ISSN 0956-9979 MARCH 2000

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Limited editions:** Make sure you don't miss out on the chance to own one of *Virus Bulletin's* 10 year anniversary T-shirts. Details of how to order and pay are on p.3.

• **Is it a bird? Is it a plane?** Our corporate case study this month comes from *Boeing's* Computer Security Product Manager, starting on p.12.

• **Denial of service with a smile:** Nick FitzGerald explains the latest menace and what you can or cannot do to avoid it, starting on p.16.

• **Exchange & mark:** We kick off our new series of groupware anti-virus product reviews with *GroupShield for Exchange* on p.20.

CONTENTS

COMMENT

“ Hoaxes are just as demanding on resources …”

What Support Technicians Really, Really Want

The concerns of the anti-virus industry differ from those of the corporate technical support community in many ways despite the fact that both are dealing with the same basic problem – malicious disruption of computer use. Thanks to the excellent work being done by anti-virus companies, technicians have very effective tools. This enables them to focus on the effective use of the tools rather than how well the tools work.

Anti-virus vendors go to great lengths to show consumers that their product outperforms those of their competitors. *Virus Bulletin* and its supporters do an excellent job of documenting the actual performance differences. However, the support technician, who has no choice but to use his company-designated software, is not concerned with the number of stickers on the box or whether one product incorrectly identified variant number N.XXX of the test virus database. If their company anti-virus product cannot disinfect the virus that just made it through his company firewall, then they are only concerned with getting an update file that can.

Any anti-virus product major enough to be selected as company-distributed software will recognize and repair the majority of known viruses. The technician still has to deal with viruses, but only relatively few. An automatic update feature can make this job easier.

However, it may not always work. Sometimes users see a message that their update failed to run unattended. Sometimes the automatic update gives every appearance that it has been working but it has not. The messages are frequently ignored, and the update schedulers are frequently turned off. The anti-virus company can best help by providing an easily accessible Web site containing timely updates that can be downloaded manually then accessed from an in-house server.

There is room for improvement in the scheduling programs that are used to run scans and updates automatically. These need to be integrated into the various operating systems used by businesses so that they run, consistently and automatically, in the background. For instance, in *Windows NT*, this should run as a service, always available. Simply running it minimized on the taskbar makes it too vulnerable to being turned off.

Corporate computer users still manage to get infected with viruses and sometimes the result is a lot of work for the technical support staff. During the hectic Y2K rollover scramble in December a little virus we were all warned about, named W97M/Thus, fired off its file-deleting payload right on the day we were told it would. Support staff already working overtime to meet the Y2K deadlines had to handle the emergency calls that resulted.

A support technician in a large company has to be an optimistic person in order to keep going. The silver lining in virus incidents such as Thus and Melissa is that they force us to update our protection and strengthen our vulnerable areas. In the case of Melissa, it did so just in time to avoid a potentially worse situation with CIH.

Hoaxes are just as demanding on resources as actual viruses. No amount of hoax database Intranet Web pages, hoax-identifying educational presentations, or simple pleading will stop the hoax warnings from circulating within companies. I have seen them in mass email from the data security department itself. People just love the opportunity to be the one to cry wolf. If you do not think hoax warnings are a big deal, take a look at the newest version of ‘It Takes Guts to Say Jesus’, the one that frantically begs the user to mail a copy to everyone in their address book (a *VB* sure indicator of a hoax).

How much worse is a Melissa-type virus that emails itself to everyone on your mailing list than a hoax that gets you to do it manually?

Daniel D Diefenderfer, Dun & Bradstreet Corp, USA

NEWS

Order Yours Now

Virus Bulletin has commissioned a limited number of 10th anniversary T-shirts featuring our 'anti-virus made easy' logo. The T-shirts are available in white, size XL only and cost £15 or US$25 each.

Payment must be in full and by credit card only. Please contact Bernadette; Tel +44 1235 555139, email your order to bernadette@virusbtn.com or fax the details to +44 1235 531889 ▮

That's a First, Again

Back in December 1997 we featured a story in this column about Israeli company *iRiS* announcing the world's first known *Windows CE* virus scanner. In February 1998's Endnotes & News page, we pointed out that *NAI* had announced the second 'first Windows CE scanner'.

Now *Computer Associates* (owner of *iRiS*) seems to have 'rediscovered' the *iRiS* scanner and has struck a deal with a *Windows CE* manufacturer to bundle said scanner with their machines. Naturally, *CA* has also seen fit to make the *third* announcement about the first *Windows CE* scanner – see http://biz.yahoo.com/prnews/000125/ny_ca_symb_1.html. Strictly speaking, *CA* is right – it now owns what really *was* the first *Windows CE* scanner.

Aware that there are no existing virus threats to *Windows CE*, the organization has included this beautiful statement in its press release – 'CA is proactively ensuring that these devices are protected on the Windows CE platform from viruses and malicious code…'. Call us cynical but our translation is 'buy our anti-virus software now in case there are virus issues with this platform in the future'. To describe as 'proactive' the provision of an inherently *reactive* 'solution' in advance of the problem must be a contender for the 'doublespeak of the year' award ▮

The End is Nigh!

The January 2000 *Dr Solomon's Anti Virus Toolkit v8.02 Update* landed on the *VB* doormat a few weeks ago, complete with a detailed reminder letter that the April 2000 release (v8.04) will be the last.

It will be interesting to see where current *Dr Solomon's* users take their business after the last monthly update has arrived. Promises of a simple transfer to the *Total Virus Defense Suite* are rife and reassuring. The problem is that many of the other AV companies are offering equally painless migration. Time will tell ▮

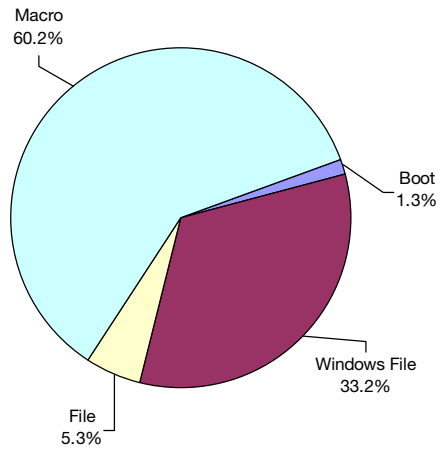
Prevalence Table – January 2000

Virus	Type	Incidents	Reports
Win32/Ska	File	140	13.3%
Laroux	Macro	124	11.8%
Marker	Macro	122	11.6%
Ethan	Macro	69	6.5%
Class	Macro	59	5.6%
Win32/Pretty	File	52	4.9%
Win32/NewApt	File	48	4.6%
Melissa	Macro	47	4.5%
Tristate	Macro	44	4.2%
Freelinks	Script	42	4.0%
Win32/Babylonia	File	32	3.0%
Cap	Macro	31	2.9%
Win32/Fix	File	31	2.9%
Thus	Macro	29	2.8%
Win32/ExploreZip	File	21	2.0%
Win95/CIH	File	19	1.8%
ColdApe *	Macro	18	1.7%
Story	Macro	13	1.2%
Fool	Script	9	0.9%
Pri	Macro	9	0.9%
Evolution	Macro	8	0.8%
Form	Boot	7	0.7%
Others [1]		80	7.6%
Total		1054	100%

[1] The Prevalence Table includes a total of 80 reports across 36 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

* In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 1055 reports in January) have been omitted from the table this month.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

Punitive Damages?

By how much are we better off as a result of the prosecution and imminent sentencing of David Smith? There is an argument, which I do not dismiss, that all crime deserves some form of punishment because, as a society, we need to declare what is acceptable and discipline those who don't conform.

The law separates crimes into different categories and takes different approaches accordingly. Crimes in which the perpetrator is motivated by personal gain seem to be the most harshly punished. Thus, punishment is clearly set to deter (amongst other things) other criminals from doing the same thing.

It is difficult to see how David Smith's punishment (any more than Christopher Pile being jailed in December 1995) is going to deter other virus authors, or would-be authors, from writing and releasing viruses.

Of course, very harsh punishment administered in front of a baying crowd may have more effect, but since we don't do this sort of thing any more (and I can't see a crowd being sufficiently incensed as to want to turn out in numbers and bay for blood), we are left wondering what positive benefits come from Smith's arrest and prosecution.

One of the big problems in the David Smith case is that he has become notorious, although this has a lot to do with the effect of his virus on the rest of us before he was 'captured'. There are clearly opportunities (and America is the land of opportunity) for Smith to benefit from his notoriety and that may encourage other virus authors to do likewise. Hardly the effect we might want.

In the early days of anti-virus development, products flashed alarms when they discovered a virus. Crowds gathered around the PC and much time was wasted. These days, the alarms are delivered elsewhere and problems are often fixed quietly with little disruption for the user.

Maybe law enforcement (and the general media) need to learn the same lesson.

Paul Robinson
Secure Computing magazine.
UK

At the Mercy of Microsoft

Mr Urzay's article (*VB*, January 2000, p.22) is indicative of the anti-virus community's reactive response to *Microsoft's* philosophy of imposing its standards on the user. Take the use of VBA for scripting *Word* macros as an example. *Microsoft* implements VBA in *Word* without giving the user the option of turning it off. If we were given that option, we would turn if off and VBA would not be used. A technology must be used to become a *de facto* standard. *Microsoft* wants its technology to become a *de facto* standard. *Ergo*, don't let the user turn it off.

It makes no difference that the technology in question enables the macro virus. That is not *Microsoft's* immediate concern. Promulgation of its standards is. The anti-virus community is forced to react to this situation by creating detection/disinfection mechanisms of ever-increasing complexity. *Microsoft* responds by making the technology more powerful, more portable, more ubiquitous which gives virus authors more tools and opportunities for their creative imaginations. The cycle repeats.

It must be noted that *Microsoft* is also a part of the reactive cycle, first with the *SCANPROT* macros, then with the 'Enable macro virus protection' option, and now with the idea of 'trusted' macros. *Microsoft's* reactions are merely *Band-Aid*™ solutions to the problem of macro viruses since the underlying technology which enables them is untouched and cannot be disabled.

Add other *Microsoft* scripting standards to the mix such as VBS, ActiveX, and *Microsoft's* own implementations of HTML and Java, none of which can be turned off, and it is no surprise that we have our current situation. With reference to the old Chinese curse, how else can *Microsoft* make our computing experience more 'interesting'?

Henry V Taber
National Imaging & Mapping Agency
USA

Quality QA

The sad fact is that, like some software developers, virus writers do not seem to QA their creations. Why not? I do not know, it could be that they are too lazy/ignorant/scared, but that is a question for another day.

We (the AV industry and the general public) should be thankful. The number of trivial bugs I see in otherwise well-written viruses is amazing. A case in point has to be the recently discovered W97M/Sylko.A, an intentionally parasitic macro virus. We have seen other pieces of VBA code that create 'sandwiches', but they have been brought about through ignorance rather than cognitive reasoning.

Cutting to the chase, if the virus detects that it is running from the Global environment and the document is not already infected, the following occurs. The virus checks the first line of code in the 'This Document' module. If the line

matches, the virus insert adds it own code but leaves the original code in the routine (about 16% of *Word 97* viruses do this). So, when the Sylko routine finishes, the original code then runs.

This infection method works beautifully provided that the check for the first line of code is successful. If that check fails then the code fails and we get a nice, familiar error box. In fact, 'fails' is not the best way to put it, catastrophically crashes would be nearer the mark!

Such a basic bug in the code is inexcusable and the fact that *Word* does not handle it very well is regrettable. The main consequence of this bug means that the virus should not spread in the wild. This is, obviously, a good thing – so maybe the lack of QA is to be praised (but only with regard to virus writers!).

