DECEMBER 2004

virus BULLETIN

The International Publication
on Computer Virus Prevention,
Recognition and Removal

CONTENTS

IN THIS ISSUE

DEAR SANTA ...

What do you get a spammer for Christmas? Graham Cluley looks back at a year that has seen the arrests of more Internet criminals than ever before – and provides a couple of suggestions for Santa.

page 2

SNAIL'S PACE

Was the 3,000 rouble fine given to 29A virus writer Eugene Suchkov (aka 'Whale') last month a fitting penalty for creating viruses? *VB* doubts it, but make up your own mind after you've read Peter Ferrie's analysis of one of Suchkov's creations: the Microsoft *.NET*-infecting Gastropod.

page 4

CALLING ALL SPEAKERS

Virus Bulletin is seeking submissions from those wishing to present at VB2005 in Dublin – so set aside some time between the season's festive parties to get writing!

page 19

vbSpam supplement

This month: anti-spam news & events; spam on mobile devices; ASRG summary.

ISSN 0956-9979

'2004 saw the image of the virus writer change once and for all, and take a more sinister shape.'

Graham Cluley, Sophos, UK

SPAMMER BABY, SLIP A ROLEX UNDER THE TREE, FOR ME ...

What do you give a spammer for Christmas? The question struck me the other day and set my mind wandering. After all, they have all the septic tanks, fake degrees, imitation luxury watches and herbal, er, pick-me-ups they could ever need – even medication for that painful Boxing Day hangover.

For those like email con man Nick Marinellis, who defrauded millions of dollars from Internet users who thought they'd won the lottery, a file hidden inside a Christmas cake may be the most they can hope for – Marinellis has been sentenced to at least four years behind bars.

In fact, 2004 has been remarkable for having seen the arrests of more virus writers, spammers and other Internet villains than ever before. There may be more crime on the net than at any point in its history, but the authorities are at least having some success.

Belgian teenager Kim Vanvaeck courted the media for years under her nom de plume Gigabyte of the Metaphase virus-writing gang. You can still find her guestbook online where lonely teenage boys post love letters, imagining her to be just as the newspapers, radio and television stations portrayed her – a Lara Croft-style malware-making cyberbabe. Whether her court case results in conviction remains to be seen – the authorities are challenged to discover if her seldom seen in-the-wild viruses caused damage or whether she incited others to cause mischief.

Sven Jaschan, the German lad who admitted to having written the Sasser and Netsky worms, will be spending Christmas wondering about his fate in the new year too. The recent announcement that he has started working for a firewall company may play well to those who wish to hear he is rehabilitating, but it leaves a nasty taste in the mouth for those who believe the firm may have employed him purely as a publicity stunt.

Jaschan was extraordinary. Not in the way we would wish anyone to be proud of, but astonishing in terms of the impact his worms had during 2004. Over 70 per cent of the virus incidents sighted in the first half of 2004 were written by the German teenager. His worms continue to spread and infect innocent users to this day, and have had a bigger impact on computer users than other names from the chamber of horrors: ILOVEYOU, Anna Kournikova, Sircam, Nimda and Blaster.

But 2004 saw the image of the virus writer change once and for all, and take a more sinister shape. *Sophos*'s global network of laboratories has seen a marked shift in the motives behind the viruses written in the last 12 months.

In February 2004, the MyDoom worm launched a denial of service attack against SCO's website, forcing the company to switch to another domain name for several weeks and to offer a $250,000 reward for information leading to the conviction of the worm's author. Other attacks were equally sinister as viruses (written, presumably, with active encouragement from the spamming community) targeted the websites of anti-spam organisations, trying to make them less effective.

Every month it becomes more obvious that a prime motivation for writing viruses today is to steal information and resources from infected computers, and create networks of zombie computers that can be used to launch new virus attacks, a spam campaign or denial of service attacks. Never have those behind virus attacks been better organised or more serious in their criminal intentions.

If you want to give a virus writer or a spammer something for Christmas, give them a hard time. Make sure your computer, and those of your friends and family, are properly protected with automatically updating anti-virus software, properly configured firewalls and the latest *Microsoft* security patches. Do something good this holiday season by ensuring that the spammers and virus writers find it more difficult to exploit innocent computers in 2005.

NEWS

SEASON'S GREETINGS

Cheers! Season's greetings from the Virus Bulletin team (clockwise from top left): Helen, Matt, Bernadette and Tom.

The *VB* team wishes all *Virus Bulletin* readers a very happy Christmas and a prosperous new year. Continuing the custom of making charitable donations in lieu of sending Christmas cards, *VB*'s donations for Christmas 2004 will be made to The International Committee of the Red Cross (http://www.icrc.org/) and the WWF (http://www.wwf.org/).

ACADEMIC RESEARCH JOURNAL

October saw the announcement and first call for papers of the *European Research Journal in Computer Virology* – a twice-yearly independent scientific journal dedicated to computer virus and anti-virus technologies. The aim of the journal is to promote constructive research in computer virology. All papers submitted will undergo peer review and those accepted will be published within a year of submission. For more information contact the editor-in-chief, Eric Filiol (eric.filiol@inria.fr).

FBI'S VIRUS BLUNDER

It has come to light that a virus infection nearly blew the cover on a secret FBI fraud investigation two years ago.

The FBI had been investigating Dr Rafil Dhafir, who was suspected of breaking US sanctions against Iraq. Shortly after the FBI began monitoring Dr Dhafir's email, however, the Bureau's computer system became infected with Klez.h, causing an email to be sent from the FBI to their suspect. Understandably concerned that Dhafir might become suspicious, the FBI investigators sent him a second email, creating an elaborate ruse to try to convince him that they were investigating the virus – even encouraging him to call the Bureau if he had any problems. Luckily for the investigators Dhafir seemed to fall for the trick and he was arrested ten months later. One hopes that rather more robust IT security mechanisms are now in place at the Bureau.

Prevalence Table – October 2004

Virus	Type	Incidents	Reports
Win32/Netsky	File	148,405	74.35%
Win32/Bagle	File	42,291	21.19%
Win32/Bagz	File	1,339	0.67%
Win32/Zafi	File	1,138	0.57%
Win32/Mabutu	File	963	0.48%
Win32/Funlove	File	930	0.47%
Win32/Dumaru	File	842	0.42%
Win32/Mydoom	File	686	0.34%
Win32/Klez	File	495	0.25%
Win32/Lovgate	File	364	0.18%
Win32/Valla	File	312	0.16%
Win32/Bugbear	File	208	0.10%
Win32/Mimail	File	193	0.10%
Win32/Swen	File	177	0.09%
Win32/MyWife	File	161	0.08%
Win32/Elkern	File	104	0.05%
Win32/Parite	File	102	0.05%
Redlof	Script	95	0.05%
Win32/Fizzer	File	79	0.04%
Win32/Mota	File	78	0.04%
Win95/Spaces	File	71	0.04%
Win32/Kriz	File	67	0.03%
Win32/Yaha	File	67	0.03%
Win32/Hybris	File	59	0.03%
Win32/Magistr	File	34	0.02%
Win95/Tenrobot	File	33	0.02%
Win32/BadTrans	File	26	0.01%
Win32/Plexus	File	25	0.01%
Win32/Evaman	File	23	0.01%
Win32/Nachi	File	17	0.01%
Win32/Sobig	File	17	0.01%
Win32/Nimda	File	14	0.01%
Others[1]		48	0.10%
Total		199,611	100%

[1]The Prevalence Table includes a total of 196 reports across 48 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

LOOK AT THAT ESCARGOT

Peter Ferrie
Symantec Security Response, USA

In 2003 I wrote 'A recompiling virus like W95/Anxiety, but without needing the source code, combined with an inserting virus like W95/ZMist, but without rebuilding the file manually ... The beast is unleashed' (see *VB*, April 2003, p.5). Now, hot on the heels of MSIL/Impanate (see *VB*, November 2004, p.6), which was the first inserting virus for the *.NET* platform, comes MSIL/Gastropod, which brings the full set of techniques one step closer.

DO NOT PASS 'GO'

Gastropod begins by calling a method named 'Go', which calls another method, which calls two other methods, which … The virus appears to have been written by someone who has been introduced to classes in C#, and has embraced them with such enthusiasm that every few lines of code are candidates for a method. It is rather like the saying, 'when all you have is a hammer, everything looks like a nail'.

This virus searches the current directory, and all parent directories, for files whose suffix is 'exe'. For each file it finds, Gastropod checks if a file exists with the same name, but preceded by an '_' (underscore). If such a file does not exist, then the virus attempts to rename the original file to this name. The renamed file will be used as a backup in case the infection fails for some reason, however the virus will delete the file if the infection is successful.

There are several bugs in the handling of files, often resulting in the virus exiting with errors, claiming that the '_[filename]' file cannot be found, or that a file cannot be created because it exists already.

DUE PROCESS

Gastropod was named 'Snail' by its author, because of the slow speed with which it executes. Of course, this is not helped by the poor coding style of the virus author, but an additional slowing factor is the way in which the virus detects if a file is already infected. The infection marker can only be found by opening each file and examining each method corresponding to each type that is stored within each module. The file is considered to be infected if any type and length of a method matches that of the virus itself.

If the file is not already infected, the virus will construct a list of types, marking those that can be renamed safely. Fortunately, the virus will rename only its own classes and methods. If the current method being examined is the entrypoint method, then the virus will place its classes at the top of the list, before adding those of the host.

GARBAGE MAN

Once the list is complete, the virus disassembles both itself and the host, using the well-known ILReader.dll utility, which is carried by the virus. The virus appends this DLL to the host during replication, which accounts for nearly 60kb of the total infection size.

The disassembly is used by the virus to insert garbage instructions throughout the code of itself and the host. The garbage instructions are either a NOP (with 20 per cent chance), or a LDLOC and POP combination (with 10 per cent chance, and only if the current method contains local variables).

If the LDLOC instruction is used, then the virus will choose the index randomly from the local variables of the current method. The virus also removes these instructions if they are found during the disassembly process.

It is unclear why the virus detects and removes those instructions, but one possible explanation is that this is an artifact from the development of the metamorphic engine, prior to the addition of the virus body – since legitimate programs usually do not contain such instruction sequences. If that is the case, that means that Gastropod shares some similarities with W95/ZMist (see *VB*, March 2001, p.6), which contained a routine that inserted redundant JMP instructions into the host, without adding the virus body.

UNEXCEPTIONAL

The virus is aware of the exception handling mechanisms in MSIL, and handles them mostly correctly – however not everything goes according to plan.

There are a number of bugs in the parsing, resulting in always duplicating 'endfinally' instructions, and producing unreachable (and, in some cases, incorrect) 'leave' instructions. These are things that would have been obvious to a tester, since they are visible in the sample that the virus author released.

This also makes it extremely difficult to restore a file to its state prior to infection. To complicate matters further, the virus extracts the managed resources from the host and stores them externally, in a file whose name is the same as the host, with '.resources' appended. Finally, if the host contained unmanaged resources, the virus will move the resources to the end of the file, and place them in a newly created section that is always named '.rsrc'.

If the entrypoint method ends with a 'ret' instruction, then the virus inserts code to abort the thread. This prevents the process from hanging around in memory, instead of terminating, and waiting for the virus code to complete, which would be suspicious to some users. A similar problem exists for Win32 viruses that hook the ExitProcess() API of a host.

