ISSN 0956-9979 AUGUST 1999

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **All write now?** Our extended Letters pages reflect the strength of feeling on anti-virus subjects ranging from testing criteria through unconventional cure to unnecessary idolization by the press. It all starts on p.4.

• **Fancy a hot potato?** *NAI's* Stefan Geisenheiner tackles the impending problems which face *Microsoft's* latest offspring, *Office 2000*. His feature starts on p.13.

• **Prior preparation prevents:** you know the rest… This month, both the Comment page and our Corporate Tutorial suggest that discipline and teamwork are essential in addressing the virus threat.

• **Hackers in black:** Ex-*VB* editor Richard Ford reports from DEFCON 7 (see the News section) and the BlackHat Briefings. Catch up with what happened where on p.17.

CONTENTS

COMMENT

The Politics of Policy

Most people know what they have to do to protect their property or which steps to take to be eligible for compensation in the event of a burglary or car accident, but are they aware of their firm's information security policy? In a recent meeting, four corporate executives strongly disagreed about private Internet use at work. Two claimed it was prohibited according to some policy nobody could recall the name of. One remembered that the CEO stated that in order to increase employees' understanding and Internet skills, it was necessary to permit them use of the Internet for private purposes at their workplace. Another said that this applied to after-work hours only.

"Fortunately, our team's experts took things into their own hands."

If executives do not know what the firm's information security and Internet policy contains we can assume that a) employees are unlikely to know the contents of such policy, and b) even if they know it by heart, why should they care unless management enforces the rules and rewards appropriate behaviour? In our research, we found that unless staff have been reminded to read the policy every sixty days, they tend to forget and thus change their behaviour in ways that may violate the policy. Something flashing across one's screen about 'Using this system… the user submits…' is no good either because, when asked, users admitted either to not reading it or not remembering it. Users must answer some policy-specific questions and be made aware that they are accountable for their behaviour in regard to information safety, AV defence and privacy.

System professionals can exacerbate the system's vulnerabilities and risks. I heard about Melissa by 27 March, while hiking in a forest, and I got worried. Student servers crash or are broken into by unauthorized users more often than most firms' servers. I hoped in vain that our system crew would do the right thing. Far from it, by late Monday morning they started to distribute the Melissa 'patch' to people who used *Microsoft Outlook*. When I got back the following Tuesday I immediately made them aware that others could pass it on too, at least the infected document. Moreover, other people may also use *Outlook* (unauthorized or authorized, always an issue on a university campus), hence, they also needed the patch. I suggested having the patch sent from our preferred anti-virus vendor to every user's desktop but most importantly, using the Internet gateway to scan all in and outgoing files. I was told that the system person responsible for this job was away.

Fortunately, our team's experts took things into their own hands. First, they broke the rules because, without prior authorization, another vendor's anti-virus software with the Melissa patch was installed on our servers and PCs. Second, we got the patch from our university's preferred anti-virus vendor's web site and installed it everywhere. Third, we provided the patch to others asking for help. Our 'unwritten policy' (i.e. what we are told by our system personnel) is that we cannot change anything in regard to anti-virus software and procedures unless initiated by them (e.g. the update is sent to every desktop when logging onto the system next time).

So what does this teach us? Most importantly, the information security policy must be understood by all employees in order to allow them to follow the policy to the letter, and the firm to enforce the policy effectively. Of course, violations will occur and if in doubt, people should be sensible and do what they find ethically and morally just, while protecting the information accordingly. Employees must be reminded of the policy at least three times a year. If our system personnel do not understand the seriousness of some of these threats and what the potential costs might be to the organization if we fail to protect our information or at least respond to a threat appropriately, a policy will do us little good. Finally, just depending on the system people to do it right, while leaving users in the dark by not providing them with the training and skills needed to understand what should and should not be done, is negligent if not downright irresponsible. While we know that humans usually do things out of self-interest, our information security policy may not necessarily reflect this. Give the users some responsibility and make them accountable while rewarding appropriate behaviour. Of course, system vendors could also make things easier for us by designing more user-friendly software but we can talk about this another time.

Urs E Gattiker, Aalborg University, Denmark

NEWS

Beware Geeks Bearing Gifts

Recently, members of the Cult of the Dead Cow (cDc) had to wipe some virtual egg off their faces. Following the much-hyped release of the Back Orifice 2000 Trojan at DEFCON 7, CDs were found to be infected with CIH. cDc's apologetic message says it all – they humbly accept responsibility for the mysterious sabotage. After all, why would they want to infect their hacking tool?

```
Somehow we must have accidentally infected
our own Defcon CDs with CIH v1.2
TTIT(Chernobyl). It was not our plan to do
this, and frankly it makes us look like
idiots.
```

VB was amused, if slightly alarmed, to receive timely press releases from *Symantec*, *NAI* and *ISS*, among others, which announced that BO2K had been summarily 'dealt with'.

BO2K, unlike its predecessor, will run on all current *MS* Win32 operating systems. It is a Remote Network Administration Tool which allows administration on any machine upon which the Server part of the software is installed. This is very comprehensive – anything from changing the Screen Saver through playing music to 'seeing' what is typed.

Once the Server program is executed on a host machine, it will run unobtrusively. Then anyone using the client program can gain remote access to the machine. Hence, it will be detected as a Trojan, though for this particular BackDoor program it will be not as effective as for others. As the source is also available, it will no doubt soon be seen in a wide variety of modified forms ▮

Parlez-vous Trojan?

In 1998, a young French hacker (JC'ZIC or Jean Christophe X) nicknamed his software 'Sockets de Troie'. The point of it was remote computer control, but he swore he just wanted to play jokes on his friends. The software consisted of two components: a Server program and a Client program. However, the Server program (the Trojan) was unusual: the Server part of the hacking engine spread with a virus and was installed in the target computer at once as an immediate payload.

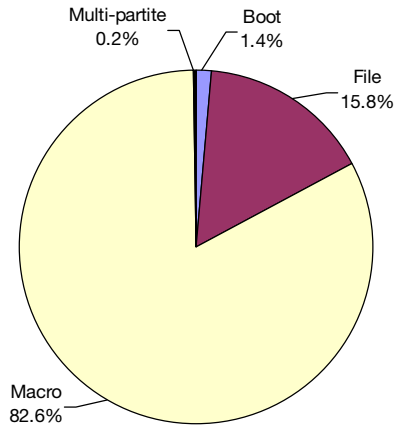
SOCKET23 was launched from his web site and immediately infected major French corporations between August and October 1998. The virus (distributing the Trojan) was known as W32/HLLP.DeTroie.A (alias W32/Cheval.TCV). Never had a virus so disrupted French industry. The author quickly offered his own remover and made his apologies on his web site (now suppressed). Jean-Christophe X (18) was arrested on Tuesday 15 June 1999 in the Paris area and placed under judicial investigation for 'fraudulent intrusion of data in a data processing system, suppression and fraudulent modification of data' ▮

Prevalence Table – June 1999

Virus	Type	Incidents	Reports
Class	Macro	515	13.4%
Ethan	Macro	474	12.4%
ColdApe	Macro	459	12.0%
Win32/Ska	File	455	11.9%
Marker	Macro	333	8.7%
Pri	Macro	285	7.4%
Cap	Macro	204	5.3%
Melissa	Macro	176	4.6%
Laroux	Macro	167	4.4%
Tristate	Macro	79	2.1%
Win32/ExploreZip	File	79	2.1%
Form	Macro	61	1.6%
Npad	Macro	58	1.5%
Footer	Macro	56	1.5%
Win95/ClH	File	43	1.1%
Walker	Macro	33	0.9%
Temple	Macro	31	0.8%
Groov	Macro	26	0.7%
Concept	Macro	24	0.6%
Taguchi	Macro	24	0.6%
AntiEXE	Boot	24	0.6%
Appder	Macro	21	0.5%
Munch	Macro	15	0.4%
Others [1]		191	4.9%
Total		3833	100%

[1] The Prevalence Table includes a total of 191 reports across 66 other viruses. A complete summary can be found at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

[*This month's letters are more controversial and topical than ever. From AV professionals to home users, everyone wants to have their say, including us!* VB's *Technical Editor Jakub Kaminski plans to respond to our virus writing correspondent JK next month. Ed.*]

Performance versus Protection

Over the last decade, we have seen a small number of significant events drastically change the nature of the virus problem and the nature of the solution for customers. The first such event was heralded by the creation of the macro virus back in August of 1995. Call me paranoid, but I honestly believe that the events of recent months have brought the industry to another critical juncture in the evolution of the virus problem.

First, the Melissa virus was definitely a 'wake-up call' for corporations and it showed how vulnerable to attack our networked systems are. Increased connectivity will make the next ten years the decade of the Worm, and corporations need seriously to consider the trade-offs between programmability and security in their applications, operating systems and email systems. Anti-virus technology and swift prosecution by law enforcement agencies will help to stem the problem; however, companies should start considering filtering all executable content including macros not signed by trusted sources at the corporate gateway and deploying (and encouraging vendors to produce) macro-free versions of corporate software whenever possible.

Second, in the last few months, virus authors have determined new ways to sneak Worms past AV software. They inserted the original Melissa virus into files with uncommon file extensions that would not likely be in an anti-virus product's default extension list (e.g. EXE, DOC, XLS, etc.). In the most recent example, the perpetrator used '.I', e.g. 'MyDocument.I' instead of 'MyDocument.DOC' to spread the Melissa virus. Since documents don't usually have a .I extension, anti-virus products neglected to scan these files allowing them to pass into corporations.

With a file extension like .I, how does Melissa spread? When a user opens an email attachment (or any file on the computer), the extension is used to determine which application to launch to edit/view the file. If *Windows* does not recognize the extension, it will examine the file's contents to determine which application to use. Consequently, if the user clicks on a *Word* document named 'MyDocument.XYZ,' the system will examine the contents of the file, determine it is a document and automatically launch *Word*. Unfortunately, *Word* will then run the document's malicious macros!

This means that email-based computer Worms can spread by sending infected document attachments with any arbitrary file extension. And the moment an unsuspecting user clicks on that attachment, they will launch *Word* or *Excel* and spread the Worm! It also means that unless customers configure their anti-virus products to scan every single file, we may see a large number of Worms sneak right past anti-virus protection!

How can companies protect against such threats? At a minimum, your company should configure your gateway and group-ware anti-virus products to scan all files, regardless of extension before allowing any attachments into the enterprise. Furthermore, given that users often receive and read private email directly through the Web, it would be a good idea to configure all desktop and server anti-virus software to scan all files too. Given the rapidity with which Melissa and other worms spread, I would suggest that *Virus Bulletin* verify that anti-virus products can provide this protection in future Comparative Reviews.

[*Since the test-sets are aligned to the latest WildList – announced two to three weeks prior to the product submission deadline – the Comparative Reviews provide an accurate measure of how each of the anti-virus products adapt to the ever-changing virus scene. Users have a right to expect that the latest updates they obtain should provide protection against the latest ItW viruses. This means not just the capability of detection, but guaranteed detection upon scanning after updating. This is the very reason why the test results quoted in* Virus Bulletin *reviews are those obtained using the default 'out-of-the-box' configuration settings for a product at that particular time. Ed.*]

Finally, it is true that configuring anti-virus products to scan all files will incur additional overhead on your machines. We have been anticipating this and are working to develop our next generation of anti-virus engines to reduce overhead and make scanning all files the only acceptable solution. In the meantime, corporations will need to choose between performance and protection.

Carey Nachenberg
Chief Researcher, Symantec AntiVirus Research Center
USA

Welcome to the Real World

You're doing a fine job helping people find out which anti-virus products are providing the best protection against in-the-wild viruses with the Virus Bulletin 100% awards. But I wonder if they're perhaps not as 'real-world' as we would like to think? It seems to me that most corporations are not using on-demand scanners for their virus protection, but on-access scanners which are capable of intercepting a virus infection in real time any time an infected file is accessed.