Paul Baccas
Sophos Plc
UK

Happy April Fool's Day!

It was with great shock and dismay that I read Vincent Gullotto's letter 'Crying Wolf!' in February's issue.

Mr Gullotto seems to think anti-virus companies have some antiquated sense of obligation to behave responsibly or, perhaps even worse, ethically not only to their competitors, but also worse, to their customers. Such misguided notions should have long ago been tossed in the dustbin of history. Nothing could be further from the truth.

As any first-year business student should be able to tell you, the goal of any business is to create value for its stockholders, or, ideally, in the event they have issued public-traded stock, to increase earnings per share. Anything else is a distraction, and, frankly, irrelevant.

By continually issuing warnings about computer viruses, anti-virus businesses have a unique opportunity to grow their brand. This increased demand for anti-virus products – preferably theirs, of course – allows them not only to grow their revenue, but to provide demi-related goods and services as well. A company which cannot unfocus its interest and instead remains laser-sharp on one specific solution will rapidly find itself losing shelf-space at computer superstores and retail outlets as their branded cohorts fill up the aisles with boxes of gold and platinum and professional editions of their software.

VB's readers can only imagine the horror of a world in which anti-virus solutions had to be judged by their technical merit – detection, removal, compatibility, usability – instead of brand recognition. All of these strike terror in my heart. What would happen to earnings per share if income from profitable update and bug fix services and lucrative per-incident support contacts faded? What good does this do the shareholders of antivirus companies? The only people who profit from such behaviour are *customers*.

Companies like *Computer Associates* and those before them who took on the difficult job of increasing their value deserve to be lauded and awarded for their brave efforts, not criticized!

Aryeh Goretsky
Colorado, USA

Gods & Monsters

The Korean cyber criminal investigation team, part of the National Police Agency (NPA), announced on Thursday 17 February that a 15 year-old middle-school student had been booked without detention for writing a worm program. He had spread his creation indiscriminately by posting it on a popular computer magazine Web page, disguised as a free updating program. When the attached EXE file is executed, the worm gets control. The infection routine opens the *Outlook Express* database, retrieves email addresses from the Address Book and then sends infected messages to the addresses found. It has a very dangerous payload routine. On the 31st day of every month, it overwrites C:\AUTOEXEC.BAT files with a command that formats the C: drive. However, it turns out that the worm needs VB6KO.DLL and MSVBVM60.DLL in order to be activated. So, it seems that it will not have much chance of getting into the wild.

The student author took computer lessons at a private institute for a year, and after gaining more computer knowledge through the Internet, he 'made' the virus in just five days. He told the police that he did it because he wanted to check how good his skills were. I am worried – our police have arrested Korean virus authors three times and every time the national mass media has portrayed them as either a hero or a genius.

Jacky Cha
Dr Ahn's Laboratories
South Korea

Doubtful Disclosures

While AV vendors fight malicious software every day, sometimes it isn't clear what we should be fighting. Visit http://www.gohip.com/freevideo and you will be prompted to download an 'enhancement'. This executable changes your home page to *GoHip's* site, modifies your *Outlook* signature file to include a *GoHip* advertisement, and on reboot, reapplies the changes. Such actions are all detailed in fine print, but who reads the fine print? Thus, one security vendor detects this as malicious code and another describes it as a 'joke program'. I ask that, instead of forcing security companies to make judgements on these programs, software vendors be more responsible in their design and disclosure policies. Otherwise, expect to find your rose labelled a weed.

Eric Chien
Symantec
Netherlands

VIRUS ANALYSIS 1

20/20 Visio

Andy Nikishin & Mike Pavluschick
Kaspersky Lab, Russia

At the end of January 2000 virus-related news sites reported a new virus for *Visio* as '… the very first macro virus for a non-Microsoft application'. However, *Microsoft* had acquired *Visio Corporation* just a few days before and two *Visio* viruses had been written at much the same time.

What is Visio?

Visio is an application – popular among students – which creates various diagrams and schemes. From version 5 it not only incorporates Visual Basic for Applications (VBA) programming language as an automation tool, it also provides the opportunity to write macros to simplify operations. Various *Visio* events can be assigned to these macros, such as Document Open, Close, Save etc. The similarity to *Office* applications is obvious – a simple but powerful programming language, automatic macro execution without notification, and macros stored inside documents. This is enough to facilitate viruses for *Visio* – it was only a matter of time.

Recently, virus writers have tired of waging endless war on the *Microsoft Office* front. Looking for new platforms for macro viruses they found *Visio*. With their experience in writing viruses for *Office* using VBA, it was easy for them to port viruses from *Word* (for example) to *Visio*. Sometimes the only change required is the event name assigned to the virus procedure – V5M/Radiant is a good example.

V5M/Radiant

V5M/Radiant infects *Visio* documents, stencils and template files (if they are open) on closing. It contains one procedure assigned with the 'BeforeDocumentClose' event. When the virus gets control it numbers and infects all opened documents. Due to the internal structure of *Visio* Radiant is able to number not only document files but stencils and templates too. When the user opens or creates a new document the application also opens the template file and a number of 'stencils', which are already infected. It is precisely this feature of *Visio* which enables Radiant to spread through files very quickly.

The virus' payload procedure features the message shown here. On every launch it creates an INDEX.HTML file in the root directory of the C drive. At the very end the virus has a small comment containing some symbols, rumoured to be encrypted information about the author of this virus. However, the type of cipher and the key used for encryption of the text string are unknown.

V5M/Unstable

This second virus also has pretensions to be the first in its class. It is more serious and complex, using encryption and special tricks to hide its body in infected files. This macro virus infects *Visio's* documents, stencils and templates, upon opening an infected document. V5M/Unstable contains three procedures in its 'ThisDocument' module, namely 'Document_DocumentOpened()', 'Unstable()' and 'ci()'. Inside infected documents the second procedure is unreadable because of encryption.

The first procedure is the main one and receives control when an infected document is opened. The virus decrypts its second procedure (using the third as the en/decryptor) and calls it. Control is passed to the decrypted procedure because VBA compiles procedures only when they are called. Thus, it compiles the already decrypted procedure.

At the very beginning the second procedure generates a random number and encrypts itself again using this random key (some *Word* viruses use the same trick). Then the virus numbers all opened documents, stencils and templates and infects them by copying its (previously encrypted) body into them. The virus adds 'Visio2k.Unstable' to the description of infected documents and does not re-infect them. It closes all opened windows inside the VBA Editor, disabling Visual Basic Editor's menus and the 'Standard' toolbar to hide its body. When users try to see macros inside infected documents they will see an empty Editor window without any menus, toolbars and child windows.

The virus has a payload that triggers on the 31st day of the month, displaying the message:

```
Visio2000.Unstable
Unstable, it's hard to be the one who's
strong
Who's always got a shoulder to cry on
Who's got a shoulder for me?
```

Conclusion

Both viruses are quite similar to many other known macro viruses. In going outside *Microsoft Office*, they prove that it is not hard to port VBA viruses to any other application that uses VBA (the *Microsoft* site reports more than 100 companies already licensed to use VBA in their products). It is rumoured that *Visio* will be included in the next version of *Office*. It *is* a powerful and useful product, but it is also a good platform for macro viruses. What next? Be ready for a new wave of macro viruses. Some say that there is already a virus for *AutoCAD* – but that is another story.

VIRUS ANALYSIS 2

Kak-astrophic?

Vanja Svajcer
Sophos Plc

Until recently, anti-virus experts advised that a computer could only become infected if an email attachment is opened or launched – simply viewing the message was considered safe. The appearance of VBS/BubbleBoy and a number of viruses exploiting an *Internet Explorer 5* (*IE5*) security vulnerability proved them wrong.

Although VBS/BubbleBoy has yet to be seen in the wild, the possibility of infection by viewing an email is now a reality for *IE5* users. The *Sophos* virus lab has received a number of reports about a worm, reputed to be in the wild, named VBS/Kakworm that also exploits this vulnerability. Unlike BubbleBoy, which uses *Outlook 98/2000* to spread, Kakworm uses *Outlook Express*, a part of default *IE5* installation. This fact may improve its chances to spread given the probability that *Outlook Express* is the mailer of choice for many *IE5* users.

Arrival

Kakworm arrives in an email that appears to be a plain HTML message without any visible signs of a viral presence. As soon as the message is opened, or even viewed in the preview pane, the ActiveX embedded code launches itself. The viral code exploits a well-known *IE5* security vulnerability – the fact that Scriptlet.Typelib ActiveX control is marked 'Safe for scripting'. The control may create files on the local hard drive, even if a remote script calls it. This vulnerability can be exploited only if the Internet Zone Security option is set to 'Medium' or 'Low'.

The worm uses the control to drop a file called KAK.HTA into the C:\WINDOWS\Start Menu\Programs\Startup folder, allowing the file to run at the next *Windows* bootup or a user-login. It also creates a randomly named hidden copy of KAK.HTA in the C:\WINDOWS\SYSTEM folder. The HTA (HyperText Application) extension represents files in *MS* binary format, containing standard HTML code. This extension is associated with a program MSHTA.EXE, which handles the execution of these files. The next time *Windows* is started, KAK.HTA runs and prepares the environment for spreading.

It first drops a hidden file C:\WINDOWS\KAK.HTM, the pure HTML version of the worm. It then copies the C:\AUTOEXEC.BAT to C:\AE.KAK and appends the C:\AUTOEXEC.BAT with the code to first execute and then delete the KAK.HTA file. Next, it changes the *Outlook Express* Registry settings, so that the KAK.HTM is automatically included as the user-signature in all outgoing messages. KAK.HTA drops the KAK.REG file into C:\WINDOWS and runs the REGEDIT.EXE with the KAK.REG as an input file. If a message is sent using HTML format, the worm will be a part of the message. If a message is sent using plain text format, it will be attached to the message as a standard ATT1.HTM attachment file.

The other Registry key altered by the worm is the well-known …\WINDOWS\CurrentVersion\Run. The worm adds a new value – 'cAg0u' – to the key and sets it to point to the hidden HTA file in the …\WINDOWS\SYSTEM folder. This alteration ensures that the file runs on *Windows* startup and re-infects the environment. Finally, the worm checks the system time and date to set up the payload. On the first day of any month after 5pm, the worm displays a message box and runs RUNDLL32, shutting down *Windows*. It cannot run under *Windows NT* due to the fact that it uses hardcoded path to the startup folder.

Removal and Prevention

All worm files must be deleted, and *Outlook Express* settings referencing the worm signature file must be changed to prevent *Outlook Express* from displaying error messages (error messages may be displayed when the 'signature' file is missing).

Users have been urged to apply *IE5* security updates and to upgrade their browsers regularly in order to prevent potentially malicious code from exploiting vulnerabilities. Alternatively, users may set the Internet Zone Security options to 'High' in order to prevent any ActiveX control from being regarded as 'safe for scripting'. This way, none of the potentially malicious code can be executed directly from an HTML page. It seems, however, that a significant number of users disregarded these warnings and, consequently, have been infected by the worm – proof, if it were needed, that this manner of virus attack is in now the wild.

VBS/Kakworm

Aliases:	VBS/Kak, Wscript/Kak.A, Kak/Worm, Wscript/KakWorm, Kak.
Type:	Worm.
Spread:	Via email. Exploits *IE5* Scriptlet.Typelib security vulnerability.
Payload:	On the first day of any month after 5pm displays a message box and causes *Windows* to shut down.
Removal:	Delete all worm files and change *Outlook Express* signature settings (see text for details).