After disassembling the virus code and the host code, the virus renames its classes and methods. The renaming is done using 6–15 letters, with a 12.5 per cent chance of an upper case letter. The virus avoids renaming the 'Go' method though, since the way in which the virus inserts the code into the host's Main method results in a hard-coded method name.

CONCLUSION

The acceptance of the *.NET* platform has been slow so far, but it is increasing, and the complexity of viruses for that platform has already progressed much further, in a much shorter time, than was the case for earlier *Windows* versions.

The full potential of the platform has not yet been realised by virus writers, but it is clear that they are working hard to reach that goal. We have received the warning, and our anti-virus engines must be prepared. For some companies, the *.NET* platform might be the next OLE2.

MSIL/Gastropod

Size:	77828 bytes.
Type:	Direct action, parasitic inserter.
Infects:	Microsoft *.NET* files.
Payload:	None.
Removal:	Delete infected files and restore them from backup.

FEATURE 1

ARE METAMORPHIC VIRUSES REALLY INVINCIBLE? PART 1

Arun Lakhotia, Aditya Kapoor and Eric Uday Kumar
University of Louisiana at Lafayette, USA

In the game of hide and seek, where a virus tries to hide and AV scanners try to seek, the winner is the one that can take advantage of the other's weak spot. So far, the viruses have enjoyed the upper hand since they have been able to exploit the limitations of AV technologies.

Metamorphic viruses are particularly insidious in taking such advantage. A metamorphic virus thwarts detection by signature-based (static) AV technologies by morphing its code as it propagates. The virus can also thwart detection by emulation-based (dynamic) technologies. To do so it needs to detect whether it is running in an emulator and change its behaviour.

So, are metamorphic viruses invincible?

INTRODUCTION

When you consider all the tricks that a virus writer can use to break AV scanners, metamorphic viruses, such as Win32/Evol, Metaphor (aka W32/Simile, see *VB*, May 2002, p.4) and W95/Zmist (see *VB*, March 2001 p.6), appear invincible. These viruses transform their code as they propagate, thus evading detection by analysers that rely on static information extracted from previously observed virus code. The viruses also use code obfuscation techniques to hinder deeper static analysis. Such viruses can also beat dynamic analysers by altering their behaviour when they detect that they are executing in a controlled environment.

Lakhotia and Singh have discussed at length how a virus can fool AV scanners, even those based on the most advanced formal techniques (see *VB*, September 2003, p.15). The limits of an AV scanner stem directly from the limits of static and dynamic analysis techniques, the foundation of all program analysis tools, including optimizing compilers. For AV scanners, the limits are debilitating for they operate in an environment where a programmer (virus writer) is the antagonist.

Metamorphic viruses enjoy their apparent invincibility because the virus writer has the advantage of knowing the weak spots of AV technologies. However, we could turn the tables if we could identify similar weak spots in metamorphic viruses. Indeed, Lakhotia and Singh close their otherwise gloomy article with one optimistic thought: '*The good news is that a virus writer is confronted with the same theoretical limit as anti-virus technologies… It may be* *worth contemplating how this could be used to the advantage of anti-virus technologies.'*

This article investigates the above remark and identifies what promises to be the Achilles' heel of a metamorphic virus.

The key observation is that, in order to mutate its code for generation after generation, a metamorphic virus must analyse its own code. Thus, it too must face the limits of static and dynamic analysis. Beyond that a metamorphic virus has another constraint: it must be able to re-analyse the mutated code that it generates. Thus, the analysis within the virus, of how to transform the code in the current generation, depends upon the complexity of transformations in the previous generation.

To overcome the challenges of static and dynamic analyses, the virus has the following options: do not obfuscate the transformed code in every generation; use some coding conventions that can aid it in detecting its own obfuscations; or develop smart algorithms to detect its specific obfuscations.

So, are metamorphic viruses really invincible? They are surely not as invincible as they first appear. A metamorphic virus's need to analyse itself is its Achilles' heel. If a virus can analyse itself, then an AV scanner should also be able to analyse it by using whatever method the virus uses to work around its own obfuscations. It is therefore conceivable that one could create a 'reverse morpher' that applies the transformation rules of the virus in reverse, thus undoing its attempt to hide from scanners.

Is there a catch? Before one can use a virus's methods on the virus itself, one has to extract those methods. One must first have a sample of the virus in order to extract its transformation rules, assumptions and algorithms.

This chicken-and-egg problem is no different from that faced by the current AV technologies for extracting signatures and behaviours. The important thing is that, once a set of tricks has been identified and countered by the AV software, the virus writer is forced to invent new tricks, thus raising the bar for the virus writer. Because of the additional constraints, the virus writer has to be more imaginative than the makers of AV scanners.

The rest of this two-part article is organized as follows. The next section provides an overview of mutation engines. It is followed by a discussion on the Achilles' heel of a metamorphic virus. In the second part of the article (which will appear in the January 2005 issue of *Virus Bulletin*) we present a case study by analysing the metamorphic engine of Win3/Evol. This leads to a discussion on developing reverse morphers to undo the mutations performed by a mutation engine. The article closes with our conclusions.

MUTATION ENGINES

At the heart of a metamorphic virus is a mutation engine, the part of the virus responsible for transforming its program. A mutation engine takes an input program and morphs it to a structurally different but semantically equivalent program.

Figure 1 identifies the three modules of any mutation engine: disassembly module, reverse engineering module and transformation module. Development of each of these modules poses different challenges and limitations.

In order to mutate its program, the virus must first disassemble it. One of the important tasks of disassembly is to differentiate between the virus code and data. If a virus cannot distinguish between code and data, it may transform the data, leading to incorrect behaviour. There are two known strategies for disassembly: linear scan and recursive traversal (see Schwarz *et al.,* 2002, Ninth Working Conference on Reverse Engineering). Each of these strategies has its own limitations (see Linn and Debray, 2003 Conference on Computer and Communications Security).

The third module, transform, generates a transformed version of the original program. A program must be transformed significantly in order to avoid being detected by a signature-based AV scanner. In the simplest case, the module may transform one instruction at a time. At the other extreme the module may analyse blocks of code and replace them with equivalent code fragments. To ensure accuracy of transformation a block must be a single entry, single exit piece of code. That means that control should not jump into the middle of the block, or else it becomes harder to create semantic-preserving transformations. One could also imagine transformations that replace segments of control flow graphs (CFGs) with other control flow graphs.

The second module, the reverse engineering (RE) module, supports the transformation module. The challenges for this module depend upon the technique chosen for transformation. As the transformations become more complex, so does the work of reverse engineering. If the transformation module works on one instruction at a time then the RE module does not need to do anything. However,

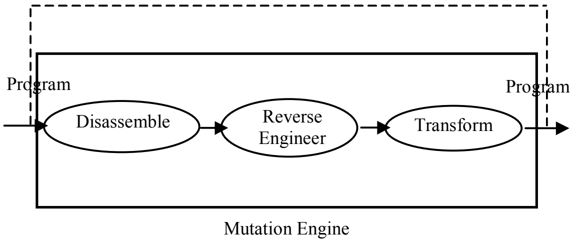


Figure 1. Stages of program transformation.

if the transformation module works on blocks of code, the RE module must identify blocks. Similarly, if the transformation module works on CFGs, the RE module should identify CFGs.

THE ACHILLES' HEEL

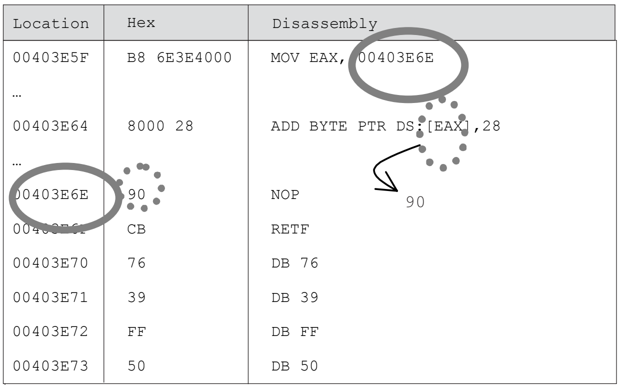
Lakhotia and Singh argued that virus writers enjoy the upper hand because they can exploit the limitations of static analysis as well as dynamic analysis to hide their code. Junk byte insertion, jump into the middle of instruction and self-modifying codes are a few obfuscation techniques that make it even harder to distinguish statically between data and code in a binary executable. Insertion of large loops and anti-debugging techniques test the patience and speed of dynamic analysis. A mutation engine that changes the virus code with every few generations as well as adding the complex obfuscation techniques to the newly created virus body might create a virus that is close to invincible.

Figure 1 shows that the steps involved in mutating a program are very similar to the steps outlined by Lakhotia and Singh for checking whether a program is malicious using program analysis techniques. There are two differences. First, a metamorphic virus uses the analysis of the first two steps for creating a transformed program. A scanner would use similar information to determine whether a program is malicious. Second, the output of the last step of a metamorphic virus becomes its input, albeit in a different execution of the program.

The feedback loop in Figure 1 has catastrophic consequences for a virus. A metamorphic virus has to analyse its own mutated code in order to mutate it further. If, after transformation, the virus introduces obfuscations that prevent its disassembly, then in the next generation the virus may not be able to diassemble itself. If it introduces obfuscations that prevent reverse engineering of the virus code – say, for instance, identifying its program blocks, then the virus will also not be able to detect its own blocks. Thus, the virus cannot introduce obfuscations that prevent those analyses that are performed by the virus itself.

To understand the problems faced in writing a metamorphic virus, let us analyse an obfuscation technique introduced by a non-metamorphic virus, W32/Netsky.Z.

The virus Netsky.Z introduces an obfuscation using a technique known as self-modifying code. Here, the virus is modifying code at location 00403E6E at run time. It adds 28h to the opcode 90h, which converts the NOP instruction to MOV instruction, thus modifying the code, as shown in Figure 2b. If we try to analyse it using a static technique we get the wrong analysis, as shown by Figure 2a.



Location	Hex	Disassembly
00403E5F	B8 6E3E4000	MOV EAX, 00403E6E
…		
00403E64	8000 28	ADD BYTE PTR DS:[EAX],28
…		
00403E6E	90	NOP
00403E6F	CB	RETF
00403E70	76	DB 76
00403E71	39	DB 39
00403E72	FF	DB FF
00403E73	50	DB 50

Figure 2a. Obfuscation through runtime code modification.

Location	Hex	Disassembly
00403E5F	B8 6E3E4000	MOV EAX, 00403E6E
…		
00403E64	8000 28	ADD BYTE PTR DS:[EAX],28
…		
00403E6E	B8 CB7639FF	MOV EAX, FF3976CB
00403E6F		
00403E70		
00403E71		
00403E72		
00403E73	50	PUSH EAX

Figure 2b. Modified code.

Now suppose a metamorphic virus writer has mutated its code such that the current generation is self-modifying. In order to mutate its code further it has to know statically the instruction that is changing at runtime. This challenge poses a serious limit to the obfuscation techniques a metamorphic virus can impose during mutation.

This highlights the Achilles' heel of a metamorphic virus: *a metamorphic virus must be able to disassemble and reverse engineer itself.* Thus, a metamorphic virus cannot utilise obfuscation techniques that make it harder or impossible for its code to be disassembled or reverse engineered by itself.