However, the VB 100% award ignores on-access detection, and puts its entire focus on on-demand detection rates. If the award changed its focus to on-access scanning I believe that not only would your readers get a better idea of which products were providing adequate protection in the real world, but the developers themselves would have further incentive to improve their products. Everyone wins! (Well, everyone apart from the poor chaps at *Virus Bulletin* who have to do the testing!)

This may even show up those anti-virus products which are using a different virus-finding engine in their on-demand scanner from their on-access scanner! I would be very interested in hearing your views on this subject, and those of your readers.

Graham Cluley
Senior Technology Consultant, Sophos Plc
UK

Cutting the Aprone Strings

I am a 17 year old male, and I have been programming for around four years in any language that I can get my hands on to learn. My evaluation copy of *McAfee* ran out and I am too cheap/poor to run out and buy it. So, to prevent my own computer from possibly being infected by a virus off the Internet, I spent some time and wrote a type of (good) computer virus to infect my computer.

I tested my creation, that I called Aprone, and it seemed to work wonderfully. It would multiply like mad and attach itself onto every EXE file on my computer. Then it would constantly monitor itself when it was executed to see if there were any fluctuations in itself.

If any (even minor) fluctuations took place (like any data beginning to be added) it would copy its 'host' EXE's contents into a temporary file. It would then send out a distress signal and self-destruct. The signal (hidden file) would be picked up by one of many 'hives' that the Aprone strains would construct on the computer. Each hive is used to store all of the raw data that is continually collected by the hundreds of strains. With the data sorted in the 'hives', Aprones that had been assigned as workers would sift through and toss out unneeded data.

Almost instantly, the signal of the Aprone that had been attacked would be checked and another Aprone (that had already been made and was waiting for its orders from the hive) would be given the small amount of sorted data that was associated with the file that was destroyed. The new strain would then rebuild the EXE file using the information dumped into the temp file and infect it, taking the dead Aprone's place and keeping the EXE untouched.

I discovered problems that would make my idea dangerous if it became as common as anti-virus scanners. Since virus writers are able to get their hands on scanners, they are able to create viruses that can usually be immune to them. That is why I feel that virus scanners are becoming more and more ineffective. If these 'good' viruses became common and used by thousands+ people, then a virus writer somewhere would eventually be able to alter it and cause it to work like an everyday 'bad' virus.

Another bad thing about this would be that the entire stage of infecting the computer without being detected would be completely removed for the virus writer because the 'good' virus had already infected the computer. This is why, as long as the virus writers don't expect it and can't get their hands on it to study it, this would be an excellent way to prevent virus infections. The only safe way that I can think of would be to make many different 'good' viruses and put them randomly in boxes. People would buy the same product but (most likely) would actually have different viruses. Thus, someone learning to alter one may not ever actually find another person with the same exact virus. My method of using a virus to stop other viruses has worked very well for my own PC. It makes sense… we use guns against guns, and bombs against bombs don't we?

Jeremy (JK)
AproneJK@aol.com
USA

A Romanian National Hero?

I have been waiting with great interest for an article in the Romanian publication *Privirea* about Romanian virus writers and hackers. I personally used to respect the quality of the articles published by the above mentioned magazine, as they used to be fair and, more to the point, correct when it came to the technical aspects of computer life.

That's why, when I got the magazine on my desk, I quickly browsed through its pages to find the relevant piece and read it. The article itself (called 'The Information Reformers') is quite interesting, as it's based on an interview with one notorious Romanian virus writer, known by the nickname muRPhy. Those of you who are directly involved in anti-virus product database maintenance and updates should be familiar with the Dodgy and RP virus families, which have been reported in the wild – RP.A is still on the WildList (http://www.wildlist.org/.).

This virus writer is well-known as one of the authors of the above mentioned two viruses, as well as co-author for other viruses, most of them with highly destructive payloads.

What is the article about? Mainly, it describes muRPhy's first experiences with computers, cracking and virus writing. More precisely, this guy started his computer career while he was in the tenth grade, where he used to 'work' over six hours each day with his PC, cracking software and playing games. Six hours are depicted as a huge, impressive amount of time for the young apprentice.

So far, to someone familiar with the generic, mediocre virus writer profile there's nothing unusual here, except that probably the majority of the readers of the publication are not familiar even with the most common computer soft-

ware, not to mention something like hacking and virus writing. If six hours seems quite a big deal to such people, to me, I have to say, it is really pathetic. Some of my friends used to do real work at the same time (not cracking or playing) over ten hours per day, and even more sometimes. It goes without saying that a good anti-virus researcher works an average of over 90 hours per week, or even more than 110 or 120 hours per week (I'm sure many of you can confirm that).

The next problem came when this hard worker discovered assembler code, and as the article proudly reports, this self-teaching guy with quite a high IQ wrote the RP.A boot virus. Nothing special, just that the virus contained a payload which triggered on 17 December, wiping the MBR of PC with trash, rendering the system unbootable.

Very happy with their 'cool' creation, muRPpy and his friend, whom we know under the nickname of 'RP', heard some time after they wrote the RP.A virus that anti-virus companies had managed to sell many programs and make good money because of their creation, and this made them really mad. They quickly wrote a better, more advanced boot virus, which specifically targeted the users of my anti-virus program, *RAV*.

At this point, I should say the original article is really quite fun to read, as it contains a nice definition of what we call a computer virus: 'a very small, well-optimized program which executes very fast, and either hangs up the computer, or deletes all the data from it'. As far as I'm concerned, no comment is needed for this 'definition'.

Oh well, our 'friend' has even more to say: 'since 1996, when I wrote the virus (which for those curious, is the Dodgy virus) I lost my interst (sic) for virus writing, as I can anytime "invent" an undetectable virus'. I can of course only welcome the fact he stopped writing viruses, but on the other hand, I regard the claim of 'inventing undetectable viruses' as puerile and stupid.

No one can claim to be able to detect any possible virus before it is actually written. However, after we, the anti-virus people, get our hands on a new virus, almost nothing can stop us from adding detection and disinfection to our programs as soon as possible.

'Master Qui-Gon, more to say, have you?' says a character from a recently released movie. Oh, yes, and our young virus writer has even more to say in the article. He describes how he and his friend used to visit company stands at computer fairs, and infect their PCs with their viruses. 'It was pretty funny' says muRPhy…

They also used to come back days later, and check the infected computers. Most of them were not working anymore, and sometimes the victim company had to close the stand because their computers were not functioning, and failed to boot because of the virus previously planted there by the two virus writers. As if that were not enough, I was myself visited by muRPhy, pretending to have a problem with his unbootable computer, and trying to make me suggest solutions for what seemed to be a CIH-damaged Flash BIOS. And this mainly to 'test' my skills, and see how smart I am, and if I would manage to guess what his problem was.

As a rather poetic and significant aside to this story, despite having such a high IQ, as the article states, muRPhy seems to have wiped out his own home computer with the CIH virus on 26 April.

One might wonder, what is muRPhy now doing for a living? Easy – cracking software for money. He 'explains' to the reader: 'I was never interested in hacking. It is far much expensive (sic), and doesn't require too much intelligence' – my comment would be that it might not take too much of your intelligence if you don't have any.

With regards to cracking software for money, I'm amazed how stupid people can be – would anyone really pay such a guy to crack a computer program in order to remove a time-lock protection?

Oh yes, it seems so, because the two reformed virus writers seem to make 'good money' from this business. And as 'big finish' conclusions, a quote from one of their friends removes any doubt anyone could harbour about such an honest and benevolent activity as cracking: 'after you become a grown-up, you can even make a vocation out of this hobby'…

So, this is my country, where someone can make a living from cracking, hacking and virus writing. No more words needed. Or, tons of words left to say. Like, for example, despite the fact I know the real names of both muRPhy and his pal RP, and having met them both face to face not one time but many times, neither I, my company, nor even the unlucky users who have had their disks trashed by their viruses (and there are hundreds of such cases, if not thousands) can do a thing about it.

I personally regard this problem as a worldwide one not just limited to my country, or to other Eastern European countries. Since there is no one in government skilled enough to understand the problem of computer fraud, (yes, that's right, and I can argue that with anyone trying to say 'not quite true, we actually have trained people, etc… ') and no one in law enforcement is able to use legislation to convict people like muRPhy and RP, such cases remain the problem only of the anti-virus developer.

We have to fight their creations and of course, gain evidence in preparation for a better time, when sending these two to jail for the evil they've done to the world computer user community will be not only possible, but really achievable and even mandatory.

Costin Raiu
RAV Team Leader, GeCAD
Romania

VIRUS ANALYSIS 1

Seek and Destroy

Steven Braggs & Richard Wang
Sophos Plc

A few months ago we were quite clear about what a virus did and what kind of threat it posed. To be effective, a virus had to spread, unnoticed, to as many machines as possible before its payload was triggered. A virus that unleashed a highly destructive payload instantly would be stopped in its tracks and spread no further. This was great for the anti-virus industry as we were most likely to see samples of new viruses long before their payloads triggered. If users kept their anti-virus software up to date and downloaded the latest patches from their anti-virus vendor's web site they were safe. Unfortunately, the recently discovered Worm ExploreZip is different.

March 1999 saw the first virus which exploited email to accelerate its spread. Like Melissa, ExploreZip uses MAPI commands and spreads between corporations and networks via email attachments.

If you can get a virus, or Worm in this case, to spread very quickly, why would you wait before executing the payload? ExploreZip is particularly nasty. As stated in last month's news story (p.3), it truncates all files with extensions C, H, ASM, CPP, DOC, XLS, and PPT to zero length. Unfortunately, this happens straight away. There is no trigger condition, no date to wait for, no counter – it just does it.

Installation

Both Melissa and ExploreZip rely upon the user launching an infected object from an email. In the case of the former, it was a document. With ExploreZip, it is an executable. Surely double-clicking on an EXE file received in an email is one of the silliest things you can do? Well, yes, but… the author of ExploreZip uses two tricks to make doing just that seem perfectly safe.

Firstly, the email itself will have come from someone you know. It will be an email you expect to receive. The ExploreZip worm searches your inbox to find emails to reply to and sends the reply:

```
"Hi Fred (read your own name for Fred)
I have just received your email and I shall
send you a reply ASAP.
Till then take a look at the attached zipped
docs bye."
```

So, it is from a trusted source and you do not think it is an EXE file, not unless you look closely. The file is called ZIPPED_FILES.EXE and what is more it has a Winzip icon. We are sure most users will think it is a Zip file. The ZIPPED_FILES.EXE program is quite large – 210,432 bytes – and it is written in the high level language Delphi.

Once the ZIPPED_FILES.EXE file has been run, a complicated chain of events is set in motion. Firstly, the worm starts four separate threads – the installation thread, two payload threads and a MAPI thread.

This section discusses the installation thread. If the program name is not EXPLORE.EXE the dialog box below appears on the screen.

An exact copy of ZIPPED_FILES.EXE is then made in the system directory under the name EXPLORE.EXE. Under *Windows 95* the appropriate directory is windows\system, under *NT* it is winnt\system32. A call is made to the WritePrivateProfileStringA function in the *Windows* API. If the operating system is *Windows 95* a line is added to the section of WIN.INI similar to this:

```
Run=c:\windows\system\explore.exe
```

Under *NT* an appropriate registry change is made. Usually the …\WindowsNT\CurrentVersion\Windows\Run is set to 'C:\WINNT\System32\Explore.exe'.

Both these changes have the effect of making sure that EXPLORE.EXE is run the next time the machine is started. EXPLORE.EXE is an exact copy of ZIPPED_FILES.EXE, though, and it will behave in exactly the same way. However, the user will not see the message about the file being incorrect. It is all too easy for users to confuse the name EXPLORE.EXE with EXPLORER.EXE and assume nothing is wrong.

Method of Spread

The Worm has two methods of spreading – via email and via networks. (The second method of spreading is closely connected to the Worm's payload and we will describe the two together later.) The MAPI thread controls spread via email. When the Worm is first installed it will search the user's inbox in their email client for new messages. If there are unread messages it will reply to them using the message above and the attachment ZIPPED_FILES.EXE.

