ISSN 0956-9979 MARCH 2002

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Feeling vulnerable:** Hot on the heels of *Microsoft* hiring a new Security Chief in order to provide its customers with ‘trustworthy computing’, Aleksander Czarnowski proves that security vulnerabilities are not the sole domain of *MS*. He looks at a selection of *Linux* vulnerabilities in an attempt to predict future threats. See p.10.

• **Tales of the uneXPected:** While the Web abounds with messages urging caution over the latest *Windows* OS release, Costin Raiu and Andreas Marx consider that an informed user should have no problems using *Windows XP*. They point out some potential pitfalls and recommend avoidance tactics, starting on p.12.

• **A social gathering:** Martin Overton believes the use of social engineering in the deployment of malware has recently ‘come of age’. He assesses the most successful techniques, starting on p.14.

CONTENTS

COMMENT

“ We serve and protect the user, but how do we reassure the user that it’s safe to go out into cyberspace again? ”

A Question of Support

When asked to write a Comment for *Virus Bulletin*, I asked myself ‘What do I have to say that other members of the AV community have not said already?’. Then, early one windy Sunday morning, a niggle formed in my mind.

The question was not ‘Why does this not work?’ nor ‘Why did the anti-virus miss that variant?’, nor even that can of worms (W32/worm.name.a or was it b) ‘Why can’t they all use the same name?’. The question was ‘What does it do?’.

One problem I have, and I suspect many *VB* readers share, is that when working with users on the end of a phone line or sitting with them in front of their infected machine, very often a question crops up to which we do not have the answer:

Support: ‘OK, you have set the AV software to delete – the worm will be removed and your machine will be clean.’

User: ‘But what does it do? How do I know nothing is left behind?’

Support: ‘I’ll get back to you.’

Support: ‘I have a user with this worm, what does it do?’

AV vendor: ‘It’s just another worm, we have so many we don’t document all of them so I can’t tell you, but we detect and remove it.’

So, what do we tell the user?

The AV industry in general does a great job in protecting against the ever-mounting threat of malware, but those of us who support many end users have another problem. Only about 70% of our task is removing the malware from the infected machine, the other 30% is about calming the users’ fears. This may be the first time the user has received a piece of malware. It may have triggered, in which case we are brought in to ‘clean up’. The AV software may have reported and stopped the malware, making ours a removal job, but the user needs to be reassured that he/she is ‘safe’.

I feel we can consider ourselves an ‘emergency service’ of the computing world. We serve and protect the user, but how do we reassure the user it’s safe to go out into cyberspace again? Or even that it is safe to turn on the computer again to write a simple memo? If we cannot tell them what may have happened to their machine it reduces the trust they have in our ability to protect them. They will probably grab a pen and paper for that memo – at least that’s safe isn’t it?

Those of us acting in computer support have a role of protection *and* reassurance, but what is the AV vendor’s role? Is it simply to stop and remove the malware or should those of us in the firing line between the vendors and their users expect more?

You detect it. You remove it. Someone must have some idea what it does – don’t you?

Dealing with remote users as well as those on site, I have found that one of the best support techniques in computing is dissemination of information and it’s not acceptable to say, ‘we can tell you what the common malware program does but not the uncommon ones’. Those of us who support users need more than that, the users expect more than that. In the wild we do see uncommon pieces of malware and we still would like to know what they do.

Is this possible? Maybe the AV vendors can answer that! What I do not want to hear is ‘it costs too much to document every piece of malware we see’. Most, if not all of us, pay for a service – I feel this should be part of the service. Even a single paragraph would be better than ‘don’t ask us’.

David Phillips, Computer Development Officer, The Open University, UK

NEWS

Three's a Crowd

They say that disasters always come in threes. Well, admittedly 'disasters' is over-dramatizing a little, but the three Microsoft Security Bulletins posted in the last week of February highlight some security holes that certainly have potential to cause significant problems.

First, MS02-008 describes a security hole affecting *Microsoft XML Core Services* versions *2.6*, *3.0* and *4.0* (and consequently *Windows XP*, *IE6.0* and *SQL Server 2000*). The hole exists in the XMLHTTP ActiveX control and may allow access to local files. Should the system be targeted for attack, an attacker would need to know the file path to access a specific file, and would have read access only.

Rather more disconcertingly, MS02-009 indicates that incorrect VBScript handling in *IE 5.01*, *5.5* and *6.0* could allow a malicious Web site operator to view files on the local computer of a visiting user. Furthermore, the vulnerability could allow the miscreant to collect information from a user's browsing session after the user had left the Web site. This information could include information such as user names, passwords, or credit card information.

Finally, MS02-010 details a faulty ISAPI filter in *Commerce Server 2000*. AuthFilter provides support for a variety of authentication methods and contains an unchecked buffer in a section of code that handles certain types of authentication requests. An attacker providing authentication data that overruns the buffer may cause the *Commerce Server* process to fail, or may run code in the security context of the *Commerce Server* process. Since the process runs with LocalSystem privileges, the attacker could gain complete control of the server.

Naturally, *Microsoft* recommends that users of the applications affected by these vulnerabilities apply patches immediately. Further details and the patches can be found at http://www.microsoft.com/security/bulletins/ ∎

As One Door Closes, Another Opens

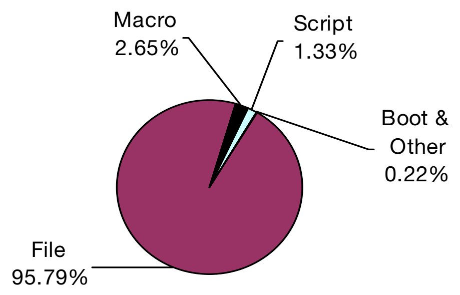
Computer Associates has announced that support for its free *InoculateIT Personal Edition* (*IPE*) software will cease on 15 May, 2002. *CA* announced in June 2001 that the product was to be replaced with the *eTrust EZ* anti-virus subscription service – however it advised that free support would be continued for users of *IPE* who chose not to migrate. Now, users of *IPE* are being given the chance to subscribe to *eTrust EZ* at a special reduced rate ($9.95 per year).
The news comes as *Softwin* unveils *BitDefender Scan Online*, a free web-based anti-virus scanning service (see http://www.bitdefender.com/). *Softwin* is soon to launch a new site, dedicated to free, downloadable AV products ∎

Prevalence Table – January 2002

Virus	Type	Incidents	Reports
Win32/BadTrans	File	1898	28.37%
Win32/Magistr	File	1416	21.16%
Win32/SirCam	File	1294	19.34%
Win32/Myparty	File	759	11.34%
Win32/Hybris	File	314	4.69%
Win32/Klez	File	121	1.81%
Win32/Aliz	File	117	1.75%
Win32/Nimda	File	115	1.72%
Win32/Goner	File	59	0.88%
Win32/MTX	File	55	0.82%
Win32/Maldal	File	49	0.73%
Win32/Gokar	File	48	0.72%
Laroux	Macro	46	0.69%
Win32/GOP	File	44	0.66%
Haptime	Script	34	0.51%
Kak	Script	34	0.51%
Win32/Shoho	File	28	0.42%
Win32/Zoher	File	21	0.31%
Bablas	Macro	16	0.24%
VCX	Macro	16	0.24%
Divi	Macro	15	0.22%
LoveLetter	Script	15	0.22%
Marker	Macro	12	0.18%
Win95/Spaces	File	11	0.16%
Melissa	Macro	10	0.15%
Ethan	Macro	9	0.13%
Others [1]		135	2.02%
Total		6691	100%

[1] The Prevalence Table includes a total of 135 reports across 60 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 …

The last words...

Those *Virus Bulletin* subscribers who attended my VB 2001 'MBCS and Office Macro Viruses' conference presentation might recall that, during the speech, I pointed out that between the countless *Windows*/*Office* combinations checked during my researching of the subject, only one was left untested. I was simply unable to obtain the required software in order to test the Arab *Windows* and *Office* installation.

Thanks to my dear friend, Mr. Petr Odehnal from *Grisoft*, who helped me with the testing of not only the Arab version, but also the Hebrew *Windows* and *Office* 2000, I can now draw my final conclusions regarding this subject.

These are the following: neither the Arab nor the Hebrew installations show the effect mentioned in my paper. These results confirm the conclusion drawn at the end of my paper, which stated that only the Japanese, Traditional and Simplified Chinese and Korean *Windows* and *Office* 2000 installations have this problem.

Moreover, following the initial discussion of the subject at the VB 2001 conference in Prague, *VMacro*, the group which deals with the classification and naming of macro viruses, has come to an agreement on how such modifications should be considered.

VMacro has agreed on the fact that, given that they are irreversible, and no possible algorithm can either determine the original form of source, nor exactly which succession of events caused the changes, the new instances should and will be considered new variants of the respective viruses.

As a practical example, W97M/Deldoc.B which initially was considered to be the same as W97M/Deldoc.A, was announced in November 2001 as a new, separate variant and named accordingly.

Finally, at the end of this addendum, I would like to express my special thanks to Mr. Yoshihiro Yasuda of *Network Associates Inc.* and Dr. Vesselin Bontchev of *Frisk Software Intl.* for their great help and support of my research on this topic.

Costin Raiu
Kaspersky Lab
Romania

CONFERENCE PREVIEW

Infosecurity Europe 2002

Helen Martin

Infosecurity Europe 2002 takes place in the Grand Hall at London's Olympia from 23–25 April 2002. More than 7000 IT security professionals are expected to pass through the doors over the course of the event, to visit exhibition stands, attend seminars, listen to the keynote addresses and meet with other specialists in their field.

The three-day event promises to include all the latest information on internal security threats, staff IT security training, business continuity, cyber-terrorism, hackers, legal issues, the introduction of employee charters, computer forensics, steganography (the art of hidden messages), managed secured services, penetration testing and electronic certificates.

More than 150 IT security vendors will be exhibiting at the show, which provides an opportunity for IT security buyers to bring themselves up to date with the latest advancements in the field and view demonstrations of new product releases. Many of the major players in the security industry are expected to announce their latest products and development news at Infosec Europe.

Keynotes

Following the popularity of last year's keynote sessions, the number of free keynote addresses has been increased to seven this year, and they will be held in theatres that will accommodate greater audiences.

The keynote sessions begin on Tuesday 23 April with the detailed findings of the DTI Information Security Breaches Survey 2002, which formed part of the DTI's work with UK industry to try to understand the impact of information security breaches. The survey was conducted between October 2001 and January 2002 and was based on 1000 telephone interviews and 100 face-to-face interviews in addition to an online questionnaire. It was designed to raise awareness among UK businesses of the value of effective information security management. (Further information is available at http://www.security-survey.gov.uk/.)

Also on 23 April, the issues of social engineering, education and dissemination of information will be amongst those highlighted in a keynote entitled 'Building the Human Firewall'. A panel of security experts on the council of www.humanfirewall.org will discuss in detail what can and is being done to help protect information assets from the perspective of changing human behaviour.