OPINION

Add-in Insult to Injury

Gabor Szappanos
Computer & Automation Research Institute,Hungary

Despite a uniform VBA language, it used to be that there was no common macro storage format that could be processed by each of the *Office* applications. However, with *Office 2000's* COM add-ins, it is possible to create a single solution for use in any *Office* application.

Creation and Installation of Add-ins

A COM add-in is simply an ActiveX DLL or EXE that implements an IDTExtensibility2 interface; thus, any tool that supports ActiveX development can create them. Since *Office 2000 Developer* (a COM add-in development extension) is capable of creating only ActiveX DLLs, as opposed to EXEs, we will discuss only DLLs that act as in-process COM servers.

The IDTExtensibility2 interface has to implement five functions (although VBA will create stubs for any left unimplemented) as shown in the table overleaf. In theory, when an add-in is loaded or unloaded, the OnAddInsUpdate event occurs in previously loaded add-ins. However, when I created some different *Outlook* add-ins, loading add-in 1 fired the 1_OnConnection event. Loading add-in 2 fired the event sequence 2_OnConnection>2_OnAddinsUpdate. This is more than suspicious given that the latter event should occur in the previously loaded add-in 1. Due to an indexing problem, when an add-in is loaded the OnAddinsUpdate event chain is started from the *second* loaded update. This is not *Outlook*-specific and each *Office 2000* application should have the same problem.

This indicates a possible security hole. A hook in the event chain is the same as hooking an interrupt. Even if a virus protection add-in is loaded, the last Trojan add-in to be loaded has the chance to rehook everything! Otherwise, all the add-ins are automatically arranged in alphabetical order and the event chain proceeds in this order. These DLLs are very loosely connected to the host *Office* applications. This connection is established via two sets of Registry keys. The first set declares that an add-in is loaded to the host:

```
REGEDIT4
[…\Word\Addins\WebPage.Connect]
"FriendlyName"="Web Page Wizard"
"LoadBehaviour"=dword:00000009
```

'FriendlyName' is the name of the add-in as it appears in the COM add-ins dialog (and the link to the second set), 'LoadBehavior' specifies how the add-in should be loaded: 0 (Disconnected), 1 (Connected). These settings can be combined with the following behaviour settings: 2 (Startup), 8 (Load on Demand), or 16 (Load At Next Startup Only). To connect the COM add-in, set the Connected flag in LoadBehaviour and clear the flag to disconnect it. Keys are stored under HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER.

This set registers the add-in as an ActiveX component:

```
REGEDIT4
[HKEY_CLASSES_ROOT\CLSID\{E5670E37-0D2F-11D2-
9E65-00A0C904DD32}]
@="Web Page Wizard"
[HKEY_CLASSES_ROOT\CLSID\{E5670E37-0D2F-11D2-
9E65-00A0C904DD32}\InprocServer32]
@="D:\\Program Files\\Microsoft
Office\\Office\\WEBPAGE.DLL"
[HKEY_CLASSES_ROOT\CLSID\{E5670E37-0D2F-11D2-
9E65-00A0C904DD32}\ProgID]
@="WebPage.Connect"
[HKEY_CLASSES_ROOT\CLSID\{E5670E37-0D2F-11D2-
9E65-00A0C904DD32}\Version]
@="2"
```

These keys link the add-in to the actual file that implements it. The ProgID key is important as it specifies the VBA reference name that can be used to access the add-in. The installation of a COM add-in is a simple matter of creating the above Registry keys – achieved by calling the appropriate Windows API functions, or importing the above text files into the Registry. The REGSVR32.EXE *Windows* program can create the second set automatically

A single DLL can serve as a COM add-in for several *Office* applications. Even the project files containing the code can be the same. The only important point, from the developer's point of view, is that a specific add-in designer is provided for each host application. This specifies the load behaviour of the component and the procedures that implement the IDTExtensibility2 interface. It is quite possible to create an add-in that loads automatically in *Word* and on-demand in *Excel*. The good news is that exactly the same macro security settings apply to COM add-ins as to application-specific macros. With the highest security level selected, only the components digitally signed by a trusted source can be executed.

However, in the Tools/Macro/Security dialog, one can select the 'Trust all installed add-ins and templates' check box. Then *Office 2000* applications will load all COM add-ins, application-specific add-ins, and templates in trusted folders without checking to see whether they have valid digital signatures from trusted sources.

Add-in Initialization Tasks

COM add-ins can be loaded in two ways. Manual loading requires the built-in COM add-ins command, available in each major *Office* application. This command is not present on the command bars by default, so the menu and command bar settings must be customized.

Automatic external loading is performed by creating the necessary Registry keys (see above). Upon startup, the host application acquires the COM add-in information and, if an add-in is registered to start up, automatically loads it. If an add-in is registered to be loaded on-demand, it will only be loaded when the assigned menu item is selected.

Event	Occurs when:
IDTExtensibility2_OnConnection	the host application loads the add-in
IDTExtensibility2_OnDisconnection	the host application unloads the add-in
IDTExtensibility2_OnStartupComplete	the host application completes its startup routines
IDTExtensibility2_OnBeginShutdown	the host application begins its shutdown routines
IDTExtensibility2_OnAddinsUpdate	the set of loaded COM add-ins changes

The OnConnection event is triggered when an add-in is loaded, and the appropriate event function is executed. This function receives a handle to the global application object as a parameter, so at this point the add-in has full control. The preferred scenario is to set up custom menu items and hook the application-level events to the appropriate add-in handlers. After establishing a connection, the add-in has exactly the same capabilities as any legitimate native macro program (or virus).

Office 2000 hooks into the CommandBarItem click event before the host application receives the event. If the OnAction string points to a COM add-in, the ProgID is converted to a CLSID. *Office 2000* checks through the Registry for the appropriate registered COM add-in and passes the click event to it. Otherwise, the host application handles the event. If the OnAction string does not match any available macro name, the host application displays an alert. The menu settings can be stored permanently in the application's global storage (for 'Load at next startup only' add-ins) or set only for this session, removed during shutdown and recreated on startup (for Startup add-ins).

Possible Attack Scenarios

The complex structure of ActiveX DLLs means it is unlikely that parasitic COM add-in viruses will pose a significant problem. However, this technology is ideal for developing worms. COM add-ins developed in C++ can overcome the limitations of the Basic programming language in order to develop highly sophisticated programs. It is as easy as coding a *Word* macro virus (the source code can even be VBA). Using the full features of Automation, these applications can automate any *Office* application.

A worm could use the code snippet from Melissa to send itself by email. Hooking events in *Office* applications is easier than patching system DLLs and hooking *Windows* API calls. It is just a question of time. It could be more dangerous than Melissa, which was (from the *Outlook* propagation point of view) a direct action virus. A malicious COM add-in could easily hook itself to the NewMail application event, thus going *Outlook*-resident. Whenever the arrival of new mail raises this event, it can mail itself back to the sender.

A malicious COM add-in can be installed by a simple file copy operation to implant the file on the target PC, then the creation of about a half dozen Registry keys (or by importing a Registry file) – all perfectly legitimate moves by any software installation routine.

Until now, cross-application macro viruses had to carry application-specific code segments and infect each *Office* application in a slightly different way. Just a couple more Registry key settings and the *Word* COM add-in will be loaded in *Excel*, too. Of course, the add-in has to include the code for the *Excel* action hooks, but most of the code is common anyway, and it can target four or five *Office* applications at the same time with a single file.

It is possible to attack *Office 2000* even before it is installed on the system. A dropper for COM malware can 'install' itself without the host application's presence, as only the Registry keys have to be created. If the host application is installed later, the injected malware comes alive – the 'sleeping Trojan' concept. The only challenge is getting the user to run the worm's install part – experience shows that it is all too easy to persuade users to execute attachments.

Challenging the AV Developers

Should COM malware ever be created, it would not be all that easy to track down. A properly installed COM add-in does not affect the host application's integrity. A couple of obscure Registry keys can lead to the malicious program which might be deeply hooked into the host application's events. The intrusion cannot be detected by behaviour-blocking programs as nothing 'illegal' is happening. Scanning an ActiveX DLL poses the same problems as scanning any other high-level language program – it can be hard to find signatures that are significantly different from the useful COM add-in programs.

It is also hard for users to spot a malicious add-in in the system. Even if it is loaded at startup, it can hook application events unnoticed – no global macros appear, no VBA code can be found. To figure it out, add the COM add-in's command button, check the friendly name, and browse through the Registry to find the specific DLL. At this point investigations stop – even a well-educated user has no way of finding out what code a particular COM add-in contains.

Reverse-engineering a compiled ActiveX DLL is about as easy as reverse-engineering any other HLL program. You thought it was tough to extract virus code from an *Office 97 Word* document? Try it with an *Office 2000* COM add-in!

EXCHANGE

Hollow Vic-tory

Brien Barlev

Delegates who made the trip to Vancouver last autumn for VB'99 may remember the outspoken Brien Barlev (aka the Millennium Viking). Having trawled the 'Net continuously for seven weeks, over Christmas 1998 Brien finally had the following 'conversation' with someone who claimed to be the virus writer VicodinES at vic@codebreakers.org.

BB: Hello, hello, hello Doug Winterspoon [the author in the index and links HTML source code of Vic's Web site], alias ALT-F11… TWG [The Weird Genius, who had forewarned Vic about excessive NY Times publicity] is too right, the BSJs [Big Stupid Jerk, a name-calling payload element within Vic's Class viruses] are watching your moves. Remember what the BSJs did to Black Baron?

Perhaps Doug Winterspoon is a red herring. What do you care? Have another Vicodin! But remember, steroids can likewise stuff you around. PS: nuke away, it will only quicken your…

Vic: Hello? I know ALT-F11, yes, but this is not his email and I have never written a nuker. Is there something specific I can help you with? No need to write in odd codes. Just say what you want to say. Peace, VicodinES.

BB: To lesser (sic) your jail term, give yourself up. Your safest bet. Ask TWG what he reckons. Otherwise the Computer Fraud and Abuse Act 1986 will roast you worse than burnt Xmas turkey!!!

Vic: I write viruses, that is true – and I put them on a Web site – that is true – neither one is against the law. Code is not illegal.

BB: Virus code *per se* may not be illegal in the USA, but, like in the UK, distribution and inciting others is, as mentioned in your Theory of better…

Vic: Pure braggadocio – I was a fool who wanted attention and respect fast. That is, yes, a theory – you can take what you want from it but none of the content is true.

BB: And my gun, though smoking, is still full of sharp pointed bullets. Get the picture??? [Many of Vic's Web pages included photos of glamorous young women in aggressive poses bearing firearms – a warning to outsiders to stay clear of his Web site?]

Vic: No, not really. Does this mean you are going to shoot me?

BB: No, you have already shot yourself, poor chap! The sharp points are to prod you in a 'better' direction.

Vic: Sir, I still am having a hard time with your emails – TWG is what? I am guessing you are assuming I either know something or I'm someone else who would understand these codes.

BB: I am your senior, but there's no need to call me Sir! Who am I? Neither AV nor VX, but in the middle. A recent Greenpeace volunteer, I'm an IT dude of a bygone generation. Have been into IT security for many a moon. And who might you imagine helped to convict 8LGM?

Vic: If you helped convict someone then just say that. Who did you get convicted, what did they do? Why do you talk like this?

BB: 8LGM – 8 Legged Groove Machine. Don't you just love it!