The MAPI thread will remain active and wait for any subsequent messages to appear and reply to them. It is capable of distinguishing between new messages and those it has already replied to and will not reply more than once to the same message.

The code mentions *Microsoft Outlook*, *Outlook Express* and *Microsoft Exchange* by name. In theory, it should be capable of working with any MAPI-compliant email client.

In our testing it appeared not to work well with non-*Microsoft* clients. It is conceivable that it will work with different clients depending on their precise configuration.

Payload

There are two payload threads. One searches all lettered drives accessible from the infected machine. The other searches all networked drives regardless of whether they are mapped or not. The first of these threads is active all the time, the second runs only once and terminates.

Both threads use the same part of the code to search all files once they have identified a suitable drive on which to search. Initially, the files found are compared to WIN.INI. If WIN.INI is found on any remote drive accessible to the infected machine, the Worm is spread to that machine. This is achieved by copying EXPLORE.EXE to the target machine under the name _SETUP.EXE. The WIN.INI found on that machine is then altered so that _SETUP.EXE is run next time that machine is restarted.

_SETUP.EXE acts similarly to ZIPPED_FILES.EXE. When it is run, the same dialog box is displayed and the installation routine is triggered in exactly the same way. This works fine on *Windows 95* machines. _SETUP.EXE also removes any reference to itself from WIN.INI. Under *NT* things are different. The WIN.INI file is just there for compatibility. *NT* will not run commands in this file on restart in the same way as *Windows 95* does. Hence, if the worm is lucky enough to find a WIN.INI file on an *NT* machine in a shared directory, it will not be effective in spreading to that machine. To activate the Worm the user would have to run the _SETUP.EXE file manually.

After WIN.INI, the Worm is interested in any files with extensions C, H, CPP, ASM, DOC, XLS and PPT. When one of these is identified a file of the same name is created which is initially 128 bytes long. This new file is then truncated to zero length and closed. If you try to recover any lost data you will find that the field in the directory entry that points to the first cluster in the chain has been set to null. Recovery of truncated files is virtually impossible.

Any file of the right extension that is accessible from the infected machine anywhere on the user's network will be truncated to zero length. As payloads go it must be one of the nastiest we have seen. A noticeable side effect of these threads is a considerable increase in network traffic, especially once several machines have become infected.

Worm or Virus?

Classification of ExploreZip is a grey area. A virus needs a carrier; an executable virus attaches itself to program files, a macro virus to documents. You could say ExploreZip attaches itself to emails, but this is not strictly true. The only email it is ever attached to is of its own creation. Email is more the medium through which it spreads. Also, as an EXE file on an infected machine it is not associated with any other EXE file. However, you could say it infects WIN.INI files, but that is just to make sure it is always being run. On balance, we have opted for Worm rather than virus, but make up your own mind. Whatever you call it, it is definitely something you do not want on your system.

Recovery

Recovery of an infected machine can be difficult – an infected network doubly so. All machines carrying the Worm should be isolated from the network as soon as it is discovered. This prevents the Worm spreading to machines once they have been cleaned. On any single machine you must shut down all tasks with the names EXPLORE.EXE, SETUP.EXE or ZIPPED_FILES.EXE. There may well be several different ones running at the same time.

Instances of the Worm are only displayed by pressing ctrl-alt-del under *Windows 95*. Under *NT* you will need the Task Manager to display the list of active processes. Then delete any reference to either EXPLORE.EXE or SETUP.EXE from the registry (*NT*), or the WIN.INI file (*Windows 95)*. Reboot the machine and check the Worm is not running under any of its various guises. Finally, search the machine for files called ZIPPED_FILES.EXE, _SETUP.EXE and EXPLORE.EXE and delete them. It would then be as well to examine any new mail for other instances of the Worm. Sadly there is no simple way to recover any truncated files other than restoring from backups.

Preventing It

The moral of the story must be never launch an executable file directly from an email, even if it is from your most trusted friend. Users must be educated. They need to ask 'What am I actually doing when I double click on this file?' If the answer is 'I am running a program', stop. On-access scanners will protect against this Worm, but will require updates to protect against new variants. To restrict the network method of spreading, consider which networked resources are shared and limit access accordingly.

W32/ExploreZip	
Aliases:	None known.
Type:	Worm.
Spread:	Via email and across networks.
Files used by the Worm:	EXPLORE.EXE, ZIPPED_FILES.EXE, _SETUP.EXE.
Recognition in memory:	EXPLORE.EXE running as a task under *Windows 95* or *NT*.
Removal:	Close all tasks relating to the Worm. Delete any instances of it in files. Check emails for attachments.

VIRUS ANALYSIS 2

Introducing the Infidel

Marius van Oers
AVERT Labs, Netherlands

The first report of a {W95,W97M}/Heathen infection at a customer site came from Denmark at the beginning of June 1999. Prior to that, we at *AVERT Labs* found viral samples of Heathen at certain 'adult' websites and newsgroups. [*All in the name of research, naturally! Ed.*]

Basically, Heathen tries to infect *Word 97* DOC/DOT files using a patched EXPLORER.EXE on *Windows 95* systems. After a reboot, when the patched explorer is being run, it searches for other DOC/DOT files to infect. Further *Word* documents can be infected without *Word* being open. However, many infected files become corrupted and will not run properly any more – they will simply crash. This behaviour was one of the things the customer noticed, and after re-installing the operating system and *MS Office 97* again, the indication that they might have a new virus became clearer.

DOC Code

Heathen code in *Word 97* files consists of a viral module about 243 lines long called 'NewMacros'. Heathen makes use of subroutines, using the Kernel32 library, with names like Callback12 and Callback24, employed in the different stages of the infection processes. It also declares a function called CreateMutexA to determine whether it needs to infect. Further, it declares functions like GlobalAlloc, GlobalFree GetModuleHandleA and GetProcAddress for resource usage/allocation/modification, to check and initialize the environment in order to be able to patch EXPLORER.EXE through HEATHEN.VEX.

The virus uses the AutoOpen macro to initiate upon loading an infected DOC file in *Word 97*. First it checks/compares whether the operating system is *Windows* by using the InStr function. If InStr(1, System.OperatingSystem, 'Windows') <> 0 then…

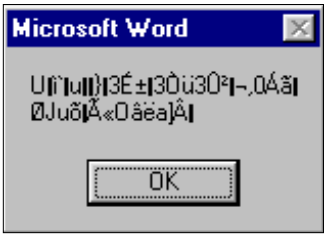
On *Windows 9x* systems the returned value for System.Operatingsystem is 'Windows'. On *Windows NT v4.0* the returned value is 'Windows NT'. Since the (In-String) check is just against the string 'Windows', the value for InStr(1, System.OperatingSystem, 'Windows') will be 1 for *Windows 95/98/NT v4.0* systems. However, Heathen does not infect *Windows 98* or *NT v4.0* systems successfully, it only spreads completely on *Windows 95* systems.

The virus determines if it needs to infect further by checking for the string value 'Heathen is here': If CreateMutexA (0, 0, 'Heathen is here') <> 0. It is not absolutely clear what the expected values for this expression are, so a variable was added to the code and it was measured.

Typical values of this comparison are in the range of 256, 312, 364… 456, 464 etc. It seems that the values are divisible by four and moreover, it seems that upon running the viral code from the same infected sample multiple times, the counter may add four or eight each time. However, the value is not constant and thus the bug results in the 'already present Checking' failing and thus the virus starts to infect over and over again each time.

The virus then declares a string variable called 'a' and sets its value using chr(xx) syntax verbs:

```
a = Chr(85) + Chr(139) + Chr(236) + Chr(96) +
Chr(139) + Chr(117) + …
```



Sometimes this trick is used in macro viruses to make it less obvious, for example, to hide a message. However, upon converting from ASCII code into Letter, in this case, no visible, easy-to-interpret code results. Another reason for 'encrypting' viral code is to try to fool heuristic scanners which may look for, for example, MacroCopy instructions.

Finally, as we will see later, the use of 'pre-compiled' code can be used for VBA viruses to drop and/or modify COM/EXE/DLL files at the system in use. Dropping infected 32-bit 'PE' EXE/DLL files is possible and a number of viruses use this trick successfully.

Here, the Heathen virus starts/prepares the file modification code. Next, it makes use of 210 'pre-compiled' code-lines (see below). These are in different blocks and are used to drop and/or modify certain files, as we will see later on.

```
s(0) = "05Y=01P0100000l00?oo0;P0000000006P1
00000000000000000000000000000000"…
```

The last 'pre-compiled' code-lines are:

```
s(208) = N3=6<i0c[C>T<kdcd3?
5=1DcC3A6=58dGSAH=6@dL3AZ=7HdPSAl
=8PdU3B>=9Xd"
s(209) ="YSBP=:`d^3Bb=;hdbSC4==0dg3CF=
>8dkSCX =?@d03Cj=@He4SD<000e00000000"
```

Heathen allocates the environment system using the declared GlobalAlloc function: b = GlobalAlloc(0, 12288). The returned value is not constant and varies slightly upon subsequent measurements. If the GlobalAlloc is not zero then it uses the callback12 routine to go over and copy its

209 codelines:

```
For c = 0 To 209
Callback12 s(c), b + c * 48, a
Next c
```

As we will see later, the HEATHEN.VDL holds the viral code for the 32-bit infection part, its file size is 12,288 bytes. The callback12 subroutine has three arguments. The first, s(c), is the 'pre-compiled' code line identifier, ranging from s(0) to s(209), string value. The second, b + c * 48: b is a numeric value, determined by GlobalAlloc(0,12,288), c is numeric value from 0 to 209, so the second variable was declared as long. The third, a, is a fixed string value consisting of the chr(xx) verbs.

Finally, the virus declares some variables using the declared functions at the beginning of the code:

```
d = GetModuleHandleA('KERNEL32'),
e = GetModuleHandleA('OLE32')
f = GetProcAddress(d, 'GetProcAddress')
Callback24 b, Word.ActiveDocument.FullName,
f, d, e, b + 1536
```

So here the virus makes use of the second callback routine, callback24. Using the two callback routines, 12 and 24, the virus can patch the EXPLORER.EXE from an infected *Word* file which searches for, and infects, DOC/DOT files. Once done, the virus uses the the GlobalFree command to free up the allocated resources which it reserved with GlobalAlloc. Finally, it disables the *Word 97* macro virus protection, setting the Options.VirusProtection to False.

Heathen's Files

After running an infected W97M/Heathen document, the virus drops files like HEATHEN.VDO, HEATHEN.VDL and HEATHEN.VEX in the \windows directory. Furthermore, to patch the EXPLORER.EXE, it uses WININIT.INI. The HTMP.DOC file is a temporary file used by the Heathen virus to get HEATHEN.VDO created. The HEATHEN.VDO file contains the viral code source in *Word* format. It looks like a regular *Word 97* OLE2 file, with DO CF file header etc., but it is slightly different. The extension VDO is not automatically linked to, for example, *Word*. If *Word* is opened and when HEATHEN.VDO is specifically loaded into it, then it is not possible to see the macros under Tools/Macro.

When trying to view the possible viral code with the Visual Basic Editor (VBE) it seems that the template project is protected by a password. There was no spreading encountered when this file was opened in *Word*. Infected DOC/DOT files also seem to become the 'template project password protection' and it is not possible to see the viral code with the VBE.

HEATHEN.VDL is a file with the viral code source in 32-bit (PE) format. This file holds the viral code for the 32-bit infection part. It is 12,288 bytes long and has four entries in the Object Table named Code, Data, Idata & .Reloc. In the code of the HEATHEN.VDL file itself are also references to HEATHEN.VDL, not surprisingly, as the viral code is fixed to drop some pre-determined files.

The virus does not write its full code but instead modifies/patches EXPLORER.EXE so that it calls the viral code from the '32-bit viral code holder' file, HEATHEN.VDL. It determines the name of the *Windows* directory with 'getwindowsdirectory' to be able to create/modify/delete files in the correct directory. It uses the KERNEL32.DLL to get/reserve memory allocation.