Network Associates and *BT Ignite* present their wireless security solutions in 'Vigilance in a Wireless World' on 23 April, while *Deloitte & Touche* look at how to reduce the risks to systems and processes through effective information security, demonstrating cost avoidance and revenue enhancement through security solutions. They will look in detail at how wireless security, malicious code threats, directory services landscapes and security policies can add value to businesses.

infosecurity
EUROPE
23-25 APRIL 2002
OLYMPIA, LONDON
www.infosec.co.uk

On 24 April the E-envoy's Office and CESG (Communications Electronic Security Group) discuss how information assurance is becoming of great importance to e-government and e-business. The session will debate how the public and private sectors can find ways to work together to ensure that the UK remains a safe place to live and work.

Thursday 25 April will bring a 'Live Hacking Session', where members of the audience will be able to hear from hackers themselves and find out more about how and why they do it. A panel session chaired by *Computer Weekly*, also on 25 April aims to explore the challenges faced by corporations, electronic service providers and governments in dealing with the problem of 'paper, people and policies'.

Seminars

As well as the keynote sessions, more than 50 seminars are scheduled to run over the course of the event. Amongst them, *Trend Micro*'s David Perry will be asking 'Who let the worms out?' in a presentation on computer viruses and evolution; *Ubizen*'s Nathan Tennant will discuss 'Managing the Cyber Terrorist Threat'; Shimon Gruper of *Aladdin Knowledge Systems* will outline 'Establishing and Managing Pro-active Content Security', while *MessageLabs*' Mark Sunner ponders how new mass-mailing viruses can be stopped in 'The Changing Face of the Virus Landscape'. Aled Myles of *Symantec* will reveal what every CEO, CIO, and IT executive should know about security and Gerry Ashton, IT sector specialist for certification body *Lloyd's Register Quality Assurance* will dispel the myths surrounding certification to the information security management system, BS7799.

Last, but not least, *Virus Bulletin* will be amongst the exhibitors at Infosecurity Europe 2002, so we look forward to meeting you there!

VIRUS ANALYSIS

Tasting Donut

Péter Ször
Symantec Security Response, USA

In early June 2001, several anti-virus companies received a copy of a new virus from its author. It was quick to be described by certain AV companies as 'the first known virus implemented in the Microsoft C# and Microsoft Intermediate Language (MSIL)'. However, this claim is simply incorrect.

Although the virus has a few bytes of MSIL code, the actual virus code is 32-bit assembly. The virus body does contain some MSIL code, but this is very short (a dozen or so MSIL instructions in the entry-point method) and is not involved in any way with the replication functionality of the virus. Furthermore, there is a minor bug which prevents the trigger function from working.

Regardless of this, the virus is interesting enough to take a closer look at, since it really is the first virus known to attack *.NET* applications.

The author of this virus, who goes by the name 'Benny', wanted it to be named 'dotNET'. Obviously such a name would not be acceptable, so I decided to call the virus 'Win32/Donut'.

.NET Framework

Two excellent papers addressed the forthcoming *.NET* virus problem at the *Virus Bulletin 2001* conference (Philip Hannay and Richard Wang's 'MSIL for the .NET Framework' and Eric Chien's 'Effects of Microsoft .NET on Malicious Threats'). However, it appears that a few things have changed in the file format since then and several things became obsolete straight after Beta 1.

The *.NET* architecture uses regular Portable Executable file format. The executable supports a platform-independent code. The actual code of the application is MSIL, which is compiled to native code on the fly by the JIT (just in time) compiler.

The Beta 1 file format is no longer supported on Beta 2. The *.NET* PE files use a six-byte platform-dependent code, a jump to a statically imported DLL function _CorExeMain() in mscoree.dll. This routine will initialize the Common Language Runtime (CLR) and thus the MSIL portion of the EXE applications. (DLLs will use a _CorDllMain() function.) The CLR is able to locate the MSIL code as well as the metadata via the CLR header defined as IMAGE_COR20_HEADER in the *.NET* SDK and available via the fourteenth entry (+0x70) of the data directory.

These are placed here for operating systems that are runtime-unaware and that are not marked for CLR-only execution. Evidently, *Microsoft* is willing to extend the system loader of its upcoming operating systems to support the file format without the need of a startup call or relocation section placed in the image.

Thus, according to the *.NET* SDK, these portions of the PE files can be safely ignored by future operating systems. However, for older systems this structure will provide backward compatibility.

Gaining Control

Donut gains control immediately upon executing an infected EXE file. The virus uses the simplest possible infection technique to infect *.NET* images. In fact, Donut turns *.NET* executables into regular-looking PE files. This is because the virus nullifies the data directory entry of the CLR header when it infects a *.NET* application.

The six-byte jump to the _CorExeMain() import is replaced by Donut with a jump to the virus entry point. The entry point in the header remains unchanged. In the past some of us in the AV industry considered this method to be an entry point obscuring technique – we call it 'obfuscated tricky jump' these days. Evidently this technique will fool some of the heuristics scanners.

The actual jump is a 0xE9 opcode followed by a DWORD offset to the start of the virus body in the first physical byte of the relocation section.

Initialization

First an exception trap is created to protect the code from unwanted GP faults. After this Donut attempts to identify the location of a loaded KERNEL32.DLL using a fairly standard approach found in 32-bit *Windows* viruses. In fact, the function handles a couple of operating systems correctly, including *Windows NT*/*2000*/*XP* and *Windows 9x*.

Donut uses CRCs of the APIs it wants to call. This saves a few bytes and makes analysis a little more complicated. Altogether there are 24 API CRCs in the virus body, starting with GetVersion(). The list ends with DeleteFileA(). However, some of the APIs, such as GetSystemDirectoryA() and SetFileAttributesA(), will not be called from the list. It is possible that the virus writer copied this section from another virus and these were remnants of that code.

Next, the virus checks the version of the OS. If the major OS version is not at least 5 the virus will not infect. Instead it will attempt to execute its trigger function and eventually the host. However, if the OS version is 5 or above the virus will attempt to infect all files with a '.exe' extension in the current directory and 20 directories above it, using a direct action method.

Since the host application has a static import to the mscoree.dll the virus will not be able to load if the *.NET* framework is not installed on the machine.

Infecting a File

First the infection routine checks the file size. If the file is too large (bigger than 4 GB) or too small (smaller than 2048 bytes), the file is ignored. Otherwise the file is opened and mapped.

Next the file is checked for the MZ mark and the PE mark using a tricky comparison. If the PE file is not a 386 image the file is ignored. Similarly DLLs, system and native applications are also ignored.

Now the relocation directory entry is checked. If there is no relocation section the file will be ignored. Otherwise the relocation entry in the data directory will be nullified. Thus infected files will no longer be considered for infection. Donut also ignores files that have a single section.

Next the virus looks for the CLR header entry in the data directory. If the RVA of the CLR header is not 0x2008 and its size is not 0x48 bytes, the virus will ignore the file. Similarly, the file is ignored if the image base is not 0x400000. Files that do not have an import address directory at 0x2000 are also ignored.

All of this is in order to search for applications compiled with the C# compiler. Thus Donut will not infect all *.NET* applications, but only a subset of them with a very specific format.

The Beta 2 *.NET* files have a 'BSJB' CLR header signature. The virus searches for this and therefore skips the Beta 1 files. The CLR entry in the data directory is also nullified. The entry point is replaced with a jump to the last section and the relocation section is overwritten with the virus code.

The virus saves the indirect pointer of the jump at the entry point, assuming a two-byte (0xFF25 opcode) jump right at the entry point. Again, this is an assumption for C# compiled executables. Since the virus does not check the stub code, in the case of a *.NET* file with an unusual entry point code the virus would fail to start the host correctly.

File Size

Donut saves 862 bytes starting at the CLR header and overwrites this with its own MSIL trigger function. The actual size of the 32-bit assembly portion of the virus code is 2180 bytes but the virus carries its trigger metadata (862) bytes, plus it will save the same amount from the original host. Thus the total size of the virus with the saved host data is 3904 bytes.

Since the virus will overwrite the relocations, in cases where the relocation section is sufficiently long the file might not increase in size after infection. This is very unlikely, however.

In the case of smaller files, and with most C# compiled executables, the file size will increase by 4 KB after infection. This is because the virus uses a 4 KB alignment to make the image acceptable for the CLR. In addition the size difference is corrected in the virtual size field of the last section header.

Next the last section is marked writeable. At this point the infection is almost complete but Donut calculates a proper checksum for the infected file by loading imagehlp.dll and using CheckSumMappedFile() API.

Finally the file is closed and the exception handling is removed. Normally this procedure should continue until each file is infected in the current directory and 20 directories above it.

Executing the Host and Trigger

Eventually the host would need to be executed. It seems Benny had the bright idea of not only executing the host application but executing the MSIL trigger routine too. Thus the procedure that originally functions as a routine to execute the host was modified as well as the infection routine.

However, Benny forgot about one thing which causes the functionality of the virus to change quite considerably.

Benny wanted to execute the MSIL routine so he copied it to the right place during infection. Obviously infected files need to start with the virus. Therefore the entry point code is replaced with a jump to the virus body.

On attempting to execute the trigger routine Donut makes a copy of the infected file. For example, on execution of 'runme.exe' the virus will create a copy of the file as 'runme .exe' (with a space inserted after 'runme') using CopyFile() API.

This image is mapped and its CLR header is fixed by assigning 0x2008 and 0x0048 values to the data directory entry. The virus uses the APIs CreateProcess() and WaitForSingleObject() to execute that image and to wait for its termination. However, the fix of the entry point code is missing. Since the virus writer failed to think about this problem, Donut will execute itself again. Thus a new file will be created, say 'runme .exe', followed by another file, 'runme   .exe', 'runme   .exe' and so on (with the name of each subsequent file containing a greater number of spaces).

The entry point is finally fixed for the execution of the host application. Thus the original host could execute many times and the MSIL trigger routine would never gain control in the CLR … Oh well.

Basically the virus is in an endless recursion. Eventually, however, the file name becomes too long and cannot be created.

This will stop the cascading effect and results in the execution of the host application sometimes more than 200 times. At least the host is executed with proper command line parameters. But is this not a little too much?

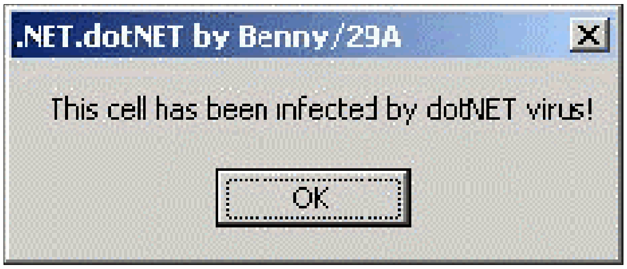
Eventually all the temporary files will be deleted by their executor so the mess is cleaned up in the current directory appropriately.

Getting the Message

The trigger would have displayed a message box with a one in ten chance. So, should the trigger function work, how would this message look?

A wide variety of interesting guesses from the AV industry can be found posted all over the Web. This is either because the virus was not replicated or the virus analysts might have simply considered themselves unlucky. I can only guess that a short VBS script did the trick for those nice write-ups.