Vic: If you want to say 'Vic, you stupid fuck, I hate you and your virus website', why won't you just say it? Because this is the second letter that I did not fully understand. Why so hostile? I don't ever remember attacking you, nor saying anything rude.

BB: All viruses are hostile. Do you not feel any shame or guilt for all the inconvenience to your helpless victims? I grant you that some AV vendors are sub-human in their behaviour.

Vic: I am in a strange place about this. I just wrote the code – and yes, they are viruses – but I did not put them on anyone's machine. So I did not intend to inconvenience anyone but the inconvenience happened anyway… so my feelings are very mixed. I do feel bad for those that panic, quite bad.

I'm gonna make a leap here. You got infected by one of my viruses – maybe someone at your work was playing on the Web and one of my _simple_ macro viruses got lose (sic)?

BB: No, on 6 November 1998, I intercepted and captured Class source code. Class.Sys told me heaps! I and others have been watching your moves ever since! Stacks of evidence for the FBI. You are well cooked! [Unbeknownst to me, ICSA had already established Doug Winterspoon as a David L Smith alias!]

Vic: This happens and now because of this (a simple, self-replicating macro) you are writing me odd letters (that could be construed as vailed (sic) threats)… I wonder, does ihug.co.nz know you are using them to write these odd emails?

BB: Do you have an out? Nope! There's that smoking gun again at your head. I will not pull the trigger but you are forced to by your own backfiring mischief. At least that way you have the choice of missing.

Vic: I write self-replicating code. In and of itself that is not evil.

BB: Not too evil perhaps, but a darn (sic) inconvenience to many, many victims worldwide. Your stated goal. In many people's books denial of service from system overload is akin to destructive payloads.

Vic: I fight with ALT-F11 about this all the time. Denial payloads are bad – it's cut and dry (sic) there.

BB: If you choose to ask, with my experience and wisdom I may be able to suggest your next moves for a softer landing. The choice is yours. Do not waste your obvious smartz. I am impressed!

Vic: This is getting even stranger. I mean no-one any harm. I have never ever written destructive code and I do not spread viruses. I write code and showcase it on a very public Web site. The AV companies have a copy the day a virus is posted.

Class was a virus I wrote eight months ago and now it's in the news. It was a new way to infect *Word 97* that had never been seen before. I posted it the day I wrote it – I even announced it in the newsgroups for all that wanted it. The anti-virus company *—* never updated their engine and therefore there was a big problem with Class.

It was in the papers because of that company and now I am getting more attention than I ever wanted because a large AV company did not write a good piece of software. Imagine the problem if someone had not called attention to this and a macro virus that destroyed data was written instead. You don't have to write me back as if this is an argument –yelling at me that 'this is wrong' and 'that is wrong'. You can still accomplish what you want by having a conversation.

BB: I will try to take account of your final comments about having a conversation.

Vic: Thanks :-) I just think that even if you hated me (not saying you do or not) that we can still 'chat' civilized.

The publicity machine for a large company is an amazing thing. Just look at all the attention now focused on 'Remote Explorer' and for what reason you have to want me in a world of trouble I do not know.

BB: Yes, marketing cowboys. But wait. Be patient. They *are* being found out. Time will surely come for some of them to pull their triggers!

Vic: He he, yea! Did you read the *—* write up on *—* and 'Remote Explorer' hype?

When I did the VDAT interview (in 1997) I was so new to the scene – like I said, I wanted to be a big VX man from day 1 (something I do not aspire to be any longer).

A popular target I have become.

BB: Yes, you have. You sounded thrilled a few days ago from all the attention and congratz from fellow VXers. So sorry, but you cannot have it both ways!

Vic: Well put yourself in my shoes – no matter what, being in the New York Times is exciting – sadly it was not something I could share with anyone in 'real' life so, yes, I was quite happy to get my congrats from the fellow VXers.

You offer that you would give me advice, I am curious (very curious to be honest), what would you suggest I do, and why? To get off topic (sic) for a second I am surprised that you used your real last name.

BB: Let's call it give and take. I'm a risk taker who carefully sized you up. You may still surprise me though, no doubt.

Vic: My explanation of what is out there for denial of service was in no way a masked threat. I do not do such things – nor would I ever. But you also cc'ed TWG and I cannot speak for him, having only talked with him a few times. He seemed like a nice young man but you never know. Oh also, (sorry for so many questions) – what is your motivation here? I can't seem to figure it out. Where does this motivation come from?

BB: A young talent like yours could be put to better use. If you need my further help and those of others, call back. Y2K is but a year away.

Vic: Well, I have to admit, for me, much has come of this. I have decided that writing viruses is too big a risk. You no longer seem hellbent on having me put away but the next person may not be so kind. As of 11am East European Time this email account will be closed and the sourceofkaos email will also be shut down soon. As for my Web site, I may just let it stay up for a week or so then kill it. Sometimes I can convince myself that I am doing a bit of service for the consumer.

I have truly enjoyed our exchanges – you gave me reason to think about lots of things and for that I thank you. Also, as I stated before, you seem much less driven to ruin my life for which I again say thanks.

I started in VX to 'infect the word' and ended up just enjoying the challenge and programming exercise of it all. Now I think this chapter in my life needs to be closed. Too much attention and unplanned events have created a big risk for me.

The publication of this exchange marks something of a departure for VB *but, on balance, we considered the content to be of sufficient topical interest for the magazine, if only to incite reactions from subscribers! Incidentally, extraneous editorial intervention was limited to the omission of specific AV companies. If you have any reactions which you would like shared concerning this article, its publication or its content, please feel free to email us at editorial@virusbtn.com. Ed.*

CASE STUDY

Boeing all the Way

Jeannette Jarvis
The Boeing Company, USA

The Boeing Company is the largest aerospace company in the world. We are a multi-national corporation with approximately 200,000 employees working worldwide. We are the world's largest manufacturer of commercial jetliners and military aircraft and America's largest *NASA* contractor. We have customers in 145 countries and operations in 27 US states. There are 335 satellites in orbit launched by *Boeing*. In any 24-hour period *Boeing* will sell more than $1 million in spare parts over the Internet. In any 24-hour period three million passengers will board 42,300 flights on *Boeing* jetliners in every country on earth. *Boeing* delivered approximately two airplanes a day in 1999; each plane contains over four million parts.

It is critical that we protect our infrastructure from any catastrophes. As virus writers get smarter it is essential that anti-virus vendors team with us in protecting our corporation. I found 1999 to be a challenging year on a number of fronts in supporting *The Boeing Company's* anti-virus needs. We had our hands full with Y2K preparation, anti-virus product software evaluations on several platforms and a couple of virus crises.

We have approximately 175,000 personal computers of all types. Employees use every operating system available (*Windows 95/98/2000*, *NT*, Mac, *Unix* etc), including legacy versions of these operating systems. I am sure that we still have *Windows for Workgroups* running somewhere as well. We are also running leading edge environments such as clustered terabyte file and database servers.

Needless to say, it is a formidable task keeping our computing environment virus-free. As we attempt to standardize it we hope to make this task easier. However, as we continue to purchase other companies (for example, the recent acquisition of *Hughes Space and Communications*), we will continue to fight an uphill battle.

Currently, two of us provide anti-virus product management support for *Boeing*: Sonja Floyd and myself. We have each taken primary responsibility for anti-virus support on different platforms and provide backup support for each other. We also rely on 'anti-virus' focal points in different geographical areas. I am grateful for every minute these focals work on anti-virus issues for us. We also work closely with all security organizations and email messaging organizations. Teaming between everybody is critical.

Each day we monitor the various Web sites to keep ourselves up to date on current issues. We can use this information to begin filtering, using a home-grown utility, or to initiate the steps necessary to keep employees updated. We also receive various virus alert reports from anti-virus vendors. Any 'heads up' information we can receive before users start calling us is very much appreciated.

Occasionally, we will get calls from an employee mentioning that they heard on *CNN*, or someone they know heard, about a new virus that formats your hard drive and asking if *Boeing* is protected from it? That is generally all the information we get, so we start tracking down what they possibly heard this time. Generally it is old information but we have been made aware of new viruses this way.

We use multiple anti-virus products and it certainly helps that we have multiple vendor contacts. The *Virus Bulletin* Web page and magazine have also proved to be useful tools. Several industry events have provided valuable networking. Keeping up to date with what is happening in other corporate environments helps. Open communication between our corporation and others made us aware of viruses such as Melissa even before we heard from our own anti-virus vendors.

We have an internal *Boeing* Web page that we like to use as a centralized point of contact for all viruses and hoaxes that we know about. This page reflects all known information about the virus and what the employee needs to do – update their definition files, clean the file, delete the message, or run for cover!

As much as we would like to say that this is the first location our employees use, we would be fooling ourselves. We know this by the amount of calls we still receive daily from employees referencing viruses that are written up on our Web site. We still see a number of hoaxes via concerned employees either trying to make sure all our employees receive their free *Honda* car or anxious about the current warning that 'It takes guts to say Jesus'.

If an employee receives suspicious email they are encouraged to send it to our computer security organization so the validity of the message contents can be substantiated. Our security organization has given us a 'heads up' on a number of viruses from the *CERT* (*Computer Emergency Response Team*) advisories they receive.

The Way Things Are

A number of issues can exist in our convoluted environment. Anti-virus vendors are not always willing or ready to support their legacy versions on our legacy operating systems. Due to our complex environment and upgrade processes, which can allow for quite some time to occur between operating system updates, we need support on older platforms. Multiple anti-virus vendor products are being used across our enterprise. As we have merged with

other aerospace companies we have had to work with multiple vendor licence agreements, in addition to employees' resistance to changing their current infrastructure to help migrate towards a standard enterprise product line.

Change is not always easy. Challenges have included getting employees – not just in-house, but also those who are travelling or telecommuting – to update their definition files on a regular basis. How can we make sure that they are keeping their computers up to date and how can we make it a non-intrusive update over phone lines in foreign countries? We have several software distribution tools to distribute updates to employees and we also provide availability to update files by utilizing centralized software distribution servers for pull-downs. It is imperative that we move to a homogenous environment that will allow streamlined processes.

Coping with Crises

Outbreaks of both the Melissa virus and the ExploreZip worm impacted *Boeing*. However, good things came out of both of these virus crises. This was reflected in the fact that they allowed us to start using an in-house, home-grown spam/virus filtering product that provides our first line of defence in protecting our company.

This utility, called *SpamJam*, is basically an email scanner. It was originally designed to block spam, but is also very effective at blocking viruses that replicate via email. This program allows us to begin filtering files or phrases as soon as we are made aware that vendors have identified a new virus. We can implement filtering before we receive and deploy the definition file updates from all our anti-virus vendors. We use this product to filter hoaxes, jokes, and spam. This product alone has made a tremendously positive impact on our protection in that it allows filtering to begin immediately. More detail about this product follows.

These particular virus crises also caused our upper management to become aware how the impact to our bottom line can be effected by a virus. Melissa and ExploreZip demonstrated, in dramatic fashion, the need to support our anti-virus security architecture.

Like most companies worldwide, we were not affected by any Y2K issues on the virus front. We monitored all anti-virus Web sites often throughout most of the night, implemented filtering when necessary and provided management with on-going 'heads ups' that all viruses being reported were low risk.

Boeing implemented a proactive exercise prior to the Y2K rollover, just in case an actual emergency occurred. We initiated filtering of several file extensions, we encouraged all employees to turn off their machines, and we forced the latest definition files through ESD (electronic software distribution) tools prior to the holiday. We also reminded employees again that they should not open attachments from unknown or untrusted sources. Had any emergency occurred the latest definition files would have been pushed to all employees through *NT* log in scripts. Although there was a risk that a virus could have been planted, we took the necessary steps to protect ourselves and were happy to see that all was calm. Now, if I can just have my holiday back!