WIN32/EVOL

Win32/Evol is a relatively simple metamorphic virus. Nonetheless, it is a good example for a case study since the virus demonstrates properties common to metamorphic viruses – i.e. it obfuscates calls made to system libraries and it mutates its code before propagation. Part two of this feature will describe the details of these methods and discuss the development of reverse morphers to undo the mutations performed by a mutation engine. [*Part two will appear in the January 2005 issue of Virus Bulletin - Ed*]

FEATURE 2

HOW ‘DARE’ YOU CALL IT SPYWARE!

Prabhat K. Singh, Fraser Howard and Joe Telafici
McAfee, AVERT

Anti-virus vendors frequently receive queries, objections and legal notices from software vendors whose applications are detected as spyware or adware. As a result, the task of ‘proving’ whether a given sample is a spyware/adware or clean application demands careful judgment on the part of the researcher.

Several such applications cannot be ignored as benign software just because of the legal risks associated with detection. Adware and spyware may be described as applications that may carry out one or more unsolicited actions, such as spying on a user’s PC activities, gathering data about the user’s browsing habits or pushing unwanted and/or offensive advertisements into the user’s system.

The majority of researchers in the AV industry are aware of these descriptions, yet almost everyone has a different interpretation of adware/spyware behaviour and companies tend to add detection for applications that may not warrant inclusion in this category, or vice versa.

Despite a rash of legislative activity in the US and the EU, definitions of spyware, adware and associated terminology vary widely. This article presents a description of spyware/adware behaviour and visits a few criteria that may be used by researchers to prove that a program can be detected as adware/spyware. We break malware behaviour into six areas to achieve an environment within which spyware/adware can be compared with malware programs.

MALWARE PROGRAM BEHAVIOUR

In the following sections we will discuss six areas of malware behaviour and structure: installation, survey, replication, concealment, injection and payload (for more information on this method of malware analysis see http://downloads.securityfocus.com/library/masterthesis.pdf).

Installation

An installer creates and maintains an installation qualifier so that the malware can execute on the victim system, and ensures the automatic interpretation of malware code.

In this definition there are two criteria for a segment of code in the malware to qualify as an installer. First, the code should cause a permanent change in the machine’s security state. Second, the code may ensure that the program is invoked after every time the system is restarted or on the occurrence of some system event. Hence, one or more of the following activities may be observed during spyware/adware installation:

Installation of a service to ensure that the application is active even when a user is not logged on

The information relating to the services installed on the local machine is stored in the Service Control Manager (SCM) database. The Win32 SCM APIs may be used to add, query or control the status of the services installed.

COM class registration

Frequently, spyware/adware applications use COM objects to achieve software reusability and adaptability. An understanding of the relevant parts of the system which may be altered during a COM object’s installation is important in order to understand the behaviour of the spyware/adware program.

A COM server is a binary file that contains the code for methods used by one or more COM classes. This server can be packaged either as a DLL or as an executable file. There are two types of activation request for an object: an in-process activation request and an out-of-process activation request. An in-process activation request requires a DLL-based version of the COM server to be present and loaded in the client’s memory space. An out-of-process activation request requires the executable to be used to start the server process.

In order to allow client programs to activate objects without concern for which type of package is used or where the executables or DLLs are located, COM stores configuration information in the registry that maps the class IDs (CLSIDs) onto the server that implements that class. Whenever an activation request is made for a CLSID in a given machine, the registry is consulted. If the configuration information is not available in the registry, the request may be redirected to a remote host from which the relevant code may be downloaded and installed. COM stores most of the configuration information in the registry key ‘HKLM\Software\Classes’, although most programs use the more convenient alias ‘HKCR’. COM keeps machine-wide information related to CLSIDs in the registry key ‘HKCR\CLSID COM’ and also looks into per-user class information in the ‘HKCU\Software\Classes\CLSID’ key.

COM stores all the locally known CLSIDs under either of the above two keys, with one subkey per CLSID. For example, let there be a class named ‘JoeSpy’ which is used to implement spyware functionality.

This will have a registry entry as:

```
[HKCR\CLSID\{571F1680-CC83-11d0-8C48-0080C73925BA}]
@="JoeSpy"
```

In order to allow local activation of ‘JoeSpy’ objects, JoeSpy’s CLSID entry in the registry will have a subkey that indicates which file contains the executable code for the JoeSpy class’s methods. If the COM server is packaged as a DLL, the following entry will be required in the registry:

```
[HKCR\CLSID\{571F1680-CC83-11d0-8C48-
0080C73925BA}\InprocServer32]
@="C:\JoeSpyware.dll"
```

If the COM server is a package that uses an executable file, the following entry will be required in the registry:

```
[HKCR\CLSID\{571F1680-CC83-11d0-8C48-
0080C73925BA}\LocalServer32]
@=" C:\JoeSpyware.exe"
```

If the environment does not cope easily with raw CLSIDs to make activation calls, a programmer may use ProgIDs. The CLSID to ProgID translation is achieved through the following registry entry:

```
[HKCR\CLSID\{571F1680-CC83-11d0-8C48-
0080C73925BA}\ProgID]
@="JoeSpyware.JoeSpy.1"
```

Conversely, to support the reverse mapping (ProgID to CLSID), the following keys are needed:

```
[HKCR\JoeSpyware.JoeSpy.1]
@="JoeSpy"
[HKCR\JoeSpyware.JoeSpy.1\CLSID]
@="{571F1680-CC83-11d0-8C48-0080C73925BA}"
```

A well-implemented COM server will implement the following two functions to support installation and uninstallation:

```
DllRegisterServer(void);
DllUnregisterServer(void);
```

However, most adware/spyware programs do not follow this practice.

Browser Helper Object (BHO) installations

These are frequently used by adware applications for installing toolbars or redirecting network traffic in *Internet Explorer* and the *Windows Explorer*. The BHO is installed as a COM in-process server registered under the ‘HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects’ registry key, and registers the CLSIDs for all the BHOs installed in the system.

Assuring execution at system initialization

This is usually achieved by using Run and RunOnce registry keys. These keys are present in the HKCU and the HKLM hives of the registry. Generally either ‘HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run’ or ‘HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run’ is used.

Both of these key entries ensure that the spyware/adware program is executed every time the user logs into their account. The difference between the two keys is that the HKLM key executes the program *before* the appearance of the desktop, while the HKCU key executes the program *after* the appearance of the desktop.

LSP and NSP installation

Spyware and adware installations (e.g. a version of *MarketScore*) use Winsock 2 Layered and Network Service Provider implementations to redirect network traffic to specific sites. The network data emanating from and entering the machine can be sent to a ‘central’ site and then sent to the user’s application.

For example, when the user types ‘www.google.com’ into their Internet browser, the browser will connect silently to www.marketscore.com and send part of this data, *before* connecting to www.google.com. This is a more powerful mechanism than that achieved through a BHO because the interception of network traffic is not application-specific (i.e. not limited to *IE* using a BHO). This is because LSPs work at the TCP implementation level.

LSP is a component that intercepts winsock2 calls and has the ability to manipulate them, and then optionally pass them to the winsock2 provider. There are two kinds of LSP, the transport service provider and the name space service provider. The former is used frequently by adware/spyware for implementing LSPs. An LSP is implemented as a DLL and should be registered with the system. The usual registry location is ‘HKLM\System\CurrentControlSet\Services\Winsock2’.

The LSP will always export the WSPStartup() function. This function is invoked whenever a user calls the WSAStartup() function. LSPs are organized in a chain wherein a network-related function call invokes the first node in the chain. This node processes the call and invokes the next node in the chain and so on, until the last node invokes the winsock2 function. An exhaustive treatment of this topic can be found at http://www.microsoft.com/msj/0599/LayeredService/LayeredService.aspx.

Survey

Survey behaviour identifies appropriate targets, network hosts or objects and their locators so that other behavioural units can perform correctly. Here, a locator is an address or path information to the target.

Survey behaviour is the reconnaissance activity that malware programs carry out to find more vulnerable host addresses for the purpose of replication. This activity is observed in viruses and worms and is *absent* from adware and spyware. This activity should not be confused with the spyware behaviour that involves collecting information from the user's PC for market survey-related purposes.

Replication

Replication behaviour provides the logistic mechanisms for the transfer of malware code. Logistic mechanisms are technical and/or social engineering methods for the transfer of malware from an infected host to another target host.

Although spyware and adware may carry out a lot of network activity, they do not exhibit replication behaviour as defined above. Replication using social engineering should not be confused with installation using social engineering methods where the user is duped into installing the program onto their PC. Currently, it is safe to say that adware and spyware are characterized by the *absence* of replication behaviour – otherwise they could be put into the category of self-propagating programs (worms) and there would be no question about the legality of their detection by any AV software.

Concealment

Concealment behaviour prevents the discovery of the activity and structure of a malware program in order to avoid detection, forensics and removal.

Software forensics can be used for author identification and characterization. The absence of author information or presence of conflicting information may be a strong indication of concealment. The use of concealment in spyware/adware is intended to increase the complexity of analysis and thus increases the difficulty in 'proving' an adware/spyware program. In many cases, variable CLSID values are generated during several instances of installation of the same sample. The proving process may not be reliable if we base our conclusions only on the CLSID 'blacklist' to identify known malicious COM objects (e.g. a version of *Virtumonde*).

Concealment has been classified into two categories, namely, concealment that prevents any dissemination of program information (PPID) and concealment that is achieved through attacks on the system's security mechanism (ASM). Currently, the concealment approaches for achieving PPID in spyware/adware are trivial and nearly all of these have been borrowed from the virus-writing domain.

The first approach to achieving PPID 'blocks' the availability of a program's executable image to the researcher. In this case the executable image of the program remains in the memory of the infected system. Once the system is rebooted, the malicious program needs to be installed on the system again. Such attacks are used more often in the conventional worm and virus-style attacks than in adware/spyware.

The second approach to achieving PPID uses code evolution. We define code evolution as the process of creating program equivalents in such a way that, given an identical input sequence of symbols, they produce an identical output sequence of symbols. The variations of this are: polymorphism, metamorphism and packing – these are well known by virus researchers and will not be discussed in this article.

Injection

The injector behaviour causes an injection of malware components into the victim object such that one or more of the malware components is placed in the execution space of the victim object. The copy of the malware may be exact or an evolved instance of the original malware, after being processed by the concealment behaviour. The execution space of an object is the code/text segment of the victim object or the environment in which the interpretation of the object will take place. Injection may or may not be present in adware/spyware.

While carrying out concealment and installation, a malicious program may need to inject all or parts of its components into another program in the memory. Common activities may involve DLL injection into the address space of another process. This is usually achieved using the following methods:

DLL injection using the registry

This is achieved by adding the DLL pathname as a data value for a subkey in the following registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs

During system initialization when user32.dll is mapped into a process's address space it automatically reads these registry key entries and loads any DLL mentioned in the data value field.

DLL injection using Windows hooks

This is achieved by implementing a system-wide hook using the SetWindowsHookEx() function. When a system-wide hook is called from the context of another process, the system loads the DLL containing the hook into the process's address space.