The actual virus code would result in the following message box being displayed:


.NET.dotNET by Benny/29A

This cell has been infected by dotNET virus!

Conclusion

We probably need to wait a little longer until 'the first known virus implemented in the Microsoft C# and MSIL' is created. This could take a little more effort on the part of virus writers since it is not a simple matter at all.

In the future I would expect to see worms written with backdoor features that utilize the *.NET* networking features. Viruses that infect other *.NET* files might not have the opportunity to become a major problem.

Name: W32.Donut

Type:	Win32 direct action virus.
Removal:	Replace infected files from backup.
Trigger:	Attempts to execute an MSIL written application to display a message box with a one in ten chance, but fails to do so because of bugs.

TECHNICAL FEATURE 1

EPO – What is Next?

Malivanchuk Taras
Computer Associates, Israel

Virus writers have always attempted to make their creations difficult to detect. As a rule, this has meant creating viruses that would cause the maximum amount of work for anti-virus researchers. In the past, when the virus writers were successful in this aim, sometimes even the leading anti-virus products of the time were unable to detect a new virus for up to a month or more.

However, once the work of the AV researchers was completed, the only perceptible change was that the size of the anti-virus file had increased by a number of kilobytes. From the customer's point of view, nothing happened unless the new virus was found in the Wild during the period when it remained undetected by anti-virus software.

Scan Time and False Alarms

Some more significant effects of difficult to detect viruses are the slowing down of the scanning process and the production of false alarms. In fact these are two sides of the same coin, since false alarms are often a consequence of an insufficiently discriminatory detection procedure, and improving the detection procedure leads usually to a slower scanning speed.

The scanning time for a given virus (or class of viruses) on the end user's hard drive is the sum of scanning times consumed by every file to which the given detection method is applied.

So, the total scanning time depends on the speed at which the supposedly infected file is scanned, and on the number of such files on the disk. Usually virus writers concentrate on the first factor, rather than the second, which is even more significant. As far as the customer is concerned only the average scanning time is important.

History

Consider the history of executable parasitic viruses from this point of view. When non-polymorphic viruses appeared, the scanning speed was very fast because it was sufficient to scan the entry point with signature to detect non-encrypted viruses or non-polymorphic decryptors. Even if the decryption process was slow, the process of scanning clean files was not slowed down.

Polymorphism was introduced in order to make detection of the viruses more difficult. Detection of a polymorphic virus requires either static analysis or emulation which, in turn, requires more effort on the part of the AV researchers.

The more effort the AV researchers were required to invest, the happier the virus creators became. A consequence of this was that scanning of executable files became significantly slower.

In the DOS days, many EXE and COM files were written in assembly language. These do not carry compiler signature and therefore cannot be distinguished as clean by simple signature scanning. Additionally, every junk file of less than 64 K in size might be a COM file. So, for these files, time-consuming emulation or static analysis are needed.

PE File Format

When the PE file format became common, the situation improved. First, most PE files are created using high-level language and therefore are easily detected as uninfected by those viruses that infect the entry point.

Secondly, as a result of the more defined PE file structure, the use of obvious infection methods – such as attaching an additional section at the end of the file, or adding a 'tail' at the end of the last section – makes an infected file appear significantly different from clean files.

EPO

Then the EPO (Entry Point Obscuring) technique, which was already used by several DOS viruses, became more widely distributed. With the EPO method, some place in the victim body is patched by virus instructions in the hope that this point will gain control somewhere.

There are two methods for detection of EPO viruses: scanning for entry point and brute force decryption of the suggested virus body. The second method may be used only against weakly decrypted viruses.

Using the first method requires quite a lengthy process of scanning the whole victim code section, which involves at least reading the whole section followed by a further examination of every suggested entry point. If this process were applied to every executable, the scanning speed would be increased very significantly, if not to unacceptable levels.

Infection Sign

Fortunately, the infection sign is very helpful for detecting EPO viruses. The intended infection sign is a marker in the file that is used by the virus to distinguish an infected file from a clean one.

An unintended infection sign takes the form of some infected file feature such as a 'tail', or section alignment, that is present in every infected file but is which is not set by the virus deliberately.

Detection of EPO viruses using the infection sign decreases the percentage of suspect files so that the overall scanning time of clean stock is altered little. Were the infection sign to be removed, however, the only thing that would save us is the relatively small number of executable files on the disk.

From the virus writers' point of view, the most obvious way to remove the infection sign is for the virus not to check for infection at all.

The virus writer 'Zombie' employed this tactic in his virus Win95/Zmist. While version A carried an intended infection sign, it was removed in the subsequent versions. Version B has no well-defined infection sign, and versions C and D carry unintended signs that HeapReserve > 32 MB.

From our point of view, the B variant of Zmist is the strongest, the only unintended infection sign that it carries is its code section physical and virtual size relation and an alignment typical to Borland compiled files. Fortunately, most files on a typical computer do not carry this sign.

Slow Self Recognition

The idea of how to remove the infection sign in a cleaner way had already been used, but not yet widely implemented by the virus community: 'Slow Self Recognition' (SSR).

With this method, the infected file carries neither intended nor obvious unintended (provided the virus has no bugs) infection signs. When the virus examines the file, it analyses it using some time-consuming method, but it is not noticed by the user because the virus runs either in a separate thread or after program termination. There should not exist a better method to detect the virus than the virus itself uses.

Win32/Kant.2016 uses this method. This virus is one level encrypted, it puts its body after the end of the victim's code section, into the 'gap' in the virtual memory between two sections, moving the next section's file location forward if necessary.

The virus searches for a call to ExitProcess using call dword ptr and jmp dword ptr methods. If no ExitProcess is imported by the victim, or no call to it is found, the file is assumed to be either infected or inappropriate for infection.

Next all the calls to ExitProcess are patched by a call (or jump) to virus, so the virus gains control when the program is about to terminate. Because of this, the relatively slow infection checking process is not noticed by the user.

Finally, the infection sign. The virus aligns the physical and virtual boundary of the infected section to 1000h and the next section starts immediately after it in virtual memory. This is obviously a bug from SSR point of view, and it neutralises the effect of a possible slowdown.

Conclusion

Quite simply, the question is: are you ready for a virus such as Win32/Magistr with EPO plus SSR features?

TECHNICAL FEATURE 2

Linux Vulnerabilities

Aleksander Czarnowski,
AVET Information and Network Security, Poland

Some readers may remember the article 'Security Bulletin Gazing' published in *Virus Bulletin* last year (see *VB* August 2001, p.11), which focused on how analysis of the security vulnerabilities highlighted in *Microsoft* Security Bulletins could give us a good idea of future security threats.

This time, we will look into *Linux* security advisories and vulnerabilities with the same objective: predicting the future.

The Differences

With *Microsoft* it is quite easy – there is one form of security advisory, the 'Microsoft Security Bulletins'. All Bulletins are listed on one page on the *Microsoft* Web site (although, admittedly, this is harder to find since the recent modification of http://www.microsoft.com/security).

With *Linux* the situation is very different. There are at least three to five distributions in wide use (for example *Red Hat*, *Debian*, *Slackware* and *SuSE*). From time to time you also need to deal with locally popular distributions or very specific ones (such as *Immunix* or *Trustix*).

While most *Microsoft* products run on the x86 platform, *Linux* systems are available on many very different architectures. Every vendor uses their own advisory format. Some vulnerabilities are typical only for some distributions or a specific processor family, while others can have an impact on almost every *Linux* installation. Add the fact that patches are provided in different formats (source, rpm, deb etc.), and you begin to see how difficult it can be to analyse more than a few dozen advisories. Fortunately, the introduction of the CVE (Common Vulnerability Exposure) standard has been very helpful (see http://cve.mitre.org/).

Divide and Conquer

Keeping in mind the problems mentioned above, I have decided to divide the chosen vulnerabilities into three separate categories: kernel vulnerabilities, remotely exploitable vulnerabilities in applications and locally exploitable vulnerabilities in applications. Such an approach allows us to distance ourselves from the issue of different distributions and concentrate on the potential risks.

One of my first steps in analysing potential infection vectors for *Linux* systems was to have a look at the Unix section of the SANS Top 20 list (this can be found at http://www.sans.org/top20.htm – if you are using any other platform you should still have a look at this document).

The Unix section of the list describes seven types of popular Unix vulnerabilities, all of which are applicable also to *Linux* systems.

Of the seven at least six of the vulnerabilities could be used by malware to infect a system. These are:

- Buffer overflows in RPC services.
- *Sendmail* vulnerabilities.
- BIND weakness.
- R* commands.
- Vulnerabilities in lpd daemon.
- Sadmind and moutd vulnerabilities.

If you consider the most successful *Linux* worms to date, you will notice that virus authors have made use of some of these vulnerabilities already.

Note that, despite some remote vulnerability, older versions of *Sendmail* are vulnerable to local attacks too, and some of them could be used either during infection or in the payload procedure.

Kernel Vulnerabilities

i) ptrace/setuid exec (bugtraq ID 3447)

Several *Linux* kernels (from 2.2.x and 2.4.x lines) contain a vulnerability in the implementation of exec() function. This vulnerability allows the modification of suid processes in memory through the ptrace() call used for the tracing process.

Actually this is a very interesting vulnerability. It allows local users to gain root privileges on many default *Linux* installations where /usr/bin/newgrp is setuid root (Rafal Wojtczuk, 'Flaws in recent *Linux* kernels', message posted on *bugtraq* 18 Oct 2001).

Usually, kernel vulnerabilities are devastating for a security system and should be resolved immediately. If virus code is able to execute with root privileges or in kernel space, we will have lost control over our system.

Remotely Exploitable Vulnerabilities in Applications

i) SSH CRC32 compensation attack detector vulnerability (bugtraq ID 2347, CVE-2001-0144)

Secure Shell (SSH) allows remote administration in a relatively secure fashion. As it is in real life, even security products can be vulnerable. Code reuse, which is very common in the Unix world, can result in the spread of one programmer's mistake into several different products. This happened with SSH (as well as with many others applications – FTP vulnerabilities for example).

Several commercial and freeware implementations of SSH server, including OpenSSH, are vulnerable to buffer overflow in the CRC32 compensation attack detection procedure. This buffer overflow can lead to illegal code execution with SSH server privileges. Many SSH servers are run with root privileges. Successful exploitation of this vulnerability in such systems would result in full system compromise. In the case of Internet worms, this is a very important fact – not only can they infect servers remotely, but they can also gain root privileges immediately. Further analysis of SSH configuration files could reveal other potential candidates for infection.