Our Architecture

Email is not the only transport method viruses use to infiltrate our company, but it is certainly the most dominant. We introduced anti-virus products on several platforms in 1999 and feel that we are now, even more than before, addressing the virus situation immediately. Special thanks are due here to our upper management, who made the decision to be proactive in this venture.

Our latest design allows our mail hubs and anti-virus servers to behave like one machine. The mail hubs receive email from the Internet and from internal sites and they forward all their mail to the anti-virus servers. These scan it and then deliver it to the next hop, whether it is our outbound perimeter servers that send email to the Internet, or the *Exchange* bridgeheads that send email to the *Exchange* environment. We will eventually roll these two machines into one as we plan to phase out the mail hub product. There were only a few anti-virus products supporting this environment when we did our evaluation. We are encouraged that the vendor we selected is taking our input and improving their product. This product has been very effective in cleaning infections on this platform.

I would like to further describe our *SpamJam* utility at this point. The *SpamJam* Administrator compiles a list of phrases that are known to be contained in spam or email replicating viruses. Each phrase is assigned a numeric value, or weight, based on its likelihood to be in spam or virus mail and the unlikelihood that it will be in legitimate email. If a phrase is known to be contained in spam/virus email and is not likely to be contained in business-related email, a higher weight rating is given to that phrase. If a phrase is known to be contained in spam/virus email but is likely to be contained in business-related email, a lower weight rating is given. This method greatly reduces false positives too.

Each email starts with a value of zero. As the email is scanned, each phrase from the *SpamJam* database that is found in the email increases the email's value by the numeric value of the matching phrase. The *SpamJam* Administrator sets a numeric threshold. Any email whose value exceeds it will be rejected as spam or a virus. Acknowledgement is due to Dean Richardson from *Boeing* who wrote this program. It truly is our first line of defence.

Suggestions and Concerns

I consider it imperative that anti-virus software vendors test their products on the same hardware and software platforms that their customers are using. This proved to be a huge issue while I was testing products on our *NT* terabyte

clustering servers. I repeatedly ran up against some major problems in evaluating products, not the least of which was the fact that anti-virus products were not utilizing resources effectively. The anti-virus product only took advantage of two of the four processors running concurrently, even when it was the only task running.

I also ran into the problem where files being backed up via fibre channel were being scanned for viruses as the anti-virus product considered them to be inbound to the machine; this caused backups to take twice as long. Since current anti-virus products do not support fail-over in clustering environments, we had to provide some workarounds to ensure virus scanning would still occur whenever drive fail-over occurred.

I encourage anti-virus vendors to address centralized management issues. If System Administrators, and users, are not examining their virus log files daily (and I know for sure that the majority are not) then, realistically, how effective is the product? As Carey Nachenberg ably demonstrated with his Virus Simulation tool at the 1999 *Virus Bulletin* conference in Vancouver, it only takes one machine to start the crisis and depending on the type of virus, it can spread rather quickly. Centralized management of log files is necessary to verify that all virus issues are taken care of in a timely manner.

Establishing a solid team relationship between our vendors and customers will only assist in improving products. I urge vendors to share known product issues so that we can address them clearly with workarounds ahead of time when feasible. Nothing is more frustrating than finding a problem with a product and being told it was a known issue, but the vendors did not share it.

I also encourage developers to come on site as frequently as time allows; seeing our environment means that you are actually aware of our issues. I will continue to support implementing the best anti-virus solution at each platform and not supporting an anti-virus suite, after recent anti-virus product evaluations again reinforced my belief that no one vendor has a product that will best meet all our requirements on all platforms.

I take this opportunity to urge all in the anti-virus industry to assist the media in responsible reporting. Recently, a local Seattle television station reported a virus wiping out hard drives. So bad was this virus that it took down all their email servers. They posted an urgent alert to their Web page and I subsequently received email from a concerned employee. It was not long before folks were calling enquiring about this new virus.

After contacting the television station I was able to determine that we were already protected from that particular virus and it turned out that the television station itself was not up to date with their definition files. They eventually removed the notice from their Web site. Sadly, there are many other examples like this one.

I would like to encourage *Microsoft* to continue to address virus protection in their products as a high priority. As email is the number one transport method of viruses and we are seeing a proliferation of viruses taking advantage of distribution through distribution lists, I would like assurance that we can keep infections under control on this platform. The *Windows 2000* and *Office 2000* environments are making strides toward prevention. Not allowing DLLs to be replaced unless signed by *Microsoft* is a good thing. I am hoping that this does not impede the progress of system updates for non-*Microsoft* products.

It is possible that taking advantage of some of the group policies in *Windows 2000*, combined with *Office 2000* features, will assist in preventing the proliferation of macro viruses. In *Windows 2000* a user is not the administrator of their local machine by default; this will help prevent security exploits such as BackOrifice.

Where we will still need to be careful, however, is when the administrators are the technical support staff. Sometimes these folks are the ones who inadvertently spread viruses as they go from one troubled computer to another. Group Policy management in the *Windows 2000* environment needs to be thoroughly thought out prior to implementation.

Companies should address virus policy and policy enforcement. What are you going to do when someone brings down your email servers with the next ExploreZip worm? What if it was unintentional, as it usually is? What if they chose to disable the electronic software distribution tool that would have allowed them to be current? Do they receive a slap on the hand? Do they lose their jobs? Do you provide even more user awareness? With whom does this responsibility/liability lie?

Lastly, I would like to encourage anti-virus vendors to work towards a standard virus naming convention. We spend an inordinate amount of time tracking down viruses that each vendor has named uniquely. When I see a virus pass through the mail hub anti-virus servers with a name I do not recognize, after noting if it was cleaned or not, I must then confirm that we are protected from that strain on all our other platforms.

One example of this is W97M/ITSMURDER. This virus is named W97M/Marker.O by another vendor. It takes time to track this information down. I see this as a minor issue that all vendors could easily agree on, if for no other reason than to make your customer's job easier!

With the advent of VB Scripting viruses, HTML viruses, and who knows what coming down the line, we have a heady future ahead of us. The challenge of coming up with newer means of protecting ourselves from these threats is compelling at the least. It has already been demonstrated that without a successful security architecture our infrastructure can collapse. I look forward to continued assistance from the anti-virus vendors, *Microsoft* and platforms such as *Virus Bulletin* to assure us that will not happen.

FEATURE

The File Virus Swansong?

Peter Morley
NAI, UK

I've been getting blasé lately about OFFVs (old-fashioned file viruses). We swap virus collections every month with eight other anti-virus vendors, and although we still get quite a few OFFVs, they rarely give serious trouble. It currently takes about 10 minutes to process a file virus written in a high-level language, and less than 30 minutes to process a polymorphic virus we've never seen before. So it was quite a shock to get a file virus which cost me over half a day!

Introducing PEPE

The exercise started normally. I took an infected file, put it with several clean goat files in the same sub directory, and ran it. It was a well-behaved, willing infector. Without going resident, it directly infected every EXE file in its own directory. It did not rampage over the rest of the drive, and it did not drop funny files to send copies of itself to Nick FitzGerald. So far, so good.

Then came the first shock. I noticed that the length increase of the newly infected files was constant, but different from the original. It seemed sensible to repeat the exercise using one of the files I had just infected, so I did. The length of the second generation was different again!

All generations added a length of more than 12,800 bytes, which was prepended to the victim EXE file. The virus was encrypted. Operation seemed to be somewhat abnormal too. Whereas most viruses replicate and then run the original file, this one ran the original file, and then replicated.

The next stage – decrypting several different generations, and seeing what's really happening – is obligatory. It proved more difficult than usual. The entry point was well up the file, and I wanted to see the whole file. However, it worked eventually, and I was able to examine decrypted files. The name was 'Pascal Extra Polymorphics Engine', which is why I called it PEPE. It was 1.1, which raised the question of what happened to 1.0? In this situation, the only viable strategy is to wait and see if it ever turns up. If it does not, no action is required, or possible, and the earlier variant will never see the light of day.

So PEPE is an HLL (High Level Language) virus, polymorphic between generations, and written in Turbo Pascal, which I thought was almost dead. It takes all sorts… After decryption, the start of the file was constant, and it was easy to select a detection string which, while unique, was common to all generations, and which was most unlikely to give a false alarm.

However, it is totally unacceptable to decrypt every file which is scanned, so I needed to answer the question, 'Are you sure you cannot detect all generations, without decrypting first?' I worked at it, and the answer was 'Yes. Sorry, but I can't'.

The next step was to find a method of elimination, so that most uninfected files would never suffer the time penalty associated with decryption. So I looked for several short strings, always in the encrypted file, near to the entry point. It is easy to check for these, and abandon the file if they are not there. The problem here, when you have found them, is to do sufficient checking to be reasonably sure they *are* always there.

Failure to do this check means the danger of missing detection of infected files. This miserable process took a goodly portion of my half day.

Repair of variable length prepending viruses is simple in theory. You just put a pointer on the M (of the MZ) at the start of the original, uninfected EXE file, and execute a verb which removes 'all bytes which precede this one'.

I have just such a verb, so what is the problem? Well, how do I find the MZ, without the risk of putting the pointer in the wrong place? The answer has always been to get as near as possible to the right place before you start looking. Since I had the decrypted file, it was not too difficult, and repair was duly added.

Philosophy

Should I have processed this virus? The chance of it getting in the wild is fairly low. However, if it did get in the wild, and AV products failed to handle it, all hell would be let loose. I blanch at the thought of having to handle this one, from five customer-submitted samples, with the five customers enquiring about progress every two hours. It is much better to have our Tech Support in a position to say 'Yes. It is detected and repaired from Version xxx. Would you like an extra driver?'.

Payload

Since you ask, I do not think it has one. The author's few comments suggest he is delighted with his technological achievement, rather than vindictive to the world at large. However, 13,000 bytes of Pascal-generated code is a lot to flog through, so I ducked it. I'll unhappily do it when we get our first field sample, if we ever do.

PEPE 1.1 is one variant of one virus, so it will be counted as one. I'm not short of numbers, so this does not hurt. However, if anyone tries to tell you it is about 50 different variants, I suggest you raise a puzzled eyebrow.

TUTORIAL

What DDoS it all Mean?

Nick FitzGerald
Computer Virus Consulting, New Zealand

Unless you were well out of touch in early February this year, you must have heard about the day the Internet died. 'Cyber-attacks batter Web heavyweights' read one headline and the story ran endlessly in on-line, print and broadcast media for more than a week. Odd that the NASDAQ reacted by strengthening…

Distributed denial of service, or DDoS, attacks disrupted some of the largest Web sites – *CNN*, *MSN*, *Yahoo* and others – sites designed to serve millions of pages per day. So what are DDoS attacks? How might they affect you and what should you do to avoid them?

History Repeating

Network denial of service (DoS) attacks are easy to understand. A malicious user attempts to exhaust some limited resource – usually network bandwidth – to deny others access to a network-based service. Apart from bandwidth consumption, other forms of DoS attack are possible. Specific versions of some network software are known to have bugs that render them unstable when 'odd' packets, or packet sequences, are received. An attacker could utilize such a weakness to DoS a site known to run an affected version of the vulnerable software.

Historically, someone planning a DoS attack would obtain code to implement an attack the intended victim would be vulnerable to, or write an implementation of the chosen vulnerability from a description of it. One of the risks of discovery would be that the attacker could lose their account on the machine launching the attack (if, in fact, the attack was ever traced). Amelioration of that risk was often accomplished by the attacker cracking some other host first, then launching the DoS attack from there.

