ISSN 0956-9979 13666614 JULY 2002

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

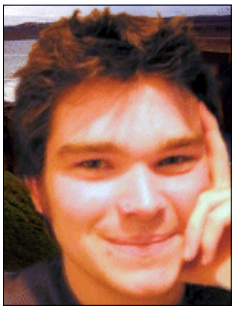
• **Mission invisible:** Stealth techniques may be old news in the virus arena, but what is new is the use of stealth on *Windows 9x*. Péter Ször traces the tracks of W95/Sma. See p.12.

• **Child's play:** Having scaled the virus prevalence tables with a hop, skip and a jump, Klez demonstrates that the combination of an old exploit with a little social engineering can still wreak havoc. Peter Ferrie investigates Klez, the new social disease. See p.8.

• **The time is nigh …** Andrew Lee explains why an effective hoax may be as damaging as a mass-mailed fast-burning virus and questions whether we should begin to classify hoaxes as malware. See p.16.

CONTENTS

COMMENT



“ Games consoles are about as ‘point-and-drool’ as it gets ”

Playing with Fire: Security on the Game

Microsoft’s *Xbox* is claimed to be ‘the most powerful game system ever built’. Screen shots from the games that have been released certainly do nothing to discredit this claim, although sales have continued to disappoint, with price wars breaking out between the manufacturers of the latest batch of consoles.

With all this powerful hardware being sold as a loss leader (there’s a lot of money to be made from the games and value-added services), memories might be stirred of hysterical reports in December 2000, which suggested that Saddam Hussein was stockpiling *Playstations* to build weapons of mass destruction.

Thankfully, that hysteria seems to have blown over now, but who’s to say that the droves of *Xboxes* and other cheap and very powerful gaming consoles aren’t providing the building blocks for devastation of an electronic ilk? It would be nice to dismiss these fears as both unfounded and unfeasible. But cast your memory back to September 1999 when security holes were discovered in *Sega*’s *Dreamcast* just days after its US launch. Who would have thought that it would be an insecure web browser at the root of the problem then, in the same way as *IE* was a root of the Nimda problem more recently?

But the security-conscious users of these gaming consoles will prove to be their saving grace, right? Sadly, games consoles are about as ‘point-and-drool’ as it gets – to rely on the users of these electronic toys to be security-aware, where even full-time sysadmins often fail, would be foolhardy to say the least.

So, what problems could possibly be caused by powerful machines running common operating systems (*Windows 2000* for the *Xbox*, *Playstation 2* is *Linux*-compatible), that are probably difficult to patch, have a broadband connection, and are being run by potentially clueless users?

We should, in theory, be relatively safe from marauding, infected *Xboxes*. Their online-gaming capabilities are supposedly restricted to a ‘safe’ *Microsoft* network dedicated to running online games. This is a nice thought, but the system can run through pre-installed broadband, meaning it has to speak TCP/IP and must be routed through somewhere; effectively one ends up with a very standardized *Windows* setup on broadband, with no option to install personal security software.

Add to this the fact that we are promised the facility to ‘download new content’ to the consoles from the *Microsoft* servers – and anyone who can masquerade as them – and you have something of a time bomb. Am I the only one who fears the phrase ‘Distributed Denial of Service’ will feature in the media a lot more frequently in the coming years?

It might be less of a daunting prospect were the *Xbox* the only threat. However, with a *Sony*-approved *Linux* distribution for *Playstation 2* – a system that can be connected to the Internet – it takes significantly less effort for a teenager with no ‘real’ computer of their own to get a *Linux* system running. No longer need teenagers concern themselves with parents demanding to know where the Start button has gone – the gaming machine is their own domain, and free for experimentation without interference. Again, when we can’t rely on some dedicated system administrators to install patches, how can we expect home users to? With ‘*Linux* for the masses’ comes, inevitably, rooting of the masses.

So where do we go from here? *Xbox Firewall* and *Bob’s Anti-Virus for Playstation*? Or will we just be left to pray that the console-creating clique patched all the holes this time? Either way, as cheap consoles gain enough power to become semi-sentient, and come online, we need to fasten our seatbelts for the roller-coaster ride of new threats that seem almost certain to arise.

Pete Sergeant, Virus Bulletin

NEWS

Quarter-byte Squaw …

This month has seen the elevation of what was thought merely to have been a minor DoS on a limited set of platforms running *Apache* to a remote-shell exploit on a large number. Worryingly, many sysadmins seem unaware both of the 'chunked encoding' bug and of their systems' vulnerability: a quick search demonstrated that at least three major AV vendors are (at the time of writing) running older versions of *Apache* that are potentially at risk. Modification of executables to contain malicious code, defaced websites, and red-faced sysadmins seem likely to become the order of the day ▮

Crying Wolf Revisited

Last month was *Network Associates*' turn to come in for a roasting over its hyping of W32/Perrun, the non-eventful proof-of-concept JPEG virus. On receipt of the virus the company was quick to distribute a press release, along with comments to the press – a move which raised the hackles of many. Although careful to acknowledge the non-severity of the virus with interjections such as 'we are not saying that this is a problem' and 'it's not serious', given the mainstream media's love of a good old-fashioned scare story – not to mention propensity for quoting out of context – one has to question the thinking behind the unleashing of this information (or lack thereof). In February 2000 (see *VB* February 2000, p.5), a concerned Vincent Gullotto of *NAI* wrote to *VB*, questioning the ethics of another AV company's marketing activities, stating: 'in all the years of pushing information, *NAI* hasn't even come close to manifesting such a barrage of unnecessary warnings.' *VB* wonders: is *NAI* attempting to make up for lost time?

But, while other AV companies may enjoy sampling the moral high ground this time around, it does appear that, for the majority, the temptation to churn out press releases at every conceivable opportunity is irresistable. Whether the result is a pile of groaningly tenuous PR 'stories' or less than helpful scare-mongering, seems to be the luck of the editor's draw ▮

VB goes to the Polls

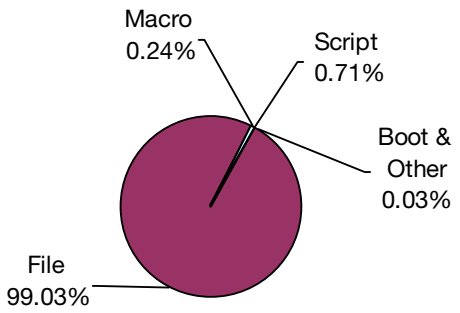
Asked 'Did David Smith get it easy?', an overwhelming 83 per cent of poll respondents on the *VB* website concluded that, yes, the Melissa author's sentencing was too light. Comments ranged from those suggesting that the focus of blame should lie with the manufacturers of vulnerable software, to those pondering whether quicker sentencing might have reduced the prevalence of mass-mailers over the last three years. Watch out for more opportunities to air your views at http://www.virusbtn.com/ ▮

Prevalence Table – May 2002

Virus	Type	Incidents	Reports
Win32/Klez	File	8344	79.65%
Win32/Magistr	File	634	6.05%
Win32/Elkern	File	286	2.73%
Win32/SirCam	File	254	2.42%
Win32/Hybris	File	220	2.10%
Win32/BadTrans	File	201	1.92%
Win32/Onamu	File	141	1.35%
Win32/Nimda	File	71	0.68%
Win32/Gibe	File	58	0.55%
Haptime	Script	34	0.32%
Win32/MTX	File	25	0.24%
Win32/Fbound	File	21	0.20%
Win32/Funlove	File	21	0.20%
Win95/CIH	File	20	0.19%
Win32/Yaha	File	14	0.13%
LoveLetter	Script	12	0.11%
Netlog	Script	11	0.11%
Win32/Aliz	File	11	0.11%
Kak	Script	9	0.09%
Marker	Macro	7	0.07%
Win32/Mylife	File	7	0.07%
Win32/QAZ	File	7	0.07%
Win32/Pretty	File	5	0.05%
Others [1]		37	0.60%
Total		10476	100%

[1] The Prevalence Table includes a total of 63 reports across 37 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

Setting the Record Straight

I am The Mental Driller, member of the 29A virus-writing group. My reason for writing is that I wish to deny publicly that the virus 'Simile', or 'Etap' (although, originally, I named it 'MetaPHOR'), was written by any anti-virus company.

Certain reactionary people read *Symantec*'s analysis of the virus and made their own malicious misinterpretation of a sentence that was intended by *Symantec* to calm user fears: 'So far Symantec has not received any submissions of this virus from customers.' The sentence means exactly what it says. Period. It cannot be interpreted in other way. The anti-virus company is not the author of the virus, *I* am the author. They have the virus in their hands because I sent it to them and, since I haven't spread the virus, I do not expect it to appear In the Wild (unless any unscrupulous person unleashes it).

Anti-virus companies do not make viruses. That urban legend is kept alive by people who, unable to achieve recognition by other means, make false statements in an attempt to attract attention to themselves and claim their 'five minutes of fame'. The only result these unfounded rumours achieve is that some users stop trusting in the protection offered by the anti-virus company – protection that may not be sufficiently robust or necessary for those who have in-depth knowledge of the subject, but which does serve well for the average user who doesn't know what PE format is, or how a virus works internally.

AV companies have no need to write viruses, there are plenty of people who create them, without financial gain. If it were true that AV companies needed to create viruses, I would have received offers inviting me to program viruses for them! I categorically deny this to be the case.

Leaving aside the cheap sensationalism generated by those who dare to pass opinion on a subject about which they know next to nothing, the worse offenders are the wannabe 'experts' who support these individuals, confirming rumours as if they were true, despite being as unqualified to do so as those who made the claims in the first place.

I get annoyed about the lies that some circles try to spread. I am not writing this to draw attention to myself, nor in an attempt to avoid problems for myself: I write because I'm fed up with the 'kiddies' who try to attract attention at the expense of the credibility of others.

Perhaps all this discussion has come to light due to the fact that the Simile virus is capable of infecting *Linux*, and this unsettles some members of the *Linux*-using community. Perhaps, instead of reacting moderately and with common sense, these people resort to fallacies and accusations, since 'a *Linux* virus cannot exist!', as some users of this magnific operating system affirm fanatically.

The fact that I managed to create a *Linux* virus in so little time (barely two weeks, including time spent learning about the system and its executable formats) indicates that all isn't as wonderful as they claim – and denying the evidence doesn't make the evidence disappear.

I hope my words are not lost in the wind and help to palliate that 'culture of the rumour' that too many people practise on the Internet.

The Mental Driller / 29A

Concerning Brand Names

On receipt of the June 2002 issue of *Virus Bulletin*, we learned that you tested the *Leprechaun VirusBUSTER II* product as part of the *Windows XP Professional* comparative review [see *VB*, June 2002, p.16]. Unfortunately, the article contains some misunderstandings concerning the relationship between *Leprechaun Software*'s *VirusBUSTER II* and *VirusBuster Ltd*'s *VirusBuster*.

