JULY 2005

virus BULLETIN

The International Publication on Computer Virus Prevention, Recognition and Removal

CONTENTS

IN THIS ISSUE

CHOOSE YOUR WEAPOX

No doubt to the surprise of some in the *Macintosh* community, the *MacOS X* platform has a rootkit. Peter Ferrie provides all the details of OSX/Weapox.
page 4

ON YOUR GUARD

Incidents of malicious applications that steal financial account information have increased dramatically over the last year. Candid Wueest demonstrates that the biggest threat to online banking is malicious code executed carelessly on the end-user's computer, and advises users of online banking to tread carefully.
page 6

AN ENEMY DIVIDED

Brian McWilliams explains why, despite the recent outbreak of Sober.Q, which showered the Internet with neo-Nazi propaganda emails, he disagrees with the notion that virus writers and spammers are in cahoots.
page 8

vb Spam supplement

In the *VB Spam Supplement* this month: anti-spam news & events and how free software Mail Avenger can be used to block unwanted mail.

ISSN 0956-9979

'With any great invention, there is always a flip side just waiting to be exploited and the Internet has proved no exception.'

Matt Peachy
IronPort Systems

ADAPT OR DIE

William Caxton introduced the printing press to England in the middle ages for the sole purpose of circulating literature to the masses, but it didn't take long before society began abusing this medium and using it to generate negative material. With any great invention, there is always a flip side just waiting to be exploited and the Internet has proved no exception.

First used as a tool by academics to carry out research and communicate with peers, the Internet is now used by spammers and virus writers to create havoc and cause chaos. According to *Ferris Research*, 70 per cent of all email traffic is now spam.

As users have deployed IT security defences to safeguard themselves from junk mail, spammers have upped their game and developed more sophisticated techniques to get around these barriers.

Just a few years ago, virus writers and spammers were two distinct groups with distinct agendas. In recent years, however, the two groups have come together as spammers have turned to the more technically adept virus writers for help. Spammers started paying virus writers to write viruses that would leave behind zombie machines – which could then be used to send spam so that the messages appear to come from a legitimate server. [*For a different take on the idea that spammers are converging with virus writers, see p.8 - Ed*]

IronPort's SenderBase Network monitors global email traffic patterns and determined that at the beginning of 2004, less than 30 per cent of spam was coming from infected zombie PCs, but by the end of 2004 this figure had jumped to more than 70 per cent.

This year will see more potent viruses appear that are designed to deliver more zombies to send ever more spam. In addition, we are seeing an increase in online fraud or phishing. Last year, analyst firm *Gartner* estimated that 57 million Americans received phishing emails and that two million US adults gave sensitive information to phishers. [*See p.6 for an in-depth look at the threats to online banking - Ed*]

I believe that it will get better though. As government enforcers pursue legal remedies, the industry has been hard at work creating a new generation of filtering technologies designed to identify and discard spam before it gets into the recipient's inbox.

Effective systems will contain a blacklist or database of known spammer addresses which can be used to cross-check messages at the gateway server. Likewise, a whitelist detailing known or trusted senders can be used to ensure that legitimate emails get through. Authentication has been coined the new white hope of email security and we will see a growing demand for it due to the increasingly sophisticated means by which spammers attempt to hide their identity.

The industry is also starting to look at things like reputation, introducing filters which control and quarantine traffic proactively from suspicious or unknown senders. Such appliances perform a threat assessment of inbound and outbound messages using a threat scale scoring system. When the score is elevated, all mail is filtered and suspicious messages are quarantined until updated signatures are in place.

It is crucial that companies don't rely entirely on signature-based filters because potentially they can create a gap in the action that needs to be taken. It's all about pre-patch management, using a system that can monitor global activity to detect an early stage outbreak and change filtering policy automatically to prevent viruses getting onto the network.

The Internet is not the first life-changing invention to be exploited by humans for personal gain, and it certainly won't be the last. What is important is that companies are ready for what spammers and virus writers throw at them. Without the right technology in place, it will be one bumpy ride.

NEWS

SYMANTEC TAKES INITIATIVE IN ADWARE LAWSUIT

Symantec has taken the unusual step of filing a pre-emptive lawsuit against marketing firm *Hotbar.com*. Rather than seeking damages from the company, *Symantec* is simply requesting a ruling from the Californian court that will allow it to label certain programs produced by *Hotbar.com* as adware.