Locally Exploitable Vulnerabilities in Applications

i) uucp (bugtraq ID 3312, CAN-2001-0873)

Some uucp implementations (Taylor UUCP) make it possible for local users to elevate their privileges by manipulating configuration files and passing them via a –config switch. This problem is not *Linux*-specific, as some BSD systems can also be attacked. What is important for us is the ease of performing an attack once the attacker is able to execute code in the user context. An exploit for gaining root privileges for *Red Hat Linux* is publicly available. Even if the user applied the patches to correct this problem the system is still not immune to attack (zen-parse, 'uucp – config patch – not sufficient', message posted on bugtraq 18 January 2002).

ii) sudo (bugtraq ID 3871)

If the system administrator is following current advice on Unix security he will be using su and sudo to perform some tasks with root privileges. Unfortunately, some sudo releases are vulnerable to local attack. Any local user can pass additional data to the executing application by using environment variables. Under some circumstances this would allow a local user to execute commands with root privileges. This can happen when postfix and the vulnerable sudo version are installed. In such circumstances postfix acts only as an attacker's 'vehicle' – the vulnerability lies in sudo, not in postfix. There is at least one publicly available exploit. It is worth mentioning that this vulnerability does not require any shellcode so it is not platform-specific.

iii) gzip (bugtraq ID 3712)

One of the most widely used tools by both administrators and attackers is the simple gzip application for file compression. Some versions of gzip are vulnerable to buffer overflow attack. The problem lies in how gzip processes file names inserted into its input. If a file name is at least 1028 bytes long a buffer overflow occurs. This could lead to execution of arbitrary code. While gzip is used across many different architectures, to exploit this vulnerability an attacker would need a specific shellcode for every cpu on which he would like to execute his code.

Preparing for the Nightmare

There are a number of network services that are run at most sites, such as DNS, SMTP, WWW, and POP3. We can draw some parallels here with *Windows NT*/*2000*/*XP* environments. For example, *Windows* servers run their own DNS server. Most *Linux* distributions come with some version of BIND. With *NT* servers, IIS is the default Web server. Most (if not all) *Linux* distributions are provided with *Apache*. Many IIS servers run *MS Exchange* also (and *MS Exchange 2000* requires IIS 5.0 or newer to be installed). In the *Linux* world, systems usually run *Sendmail* by default.

We don't even need to imagine what would happen if a remotely exploitable vulnerability were to be found in one of those applications. It already happened in *Microsoft IIS* (Directory Traversal) and BIND (Tsig buffer overflow).

In fact, a BIND vulnerability could have a very serious impact on Internet infrastructure due to the fact that many other network services, such as WWW, rely on DNS. If you look at Web server statistics for Internet-connected sites, *Apache* is number one. Imagine what would happen if a similarly devastating bug were found in it.

Fortunately, *Apache* does not run with root privileges so the impact of such a potential vulnerability could be limited. Also, *Linux* provides additional security mechanisms such as chroot environment, for example, which allows an application to operate on a separate file system with bogus configuration files and without access to the real root directory. Unfortunately chroot environment can be very time-consuming in setup and there are at least three different ways of escaping from it. The most popular is to use one of the links pointing outside chroot. Such links are often left in chroot environment by mistake.

One of the best options is to limit network services to a minimum and provide them though an application with a good record in the security field.

There are many servers like *Apache* that are reasonably secure and written in a secure fashion. Some MTA servers like qmail or postfix have been designed with security in mind. If you do not feel secure running BIND you can use djbdns from the author of qmail. There are also sets of kernel patches to tighten security in *Linux* systems.

Conclusion

So what will happen in the future? That is pretty simple. Someone will find a remote vulnerability in one of the popular services like DNS or RPC. A week or two later someone else will release an Internet worm that infects servers by exploiting the same vulnerability. Millions of unpatched servers will be infected in hours (unless the worm's IP address generator is poor).

'What's new in this scenario?', you might ask. Actually there is nothing new, I just train my skills in predicting the future by looking back at the past. Will millions of users update their anti-virus software and install patches on a regular basis? My guess is they will not …

TECHNICAL FEATURE 3

eXPect the uneXPected

Andreas Marx, AV-Test.org, Germany and
Costin Raiu, Kaspersky Labs, Romania

Long gone are the days when *Windows 3.11* was the latest and the greatest OS version from *Microsoft*, or when *Microsoft*'s flagship servers were running *Linux*. Nowadays, if you care to 'fingerprint' them with a tool such as 'Queso' or 'nmap', you'll see that most will be running some flavour of *Windows*, probably *2000*.

I say probably, because it's hard to determine exactly which *Windows* version a system is running based only on the replies of its TCP/IP stack. However, I have no doubt that a lot of them may be running the newly released *XP* version as well, as there's no doubt that some servers carrying *Microsoft*'s name might still be running some version of *Linux*.

New Kid on the Block

But it is obvious that, as time passes, more and more systems connected to the Internet will be running *Windows XP*, especially after the 'traditional' six-month transition period is over.

The fact is that, as for any new operating system which brings a host of new features and connectivity options, it is highly likely that *Windows XP* also carries a certain number of bugs, some already known, and some unknown.

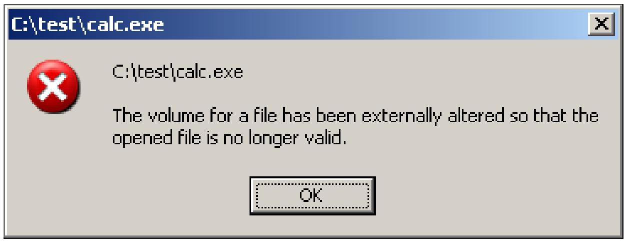
Since the release of *Windows XP*, we have come by a set of problems of which the most important are described in this article. Some of these have been fixed by *Microsoft* already, while others remain unfixed since they are more or less regarded as 'features' of the OS version, or that they work this way 'by design'.

We intend this article to be a useful reference for IT staff or system administrators who have to deal with *XP* systems in their networks, or for the casual *XP* user whose computer is running, according to a *Microsoft* quote, 'the most secure *Windows* version ever'.

1. Manifest Files

The phenomenon of the '.manifest' extension may not be widely known amongst new users of *XP*.

If you create an empty file named with the same name as an existing executable on your disk followed by the '.manifest' extension (for example, 'notepad.exe.manifest') and then save the file in the same directory as the respective program, then *Windows XP* will steadfastly refuse to execute the program anymore.


C:\test\calc.exe

C:\test\calc.exe

The volume for a file has been externally altered so that the opened file is no longer valid.

As can be seen above, on attempting to execute the file, a more than cryptic message appears: 'The volume for a file has been externally altered so that the opened file is no longer vaild.'

If you attempt to start the respective program from the command prompt, a more or less generic message of the same form appears, which says: 'The system cannot execute the specified program.'

Curious things, '.manifest' files are new additions to *Windows XP* – their main purpose is to allow developers to specify the so-called 'shared assemblies' between modules and applications.

Unfortunately, the problem is that in order to render a system unusable, one does not need to have the right to change important system files – the permission to add an innocent empty '.manifest' file into the right place is more than enough.

Moreover, '.manifest' files are supposed to be located in a special sub-folder of the *Windows XP* installation directory. So, for example, there should be no legitimate reason for one to exist in the 'system32' directory.

Such a problem is likely to be very hard to diagnose, especially given that no existing file has been modified, and no change has been made to the registry. That's why you may want to look for zero-byte-sized '.manifest' files if you ever happen to encounter one of the messages listed above.

2. Universal Plug'n'Play

By default, on the TCP port 5000 and UDP port 1900 of *Windows XP* systems there is a service listening for connections called the 'SSDP Discovery Service'.

Basically, this service provides an interface between the network and the 'Universal Plug and Play Device Host', the service taking care of Plug'n'Play devices. The main purpose here is, of course, to allow your computer to discover and use automatically any Plug'n'Play devices connected to the network (such as 'smart' printers, or remotely controllable microwave ovens).

On 20 December 2001, *eEye Digital Security* released an advisory which covers three major bugs in the Universal

Plug’n’Play (UPnP) implementation included in *Windows XP*. These three bugs allow a remote attacker to launch a DoS attack against the respective system, to use the system to make connections to other arbitrary addresses on the Internet and, worst of all, to execute code on the system with higher privileges.

Microsoft’s response can be found in the MS01-059 Security Bulletin (see http://www.microsoft.com/security/bulletin/MS01-059.asp), in the form of a 585 KB executable that replaces a number of system files, amongst them ‘ssdpapi.dll’ and ‘ssdpsrv.dll’. This package takes care of the vulnerability and protects the affected systems against the three types of attack.

Given that it took the author of CodeRed about one month to write the worm after a public exploit for the respective vulnerability was released, we wonder how long will it be before we see a similar thing exploiting the *XP* UPnP hole.

And unfortunately, not only will such a thing have a much larger target base than CodeRed (we expect the number of *Windows XP* systems on the Internet to outrank the number of *Windows 2000* systems running IIS), but also it should be more compatible than CodeRed and, technically, be able to spread much faster.

3. Outlook Express 6.0 and the German Version

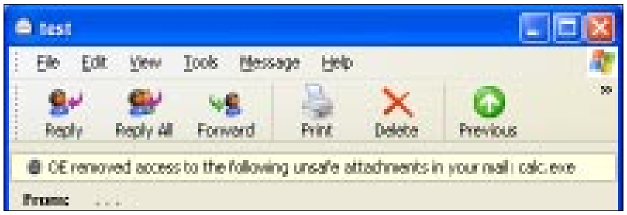
Installed and configured by default into *Windows XP*, *Outlook Express 6.0* and *Internet Explorer 6.0* have the task of acting as ‘email’, ‘news’, ‘Web’ and ‘ftp’ access clients.

Of these, it is interesting to note that *Outlook Express 6.0* includes some basic ‘virus-protection’ options which can prevent the user from accessing attachments with a certain set of extensions belonging to executable or script files, such as .VBS, .EXE or .BAT.

This is intended as some form of simple protection against email viruses, and despite the fact that it greatly reduces the usability of the product, it might actually prove useful in some cases.

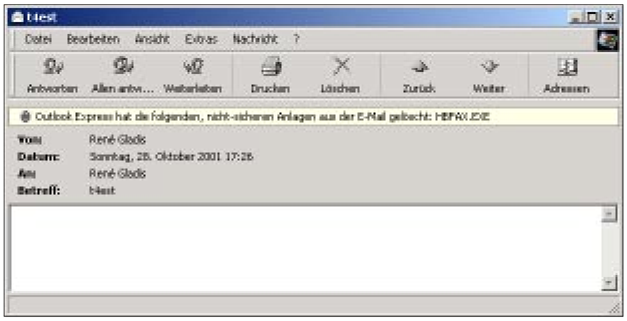
Whenever the user receives an attachment in the form of a file with one of these extensions, *Outlook* will display the ‘status’ message: ‘OE removed access to the following unsafe attachments in your mail: filename.extension’

However, it seems that the team that translated *OE6.0* into German, made a little mistake, so instead of ‘removed access’ the German version of *OE6.0* says it has ‘deleted’ the attachments.



OE removed access to the following unsafe attachments in your mail: calc.exe

From: . . .



Outlook Express hat die folgenden, nicht-sicheren Anlagen aus der E-Mail gelöscht: HBFAX.EXE

Von: René Gladis
Datum: Sonntag, 28. Oktober 2001 17:26
An: René Gladis
Betreff: t4est

So, quite understandably, an unsuspecting user might imagine that the attachment had actually been deleted from the message. Thus a false sense of security is created.