As you can see, the typographical setting itself is different – a fact that the typesetting of the article ignored. The reason is that *VirusBUSTER II* is not a *VirusBuster* version. *Leprechaun Software* and *VirusBuster Ltd* have been cooperating since July 2001. As a consequence, *VirusBUSTER II* is developed on the Hungarian *VirusBuster* engine, but *VirusBUSTER II* itself is not our product. The cause of the misunderstanding is possibly the similar (but not identical) brand name.

Naturally, we always supply *Leprechaun* with the latest version of our scan engine, therefore, we do not understand the huge difference between the test results of the two products. We have already initiated consultation with *Leprechaun* in an attempt to resolve this matter.

We have published an official *VirusBuster* press release concerning this issue on our website. Please see http://www.vbuster.hu/.

Péter Agócs
VirusBuster
Hungary

VB Replies

We are happy to confirm that the two products, although sharing the same underlying engine, are understood by *VB* to be the products of two distinct companies. Following

past enquiries from readers, however, we felt it necessary to stress that the link between the two products runs deeper than simply the product names. We acknowledge that the typographical setting of the *Leprechaun* product was somewhat misleading and apologise for any confusion that arose as a result.

Matt Ham
VB Technical Consultant

Why Virus Writers Win (and how we can stop them)

There is a scene in the second *Godfather* movie in which, on a visit to Cuba, Al Pacino's character Don Corleone witnesses a rebel blowing himself up rather than face capture. He realises that the rebels will never be stopped as they are motivated by a cause that is not financial. A comparison can be made with today's virus writer – admittedly, virus writers do not tend to take such drastic action, but they are more committed to writing viruses than you are to keeping your organization virus-free.

Why is this? There are a number of reasons:

1. You simply do not have the commitment level of the virus writer who sacrifices his/her time and energy to write viruses that satisfy his/her own, varied motives, all without financial gain.
2. You are doing the best job you can for the money, but at the end of the day, it's one of the many hats you have to wear and you can't keep your eye on everything …

Unfortunately, the result is that your organization is vulnerable to virus writers who have the time and motivation to cause as much disruption as possible. Your messaging or desktop infrastructure is in pieces and you have the thankless task of cleaning up the mess.

To add to this, the virus writer is only one of three challenges you have to face daily: the virus writer, over whom you have no control, internal users who unwittingly spread viruses, and the management team who are conscious of cost and focused on investments that add value to their business.

You have no real control over the virus writer, nor, indeed, internal users, as many businesses have discovered – and often the internal challenges are linked to persuasive ability: how do you convince a team that has already invested in AV technology to review the situation regularly and, if needed, invest in new solutions?

One approach is to steer AV purchasing decisions away from the typical three-step process:

1. Which anti-virus products have we heard of?
2. Which is the cheapest?
3. We'll take it! Plus all the anti-virus tools you have – at the end of your sales quarter, for the best price.

This thinking does not bode well for a successful, long-term anti-virus strategy. The trick is to protect the organization comprehensively, not partially. Reliance on the product of a single anti-virus vendor for protection against a wide variety of threats can be the quickest way to allow viruses to breach defences – there is a single point of failure.

After every major virus incident in recent years, the majority of new anti-virus sales were to organizations that wanted another tier of anti-virus technology, in addition to the software they were using already.

Typically, cheque holders take an interest in new anti-virus solutions only *after* a virus incident has caused significant amounts of disruption, which forces a repetitive cycle of events. As a result, customers tend to pay for new AV software twice. First for the virus clean up costs and again with payment for the full licence of the evaluation software they used to clean up the mess in the first place.

Furthermore, AV solutions purchased out of pure necessity are quite often far from ideal. Many customers have good products, but with only some of their functionality implemented due to the haste with which they were installed – when stress levels are high, users are vocal and a solution is needed quickly.

Managing an anti-virus strategy this way can mean that the virus writer and the anti-virus software vendor win; you, the customer, pay twice.

What can be done? Businesses must continuously review their anti-virus strategies and focus on reducing the odds of viruses breaching defences, rather than waiting for virus signature files to become available. New technologies that aid messaging and collaboration should be evaluated for their potential to spread viral code in advance of deployment. This can prove a difficult task when different groups in the IT decision-making structure adopt new solutions without consulting the IT security group.

Keeping security at the top of the agenda is one way to ensure that your internal users are aware and educated – there is no doubt that this is a challenge – however, history has shown that anti-virus strategies become much more important when they are proven to have flaws.

Finally, if you are reading this, you are obviously committed to virus prevention. Just bear in mind that your real opponent may not be a faceless virus writer but someone you know and whom you can influence.

James Clifford
Sybari
UK

Patchy Apache

The initial assumption after its discovery was that the *Apache* 'chunked encoding' bug was harmless, and much was made of this 'fact' – how lucky the world was that the

open-source poster-child *Apache* was still very secure, and that the bug was able to cause only minor damage and on a small subset of platforms.

Thus, it will have come as a surprise to a lot of people when Gobbles Security posted a functional 'proof-of-concept' program to exploit the bug on *OpenBSD*, and give the executor a remote shell.

While some may question the ethics of Gobbles and his team, it has become apparent that there is a real-world threat here. As Gobbles claims to have versions that work on *Linux*, *FreeBSD* and *Solaris*, it seems likely we'll have a spate of defacements of well-known websites with lousy admins in short order.

Unsurprisingly, it didn't take long for talk to start of a 'Code Red equivalent' for *Apache*. At the time of writing, *Security Focus* are on ThreatCon 3 – much the same as when Nimda started to spread. However, the idea of this bug being exploited to create a viable and fast-spreading worm dissolves under closer scrutiny.

First, the bug is hard to exploit. Gobbles claims it took him and his team about two months to achieve working versions for the four operating systems he has mentioned.

Furthermore, he says that each works in a different way, taking advantage of a 'peculiarity' in each operating system. Again, certainly for the *OpenBSD* version, you either need to know exactly which versions of *OpenBSD* and *Apache* are being run, or tap your foot for two hours or so as the brute-forcing takes place. Multiply that figure by four if you don't know the OS, and we're talking about potentially eight hours work in order to infect a machine, compared with a couple of seconds for Code Red.

So how about TCP/IP fingerprinting to discover a remote host? At the best of times this is difficult to do with the required degree of accuracy – if you don't have root, and can't bind raw sockets (as one would imagine is the case for a worm), TCP/IP fingerprinting is out. Exploiting the bug will give you an account with the same privileges as *Apache*, and that tends to be not very many. You could resort to banner-checking, but who knows how accurate that'll be? Each machine on which the OS and *Apache* version cannot be seen straightaway represents the need for a significant investment of time in order for a potential worm to infect. Expect no Warhol-esque spreading …

Another factor in Code Red's favour was the number of people who were unaware of the fact they were running *IIS*, and consequently unaware that any patching was needed on their machines. It's arguably more difficult to install *Apache* on a machine unintentionally – most *Linux* distros give you at the very least a choice between server or workstation at install time, and most make you install *Apache* explicitly if you want it. Even if you have installed it, all but the most clueless user should realise pretty quickly, the relatively common 'ps -aux' showing many 'mysterious' *Apache* children running merrily amongst the daemons and zombies.

Let's assume the worst happens: a worm is written and spread. Upon infecting the machine, the worm can quite possibly modify pages on the site, perhaps dropping an HTML infector, like Nimda, or perhaps adding pseudo-political messages to the site by way of defacement. But what then?

Usually, *Apache* is run as an unprivileged user for exactly this reason – if exploited, all that's rendered is an unprivileged account. Viruses that are dropped will have difficulty infecting system files they don't have write permissions on. Dangerous, certainly, but not the end of the world.

To gain root and do really nasty things to the veritable patchy quilt of possible setups on which people run *Apache*, a worm would have to have a knowledge base approaching the self-aware. And who wants to run the risk of Arnold Schwarzenegger travelling back in time to kill them because they created sentient software? Time will tell.

Name and address withheld

Fair Comment?

VB wants to hear your views – whether it's a response to an article in the magazine or an opinion on the AV industry, get it off your chest and email editor@virusbtn.com.

VIRUS ANALYSIS 1

Polymorphism comes to Unix

Taras Malivanchuk
Computer Associates, Israel

Infectors of ELF binaries have been around for a long while. However, although this file format has seen simple encrypted viruses (e.g. Linux/Mandragore.666), until now there have been no polymorphic viruses that infect ELF format executable files. The first, Linux/Etap.1C (aka Simile), appeared recently. Not only is this the first polymorphic virus to infect ELF binaries, but it is also the first EPO (Entry Point Obscuring) virus to infect this format.

A previous version of this virus was described in the May 2002 issue of *Virus Bulletin* (see *VB*, May 2002, p.4). The virus is extremely complex and is written completely in assembly language, with an overall source size of 700 Kb; despite this, the virus can be detected relatively easily.

The virus is a direct infector that searches for files, then examines those with EXE and SCR extensions to determine whether they are in PE executable format, and others to determine whether they are in ELF executable format. Under *Linux*, the virus also checks whether the file is marked as executable.

When the virus is executed on a *Windows* platform, it uses GetModuleHandle or LoadLibrary and GetProcAddress – both ASCII and widechar variants – which have been imported by the host. If the host does not have these imports, it will not be infected. Next, the virus imports the necessary functions. Under *Linux*, it uses the int 80h syscall interface. The virus examines the file format and maps the file for further analysis.

When the virus starts processing the file, it checks for an infection mark – three random bits in dword at offset 24h in the ELF header (e_flags), which is unused in x86 executables – and checks whether the file is x86 executable.

Then the virus searches for a place to put its decryptor and body. The ELF file is analysed in two parallel views: execution view, which is represented as an array of segments, and link view, represented as an array of sections. The sections never overlap and are sorted by offset.

Usually the last sections in the list, such as the string table, notes etc., are not loaded into memory, and the last section to be loaded into memory is ‘ .bss’. The segments may overlap, but usually there are two non-overlapping program segments that include the rest of the loadable segments. If the file does not have this segment and section topology, it is not infected.

The virus searches through segment headers to find two main segments and reserves space for itself in the last segment, increasing its physical and virtual size. The virus creates an additional section after the .bss, so that the virus section becomes the last loadable section in the file. In order to create the entry for this section in the section header table, the virus moves the section header table to the end of the file.

Then the virus creates a random name for its section, starting with dot and containing up to seven lowercase letters, and places it into the string table – this is not difficult because the string table is the last section in the pre-infected file. This step is unnecessary because it is possible to take any name from the string table. When the place for the virus is ready, a decryptor is generated and placed precisely at the beginning of the section created by the virus so that the entry point will point here. The maximum size of the decryptor is 1000h but the encrypted virus body is always located at offset 1000h from the beginning of the decryptor, leaving some zero padding between the decryptor and virus body.

The decryptor is polymorphic, but works in a straightforward manner and is easy to analyse. In the same way as a decryptor in a Win32 infected file, it allocates as much as 3.4 Mb memory, copies the virus body to here, decrypts it, calls the virus, then frees the memory and executes exit(). This is sufficient to enable detection of the virus. The allocation is made using the malloc() function imported by the host program, and if it has not been imported, the file is not infected. It is interesting that the virus writer does not use the int 80h interface here.