It is hard to modify files if they are in use, so Heathen uses a WININT.INI file to make the necessary (renaming) changes upon the next reboot. Once done, this WININT.INI gets deleted. Its previous existence can usually be traced by searching for the WININIT.BAK file:

```
C:\WINDOWS\Explorer.exe=C:\WINDOWS\Heathen.vex
```

The now patched EXPLORER.EXE searches for DOC and DOT files to infect. After startup, the virus may search for and infect *Word 97* DOC files without *Word 97* being open. Using FindFirstFile, FindNextFile it searches for target DOC /DOT files to infect using HEATHEN.VDO as viral source. It searches for them on 'LogicalDrives', opening and checking them. It searches the WordDocument 1Table for ScanData and HeathenWC macros for 'Heathen is here', using KERNEL32.DLL, USER32.DLL and finally OLE32.DLL, modifying them and write/closing them. ScanData is used to keep track of 'target-document' search progress. It seems that the virus searches at time-intervals, after about four minutes.

The viral routine to infect DOC files from the patched EXPLORER.EXE has some severe bugs and may result in many *Word 97* DOC files that will not run properly any more. Nevertheless, these files are infected and the Heathen virus caused a lot of trouble at a huge customer site. Heathen's payload may erase the following system files: USER.DAT/.DAO and SYSTEM.DAT/.DAO. On the test system, the payload triggered after setting the clock more then six months ahead after first infection date. As the virus rapidly corrupts many DOC/DOT documents, it will certainly not go unnoticed and with updated anti-virus software it is expected that the payload will not occur.

{W95,W97M}/Heathen

Aliases:	{Win95,W97M}/Heathen.12288.
Type:	Patches EXPLORER.EXE to search for and infect DOC/DOT files.
Payload:	Might erase USER.DAT/.DAO and SYSTEM.DAT/.DAO.
Detection and Removal:	Use current anti-virus software to detect and clean the *Word* infections. Delete infected files and replace clean EXPLORER.EXE.

INSIGHT

King of the Hill

[*Ian Whalley has been Editor of* Virus Bulletin, *prominent in* Sophos *and now he has crossed the pond to the Big Apple to work for* IBM. *And he's still only 26! Here, Ian tells his own story and reassures us that viruses are still his major preoccupation. Ed.*]

I was born in 1973 – that will make me 26 at the time of publication – in the university town of Oxford, slap bang in the middle of England's green and pleasant land. Education inevitably ensued, as it so often does – in my case, it was the fairly predictable set of science-related subjects at school. University came along in 1991. I studied (in a fairly loose sense) Physics and Computer Science at Manchester University in the UK. The degree was what American universities would call 'double major', but in the UK it is called 'joint honours' – both subjects count for an equal part of the final mark.

It was this fact that cost me a better degree. Whilst Computer Science was a walkover, and great fun, Physics came across as a long hard struggle away from rationality into madness. Physics at university is depressing from the start, when the lecturers all tell you that everything you learnt at school was, in fact, entirely fabricated. It was shortly after this that expressions like 'quantum uncertainty', 'wave-particle duality', and 'non-linear equations' left their lips – it was downhill from there.

In spite of the Physics problem, university was exactly what I believe it should be, terrific fun. I stuck with the Physics, due mainly to irrational determination to finish what I'd started. In the meantime, however, I was teaching myself about the many wonders of Unix – my final year project in computer science was a system to monitor traffic on the research network – a project which, in retrospect, seems like an absolute gift to someone like myself.

The first time I encountered a virus was on a rare foray into DOS at university. In the house where I lived, we had a single PC (running *Linux* – which at the time was completely unknown and very new). It was a 386 with 4 MB RAM, and we used VT100s to log into it from other rooms to work on our projects, and (to a somewhat greater extent) to develop a multi-user game that some of us ran from the university computers.

When it was not being used for work, games were played, from DOS, of course. One day, things were a little peculiar – mysterious messages about being out of memory, and plaintive complaints talking about 'executable formats' would appear for no readily apparent reason. It transpired that the machine had a variant of the Athena virus. How did we find this out? We ran *MSAV* and it found and removed all instances of the virus. Perhaps not very exciting, but notable for one thing – proof that *MSAV* did help real users, and was, at one point, actually useful.

When I left university in 1994, I interviewed with a large number of computing-related companies, in a wide variety of fields. Almost all either came over as ponderous and unimaginative, or (and let's be honest) did not make offers. In the end, I got a few – I was only interested in the one from *Sophos*. I have vivid recollections of the non-traditional interview process, and once I had received their offer, it was fairly clear that the decision had been made, in spite of the fact that it was by no means the best offer.

So, for several months at *Sophos* I learnt about computer viruses. At the time, the industry was deep into the anti-polymorphism game, almost everything was for DOS, and life was complex, but compared to now, simplistic. The anti-virus industry was much like myself at that point – young and fairly naïve. This is not to say that the work was easy, far from it, my mind is not ideal for analysing viruses, a pursuit that requires a very specific type of person. Following this work, I worked on *SWEEP for Windows NT* – the first anti-virus product for *Microsoft's* new operating system – so new that even on a top-of-the-range PC, it was utterly painful compiling.

A few months into 1995, the opportunity arose to move to *Virus Bulletin*. Physically, this was not much of a move, as *VB* was, and is, a sister company of *Sophos*, but it was a completely different type of job. Whilst I had written a couple of pieces for *VB*, with mixed success, working as Editor was a different thing altogether. It was definitely a challenge. For a while, I was accused by one so-called expert (he knows who he is) of being merely a *nom-de-plume* of one of the directors of *Sophos*.

For the uninitiated, being Editor of *VB* is considerably more effort than it might at first appear [*You tell 'em, Ian! Ed.*]. I have deduced, by observation and personal experience, that the meantime to failure of a *Virus Bulletin* Editor is just over two years – all Editors so far have lasted between two and three years – I was no exception.

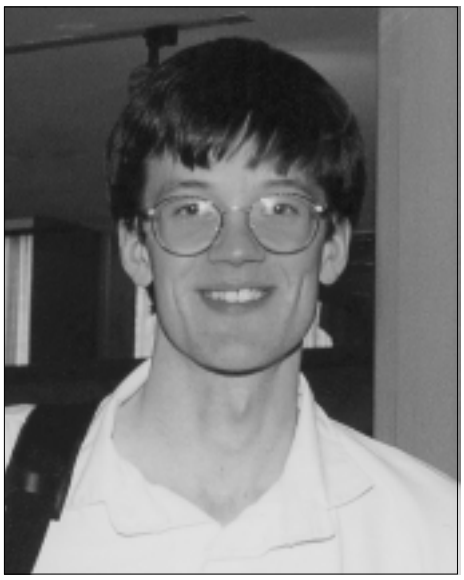
In 1997, I left *VB*. I returned to *Sophos* where I took on a number of tasks ranging from utilising my knowledge and fandom of Unix to run a variety of machines providing Internet services, through development work on components of the next-generation of *Sophos'* virus detection technology, to leading the team developing versions of *Sophos Anti-Virus* for a large number of different flavours of Unix. I continued to work at *Sophos* until mid-1999.

I left the company, and indeed the UK, at the end of May, 1999. At that time, I moved to a location a little north of New York city (allegedly the greatest city in the world, if

the words 'greatest' and 'city' are not contradictory), and began work at *IBM's TJ Watson Research Center* in Hawthorne, NY. The group in which I now find myself is the group designing and developing on the now-famous Immune System.

I am currently gathering observations on American life that will no doubt, one day, form a Bill Brysonesque series of memoirs documenting the culture-shock that only the truly out-of-place can feel… I am confident that chapters entitled 'The American Behind The Wheel' will take up large sections of these books.

For the moment at least, I am staying with viruses – anti-virus is, after all, the field in which I am best known. I am also keeping up with my skills in other areas of computer security. For the anti-virus 'industry', as I have always called it, in spite of the fact that the term is far too grand-sounding, the field is going to continue to change rapidly.



We have seen the start of this metamorphosis over recent months, what with Melissa, ExploreZip, PrettyPark, and JulyKiller – all pieces of malware which burned brightly, but only for a short period of time, fast burners all.

Malware of the future will have a much shorter half-life than we have been used to in the past. Back in the 'good old days', viruses appeared, stayed in the wild for months and then years, only to eventually die out, due predominantly to environmental factors.

Malware will appear in the wild, and suddenly spread extremely widely – a fantastically high burn rate. It will be necessary to create products and systems that cause the attrition rate to be appropriately scaled up also – environmental factors will not be enough to kill off such malware in a reasonable timeframe, nor will monthly updates on CD.

One does not have to be a genius to make certain predictions in the short term – macro viruses will, naturally, continue to be the most problematic items of malware in the 'real world', and we will see more and more network-aware viruses. It is more intellectually interesting to speculate on the long term – or at least the long term as far as the world of IT is concerned – what will be happening in five years? If anyone can accurately predict that, I would be grateful if they would get in touch and let me know where to invest!

Elsewhere, we are gradually going to have to stop concentrating on 'viruses'. This is currently a very restrictive area for people to be working in, and products and companies that only deal with viruses will find that they are not protecting their users adequately. The customer wants something that will take care of all malicious programs ('malware') which that user encounters. Increasingly, these will not be things which fit comfortably into traditional definitions of the word 'virus'.

In my opinion, the current approach – traditional scanning methods with a variety of heuristic techniques of varying levels of complexity and usefulness – will continue to be the best approach for some time to come. There is always room for a single, revolutionary step that will change the world of anti-virus forever, oh well, let's just say 'I'm still working on it!'.

For some time, AV has been in a phase of evolutionary development. Gradual refinements and modifications to current techniques are the order of the day, at least as far as the product that the user sees on his desktop.

Inevitably, however, some companies may start to find that the burden of the back-room work becomes too much. Already, anti-virus companies create products for numerous different platforms and operating environments, doing an incredible number of things most of which have nothing to do with viruses, and so on and so forth.

More obviously critical, however, is the steady, inexorable rise in the number of viruses that are 'catalogued' every month – whilst a given fraction of these will be simple, more and more (in terms of absolute numbers) will be more and more complex, and will take more and more time to analyse. Either manufacturers will have to automate the analysis of viruses for which the problem is tractable, or employ ever more people to do it by hand – and analysing viruses by hand is not the most exciting thing in the world to be doing with one's time (trust me on this).

Trojans are one obvious example of fast-burn malware. The life-cycle of a Trojan Horse program is so short as to be missed by most people – someone distributes a single program, some people download and run it, and they get stung. The Trojan is removed, and never seen again – end of that Trojan's life story. The fact that detection for that specific Trojan is added to anti-virus (anti-malware!) products a week or so later is not really much help – the threat has passed already, it will never come again. It's a *cliché*, but we do live in interesting times.

For myself, times are no less interesting. My relocation has brought me closer to the love of my life (no coincidence there), and I look forward to the next stage of my personal life. In many ways, I also look forward to the next stage of my professional life – in other ways, I do not. I concur with my friend Carey (see p.4), we are at a critical juncture in the anti-malware game.

FEATURE

The O2K Problem

Stefan Geisenheiner
Network Associates Inc, USA

Microsoft has just released *Office 2000* – the biggest and most integrated suite of *Office* applications – just in time for the millennium. The suite promises to change corporate document collaboration and information sharing through tight Web and Intranet integration. In this article we will take an initial look at the new security features, programmability and their possible effect on macro viruses.

We Are Family

Since 10 June 1999 *Office v9.0* has been available in the USA in five different suites – Standard, Small Business, Professional, Premium and Developer. Four basic applications, part of every configuration, are formed by *Word*, *Excel*, *Outlook* and *Internet Explorer 5*. The Premium suite, (the largest) includes *PowerPoint*, *Access*, *Publisher*, *Frontpage*, *Photodraw* and *Small Business Tools*. *Office* developers are guided by additional resources to ease the building of Web components and team development.