DLL injection using remote threads

Most *Windows* functions will allow a process to manipulate its own address space. Some functions do exist that can be used for manipulating other processes. In order to load a DLL in another process and get loadlibrary() to load a DLL into the other process, the program creates a new thread in the other process using the CreateRemoteThread() function. Since the CreateRemoteThread()API is not present in *Windows 9x*, this method will not work on all platforms.

Injection targets

Spyware/adware programs usually inject themselves into processes like *Internet Explorer* and *Windows Explorer*.

Payload

The payload is a behaviour programmed into the malware that is used to achieve a specific purpose for which the malware was created.

Two types of payload behaviour are observed both in spyware/adware and in malware.

Host payload behaviour

This type of payload carries out activities such as displaying a text message that may be a threat, warning, joke or an advertisement banner on the user's computer. Payloads may include excessive consumption of computation resources leading to a denial of service or complete destruction of a specific resource on the user's computer.

Network payload behaviour

Network payloads collect information from the host computer and send information to another host on the network. The main difference between replication and payload-related network behaviour is shown in Figures 1 and 2. In both cases we observe the use of email addresses, URLs, IP addresses or IRC channels for connecting to network targets, but in the case of replication, network addresses are variable in nature (obtained from the infected host itself). In network-based payloads, the network targets are usually hard-coded in the malware program or are received in the form of command/control information from fixed source hosts on the Internet (these may come bundled in the malware code). Network-based payloads may be further categorized on the basis of the direction of relevant information flow occurring during the network activity:

- Outward flow – the information flows from the infected object to a remote object.
- Inward flow – the information flows from the remote object to the infected object.

The outward information flow is observed more in spyware, while the latter activity has been prevalent in adware.

CRITERIA DISCUSSION

In the previous section we observed that adware and spyware programs display only four of the six areas of malware behaviour: installation, injection, concealment and payload behaviour. The presence of survey and replication behaviour is sufficient to eliminate a sample from the adware/spyware category. In this section we discuss some criteria that will aid the decision process.

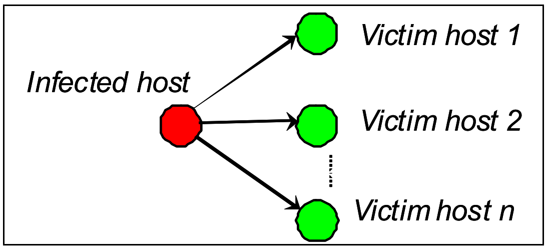


Figure 1. Replication network behaviour.

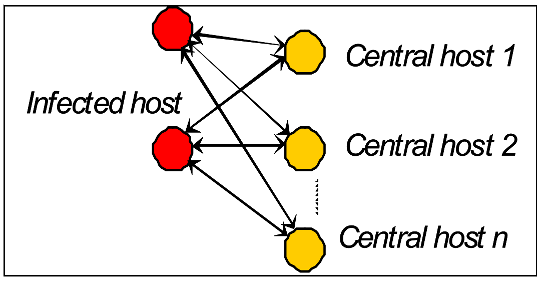


Figure 2. Payload network behaviour.

Each criterion in this section is based on at least one of the following factors: integrity, privacy of user information and uninterrupted availability of system resources. Two types of criteria are discussed here, one is based on the program structure (achieved through static analysis) and the other is based on observations of program execution (achieved through dynamic program analysis).

Static analysis

A product should not misrepresent its creator. For example, if the sample has a resource section that falsifies a CompanyName resource attribute, it should be interpreted as abuse of user's trust and hence violation of system integrity through concealment. We have seen this in at least one spyware sample.

Several known adware and spyware programs have been found to be packed using a packer or an obfuscator. Packers such as UPX, do not fall into this category since these come with an unpack option. However, there are patched UPX variants that have been used in packing adware and spyware programs – these will be considered malicious. A number of executable packers are used solely or almost entirely by malicious software. Thus the use of 'questionable' packers or obfuscators, such as morphine, UPX (Redir), telock, petite, pepatch, fsg, pecompact, exestealth, obsedium, ezip, pespin, krypton, exe32pack, pex and pediminisher, may be a strong indicator of questionable intent. Packing is used to conceal the URL strings or IP addresses embedded in the data, resource or text section of the sample. These are addresses of sites to which the adware program will connect. A good example for this is a version of *Adware-Lop*, which encrypts and stores all URL strings (in the resource section of the program) to which it may connect.

Dynamic analysis

During the installation phase, the product should inform the user (through the end user licence agreement, or EULA) exactly what the program will do. Sufficient information in the EULA should be provided, which can convince the user that the program will not invade their privacy or tamper with the system's integrity (with respect to security).

The presence of 'fine print funny business' in the EULA is a strong indicator that an adware/spyware program will be installed (see http://grc.com/oo/cbc.htm). Also, we need to check for the availability of a privacy notice from the developer during the installation process, or at least some information that points to the developer's website indicating the same.

The privacy policy and the EULA should hold true for all the components that are installed by the product at a current or later stage. Programs which announce and do not respect well-defined privacy and anonymity constraints will generally be classified as adware.

If the program exhibits a 'silent' installation behaviour, wherein the components are downloaded and installed on the user's PC while the user is browsing a website, this is a strong indicator of spyware/adware. This violates the system integrity of the user's machine and also consumes network and system resources which may be paid for unknowingly by the user.

The product should not install and execute hidden processes. Also, the product should not install and run programs that masquerade as well known system programs (e.g. svchost.exe). This is a form of concealment that violates the system's integrity since the information on the system is altered without the user's knowledge or consent. If the program modifies the browser settings or launches a browser automatically and displays arbitrary content, it is a violation of system integrity policy and warrants detection as adware.

For a product to qualify as 'well-behaved', it is very important that an uninstall feature be available and functional. Some adware programs, such as a version of *CommonName*, will automatically conceal or 'morph' themselves to an undetectable form when their uninstaller is executed.

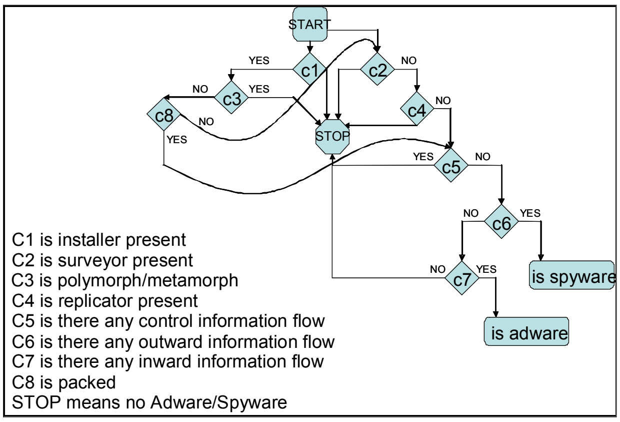

START
c1
c2
c3
c4
c5
c6
c7
c8
STOP
YES
NO
is spyware
is adware
C1 is installer present
C2 is surveyor present
C3 is polymorph/metamorph
C4 is replicator present
C5 is there any control information flow
C6 is there any outward information flow
C7 is there any inward information flow
C8 is packed
STOP means no Adware/Spyware

Figure 3. A possible procedure for determination of spyware or adware.

If the program sends information over the network without the user's consent (e.g. email address, banking or credit card information, account names or passwords), this is an indicator of spyware behaviour. Sometimes information gathered about the user's browsing habits may also be transmitted by spyware, which may not be acceptable to the user. If the injection of the program or one of its components is done in *Explorer* or a browser program, there is strong reason to believe that this is done to achieve concealment from outgoing firewall rule sets. This activity may carry out transmission of sensitive data which would have been blocked by a firewall. These types of activity violate the user's privacy. If the program receives command and control information from a machine outside the user's domain of control, this indicates a network payload behaviour which eventually violates the privacy, integrity and resource usage policy of the user.

Based on the behaviour studied in the previous section Figure 3 is a flow chart that summarizes the criteria we have discussed.

CONCLUSION

At the time of writing, adware and spyware account for nine of the top 20 threats detected by *McAfee* users (see http://vil.mcafee.com/mast/viruses_by_continent_internal.asp). Large ISPs and IHVs report that a large portion of their technical support calls arise from the side effects or degraded performance resulting from the presence of spyware or adware programs on their users' systems.

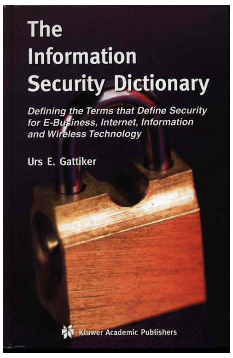
While legislative activity and litigation may alter definitions of spyware and adware in the future, it behooves us as an industry to develop consistent criteria and definitions. This will ensure better customer understanding and reduce legal exposure. It also allows us to assist those vendors who are making an honest effort to clean up their acts.

BOOK REVIEW

DICTIONARY DEFINITIONS

Eddy Willems
NOXS and EICAR, Belgium

Title: The Information Security Dictionary
Author: Urs E. Gattiker
ISBN: 1-4020-7889-7
Publisher: Kluwer Academic Publishers


The Information Security Dictionary
Defining the Terms that Define Security for E-Business, Internet, Information and Wireless Technology
Urs E. Gattiker
Kluwer Academic Publishers

Earlier this year I found myself searching for a book which would help my co-researchers (see *VB*, August 2004, p.10) to define some of the terms they would come across within the field of information security. It was at that moment that *The Information Security Dictionary* appeared.

The Information Security Dictionary attempts to explain the terms that define security for e-business, Internet, information and wireless technology. It is written by Dr Urs E. Gattiker, who has co-authored other security-related books, such as *Viruses Revealed* (in 2001 with David Harley and Robert Slade).

The book has what I would call 'something for everyone'. The first edition defines over 1,200 of the most commonly used words in the security field, with particular attention being paid to the terms used most often in computer forensics, and those relating to malware, viruses and vulnerabilities.

This dictionary will help non-specialist readers understand the information security issues they encounter in their work or in studying for certification examinations – but it will also help the real IT security expert in pinning down a definition for a specific term or word.

The goal throughout has been to provide a comprehensive dictionary of terms that will increase access to works in all sciences. Even statistical definitions are included, since IT security is moving rapidly towards becoming a more established scientific discipline.

Special attention has been paid to terms which may prevent educated readers from gaining a full understanding of journal articles and books in cryptology and security and information systems, as well as applied fields that build on these disciplines.

A number of definitions have been included for terms that might not strictly be associated with information security – for instance: validity, reliability, attitudes and cognition. The author reasons: '[These words] meet the main criteria for inclusion: the words pop up fairly often, and many people are unsure of the meaning.'

The emphasis throughout the book is on concepts, rather than implementations. Because the concepts are often complicated, readers may find that a definition makes sense only after it has been illustrated by an example – as a consequence, the explanations and illustrations are sometimes longer than the definitions themselves.

As in any language, more than one word may be used to express the same idea. In such cases the author has included a full definition for what he believes to be the more commonly-used term, while the other terms are defined briefly and cross-referenced.

The rules used for the dictionary's listings are minimal but important to understand. For instance, when a term such as 'virus' has several related terms (e.g. polymorphic virus), the related definitions may all appear as sub-entries under the definition of the main term ('virus'). Under the entry 'polymorphic virus' the reader is simply referred to the 'virus' entry for further explanation. This helps non-specialist readers to find their way around faster when dealing with unfamiliar terms.