Then the virus inserts EPO entry points: it locates calls to exit() and replaces them with jump or call near (E9 or E8) to the decryptor – exactly to the beginning of the new section created by the virus. If exit() is not imported, the file is not infected.

So we see that the obvious weaknesses of the infection site, EPO and decryptor implementation render the permutation engine worthless. Restoration of infected files is achieved easily, by replacing the calls to the virus by calls to exit(). Additionally, the virus corrupts some files during infection.

Linux/Etap.1C

Aliases:	{W32, Linux}/Simile.D.
Infects:	Win32 PE files, *Linux* ELF binaries.
Payload:	‘Free Palestine!’ is written to STDOUT on 14 May, and ‘MetaPHOR v1C by The Mental Driller/29A’ on 17 March.

VIRUS ANALYSIS 2

Raised Hacklez

Peter Ferrie
Symantec Security Response, Australia

When W32/Klez first appeared, it seemed like just another mass mailer of little note, but its later variants have spread so widely and rapidly that the Klez family has generated more interest. At the time of writing, there are 12 known variants of Klez. Despite the speed with which anti-virus developers released detection updates, despite the fact that some anti-virus products detected the later variants even before they were released, and despite its destructive payload, Klez remains a problem that shows no sign of being resolved in the near future.

The Buck Stops Here

All known variants of Klez begin with a call to a function in a dll that does not exist in *Windows 95* (*Windows 95* does not support Winsock 2.0), and import a function that does not exist in another dll in *Windows NT* (*Windows NT* does not support the Toolhelp interface). Therefore, Klez cannot replicate under either of these platforms. However, this has clearly proved to be not much of a limiting factor.

Copy Me, I want to Travel!

Klez creates several threads in order to perform a number of functions simultaneously. The first thread terminates certain applications – anti-virus and firewall programs – based on application name. Later variants also search for strings in process memory, and will terminate processes and delete files that contain them. Initially, this search was restricted to viruses, such as Nimda and SirCam, but the feature was extended later to include searching for anti-virus programs and the deletion of Registry keys.

Under *Windows 98*/*ME*, Klez writes itself to the Registry key 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run'. Early variants use 'krn132' as the value name and data, while the later ones use a name that begins with 'Wink' followed by two to four random letters. The result is that *Windows* launches the Klez file whenever the computer is booted. The thread is set to execute ten times per second, making it impossible to run on-demand anti-virus software for long enough to remove the virus. Later variants of Klez also run this routine (thousands of times) as part of the payload, but in such a way that processes will be terminated and files deleted, regardless of their content.

Dropping your Bundle

The second thread drops and runs the W32/Elkern virus, which is carried as a compressed file within the body of Klez. Klez decompresses this file and drops it using a random filename in the %temp% directory. Once execution of the file is complete, Klez will delete it.

When Elkern is run, it copies itself to the %system% directory, using a filename whose suffix depends on the platform upon which it is executed. Under *Windows 98*/*ME*, the filename is 'wqk.exe', and under *Windows 2000*/*XP* it is 'wqk.dll'. Klez is aware of this behaviour and under *Windows 98*/*ME* it will run wqk.exe; under *Windows 2000*/*XP*, it will load wqk.dll into its own process memory. This action will prevent the wqk file from being deleted, unless the Klez process is terminated first.

At this point, Klez copies itself to the %system% directory, using the same name as it used in the Registry. Under *Windows 98*/*ME*, Klez will then write itself to the Registry again, as above. If the RegisterServiceProcess() API exists, Klez will use this to register itself as a service, which removes it from the Task List. If the copied file is not running already, Klez will run it now.

Under *Windows 2000*/*XP*, Klez determines whether it is running as a service, using a rather complicated-looking method involving tokens and security IDs. If it is not running as a service, Klez will create a service, using the same name as it used in the Registry. If the copied file is not running, Klez will run it now, as a service. The most recent variants assign random values for the copied file's date and time, in an attempt to conceal its presence within sorted directory lists that would otherwise show the Klez file as the file created or modified most recently. Those variants that infect files will decompress and run the host file at this time.

Little Black Book

The third thread is used to send email. Klez uses the *Windows* Address Book as a source of email addresses, and assumes that the address book can be located from the Registry key 'HKCU\Software\Microsoft\WAB\WAB4\Wab File Name'.

This key is created by email products such as *Outlook* and *Outlook Express*, although others, such as *Exchange* and *Windows Messaging*, store the location of the address book using a different Registry key. Later variants of Klez also search for *ICQ* data files, which begin with 'db' or are called 'user.db'.

If it finds either the address book or an *ICQ* data file, Klez reads from there as many addresses as will fit into its 4 Kb buffer. Klez has two routines for reading email addresses. One supports the ANSI character encoding for addresses, as used on *Windows 98*/*ME* by *Outlook Express*, *ICQ*, and *Outlook* prior to *Outlook 2002*. The other routine supports

the Unicode character encoding for addresses, as used by all versions of *Outlook* and *Outlook Express* on *Windows 2000/XP*, and *Outlook 2002* on all platforms. However, Klez stores the Unicode addresses in ANSI format. Klez considers an email address valid if it contains one '@', followed by at least two characters, then a dot ('.'). Later variants of Klez check that there are additional characters following the dot.

If early variants find fewer than ten email addresses, Klez generates a random number of addresses (between 20 and 29), each containing three to nine letters, with the domain selected randomly from yahoo.com, hotmail.com and sina.com.

For each email address in the list, all known variants will select another address at random and use this as the 'From:' address. Klez prepends 'smtp' to the domain name in the 'From:' address, and attempts to connect to this server. If the connection is unsuccessful, Klez will enumerate the entries in 'HKCU\Software\Microsoft\Internet Account Manager\Accounts\' to find SMTP information and attempt to connect to the server that is found. If the connection is successful, Klez will attempt to send itself to the chosen email address. Thus, person A's computer will be used to send an email to person B, but the email will appear to have come from person C.

Get the Message

The early variants of Klez choose the subject of the email randomly from the following:

Hi
Hello
How are you?
Can you help me?
We want peace
Where will you go?
Congratulations!!!
Don't cry
Look at the pretty
Some advice on your shortcoming
Free XXX Pictures
A free hot porn site
Why don't you reply to me?
How about have dinner with me together?
Never kiss a stranger

Later variants use more complex subject generation. With a one in three chance, the current date will be checked against a list of specific dates. If the dates match, then the subject will begin with 'Happy' or 'Have a'. With another one in three chance (or always if the subject begins with 'Have a'), these variants will select one of the following words: new, funny, nice, humour, excite, good, powful [*sic*], followed by the name which relates to the date. The dates and names are as follows:

1 January: New year
6 January: Epiphany
2 February: Candlemas
14 February: Saint Valentine's Day
25 March: Lady Day
1 April: April Fools' Day
15 August: Assumption
31 October: Allhallowmas
2 November: All Souls'Day
25 December: Christmas

So the result may be, for example, 'Have a powful Candelmas', 'Happy Christmas', or 'Happy excite Lady Day'.

If no subject has been chosen yet, it may be left completely blank or begin with one of the following texts:

Undeliverable mail–
Returned mail–
Hi,
Hello,
Re:
Fw:

followed by any one of:

how are you
let's be friends
darling
don't drink too much
so cool a flash,enjoy it
your password
honey
some questions
please try again
welcome to my hometown
the Garden of Eden
introduction on ADSL
meeting notice
questionnaire
congratulations
sos!
japanese girl VS playboy
look,my beautiful girl friend
eager to see you
spice girls' vocal concert
japanese lass' sexy pictures

Alternatively, the subject may be a random string from a data file, or chosen from this list:

a %s %s game

a %s %s tool

a %s %s website

Each %s is replaced by a word from the adjective list described previously (new, funny, etc.).

Other subjects include 'a %s %s patch', where the first %s is replaced by an adjective, and the second by 'WinXP' or 'IE 6.0', and '%s removal tools', where %s is replaced by 'W32.Elkern' or 'W32.Klez'. The most recent variants of Klez may use the subject 'Worm Klez.E immunity'.

The email body of early variants contains a message which appears to be from the virus author, describing his financial situation. However, this message is not visible if the email is viewed in HTML format.

The message body in later variants remains empty unless the subject is one of those that contains a %s, the subject refers to Klez.E immunity,
or the subject begins with 'Undeliverable mail–' or 'Returned mail–'.

If the subject refers to an undeliverable or returned mail, the message body will read 'The following mail can't be sent to %s:', where %s is the random 'From:' email address, followed by 'The %s is the original mail', where %s is 'attachment' or 'file'.

If the subject refers to a removal tool, the message body will contain one of the following names: Symantec, Mcafee, F-Secure, Sophos, Trendmicro, or Kaspersky, followed by 'give you the %s removal tools', where %s is 'W32.Elkern' or 'W32.Klez'. The following line is either 'W32.Elkern is a %s dangerous virus that can infect on Win98/Me/2000/XP' or 'W32.Klez is a %s dangerous virus that can spread through email', where %s is 'very' or 'special'. This is followed by 'For more information,please visit http://www.%s.com', where %s is the name of the anti-virus vendor from the list above. The filename of the attachment is 'setup.exe' or 'install.exe'.

For emails whose subjects refer to a game, tool, or website, the message body will begin 'This is', then repeat the subject, followed by 'I %s you would %s it.', where the first %s is replaced by 'wish', 'hope' or 'expect', and the second is replaced by 'enjoy' or 'like'. The message may begin with 'Hi' or 'Hello'.

If the subject refers to a game, the message will continue with 'This game is my first work. You're the first player' and the name of the attachment will be one of the following: 'setup', 'install', 'demo', 'snoopy', 'picacu', 'kitty', 'play', 'rock'.

If the subject refers to Klez.E immunity, then the message body will read:

```
'Klez.E is the most common world-wide
spreading worm.It's very dangerous by
corrupting your files.

Because of its very smart stealth and
anti-anti-virus technic,most common AV
software can't detect or clean it.

We developed this free immunity tool to
defeat the malicious virus.

You only need to run this tool once,and
then Klez will never come into your PC.

NOTE: Because this tool acts as a fake
Klez to fool the real worm,some AV moni-
tor maybe cry when you run it.

If so,Ignore the warning,and select 'con-
tinue'

If you have any question,please
<a href=mailto:%s>mail to me</a>.'
```

where %s is replaced by the random 'From:' address.

If the subject does not refer to a removal tool, the suffix of the attachment will be .exe, .scr, .pif, or .bat.

Repeat as Required

In addition to the message body, there is HTML code that exploits a vulnerability in unpatched *Outlook* and *Outlook Express*. There are two parts to this vulnerability. The first is that applications can be launched automatically from an IFrame, without any prompt. The second part is that the MIME content type is trusted explicitly, without reference to the filename (and thus the file content), yet the launching of the application is performed by a part of *Windows* that does examine the filename. The result is that certain multimedia content types can be used to launch *Windows* executable files.