The detection of adware is something of a legal minefield for security vendors, since this type of program falls into the 'grey' area of programs that are not strictly malicious, but which many customers want removed from their systems. Many vendors have found themselves at the wrong end of lawsuits filed by disgruntled companies whose programs their products have detected and removed.

In this case, *Symantec* chose to take matters into its own hands after being threatened with litigation by *Hotbar*. According to the suit, *Hotbar* first contacted *Symantec* in July 2004, asking the vendor to remove all *Hotbar* programs from its detection database. *Symantec* says that this prompted it to review its classification of the programs, but that after some consideration it decided to continue labelling them as adware. *Symantec* subsequently received five threats of litigation from *Hotbar* and chose to take pre-emptive legal action.

With many security vendors still treading a very cautious path between meeting customer demands and avoiding litigation, the outcome of this case could be a significant step towards clarifying what vendors should and should not classify as adware.

MICROSOFT ANNOUNCES PLANS FOR SYBARI

Microsoft completed its acquisition of security firm *Sybari Software* last month and announced its plans for *Sybari*'s products. *Microsoft* intends to discontinue sales of the company's anti-virus solution (*Antigen*) for Unix and *Linux* platforms, while its other products, including *Antigen for Domino* (on *Windows*), will continue to be marketed. *Microsoft* has said that it will continue to offer support to existing users of *Sybari* products on both *Windows* and non-*Windows* platforms, and that *Sybari*'s existing pricing and licensing model will be retained. It is also likely that *Microsoft* will add its own scanning engine to those that can be used with the multiple-engine *Antigen* product.

With *Microsoft*'s desktop anti-virus product for the consumer market, *OneCare*, scheduled for beta release later this year, the company finally looks to be making serious inroads into the anti-virus market.

(ATTEMPTING AN) ALTERED IMAGE

Computer Associates is considering changing its name in a bid to rid itself of the negative perceptions that have resulted from its highly publicised accounting scandal.

In September last year the company's former chairman and CEO was charged with securities fraud, conspiracy and obstruction of justice in connection with the multi-billion dollar accounting scandal. The company agreed to pay $225 million to shareholders as part of a settlement that would allow it to defer criminal prosecution and that would settle securities fraud charges brought by the Securities and Exchange Commission. Nine months on, the company is set to launch a $7 million promotional campaign in an attempt to shake off negative images, and says that it has considered adopting '*CA*' as its formal name, as well as redesigning the company's logo.

VIRUS REVEALS POWER PLANT DATA

Confidential information about Japanese nuclear power plants was leaked last month as the result of a virus-infected PC. According to Japanese news reports the personal PC of an employee of *Mitsubishi Electric Plant Engineering* – the contractor responsible for the inspection and maintenance of equipment at numerous Japanese power plants – was infected by an unnamed virus that sends data through the *Winny* file-sharing software. As a result, a substantial amount of confidential data was leaked to users of the *Winny* peer-to-peer file-sharing system, including power plant inspection reports, a repair manual, the names of inspection workers and photographs of the interior of the power plants.

Japan's chief cabinet secretary Hiroyuki Hosoda said that the government believed the leak did not involve any information that would pose a risk to the protection of nuclear materials.

EAGLE EYES

Readers with a keen eye will notice that there has been a significant increase in the number of incidents reported in *VB*'s prevalence table (which, for this month only, has been re-homed on p.4). This is thanks to two new contributors coming on board.

VB is always looking to increase the scope of the prevalence table. If you have regular monthly access to virus statistics and are willing to submit your figures to *VB*, please email Matthew Ham (matthew.ham@virusbtn.com) to discuss becoming a contributor. Figures are submitted in confidence, and will only be used in the statistics presented in the prevalence table.