First of all, the problem is that the attachment has not been deleted. If the *OE6.0* security option is disabled, the attachment can be accessed again without problems.

Secondly, the attachment could very well have been something useful to the receiver – a legitimate file that the user was expecting and wanted to receive. This way he/she may be tricked into believing that the attached file has been lost.

And finally, if the attachment was indeed infected with a virus, imagine the surprise of the user who is certain that the attachment has been ‘deleted’ from his mails, while an anti-virus product able to scan the *OE6* mailbox reports the virus still to be present in the message.

Of course, a proper fix would be required in this case, which translates to the right meaning in the German *Outlook Express 6.0*, but until then, users should be aware of this fact.

4. The Windows XP Personal Firewall

Of the many security features *Windows XP* can provide, of great interest, especially to home users who connect their systems to the Internet through a dial-up, cable or DSL link, is *XP*’s embedded Personal Firewall (PF).

Once activated, the *Windows XP* Personal Firewall does a very simple, yet very effective thing – it will prevent remote machines from initiating connections towards the protected system on a large array of TCP/IP ports, thus greatly reducing the possibility of external attacks.

Of course, the Personal Firewall can be explicitly permitted to allow certain ports to pass the lock, which is very useful if someone wants, for example, to run an ftp server.

The only problem with the Personal Firewall is that, under various circumstances, it will open a server port automatically for connections from the outside. In doing so this allows remote access to the machine virtually from anywhere on the Internet, without even notifying the user.

This problem occurs when someone has the Personal Firewall running, and tries to activate the *XP* Remote

Desktop Server. During this process, *XP* will add the Remote Desktop port to the 'allowed' server ports automatically, and silently give remote parties the ability to initiate a Remote Desktop session with the machine.

Of course, to initiate the Remote Desktop session, one would also require a valid username and passport. However the fact remains that the first step has been made, and along with it, a door has been opened into the security defences of the machine, without any warning at all to the unsuspecting user.

Microsoft was notified of this problem in November 2001, and the issue was said to be under investigation, maybe scheduled for fixing in the future.

One other thing we should mention is that this effect could not be reproduced on all of our test configurations. It did not occur on an English test installation of *XP*, but initially it was found and reproduced on the German MSDN *Windows XP Home* and *Pro* versions.

Some Conclusions

The recommendations to practise caution with *Windows XP* are posted virtually everywhere on the Internet. That's why we are not going to add any fuel to the topic.

On the contrary, all of the problems we have mentioned in this article can be avoided through very simple means, and an informed user should have no problems (provided the translation issues in the German *Outlook Express 6.0* are dealt with).

So, if you want to take advantage of all the new features in *Windows XP*, just go ahead.

But wait! When you install *Windows XP* don't forget at least to take care of the UPnP problem by installing the patch or disabling the SSDP service.

If you have a firewall, we recommend that you close the TCP port 5000 and the UDP port 1900 – there's absolutely no reason why someone from the Internet should connect a Plug'n'Play device into your network.

Also, especially if you use an isolated computer, it would be better to install a separate, more configurable personal firewall with more features than *XP*'s built-in implementation which is designed to provide only a very basic line of security. There are many very good personal firewall applications available on the Internet, many of which are free, and most of which are reported to work without any problems on *XP*, even if *XP*'s built-in personal firewall is running as well.

And finally, if you notice any of the strange error messages indicated in the '.manifest' section of this article, you may want to take a look for any such files in the system or *Windows* directories since, most likely, they have no legitimate reason to be in there.

FEATURE

You are the Weakest Link, Goodbye! – Malware Social Engineering Comes of Age

Martin Overton,
ChekWARE UK

The Year of the Snake in the Chinese astrological calendar ran from 24 January 2001 to 12 February 2002. In Western culture snakes are seen as treacherous, sly, sneaky and the very embodiment of evil. They are credited as being able to 'hypnotize' or 'charm' their prey, so that they appear entranced and pliable to the will of the snake (this has no apparent basis in fact, as the intended prey is probably just frozen by fear).

Does this sound familiar in the realms of malware, with the virus authors, or more specifically the social engineering aspect that has become prevalent in malware recently 'hypnotizing' their victims? Certainly, it would seem that 2001 could be dubbed the Year of the Malware Social Engineer.

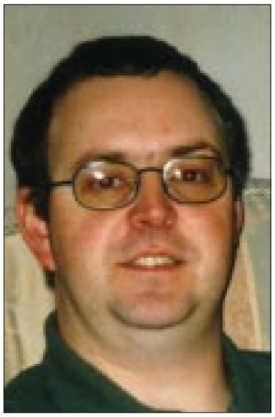
More Social than Most?

It has been obvious for some time that many virus writers have been watching and learning from the hackers (I use the term in the way most 'non-anoraks' use the word – those real 'old-time hackers' out there, please forgive me for using the media definition).

The appearance of W32/MyParty@MM and W32/Porman@MM (not to mention W32/Nimda@MM) has shown that scenarios (nightmare or not) discussed by researchers behind closed doors have also been thought of by those in the wider world – specifically those in the malware-writing world (much to the vexation of the virus researchers and those of us in large organizations, who are forced to play 'chicken' with the latest and greatest of the 'just-released' threats).

Several months ago, I pondered with a few other security professionals why virus writers hadn't taken the next obvious route and used Web (http-like) addresses (as attachments to the email, rather than a URL in the body of the message which links to a Web site) as a means of increasing the likelihood of their creations being executed by the less vigilant of their victim pool.

The obvious attachment names to use, from a social engineering point of view, are Web addresses ending in '.com' (minus the http:// prefix), as most computers will happily run such a misnamed file as an executable – especially if it is a binary file format (i.e. *.COM, *.EXE).



The use of social engineering is not new in the world of malware; however, I believe it has now ‘come of age’ (just coming up to the teenager stage, in fact, and bringing all its problems, attitude and teenage angst with it).

At the very least, social engineering is the current flavour-of-the-year for ‘suckering’ potential victims, by using ‘new’ ways to tempt them into taking the bait.

Luckily, some of the victim pool have wised up to the fact that email attachments (from strangers) may spell danger, and treat them as the electronic equivalent of ‘Typhoid Mary’. However, that is not usually the case when the unexpected files are received from someone they know.

Yet again, we have seen malware authors ‘steal’ another recommended way to deal with the thorny issue of email attachments. Many companies (both AV vendors and IT Security departments) have suggested that directing recipients to a URL instead of sending a file attachment is ‘safer’. Time to think again, maybe?

What is Social Engineering?

Here’s a short, but very apt, definition from Jargon File: ‘***Social engineering n.*** *Term used among crackers and samurai (hackers for hire) for techniques that rely on weaknesses in wetware (people) rather than hardware or software.’* (See http://www.tuxedo.org/~esr/jargon/html/entry/social-engineering.html.)

In his book *Secrets and Lies*, Bruce Schneier lists social engineering as one of six ‘aspects of the human problem’ when focusing on information systems security. He states that social engineering is ‘very effective’, and that it goes straight to the ‘weakest link in any security system: the poor human being trying to get his job done, and wanting to help out if he [or she] can.’ (See http://www.rr.sans.org/securitybasics/awareness.php.)

In reality, however, social engineering is lots of things and it is even harder to pin-down when it is used in relation to malware, but the key to it all is the following: ‘Someone wants something you have (or have access to) or wants you to perform an action (such as disclose information, run a program). To achieve this, the would-be Social Engineer will lie (claim to be someone or something they are not, or that they have access to something they are not entitled to), cheat (forge credentials or get you to run code that does something to escalate rights or install a backdoor by convincing you that it is something else) and steal (data, passwords, identities, availability of system resources)’.

In short, the would-be Social Engineer plays on the natural human tendency to trust, and to want to help others.

Damned if we do, Damned if we don’t

Part of the problem is the fact that ‘we’ (that is a collective we, not a royal one) have anti-virus solutions in place in many companies: at the mail gateway, on file and print servers, internal mail servers, http and ftp gateways, and on our final bastion, the desktop.

This ‘multi-layered-approach’ actually appears to exacerbate the problem, as end users seem to be more willing to take a risk with an attachment or link because they know that the company (and therefore their computer) is ‘protected’ by anti-virus software. This leads them into an ‘it-can’t-happen-to-me’ attitude, and even if it does happen, the users tend to see it as being not their problem, but an IT problem.

Sex, Lies and Topical Themes

Let us have a look at the some of the various methods and themes that have been tried, and how successful they have been:

Sex:	W32/Pops@MM, W32/Toget@MM, W97M/Melissa@MM, VBS/Loveletter@MM, VBS/VBSWG@MM (Anna Kournikova)
Fear:	W32/Whitebait@MM
Greed:	Nigerian Money Transfer and its many variants[1]
Altruism:	W32/SirCam@MM, W32/Myparty@MM, SULFNBK.EXE Hoax
Authority:	Appears to come from someone you know or trust (such as the almighty MS or an Anti-Virus/Security company)
Humour:	W97M/Comical@MM, W32/Roach@MM
Games/ ScreenSavers:	W32/Maldal.d@MM, BudFrogs Hoax
Topical:	W32/Ska@M (Happy99), W32/Maldal.c@MM
Anti-virus or Security updates:	W32/Whitebait@MM
Double extensions:	W32/Magistr@MM, W32/Sircam@MM, W32/Badtrans@MM
‘Polymorphism’ :	W32/SirCam@MM, W32/Gokar@MM, W32/Klez@MM

(‘Polymorphism’ in this instance refers not to true polymorphism, but to random subject lines, body text, attachment names and extensions.)

Well, how successful were some of those?

Very successful (either very fast-burners or had a long-term presence): W97M/Melissa@MM, W32/Sircam@MM, VBS/Loveletter@MM, W32/Badtrans@MM, W32/Hybris@MM, W32/Magistr@MM, VBS/VBSWG@MM.

Quite successful (either fast-burners or had a long-ish presence): W32/Goner@MM, W32/Maldal.d@MM, W32/Ska@m, W32/Maldal.c@MM, SULFNBK, BudFrogs Hoax.

Average success (never really took off): W32/Klez.

Poor (a mere drop in the ocean): W32/Whitebait@MM, W32/Roach@MM.

The Scores on the Doors

Why did some of those listed above do far better (i.e. spread further and/or more rapidly) than others?

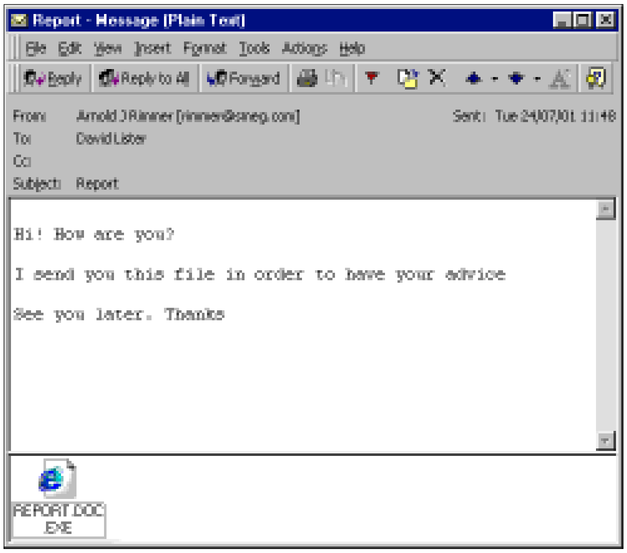
Friend or foe? a major plus point for successful distribution of malware seems to be the arrival of the file from someone the intended victim *knows*. This appears to be the key factor for 'Electronic Ephemera' (hoaxes and their kin).