Klez uses this vulnerability to launch itself automatically. In addition to the viral attachment, if a data file is found (see below), there is a 50 to 100 per cent chance (depending on the variant) that Klez will attach this file to the email as well.

Once the email has been sent, the recipient's address is added to a master list. If the email connection proved unsuccessful, Klez will try five other addresses, selected at random from the email list. If the connection is still unsuccessful, Klez will try five addresses chosen randomly from its master list. Later variants of Klez also carry a list of open relays and will attempt to connect to one chosen at random from this list.

Regardless of whether the email has been sent successfully, the master list is updated each time, by removing the first entry and shifting the others up. This thread is executed repeatedly, at intervals of between 10 minutes and five hours, depending on the variant.

Share and Enjoy

The fourth thread that is created searches for open shares on the local area network. Klez will copy itself once to each shared directory. If a data file is found (see below), then Klez will use its filename without extension as its base filename, otherwise it will generate a random name, consisting of two to five letters followed by a number. To this will be attached two suffixes. The first is chosen randomly from txt, htm, doc, jpg, bmp and xls. The second is always '.exe'.

Later variants of Klez can also drop RAR archives, containing only the Klez file, into these directories. Under *Windows 2000/XP*, Klez will launch the file as a service on the remote computer. The more recent variants will also connect to the remote Registry and add an entry to the 'HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce' key to run the copied .exe file when the remote computer is rebooted. The danger of this action is clear: not only can Klez send an executable to another computer, but it can cause the file to execute, too. This thread is run repeatedly, at intervals of between 30 minutes and eight hours, depending on the variant.

Here's One I Made Earlier

Klez searches for data files to use as filenames on remote computers and as decoy attachments in emails. Later variants of Klez also look in these files for email addresses. Klez searches for files by creating 26 threads, one for each possible drive letter. On hard drives and network drives, Klez searches for files whose extension is in the following list: txt, htm, html, wab, doc, xls, jpg, cpp, c, pas, mpg, mpeg, bak, mp3.

Although only one filename is saved, the use of threads raises the possibility that the email and network routines will see different filenames. Later variants of Klez also delete anti-virus integrity database files whenever they are found, and replace RAR archives with new archives containing only the Klez file. Early variants of Klez execute these threads only once, but later variants execute them repeatedly, at intervals of between 30 minutes and eight hours, depending on the variant.

We're the Infectious Grooves

Later variants of Klez infect files. Klez enumerates the entries in the 'HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths' key, then searches all directories whose name contains the string 'Program'.

A file is a candidate for infection if it is not infected already, is not protected by the System File Protection that exists in *Windows 98/ME/2000/XP*, is between 67 Kb and 3 Mb in size, and its filename does not contain EXPLORER, CMMGR, msimn, icwconn, or winzip.

Klez infects by copying the original file to a random filename and replacing the original contents with itself. The size of the infected file is not altered, which is why infectable files must be at least as large as Klez itself. The copied file is then compressed using a Run-Length Encoding algorithm, and the file attributes are set to Hidden, System, and Read-Only, to hide it from the default directory listings. This thread is executed once every hour.

The Wait is Over

It is at this point that Klez checks whether its payload should trigger. The payload activates during odd-numbered months (January, March, May, July, September and November). The early variants of Klez activate on the 13th of those months, while later variants activate on the 6th of those months.

When the payload activates, Klez creates 26 threads, one for each possible drive letter. If the drive is a hard drive or a network drive, Klez will search for files in all directories, and overwrite the files entirely with data from memory.

For early variants of Klez under *Windows 98/ME*, the data will be random, and under *Windows 2000/XP*, the data will be the characters 'BA AD F0 0D' (a *Windows* default value); for later variants of Klez, the data will be zeroes for all platforms.

Early variants affect all files in this way, but later variants of Klez affect all files only during January and July. At other times, only files with extensions that are included in the data file list above are affected. Early variants of Klez execute this thread every 30 minutes, but later variants execute it only once.

Conclusion

What conclusions can be drawn from Klez? It seems that the combination of an old exploit with social engineering can convince an enormous number of people to open attachments from people they don't know.

Klez does not present new ways to replicate, only new words to entice people to help it replicate. Some computers experience symptoms and their owners seek a cure, while others do nothing and allow the replication to continue.

It will take a change in people's behaviour to halt the spread of viruses like Klez... Klez: the new social disease.

W32/Klez	
Type:	Memory-resident, direct-action companion infector.
Infects:	*Windows* Portable Executable files.
Payload:	Date-triggered file deletion.
Removal:	Delete infected files and restore them from backup.

VIRUS ANALYSIS 3

Stealth Survival

Péter Ször
Symantec Security Response, USA

Virus writers have always attempted to challenge users, virus researchers and virus scanners alike. In fact, some of today's viral techniques, such as anti-heuristics and anti-emulation, were invented by virus writers in response to virus scanners becoming increasingly powerful. However, stealth viruses appeared very early on in the history of malware.

In fact, one of the first known PC viruses, Brain (a boot virus), was a stealth virus. Brain showed the original boot sector whenever an infected sector was accessed and the virus was active in memory, hooking the disk interrupt handling. This was in the golden days when Dr Alan Solomon was almost driven to distraction in his attempt to figure out what exactly was going on.

It was not long before stealth techniques appeared in DOS file infectors too – this method enabled the virus to remain unnoticed for long enough to replicate. In the DOS days, users would memorize the sizes of system files in an attempt to apply their very own form of integrity checking. Knowing the original size of files such as command.com was half-way to the success of detecting an ongoing infection.

There are a number of different names for the stealth technique, according to how difficult it is to detect the virus and what kind of method it uses. 'Semi-stealth' viruses conceal the changes in file size, but the original content of the infected object remains visible via regular access. 'Read stealth' is a technique where the original content of the object is simulated, usually by altering seek and read functions. The technique is described as 'full stealth' when all possible access is virtualized to a certain infected object. Finally, the highly sophisticated 'hardware-level stealth' was used by the Russian virus Strange, which hooks INT 0D, which corresponds to IRQ 5. In all of the stealth techniques the virus code must be resident in memory.

Whatever Happened to Win32 Stealth Viruses?

I don't know of any *Windows* users who would bother to memorize the size of notepad.exe or other such files. Who would pay attention to this information these days, when applications are typically so huge that they barely fit on a diskette? Evidently this is the primary reason why there have been only a few attempts so far to develop stealth viruses on 32-bit *Windows* systems.

Nevertheless, one of the first known Win32 viruses, Win32/Cabanas, used a semi-stealth technique (or so-called directory stealth). Cabanas tried to hook the host's access to APIs which returned file size information in order to hide file size changes. Cabanas certainly did not use a very advanced technique, but we can say that this was the first step in stealth on *Windows*.

There have been no other major attempts to write semi-stealth viruses since Cabanas was released. The fact is that semi-stealth does not make sense from the point of view of virus replication on Win32.

Read Stealth in W95/Sma

A few weeks ago a new virus, W95/Sma, was sent to me by a fellow anti-virus researcher. The virus was marked as 'interesting polymorphic', so I was very keen to look into it immediately. After a while I figured out why it seemed impossible to replicate the virus.

At first I believed that I had replicated the virus. I knew this because the size of my goat files changed on the hard disk. Next, I copied the infected files to a diskette in order to move them over to my virus research machine. To my surprise I found I had copied clean files! I repeated the procedure twice more until I started to suspect that something was just not right with W95/Sma.

Using my Windows Commander tool I looked into the file on the infection machine. Sure enough, there was nothing in the file. In fact, the file was larger, but there appeared to be nothing appended to it. Then I accessed the file on the diskette one more time. Sure enough, the size of the file changed on the diskette too. I quickly moved to my virus research system and, finally, found W95/Sma in there. Gotcha!

W95/Sma is the first known working *Windows 95* stealth virus. Previously W95/Zerg attempted a stealth technique, but the virus crashed so quickly that it did not prove difficult to detect at all.

W95/Sma does work, but there is a minor bug in its technique. The virus attempts to set the second field of the infected PE files to 4 in order to hide its size in specially marked infected files. However, the virus clears the bit that it wants to detect before it compares and thus it will always fail to hide the size change. Infected files will appear 4 KB longer.

Decrypt Slowly

W95/Sma is an oligomorphic virus. The virus does change the main entry point in infected applications. However, it places its first decryptor into a cave of the code section itself. Such a cave usually exists in PE files and the first decryptor is only a few bytes.

The first decryptor will decrypt the main virus body in the last section and jump to that point. However, there is another decryption layer, which will decrypt the virus body little by little. Finally, the decrypted virus body is executed. On returning from this call, the original main entry point will be executed.

Sma in Kernel Land

The virus uses a call gate mechanism. It modifies the Global Descriptor Table (GDT) to create its own descriptor entry 1F0h. From there on, Sma can execute its code in kernel mode. This will be vital, since the virus wants to hook the file system.

First, W95/Sma allocates kernel heap memory, then it copies itself there and jumps to that location. It hooks the file system so it will be able to see file access functions (SEARCH, OPEN, SEEK, etc.). In addition, the virus hooks the TCPIP service, which it uses maliciously (see later). Eventually, control is returned to the original host.

The virus will ensure that it hooks the file system only once. Should the file system already be hooked by Sma, the previously reserved heap block will be freed properly.

Fast Infection of PE Files

On each file open the virus will check the file content and attempt to infect PE files regardless of their attributes (it clears attributes) or their extension. It modifies the pointer to the symbol table entry in the PE header, giving it a non-zero value, in order to mark the infection internally. The entry point will be modified to point to the first decryptor in the code section.

The decryptor of the virus is oligomorphic (it does not change very much, but enough to break pattern matches). The size of the image field is also altered and the virus body is placed into the last section. The last section is marked as writeable and the encrypted virus is placed into it with the additional decryption layer on top. Infected files will grow 4 KB in size. (The change remains visible in this release of the virus.)

The virus structure looks very advanced internally. This is due to the modularity of the code. The virus does not patch the usual CD 20 xx xx xx xx patterns all the time as first-generation kernel mode viruses do on *Windows 9x*. Instead, it uses a single function and calls that with passed function IDs according to its needs. This simplifies the coding, and makes analysis of the virus code more difficult.

Stealth

When an infected PE file that has been marked as infected is opened, the virus virtualizes the file content. In fact, it hides the changes so well that it is very difficult to see any at all. The virus assumes zeroes for all the places where unknown data was placed. Otherwise, original content is returned for all previously modified fields of the PE headers and section headers.

Evidently, if there were no bug in the code the virus would be totally hidden from the eye. So, is it hidden? Yes and no. The virus code remains hidden from regular file _open() and _read() functions. Consequently, when an infected file is copied via such functions, the copy will, at first, appear clean from the virus.

However, we should note that W95/Sma does not hook memory mapping at all. This means that a sequence of memory mapping APIs can lead us to the proper file content! This is good news. (Although it would be relatively simple for the virus to convert to full stealth and hook such events as well.)