Almost all *Office 2000* applications (with the exception of *Publisher* and *Photodraw*) support at least one built-in programming language, Visual Basic for Applications 6. In addition, *Word*, *Excel*, *PowerPoint*, *Access*, *Frontpage* and of course *IE5* support Visual Basic scripts. That support also provides a powerful script design environment familiar in *Visual Studio* – the *Microsoft Script Editor*. Now native binary office formats also store script code besides VBA macros to ensure total Web compatibility.

Yes, Ignore Macros!

Possibly the most dramatic changes (as far as we are concerned) are the new macro security levels. Set to High by default, all document macros are ignored in *Word*, *Excel*, *PowerPoint* and *Outlook*, even if they should be executed at certain events. The document is opened with no warning – instead information about the macro's disabled state is presented at the event when a macro is supposed to run.

As always, there are exceptions to this rule. Developers can digitally sign their macro projects to allow users to trust them. Once the certificate information is added to the application's list of trusted sources, signed macros using this certificate will always be enabled without warning.

Similarly, all macro projects in installed add-ins and templates are trusted by default. Thus, for example, macros in NORMAL.DOT are silently enabled, unless the user has turned this default trust relationship off. To be exact, the security levels in *Word*, *Excel*, *PowerPoint* and *Outlook* only control code written in VBA. Visual Scripting is not disabled and macros written in *Excel 4* formula language cannot be signed or disabled either.

The remaining levels, Medium and Low, mostly represent the existing 'macro virus protection' in *Office 97*. While in Medium the user has a choice whether to enable or disable macros when opening a document, unless a signature of a signed macro project is invalid, causing all macros to be disabled automatically. Low allows any macro to run without confirmation prompt.

Obviously, the High security setting will protect those users who never create or use their own macros. Such people will not have to worry about the more complex settings to get a macro executed. More experienced macro users should choose the Medium setting, otherwise they will not be able to test their just-created macro.

Given that they have the ability to sign macros, developers are advised to guarantee a virus-free environment. Since a digital signature is automatically reapplied on a machine with access to the certificate, project changes due to a macro virus infection would be silently authorized. If it were to be distributed, this signed macro virus would run without warning on all *Office* installations trusting the developer or company certificate.

Fortunately, *Office 2000* also provides a virus-scanning API for third party anti-virus products to check a file during the open event of *Office* applications. Also, a macro project should always be locked for viewing by the developer, to avoid any chance of modification afterwards.

Finally, macro support in *Outlook* and *Frontpage* is restricted to one global template only ('VbaProject.OTM' and 'Microsoft Frontpage.FPM'). Still, no email or web site can contain a VBA macro. Here the VBA security settings only apply to macros in the global template.

Up and Down

Despite the macro signing and execution restrictions, *Microsoft* has kept the binary file formats compatible with *Office 97*, with the exception of *Access* database files. Furthermore, the automatic conversion of WordBasic to VBA and support for *Excel 4* formula language are preserved. Fortunately, all module copy related restrictions introduced in *Office97* SR1 are still in place, too. In general, macros written in *Office 97*, which include today's macro viruses, will work the same way in *Office 2000*, if they get executed under the security settings.

Independent of macro security, opening an *Office 2000* document in *Office 97*, or the other way around, causes a recompilation of the code, because *Office 97* only supports

VBA5. Although the differences between VBA5 and VBA6 are minor, scanners detecting W97M, X97M and PP97M viruses in *Office 97* had to be adapted to find upconverted viruses in *Office 2000* documents. Most anti-virus vendors have applied the existing detection signature for *Office 97* viruses to their upconvert. Thus, besides a scanner update, no additional signature had to be created to detect these existing viruses in *Office 2000* documents.

Taken for granted that macro execution is enabled, we should look at today's most widespread macro viruses. W97M/Class.B will not spread under *Word 2000* because its replication part depends on a default number of VBA5 project attribute lines. Still, the next time an infected *Word 97* document is opened, it will become a virus again.

O97M/Tristate.C and W97M/Melissa.A have no problem replicating under *Office 2000* and continue their spread to other *Office* applications the same way they did in *Office 97*. Fortunately, a few other *Office 97* viruses which depend on default template path names or registry keys fail because *Office 2000* introduces a new templates storage location. However, X97M/Laroux variants are able to spread in *Excel 2000* using the new template location.

Distributing Code

One of the key features of *Office 2000* is the ability to save and publish *Office* documents in HTML format, by preserving all *Office* properties not supported in HTML. When saving a document as HTML, various companion files are created in a new subfolder to assure that embedded objects, styles and application-related properties, including macros, are stored. Among those files 'OleData.mso' stores embedded streams, whereas 'EditData.MSO' holds the VBA storage information. All these files are bound together by XML tags, which are evaluated by *Office 2000* applications and *IE5* in order to assemble all the information normally stored in binary format only.

To edit the HTML file, for example by clicking on the EDIT button in *IE5*, the appropriate *Office* application is launched and all companion files containing VBA macros are silently retrieved, including 'EditData.MSO'. New data access pages in *Access* and *Excel* automatically invoke a so-called Web component – an ActiveX control (PE executable) – to provide an application-like view of the published Web site in the browser.

If *Microsoft's* long-held vision of collaborating and managing documents in one place – the Intranet – becomes reality, the average user will not notice what particular files are loaded on their machine. Other new ways of sharing documents are labelled as Online Meeting, Presentation Broadcast and Web Discussions, where users can access and edit documents created by co-workers. Monitoring and tracking the distribution of code will become difficult, despite the advantages of document-focused collaboration. Administrative policies and on-access network protection will become more important.

Administrative Power

Using the extended capabilities of the 'Office Custom Installation Wizard', administrators can enforce certain security policies. A network administrator is able to set and lock the security level users need and to create installation profiles for the company. The administrator can also lock the list of trusted sources and assign developer certificates to control exactly which macros should be allowed execution only within the High security setting. Keeping the user's profile on a server could be another way to check macro security with the added benefit of allowing user-independent backup and personalized user-access from any computer on the network.

Virus Problems

Besides scanning for existing *Office 97* macro viruses, *Office 2000* introduces a couple of new binary file formats, which need to be scanned by anti-virus products for potential viruses. It is possible that future viruses will make use of *Office 2000* applications having limited macro support like *Outlook* or *Frontpage* to plant a payload or just survive in that state.

Also, companion files of *Office 2000* documents saved as HTML have to be examined by default, particularly the macro container 'EditData.MSO'. Furthermore, the combination of VBA and Scripting support could lead to multi-application viruses containing macro and scripting code trying to exploit every possible entry to the system. Using both sides to their advantage, the VBA security settings could be turned off and *Office 2000* application email support especially would provide faster distribution potential than that we have seen in W97M/Melissa. On the other hand, distribution of ActiveX controls increases the Portable Executable exchange between systems, potentially providing more targets for PE file infectors.

In summary: High macro security in *Office 2000* provides a method to avoid the execution of unauthorized VBA code, reducing the potential of existing macro viruses to spread. When administrators and developers enforce digital signing of macro projects, users can begin to trust a macro to run. However, the developer is responsible to assure that the code signed is virus-free.

Users who have chosen a Medium or Low security setting are still exposed to today's macro virus vulnerabilities. However, the complexity of *Office 2000* documents creates uncertainty about what actually happens behind the *Office* application or browser. Considering all the integrated automation techniques *Microsoft* merged in *Office 2000*, users will get confused about where to watch for security risks and hackers will try to exploit many ways to find any open backdoor. Anti-virus products which secure network traffic will become more important in an *Office 2000* environment. It is also mandatory that the administrator studies the new features and vulnerabilities in *Office 2000* first, before installing and configuring the critical settings.

CORPORATE TUTORIAL

Early Warning Systems

Christine M Orshesky & William A Cribbs

'When Barriers Break Down' (see *VB*, June 1999, p.16) established that traditional barriers, such as anti-virus protection, no longer suffice as standalone defences against virus and malware threats. Many recent threats of this nature have attempted to exploit user *naiveté*. Knowledge of how organizations work and how human factors can be exploited are becoming more prevalent in malware attacks.

Educating users about such threats and their own role in the protection of computer systems from malicious logic can provide a solid foundation and a viable extension to the anti-virus protection program. In fact, educated users can act as an *early warning system* (the eyes and ears at nearly every level of a system) to detect malware attacks and to mitigate its impact by taking appropriate countermeasures and not further spreading/enabling the virus.

For the user population to function this way and to be able to apply limited countermeasures for malware (and other threats) initial and on-going/refresher training focused on Information Assurance (IA) must be provided. The concept of IA is to assure the security of the complete system – hardware, software, data contained in the system, and users. The purpose of the two types of training is to establish an initial awareness baseline to help users operate the systems safely and then maintain and build upon that baseline.

Licensing the User

The first step to equipping each user with the knowledge to act as an early warning system is to provide an introduction to the overall system, the threats, the systematic countermeasures, and the role of the user in the defence posture or Information Assurance process. One method is to provide initial training in the form of 'user certification/licensing'. Similar to a licence to drive on the road, user certification is like a licence to 'drive' on a network or within computer systems and ensures that the user has enough information to operate the system(s) safely. To be effective, the user must receive this information or licence *prior* to system access.

To enable a user to act as an early warning system for overall security threats, user certification should include the topics shown in the list below. Covering these topics will provide overall security or Information Assurance posture as well as the specific items to address malware threats.

Initial Training or User Certification Components:

- System Responsibilities and Procedures
- Effective Password Selection
- Physical Protections of the System
- Proper Use of the System
- Incident Reporting
- Proprietary Concerns

Malware-Specific Components to Include:

- Types, Functions and Sources of Viruses and other Malware
- Common Symptoms
- Protection Strategies
- Applicable Policies Governing the Use of AV Installation
- Applicable Policies Governing Virus Propagation
- Licence Agreements for Anti-Virus Protection
- Incident Response Procedures

All the above may be accomplished via computer-based training (CBT) or other automated means, so long as it leads to user certification or licensing. The initial training should not stop at this level if the user is to be an effective tool in the system defence. To complete the initial training, the security representative must make the basics more memorable for the user by following these four simple rules: keep it basic, succinct, current, and personal.

Keeping it basic and succinct means describing the anti-virus protection strategies clearly and in a short space of time, maybe 10 to 15 minutes. It is important to make sure that the material covered is current by using recent (within the past year) incidents or metrics. Making it personal includes specific case studies or incidents on the users' systems, illustrating the importance of the user in the protection scheme and ensuring that they have a stake in the outcome. Providing this level of training in small groups or on a person-to-person basis makes the user an active participant in the anti-virus program.

Applying the 'Last In' Principle

There are two concepts or 'laws' that come into play in the training of any user but particularly when they are to function as part of your early warning system – the Law of Recency and the Law of Repetition. The former states that the latest or 'last in' information/learning provided is most likely what will be retained and used in a given situation.

The latter states that the most critical/sensitive training must be reinforced periodically or the Law of Recency will reduce the effectiveness of your initial training as the learner moves further from the point of presentation. Taking these laws into account when designing and conducting security training ensures that the user operates as an extension of your anti-virus or security program.

As a general rule, the content of the training should progress from the general to the specific and be fairly limited. Bearing in mind that the goal is for users to retain critical information, the use of these laws helps to ensure that the user actually operates as an extension of your anti-virus or security program.

Maintaining the Early Warning System

An effective warning system requires maintenance. From the perspective of user awareness, maintenance means keeping information current and fresh in the user's mind. The longer the time between receiving and using information, the less likely the user will be able to apply it, thus becoming ineffective in the early warning system that is required in today's technology.

Multifaceted approaches to user awareness provide both extra ways to receive user information and a 'Defence in Depth' approach – meaning that the user cannot escape the information in some form. Automated means of distributing information include email, bulletins, screen savers, broadcasts across the network, logon banners, and CBT. Each of these methods have their own merits and can be used effectively depending on the type of information needed and the urgency of any actions that might be necessary as a result of the information provided.