Prevalence Table – May 2005			
Virus	Type	Incidents	Reports
Win32/Sober	File	733,358	86.66%
Win32/Mytob	File	46,589	5.51%
Win32/Netsky	File	29,575	3.50%
Win32/Mydoom	File	16,318	1.93%
Win32/Bagle	File	5,371	0.63%
Win32/Lovgate	File	4,987	0.59%
Win32/Bagz	File	1,466	0.17%
Win32/Zafi	File	1,309	0.15%
Win32/Mabutu	File	1,227	0.15%
Win32/Funlove	File	983	0.12%
Win32/Bugbear	File	863	0.10%
Win32/Klez	File	485	0.06%
Win32/Mimail	File	381	0.05%
Win32/Dumaru	File	355	0.04%
Win32/Gibe	File	344	0.04%
Win32/Wurmark	File	324	0.04%
Win32/Pate	File	288	0.03%
Win32/Valla	File	248	0.03%
Win32/Mylife	File	152	0.02%
Win32/Eyeveg	File	151	0.02%
Win32/SirCam	File	142	0.02%
Win32/Swen	File	126	0.01%
Win32/Sobig	File	105	0.01%
Redlof	Script	104	0.01%
Win95/Tenrobot	File	104	0.01%
Win32/MyWife	File	71	0.01%
Win95/Spaces	File	70	0.01%
Win32/Mota	File	69	0.01%
Win32/Fizzer	File	58	0.01%
Win32/Yaha	File	54	0.01%
Win32/Kriz	File	36	0.00%
Win32/Maslan	File	31	0.00%
Others[1]		458	0.05%
Total	846,202		100%

[1]The Prevalence Table includes a total of 458 reports across 63 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

VIRUS ANALYSIS

GOT [MAC]ROOT?

Peter Ferrie
Symantec Security Response, USA

There is a long history of rootkits on Unix-based platforms, such as Unix itself, *Linux*, *BSD*, etc. No doubt to the surprise of some in the *Macintosh* community, the *MacOS X* platform now has one too. We call it OSX/Weapox. It is written by someone who calls himself 'nemo'.

Weapox is based very heavily on the AdoreBSD rootkit. In fact, some of the original Adore code remains in the Weapox binaries, although it is never called. Even the function names have been retained (since the *MacOS X* kernel-extension file format is an object file, a lot of textual information is visible, including the function names). Since the *MacOS X* platform essentially has *BSD* at its heart, it was not a surprise that a *BSD* rootkit was used as the basis for a *MacOS X* rootkit. Weapox did not load on a test machine running *Jaguar* (*MacOS 10.2*), but it did load on a test machine running *Panther* (*MacOS 10.3*).

Whenever the rootkit is executed, it begins by hooking the functions 'setuid', 'kill', 'write' and 'chmod'. The rootkit also contains hook functions for 'writev', and 'getdirentries', but since those functions are never hooked, the hook functions are never called. In any case, those hook functions seem to be incomplete, even though they are fully functional in the AdoreBSD rootkit. Perhaps *MacOS X* is sufficiently different that the Weapox author couldn't get them to work properly. Other functions that are not called are 'activate_cloaking' and 'hide_process'. The latter is another remnant from the original AdoreBSD rootkit.

EATS, ROOTS AND LEAVES

The hooked 'setuid' function checks if the UID is set to a particular value (1337 – 'leet'). If it is, then the function sets the UID to 0 (root) instead. Otherwise, the function calls the original handler. (Leetspeak [or 13375p34k] is a cryptic form of transliteration adopted by some hackers and gamers as a way of excluding the non-leet from their conversations on open channels. 1337 is devolved from the word 'elite' via 'lite' -> 'leet' -> '1337'.)

KILL OR BE KILLED

Despite its name, not only can the 'kill' function terminate a process, but it can also signal a process. If the hooked 'kill' function is used to signal a process, then the function checks if the signal is one of two particular values. If the value is 1337 ('leet' again), then the function escalates that process's privileges to level 0, effectively giving it full control over

the system. If the value is 9047 (which, in a less common use of 1337, means 'PORT'), then the passed PID is interpreted as a port number, and the function prepends that port to a list of ports to hide. That list is used by the hooked 'write' function (see below). If the signal is neither of these special values, then the hooked 'kill' function calls the original handler.

The hooked 'chmod' function checks whether the mode to set is a particular value (378, a value that would not normally be allowed). If it is 378, the passed path is interpreted as a logged-on username, and the function prepends that username to a list of usernames to hide. This list is used by the hooked 'write' function. Otherwise, the function calls the original handler.

This signalling method is interesting in that the hooked functions are in no way related to the information that they hide – but perhaps that's the idea.

WRITE YOUR OWN TICKET

The hooked 'write' function checks the name of the application requesting the write. If the requesting application is 'netstat', and if any entry in the hidden port list appears anywhere within the text to be printed, then the entire line is discarded instead of being printed. If the requesting application is 'w' or 'who', and if any entry in the hidden user list appears anywhere within the text to be printed, the entire line is discarded. This is a very simple method of stealth, which can be defeated, for example, by renaming the application, but it works well enough against the average user. It is also the method that AdoreBSD used.