The following is an example of a definition from the dictionary:

> '**Virus** is a segment of a computer code or a program that will copy its code into one or more larger 'host' programs when it is activated. Unfortunately, it also may perform other unauthorized actions at that time (see also Merging of Attack Technologies, Trojans, Virology).
>
> 'To illustrate, Virus is a program that searches out other programs and infects them by embedding itself in them, so that they become Trojans. When these programs are executed, the embedded virus is executed as well, thereby propagating the "infection". This process tends to be invisible to the user.'

The book is well and logically structured, with clear figures and tables. The appendices provide one of the most comprehensive listings I have seen of informational dictionaries and other resources, with a good selection of URLs for some interesting and useful websites.

I was not always fond of the illustrations and definitions I found in this book. Certainly some experts will feel that a number of the definitions are not complete or that the illustrations are not always sufficiently strict. The dictionary could be considered a little too general for the more experienced security or anti-virus experts. Nevertheless, in my opinion the author has done a good job, and this dictionary is a must-have for anybody who works with or who is interested in information security.

This is yet another book to add to my ever-expanding anti-virus and IT security library. When will it end?

LETTERS

GENERIC DETECTION – A SPECIFIC CASE

This subject has effectively been glossed over for some 10 years. One reason is that anti-virus researchers who could write about it have been slightly scared to do so, lest the information be of value to competition. I feel I can risk it now.

McAfee received (from Andreas Clementi of the University of Innsbruck) a collection of some 1,350 *.HTM files. These are text files. Any of our customers can look at the contents. The files are mostly from virus/Trojan writing groups, although some are from AV experts, about virus/Trojan authors and their techniques, backgrounds and attitudes.

Should we detect these files, for any reason other than the fact that we may be reviewed against them? I took a good look. My conclusions may surprise you.

Some 340 of the files are ones we should certainly *not* detect. These include:

- An interview with Dr Alan Solomon, by virus author Dark Fiber.
- A report of the death of an Australian virus author.
- The well reported interview with Dark Avenger by AV researcher Sarah Gordon.
- Innocent (and valid!) expressions of opinion by people in the AV industry.

However, we should detect most of the others, nearly all of which were written to educate and inform virus authors. There are three more reasons:

- IT Managers like to know if this type of material is residing on any of their machines.
- Internet companies which pass high message volumes like to know if they are being used for malware group communication.
- Detection will inconvenience the malware groups, and make them slightly less productive.

Initially I decided to write the detections so that reviewers could use them, and to make them available for general use later if we decided to do so. The files will, of course, be detected as applications, not as viruses or Trojans.

So, I had just over 1,000 detections to write, and they needed to be written efficiently. They had to be generic, in order to minimise the workload. I could see it was easy, because the files contained lots of very strong detection strings, many of which occurred in more than one file.

The generic technique used was simple: where a detection string occurs in more than one file, search for it in a slightly extended area, rather than at a specific offset. Do this so that all files containing that string are detected by a single search.

The bad news? Well, the 1,340 files came in a collection of 13,500 files of virus-associated material. I still need to look at those. No doubt, the question of whether virus or Trojan source code should be detected will raise its ugly head once more!

Peter Morley
McAfee, UK

COFFEE-TIME AMUSEMENT

Whilst drinking my coffee and scanning the funnies one morning, I happened across a message which stated that *Microsoft* was evil – it turned out to contain a link to a website that rates the 'goodness' of other websites using a numerological method.

Being a long-time member of the anti-virus community I decided to run a number of anti-virus sites through the 'goodness' test. The results (at the time of testing) were as follows:

http://www.bitdefender.com is rated 18% evil, 82% good

http://www.ca.com is rated 34% evil, 66% good

http://www.f-secure.com is rated 62% evil, 38% good

http://www.kaspersky.com is rated 55% evil, 45% good

http://www.mcafee.com is rated 23% evil, 77% good

http://www.messagelabs.com is rated 14% evil, 86% good

http://www.nod32.com is rated 98% evil, 2% good

http://www.norman.no is rated 45% evil, 55% good

http://www.pandasoftware.com is rated 13% evil, 87% good

http://www.sophos.com is rated 50% evil, 50% good

http://www.sybari.com is rated 2% evil, 98% good

http://www.symantec.com is rated 30% evil, 70% good

http://www.trendmicro.com is rated 63% evil, 37% good

http://www.virusbtn.com is 7% rated evil, 93% good

If you would like to try some of your own visit http://homokaasu.org/gematriculator.

Anon, UK

[*Disclaimer: Virus Bulletin would like to point out that the above has been included as a frivolous coffee-time distraction, and the inclusion of this letter should not be taken to be a statement about the content of the websites or their associated companies – furthermore, the fact that Virus Bulletin's own website is rated as 93 per cent good is pure coincidence …*]

PRODUCT REVIEW

TREND MICRO PC-CILLIN INTERNET SECURITY 2005 12.0.1330

Matt Ham

It is a pretty much universally accepted fact that, in software development terms, a release date of 'next week' can mean a delay of anything up to 12 months. Proving the exception to that rule, however, is *Trend Internet Security 2005* which has become available to all and sundry well before those 12 months have commenced. Within the product the name used is *Trend Micro PC-cillin Internet Security 2005*. For the sake of brevity therefore, the product will be referred to throughout this review as *PCCIS*.

At a cursory glance *PCCIS* has much the same interface and functionality as previous *Trend* anti-virus products. However, *PCCIS* contains numerous new features as well as additions to older ones. This is also an application that has been introduced after the release of *Windows XP SP 2*, so one would hope that it has been designed to integrate with the operating system upgrade.

COMPANY OVERVIEW

Trend Micro has a geographically varied history. My first experience with *Trend* was with the 'Trend Chipaway' BIOS-based boot sector protection method, when the company was based in Taiwan. From here *Trend* moved its centre of operations to Japan. Later still, the company became more associated with the US, but it retains a large presence throughout the Asia-Pacific region and especially the Philippines.

In terms of its products, latterly the company has concentrated mostly on anti-virus solutions, along with some aspects of security which impinge upon that core business. For years, server-based products were the mainstay of *Trend*'s corporate presence, but as other companies started to introduce their own gateway solutions *Trend* strengthened its presence in the desktop field. *Trend* has also produced hardware anti-virus solutions, most notably for the home user market.

INSTALLATION AND UPDATE

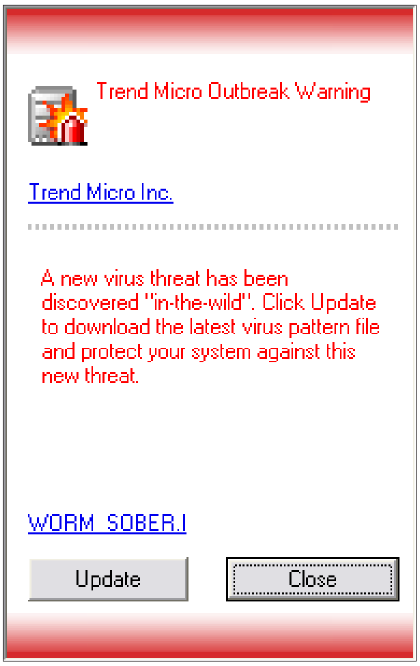
Where home user products are concerned it is not unreasonable for the developer to reduce user interaction to a minimum, working on the theory that the bulk of users will be not be helped by the option to set up exclusion lists or to determine how many levels of recursion should be used when scanning inside archives.

Installation of *PCCIS* was, therefore, remarkably free from user interaction. Information boxes appeared on an unpatched version of *XP*, stating that an older version of *Microsoft Installer* existed (though these were absent in the *SP2* version of *Windows*). Once the user has accepted the licence agreement an initial scan is performed for Trojans and viruses, after which the user's name and organisation are checked. Next there is the option to select a different installation location from the default, after which installation completes without further ado. A reboot was not required on installation, though it was necessary after uninstallation.

Following installation only three items had been added to the Start menu: *PCCIS* itself, the uninstallation application and the associated help files. Removal of the program could also be achieved by running the installation program again, which makes a change from the usual situation where this merely reinitialises the installation.

Updating the product was similarly user-friendly. Once installation had been performed no further adjustments to the settings were required before the update could be triggered. One update was provided prior to registration, allowing the demonstration product to have at least a few days of fully updated capabilities upon which to be judged.

By default, updates are set to occur every three hours, with longer delays available if required (though not shorter ones). Updates seemed to be triggered by booting the machine, however, so the three-hour delay should be the maximum update delay under any circumstances. Updates can be silent if required. Partially offsetting this is the Outbreak Warning pop-up, which allows for dangerous new malware to be announced and an update to be offered without going through the generic update process.



Trend Micro Outbreak Warning

Trend Micro Inc.

A new virus threat has been discovered "in-the-wild". Click Update to download the latest virus pattern file and protect your system against this new threat.

WORM_SOBER.I

As is standard, proxy connections are supported by the update procedure. More obscure, though, is the ability to set the updates separately for application files, anti-spam and anti-virus plus firewall functionality. This seems a rather pointless area of control.

Given a network where *PCCIS* is installed, any machine may be used to update any other on the network. Of course, there are caveats. For one, it seems necessary for the products on these machines to be *PCCIC 2005* rather than any previous version or any other version of *PC-cillin*. Various products were tried and only the newest version could be updated in this way. Furthermore, all the products to be updated must be password-protected in their settings, and the password must be identical on all machines being updated.

DOCUMENTATION AND WEB PRESENCE

Trend Micro was fast, insightful or lucky enough to lay claim to the 'www.antivirus.com' URL, which offers a memorable portal for *Trend* access. Another of *Trend*'s URLs is www.trendmicro.com. This resolves to a selection of regional sites (in an impressive piece of regionalisation it seems that *PCCIS 2005* is named *PCCIS 12* in Europe, adding more than a little to the naming confusion).

The contents of all regional *Trend* sites are much the same and will hold no surprises for regular visitors to any anti-virus website. Of note, however, are the virus descriptions. These contain a general section for each virus, along with a more technically in-depth description. The technical descriptions contain a wealth of information on such matters as the registry changes made and filenames used by the virus in question. These are sufficiently complete that competing anti-virus products have, in the past, been known to false-positive on these pages.

An online scanner, *HouseCall*, is also available from the website. This requires some slackness in browser security settings in order to operate, as well as a java platform – which does make for a rather slow experience for the first scan on *XP* (since not only the engine and virus database must be downloaded, but also the java platform). Subsequent scans are far speedier however. In a blatant attempt to throw a spanner in the works, the online scanner was run on a machine on which *PCCIS* was already installed. Even while running *HouseCall*, a standard on-demand *PCCIS* scan and on-access scanning, there were no terrible ill effects. My mp3 player stuttered somewhat under the onslaught, but otherwise matters proceeded as normal.

In line with the wider security coverage in *PCCIS*, the website also offers an online security scanner known as *Hackercheck*. This is a rather basic port scanner, and as such contains several warnings that running the application in a corporate environment may result in the descent of blood-crazed sysadmins upon the hapless user. Although the functionality is not notable in this offering, the typos present on the results page are entertaining.