TCPIP

The virus hooks TCPIP in the initialization code. I had many discussions with Peter Ferrie and other researchers at *Symantec* on this matter until all the pieces fell into place and started to make sense. The virus attempts to open PORT 53357, specifying UDP, and sets up a notification request using TdiOpenAddress() and an event handler for TDI_EVENT_RECEIVE_DATAGRAM with TdiSetEvent() using kernel mode functions.

The idea is to execute code that is received via such broadcast in kernel mode. Memory is allocated for the incoming data, then _VWIN32_CreateRing0Thread() is used to run the content that is received.

Such a technique could be used for many malicious reasons, including (but not limited to) DDoS attacks as well as backdoor features. The 'NetSt0rm 1.0, G.7 (c) Smash Inc.' string in the virus suggests that this is a version 1.0 release of the code. 'G.7' could be the sub version number of the virus – it definitely appears to be an early release of the code.

Conclusion

The stealth technique has finally arrived on *Windows 9x*. The next step for the dark side will be stealth on *Windows NT/2000/XP*. We can only hope that this happens later rather than sooner.

It seems that the merging of the malicious hacker and virus writer knowledge base is continuing to produce increasingly sophisticated attacks.

W95/Sma	
Type:	Fast PE infector, oligomorphic, stealth.
Size:	4096 bytes.
Detection string:	Not possible.

CONFERENCE REPORT

It's Not Just About Viruses Any More: EICAR 2002

James M. Wolfe
Independent Researcher, USA

The first I heard about the European Institute of Computer Anti-Virus Researchers (EICAR) conferences – several years ago – was that they were strictly an academic endeavour and that much of their content was both theoretical and not particularly practical in terms of application in the real world. Therefore, when I attended the EICAR conference last year I was pleasantly surprised to discover that such bias toward the theoretical no longer exists. In fact, following my experience of last year's conference, I was quick to make my reservations as soon as the 11th Annual EICAR Conference was announced.

EICAR describes its conferences as '[combining] universities, industry and media, as well as technical, security and legal experts from civil and military government and law enforcement and privacy protection organizations for a major European forum.' All I can say is that EICAR isn't just about viruses any more.

On an AV Theme

Of course, this year's conference programme included many topics that were directly relevant to the anti-virus field. These included Jeanette Jarvis' five key components for a successful anti-virus strategy, Andrew Lee and David Harley's 'Back to the Future', which presented a 'fresh look' at malware, and Randy Abrams' presentation, 'Corporate Virus Checking'.

Randy tantalized his audience with a look at his internal (to *Microsoft Corp*) program *Scan-O-Matic*. The program allows his internal clients, with a few simple clicks, to submit virus samples for checking – once analysed the results are reported back to the user. Randy, just what will it take for you to let the rest of us get hold of the program? Can you show this article to Mr Gates and let him know that we want it?

Allan Dyer, of the Association of Anti-Virus Asia Researchers (AVAR), and Robert Vibert, CDO of the Anti-Virus Information Exchange Network (AVIEN), were on hand to discuss the activities of their respective organizations.

Security Focused

Although not specifically virus-related, I found some of the presentations in other tracks particularly interesting. There was 'Cyberterrorism and the Real World', which I enjoyed because it allowed me the opportunity to argue just how ludicrous the term 'cyberterrorism' is. Eddy Willems presented 'Towards an Early Alert System', which discussed the European system that distributes real-time public service announcements warning of new virus threats.

Rainer Fahs, Chairman of EICAR, and *Symantec*'s Vincent Weafer led an enlightening talk on 'Effective Protection of Critical IT Infrastructures'. My personal favourite, though, was 'Windows XP Sentinel Systems for Academic Research' by Dr. Larry Leibrock from the University of Texas. Dr. Leibrock is a newcomer to EICAR, but certainly made an impact both with the conference audience and with this author. I hope that EICAR continues to encourage submissions from the IT security and legal fields as well as the anti-virus regulars that we've come to expect.

See *You* Next Year …

I do have one question though: where were all of you? There were roughly 80 delegates at this year's conference – far too few in my opinion. EICAR continues to evolve – changing proactively with the times in order to give its members and conference attendees information that is both current and relevant.

Following last year's 9/11 tragedy, security has been on nearly everyone's mind. In fact, EICAR was ahead of the times last year, when its conference was dedicated partly to security issues. This year the organizers really pulled out all the stops and put on a first class programme that dealt as much with general security issues as it did with virus-related issues.

For global corporations EICAR should be a very attractive conference. The registration fee is not over-priced and, in general, EICAR picks locations that are in or near major cities, which results in lower travel costs. EICAR provides an informative programme that contains real-world information that can be used by all.

My one request is that I would like to see a North American version of the conference some time in the future. A strong North American conference – whether in lieu of or in addition to the European version – could gain EICAR much needed membership and additional funding.

Thanks to Rainer, Urs, Sarah, Eddy, Christine, and the rest of the EICAR staff for an excellent conference. I look forward to seeing all of you next year, 10–13 May, for EICAR 2003 in Copenhagen.

OPINION 1

Defence of the Realm

Paul Baccas
Sophos Anti-Virus, UK

In *A Short Course on Computer Viruses*, Fred Cohen writes about the 'Moated Wall' approach to protecting a system from viral attack. The defence-in-depth approach works because where one solution does not prevent infection, hopefully, the others will stop the virus. The only real problem with a multi-tiered approach is complacency – administrators may become blasé about the protection they have and the proverbial pride comes before the fall.

The smaller the business, the smaller the IT/IS department and the greater the workload of the person in charge of implementing the company's anti-virus strategy. In general, most people will opt for the easy solution in preference to the potentially labour-intensive ones. However, a large proportion of these people will not have looked beyond the initial problems of setting up a system and, unwittingly, will choose the more labour-intensive solution because it appears to be the easiest. In large corporations, while it is still true that people will look to the easy solution, generally more thought is given to the long-term effects.

Computer security experts, journalists and marketers use various analogies to help explain problems in terms that make it easier for their audience to understand. Most of the analogies work only at a superficial level but, if chosen well, they can provide insight into the issues at stake.

Defensive Walls of Yore

Historians have argued for some time about the purpose of defensive walls of the type built by Emperors Hadrian and Antoninus, as well, of course, as the Great Wall of China. Thought originally to have been of a purely defensive nature, more recently it has been postulated that they were built so as to restrict access to the Empire. Access was granted via manned gatehouses where imperial officials could do as was their wont – tax, monitor and, of course, stop the ravening hordes.

The computer, or computer network, does not have to go to the extremes of monumental civil engineering; access is funnelled through certain points already. As analogies go, the defensive walls of yore and the protection of a computer network may be stretching the point a little – but bear with me and I shall elucidate. I shall examine only one point of entry, the email gateway, with the AV software representing the gatehouse, and the configuration of the software the imperial official.

At the gatehouse the imperial official could, of course, let the raider(s) through to ransack the Empire. Obviously, however, this is not a feature we would desire in an email gateway.

Raiders could be refused entry forcibly – the ideal solution. Problem solved. However, not all of the officials seem to do this.

Occasionally, the officials disarm the raiders of their weapons and let them through. The raider still arrives at the doors of the peasant farmers of the Empire. Moreover, while the official may have removed the raider's sword, this may not be the only weapon with which the raider is armed – a commando training manual from the last war included details of how to kill with a matchstick! Disarming or disinfection may appear to be a good solution, but what happens when there are multiple infections? And does disinfection interfere with the detection of the other viruses?

Some officials stop the raider, then proceed to dress a young messenger up in the raider's clothes and send the messenger onward. This may be straining the analogy somewhat, but what do the peasant farmers think when they see the messenger? What is a computer user to think when they see an email with the same subject, message body and attachment name as a virus? The attachment size is not the same and its extension may be TXT but, in any SafeHex rulebook, not opening files is a prerequisite and providing the user with a text file saying, 'Product X has removed Virus Y from the mail from user Z' is hardly ideal.

Some officials do not dress the messenger in the raider's clothes but send him off on the raider's horse. The messenger still turns up at the farmer's door uninvited. What happens after a particularly good celebration of the Emperor's birthday, when the official has done his duty, and the raider attempts to sneak through disguised as a messenger? The solution of using a fixed name for the warning message is just as flawed as that of adapting the virus name.

When a group of travellers attempt entry to the Empire what is the official to do? Let the wagon the raiders brought with them through? What happens if the wagon is not all it seems? I saw my only ItW copy of VBS/Redlof.A in the 'clean' attachment sent with W32/Klez.H.

Conclusion

In my opinion, the easiest/best solution in the long run may not seem like the easiest solution in the short term. That is to quarantine all emails and parts of emails that are infected and send an inline notification to the user. Such a system will provide a large initial overhead for the administrator, most of which can be automated subsequently. However, stopping things at gateways is a great deal easier than cleaning infected machines.

OPINION 2

Memetic Mass Mailers: Time to Classify Hoaxes as Malware?

Andrew Lee
Team Anti-Virus, UK

This article is intended to provoke discussion, rather than to provide hard and fast answers; it arises after observing the statistical tracking of hoaxes (in a limited and fairly unscientific manner) over the last two years. From the trends shown by this tracking one can extrapolate that an effective hoax (which I shall define in a moment) can be as damaging as a mass-mailed fast-burning virus – and sometimes more so.

The Effective Hoax

A 'successful' or 'effective' hoax is one that works on three levels:

1. It is sufficiently attractive to draw recipients' attention to it (the subject line 'New Virus Alert' usually achieves this).
2. It spreads rapidly and widely enough, with or without modifications, to catch recipients unawares – and does so before it can be debunked.
3. It is believable enough for the recipient to deliver the payload – whether that be simply propagating the hoax further, or carrying out given instructions before spreading the message.

There are some striking similarities here with actual malware. Some of the more successful worms and viruses of recent years, such as Melissa, Loveletter, Anna (I apologise for using the populist forms of these names, but it makes for easier reading – and gives the pedants something to get their teeth into), have followed the same rules.

1. They were sufficiently attractive for people to pay attention to them: Melissa offered porn site passwords, Loveletter offered, well, love, and Anna offered pictures of a nubile sports personality. This is getting the foot in the door, and is essential for achieving a successful spread. (Let's leave the true worms out of this for a while.)
2. They spread rapidly and widely enough that many people 'contracted' them before their AV software was updated, and before alerts had been issued.
3. They were believable enough to make the recipient deliver the payload. I chose these examples specifically (rather than, for example, Klez or Badtrans.b) as they demonstrate user involvement. No software exploit was involved in these viruses – in each case, it was the recipient who delivered the payload by double-clicking on the file.

Usually, of course, a hoax requires the user to carry out its replication as well as its payload, but just as the aforementioned trio of malware delivered mass mailing as part of their payload, the successful hoax has the same result. Whether it is malware, which does the work itself, or a hoax which gets the user to do the work on its behalf is really an irrelevance, since the end result is the same.

Let's examine a couple of successful hoaxes. The technique used by each is substantially similar, with the execution being the only real difference. Rather than looking at the hoaxes from the point of view of finding out why they are hoaxes, I shall look at why they work.