Psychological buttons: those pieces of malware that use the typical 'high-interest psychological buttons' seemed to do best – sex, greed, altruism and topical themes.

Timing: this can make a big difference, but it plays a small part in the overall 'malware-that-got-lucky' model.

Holes: those pieces of malware using known exploits in target operating systems or applications worked very well. This is a trend I expect to be taken to new levels this year, making W32/Nimda@MM seem like a minor annoyance.

Originality: those pieces of malware that are significantly different from any other (the first of their ilk) tend to be more successful, when combined with other factors listed above, as they are less likely to be detected either heuristically or via generic (family) signatures.


Report - Message [Plain Text]
File Edit View Insert Format Tools Actions Help
Reply
Reply to All
Forward

From: Arnold J Rinner [rinner@sneg.com]
Sent: Tue 24/07/01 11:48
To: David Lister
Cc:
Subject: Report

```
Hi! How are you?

I send you this file in order to have your advice

See you later. Thanks
```

REPORT.DOC.EXE

Own SMTP engine: this seems almost to be the *de facto* standard for today's mass-mailers, and tends to improve the ability for a mass-mailer to spread.

The Solutions

Let's have a look at the possible solutions to some of the problems which social engineering malware adds to the 'overall' malware problem in an organization:

i) Technology

Can the problem be solved completely purely by throwing technology at it?

Of course not, you'd need a full integrity management system (to identify/block changes) as well as a change management system (to check whether any changes were authorized changes) and an expert system to attempt to manage the whole process. Even with all of these precautions in place there is no guarantee that an unauthorized change couldn't happen.

Even if we succeeded in building a better snake trap, the snake would evolve so that it could avoid or bypass the trap. It quickly develops into a snake and mongoose (malware writer and anti-virus/security company) game, and the victims (end users and their companies, aka the 'mice') are the innocent casualties of the game (i.e. 'lunch').

If technology is the answer, then why have many companies suffered with the likes of W32/Nimda@MM, W32/Goner@MM and many other mass-mailers (many of which have inbuilt social engineering elements)? Obviously if technology is the 'whole' answer, then something is not working.

ii) Education

I am on record as stating that trying to educate end users about malware is mainly a waste of time and I stand by that still, even though I made the statement as long ago as 1996 when I was a virgin *VB* Conference presenter.

In large companies educating the end users is like painting a very long bridge: once you've reached the end of the bridge, it is time to start painting from the beginning again.

Furthermore, trying to educate end users about malware issues is like expecting a car owner to understand how the car works, when all they really need to know is how to use it, when to fill it up, check the tyres, send it in for repair (or get it towed), and what to do when things go wrong (who to call).

In other words, a simple check list is all the end users need. They don't need to know how to strip the engine, nor understand the mechanics or physics involved.

To take this metaphor further, your support staff can be likened to garage mechanics; *they* are worth training. This is also true with regard to your in-house developers and systems administrators and other operations staff.

Finally, educating end users in large companies is very expensive, both in terms of the cost of hiring suitably qualified trainers and in the lost productivity of the trainees.

In the light of this, many companies see the occasional outbreak as an acceptable risk that they are (currently) prepared to accept and sign off.

iii) Policy and Procedures

So what will work with your end-users? Give them simple guidelines, policies and procedures for them to follow, which are easy to understand (and which do not consist of lots of 'techie-speak').

Below are some simple guidelines for end users that could help in combating the threat from social engineering-based malware:

- Follow 'Safe Hex' guidelines, which should be available at http://yourcompanyintranet.com/safehex/. (Most AV companies have some basic guidelines which you can adapt for your own company. Otherwise see: 'Safe Hex in the 21st Century', *VB* June 2000 p.16 and *VB* July 2000 p.14 for some suggestions and guidance.)
- Send all received warning emails, or suspect files to 'suspect@yourcompany.com' – a central email drop-box for your company which is monitored by a team (or member of staff) that understands malware and related issues and knows where and how to verify or debunk received files or warnings.

For your support and operations staff:

- Create an intranet site, and put policy, procedure and FAQs, virus warnings, hoax information and other pertinent documents there.
- Publicize a 'hot-line' number and encourage employees to use it.
- Join Security Alert Mailing Lists.
- Monitor AV/security/hoax Web sites.
- Join an Early Warning System-type service, such as AVIEN's EWS (see http://www.avien.org/).
- Ensure your systems are patched.
- Roll-out anti-virus updates as soon as you can (after testing them of course).
- Ensure that you educate your users via a simple, easy-to-understand security policy, which is underpinned with good and well-documented processes and procedures.
- Make it clear to staff that not following the guidelines, etc. could lead to disciplinary action, and it may even cost them their job.

As with most things in life, a mix of approaches or a 'holistic' approach offers the most appropriate way of dealing with malware threats, including social engineering malware. Use whichever of the suggestions listed in this article will work in your environment. There are almost certainly others that I haven't covered here.

Conclusions

It seems clear (to me) that the use of social engineering in malware is likely to increase for the rest of the year and I believe it will be the most effective way for malware authors to ensure that their creations achieve a wide and receptive audience.

The current increasing 'trend' of mass-mailing malware can be likened to rape. Statistics show that most rapes (71 percent in the US and as many as 97 percent in the UK) are committed by someone the victim knows and probably trusts (see http://abc.eznettools.net/D302506/X329849/stats.html and http:/www.rapecrisis.co.uk/statistics.htm). In the case of mass-mailing malware, both the sender and recipient are, more often than not, both 'victims'.

We need to ensure that end-users are more careful with all electronic information that they receive from those they know and trust, as well as from strangers.

We need to establish in our user-base the need to be more cautious, suspicious and a little more paranoid, to help minimize the chances of them becoming part of the problem, rather than part of the solution. If we fail to do this, then we too are also part of the problem, and not part of the solution (which should be our goal).

However, you must remember that (in most cases) it is ultimately a human being pushing the buttons (and having their psychological buttons pushed) and this is the root of the problem.

The 'key' to breaking this cycle, I believe, is to make the end users accountable, make security their problem too, and remove the focus from it being an IT (a technology) issue, to what it (social engineering) is – a 'human problem'.

The old maxim states: 'Curiosity killed the cat'. How many lives do your end users have left? In fact, the character Fox Mulder from the *X-Files* TV series may have the ultimate mantra for our staff to learn … 'Trust No One!'

Footnote

[1] There are many reports from both the UK and the USA that a surprising number of mugs … er, I mean unsuspecting victims lost a significant amount of money, and occasionally their lives as a result of being taken in by the 'Nigerian Money Transfer'. So much so, in fact, that it has been subject to both an FBI fraud alert (see their Web site http://newyork.fbi.gov/contact/fo/nyfo/fraudalert.htm) and a warning from the US Secret Service (see http://www.ustreas.gov/usss/index.htm?alert419.htm).

PRODUCT REVIEW 1

Kaspersky AntiVirus

Matt Ham

As mentioned in the last issue of *Virus Bulletin*, recently *Kaspersky AntiVirus* underwent a twofold change. Of course it is the surface change which is the more obvious upon cursory inspection, but this striking re-ordering of the visual aspects of the product is coupled with the introduction of a new scanning engine. With so many changes afoot, there is clearly scope for a more detailed inspection of the product than was afforded by last month's Comparative.

The Package

The *Windows NT* availability of *Kaspersky AntiVirus* (*KAV*) stretches over four product lines. Of these, *KAV Lite, Personal* and *Personal Pro* are the more home user-oriented lines, while *Business Optimal* is a package which covers a wide range of operating platforms with Workstation and Server components available for *NT*.

For the purposes of testing, a *Windows NT 4* server and *Windows 2000* client configuration was the base from which the observations in this review were intended to be made. However, there was a major new release of the *Business Optimal* product late during the period of the review and thus the new *Personal Pro* version was tested. Certain customary comments on box contents will, as a result, not be present in this review since these are likely to change drastically before the review is published.

The *KAV Lite* product is a minimalist offering and very much designed for home use, as a result it was not inspected except in regards to interaction with other *KAV* products.

Documentation and Help

Context-sensitive help within the programs is limited to floating information in the icon bar area. The information available in terms of the help function, however, is much more detailed and offers a useful array of detail about even the smallest features.

Finding the appropriate information can take some time due to the sheer volume available, and the addition of context-sensitive help would be a significant improvement. No hard copy versions of documentation were supplied.

On-line Resources

Kaspersky Lab has two main points of presence on the Web: the most obvious http://www.kaspersky.com/ and the less intuitive http://www.viruslist.com/. Both sites are administered directly by *Kaspersky Lab* and are in addition to the usual selection of VAR and partner sites (which were not inspected at this time).

The www.kaspersky.com site is typical of anti-virus vendor Web sites in content if not in initial layout. The home page is dominated by a large advertising byline, with a smaller pair of graphical links to (at the time of testing) information on the latest *Kaspersky* product and on the W32/Klez worm.

To the right of this graphical dazzlement is the much smaller and more restrained links area which leads to general site sections and a selection of news items. Every page seems to include advertising and a selection of example awards, certificates, user and media responses. At times the feeling of hard sell is somewhat overwhelming.

A limited selection of demonstration *Kaspersky* products is available at the site, though these versions lack disinfection as part of their deliberately limited functionality. Virus definition updates are supplied here too. As would be expected, these are limited in the case of a demonstration version inasmuch as the demo version cannot download updates (hopefully this should free up the download bandwidth for registered customers, not to mention being an incentive to purchase).

There is a whole host of purchase options available on the site, and the areas for partners and affiliates are rather more obvious than in most corporate Web sites. The obvious presence of such information shows a company dedicated to aggressive expansion, and this expansionist mentality is reflected in the *Kaspersky Lab* Mission to 'become the world leader in anti-virus software production' within five years.

The www.viruslist.com site is much less typical of current anti-virus sites, being more akin to some of those news sites which attempt to distinguish themselves by a certain degree of quirkiness. This manifests itself in such features as user polls with some bizarre stock answers (e.g. a 'what should be done to virus writers?' poll listed 'send them all to a desert island' as one of the available options) and a selection of Murphy's Law quotations, some of which border on the risqué.

However, the information provided at this site is very complete, and contains a long tract on virus history, summaries of most virus types and their properties and a vast collection of individual virus write-ups. For more immediate appeal there are daily news items – predominantly concerned with viruses, but covering security and general computing subjects in addition.

Two things struck me as significant about the contents of this Web site. The first is that, although it is not specifically declared to be a *Kaspersky* site, there are quite a number of

Kaspersky press releases and very few for other companies. This bias does detract somewhat from the usefulness of the site from the viewpoint of a user requiring a broad vista of industry information.