An easily compromised system, giving the attacker root or administrative privileges, has two advantages. First, it moves the attacker one step further from possible banishment since it is not the attacker's own system. Second, and more importantly, the attacker further reduces the chance of being discovered because if the site was easily compromised (say, with an old exploit), by definition it is a poorly administered site. Also, with root access, the attacker could alter system logs and the like, further obfuscating the real source of the attack, or at least the person responsible for it.

As widespread DoS'ing of sites became something of a sport among elements in the hacking underground, a new challenge arose. With the attacks becoming more common, some potential targets were increasingly armoured against one or more of the well-known attacks, through improved firewall and router configurations and use of network intrusion detection systems (NIDS). Further, the very large (and, therefore, most brag- and news-worthy) sites were daunting targets because of the sheer bandwidth a successful attack would have to use up.

Distributed DoS attacks were the obvious next step, solving both problems by implementing several attacks in one tool and providing a means to coordinate and synchronize attacks from very large numbers of machines. Given the alternative for an attacker having to maintain a motley crew of tools, and possibly accomplices to help launch attacks from a handful of compromised sites, the advantages of DDoS tools should be clear.

Are DDoS Tools New?

From the media coverage, you would probably assume the answer to this question is 'Yes', but they are not that new. The concept has been around for some time, but although there have been examples of DDoS and other distributed hacking tools, they certainly have not been common.

In September 1999's Editorial I mentioned a Trojan that had become widely distributed by mass-emailing. When the attached program was run, rather than installing the latest security patches to *Internet Explorer*, it installed a program to monitor whether an active Internet connection existed, and if so, sent a large amount of abusive email to the *Bulgarian National Telecommunications Company* and ISP.

Over the following few months, variants with different network-based, resource-wasting attacks were also seen. These reportedly caused a great deal of inconvenience to the real target – the Bulgarian ISP – but typically were of nuisance value only to those tricked into running them guilelessly. These Trojans may have implemented the first widespread, programmatic DDoS attacks.

Released shortly after Melissa, X97M/Papa contained not only a mass-email distribution mechanism, but a distributed 'ping' DoS attack directed at two machines of a well-known network security researcher. Perhaps fortunately for the target of Papa's ping flood, Papa did not become anywhere near as widespread as Melissa.

Between the appearance of these two early, simple, PC-based DDoS agents, W97M/ColdApe was released. As the target of that virus' email payload, it was very ineffective if it was intended as an email DoS attack against me or the magazine. So ineffective, in fact, I would not have considered this a possible motive for that part of its payload. However, several newsgroup posts by a virus writer affiliated with one of ColdApe's writers suggests that the pro-virus/VX underground saw it as such.

Outside the world of personal computers, DDoS tools started to appear in the wild in early to mid-1999. The best known are Trinoo (or Trin00), Tribe Flood Network (TFN), Stacheldraht and a recent update to TFN known as TFN2K. These tools have gained quite some media coverage, probably because they have been closely analysed by security experts and source code for them is readily available. However, in a recent article, the hacker known as Mixter (author of TFN and TFN2K) claimed to know of four other DDoS tools, that he named. They have not been publicized, but may be in use, and how many other DDoS tools are in use that Mixter does not know of?

The Shields are Down Cap'n…

So how do these recent network DDoS tools work? Perhaps the most important thing to realize about them, which the mainstream media has mainly overlooked, is that there are really two separate targets in these attacks. Obviously the big-name Web sites in the early-February headlines were targets, but they could not have been targeted (as successfully) without the first set of targets – a large number of poorly secured and under-administered Internet servers.

Trinoo, TFN, TFN2K and Stacheldraht have similar general architectures, varying in implementation details. All four have two software components installed on compromised machines. Let us refer to these two components as 'master' and 'slave'. An attack with any of these begins with the attacker locating and compromising many suitable machines, on which the slave is installed. A few machines are also compromised and the master software is installed.

Together, these machines constitute an attack network. Launching an attack is simply a matter of contacting the master(s) and providing them with the address(es) to attack and the type of attack to use. Trinoo is the simplest of these well-known DDoS kits and it only implements one network DoS attack – a UDP flood. The others add ICMP and SYN floods, and the Smurf attack. Most of these attacks either depend on IP spoofing (sending packets with forged source addresses) or use spoofing to confuse and slow diagnosis and resolution attempts by the target further.

Captured and/or released source code for these kits shows various 'fingerprints' the tools leave in a system or on a network. Later tools, especially TFN2K, are more sophisticated in this regard, making several attempts to disguise their presence further. Some of these obfuscations include: the encryption of all control messages between masters and slaves with compile-time keys; depending on probabilistic delivery of control messages, so the slaves never respond to masters, and; use of ICMP packets which extant network tools have unsophisticated handling of and that generally are allowed through firewalls.

As this article was being completed, reports arrived of a US university discovering a Win32 port of the Trinoo slave installed and active on PCs in its student residence network. All the affected PCs had also been compromised with BackOrifice, suggesting that either BO has been 'bundled' with this Trinoo executable or Trinoo was installed once the PCs were accessible via the BO client.

Protecting Yourself

The bittersweet irony of these DDoS tools is that you cannot protect yourself. The best an individual site or firm can do is ensure its machines are as secure as they can be. After that, you can only hope the 'white hats' find the easy exploits in a timely fashion relative to the 'black hats', then install any security patches your vendor produces.

Having done all that, you are protected as best you can be against becoming a DDoS slave, but you can do little about attacks that may be launched against you with these tools. Depending on various technicalities, there are some newer router and firewall options that can reduce the impact of some of the DoS attacks the slaves launch without rendering your network unusable for its intended purposes.

NIDS have been updated to detect traces of Trinoo, TFN, TFN2K and Stacheldraht in the network. If you have a NIDS and have updated its profiles, do not be complacent that this is sufficient to detect these tools. They are available in source form and tend to be in the hands of more sophisticated users than the script kiddies. The source recommends users alter many of the defined constants precisely to avoid such 'signature' scanning methods. Evidence that attackers are heeding this advice is available in the Win32 port of Trinoo mentioned above. It does not use the 'default' ports described in the first detailed analysis of Trinoo, although from a rudimentary first look at the program it appears that the rest of the Trinoo protocol is fairly standard in this case.

Not to be left out, several anti-virus developers have added detection of the 'big four' DDoS tools to their products. This, of course, raises even more problems than the NIDS face. A good NIDS may be able to detect some tell-tale changes in traffic flow shapes, 'odd' packet types and the like, raising an alert for the network manager to apply some human intelligence to a trace. However, with the tools distributed as source, and intended for building on many systems, imagine the number of combinations of compilers, linkers and strippers. Cross that with the number of standard libraries, allow that two (or more) different sets of development tools are available for most of the likely target systems and factor in how many versions of these tools? We are talking a staggering number of potential binary variants, and that is before allowing that attackers may alter the known code, which they *are* doing.

The machines that most need detection added are the ones that responsible, concerned admins cannot affect. Your best defence is to secure your own sites and harden your network boundaries against the known attacks. Finally – and marketing departments will not like this – you had best hope that your Web site or company is not interesting or newsworthy enough to be targeted!

OVERVIEW

Testing Exchange

Fraser Howard

This article is intended to serve as a guide to the procedures used and considerations given during testing in the *Exchange* product review featured on p.20 of this issue.

Exchange AV – a 'Job Desciption'

There are a number of requirements that an *Exchange* anti-virus product is expected to fulfil. Aside from the rather obvious one – that of providing on-access scanning of both incoming and outgoing emails (local or Internet) – other features are also of interest. These include:

Administration and configuration issues. There is a genuine need for central administration from the server. In a large organization this can mean administration throughout the company (possibly containing hundreds of *Exchange* servers) from a single seat. Thus, the integration between the product and the *Exchange* hierarchical object tree is important.

On-demand scanning. The ability to perform on-demand and scheduled scanning of user mailboxes and public folders is essential.

Installation and updating. Again, it is desirable to roll out updates and installation templates to multiple servers from a single seat.

Central quarantine. Within a large organization it can be desirable to have a single, central quarantine in which to contain suspect samples.

Content filtering. Some form of content filtering is a valuable supplement to scanning mail for known viruses. Such solutions can prove useful during outbreaks of mass-mailing viruses/worms.

Alerts and notifications. Exactly what is needed is very much dependent upon the size of the organization. Thus, flexibility is the key – the ability to configure alerts and statistics exactly as needed.

The brief feature last month (see *VB*, February 2000, p.22) outlined the basic princples behind message flow and storage within *Exchange*. Also mentioned was the choice of product architecture, specifically relating to the on-access scanning of messages. All currently available *Exchange* anti-virus products, except for *Antigen* by *Sybari Software Inc*, use the MAPI interface to the *Exchange* Information Store (IS) is used. In this scenario, on-access scanning of both inbound and outbound messages is reliant upon notification events as the messages are routed through the server. Thus, under heavy inbound mail load there is a race condition between messages being scanned, and the messages being written to the IS.

The biggest consequence of this, as far as product testing goes, is that a simple log of writes to the *Exchange* IS versus time cannot be used to determine on-access scanning overheads. Only if the messages are queued for virus scanning *prior* to being written to the IS can the above approach be used. Thus, in the following *Exchange* product review (p.20), the scanning overhead has been gauged by monitoring the '% Processor Time' with the *NT* performance monitor on the *Exchange* server.

Personal Folders

The concept of personal folders within *Exchange* brings mixed reactions from various sytems Administrators. The bone of contention lies in the fact that the PST files that comprise the folders are typically stored locally on the client workstation within the users' profile. Thus, the convenience of central backups of all data at the *Exchange* server is removed (although the PST files could be stored on a central server).

PST files are essentially smaller versions of the *Exchange* IS – they are capable of storing the same wide array of messages. This results in another area in which virus scanning must be performed. Assuming that the policy within an organization permits users to create and use personal folders therefore, the ability to scan the associated PST files, be they workstation or client-based, is essential.

Bulk Email Automation

Having identified product features central to the choice of a suitable *Exchange* anti-virus product, the next step is to consider product performance.

Detection rates. A fundamental consideration – this is, after all what the product is there to do.

Overheads. Though perhaps less important than on the desktop products, the overhead of the on-access mail scanner is still a worthy concern.

Scanning Speed. Since scheduled scans are a much used feature of any groupware product, the scanning speed is also a genuine consideration in choosing a suitable product.

In order to test the detection rates and scanner overheads of products, it was necessary to automate the dispatching of numerous emails, each bearing single file attachments (either viral or clean). For internal mail (between users within the same *Exchange* site) a simple VBA procedure was written which created and dispatched one email for each file within a specified path, attaching the file to each email in turn. A small executable running on a *Linux* machine was used to automate incoming Internet (SMTP) mail – again, one file attachment per email.

PRODUCT REVIEW 1

February 2000 Comparative Review Addendum

VB offers its apologies to the Icelandic anti-virus company *FRISK Software* for omitting their results from the DOS Comparative last month. The full set of *F-PROT's* results are set out below and set against the rest of the pack.

The detection tests were performed using a test-set of the usual *VB* Polymorphic, Standard, Macro and In-The-Wild sets. Importantly, the ItW set was aligned to the October 1999 WildList, which was announced two weeks prior to the product submission deadline (01/11/99).