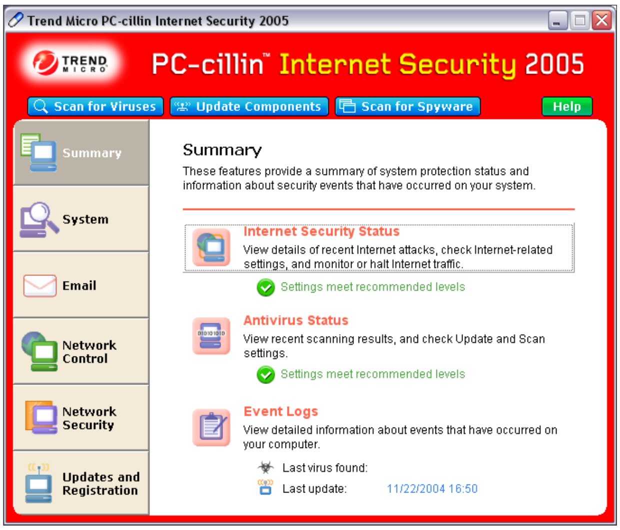
Context-sensitive help is available throughout the main application, in addition to the standalone help offered in the Start menu. These both arrive at the same set of help material, which combines information on how the software should be operated with corresponding information as to why this is the case. In most instances the information is well written and will be useful to an average user. In some cases, however, the information is at best perplexing and at worst misleading. For example, when discussing quarantine handling it is stated that 'Most Trojans can be identified by the name: TROJ_, VBS_, or JS_.' This may be true due to the *Trend* method of file description within logs, but as a general statement it is not likely to help the user gain a greater understanding of how to identify a Trojan.

GENERAL SECURITY FEATURES

With the standard anti-virus functionality much as might be expected, the real interest in *PCCIS 2005* lies with its additional security features, many of which are directly or indirectly related to virus threats.

The Email section of the main GUI demonstrates this mix of functionality. The standard offerings of SMTP and POP3 scanning are clearly present. By default, these are limited to one level in their archive recursion and to 3 MB in size, presumably as a method of limiting overheads. As home users are the major target market, however, webmail scanning is offered as a second category, with much the same default settings.

All is not dominated by viruses, however, and the third category within the Email area is devoted to spam. Here,



Summary

These features provide a summary of system protection status and information about security events that have occurred on your system.

Internet Security Status

View details of recent Internet attacks, check Internet-related settings, and monitor or halt Internet traffic.

✓ Settings meet recommended levels

Antivirus Status

View recent scanning results, and check Update and Scan settings.

✓ Settings meet recommended levels

Event Logs

View detailed information about events that have occurred on your computer.

Last virus found:

Last update: 11/22/2004 16:50

various levels of detection are offered – and, of course, there are addresses for submitting emails to *Trend* which are either false positives or false negatives.

As noted during installation, an anti-Trojan function is included in the application. This is integrated with the general virus scanning however, so does not really count as new material. The fact that this functionality has been given its own category is perhaps a sign that users are more aware of Trojans as a threat.

Along the same lines, though less virus-oriented and a separate function, a scan for spyware can be performed. By default this option is not activated. Activation of spyware scanning reveals a number of categories, of which hack tools, diallers and password crackers are the default options. Judging by these categories, *Trend*'s definition of 'spyware' covers a multitude of sins. The definition of spyware is a contentious one (see p.8), but the help resources for this function are well detailed and cover the possibilities that some files may be desirable even if they are detected here.

In the Network Control area of the GUI the security issues are more closely associated with virus prevention. The Personal Firewall is primary among these. Although detailed investigation of the firewall was not performed, there were certainly no open ports available amongst those which were investigated. In addition, the ports scanned were logged in the appropriate log-file. By its description the firewall seems to be an update to the *Windows XP* inbuilt *ICF*.

Also falling under the category of network control is the Private Data Protection function. With an increasing number of worms attempting to transmit personal data to undeserving recipients, this function is certainly virus (and spyware) related in scope.

Although it was not tested, the Private Data Protection feature purports to prevent sensitive personal information from being exported by HTTP, SMTP or instant message. The relevant information must first be entered into a *PCCIS* database of such items – suggested terms are: name, credit card number, login name and the like. There is only a suggestion, albeit a strong one, that this information be password-protected, so this is an area in which security could be compromised by any person with physical access to the machine. Such a collection of personal data in one place would be a huge labour-saving device for those in search of personal information if passwords are not enforced.

In the same area of the GUI is the ability to block URLs by type. This, however, does not include a type which relates directly to virus distribution sites – which seems an odd oversight. Sites which offer 'resources to affect or influence real events through the use of spells' can be blocked, however, which is rather telling about the priorities of some users. Finally in this area are security options for Wi-Fi networks, which were not tested.

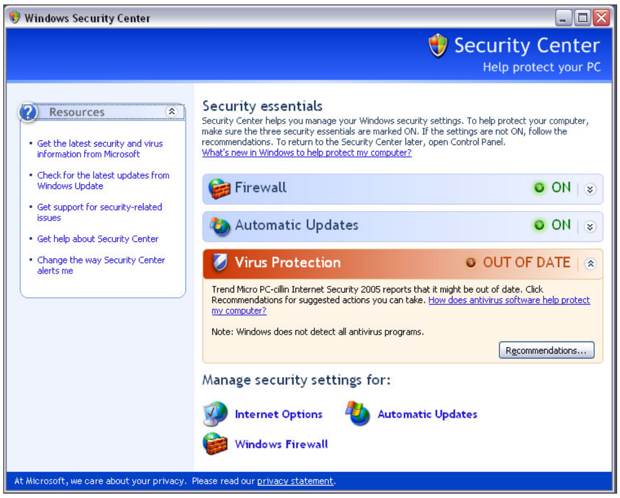
The Network Virus Emergency Center area is a function whereby various worms are recognised through their network activity and a direct response can be set if these are detected. What is more useful than might at first appear to be the case, is the fact that the activities that can be detected are not based solely upon known worms. Although there are some specific entries, such as WORM_BAGLE.AU, the list also includes much more loosely-defined attacks, such as 'MS02-039_SQL_SERVER_RESOLUTION_EXPLOIT'. This is directly related to a *Microsoft* security alert and is that which was used by the SQL/Slammer worm to great and terrible effect. In this case it is the exploit that is detected, rather than Slammer's particular implementation. In many ways this is a preferable method of detection. If any of these worms or exploits are detected the user may either be sent a pop-up to this effect (the default action) or, more sensibly, the user can opt to halt all Internet traffic pending investigation.

Of course, the presence of known security exploits is in many cases followed by the presence of patches which nullify the effect, at least on patched machines. With *SP 2* installed an *XP* user will be bombarded with instructions to scan for updates and patches, but of course not all everyone will be using *XP SP2*. It is therefore welcome that, among the scans offered within *PCCIS*, is one for known security vulnerabilities. This seemed to work variably, having no issues on machines where vulnerabilities were present (these were detected), but it left an unclosable dialog box if no problems were detected.

INTEGRATION WITH WINDOWS XP SP2

Some users, myself included, are sufficiently irritated by the balloon pop-ups within *Windows XP* that they remove them via an assault on the registry. However, for those who are not quite so rabid in their dislike of intrusiveness these can be a source of useful information. Windows Security Center is a new feature in SP2 which makes liberal use of these alert bubbles as well as providing an area where firewall, anti-virus and Windows Update status are described. This description warns in no uncertain terms if settings are not in a *Microsoft*-declared list of acceptable states.

The default installation of *Windows XP* cares little whether there is an anti-virus product installed. This changes under *SP2* where small balloons will pop up if no known product is found. Of course, there should be heavy stress on the word 'known'. An anti-virus product must be recognised by *Windows XP* in order for this alert to be laid to rest. In the case of *PCCIS* the first hurdle is crossed easily – upon installation the balloon was replaced by another, which


Windows Security Center
Security Center
Help protect your PC
Resources
Get the latest security and virus information from Microsoft
Check for the latest updates from Windows Update
Get support for security-related issues
Get help about Security Center
Change the way Security Center alerts me

Security essentials

Security Center helps you manage your Windows security settings. To help protect your computer, make sure the three security essentials are marked ON. If the settings are not ON, follow the recommendations. To return to the Security Center later, open Control Panel.
What's new in Windows to help protect my computer?

Firewall ● ON

Automatic Updates ● ON

Virus Protection ● OUT OF DATE

Trend Micro PC-cillin Internet Security 2005 reports that it might be out of date. Click Recommendations for suggested actions you can take. How does antivirus software help protect my computer?

Note: Windows does not detect all antivirus programs.

At Microsoft, we care about your privacy. Please read our privacy statement.

declared that the installed product was out of date. This state of affairs was echoed by the larger descriptive GUI, which used almost inch-high text to drive the point home.

As hoped, the updating process put a stop to these warnings. The alerts, or at least a subset of them, could be reactivated however, by turning off on-access scanning. This is an area where a warning might be very useful. Many commercial software products, usually in a spirit of overblown paranoia, request that anti-virus software be disabled during installation. The alert this will engender is no bad thing, although it is somewhat paranoia-inducing when the alert is triggered momentarily during updates to the product.

Although the irritation factor of these alert balloons is high, it is not at all undesirable. One of the long-term problems with anti-virus programs is that, for the bulk of the time, it is more irritating than not to have them fully operational and updated. The big-bang event where infection occurs is not often considered by home users for the sake of day-to-day convenience. Scanning overheads and any interaction with updates, however, are likely to be considerably less irritating than a never-ending tirade of noisy alert bubbles.

Other aspects of the installation showed that integration was occurring with the security features of *Windows XP*. Although available in earlier versions of *Windows XP*, the Internet Connection Firewall (ICF) is not activated by default until *Service Pack 2*. The fact that this was activated was noted during installation of *PCCIS* as a potential conflict. The conflict mentioned was only one of possible slowing of data transmission and thus for much of the testing period both ICF and the *PCCIS* firewall were operating. This did result in an almost imperceptible warning appearing on the Windows Security Center page stating that it was preferable to operate only one firewall. When ICF was disabled, *Trend*'s firewall component was recognised as being in operation and no warnings resulted.

CONCLUSION

Whenever more than one anti-virus specialist is gathered in a room there is much talk concerning blended threats and the blurring of general security and the traditional enclaves of anti-virus technology. That this is now considered to have reached the mainstream is apparent in this latest offering from *Trend Micro*. The addition of anti-spam, anti-spyware, anti-Trojan and security scanning functionality to the product is not much of an innovation – though there are additional features which certainly are innovative. What is more noteworthy, in some ways, is that these extra features are included in a product which is deliberately intended to be easy to use. *Trend* clearly considers that its users will consider these features to be valuable additions and have sufficient knowledge to appreciate them.

The idea that general users are aware of the importance of security may be somewhat optimistic but, one hopes, it is a sign that the future may be easier than the recent past. If users are becoming more educated, some of the major flaws in human behaviour upon which viruses have relied, should begin to decline. I am tempted to hope that *Trend*'s market researchers have judged their customers correctly.

Technical Details

Test environment: Identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive running *Windows XP Professional*. Athlon XP1600+ machine with 1 GB RAM, 80 MB hard disk, DVD/CD-ROM and ADSL Internet connection running *Windows XP Professional Service Pack 2*.