The hooked 'writev' function checks if the text 'promiscuous mode' appears anywhere within the text to be printed. If it does, then the entire line is discarded instead of being printed. Otherwise, the function calls the original handler. It is not clear why this text would be ignored, unless perhaps it would otherwise appear in a network security log. In the AdoreBSD rootkit, the function is used as an 'I'm here' routine – literally, if the hooked 'writev' function is called and if the text 'promiscuous mode' appears anywhere within the text to be printed, AdoreBSD prints 'I'm here'. Otherwise, the function calls the original handler.

The hooked 'getdirentries' function contains several bugs, one of which results in an infinite loop, but that doesn't matter since the function is never called. The function is intended to check for a particular directory name within a directory structure, presumably to avoid it being printed. However, the directory name is never copied to the buffer to compare. Even if the name matched, the function simply prints 'MATCH!' and does nothing further with it. Additionally, the original function is not called afterwards.

The AdoreBSD code, for comparison, calls the original function to retrieve the real list of directory entries, then removes the directory to hide from that list, before returning the list to the caller.

The 'active_cloaking' function is intended to remove the current module from the kernel module list. This list is used by, for example, kextstat. It is not clear, though, if the removal from the list results in the process no longer being executed. Perhaps that is why it is not called. The 'hide_process' function doesn't hide anything at all. It simply searches for the requested PID and returns success or failure. The rest of the code that was present in the corresponding AdoreBSD rootkit function has been removed from this function.

OPEN SESAME

A 'rootkit' called SH/Renepo ('opener' spelled backwards) preceded Weapox by a few months. (In fact it was less of a rootkit and more a collection of hacking tools.) The package contained a script and three binaries. The script displayed some messages, including user information and passwords. It added a new user to the system, turned off the firewall, and attempted to terminate the LittleSnitch process (a program that tells the user when a program is attempting to send information to the Internet). It downloaded and installed the rest of the package, started a backdoor, created a screen dump, and deleted its temporary file.

The first binary was a 'backdoor' for the xinetd program. It simply ran xinetd with a custom configuration file that caused xinetd to listen on port 31337 ('eleet', surprise!), and return a command-shell with root privileges on connection. The second binary was the well-known netcat program, a very useful tool for doing all kinds of network-related things. The third binary was a *Unix* log-file cleaner. There was no active stealth technology at all in the package.

CONCLUSION

There's always one person who spoils it for everyone else. The *Macintosh* community has been relatively unaffected by recent malware, at least when compared to the *Windows* community, but perhaps that is set to change after all.

OSX/Weapox	
Size:	27,608 bytes
Type:	Rootkit
Payload:	None
Removal:	kextunload, then delete the files.

FEATURE 1

THREATS TO ONLINE BANKING

Candid Wueest
Symantec Security Response, Ireland

A Miami businessman is suing his bank for the loss of $90,000. He claims that, in February 2005, this money was stolen from his online bank account via an unauthorized transaction. Investigations have revealed that the businessman's computer was infected with a Trojan capable of logging keystrokes, including his full account details. It is likely that the theft of this information was the trigger that led to the unapproved transaction to a foreign bank account. So far, the businessman's bank has refused to compensate for his loss [1].

This fraud case is not an isolated incident. The incidents of malicious applications that steal financial account information have increased dramatically over the last year, often resulting in victims losing hard currency. In May 2003, only around 20 such Trojans existed in the wild. Two years later, the number is reported to have increased to more than 1,300 [2]. Why?

There are several factors that may have influenced the evolution of this type of malware, but maybe the dramatic increase in their prevalence is just because they have a higher chance of succeeding than expected. The case of the businessman in Miami is just one example of many that have succeeded.

EVOLUTION

In the early stages, Trojans that steal financial account information targeted only a handful of online banks. For example, in August 2003, PWSteal.Bancos.B stole account information from only five banks.

This has changed dramatically. PWSteal.Bancos.T, discovered in April 2005, contains a list of 2,764 URLs from 59 different top-level domains. The corresponding organisations range from small local bank branches to international banking groups.

In February 2005, a Trojan named PWSteal.Goldun.B was discovered stealing account information for an online payment service called *e-gold*. The Trojan disguised itself as a security update for *e-gold*. When a user executed the deceptively-named file 'SecurityEgold.exe', the Trojan registered itself as a browser helper object (BHO) and monitored *Internet Explorer* for visits to pre-defined URLs.