From a more devoted reader's viewpoint there are also some interesting quotations on the site if some of the articles are inspected carefully. Particularly impressive is a comment concerning the rise of polymorphic virus creation kits: '…lazy people join the ranks of virus makers, downgrading a respectable and creative profession of creating viruses to a mundane rough trade.' Surprising words indeed from the mouth of Eugene Kaspersky.

Checking with Eugene himself it was, thankfully, determined that this was a classic example of less-than-perfect translation of Russian humour (though there was a little disappointment here at *VB* due to the loss of juicy headline material).

Installation

With a server assumed to be the logical starting point for installation, the ultimate plan was to use the server to deploy to other machines while maintaining a minimum of physical contact with the target machines. One feature which soon became apparent is that the *Personal* version cannot be installed upon *NT* servers (though why anyone but a reviewer might wish to try this is open to question).

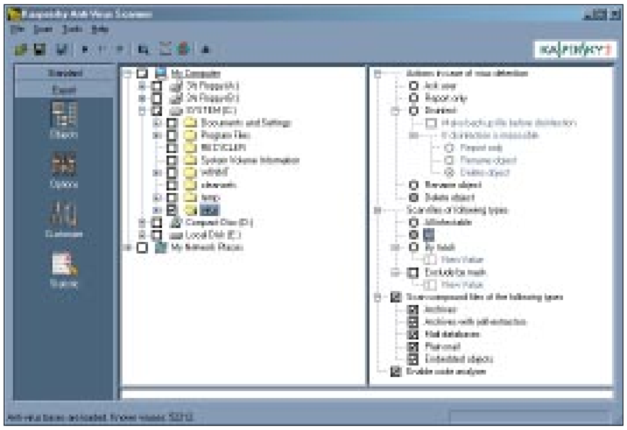
The *Personal Pro* version was first to be installed. Although using the ever-popular *InstallShield* package as a base for the installation process, *KAV* was at variance with the usual set of installation choices – offering Custom, Easy and Typical. There is a certain degree of strangeness in the Easy option, having the requirement to choose a non-default and thus less-easy option.

The Custom installation is customarily the most interesting of those on offer, and *KAV* proved no exception on that front. There are the generic on-access and on-demand scanners available here, in addition to update functionality, script checker, mail scanner, *Microsoft Office* file scanner, integrity checker and a control center to rule them all and bind their functionality. Rescue disks can also be selected for creation here.

The Easy option does not install the integrity checker, *Office* file checker or control center, and rescue disks are not produced.

The Typical option is identical to the Custom installation default with all components installed, though no rescue disks are produced, and this was taken as the tested configuration.

Moving through installation, the next option offered is whether to associate either .REP or .RPT files with the Report Viewer utility and a choice of where reports should be stored. After this the installation process is essentially over for a standalone machine. The Report Viewer is one of several utilities which are installed without being chosen during the component selection portion of installation.



Another of these default-installed programs is the virus list generator, which outputs a list of all malware detected by the currently installed *KAV*. These, together with the components selected to be installed, appear as accessible options via the Start button.

Updates and Upgrades

Updates are managed through the Updater or indirectly through the control center. In either case the process is very much transparent to the user unless errors occur.

A particularly good feature is the use of multiple servers for attempting updates. Half a dozen or so sites are used for these downloads and a failure on one update attempt simply triggers an attempt using another server. This default behaviour can also be tuned to update from a local folder or a dedicated *KAV* server and the choice exists as to whether the updates occur for Antivirus Bases, i.e. virus definitions, Executable modules, both or neither.

A set of tests were performed by installing version 3.5 of *Kaspersky Antivirus* and installing, on top of this, version 4. This was not only tested on the same product line but with all combinations of *KAV Lite*, *Personal* and *Personal Pro*. Upgrades from *KAV Lite* to *Personal* and from *Personal* to *Personal Pro* were also attempted. It soon became apparent that the three product lines must be considered as distinctly separate when considering upgrades.

In all cases it was possible to install version 4 of a product over version 3.5 of the same product. This was done through a request to uninstall the existing product where *3.5 Lite* was the base installation.

Where *3.5 Personal* or *Personal Pro* were being upgraded the process was much more fluid, with the older version being automatically halted and overwritten. In the latter two cases the existing configuration files for *KAV* could be preserved, overwritten, or in some cases merged with those provided by the new installation.

Attempts to upgrade from any version 4 to version 3.5 resulted in a declaration that this was not allowed – a situation not entirely conducive to rollbacks in case of problematic upgrades.

Rollbacks were tested by manually triggered uninstallation of the version 4 products before installing version 3.5.5. It was noted that this process left stray files from the newer version if potentially shared files were not selected for deletion – which resulted in strange graphical problems and intermittent instability in the second product installed.

It was also impossible to upgrade from any version of 3.5 to a different product in version 4 – with the exception of *KAV Lite*, which could be upgraded, though again this was by a manual triggering of the uninstall process.

Features and Interface

The *KAV* interface has a different feel from that of many products, though without straying so far from the normal control methods as to become totally incomprehensible.

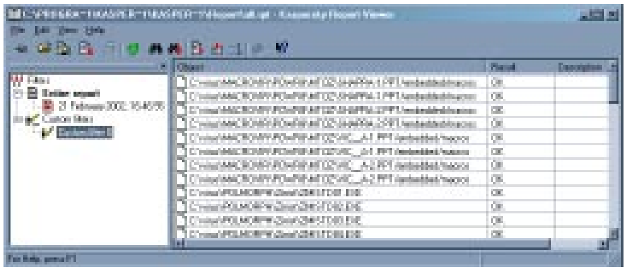
There is one feature which caused momentary confusion at the beginning of testing, at which point there appeared to be a disconcerting lack of control over, for example, which areas were to be scanned. This turned out to be a very easily solved issue, there being an advanced mode selectable for the interface. The method of entering this mode is through an easily missed button in the bottom left of the interface, hence the initial confusion.

The features here will come as no surprise, but the interface offers more of interest. Rather than the more commonly encountered dialog boxes, all interaction is performed on a tree structure of choices. This method of control works impressively well and is in many ways more convenient than the tabbed pages so often wrestled with in past reviews. This is probably a subjective opinion but it is hoped that real-world users will find the same.

Since the features are fairly standard and other more interesting subjects remain to be discussed, the subject of exact product abilities is rather more swiftly dealt with than usual.

Of the additional components in the installed product, one stands out as being rather more powerful than is generally the case – and of great use when reviewing. This is the report viewer, which as a separate module can be run alongside the on-demand or on-access scanners, or after the event in order to analyse reports from those sources.

More important to some will be the ability to set filters within the report file, so as to be able to display, for example, clean files or those logged as corrupt, while ignoring other categories. The filter mechanism allows for complex sets of conditions to be used and was a helpful first-line analysis tool when performing tests. Unfortunately, if understandably, unscanned files are not logged and thus old-fashioned log parsing techniques were used for final test results.



Detection Tests

With the recent Comparative's results still fresh from the press it might seem that there would be little to gain by re-testing the abilities of what is essentially the same product.

With a different platform as the primary test environment, and many settings to fiddle and experiment with, however, ample opportunity was found to produce new results for comparison. The only concern upon embarking on such an endeavour is the worry that no differences may come to light under any circumstances, giving little to discuss.

KAV allows for the alteration of two main scanning engine settings; one controlling which files are to be scanned (all or a predefined set), and another which activates or deactivates heuristics.

The default setting is to use the *Kaspersky Lab* set of potentially dangerous files and heuristics. In this particular test set it soon became apparent that the heuristics were not destined to play any part in the analysis – since all files detected were detected as a specific virus rather than as suspicious or likely to be a virus. The comparison is thus between the effects of scanning using the *Kaspersky* file list and the all-files option. In both cases the W32/Zmist.D samples were missed in their entirety.

More confusing was the matter of *Powerpoint* infectors. The .PPT and .POT files in the test set were marked both clean and infected by *KAV* simultaneously. Thus, scanning these files gave entries in the 'OK' and 'Infected' portions of log files. At first this was considered to be simply a matter of *Powerpoint* being treated as an archive – with the infected objects being identified as infected while the container is clean.

The confusion as to what to do with these files was not, however, restricted to the reviewer. When configured to delete infected files these *Powerpoint* files were ignored – though an instruction to disinfect the files resulted in a file which no longer registered as infected. This mystery was solved by the discovery of an option to enable deletion and renaming of infected archives, and thus the initial theory being proven.

This leaves only the files which were detected with a scan of all objects, but not when the *Kaspersky* set of potentially

Scanning Speed Tests	Default Settings		All files, Heuristics		Infectables, No heuristics		All files, No heuristics		Version 3.5.5	
	Time (s)	% of Default	Time (s)	% of Default	Time (s)	% of Default	Time (s)	% of Default	Time (s)	% of Default
Clean set	242	100%	268	111%	229	95%	256	106%	172	71%
OLE set	23	100%	23	100%	22	96%	23	100%	23	100%
Clean zips	137	100%	171	125%	132	96%	164	120%	106	77%
OLE zips	38	100%	42	111%	37	97%	42	111%	33	87%
Win2k	209	100%	224	107%	192	92%	210	100%	165	79%
Viral sets	438	100%	437	100%	421	96%	422	96%	332	76%

infectable files is used. The samples missed here were both .ASP samples of W32/Nimda.A. As an .ASP 'virus' this is not that great a threat – the .ASP in question simply appends to existing .ASP files and redirects to an infected file which will be detected by the scanner.

Speed Tests

Another facet of transition between the old and new *KAV* engines is the matter of whether these will have any noticeable effects for the user in terms of speed of scanning.

In addition to the four possible combinations created by the selections used during detection tests, it was decided to test the speed of version 3.5.5 in its default mode. The speed tests were performed against the four usual *VB* clean test sets, executable, OLE, zipped executable and zipped OLE, plus a clean *Windows 2000* machine and the full *VB* virus test sets. With this combination of information producing quite a spread of numbers it seemed wisest to tabulate the results.

The raw figures are given in terms of seconds to perform the tests – though, for reference, these have been displayed in terms of a percentage also. The base upon which these percentages are calculated is that of the speed test on *KAV 4* using the default settings for that platform. Some of the results were as anticipated – others were less expected, but followed a logical pattern.

The most disappointing from *Kaspersky Lab*'s viewpoint is that the new 4 engine performs distinctly more slowly that the 3.5.5 engine. The only area where this is definitely not the case is for the pure OLE set, thus it would be fair to suppose that changes were made mainly in the treatment of executable files and perhaps as a type of future proofing against perceived threats.

Elsewhere, removal of heuristics tended to speed up the product somewhat, while the scanning of all files tended to slow it down – no real surprise at all. Again, this was more marked where non-OLE files were included in the scanned sets. Also tending to increase the effects of heuristics was the scanning of files within .ZIP archives – where the engine is no doubt closer to its throughput limit when the additional loads of decompression are simultaneously in operation.