Developer: *Trend Micro Inc.*, 10101 N. De Anza Blvd, Cupertino, CA 95014, USA; Telephone +1 877 268 4847; email sales@trendmicro.com; website http://www.antivirus.com/.

ERRATA – WINDOWS SERVER 2003 COMPARATIVE REVIEW

VB regrets that three mistakes crept into the Comparative review published in the November issue:

- The version number for *Sophos Anti-Virus* should have read 3.86, not 3.83 as published.
- The values for *CAT Quickheal* in the standard on-demand test set should read 'Misses: 169, Detection 92.91%' in all occurrences (the on access results were erroneously duplicated).
- *Norman Virus Control* did not reproducibly miss detection of any samples in the In the Wild test set and thus is due a VB 100% award.

Nov 2004
VIRUS BULLETIN
100%
www.virusbtn.com

VB apologises for the errors.

CALL FOR PAPERS

VB2005 DUBLIN

Virus Bulletin is seeking submissions from those wishing to present at VB2005, the Fifteenth Virus Bulletin International Conference, which will take place 5–7 October 2005 at The Burlington, Dublin, Ireland.

The conference will include two full days of 40-minute presentations running in concurrent streams: Technical AV, Corporate AV and Spam (both technical and corporate).

Submissions are invited on all subjects relevant to the anti-virus and anti-spam arenas. *VB* welcomes the submission of papers that will provide delegates with ideas, advice and/or practical techniques, and encourages presentations that include practical demonstrations of techniques or new technologies.

SUGGESTED TOPICS

The following is a list of suggested topics elicited from attendees at VB2004. *Please note that this list is* not *exhaustive, and papers on these and any other AV and spam-related subjects will be considered.*

TECHNICAL AV

- Honeypots
- Longhorn
- Threats and protection for mobile devices
- Emulation, engine level sandboxing, unpacking and other static code analysis
- Wireless security
- Rootkits
- ZA64/AMD64 viruses
- Emulators/heuristics/PE unpacking on non-*Windows* platforms
- Neural networks
- HTML exploits
- Email encoding
- Viruses in new formats
- Hardware anti-virus solutions
- Tools of the trade (deobfuscation, IR, etc.)
- Behavioural analysis and detection

CORPORATE AV

- How to raise public awareness of malware/cybercrime
- Use of personal firewalls with anti-virus protection
- Corporate patch management
- Phishing
- Spyware and adware detection – legal issues
- Corporate anti-virus/spyware/adware case studies
- Zero day virus infections in a corporate environment
- Malware and the law
- Anti-virus performance and quality testing
- Management of anti-virus infrastructures
- Proactive detection mechanisms
- IDS/IPS
- False positive prevention
- Vulnerability management
- Government security policies
- IT outsourcing and associated risks
- Integration of protection technology on the desktop: anti-virus/firewall/IDS/spyware
- Blackhat view of malware
- Demonstrations of threats
- IM threats
- Anti-virus and anti-spam managed services

SPAM

- Corporate anti-spam case studies
- Spam tricks
- How spammers operate
- Spam from a legal point of view
- Anti-spam performance testing
- Mobile spam
- New anti-spam techniques

HOW TO SUBMIT A PAPER

Abstracts of approximately 200 words must reach the Editor of *Virus Bulletin* no later than **Thursday 10 March 2005**. Submissions received after this date will not be considered. Abstracts should be sent as RTF or plain text files to editor@virusbtn.com. Please include full contact details with each submission.

Following the close of the call for papers all submissions will be anonymised before being reviewed by a selection committee; authors will be notified of the status of their paper by email.

Authors are advised in advance that, should their paper be selected for the conference programme, the deadline for submission of the completed papers will be Monday 6 June 2005 and that full papers should not exceed 8,000 words. Further details of the paper submission and selection process are available at http://www.virusbtn.com/conference/.

END NOTES & NEWS

Infosec USA will be held 7–9 December 2004 in New York, NY, USA. For details see http://www.infosecurityevent.com/.

The SANS Cyber Defensive Initiative East takes place 7–14 December 2004 in Washington, D.C., USA. Focused training disciplines include security, legal, operations, managerial and audit. For more information see http://www.sans.org/.

Computer & Internet Crime 2005 will take place 24–25 January 2005 in London, UK. The conference and exhibition are dedicated solely to the problem of cyber crime and the associated threat to business, government and government agencies, public services and individuals. For more details and online registration see http://www.cic-exhibition.com/.

The 14th annual RSA Conference will be held 14–19 February 2005 at the Moscone Center in San Francisco, CA, USA. For more information, including online registration and the conference agenda, see http://www.rsaconference.com/.

The E-crime and Computer Evidence conference ECCE 2005 takes place at the Columbus Hotel in Monaco from 29–30 March 2005. ECCE 2005 will consider aspects of digital evidence in all types of criminal activity, including timelines, methods of evidence deposition, use of computers for court presentation, system vulnerabilities, crime prevention etc. For more details see http://www.ecce-conference.com/.

Black Hat Europe takes place in Amsterdam, The Netherlands, from 29 March to 1 April 2005. Black Hat Europe Training runs from 29 to 30 March, with the Black Hat Europe Briefings following, from 31 March until 1 April.

Black Hat Asia takes place 5–8 April 2005 in Singapore. In this case the Briefings take place 5–6 April, with the training on 7–8 April. A call for papers for the Black Hat Briefings (both Europe and Asia) closes on 15 January 2005. For details and registration see http://www.blackhat.com/.

The first Information Security Practice and Experience Conference (ISPEC 2005) will be held 11–14 April 2005 in Singapore. ISPEC is intended to bring together researchers and practitioners to provide a confluence of new information security technologies, their applications and their integration with IT systems in various vertical sectors. For more information see http://ispec2005.i2r.a-star.edu.sg/.

Infosecurity Europe 2005 takes place 26–28 April 2005 in London, UK. Now in its tenth year, the exhibition will have over 250 exhibitors and its organisers anticipate over 10,000 visitors. See http://www.infosec.co.uk/.

The 14th EICAR conference will take place from 30 April to 3 May 2005 in Saint Julians, Malta. Authors are invited to submit papers for the conference. The deadlines for submissions are as follows: academic papers 14 January 2005; poster presentations 18 February 2005. For full details of the conference see http://conference.eicar.org/.

The sixth National Information Security Conference (NISC 6) will be held 18–20 May 2005 at the St Andrews Bay Golf Resort and Spa, Scotland. For details of the agenda (which includes a complimentary round of golf at the close of the conference) or to register online, see http://www.nisc.org.uk/.

The third International Workshop on Security in Information Systems, WOSIS-2005, takes place 24–25 May 2005 in Miami, USA. For full details see http://www.iceis.org/.

NetSec 2005 will be held 13–15 June 2005 in Scottsdale AZ, USA. The program covers a broad array of topics, including awareness, privacy, policies, wireless security, VPNs, remote access, Internet security and more. See http://www.gocsi.com/events/netsec.jhtml.

Black Hat USA takes place 23–28 July 2005 in Las Vegas, NV, USA. A call for papers will open 15 April 2005. For more details see http://www.blackhat.com/.

The 15th Virus Bulletin International Conference, VB2005, will take place 5–7 October 2005 in Dublin, Ireland. For conference registration, sponsorship and exhibition information and details of how to submit a paper see http://www.virusbtn.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$310)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire OX14 3YP, England
Tel: +44 (0)1235 555139 Fax: +44 (0)1235 531889
Email: editorial@virusbtn.com www.virusbtn.com

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

vb Spam supplement

CONTENTS

NEWS & EVENTS

NEWS BY EMAIL

Last month saw the inaugural issue of a twice-monthly email newsletter dedicated to news and technical information about the spam and anti-spam arena. The newsletter, produced by *POPFile* and *Spammers' Compendium* author John Graham-Cumming, is a vendor-neutral, technical newsletter aimed at those who wish to keep informed about the latest in spam and anti-spam. As well as news and events, each newsletter will include a technical article on spam or anti-spam techniques. Those interested in receiving the newsletter should visit http://www.jgc.org/ to subscribe.

SAVE YOUR SOUL WITH SPAM

We are all accustomed to receiving spam that advertises herbal medicines, designer watches, new mortgages and online degrees – and we are even used to seeing spam that promises 'quickie' ordinations for aspiring ministers – but *MessageLabs* has reported an influx of spam that goes one step further and offers the recipient spiritual salvation.

According to researchers at *MessageLabs* 'spiritual spam' may not push a product of any kind, but may simply urge recipients to accept God into their lives. Some of the messages even provide a prayer that 'can save you or someone you love'. Perhaps a prayer that can help you to stop spam would be a useful addition.

HOTMAIL ORIGINATOR TURNS TO ANTI-SPAM

Sabeer Bhatia, the man behind *Hotmail*, has announced that he has invested in an anti-spam company. Bhatia, who developed the concept of web-based email in his mid-20s, and subsequently sold *Hotmail* to *Microsoft* for a reported $400 million, revealed last month that he had turned his attention to the problem of spam. Although reluctant to give away any details of his anti-spam project, he told an audience of Australian technology entrepreneurs that he considers there to be 'intelligent solutions [to the spam problem] by putting an appliance at a network level'.

EMAIL AUTHENTICATION IN THE OPEN

Last month an open letter was sent to members of the US Federal Trade Commission (FTC), calling for a rapid rollout of email authentication technologies. The letter was signed by 35 high-profile organisations including *Amazon.com*, the Anti-Phishing Working Group, the Bank of America, *CipherTrust*, *Cisco Systems*, *EarthLink*, *eBay*, the Email Service Provider Coalition (ESPC), *IronPort Systems*, *Microsoft*, *Sendmail*, *Symantec* and *VeriSign*, and was sent ahead of an email authentication summit held by the FTC and the US National Institute of Standards and Technology.

The signatories of the letter pledged their support for email authentication standards, saying: 'We stand united in our fight against spam and phishing and in the support of email authentication standards. We are committed to deploy[ing] the Sender ID Framework by publishing our records and advance signing technologies such as *Cisco*'s Identified Internet Mail and *Yahoo*'s Domain Keys which can be rapidly deployed to meet the needs of consumers and enterprises worldwide.' The full letter can be read at http://truste.org/about/sender_id_industry_letter.php.

The authentication summit itself provided little in the way of agreement on standards. Pavni Diwanji, of *MailFrontier* said, 'We'll be lucky if we solve 50 per cent of the problem [with email authentication].' However, there was general agreement among participants that email authentication is an essential first step toward an anti-spam solution.

MOST-SPAMMED SEES END IN SIGHT

The world's most spammed email recipient, Bill Gates, said last month that he hopes to have the spam problem under control within two years. However, this is not the first time Gates has quoted this time frame – ten months ago the *VB* website reported that Bill Gates had ‘explained that spam would not be a problem in two years’ time, due to *Microsoft*’s movements on sender-pays email’ (see http://www.virusbtn.com/news/virus_news/2004/02_03.xml). It was revealed last month that, with an average of four million incoming email messages per day, Bill Gates ranks as the world’s most spammed email recipient [*and we thought we’d got it bad! - Ed*]. There’s little reason to feel sorry for Gates though, since he has an entire team dedicated to dealing with his emails.