Any account information that was gathered by the Trojan was posted via a PHP script on a domain controlled by the attacker. The log file on the server storing the data was accessible by anyone, most likely due to a misconfigured web server. A quick look at this log file showed a growing list of account numbers and corresponding passwords. Within an hour, the PHP script had added another 13 valid-looking account credentials. The site was online for another 24 hours before being shut down.

Interpolation of this data leads to the conclusion that the PWSteal.Goldun.B attacker had received details for a large number of accounts, providing him with the opportunity to steal hard currency from the victims.

The attack vectors used by this kind of malware can be categorised in two groups: local and remote attacks. Local attacks happen on the local computer during an online banking session. Remote attacks do not execute code on the local computer, but redirect the victim to a remote site.

LOCAL ATTACKS

A common mistake made by end users is believing that their online banking session is perfectly safe when they use an SSL connection.

Security experts state repeatedly that everything is safe if there is a yellow padlock symbol in the browser window. But SSL is designed as a secure tunnel from the end user computer to the bank mainframe and does not protect the end points such as the end user's computer.

The PWSteal.Bankash Trojan exploits this fact. The Trojan drops a DLL and registers its CLSID as a browser helper object in the registry. Thus the Trojan is able to intercept any information that is entered into a web page before it is encrypted by SSL and sent out.

This functionality can also be achieved by injecting the Trojan directly into the web browser's memory space, which also can often bypass desktop firewalls when making outgoing connections.

Other local attack methods include running a layered service provider (LSP) monitoring all network traffic, writing its own network driver, or displaying a carefully crafted copy of a website on top of the official website.

The PWSteal.Bancos family does the latter. When an infected user visits one of the pre-defined domain names, the Trojan generates a pop-up window which overlays the current browser window. The pop-up window contains an exact copy of the original service website login page. When information is entered into the fake form and the send button is pressed, the spoofed pop-up window closes, leaving the old browser window. Meanwhile, the harvested account credentials are sent to a remote server.

These are only some of the possible methods that will work even if the session is SSL encrypted. These procedures will

also bypass the virtual keyboard – a countermeasure that has been introduced by some online banking systems against key loggers. Here, the user clicks on a virtual keyboard displayed on the screen, rather than pressing the real keys on their keyboard, but this only shifts the problem: screen captures, fake website pop-ups and malicious code running inside the web browser can record exactly the same information as key loggers. No matter how the information is entered into the web form, once it is entered, it can be intercepted.

A further step towards better security therefore, would be the use of non-static user credentials. A user name and a static password are simply no longer enough to protect online banking sessions. Some companies have already responded to these threats by introducing dynamic passwords including RSA-secured ID tokens [3] or one-time passwords on paper lists called transaction numbers (TAN).

Unfortunately this does not solve the problem entirely. Since the method for entering authentication data has not changed, the password still can be intercepted. The only additional hurdle is that the attacker must use it first, before the legitimate user does.

This behaviour has not yet been observed in the wild, but consider the following scenario: a Trojan, intercepting a password by any of the discussed methods, simply has to send this information to the attacker. Meanwhile, it blocks every other connection from that computer. Thus the current online banking session is never completed. A network driver, LSP, BHO, or rootkit that hooks network API calls could do this. All the attacker needs to do is to use the unused, one-time password quickly to establish an online banking session from his own computer, enabling the attacker to do whatever he or she likes.

Some companies therefore ask for multiple TANs during a session, one for login and one for each transaction made. Others ask for a specific TAN on a list and the position is chosen at random each time. Still you do not achieve 100 per cent security.

One next step towards better security may be PKI (public key infrastructure) smartcards, which have already been introduced by some banks. These cards can be attached to the computer using a USB card reader and can act as a challenge-response authentication or use a zero-knowledge authentication, leaving the attacker with little useful information.

REMOTE ATTACKS

Usually, the attacker sets up a copy of the web page he wants to impersonate on a server he controls. In the past attackers often linked directly to the original images on the legitimate web server, which left easy-to-follow traces in the webmaster's log files. Nowadays, attackers tend to keep resources locally. Once the bait server has been set up, the attacker sends out emails that trick the user into visiting the spoofed website. These emails often prompt the user to visit the online service in order to provide some urgent data verification, or indicate that the user is required to visit the website because of some update process in the main database of the service provider.

This form of social engineering attack, with the goal of acquiring user account information, is also known as phishing.