From the results derived on-demand it would seem that there is little need for removing heuristics to save time, since the time saved is minimal. It would also seem that there is no real advantage to not scanning all files – though this is a more contentious issue.

The *VB* clean test sets and the areas scanned on the *Windows 2000* machine were devoid of files which caused great overheads from all-file scanning, but this does not necessarily mean that all users would find the same to be true.

Conclusion

Kaspersky AntiVirus proved an easy product to review in most ways – the only major problem being that of new releases rendering parts of the review potentially out of date before they were published. While this may prove a problem from a reviewers point of view, it is a blessing from a user's viewpoint when the result is the addition of new functionality.

Overall, *KAV* performed well, with the only potential criticism being the slowdown in the newer version 4 as compared to version 3.5. The modular nature of the software provided numerous features which have not been covered in any depth here – I hope to return both these and the network functionality at a later date.

Technical details:

Product: *Kaspersky AntiVirus 4.*

Developer: *Kaspersky Lab*, 10 Geroyev Panfilovtsev St, Moscow, Russian Federation 123363; tel +7 095 797 8707 or +1 925 4634881 (US); email sales@kaspersky.com; Web site http://www.kaspersky.com/ and http://www.viruslist.com/.

Test environment: 750 MHz AMD Duron workstation with 128MB RAM, 8 GB and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *Microsoft Windows 2000 Professional*.

PRODUCT REVIEW 2

RAV AntiVirus for Sendmail 8.3 – Part 2

Matt Ham

In December 2001 *VB* brought you the first part of this *RAV for Linux* review (see *VB* December 2001, p.18), with the promise of more to come. After a good deal of wrestling with *Sendmail* that promise can finally be made good.

Changes

Over the intervening months the *RAV for Sendmail* products have seen a number of improvements. Of these, the most welcome in terms of reviewer satisfaction is the improvement in the documentation available. Although not epic in scope, the revised installation documentation contains those additional snippets of information which make all the difference.

As a prime example of this, exact details are supplied now as to the changes that must be made to sendmail.cf files in order to make the product operational. Previously this was described only at the level of sendmail.mc file editing, which caused great irritation on *SuSE Linux* since that distribution does not make use of m4 scripts but instead has a selection of pre-built sendmail.cf files. That is not to say, however, that the path to installation ran as smoothly as hoped.

The majority of problems came in the compilation of *Sendmail* with LIBMilter enabled, a process which required several packages not supplied as standard on *SuSE* distributions and prompted a move away to *RedHat Linux* – where LIBMilter is currently included as a standard feature.

Unfortunately, as noted in the previous part of this review, the installation of *RAV for Sendmail LIBMilter* was fraught with problems and the testing was performed on the older *RAV for Sendmail* product.

With the new documentation this process took about an hour – most of this time being spent double-checking alterations to the *Sendmail* configuration files.

This paranoia was possibly the reason why the email scanner worked on the first attempt. The settings in their default form are such that only a portion of the product's capabilities are enabled, and thus a fair amount of extra fiddling was required in order to test many of the features.

Configuration

The configuration for the mail scanner is registered in some not entirely easy to read configuration files. Clearly, *RAV* have realized that this might make configuration choices somewhat less than clear. Thus, for the checking of the current parameters, command line instructions can be given to *RAV for Sendmail*, which result in a detailed and human-friendly breakdown of exactly which options have been selected.

It quickly becomes apparent when this feature is used that the feature set is much greater than simply a virus checker – but those dedicated anti-virus features are a good place to begin.

As would be expected, there are options to allow for a variety of treatments when an infected object is found in an attachment. These include the usual deletion, quarantining and disinfection.

Since all actions must be totally automated for the scanner to be of any use, the options are selected in the order in which they should be attempted and messages may be set for failure at any stage.

For this particular mode of operation the default settings are complete and certainly would enable an easy first configuration for protection against viruses.

Options

Some of the associated default options are not, perhaps, so suitable. For example, for every virus detected an email is sent to administrator, recipient and sender.

While this might be acceptable in a small organization, in a larger company the volume of these mails could be excessive. In cases such as W32/Nimda, where much of the initial damage and disruption was due to the volume of infected messages, this might cause concern.

The matter of alternative methods of virus blocking is also taken up by *RAV for Sendmail*. Filters may be applied to block particular attachment names, subjects or body text.

These are applied as regular expressions, which allows for a large degree of control as to what is considered to be undesirable. As an example of a virus for which this feature might have proved extremely useful, W32/Nimda comes to mind again as far as file names are concerned, setup.exe being an easy target.

In a welcome departure from some desktop scanners, it is also possible to define undesirable objects as not being viruses and gain an appropriate classification in alerts.

Detection Testing

When handling single mail files sent at reasonable intervals the mail scanner proved able to detect all those samples which were detected also by the on-demand scanner of the same version.

Despite providing the scanner with attachments divided into chunks, detection did not seem impaired as the reconstituted attachment was used for scanning. Of course, with the large number of files in the test set, individual mails are not a feasible method of testing and an automated method was contrived.

For this process metasend was used to create an individual mail file for each virus in the *VB* test set. These files were then passed on to *Sendmail* through forwarding, so as to produce a rapid stream of infected objects.

Problems did occur, however, when this fully automated email sending was attempted. It was not clear exactly where the problem lay, since the mail client seemed to be suffering from instability due to volume of traffic. For future tests other methods are planned to overcome this potentially weak link.

Having been denied a mass of results from this source, the file name and contents scanners were put under scrutiny by way of consolation. These performed as advertised, leaving little more to discuss as far as the mail scanner is concerned.

Linux Malware

With the forthcoming *Linux* comparative review only a month away it is logical that *Linux* malware be added to the *VB* test sets. This review saw some, though admittedly not a vast number, of malware samples in the test sets.

The results here were slightly disappointing in comparison with the success rate on DOS and *Windows*-related viruses, though with no comparable data from other *Linux*-based scanners, in addition to a very small sample set, the level of criticism due is uncertain.

For these tests the command line scanner ravav was used from version 8.5-1, this being installed as part of the prerequisite ravcore package required for ravsendmail. Also installed and used by ravsendmail was ravmd-8.3-2.

On the positive side, the majority of those *Linux* worms and viruses which have made the news in recent times were detected. Such creations as Linux/Lion and Linux/Ramen fall into this category, having been newsworthy for a while and, the latter at least, common by *Linux* malware standards.

Of the remaining detections the levels of noteworthiness are distinctly lower. Such specimens as Linux.Lindose may have made their mark as concept viruses but cannot rank anywhere near having been a real world threat.

Overall, 14 detections were made. These were as follows: Adore, Bliss.A, Bliss.B, Kork.A, Lindose.2132.A, Lion.A, Lion.B, Mandragore.666, Ramen.A, Ramen.C, Silvio.6065, Silvio.7381, Telf.8000 and Vit. (The names used here are those given by *RAV* itself.)

Misses were encountered in a further two samples, Linux/NuxBee and Linux/Cheese. While the former is very much in the category of a zoo virus, the miss of the latter is more worrying. That misses did occur makes the forthcoming *Linux* comparative one which might prove less predictable than some of those of late.

Conclusions

The two parts of this review taken as a whole illustrate some interesting changes within *RAV*'s *Linux* offerings over the period covered.

Criticism was levelled at the earlier products for lack of documentation, yet this area has shown great improvement. The same improvement was apparent in the ease of use of the product, with a number of small changes adding up to a considerable improvement.

Although installation still proved a challenge where *RAV for LIBMilter* was concerned, this was due more to the distribution used for this purpose than the installation information being incorrect.

This problem of distributions may prove to be one of the greater issues when installing anti-virus software – since environment, application version and installed packages are not predictable to any reasonable degree. The later versions of *RAV* pay greater attention to this fact, but it is quite clear that attempting to use anything other than the most common distributions could be inviting problems.

This is almost certainly a general problem, rather than *RAV*-specific, but there is a more helpful aspect of the same phenomenon.

With application portability problematic even for legitimate applications, viruses and worms might be expected to encounter even more problems. User editing of config files is more likely an event when inspired by a manual than by malware.

Technical details:

Products: *RAV AntiVirus for Sendmail 8.3*, *RAV AntiVirus Desktop for Linux 8.3*.

Developer: *GeCAD Software*, B-dul Mihai Bravu, nr. 223, Optodol Business Center, et.2,sect.3, Bucaresti, Romania; tel +40 1 1321 78 03; email sales@gecadsoftware.com; Web http://www.ravantivirus.com/.

Test environment:

Servers: Two 750 MHz AMD Duron Servers with 128 MB Ram, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy.

Operating systems: *SuSE Linux 7.2 Professional*, *Kernel 2.4.4*, *glibc 2.2*; *RedHat Linux 7.2*, *Kernel 2.4.7*, *glibc 2.2.4*.

Virus test sets: Complete listings of the test sets used are at
http://www.virusbtn.com/Comparatives/WinNT/2001/08testsets.html
http://www.virusbtn.com/Comparatives/WinME/2001/12testsets.html.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 9731266, Fax (781) 9731267

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

END NOTES AND NEWS

CeBIT 2002 runs in Hannover, Germany, from 13–20 March 2002. Easier navigation and a more compact, convenient layout is promised for visitors to CeBIT 2002. For more information or to order tickets online, see http://www.cebit.de/.

Cost-Effective Risk Management for Information Security takes place at the Café Royal, London, 19–20 March 2002. Corporate case studies examine the strategic issues surrounding cost-effective risk management for information security. For more details visit http://www.iqpc.co.uk/GB-1759/ediary/.

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

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

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

Infosecurity Europe 2002 will run from 23–25 April 2002 at London's Grand Hall, Olympia. For more details see this issue p.5 or visit the Web site at http://www.infosec.co.uk/.

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

Infosec 2002 takes place 28–30 May 2002 at CNIT, Paris La Défense, France. This three-day event will run concurrently with SIMBIOM, the First International Biometry Exhibition. For more information, including an exhibitor list and details of the conference and tutorials, visit http://mci-salons.fr/infosec/.

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

Papers and presentations are now being accepted for the Black Hat Briefings 2002 conference. The conference is to be held from 31 July to 1 August, 2002 at the Caesar's Palace Hotel in Las Vegas, USA. Submissions must be received by 1 May, 2002. For further details of the conference see http://www.blackhat.com/.

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 9th International Computer Security Symposium, COSAC 2002, takes place 8–12 September 2002 at Killashee Hotel, County Kildare, Ireland. Cost of registration will include 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 in New Orleans, USA from 26–27 September 2002. Watch out for the full programme details at http://www.virusbtn.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/.

F-Secure and *CyberGuard* have announced that ***F-Secure*'s anti-virus technology will be integrated with *CyberGuard*'s family of firewall/VPN appliances**. See http://www.F-Secure.com/.

Following its move to the New York Stock Exchange last month, ***Network Associates* has changed its ticker symbol to 'NET'**. For more details see http://www.nai.com/.

The pizza delivery company *Domino's Pizza* is to use *Sophos Anti-Virus* on the computer networks of its UK and Irish master franchisee. The chain has almost 240 stores across the UK and Ireland. See http://www.sophos.com/.