For example, one approach might be to provide a monthly bulletin on the viruses seen in the organization the previous month and the new ones released that might affect it. The bulletin should provide basic information on each virus such as behaviour, any symptoms it may display, the ways in which the user may be exposed, and how to prevent an infection. However, if an epidemic were occurring in your organization, do not wait until the monthly bulletin to tell users about it – they could be an effective line of defence against the virus or malware continuing to spread. Here, a logon banner or broadcast message that could be sent quickly to all users with relevant information would be more appropriate. There are also more traditional means to get information to users – posters, bulletin board displays, and videos are good ways of reinforcing information and keep it conspicuous.

Engaging the Early Warning System

There is no effective substitute for interactive learning. It gives both the security representative and the user(s) an opportunity to discuss the information. There are at least two major ways to provide for interactive information exchanges: 1) meetings and 2) on-line mailing lists/ discussion groups. Organizations should consider some form of 'face time' – when an expert security representative provides information directly to the user in a formal or informal setting – on at least a quarterly basis.

On-line mailing lists and discussion groups also provide a means for the user to ask questions or make comments and receive feedback from an expert. This is a good way for the latter to ensure accurate information is disseminated and to correct misconceptions. From a user perspective, the security representative is taking time to 'speak' to them directly – that gives the impression that security is important. If the expert can make the information real by using examples of actual incidents or situations where the user is affected, the user can better relate to the information.

A Corporate Case Study

Wacky Widgets was having a tremendous problem with virus hoaxes. Over 40% of their virus-related incidents stemmed from hoax messages being sent and people reporting that they were infected with all kinds of interesting things from 'Good Times' to 'Bad Times'. In addition, their email servers were being flooded and in some cases crashing due to the massive forwarding of false information. *Wacky Widgets* knew that there were no technical solutions to the problem, aside from disabling email, so they decided that the only way to stop these 'incidents' was to rely on the users to stop forwarding them.

Processes had been in place for users to validate any suspicious or hoax message and also for disseminating information about viruses or hoax messages. The aim was to provide the users with a way to know that only certain individuals within *Wacky Widgets* were supposed to provide this type of validation or information. The problem seemed to be that few users were using the process and neither did they know what their responsibilities were.

Wacky Widgets then set up a series of security-related lectures during which information on hoaxes was presented using examples from the organization, and the processes and expectations were defined. All employees were required to attend at least one of the lectures. *Wacky Widgets* found that after the lecture series was completed, the number of hoax messages being spread dropped significantly and the number of hoaxes responsible for virus 'incidents' dropped below 5%. *Wacky Widgets* also experienced many actual virus incidents since the lectures where logon banners and warning notices were posted, helping to re-enforce the information that had already been provided.

Wacky Widgets has maintained a consistent drop in hoax incidents and has seen a marked increase in user reports of suspicious behaviour that led to the early detection of recent virus incidents such as Melissa and ExploreZip. While these results are not conclusive, *Wacky Widgets* was provided with informed users who report suspicious email messages and system behaviours – an early warning system. In addition, users do not continue to flood internal resources with hoax messages, thus helping to improve the overall security posture of the *Wacky Widget* network.

Summary

We recognize that the prevention of infection is the primary job of the anti-virus product(s) being used. However, it is in situations when the barriers have broken down that reliance on the users for early warning is essential. A clear and consistent method for getting current and useful information to the user is critical. Using a variety of the methods described in this article can provide a well-rounded or 'defence in depth' approach and it gives your early warning system the tools it needs to function effectively. It takes a team approach – software, hardware, and people – to provide maximum protection for your system.

CONFERENCE REPORT

Live from Las Vegas: Viruses and Disclosure

Dr Richard Ford

There is always something irresistible about *DEFCON*, an annual 'hacker' conference held in the surreal surroundings of Las Vegas, USA. Amidst the 24-hour a day lights, electric tinsel and sounds of money falling from the slots (or not, as the case may be), the conference attracts people from all over the globe. I say people, rather than 'hackers', because this year's offering seemed much more counter-culture than ever before.

Black is certainly the colour to be seen in, though blue, magenta and green hair also had their place. Numberless hoards of variously pierced youths stalked through the lobby of the *Alexis Part Hotel*, much to the surprise of other guests. *DEFCON* was back in town.

Despite the somewhat *clichéd* picture painted above, *DEFCON* is about a lot more than simple counter-culture. While the mix of attendees seems to be shifting away from hackers towards more of the general cyber culture scene, some of the 'best' (or shall we say most talented) hackers in the world regularly attend this conference. Only if you actually know their faces or if they wish to disclose their presence can you pull them out from the crowd. There amidst all this chaos, me. What on earth was I doing here? I had come to participate in a panel entitled 'Viruses On and Off the Internet' [*quite apart from being hired as* 'Virus Bulletin's *Man in Las Vegas'. Ed.*].

My Con-Fu is Tested

DEFCON started off in fine form this year, as I called the hotel to find that my room had been cancelled… which was okay, because someone else had booked me in with a different confirmation number. I decided to stay elsewhere, just to avoid any further mix-ups. I set my alarm for a quarter to nine, but need not have. I received an impromptu 8am wake-up call the next morning. Still, things could have been much worse... this was more of a gentle hello from friends than a hostile act. Tales of derring-do from previous years can be hair-curling.

The conference itself was its usual mix of joyous chaos which all seems to hang together in a wonderful sort of way. *Ad hoc* conversations struck up over a computer expand into some fairly sophisticated areas, and there is plenty to be learned. This year's conference was different to previous ones I had attended as it was multi-stream, with a 'Newbie' area, a main auditorium and an overflow area which oscillated from a third stream to a disco where DJs and bands could 'do their thing', whatever that may be.

Viruses and DEFCON

There is so much at *DEFCON* that it is impossible to do it justice unless you have been there, so in the remainder of this article, we will concentrate on those areas relevant to computer viruses and Malware. Three sessions come to mind; a panel discussion moderated by the lovely Sarah Gordon, an 'Introduction to Viruses' by the less-lovely Robert Lupo, and the official launch of Back Orifice 2000, by the media-grabbing Cult of the Dead Cow (cDc).

The most informative session of the three was the debate concerning full disclosure with respect to virus source code. Fuelled by the fracas surrounding the publication of the source to Melissa, Gordon's panellists represented several different positions: that of the VXer (played by The Attitude Adjuster), the List Manager (Elias Levy, aka Aleph1), a corporate security guy (myself, in a somewhat familiar role nowadays), the anti-virus industry (Torralv Dirro and Jon David), the latter convincingly representing the 'hang 'em and flog 'em' contingent.

At this point it only seems fair to note that the points made by each panellist differed, sometimes radically, from their own positions – everyone was there to present a particular point of view. This was duly explained to the audience, which packed the session, held in the largest of all the Conference rooms.

'Hi Boys, I'm Back'

The panel session consisted of three questions to the panel from Gordon, followed by open questions from the floor. The first of the questions concerned the public availability of viruses on ftp/WWW sites. Here, the panel was split along fairly predictable lines. The logic employed by 'the virus writer' was simply that this information should be available to all – that keeping information only for the cognoscenti was inherently wrong.

The world was too dependent upon technology – people needed to know how it could fail. The flipside of the argument, was presented by David and Dirro, namely that virus distribution needed to be legislated against as criminally irresponsible. 'These people must be stopped' commanded David, and to my total shock, Gordon got almost the entire audience to applaud his statement just by asking them to clap for him.

'Be nice', she said, 'he volunteered to come up here and say what some others in the anti-virus industry believe but couldn't be bothered to come and say, so give him some respect'. The audience applauded to which Gordon responded 'I bet you never in a million years thought you'd be sitting at DEFCON *applauding* some guy who says you all should be locked up'. There was laughter, and the *CNN*

cameras kept rolling. Yes, that's right: this panel generated so much interest that film crews from around the globe were there to discuss virus writers, and the distribution of viruses. This was not just a 'hacker conference'. This was a media event.

Interestingly, the corporate position was echoed by several attendees: corporate security officers I spoke with after the session felt that they have a need for access to computer viruses, and are therefore glad of the existence of Vx sites. They argued that viruses are 'out there' already, so anyone who wants them can get hold of them. Why should they be driven underground?

Indeed, this point was made during the question and answer session as well. Gordon interjected that the virus 'collections' which are available can be of very poor quality – they are not generally suitable for the testing of anti-virus software. Also, there may be liabilities involved should one accidentally unleash a nasty within the company and should that nasty find its way to the outside world.

As an aside, during the BlackHat Briefings (the serious side of *DEFCON*), the audience of several hundred security professionals was polled regarding their opinion on the public availability of viruses on WWW sites and the Internet. A surprising number of hands went up, more than went up when the same question was asked at *DEFCON*, of over a thousand people (many of whom represent themselves as the archetypal bad guy in black!).

When asked if viruses were cool, a surprisingly small number of hands went up amongst the *DEFCON* crowd. 'Viruses are cool only if its some kind of work toward evolutionary code'. 'Viruses are cool if they actually do something'. Gordon countered these arguments with the fact that the part of the virus that *replicates* is the most boring part, and that 'cool programs' can do things without having to spread themselves. 'How cool can it be if you can do it in 21 bytes' I quipped, driving the point home.

Interesting Perspective

The most interesting debate ensued when the topic of publication and availability of source and samples on 'white hat' lists was broached. This question was chosen as part of our research for the 1999 *Virus Bulletin Conference* presentation. Not only was there a panel discussion, but the audience was queried; the panellists were available after the session to discuss the issues further. Here, the issues are considerably more complex. The entire panel agreed that the publication of 'just another PS-MPC' virus was pointless, even for the virus writer! However, debate surrounded the legitimacy of publishing 'new and important' viruses.

The first response was from Levy, moderator of *Bugtraq*. He stated that administrators need to be aware of these types of exploits. Only by heightened awareness can pressure be put on the developers of applications and OSs in order to make viruses less viable.

Once again, the charge for the anti-virus community was led by Dirro. In fine form, if definitely outnumbered, he pointed out that there is usually little a company can do to take precautions against a new virus except by updating its anti-virus software. Similarly, he explained, the publication of the source can quickly lead to the development of new variants, making detection and protection more difficult.

Levy politely asked to address Dirro, responding with the point that we need solutions to the problem, not temporary fixes offered by the scanner makers. The audience applauded. Chalk one up for the 'other side'.

At this point hands in the audience were waving wildly, so Gordon opened the session up for questions. 'Why do people confuse prosecution with prevention' asked one neatly dressed delegate. 'People do what they can. We can't prevent, so we prosecute', countered David. 'I just want to say that we need viruses like this to show Microsoft they MUST secure up their products'.

'Do you think it's working?', I countered. 'You can't catch all the virus writers. It's wrong to catch only the ones you can catch because then you're just catching the stupid ones'. The logic of this last question eluded me. On that note, it was time to clear the stage for the offering by the cDc boys. We left the stage and were mobbed by press and audience alike. *DEFCON* was in full swing.

Closing Thoughts

As one journalist wrote just days before the conference, 'the genie is out of the bottle'. Echoed by many IT professionals, sentiments ran high that viruses are out there, and they will not go away simply because we (the white hats) choose not to publish them. The general level of expertise on viruses in the *public* areas of *DEFCON* is abysmal. That is a good thing.

However, as I sat back on the plane and passed up a Gin and Tonic I was offered, I reflected on my few days at *DEFCON*. At that point I was reminded that things are not always as they seem. I had seen piercings where I did not even know there were body parts; trust me on this. I had seen 'Feds' running around like ants on amphetamines. I had taken countless taxis back and forth from my hotel to the conference hotel (hint: *never* stay at the same hotel where the actual *DEFCON* is taking place. Sleep will not be allowed). These are the 'external signs' of *DEFCON*.

More importantly, what I came away with was this: while I had seen some people 'learning' about viruses much the same way they did back in the early 1990s (and at a very low level of expertise), I had discussed very real security issues at a level I would be hard pressed to find anywhere in the mainstream anti-virus world. Finally, I was reminded of one eerie fact – some people, both black hats and white hats, have a real, thorough, deep and abiding knowledge of what can be done with computer viruses. I am afraid it is only a matter of time.