FRISK Software F-PROT 3.06a (31/10/99)

ItW Boot	100.0%	Macro	99.8%
ItW File	100.0%	Standard	100.0%
ItW Overall	100.0%	Polymorphic	97.1%

A quick glance at the results below is sufficient to satisfy expectations of the *F-PROT* engine by *FRISK Software International*. Skipping through the ItW file and boot sets, detecting all the samples along the way, earns the Icelandic product its second VB 100% award.

Results across the board parallel those observed for *Command AntiVirus* – unsurprising since the *Command* product uses the *F-PROT* engine. Detection in the Standard set (to which a variety of the recent *Windows* file infectors had been added) was faultless and only three samples (those infected with W97M/Astia.Y) were missed from the Macro set. The weakest area was detection in the Polymorphic set, in which samples infected with ACG.A and Win95/SK.844 were missed.

In terms of scanning speed, *F-PROT* excels (pardon the pun) at scanning OLE2 files, returning a throughput of approximately 3750 KB/s. Executable scanning was less impressive, but happily, no false positives were registered against either test-set.

On-demand tests	ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Missed	%	%	Missed	%	Missed	%	Missed	%
Alwil LGuard	3	99.8%	99.8%	123	96.3%	91	91.6%	11	98.9%
CA Vet Anti-Virus	0	100.0%	100.0%	60	98.4%	264	94.4%	1	99.9%
Command AntiVirus	0	100.0%	100.0%	3	99.8%	62	97.1%	0	100.0%
Data Fellows FSAV	3	99.8%	99.8%	30	99.1%	0	100.0%	2	99.9%
DialogueScience DrWeb	0	100.0%	100.0%	11	99.6%	0	100.0%	6	99.7%
Eset NOD32	0	100.0%	100.0%	60	98.3%	21	97.2%	8	99.7%
FRISK Software F-PROT	0	100.0%	100.0%	3	99.8%	62	97.1%	0	100.0%
GeCAD RAV	23	96.3%	97.0%	92	97.2%	8792	43.3%	236	85.0%
Grisoft AVG	0	100.0%	100.0%	52	98.4%	355	86.1%	90	96.4%
Kaspersky Lab AVP	0	100.0%	100.0%	19	99.3%	0	100.0%	0	100.0%
NAI VirusScan	0	100.0%	100.0%	12	99.6%	17	97.7%	0	100.0%
Norman Virus Control	0	100.0%	100.0%	11	99.7%	195	94.4%	6	99.7%
Sophos Anti-Virus	0	100.0%	100.0%	73	97.7%	191	94.9%	18	99.3%
Symantec Norton AntiVirus	0	100.0%	100.0%	34	98.9%	305	88.8%	1	99.7%

PRODUCT REVIEW 2

NAI GroupShield v4.0.4 for Exchange

We kick off groupware anti-virus product testing with the *Network Associates* product for the *Microsoft Exchange* platform. According to the various forms of product advertising and press releases, two particular features of *GroupShield for Exchange* (*GSE* hereafter) distingiush it from competitor products – 'locking' messages at the inbox, and a client-side scanner.

These two features and the performance of *GSE* as a whole were investigated in this, the first of *Virus Bulletin*'s *Exchange* product reviews.

The Package

A CD containing the entire *Network Associates Total Virus Defense Suite* product range, and another containing product updates, were submitted to *VB* for testing. A single folder on the latter contained the files relevant to the *GSE* v4.0.4 product – installation files and a 184-page manual in PDF format.

Installation

Prior to installation, a suitable service account for the *GSE* services to be used subsequently needs to be available. Additionally, the installation process itself must be performed from an *NT* account with specific privileges (Service Account Admin rights to the *Exchange Directory* and Domain Admin rights to the *Exchange* server).

The friendly InstallShield installation routine presents the options that you would expect from such a product:

- select the quarantine folder.
- set access permissions to the quarantine (*GSE* supports either one central quarantine folder for an entire organisation, or distinct folders for each site or server within the organisation).
- select mailbox(es) for notifications.
- set a single scheduled on-demand scan of the *Exchange* message store.

For Administrators overseeing larger sites, there is an option to copy the installation to any other *Exchange* servers within the site.

Unfortunately, even with the necessary service and installation accounts, problems were still encountered when installing *GSE*. Briefly, the problems centred around an inability to alter the permissions of objects in the *Exchange Directory*. These problems exist thanks to some of the changes introduced in Service Pack 3 for *Exchange* (which was released after *GSE v4.0.4* shipped).

Aside from installing *GSE* on the *Exchange* server itself, two options are also presented for installing the *GSE Client Extension* (discussed below) to workstations – either using a disk set, or emailing the extension to all mailboxes on the server. Though more convenient, the latter option can place a rather large load upon the server (since the installation message tips the scales at a little over 12 MB).

Upon selecting either of these options, an additional screen appears during installation, enabling the default client extension configurations to be set. Selecting either of these options unfortunately led to further SP3-related errors. To solve the problem, the user administration extensions (necessary for modifying the client extension settings) were added to the 'Recipients' container manually by using *Exchange Administrator* in 'Raw Mode'.

Configuration of GSE

During installation, a 'Network Associates GroupShield' object is added to the *Exchange* object hierarchy, within the server container. Double-clicking this object opens up the *GSE* property pages that are used for configuring the product from within *Exchange Administrator*.

The property pages are split logically amongst the various features of *GSE*, and all are presented in a straightforward and neat manner.

On-Access Protection

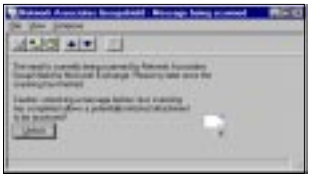
Configuration of on-access scanning is split between two pages – scanning of either mailboxes or public folders. The options presented on both of the pages will be familiar to users of conventional workstation anti-virus products. For example, these include the ability to enable file and/or macro heuristics, scan inside archives, scan inside compressed files and to specify the file types to be scanned.

By default, all mailboxes and public folders are monitored by the on-access scanner, although if desired, specific mailboxes or folders may be selected and this custom list included or excluded. Any changes made to the exclusion

list necessitate the stopping and restarting of the *GSE* Services. A prompt informs the Administrator of this, and if desired, the process is performed automatically.

GSE utilises the MAPI interface to the *Exchange* Information Store (IS). In order to prevent access to unscanned messages, *GSE* employs a custom message property for flagging scanned and unscanned messages. Access to unscanned messages is prevented through the use of the 'Lockout' form (IOFORM.EXE). If so desired, the message locking feature can be disabled, but, as you would expect, it is enabled by default.

Under the heavy inbound mail flow employed during the detection tests, the message locking can be viewed in real-time by simply observing the contents of the recipient's inbox. As new messages appear in the inbox, the 'unread message' icon rapidly (within one second or so during testing) changes to a custom *GSE* 'locked' icon, indicating that the message has become locked.



Attempting to view a locked message by double-clicking it results in the lockout form being displayed. If the message has been locked for more than 10 minutes, this form provides a button for the user to unlock the message manually. If this should fail, then there is an additional utility available for Administrators.

A weakness in the message locking feature was uncovered during testing. Right-clicking a locked message in the *Outlook* inbox, and selecting 'View Attachment' from the resulting context menu enables access to the possibly infected attachment.

There are four options to choose from for dealing with infected messages. The action can be set to disinfection (quarantine if unsuccessful) or immediate quarantine, of either the entire message or just the infected attachment.

On-Demand and Scheduled Scanning

Once again, regular users of AV software will feel at home with the configuration and use of the *GSE* on-demand scanner. Whilst a scan is in progress, the main window displays the scan summary, including names of the scanned mailboxes and/or folders, the number of viruses found in each and the number of files cleaned or quarantined in each.

Configuration of the on-demand scanner (options to use heuristics, expand file archives, expand compressed files, specify file types to scan) is possible by accessing the settings page. Once again, a custom list of mailboxes and folders can be created, and subsequently included in or excluded from the scan.

Perhaps more important than performing immediate on-demand scans is the facility to schedule scans. A plethora of the usual schedule options are available, enabling one-off and repeating (hourly, daily, weekly, monthly) scans to be scheduled. The configuration of each of the scheduled scans can be altered as for the on-demand scans described above.

A summary report is compiled from the results of all on-demand and scheduled scans, and sent to the Administrator using the *GSE* form SUMFORM.EXE.

Updating GSE

Product updates are performed from the 'Updating' page. Three options are provided for obtaining updates – from the Internet via a proxy server, from a remote server path or from the local machine. Multiple *Exchange* servers running *GSE* can be updated at once, and there is also a checkbox to select whether or not client updates should be issued at the same time.

Users accustomed to the *Network Associates* product range will be familiar with the SuperDAT files that are used to provide both signature and driver updates. In order to update by running an SDAT file the Administrator must be seated at the server. Remote updating is possible from within *Exchange Administrator* by pointing *GSE* to a local drive containing the necessary DAT files.

During testing, therefore, *GSE* was updated by pointing it to a directory containing the manually extracted DAT files from the downloaded ZIP (DAT4065.ZIP).

The GSE Client Extension

As mentioned above, *GSE* gives Administrators the option of installing client-side anti-virus protection. If not distributed during the main *GSE* installation process, the client extensions can be distributed at a later stage from within the *GSE* configuration property pages accessed from *Exchange Administrator*.

To aid the installation and upgrading of the client extensions, one of the organization forms installed with the product is used (INSTFORM.EXE). Subsequently, the chosen clients receive an installation message which, upon downloading, is used by the email client (*Outlook* or *Exchange*) to install the *GSE* client extension. The same form is used to aid the distribution and installation of client extension updates from the *Exchange* server.

Once the client extension is installed, an extra drop-down menu ('GroupShield') is enabled, which provides options to view the configuration (and adjust if permitted), and scan the mailbox or selected items. Shortcut buttons are also added to the *Outlook* toolbar providing a shortcut to these facilities. Furthermore, an addition to the 'Help' drop-down menu within *Outlook* provides access to the *GSE* client extension help pages.

The client-side on-access scanning options enable message scanning when reading (resulting in a short delay before the message is displayed) or writing (short delay before message is routed to the *Exchange* server) messages, and when new messages are delivered. On-demand options include the ability to scan the contents of mailboxes (those accessible to the user), public folders and personal folders (PST files) – peronal folders are commonly stored on the local workstation.

Upon attempting to view, send or post a message with an attachment which the *GSE* client extension determines to be infected, the action is blocked and an alert box displayed. The specific action taken by the *GSE* client is determined by the configuration of the client extension, which in turn is controlled by the Administrator – by default users have no control over the settings. The degree to which users may have control can be set either during installation of *GSE* (assuming the client extension is installed at that stage), or from the property pages of the particular mailbox within *Exchange Administrator*.

Alerts & Notifications

Upon detecting a virus during either on-access or on-demand scanning, *GSE* can issue a notification message to the Administrator, the intended recipient, the message author or any combination of these parties. The subject line of the notification message can be customized using a standard series of tags (virus name, recipient, quarantine etc), as can the text that will be written in place of any stripped attachments.

As well as virus incident notifications, *GSE* provides the administrator with a facility to customize the alerts that are written to the *NT* event log. Furthermore, it is possible to link the alerts to the familiar *NAI Alert Manager*, which enables nine possible alerting methods (including SMTP, network message, pager, printer, SNMP, program execution and audible alert). Any combination of these methods may be used, ensuring no respite for Administrators, be they in the office, at home or on the road!