Elf Bowling

First, let's look at the Elf-bowl hoax and why it was successful. (The full text of this hoax can be found at http://www.umich.edu/~vbuster/hoaxes/elfbowl.html.)

First, the hoax plays on our fears – we have all heard or read endless warnings about accepting unsolicited email attachments. Most users know they shouldn't open them, but do so anyway. In the case of Elfbowl.exe, they received it from a friend, who was sent it by a friend, who got it from who knows where, all of which adds up to a hefty uncertainty factor.

Secondly, the hoax was timely. The message appeared only a few days after the original file had been circulated, which meant that the game was still fresh in people's minds. This raised the profile of the doubt in the recipient's mind. Had the hoax message been sent a few months later, it is debatable whether anyone would have remembered the original file, and the hoax would not have had the same impact.

Finally, the message was not confirmable as a hoax for some time – no one (including the AV companies) knew for certain whether the file had been infected, or Trojanised. There was no way of knowing whether such a modified variant was out there, all that could be determined was that the *original* Elfbowl.exe was not malicious. Confusion is a wonderful vector for rumour and insinuation.

A Picnic of Teddy Bears

Now let's look at another successful and more recent hoax, the JDBGMGR.EXE hoax. (See http://www.umich.edu/~vbusters/hoaxes/jdbgmgr.html for the full details of this hoax.)

There is some discussion about whether this really is a hoax, or whether it is just well meaning misinformation – let's put that aside for one moment, and concentrate on the reasons for its success. I find it particularly surprising that this hoax was such a success, as it is almost identical to the SULFNBK.EXE hoax (with the same caveat on the use of the word hoax), which appeared almost a year earlier. So why did this one work?

First, there is a heightened awareness of malware in the media at the moment. Nimda, CodeRed, SirCam, Badtrans.b and Klez have each had many media column inches devoted to their 'Internet-destroying' properties. This heightened awareness is usually fairly undirected – in other words, there is a lot of fear, uncertainty and doubt, and no greater level of knowledge. Arguably, this leaves the less clued user open to exploitation by new hoaxes that play on these shifting fears.

Secondly, the message is cleverly worded. Well, I mean 'clever' in that it exploits natural human naïveté. Many people still believe things that are written down – newspapers being a classic example – and apply little actual thought beyond the face value of information. So, when something reads, 'This is not a hoax, I found it on my machine', combined with 'I think I may have sent you a virus', it creates a powerful rationale which the reader accepts unswervingly.

Finally, this hoax will almost always work. The fact that the file named by the hoax exists on 99 per cent of normal *Windows* installations will almost certainly fool some people. Add to that the file's unusual icon (a teddy bear of all things), and you have the makings of a great hoax.

Destructive Payloads

The aim of much malware is to deliver a payload (of course, many viruses simply replicate and have no payload). Payloads range from nuisance value, such as intermittent beeping or displaying a graphic, through mass-mailing, right up to destroying data on the infected machine.

The traditional payload of hoaxes is time wasting and increasing user anxiety about the virus threat. Replication is achieved by suckering the punter into sending it on – effectively, a simple user-assisted replication.

Hoaxes such as SULFNBK.EXE and JDBGMGR.EXE add a level of destructiveness to the payload. In these cases it is the deletion of a single system file, and the files in question are fairly irrelevant – at worst their deletion causes an inconvenience – but it would be naïve to assume that this will always be the case. What if the recipient were instructed to delete a folder, or a more important file or set of files? The possibilities are endless, and because there is little technological detection available for such hoaxes (though some products do claim to detect them), the chances are high that such hoaxes will replicate successfully and deliver their payload.

It has been argued (convincingly) that the SULFNBK.EXE and JDBGMGR.EXE hoaxes are both instances of 'well meaning misinformation'. This may be the case – certainly SULFNBK.EXE was a very commonly mailed file when W32/Magistr.a was at its peak, and there is some justification for believing that someone put together the instructions for its removal in good faith. However, this seems less likely with the JDBGMGR.EXE hoax – mainly because it appeared almost exactly one year after its earlier variant when W32/Magistr.a has long been known about. Perhaps we shall never know, but it may be wise to consider this a glimpse of a possible future trend.

The fact is that there will always be malicious (or just silly) pranksters who take great delight in knowing that their creations have caused widespread damage and/or panic. In fact, I would go so far as to say that, as more virus writers are dragged through the courts, hoaxing may become a safer way of spreading an idea.

Recently, some AV vendors have begun to provide up-to-the minute hoax metrics and alerts, in much the same way as they have traditionally done for viruses. This in itself is a double-edged sword. There has always been a certain 'respect' to be gained amongst writers for getting a creation onto the WildList or a vendor site, and this may be the same for hoaxters. But, of course, such sites and lists are also valuable (and eventually essential) tools for overworked system administrators.

Whatever the state of play at the moment, there is no doubt that the potential for damage becomes greater with each new hoax. When I first connected to the Internet (or at least its rudimentary beginnings) I could count on one hand the number of friends who had email addresses. Now it seems that everyone and their dog has (sometimes several) email addresses. This has proved rich pickings for the fast-burners like Melissa, Loveletter and Anna, but without doubt the hoaxters have had their fun too. Hoaxes account for close to 95 per cent of the 'alerts' that I see every week, and I know I am not alone.

There is a huge cost loss associated with hoaxes, and it is way beyond that which most viruses cause. There are psychological costs too. The worry caused by deletion of files that should not have been deleted. The humiliation of realising that one has been duped. The fear, uncertainty and doubt that is caused by thinking that there is an 'Undetectable Virus' on one's computer – and anyone who underestimates the power of that fear has never worked on an AV support desk.

I predict (or at least have a fairly large prescient twitch) that hoaxes will evolve in complexity over the next few years, until they are, effectively, indistinguishable from malware. Techniques for detecting hoaxes have always been based on pattern matching and intuition – the basic model for heuristic scanners. This has become increasingly difficult, and our 'scanning engine' (the brain) has had to be fed all sorts of new information to keep up. Goodbye Good Times.

PRODUCT REVIEW

HAURI ViRobot Expert 4.0

Matt Ham

HAURI has submitted its product to *VB*'s comparative reviews on a number of occasions over the last few years, and the changes to the program interface over this time have been notable. Detection rates – In the Wild at least – have shown a similar change over this time and thus a more detailed examination of the product seems due.

Origins

HAURI is a South Korean company and its country of origin is one which might have an influence on some of its products' features and abilities – a matter which bears some further inspection.

Starting from a historical point of view, *Microsoft Word* has, traditionally, been relatively less popular in the Korean market than *Excel*. This reversal of the Euro-American trend was due, at least in part, to the number of home-grown word-processing products which offered better support for the Korean hangul character set.

From a more modern perspective, Korea is currently one of the countries with the highest penetration of high-speed Internet access. Thus, even home users can be expected to have high-speed access, along with all the problems and benefits it brings.

From a pragmatic point of view, *ViRobot* might be expected to have better *Excel* than *Word* detection. There is also a likelihood that it will have a special dedication to the detection of recent mass mailers, while taking advantage of the high probability that the user will have a solid Internet connection.

So, how did the product fare? Since detection was covered in last month's comparative review (see *VB*, June 2002, p.16), it is afforded little space here and I shall focus instead upon the product's features.

Product Range

HAURI offers a suite of anti-virus products named *ViRobot*, with the *ViRobot Management Server* tool for the administration of these over a network. The company also produces a data recovery tool, *DataMedic*. Finally, with rather more scope than simply the control of anti-virus applications, *Sysers* is *HAURI*'s remote application administration program.

Currently, the platforms supported by *ViRobot* are *Windows* of the *95*, *98*, *ME*, *NT Workstation* and *Server*, *2000 Professional*, *Server* and *Advanced Server* and *XP* varieties.

In addition, *RedHat Linux* is supported (support for other distributions may also be available, but the advice on *HAURI*'s website is that enquiries should be made in these cases). Somewhat more hidden on the website is support for *Solaris*, HP_UX and AIX.

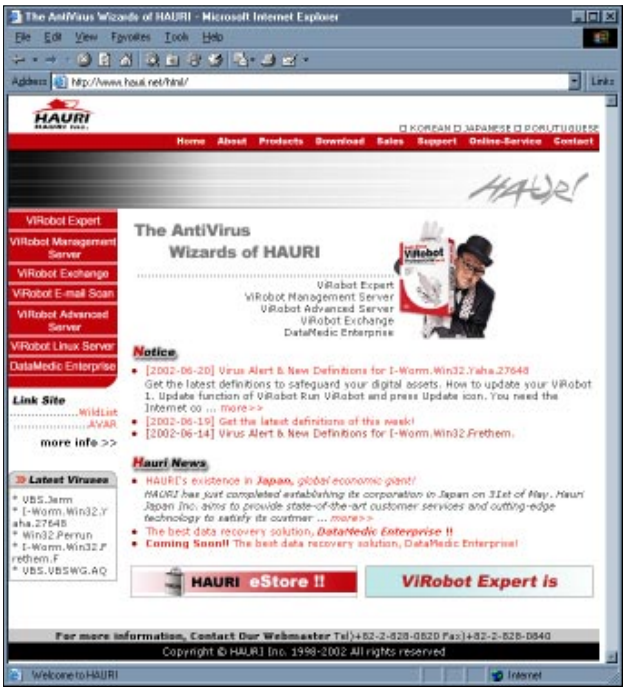
Furthermore, *ViRobot E-mail Proxy* can be used for mail scanning at the proxy level, while *ViRobot Domino*/*Notes* and *Exchange* is available for mail scanning through groupware.

Web Resources

HAURI's web presence is multilingual, covering English, Japanese, Portugese and, predictably, Korean. The various sites share a common look and feel, though the content varies a little between the sites.

Since my Korean is somewhat rusty these days, I concentrated on the English language version. This is available either via the central site, http://www.hauri.co.kr/, or can be accessed directly at http://www.globalhauri.com/.

The first obvious feature of the site is a pop-up box which delivers *HAURI*'s latest message of urgency to the world. For example, on my visits to the site I encountered a warning concerning W32/Elkern.C, which linked directly to a write-up of W32/Klez.H (the dropper for W32/Elkern.C), a short description/advertisement and a link to the latest update files for *ViRobot*. Since pop-ups can be annoying if triggered frequently, there is an option to suppress the pop-up for the remainder of the day.



Major portions of the website are exactly as might be expected – with downloads, contact information, sales details and other similarly worthy, but dull, content. More interesting parts of the site are the Support and Online-Services sections.

The Support section is the part of the website on which the product is registered. Registration allows a user to subscribe to update or alert mailing lists and 'qualify for special benefits, services and promotions' from *HAURI*.

The Online-Services section encompasses the LiveCall and LiveMedic features. LiveCall is an online scanning system, which requires a short download session to operate. This does not offer disinfection, but does offer a very good rate of scanning in comparison with similar services from other companies. LiveMedic is, presumably, related to *HAURI*'s *DataMedic* data recovery program, though I was unable to investigate since the link was not operational when I visited the site.