PRODUCT REVIEW 1

Tripwire v2.1 for Windows NT

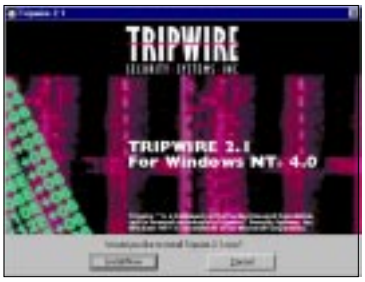
The first of the product reviews this month takes a look at a product that has long been familiar to Unix users, the integrity checker *Tripwire*. Originally developed as a freeware application by researchers at Purdue University solely for Unix platforms, the product is now commercialised under the wing of *Tripwire Security Systems Inc*. In this review we assess the latest version of this product, the first to support the *Windows NT* platform.

The Package

The package submitted for review consisted of a small envelope bearing the *Tripwire* logo and licensing details. Within the envelope was the documentation (presented in a neat, spiral-bound A5 booklet), a quick reference card and the CD itself.

The product as a whole is designed for use by Systems Administrators, or at the very least experienced users. It is not complex to configure and use, but neither is it simple. The documentation is well set out, and split up into logical sections: installation, configuration, and use. Finally, the cryptographic methods *Tripwire* supports are described.

Installation



Installation is achieved using a standard *InstallShield* interface. The destination directory can be changed, which is probably advisable given the length of the default path. Since the program is designed to be run from the command line, such a long path (C:\PROGRAM FILES\TRIPWIRE SECURITY SYSTEMS\TRIPWIRE 2.1\) leads to unpleasant line wrapping.

Various configuration options are offered during installation, although since this information is simply stored in a plain text ASCII file, such options can be easily altered later on. Of importance to administrators faced with multiple *Tripwire* installations, there is an option to perform a silent installation, where the configuration settings are read in from a predefined text file. For installation across a network, this text file can be read from a network drive.

The final stage in the installation process consists of entering pass phrases which are needed to generate keys for cryptographically signing *Tripwire* files. Two pass phrases are needed – a 'site' one for signing files used by multiple machines, and a 'local' one for signing files that are machine specific. For security reasons, the documentation recommends that users adjust the permission settings for the *Tripwire* directories following installation.

Principles of Operation

Quite simple really. As with other file integrity checkers, the mode of operation is simply (i) record information from a clean, uncompromised system, (ii) regularly check against this information database, (iii) report discrepancies, and (iv) update information to reflect legitimate changes or (v) report change as an undesired attack.

So does *Tripwire* offer any advantages over the integrity checkers supplied in other all-in-one AV product bundles? At first glance, user friendliness does not appear to have been a high priority. Mouse jockeys will baulk at the very idea of a *Windows* product being operated solely from the command line. Forget the cosmetic niceties, *Tripwire* is a behind the scenes workhorse. Once the necessary principles behind its operation have been grasped, the sheer flexibility this product offers soon becomes apparent.

In addition to monitoring the integrity of the filesystem, *Tripwire* can also monitor the *Windows* Registry (both keys and values).

Using Tripwire

All of *Tripwire's* operations are controlled from the command line. Following installation it can be 'initiated' from the Start Menu – the shortcut merely opens a command prompt window (CMD32.EXE) with the startup location set to the *Tripwire* program file destination directory.

Administration of *Tripwire* is achieved by the editing of two files, the configuration and policy files. The former stores information entered during the installation process, including details such as the location and name of policy file(s), database file(s), cryptographic keys, and text editor. The latter file provides the main control facility for *Tripwire*. Quite simply, it contains the rules by which *Tripwire* looks at the system.

Both can be created or edited as plain text files, which must be encoded and signed prior to use by *Tripwire*. For this the TWADMIN program is used. To prevent unauthorized access to the configuration of *Tripwire*, once the necessary manipulations to these plain text files has been achieved, their deletion is recommended.

As mentioned above, the policy file is the control centre of operations. To aid the administrator in setting *Tripwire* up for the first time, a generic policy file for *Windows NT v4.0* is installed as part of the package. In its simplest form, the

policy file consists merely of a set of instructions telling *Tripwire* what to monitor (objects), and what properties of each object to monitor (property mask). Depending upon whether the object pertains to the NTFS or the Registry, a whole variety of object properties can be monitored with *Tripwire* – ranging from file size, file attributes and checksums of the NTFS data streams, to the in-built *Windows NT* security attributes.

With this degree of flexibility in the rulemaking, it is easy to imagine the policy file becoming extremely complex and cumbersome. Such a scenario is avoided however by the use of a set of predefined variables for both the NTFS and Registry objects. For filesystem objects, these include ReadOnly (for widely accessible files), Dynamic (for files whose content is expected to change), IgnoreAll (checks only a file's presence) and IgnoreNone (checks all file properties). To aid setting up the Registry section of the policy file, predefined variables also exist for each of the key names (e.g. HKCR for HKEY_CLASSES_ROOT).

Thus, a particular rule in the policy file might be:

```
C:\DATA   ->    $(Dynamic)-&owner
```

which instructs *Tripwire* to monitor the C:\DATA directory according to the property mask defined for the Dynamic variable, excluding the NTFS Owner SID. Additional variables (both local and global) can be defined within the policy file as needs must. Due to the number of object properties that can be monitored, during testing it was found that the pre-defined variables were extremely useful as templates for personalised variables.

A nice feature of the policy file is the way in which rules can be divided up into specific sections. This enables different actions to be taken for violation of rules within different sections. For example, certain system areas could be defined in a 'Critical objects' section, where if violation occurs an email is sent directly to the administrator. The policy files also support conditional interpretation of the rules depending upon the host computer. This facilitates the use of a single centralized policy file for computers on a network, rather than separate files for each.

Brief tests of the throughput of *Tripwire* were then performed. For this a database of information concerning 4,500 files (377 MB) was generated and integrity checks performed using a variety of policies. Processing rates of between 0.1 and 1.8 MB/sec were observed during integrity checks, depending upon the level of information monitored. Though while-you-wait checks of small systems may be acceptable, scheduling is certainly needed for the checking of large systems. Scheduled checks can be enabled using the *Windows NT* 'AT' command.

Changing the Tripwire policy

OK, so *Tripwire* has been installed and running happily for a few months. However, what if the administrator decides to step up security and monitor certain files more closely? For this, changes to the policy file need to be made. To avoid incorporating any system violations that may have been made since the last integrity check, *Tripwire* uses a special mode for policy file updating.

This mode enables changes to the policy file to be made without having to re-initialize the database completely. Upon initiating update mode, the system is checked according to the new policy file, and any discrepancies between the object information obtained from rules present in both the old and new profiles are reported.

Such discrepancies should not be incorporated into the new database without first being checked by the administrator. It is peculiar therefore that the default action during policy file updates is to use the 'low' security mode, in which the database files are updated silently. To prevent such changes being incorporated into the new database, it is necessary to include the 'high' security command line switch manually.

Cryptographic Security

Four cryptographic signature routines are shipped with *Tripwire* – CRC (32-bit), MD5 (128-bit), SHA (160-bit) and HAVAL (128-bit). Being the fastest to use, CRC signatures are also the least secure, and there are programs freely available which allow modification of files whilst maintaining the original 32-bit CRC signature. Use of one of the three other more secure routines is recommended.

Conclusions

One thing is clear with *Tripwire* – it is aimed at the administrator end of the market, both in terms of use and price. It finds its niche where the implementation of large scale security plans are needed, not for the home user who wants a 'point and click' approach to integrity checking.

Though requiring a sound knowledge of the manual, the nature of *Tripwire's* configuration is its best asset, giving it huge flexibility. Inevitably however, with flexibility comes complexity. It is unsurprising, therefore, that configuration is also the product's weakest point. Future versions may well resolve this by enabling *Windows*-based configuration through a user friendly GUI. Though welcome to some, let's hope that this will not be at the expense of the flexibility that *Tripwire* currently offers.

Technical Details

Product: *Tripwire 2.1 for Windows NT.*

Developer: *Tripwire Security Systems Inc*, 1631 NW Thurman, Portland, OR 97205, USA; Tel +1 503 223 0280, Fax +1 503 223 0182
email tripwire@tripwiresecurity.com,
WWW http://www.tripwiresecurity.com/.

Availability: *Windows NT 4.0, Internet Explorer 4.0* or later is required to view on-screen help files.

Price: $495 including 1 year's support & maintenance contract.

Hardware Used: 166MHz Pentium-MMX with 64MB of RAM, 4 GB hard disk, CD-ROM drive and 3.5-inch floppy, running *Windows NT 4.0*, SP5.

PRODUCT REVIEW 2

Norman Virus Control v4.70 for Windows NT

Martyn Perry

Since the withdrawal of support for their *ThunderBYTE* product (March 1999), this is the sole player in the anti-virus arena from *Norman Data Defense Systems*. With a strong pedigree in *VB* Comparative Reviews over the past twelve months or more how does *Norman Virus Control* (*NVC*) stand up to closer inspection?

Presentation and Installation

The software is supplied on CD. The installation auto-loads and presents a screen in both English and Norwegian. After selecting English installation, the program prompts for a CD key. This key is a label found on the rear of the CD case. It is a thirteen character code split into three sections. and it is case sensitive. A small problem here is that the key characters in each section are printed in different font sizes. This initially led to the assumption that all characters might be in upper case. This was not true since the first section needed to be in lower case.

Having entered the key code correctly the next screen offers a choice of installations: Typical – common options and recommended for most users, or Custom – providing a choice of installation options and recommended for advanced users. There was an additional choice to update (greyed-out) as well as the opportunity to create a diskette for scanning from a clean boot.

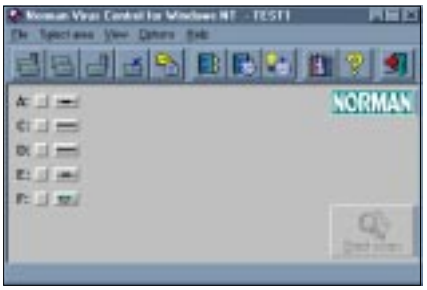
The next screen offered the destination directory for the application C: \NORMAN with icons being added to the program folder. The subsequent screen displays a summary of the selections made, requesting their confirmation.

There is a further option to select an update diskette (for updated signatures), if shipped with the CD. Finally, the installation prompts for the Norman Internet Update Authentication key. If there is no network connection available, one need only press cancel to continue.

We now return to the front screen in English/Norwegian. The options for documentation are only provided in Norwegian. In my opinion, it would have been better, having established that it was an English installation, if the final screen was also set up in the appropriate language.

General Configuration Options

The main screen provides access to a number of facilities – Files, Select Area, View and Options. Under Files, it is possible to view the contents, in hex, for a specific file, Master Boot sector or System Boot sector. This viewing facility is a useful feature as it allows the administrator to investigate suspicious activity further.



With Select Area, drives can be selected either locally or on the network. In addition, a separate selection can be made to select a specific directory and its subdirectories. It is not possible to make multiple directory selections except by moving back up a level and having all directories of the same level scanned.

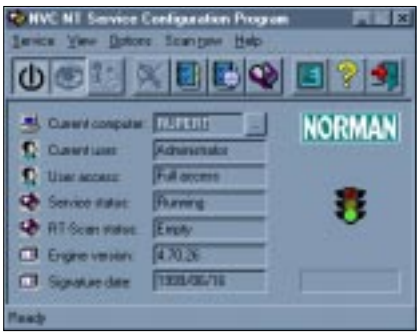
View is the facility to read the log file as well as to link to the Virus Library (description of the infection mechanisms) and the Book on Viruses (giving histories and infection mechanisms). Trying to select this from the menu gives an error looking for NVCBOOK.HLP file. The book was in fact found as BOOKON.PDF. Finally, the Options Menu splits into Scheduled scan and setting up options for the on-demand scan.

Managing infected files allows the possibility to repair a file. When this is not feasible, the following choices are available: no action, delete infected files or move infected files to a quarantine directory (C:\NORMAN\INFECTED). However, during testing for this review, the read-only samples would not delete.

Therefore, the move option was used to transfer infected files to the quarantine directory. *NVC* does not delete files in the following situations: a) if the file is on write-protected media, b) if the file resides on a network drive and is write-protected, and c) if the file is in use.