The location of the real server is obfuscated or masked using exploits or by other, not so well-known methods. For example, one can translate the quartet of an IP address (such as 216.239.37.99) into a decimal number (such as 3639551331). Then a fake user authentication can be added that is made to look like the impersonated domain (http://mySecureBank.vb@3639551331). This trick fools some users into believing that they are clicking on a link that leads to the mySecureBank.vb domain. Instead, it goes to 3639551331, which is the IP address of Google.com represented as a decimal number.

The use of international domain names (IDN), introduced by ICANN in June 2003, adds a further complication to the matter of identifying URL obfuscation. The fact that international characters can be used in domain names raises the issue of domain names that have been spoofed simply by replacing some of the letters in their name with letters from different alphabets that look the same.

For example, an attacker could register the domain 'mySecureBank.vb', where the 'a' is replaced with an 'a' from the Cyrillic character set, which looks identical. If a hacker finds a domain-authenticated SSL service, he or she can even add an SSL padlock in an attempt to fool the end user, as demonstrated by Eric Johanson in 'The state of homograph attacks' [4].

Trojan.Blinder utilizes another example of obfuscation. Trojan.Blinder uses JavaScript to layer a white box over the location field of the browser hiding the fake URL. The box contains a spoofed URL that looks like a legitimate website. The position of the box is even recalculated and reapplied multiple times per second to ensure a seamless integration with the browser.

Furthermore, obfuscation is not even necessary. As seen in February 2005, large DNS poisoning attacks can lead to browser redirection, without even modifying the end user's computer directly. Once the user is on a spoofed website everything entered there can be captured.

JOINT FORCES

If an attacker combines local and remote attacks more serious damage can result. For example, a Trojan running on an infected computer can alter the local hosts file to redirect any requests for mySecureBank.vb to an IP address controlled by the attacker. This behaviour has already been observed in a number of adware threats in the wild.

To complete the illusion, the Trojan can also install a self-signed root certificate on the infected computer. Free tools like *OpenSSL* can be used to help create these certificates. This enables the attacker to generate official-looking SSL connections from the infected computer to the malicious web server hosting the spoofed website. The chances of an average user noticing these changes are very slim.

Once the user has been trapped on such a spoofed website, the attacker can act as man-in-the-middle and relay any challenge-response protocol that might be implemented by the original online banking system. At the moment we are not aware of a Trojan in the wild performing such an attack, but that does not mean that there couldn't already be one doing this. Such an attack could be countered by carefully checking the IP addresses involved in the session and their owners.

CONCLUSION

These examples show that the biggest threat to online banking is still malicious code executed carelessly on the end-user's computer. The attackers tend to target the weakest link. Once the attacker has control over a user's computer, he or she can modify the information flow to his or her advantage. This may have happened in the case of the businessman from Miami.

The situation most likely will not change until new transaction methods are introduced. So, whenever using an online financial system, ensure that your system is still under your control and not a spoofed puppet, or you could end up featuring as the businessman in the next fraud case article.

REFERENCES

[1] John Leyden, 'Florida man sues bank over $90k wire fraud', http://www.theregister.co.uk/2005/02/08/e-banking_trojan_lawsuit/.

[2] Joakim von Braun, *Symantec*, Sweden, personal communication.

[3] RSA SecurID Authentication, http://www.rsasecurity.com/node.asp?id=1156.

[4] Eric Johanson, 'The state of homograph attacks', http://www.shmoo.com/idn/homograph.txt.

FEATURE 2

SPAMMER README

Brian McWilliams
Independent writer, USA

The recent outbreak of the Sober.Q worm, which showered the Internet with neo-Nazi propaganda emails, is likely to reinforce the notion that virus writers and spammers are deeply in cahoots.

Many anti-virus and anti-spam providers, with the help of the computer press and even the mainstream media, have been warning that mercenary VX'ers are collaborating with spammers and are bent on turning unprotected PCs into unwitting accomplices in spamming. I helped perpetuate this notion in chapter ten of my book *Spam Kings* [1], where I discussed the rise of spam zombies and the SoBig worm.

But the real message of Sober.Q, contained in a small text file dropped by the worm, is quite different. The file, spammer.readme.txt, included hyperlinks to a May 2005 press release issued by a Californian email management firm. The press release warned that computers infected by Sober.S were 'being transformed into spambots'.

Beneath the links, the author of Sober.Q had written (in German), 'I am still not a spammer! But perhaps I should become one.'

The Californian company's press release was pure FUD (fear, uncertainty, and doubt), and it took a malware writer to say it like it is.