Installation and Updates

ViRobot is one of the many products that use InstallShield for standalone installation. Following a standard licence agreement, the installer prompts for user information and a serial number, then progresses onto the more interesting parts of the process.

First, the installation location may be selected, then the option is given to scan any currently running processes for viral activity. If this option is selected a dialog box is produced declaring, in the case of the standard clean installation, that no viruses were found. At this point the bulk of the file copying proceeds.

The next option is whether to produce emergency disks. Emergency disk production is an automated five-stage process, the first stage being to launch a format of the chosen disk. After this, system files are created and then *Windows* system files are copied and backed up. Finally, *ViRobot* utility files are copied to the disk. A caveat states that *ViRobot for DOS* will not work under NTFS – which leads to the assumption that the DOS product is part of the disk produced. This is, indeed, the case.

Since the rescue disk uses a *ViRobot* product, it comes as a surprise that the next installation option is to update the main program virus definitions – a task which it would, perhaps, be more logical to perform before writing data to the floppy disk. Since the rescue disks are static, it is likely that they will be out of date when required in any case. However, this problem is not as great as it might be, since the disk can be produced from within the main program. If Engine Update is selected a new dialog box appears, offering the option to start the download or configure update settings.

Should the configuration be changed, several options are available. First, the source of the updates is configurable amongst Internet, a *ViRobot Advanced Server* machine, a network source, or a floppy disk. Of these, the update floppy allows the creation of update disks as well as the setting of these as the default update method, while network and *ViRobot Server* allow browsing or searching for an appropriate source. Use of the Internet presents a choice of *HAURI* servers or a custom set server. It is possible to specify HTTP or FTP as the update method and proxy servers are supported.

Updates were between 0 K and 900 K when triggered daily – with a waiting period of one month leading to a 2 MB update size. The program distribution files were around the 15 MB mark for all these versions before updates, so it is safe to assume that the program files are being replaced, rather than added to, by the update process.

Documentation

The documentation was reviewed in its electronic format. The manual is clear, with very few stilted phrases having resulted from the translation from what was, presumably, the Korean original.

Although all portions of the program are covered in the manual, including installation, removal and general operation, the emphasis is placed upon correct configuration and the aftermath of scanning – which is where most problems are likely to lie. Even such cases as multiple infections and *Windows* locked files are explained clearly, the latter being an area by which many users seem stumped.

The section on actions to take when a new virus is detected is among the more revealing, since the information given on submitting virus samples is much more detailed than is usually the case.

In addition to the standard executable and macro virus submissions that are performed easily, *HAURI* supply as the default a set of boot sector extraction utilities and the documentation provides information on how and when these should be used.

One minor niggle remains, however, in that no mention is made of the use of encryption or encoding of infected files, so that, in many cases, the suspect files will not succeed in reaching *HAURI*.

No content-sensitive help is available within *ViRobot*, although an HTML help file is available from within the program. Clearly this has been based upon the manual, but it is sufficiently well adapted that it proves more useful than a straight copy would have been. The few queries I had about the program operation were, by and large, covered here, with only a few small omissions.

Scanner Features

As is the way with popular configurations, the default layout of *ViRobot* is very similar to those of many of its competitors. From top to bottom come drop-down menus and an icon bar, after which the interface splits.

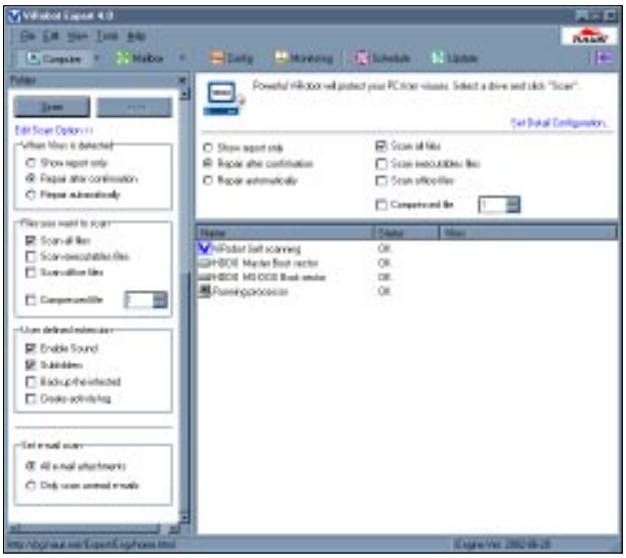
At this point the left pane contents vary, with the top right-hand pane being similarly changed depending upon which feature is under control, and the bottom right-hand pane being reserved for status reports.

The word 'default' as a descriptor of layout is important, however. The options on the icon bar are Computer, Mailbox, Config, Monitoring, Schedule and Update. Of these, Schedule and Update launch separate mini-application dialogs, while the remaining four alter the screen layout.

The Computer icon is selected by default. In the left-hand pane this displays a tree view of the local machine, with an option to extend this over the Network Neighbourhood. Here, areas to be scanned may be selected on a drive or directory basis, though not on a file-by-file basis.

This pane also contains buttons to initiate the scan and repair infected objects, as well as links to open the Edit Scan Options and Set Detail Configuration options.

Edit Scan Options opens up more choices within the pane itself. First, these determine whether there should be a report only if a virus is detected, automatic repair, or (as the default) repair after confirmation.



Furthermore, the file types to be scanned are selectable from all files (the default), executable only, or *Office* files only. In a rather redundant fashion, these options are all present already in the top right-hand pane.

Options included in the left-hand pane only are whether to enable sound (default on), scanning of subfolders (default on), backup of infected files (default off), or log files (default off).

From a dedicated email perspective there is a choice as to whether all email attachments should be scanned (the default) or only those that are marked as unread.

The Set Detail Configuration option from the left and top right-hand panes opens a tabbed dialog which controls many of the program features. This has tabs for Scan, Action, Startup, E-mail, Exclude Zone and Option. The Scan tab, among others, presents further choices in the determination of scanned objects, though some of the options are the same as those discussed previously. The status of the tabbed Configuration dialog is reflected in those other areas when Configuration is exited.

Scan gives the aforementioned choices of whether to scan all files or executables and/or *Office* files, as well as the opportunity to add a custom extension list. This last option is not a choice that is visible on the information displayed upon the general GUI – therefore, if only custom extensions are selected for scanning, the result is an interface which appears to state that nothing is being scanned.

Also on this tab, the option for scanning compressed files is repeated, along with the level of compression which will be scanned, enabling sound, backing up infected files, scanning subdirectories and creating a log file.

Again, there is an additional option here, which claims to determine the maximum size of the report file. There are two problems in this case: first, the maximum file size is noted as being 65 Kb and the second, more serious, problem is that there seems to be no dialog available in order to change this – nor any display of the current setting.

Again, the Action tab expands upon the choices displayed, and configured, from the general GUI. The action on infection list is maintained at the rather limited asking of the user, ignore and continue, or repair automatically. In addition, the choice is provided separately as to whether overwriting viruses should have the affected files deleted automatically. Whether or not this applies equally to worms, where there is no solution other than deletion, is not made clear.

The Startup tab offers a completely new set of configuration choices. This is where the automatic loading of *ViRobot Resident* is controlled. Also determined here are the options of whether to scan running processes or the hard disk boot sectors when *ViRobot* is started up. Finally, it is possible to flag the boot sector or any selected folder for scanning at operating system restart. Although labelled as a folder

option, this option is able to select drives for scanning – though only one area may be selected for this treatment.

Mirroring the GUI's small email options interface, the E-mail tab adds more detailed configuration. The server and application to be scanned are selectable here, with the server, like the proxy setting used earlier for updates, being detected automatically through *Internet Explorer* settings. *Outlook*, *Netscape Messenger* and *Eudora* are supported, though a full range of products were not tested here.

The section on mail monitoring gives finer control of *Outlook*-based mail scanning. This allows scanning of incoming or outgoing attachments from within *Outlook*, scanning of compressed file attachments and the setting of a size limit, above which attachments will not be scanned. These features are fully supported in *Outlook*; in *Outlook Express* there is scanning functionality but much less integration.

The Extensions tab is relatively self-explanatory – though it offers slightly more control than might be expected. Three separate areas exist: folders, files and extensions. Each of these may be selected individually for exclusion. It is not stated explicitly whether subdirectories are included when exclusions are set in this area.

Finally in the Configuration dialog is the catch-all Option area. Here is the option to display '*HAURI* news upon launching *ViRobot*'. This option defaults to off and, when selected, creates a pop-up link box when *ViRobot* is run.

At the time of writing, '*HAURI* news' consisted of three sections, each of which contained three links. The 'Notice' section contained links to information on the alert and definitions for I-Worm.Win32.Yaha.27648 and I-Worm.Win32.Frethem (both as labelled by *HAURI*). Also under this heading was a link to the general virus definitions updates area. The next section, '*HAURI* news', is more of a press release area, on this occasion containing links to two pieces of information on *DataMedic Enterprise* and an announcement about *HAURI*'s new Japanese offices. Finally, a section on new, but not particularly rampant, viruses provided links to three of these.

Additionally, the pop-up box presented a number of links to download areas and the latest readme.txt – which details the contents of the latest update.

At the Option tab, the ability to perform right-click scans, which is enabled by default, may be disabled. Also controllable here is MacRobot, a feature which might gain prizes for its confusing name. Rather than having any connection with Macintosh computers, this feature scans files before they are opened in *Internet Explorer* or *Microsoft Office*. Finally in the Option tab – and finally in the Configuration GUI – is the option to connect directly to *ViRobot Advanced Server* for updates when *ViRobot* is run.

This completes the options and controls available from the default Computer view. In this view, the great bulk of the screen is devoted to the scanning report. After having started the program this is not exactly swarming with information – in a default installation it contains information on the status of the *ViRobot* self scan, boot sector scans and running process scans.

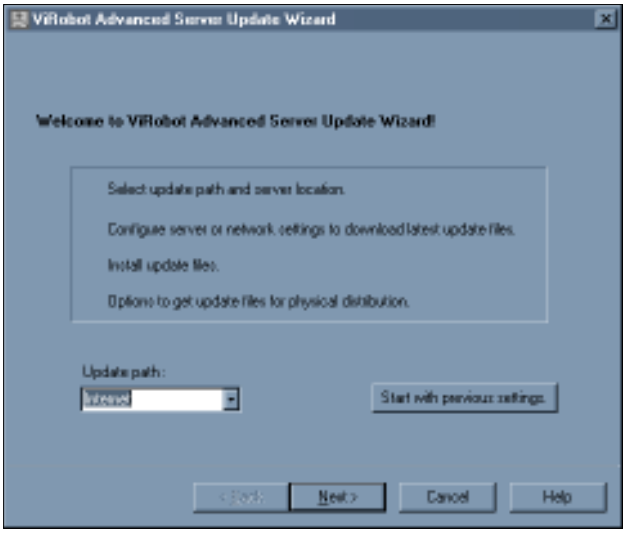
When scanning has resulted in detection of a virus the status of the files may be one of a wider-than-average collection when disinfection is attempted. Repaired, Deleted, New Virus, Suspected and Failed to Repair are all fairly standard. More novel is the inclusion of Repair after Decompression and Repair after Decryption (reserved for known infected files which are not repairable in their current state). This also applies to those files which have been declared as Access Denied – commonly this is due to a file being in use by *Windows*. Finally, Overlapped designates a file which has been disinfected of one virus, but which contains further infections which require that another attempt be made at disinfection.