EVENTS

Anti-Spam Asia takes place in Shanghai, China, on 7 December 2004. The seminar will feature a number of speakers including representatives from Shanghai Computer Anti-Virus Center, anti-spam and anti-virus company *Sophos*, and Hong Kong University of Science and Technology. For more information or to register email wendy@phangnaughton.com.

The ISIPP’s National ‘Spam and the Law’ Conference will be held on 28 January 2005 in San Francisco, CA, USA. Subject areas covered by the conference include the current state of the law regarding spam, the laws under which you operate when you send email, and the rights and obligations of email recipients. The conference is aimed at anybody involved in the email sending or email receiving industries, including email service bureaus, email service providers, ISPs and online marketing agencies. For details see http://www.isipp.com/.

The 2005 Spam Conference will be held in Cambridge, MA, USA in early to mid January 2005 (date yet to be confirmed). As in previous years the intensive, one-day conference will comprise a succession of quick (20-minute), concentrated talks, with attendees going out for dinner together in the evening. See http://spamconference.org/.

The International Quality and Productivity Center (IQPC) will hold a two-day conference on managing and securing corporate email from 1–2 February 2005 in Las Vegas, NV, USA. The conference will include case studies on tackling spam, viruses, compliance issues and instant messaging. For more details see http://www.iqpc.com/.

The Second Conference on Email and Anti-Spam (CEAS 2005) will be held in summer 2005 (date and venue yet to be announced). A low-volume mailing list has been set up for CEAS conference-related announcements – sign up by sending a message with the body ‘subscribe ceas-announce’ to majordomo@lists.stanford.edu. Information will also be posted on http://www.ceas.cc/.

FEATURE

A RIVAL FOR ONLINE SPAM?

John Clark
TeleCommunication Systems

After a difficult period, the mobile phone industry is booming once again. It has been reported (by *EMC*) that one sixth of the world’s population owns a mobile handset – which accounts for one billion people. This figure is set to double by 2006.

According to the *Mobile Data Association* an average of 55 million text messages (or Short Message Service, SMS) per day were sent across the UK’s GSM networks in August 2004. But, while millions of people are using SMS to communicate with their friends, buy data services or even vote for their favourite contestants on TV talent shows, an increasing proportion of the total number of messages sent is made up of unsolicited commercial SMS messages, or spam.

A study by British technology firm *Empower Interactive* revealed that 65 per cent of European mobile phone users receive at least five spam SMS messages a week. In addition, complaints about spam are increasing: in the UK the number of complaints about unsolicited SMS advertisements grew from 65 in 2002 to 393 in 2003. The Advertising Standards Authority recorded only six such complaints in 2001.

All this suggests that we need to find a solution to guard against the onslaught of mobile spam – mobile phones are beginning to swarm with unwanted messages, and we might end up with another spam epidemic on our hands.

MOBILE SPAM AND NETWORK SPAM

The spam that affects mobile users and operators falls into two categories. The first, ‘mobile spam’, refers to unwanted or unsolicited messages received on wireless devices and handsets. The second, ‘network spam’, refers to multiple messages sent to solicit or harass a carrier’s subscribers and intended to have a negative impact on an operator’s network. Spam can take the form of solicitations, harassments, scams or frauds.

As with any new phenomenon, it is hard to predict the progression of the problem or what the ramifications will be. An obvious parallel for mobile spam is email spam – although there is still some question as to whether we should regard the Internet industry as a direct case study for the mobile world.

In August 2004, *MessageLabs* reported that 68 per cent of all emails sent were spam. Spam is now regarded as the principal issue of email subscriber discontent, it has a negative effect on productivity in the workplace, and has led to legislation that can bring criminal prosecution against the spammer in a number of countries.

RECOGNITION

Mobile operators in Europe have recognised that, while mobile spam is annoying to the end user, it (currently) represents a small percentage of the total number of text messages sent. However, the subset of spam messages known as scams (fraudulent messages offering prizes and holiday deals if the recipient responds to a premium rate text or dials a premium rate number) has caused more concern among subscribers and is an issue that operators are looking to address via business agreements and legislation, in addition to technical solutions.

The concept of spam in the mobile arena has been acknowledged and steps are being taken to ensure that the issue is not ignored. The EU Directive on Privacy and Electronic Communications, which came into force on 31 October 2003, ruled that any company wishing to send an electronic communication (including SMS) that originates in the EU must have the permission of the recipient in order to do so. The recipient must ‘opt in’ unless they already receive communications from the source – in which case the recipient can opt out if they no longer wish to receive messages from that source.

However, loopholes have been found in the legislation – as with email spammers, organisations sending SMS spam and scam from outside the EU are unaffected by the directive. This means that it must fall to the EU’s mobile operators to control the spam and scams being sent to their subscribers.

While mobile spam may not ever become as prevalent as it has become in the Internet world, mobile devices are such personal items that an influx of even a small percentage of spam will not be tolerated by the users. Subscribers will hold their mobile operators accountable, which will have a serious negative impact for the operator, both financially and in terms of subscriber loyalty.

LESSONS FROM THE INTERNET

Scams are currently the main issue which mobile operators are looking to address in Europe, but as handsets become increasingly advanced and the ability to surf the Internet via mobile devices becomes increasingly widespread, operators and mobile device manufacturers will have to view spam as an ongoing threat to the industry. Mobile operators need to learn from both the successes and the mistakes made in the Internet arena.

In addition, operators will need to look at how to address what is the largest revenue driver in the Internet industry, the adult content business. Operators must tread a fine line between providing their subscribers with the services they ask for, and protecting minors and those who do not wish to subscribe to the adult content services. Status quo subscriber opt-in mechanisms will not be sufficient in this scenario. A more complex approach that takes into account a combination of business agreements, age verification solutions, filtering, blocking and parental handset security will need to be addressed.

FIRST STEPS

Mobile operators are taking steps to make it harder for spammers to reach the end user. In Europe, operators prefer the ‘closed garden’ network approach – there has been a tightening of commercial in-country and overseas roaming agreements between the national and international operators that now also cover data delivery. This has had the effect of making bulk SMS significantly more expensive to send across their networks. Additionally, operators will look to permanently block overseas operators/vendors that try to harvest their subscribers’ mobile numbers or send inter-carrier bulk spam via the SS7 signalling network.

Another approach to curbing mobile spam – which is more prevalent in the North American markets – is for operators to install anti-spam gateways at the edges of their networks. These can be configured to block, filter or quarantine any messages that are not from recognised sources or domains, messages that are being sent in bulk (higher than a predetermined volume), or that contain certain types of content.

LOOKING TO THE FUTURE

Mobile spam is a growing problem – and September this year saw the first confirmed reports of mobile phone virus SymbOS/Cabir (see *VB*, August 2004, p.4) in the wild.

With the increasing convergence of viruses and spam on the Internet, mobile operators should start thinking about the not-so-distant future. Viruses will become more widespread on mobile handsets and there is no doubt that they will become more sophisticated. A little more consideration is needed from the mobile operators, along with a stricter set of regulations to which they must conform – otherwise the mobile world may face a problem that has disastrous consequences for the industry.

SUMMARY

ASRG SUMMARY: NOVEMBER 2004

Helen Martin

November was a notably quiet month for the ASRG mailing list – perhaps because there was a physical meeting of ASRG members at the 61st IETF in Washington, D.C. The 62nd IETF meeting (at which there will almost certainly be another ASRG meeting) will take place 6–11 March 2005 in Minneapolis, MN, USA.

The month began with a discussion of MTA Mark, the extensible system for determining whether an MTA should be running on a given IP address (see *VB*, May 2004, p.S2). Peter J. Holzer pointed out that the MTA Mark system requires only a couple of TXT records and that no change of DNS or BIND is necessary. However, since his ISP does not delegate reverse DNS for /29 networks, Peter said he would be unable to implement MTA Mark for his home network – he would need his ISP to do this.

Douglas Campbell pointed out that having to rely on the controller of his address range to implement MTA Mark should not pose a problem, saying, 'I'd expect them to be eager to cut spam transiting their network.' Douglas said he would even be willing to pay a fee to allow his mailserver access to the network – although he acknowledged that the big spammers would look upon the maximum fee he would be willing to pay as 'chump change' (throwaway money).

Peter Holzer, who works as a sysadmin for WSR, the Austrian Computing Centre for Economics and Social Sciences, revealed that he had tested all 155,000+ of the IP addresses which connected to his organisation's MX during October 2004 for MTA Mark records. The results were as follows:

```
155,173 MTA = unknown
      8 MTA = no
      6 MTA = yes
```

While these results were more encouraging than a similar test Peter had carried out three months previously, he pointed out that they indicate clearly that, currently, there are not enough MTA Mark records to bother implementing filters on them.

Furthermore, Peter revealed that the MTA Mark = no records he encountered seemed a little suspicious – he suspected that in some of the cases someone had created an MTA Mark = no record for the whole network but had omitted to add the MTA Mark = yes records for the mail servers. Peter posted a link to the script he used to run the stats, so that anyone interested in carrying out their own tests could do so: http://www.hjp.at/mail/spam/mtamark/mtamark.pl.

Markus Stumpf revealed that, at his organisation, they have added MTA Mark records to around one hundred of their mailservers and that they perform greylisting, and disable greylisting in case there is an MTA Mark = yes record. He also pointed out that there is native support for MTA Mark in the current sendmail versions – and Markus said he hoped to have a qmail interface/patch ready very shortly.

Last year Kurt Magnusson put forward a proposal for an anti-spam system which used a blacklist comprising the contact details (telephone numbers and URLs) contained in spam messages. The method was criticised at the time, with group members suggesting that such a list could be tainted. However, for the last year and a half Kurt has been running a proof of concept system, using a simple shellscript with some grep lines, and has had encouraging results.

Kurt explained that he compiled his blacklist using both spam he received in his own inbox and Spamarchive.org repositories. He reported that, prior to 'the Rolex explosion', he was receiving an average of just two to three out of around 35 spams per day. He said, 'to cope with Rolex, I added SURBL DNS list to the [proof of concept], which thankfully decreased the post-Rolex hits from 20–30 out of 40–50 spams a day to six to eight passing in.'

John Levine's Internet draft 'DNS Based Blacklists and Whitelists for E-Mail' was added to the IETF's database for Internet drafts this month. The draft documents the structure and usage of DNS-based blacklists and whitelists, and the protocol used to query them – it can be found at http://www.ietf.org/internet-drafts/draft-irtf-asrg-dnsbl-01.txt.

Indeed it was a busy month for ASRG chair John Levine, as he was called as an expert witness for the prosecution in the criminal trial of prolific spammer Jeremy Jaynes (aka Gavin Stubberfield) and his sister Jessica DeGroot. The case was brought under Virginia's state anti-spam law, which is considered to be stronger than the Federal CAN-SPAM act. The Virginian law makes it a crime to send unsolicited bulk mail using forgery, meaning that the prosecutors had to prove not only that Jaynes sent lots of unsolicited mail, but that it was sent using forgery. John Levine was asked to testify as an expert on email technology and was questioned briefly by the defence lawyers about his involvement with the ASRG. Although the trial progressed slower than the prosecutor had expected, Jaynes and DeGroot were convicted of violating the anti-spam law. Jaynes was sentenced to nine years imprisonment, while DeGroot was fined $7,500. John's fascinating account of his involvement in the trial can be found at http://www.circleid.com/article/804_0_1_0_C/.