Virus Detection

The detection rates for on-access scanning of both Internet and internal email (internal is used here to represent email sent between users within the same *Exchange* organization) were investigated in this review. For this, the two automated routines mentioned on p.18 were used. A complete listing of the test-sets used for testing can be found at the URL listed at the end of this review. The ItW set was aligned to the January 2000 WildList, and thus would be expected to be well within the detection capabilities of *GSE*, which was updated according to signature files dated 16/02/2000. The action upon detecting a virus was set to quarantine the entire message, and so detection rates could be determined quite simply from what 'got through' (on-access scanning) or what 'remained' (on-demand scanning). To eliminate any peculiarities, the log files were also used to double-check the detection rates.

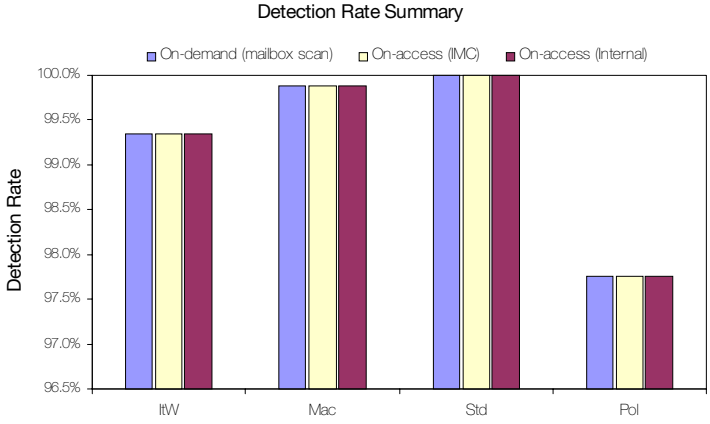
A quick glance at the detection rates confirms that for the most part, *GSE* has performed as you would have expected it to, given the generous 'timing' of the test-set and the product updates. Unfortunately for *GSE*, failing to detect samples of W97M/Hubad.A led to incomplete ItW detection. This is somewhat surprising since the necessary signatures to detect this virus are included in the DAT files used for testing (4065), as verified with brief tests of the *GSE's* desktop brethren – *VirusScan*.

Elsewhere, the majority of the samples of the complex polymorphic Win95/SK.844 accounted for the misses in the Polymorphic set.

On-access Scanning Overhead

The overhead of any conventional desktop on-access scanner is an important factor to the success of the product. Significant imposition upon the user will simply result in the scanner not being used, and protection being lost. The overhead of on-the-fly email scanning is not so critical to product success however, for the simple reason that any slight delay is not so directly 'visible' to users. Provided that the overhead is not absurdly large, a slight delay in the receipt of emails will not be noticed.

The architecture of *GSE* raises two important issues. The first has been discussed above, and is concerned with the need for message locking such that under heavy server load, users cannot access unscanned mail. The second refers to scanner overhead, and arises as a result of the message locking functionality. A user logged on to his or her mailbox will see new messages arrive, become 'locked', and then subsequently become 'unlocked' once scanned and verified clean. The user is thus presented with a form of direct 'contact' with the scanner, which has the consequence of raising the importance of scanning overhead.



As mentioned in the article on p.18, measurement of the on-access scanning overhead is not a straightforward process because messages are delivered to the destination mailbox prior to being scanned. Instead of using a log of message writes to the *Exchange* Private IS therefore, a log of Processor activity, specifically '% Processor Time' has been used. This provides a direct measure of the fraction of time that the *Exchange* server processor has spent doing useful work.

To measure the on-access scanning overhead for scanning internal and Internet mail, 1,000 emails (each bearing a single file attachment from a set of 1,000 executables) were despatched with the *NT* performance monitor running on the *Exchange* server. The process was repeated at least five times for each of the following configurations:

- *GSE* not installed on the server
- *GSE* installed, but on-access scanning disabled
- *GSE* installed, with on-access scanning enabled

Within each configuration, the observed percentage processor times were consistent between successive mail runs, and the values plotted in the graph here represent the simple average. As can be seen, on-access scanning of both internal and Internet mail increases the workload of the *Exchange* server processor by a factor of approximately 1.8 and 1.4 respectively.

On-demand Scan Rates

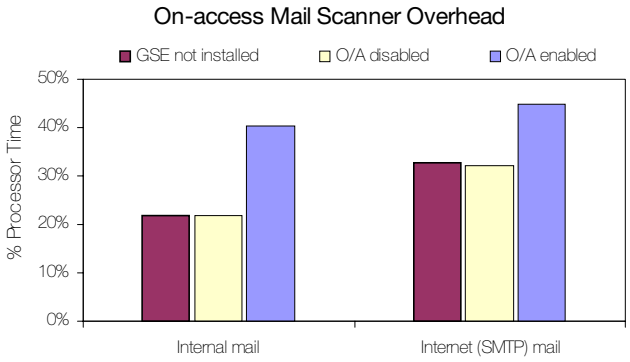
Next, the throughput of the on-demand scanner was assessed. For this, the contents of the customary *VB* Clean set (consisting of executables and OLE2 files) were mailed (one file per email) to a specific mailbox, and then an on-demand scan of that mailbox performed. The throughput of the client-side scanner was assessed in a similar manner.

On-demand scanning rates	Executables (KB/sec)	OLE2 files (KB/sec)
Server-side scanner	1098.3	1101.9
Client-side scanner	488.3	349.5

Further Product Developments

Finally, a sneak preview of *GSE v4.5β* – which includes support for the new virus scanning API from *Microsoft*, incorporated in *Exchange 5.5* SP3 – was undertaken. The installation routine was a simpler affair (the SP3-related problems corrected), and the *GSE* property pages had been tidied. One welcome addition is a mild form of content filtering – the ability to block all or specific file attachments, either by filename or file extension.

Another new feature incorporated in *GSE v4.5* is the *Outbreak Manager* (*OM*) – quote, a 'virus outbreak monitor'. The monitor operates using series of rules that can be added and configured by the Administrator. Within the rules there are four triggers upon which *OM* may act:



- no. of viruses in a time period
- no. of identical viruses in a time period
- no. of identical attachments in a time period
- no. of identical attachment types in a time period

The actions that may be taken upon a trigger condition being satisfied vary from deleting attachments or performing a DAT update to shutting down the *Exchange* server.

Summary

Reducing a product review to a simple score out of ten is always a difficult and, in many ways, worthless exercise. This is even more the case for a groupware product review where so many factors determining a good or bad product come into play.

What can be said without hesitation about *GSE* is that once the minor problems encountered during installation (thanks to the *Exchange* SP3 conflict) were solved, it was an enjoyable product to test. The prime area of concern in this review lies with *GSE's* failure to detect the samples of W97M/Hubad.A – a virus that has been on the WildList since December 1999. Exactly why this was missed is a mystery, since its detection is catered for in the DAT files used during testing.

Sitting with all but one of the other *Exchange* anti-virus products on the MAPI side of the fence, the importance of preventing access to messages prior to them being scanned is obvious. Thus, the message locking functionality which *GSE* boasts is without doubt an attractive feature, and worryingly, a feature that is currently not present on all other commercially available products.

Technical Details

Product: *Network Associates GroupShield for Exchange v4.0.4.4065* (16/02/2000).

Test Environment: *Exchange Server*: 450 MHz AMD *K6* with 128 MB of RAM, 8 GB hard disk, running *Windows NT 4.0 (SP5), and Exchange Server 5.5 (SP3)*. *Workstations*: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, running *Windows NT* or *Windows 98* with *Microsoft Outlook 98 v8.5.5603* (security patch applied).

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Ex/200003/test_sets.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, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

VB 2000, Virus Bulletin's 10th annual international conference, takes place on Thursday 28 and Friday 29 September 2000 at the Hyatt Regency Grand Cypress Hotel in Orlando, Florida. The inaugural welcome drinks reception will be held on the evening of Wednesday 27 September. There are currently exciting opportunities for both event sponsorship and the conference exhibition. For details on all aspects of the conference email VB2000@virusbtn.com or visit http//www.virusbtn.com/.

The fifth Ibero-American seminar on IT security and computer virus protection will take place from 22–27 May 2000 at the Informatica 2000 International Convention and Fair in Havana, Cuba. The principal topics include anti-virus software, Internet security, e-commerce security and systems audits. For further details contact José Bidot, the Director of *UNESCO's* Latin American Laboratory; Tel/Fax +53 7335965 or email jbidot@seg.inf.cu.

***Network Associates Inc* announces the immediate availability of the *WebShield 300 E-ppliance*.** The company claims that this is the first security device to combine anti-virus, firewall and VPN software in an easy-to-use e-business appliance. For details contact *NAI* in the UK; Tel +44 1753 827500 or email Caroline_Kuipers@nai.com.

Symantec has brokered a deal with *L-3 Communications Corp* to acquire its subsidiary *L-3 Network Security's* vulnerability management solutions, consulting business and employees for a one-time US$20 million cash payout. By doing so, ***Symantec* take control of the *Retriever* and *Expert* security assessment and management product lines.** For details email Lucy.Bunker@symanetc.com or visit http://www.symantec.com/.

***Kaspersky Lab* announces the launch of its brand new Web site – http://www.avp2000.com.** The site is dedicated exclusively to its new product *AVP for MS Office 2000*, powered by macro virus management technology *AVP Office Guard*. Aside from offering a '100% guarantee against macro viruses', *AVP for MS Office 2000* also includes a 'watchdog' virus interceptor for all the main *Office* applications (*Word*, *Excel*, *Access*, *PowerPoint*), a plug-in filtration utility for email programs and the flexible anti-virus protection administration utility *AVP Control Centre*. For further details contact Denis Zenkin; tel +7 095 9485650 or email denis@avp.ru.

A two-day course entitled Investigating Computer Crime and Misuse will be run by *Sophos* on 5 and 6 April 2000 at the organization's training suite in Abingdon, Oxfordshire, UK. For further information, or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, email courses@sophos.com or visit the company's Web site http://www.sophos.com.

InfoSec 2000 will take place at the National Hall, Olympia, London from 11–13 April 2000. The show includes exhibitions and talks on various subjects including virus protection, firewalls, network security, e-commerce and Web security. There will also be a series of 46 free, on-floor seminars on topics such as *Windows 2000* and *Linux*. For more details or to make a booking contact Yvonne Eskenzi; Tel +44 2084 498292 or email yvonne@eskenzi.demon.co.uk.

The *Computer Security Institute* (*CSI*) has released details about its 10th annual Network Security conference and exhibition this year. **NetSec 2000 will be held at the Hyatt Regency Embarcadero in San Francisco from 12–14 June.** For more details contact *CSI*; Tel +1 415 9052626 or visit http://www.gocsi.com/.

F-Secure Corporation has recently released virus protection software for the Wireless Application Protocol. ***F-Secure Anti-Virus for WAP Gateways*** is, the company claims, the first product to protect wireless communications, transactions and e-commerce from new and emerging vulnerabilities and exploits. For more information email Pirrka.Palomaki@F-Secure.com or visit http//www.F-Secure.com/.

The fourteenth annual Vanguard Enterprise Security Expo 2000 will be held at the Atlanta Hilton and Towers, Atlanta, Georgia, on 15 and 16 May 2000. For further information contact *Vanguard*; Tel +1 714 9 390377, or see http://www.vipexpo.com/.

***Content Technologies Ltd* has released *e-Sweeper*, a tiered content security solution aimed at ISPs and ASPs.** The product, powered by *MIMEsweeper*, scans email before it is delivered checking for spam, viruses, hoaxes and malicious content and quarantining suspicious messages. ISPs retain control of their email and can configure all aspects of *e-Sweeper* via co-branded Web sites. For more information contact Catherine Jamieson; Tel +44 118 9301300 or see the Web site http://.www.mimesweeper.com/.