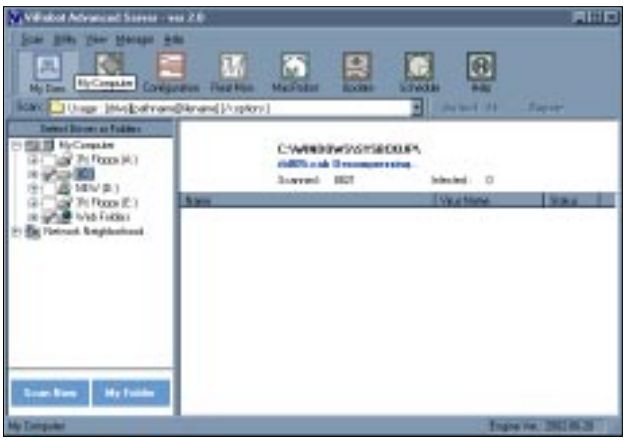
Alternative Views

The Computer view covers most of the commonly used functionality of *ViRobot*, there being a further three views which change the format of the interface: Mailbox, Config and Monitoring. None of these views add any configuration options to those discussed above, each presenting simply a more convenient way of changing some of these options.

Despite being arrayed with this collection of views, Schedule and Update on the icon bar are, in fact, simply launchers for other pop-up configuration GUIs, rather than changing the screen view or configuration themselves.

Update launches the Update dialog, which is identical to that which appears when installing *ViRobot*. Schedule is of more interest. This offers the ability, unsurprisingly, to trigger virus scans and updates of the software. Although much of this is standard scheduler fare, there are two trigger conditions which are out of the ordinary. It is possible to trigger updates based upon screensaver use – this,





presumably, being a time when the machine is not busy and thus updates will cause little inconvenience. For those users who like mystery and suspense it is also possible to trigger update events by use of a random timer. This is not quite as strange as it might seem, since it will impose a staggered load on update servers which may be desirable.

Management Features

ViRobot Advanced Server is recommended as usable on *NT* and *Windows 2000*, although it installs perfectly happily on *Windows 98*.

The program itself was only 5 MB in size. However, during install, it downloaded a further 35 MB of files for its use. These can be downloaded either from the Internet, or from another Advanced Server. Since at least one Advanced Server machine in a network can be expected to have full Internet access for updates, this method of distribution is not surprising.

ViRobot Advanced Server has a very similar feature set to those found in *ViRobot Expert*, though there are some additional features and a somewhat different GUI.

The difference lies primarily in the ability to act as a central repository for update files. These files can be downloaded actively by clients (as described in the comments concerning the *Expert* program above), alternatively it is possible to inspect and force updates from the *ViRobot Advanced Server*. In addition, there are several informational features available in *Advanced Server* which are not included in the *Expert* version. One feature notable by its absence is the ability to install *ViRobot* remotely from the server.

Scanning

As mentioned earlier, this review concentrates primarily in areas other than detection, though I shall include a little more detailed discussion of those results produced in the recent comparative review.

In those tests, the In the Wild misses were few in number and compare well with those obtained by other products. However, many more misses occurred in the standard, polymorphic and macro test sets.

In the polymorphic set these misses were distributed across two main groups, the old and the new. This may appear to cover the whole ground, but in reality the middle ground is a major factor and was detected well.

Detections were low on the 'more-complex-but-as-yet-not-in-the-wild' polymorphic viruses as well as those which can be categorised as sufficiently old to pose no major current threat. By and large, however, the viruses which have been seen more recently in the wild were detected, with the notable exception of W95/Marburg.A.

In terms of the standard test set, the same pattern occurred, with the misses predominantly being the very new and very old samples in that set.

This concentration of effort in detection is, if more than an illusion, a system with both pros and cons. On the one hand, the speed of scanning should be increased and false positives lessened, yet on the other hand, the chance of encountering one of the missed viruses is, although minuscule, not zero. In the past other developers have considered cutting down on their detection of older viruses, though this has not yet become noticeable in *VB*'s tests.

The division of detection in the macro test sets is one which backs up the theory that the product's *Excel* detection might be better than that in other *Office* applications by reason of a greater historical prevalence of *Excel* infections in Korea and the Pacific rim in general. This is still the case to a certain extent – recent prevalence data from Japan still identifies the number of *Excel* infections as representing 75 per cent of *Office*-borne infections. In the macro test sets *Excel* viruses were detected very well indeed, while it was the *Word* viruses which suffered relatively poor detection.

The Emergency Disks

The contents of the emergency disk created through the install procedure (also creatable from within the program) were examined. As mentioned previously, the option to create within the program is preferable, since this should ensure that the virus definitions are in their most up-to-date form.

When booted, the disk itself simply performs an automatic scan using *ViRobot for MS-DOS*, which takes some time on a machine with a large quantity of temporary Internet files. After the automatic scan the machine returns to an A:\ prompt. The only other *HAURI*-specific content on the disk is rebuild.exe – which restores boot sector information when run, providing the expected disclaimer as to the effect this might have if run on a different or altered machine.

Speed and Other Related Tests

ViRobot has a number of scanning options which might be expected to have an effect upon the speed of the scanning process – though the use of these in real-world situations turned out to be limited. The three main options available (and easily testable) are all files (the default), executables

only and *Office* files only. In many products 'executables' represents a rather broad category, including macro, script and other assorted malware. The *HAURI* way of thinking, however, is that executable means strictly that. It came as no great surprise, therefore, that a scan of this type took substantially less time that a standard scan, while detecting far fewer files. Selecting the option to scan only *Office* files had a similarly predictable result.

Examination of the files detected determined that combining the executable and *Office* detections and their respective scan times resulted in detection rates which nearly totalled that when all files were scanned, though engendering some additional misses. Whatever the exact details, the use of any but the 'all files' scanning option would seem inadvisable as a standard setting.

Conclusion

As noted in the introduction to this review, *ViRobot* has seen many changes over recent years, both in interface and in an improvement in ItW detection rates. However, detection rates are distinctly slanted towards the detection of certain varieties of virus, with weaknesses outside of these. That these weaknesses lie in the area of unlikely infections for most of *HAURI*'s existing customers may or may not be of relevance to a would-be user. As mentioned already, there are larger anti-virus companies who admit privately that they would like to take this approach but are unwilling to risk the potential customer concerns.

The features on offer also display this somewhat idiosyncratic way of operating, with the options to scan only executables or *Office* files being a good example. Management over a network is improving, though the lack of remote installation features is something of an irritation. For those who are concerned by such matters, *ViRobot* has recently been awarded the 'Designed for Windows XP' logo.

ViRobot does have specific detection strengths, as addressed at the start of this review, and it does make use of the Internet in a slightly more intensive way than other products – for example with *HAURI* news and the download-during-installation of *Advanced Server*. As for the future, speculation would be that the detection rate becomes better on a wider range of viruses, while the features continue to improve.

Technical Details

Test environment: Four 1.6 GHz Intel Pentium 4 workstations with 512 MB RAM, 20 GB dual hard disks, DVD/CD-Rom and 3.5-inch floppy, running *Windows XP Professional* and *Windows NT Server*. 1600+ Athlon XP workstation with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and USB ADSL internet connection, running *Windows 98 SE*.

Developer: Global HAURI, 3003 North First Street #304 San Jose, CA 95134, USA; Tel +1 408 232 5463; Fax +1 408 232 5464; email sales@globalhauri.com; website http://www.globalhauri.com/.

ADDENDUM

Windows XP Professional Comparative Review

Matt Ham

Since the publication of the *Windows XP* comparative review in the June edition of *Virus Bulletin* (see *VB* June 2002, p.16), a number of the tests have continued in the interests of determining the cause of problems which arose during these tests. The following conclusions have been drawn.

Panda Antivirus Platinum

The review noted that *Panda Antivirus Platinum*'s on-access scanner did not function when tested. Clearly this was an issue about which the developers were concerned, and the tests were repeated at that time, gaining the same result.

However, more recent tests, using the same hardware and software, have not demonstrated these problems. The lack of functionality noted in the review cannot, therefore, be taken to be indicative of a reproducible problem with the software.

Discussions with other developers have confirmed that the type of problem described is not uncommon with *Windows XP*. One theory put forward is that, at boot-up, *XP* does not always load all operating system components in the same order. With anti-virus programs being interwoven with the OS to an extreme degree, this might be a cause of such oddities.

NAI VirusScan

Also noted in the review was the fact that the sample of W32/Gibe.A was missed In the Wild by *NAI VirusScan*. This proved to be the result of an update method which, despite updating virus definitions, did not fully update the underlying engine. While this was the update method provided by the vendor, the results are not indicative of those which would have been obtained had SuperDAT files been used rather than DAT files.

The test results as published in the June issue are correct for the older engine tested, however, it should be noted that when subsequent tests were performed using SuperDAT files as an upgrade method, no files were missed by *VirusScan* In the Wild. Therefore, with the current 4.1.60 engine the product would qualify for the VB 100% award.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 9731266, Fax (781) 9731267

END NOTES AND NEWS

The Black Hat Briefings and Training 2002 take place from 29 July to 1 August 2002 at the Caesar's Palace Hotel in Las Vegas, USA. The briefings will consist of eight separate tracks over two days (31 July to 1 August), with ten different classes on offer for training (29–30 July). For further details or to make an early reservation see http://www.blackhat.com/.

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 visit http://www.isaca.org/, email conference@isaca.org or tel +1 847 253 1545 ext. 485.

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. Rather than formal speaker presentations, 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 on a first-come-first-served basis and is restricted to 50 people. See http://ntbugtraq.ntadvice.com/.

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 will take place at the Hyatt Regency, New Orleans, LA, USA from 26–27 September 2002. Take advantage of special *VB* subscriber rates and register now. Contact Bernadette Disborough; tel +44 1235 555139, or email VB2002@virusbtn.com. Visit the *VB* website for full programme details: http://www.virusbtn.com/.

Black Hat Asia 2002 takes place in Singapore, 1–4 October 2002. For further information see http://www.blackhat.com/.

Information Security Systems Europe 2002 will be held in Disneyland, Paris, from 2–4 October 2002. For more information visit 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 at the Queen Elizabeth II Conference Centre, Westminster, 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/.

Central Command* has released *Vexira Antivirus* for *Linux. For a limited time, a 25% discount is being offered on *Vexira Antivirus for Linux Server*, *Workstation* and email messaging server product *Vexira MailArmour*. See http://www.centralcommand.com/.