Options to give more specific virus names, scan all files, ignore system areas, look for EXE headers, delay between files, beep on infection, and minimise while scanning can also be selected. To facilitate the configuration of scans, the various selections can be grouped together and saved as a 'Style'. Thus a number of styles can be pre-configured for specific tasks, and accessed with a single click. This is particularly useful when running styles under the scheduler. It enables different parts of a machine or network to be scanned using different schedule periods.

For configuration of the real-time scanner, the *Norman Virus Control NT* Service Configuration Program is used. This is loaded by running the NCFGW.EXE file (in the default destination directory). Configuration of the real-



time scanner can be included at the command line during the loading of this service, although it can also be configured, stopped and started from within the service.

Scanning Options

This can be called either from the control program or, when the NT Service is running, from the Configuration program. The default file extension list is particularly large, including a total of 52 file extensions!

The scanning options include: Don't stop on virus detection, Ignore locked files, and Look for OLE2 headers. The latter option can be chosen to overcome the problem of document files being stored with non-standard file extensions or no extension at all. The downside of this is that every file will need to be checked. Further options include: Scan multiple diskettes using the same settings, Scan archive files (supported formats are ARC, ARJ, LZR, PAK, ZIP, ZOO) and finally, Scan compressed program files.

With regard to reporting, the report can go to a printer or to a file (default C:\NORMAN\NORMAN.RPT). The choice can be made to log just infected files or to include scanned directories or all files.

When managing infections, the scanner will attempt to repair if possible. If it determines that it is not possible, then the choices are: No action, Delete infected files or Move infected files to a quarantine directory (default C:\NORMAN\INFECTED). Additional options include: Provide more specific virus names, Scan all files, Ignore System areas, Look for EXE headers, Delay between files, Beep upon infection and Minimise display while scanning.

During on-access scanning, the scanner allows separate read and write access checking. Floppy diskettes are also checked when first installed locally. The handling of virus infections is similar to that for on-demand scanning.

Scheduled scans can be configured and activated separately. The scheduled task can be run Hourly, Daily, Weekly or Monthly. A Style can be chosen with all the pre-defined settings required. Scheduled scans can be configured similarly to on-demand scans, with the following options: do not stop when a virus is detected, ignore locked files, look for OLE2 headers, scan multiple diskettes, scan archive files and scan compressed program files.

An additional interface to NVC is available through *Explorer* by right-clicking on any file, directories or drives. The interface is simpler than that of the main program, providing a convenient means of scanning specific files. If infected files are found, options to delete, move or attempt disinfection are then presented. The contents of archive files are automatically scanned when *NVC* is initiated through *Explorer*.

Administration

For *Win 9x* workstations there is an additional program called *Cat's Claw*. This can scan for file, boot sector and macro viruses. For those who use macros regularly, there is an option to certify legitimate macros. For this, checksums of the approved macros are generated and stored.

An extra facility is the option to create a company-specific help file called USERDEF.HLP. This can be employed to inform the user of company-specific procedures. To distribute the product, there is a separate program, N_DIST. This can be used in conjunction with script files to control the distributed installation of the software to workstations and remote servers.

Updates

Updates to the virus definition files (NVCBIN.DEF and NVCMACRO.DEF) and the scanner engine DLL can be obtained using the *Norman* Internet update facililty. The files can be downloaded either individually, or within a self-extracting, self-installing executable.

Scanning Overhead

To measure the extra work performed in detecting a virus, diskettes comprising 26 EXE and 17 COM files were scanned. An overhead of 43.2% was measured when scanning the files infected with the Natas.4744 virus. Next, the *VB* Clean set was scanned to measure scanning speed and check for false positives. It took 36 minutes and 30 seconds to scan the 5,500 file collection (520 MB), with zero false positives alerted.

Detection Rates

The scanner was checked using the official *Virus Bulletin* test-sets – ItW, Standard, Polymorphic, Macro and Boot Sector. See the summary below for counts. The tests were conducted using the default scanner file extensions supplied. The scanner was configured to move infected files, and the residual file count was then used to determine the detection rate.

The results were impressive, starting with a 100% success rate against boot sector viruses. Complete detection against the ItW file set was denied only by an extensionless *Excel* file infected with O97M/Tristate.C. The corresponding samples of the A, B and D variants of this virus were also missed in the Macro test-set. Setting *NVC* to scan All Files

(or those containing OLE2 headers) resulted in detection of these extensionless samples. Elsewhere in the test-sets misses were few. One of the Win95/Navrhar infected VxD samples was missed in the Standard test-set, and samples infected with A97M/Accessiv (A and B variants), PP97M/Vic.A, W97M/Triple.B, and W97M/IIS.H were missed from the Macro test-set. As with many of the other products tested recently, *NVC* is yet to include detection of a newcomer to the Polymorphic test-set, ACG.A.

Real-time Scanning Overhead

To determine the impact of the scanner on the workstation when it is running, the following test was executed. The basis of the test was to time the following activity.

200 files of 23 MB bytes (a mixture of DOC, DOT ,XLS, XLT, XLA, EXE and COM files to reflect typical file types being moved) were copied from one folder to another using XCOPY. The folders used for the source and target were excluded from the virus scan so as to avoid the risk of a file being scanned while waiting to be copied.

The default setting of Maximum Boost for Foreground Application was used for consistency in all cases. Due to the different processes which occur within the server, the time tests were run ten times for each setting and an average taken.

- Program not loaded: establishes the baseline time for copying the files on the server.
- Program unloaded: run after the server tests to check how well the server is returned to its former state.
- Program installed, scanning files being written: tests the impact of the application scanning files being written on the server.
- Program installed, reading files: tests the impact of the application scanning files when they are being read.
- Program installed, scanning both writing and reading files: tests the impact of the application scanning files when being written and read on the server.
- Program installed, scanning both writing and reading files while running manual scan: tests the impact of the application scanning files being written and read on the server with an application accessing files.

Summary

NVC must take the prize for the sheer number of file types being scanned. It serves as timely reminder just how many different file types can be attacked by viruses. However, this was not sufficient to detect all of the ItW virus samples, thanks to the extensionless samples of O97M/Trisate.C. This, and other similar files are detected if the scanning options are altered to scan files containing an OLE2 header irrespective of extension. Whether this option should be included in the default 'out-of-the-box' configuration is a matter of current interest. As has been expressed throughout recent *VB* reviews, the days of file extension lists are certainly numbered. Despite the increased overheads of scanning all files by content, adequate user protection against viruses such as the O97M/Tristate variants demands such a scanning method.

The retention of command line options is a useful and welcome feature of *NVC* allowing the service to be started with options rather than having to rely solely upon set up from the configuration program.

Another nice feature is that of having a separate facility for scanning floppy diskettes. This may at first seem rather trivial, but for anyone who needs to scan large numbers of diskettes, the ability to select the next diskette with a single command rather than having to step through a complete drive selection sequence each time will be very welcome.

So, will *ThunderBYTE* users be disappointed with their conversion to *NVC*? In short, no, probably not. The high detection rates that have become something of a trademark for *NVC* continue, and its scanning speed, though not as fast as *ThunderBYTE*, is perfectly adequate.

NVC for Windows NT v4.70

Detection Results

Test-set[1]	Samples Detected	Score
In the Wild Boot	45/45	100.0%
In the Wild File	548/549	99.8%
Standard	1282/1283	99.9%
Polymorphic	14444/14618	98.8%
Macro	2887/2898	99.6%

Overhead of On-access Scanning:

The tests show the time (in seconds) taken to copy 200 files (23 MB). Each test was repeated ten times, and an average taken.

	Time	Overhead
Not loaded	23.9	–
Unloaded	24.0	0.2%
Loaded, writing	37.2	55.0%
Loaded, reading	47.6	98.8%
Loaded, writing, reading	63.3	164.0%
—— + —— + manual scan	157.5	557.0%

Technical Details

Product: *Norman Virus Control v4.70.*

Developer: Norman Data Defense Systems, Postboks 43, N-1234 Lysaker, Norway. Tel +47 671 09700, Fax +47 675 89940, email sales@norman.no, WWW http://www.norman.com/.

Price: contact *Norman.*

Hardware Used: Workstation: *Compaq* Prolinea 590, 80 MB of RAM, 2 GB hard disk, running *NT Server v4.0 (SP5).*

[1]Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NetWare/199907/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:

Virus Bulletin, 18 Commerce Way, Woburn, MA 01801, USA

Tel (781) 9377768, Fax (781) 9320251

END NOTES AND NEWS

According to London-based data security vendor *Reflex Magnetics Ltd* the ExploreZip worm was the fault of commercial anti-virus organizations. *Reflex* accused the industry of encouraging user complacency through spreading the 'myth' of total virus protection by updated scanners alone. *Reflex* claims that only generic protection like that afforded by its *Reflex Disknet Data Security for Windows NT* (see June issue's End Notes and News section) defends against known and unknown malware. At the forefront of *Reflex's* argument is the exploitation of human psychology and its vulnerabilities, as featured in the Corporate Tutorial on p. 15. More information about *Reflex* and its products can be found at http://www.reflex-magnetics.co.uk/.

A Practical Anti-Virus Workshop will be run by *Sophos* on 15 and 16 September 1999 at the organization's training suite in Abingdon, UK. For more details or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, fax +44 1235 559935, or visit the company web site at http://www.sophos.com/.

The *Computer Security Institute's* 26th annual conference and exhibition is to be held from 15–17 November 1999 at the Marriott Wardman Park Hotel in Washington DC. For more information on the 85 featured presentations or pre- and post-conference seminars, contact *CSI*: Tel +1 415 9052626 or visit http://www.gocsi.com/.

***Network Associates Inc* has included a new feature in its enhanced version of *VirusScan*.** In what it claims to be an 'industry first', the corporation claims that its 'email x-ray' makes *VirusScan* the first desktop anti-virus product to scan email attachments at mailbox level, before they are opened by the user. This offers protection against viruses like Melissa and Trojans like ExploreZip. The new version is also said to provide the only anti-virus protection for the latest Java and ActiveX threats as well as updated *Office 2000* protection. For details, contact; Tel +44 1753 827500 or visit http://www.nai.com/.

CompSec'99, the 16th World Conference on Computer Security, Audit and Control will take place from 3–5 November 1999 at the QE2 Centre, Westminster, London, UK. A Directors' Briefing will be held on 4 November. Conference topics include malicious software, firewalls, network security and Year 2000 contingency planning. For more details contact Tracy Stokes at *Elsevier*; Tel +44 1865 843297, fax +44 1865 843958, or email t.stokes@elsevier.co.uk.

***Norton System Works* from *Symantec* has been chosen by *Microsoft* to partner its *Office 2000* software application suite across the UK.** In an introductory offer *Norton System Works* is available at 50% off its recommended retail price when purchased in the same transaction as *Office 2000*.

In an unrelated announcement, the latest version of *Symantec's pcAnywhere v9.0* allows remote access to internal company networks through Internet and intranet whilst protected by *Norton Anti-Virus*, low-level encryption and *VPN* (*Virtual Private Network*) technology. This product is available now for £144, or as an upgrade for £57. For more details contact Charlotte West at *Harvard PR*; Tel +44 181 7590005 or email charlotte@harvard.co.uk.

***Data Fellows* plans to extend its existing line of *Linux*-compatible security products.** *F-Secure Anti-Virus for Linux* is now shipping and later this year *F-Secure VPN+* (secure networking software) will be available. Contact Jason Holloway; Tel +44 1223 257747 or email Jason.Holloway@DataFellows.com.

The seemingly exponential growth in BackDoor hacks continues with **the Trojan Subseven.backdoor.C**, discovered in mid-June and now confirmed as in-the-wild, distributed under various aliases via newsgroups and email.

In Brussels, Belgium, from 4–7 March 2000, **the ninth annual *EICAR* conference**, also known as the first European Anti-Malware Conference, takes place. For more information, to place a booking or to order a timetable visit the web site at http://www.eicar.dk/.