NO EVIDENCE

The fact is, there is no evidence that systems compromised by earlier versions of Sober have joined the legions of machines known as spam zombies. Aside from the author's own blasts of political 'spam' (and I use that term loosely), I have seen no proof that Sober-infected systems have joined the 'botnets' being used as proxies by commercial spam operations.

I would even go so far as to say that, in the wake of the big outbreaks of SoBig in 2003 [2] and Bagle in 2004, virus authors have discovered that there isn't much of a market for worm-infested spam zombies. Just look at Netsky, the biggest worm of 2004 [3], which has remained atop the *Virus Bulletin* prevalence table so far for 2005. Netsky and its variants employ numerous techniques for spreading. But ultimately the worm is all about propagation; it installs no backdoors or other code that could enable the author to access the victim computers remotely at a later date.

To be sure, a number of self-replicating malicious programs of late have installed remote-access code or proxy software.

For example, Zafi [4, 5] opens port 8181. MyDoom [6, 7] listens on ports in the range of 3127 to 3198. Bagle installed a backdoor port on 2745. But none of these worms have made a serious run for the top of the prevalence charts.

Hence, I have a hard time concluding, as did Joe St Sauver, the author of an otherwise stellar article about zombies [8], that 'the prime focus of many recent viruses is the conversion of end user hosts into spam zombies.'

SPAMMER HEAVEN

There's no doubt that a spammer's idea of heaven includes plentiful and freely available proxy computers. Routing spam through proxies helps junk emailers conceal their identity and makes them a tougher target for blacklisting. Almost all of the most popular spamware programs are designed to import lists of proxy computers – either in the form of a text file or a URL. By some estimates, up to 80 per cent of all spam currently emanates from proxies.

Gone are the days when spammers scanned the Internet manually for misconfigured SOCKS and other proxies on well-known ports such as 1080 and 3128. Spamware companies like *Send-Safe* still sell proxy scanners, but using them is a laborious process. Most impatient spammers (aren't they all?) simply visit one of the many websites, searchable via *Google*, that offer lists of open proxies.

However, anyone who's been in the spam business for any significant length of time has learned to bite the bullet and buy proxies from one of the many underground purveyors. Visit any 'bulk email' message board or Internet relay chat (IRC) channel for spammers, and you'll often see people selling proxies (or 'peas', as they're called).

Typically, proxies are rented by the week, with prices ranging widely. I've seen ads for 4,000 peas for as little as $50 per week, but proxies advertised as 'fast' or 'not beat up' can go for $600 per thousand per week.

ZOMBIES FROM ZOMBIES

It is tempting to assume that all these proxies are the work of money-grubbing worm and virus writers in the employ of spam kings. Conversely, some have voiced suspicions that SoBig was created by *Send-Safe*, in order to generate a ready pool of spam proxies. (Ruslan Ibragimov, owner of Russia-based *Send-Safe*, told me that a document published anonymously in 2004 wrongly accused him and his company of authoring SoBig.)

Talk to anyone who monitors botnets closely, and they'll tell you that spam zombies are usually created by other spam zombies, not by viruses or worms.

'It's a really nasty situation to watch,' says Andrew Kirch, one of the operators of the anti-spam Abusive Hosts Blocking List [9]. He has been known to sit in private IRC channels, gawking as newly compromised drones connect to the channel by the hundreds or thousands per day. Soon, the drones respond to orders to begin attacking other hosts, and report back to the channel any successful system compromises.

Zombie code – programs like rbot, sdbot, and phatbot – may capitalize on backdoors opened by worms. But Kirch says the botnet Trojans are much more effective at scanning and compromising new hosts. 'Worms such as MyDoom and others are pretty limited in functionality, despite all the hype about the open ports they leave behind,' said Kirch.

To test this assertion, I posed a hypothetical question to a handful of white-hat hacker acquaintances. 'If you were tasked with turning lots of *Windows* PCs into spam zombies ASAP,' I inquired, 'what method of attack would you choose?'

To my surprise, none of the security experts said they would release an email worm along the lines of SoBig or Bagle. Instead, they almost universally favoured using a browser exploit embedded in a web page.

'Stupid, impatient, greedy hackers use viruses and direct-spammed Trojans,' said Joe Stewart, security researcher with *LURHQ*, who authored a fascinating treatise on SoBig and spam [10]. 'Smarter, more long-term-thinking hackers use drive-by downloads,' he opined.

One good-guy hacker disagreed, saying the ideal method for building a spam zombie network would be to act like a zombie: scan the Internet and exploit (or 'own') any system found with *Windows* vulnerabilities. 'If you want stealth and have the patience, you will scan and own,' said Steve Manzuik, a security product manager with *eEye Digital Security*.

Indeed, stealth seems to be a key issue in assembling a large botnet that can be rented out to spammers. Dmitri Alperovitch, security researcher with *CipherTrust*, reminded me of the 'Warhol worm' discussions of a few years ago. 'If the goal is to get the [largest] number of machines in the fastest amount of time, the choice would definitely be an automated worm that is exploiting some particular popular vulnerability,' said Alperovitch.

But unlike some virus writers, the goal of botnet operators isn't to make headlines on *CNN* or *CNET*. According to Alperovitch, 'The more noise you create with it, the more likely you are to attract attention from both law enforcement and also volunteers and security companies.' That's why Alperovitch says a browser exploit is currently the best way to assemble a zombie army.

Experienced spammers know it's pointless to try to help themselves to some free proxies by scanning for zombies.

Botnet operators configure their zombies to listen on random, high-numbered ports (the Mitglieder Trojan, for example, creates proxies that listen on ports such as 35555 and 39999) and to 'phone home' to the zombie master. According to Stewart, botnet operators also take great pains to secure their zombies against takeover by others.

SCUM OF THE EARTH

Of course, it's certainly possible that new, less widely spreading worms designed to create spam proxies will appear, even if none of the most prevalent current email worms appear to have this goal in mind.

Then again, changes underway in the behaviour of botnets suggest that mercenary worm-writers will instead turn their focus to propagating spyware. Stewart reports that botnet operators are moving away from renting their zombies out as spam proxies, and instead are using the compromised machines to install adware and quietly rack up big commissions.

I don't claim to know what motivates virus writers in general or the author of Sober-Q in particular. But I think his little readme file was illuminating. Even a neo-Nazi malware creator apparently thinks spammers are the scum of the earth.

REFERENCES

[1] Brian McWilliams, *Spam Kings*, O'Reilly, 2004 ISBN 0-596-00732-9.

[2] Peter Ferrie, 'Sobig, sobigger, sobiggest', *Virus Bulletin*, October 2003, p.5.

[3] Mircea Ciubotariu, 'Netsky: conflict starter?', *Virus Bulletin*, May 2004, p.4.

[4] Gabor Szappanos and Tibor Marticsek, 'Patriot games', *Virus Bulletin*, July 2004, p.6.

[5] Gabor Szappanos, 'More patriot games', *Virus Bulletin*, August 2004, p.9.

[6] Gabor Szappanos, 'We're all doomed', *Virus Bulletin*, March 2004 p.9.

[7] Gabor Szappanos, 'Doomquest: life after Mydoom', *Virus Bulletin*, April 2004, p.8.

[8] Joe St Sauver, 'Spam zombies and inbound flows to compromised customer systems', MAAWG General Meeting, March 2005, http://darkwing.uoregon.edu/~joe/zombies.pdf.

[9] Abusive Hosts Blocking List, http://www.ahbl.org/.

[10] Joe Stewart, 'Sobig.a and the spam you received today', http://www.lurhq.com/sobig.html.

FEATURE 3

THE IDEAL TOOLS OF AN IDEAL VIRUS LAB

Jozsef Matrai
VirusBuster Ltd, Hungary

Each profession has its own set of tools, and whenever there is an improvement in those tools, the work of that profession becomes more efficient.

Every company in the anti-virus industry has its own confidential technology for studying malicious and potentially malicious code. However, creating all the necessary tools for malware analysis in-house is not always economical, particularly for small companies. This article is aimed at anyone who is a potential user or creator of malware analysis tools.

RUNNING, DEBUGGING AND WATCHING

Malicious code tends to involve a lot more computational effort than non-malicious code. For example, non-malicious code might say: 'I want to UrlDownLoadToFileA http://xxx.yyy.com/zzz.exe to the local file'. However, to avoid analysis, malicious code might scan through the memory for the 'M', 'Z' signatures of DLL files, get the entry point of UrlDownLoadToFileA using only a checksum made from the characters of the function name, and compute the URL string using a long formula.

As a result of this complexity, we need a tool for running, debugging and watching the resource usage of malicious code.

A VIRTUAL WORLD

What is the problem with current commercial products? Commercial debuggers are, in general, hardware-level debuggers. On *Intel x86*-compatible machines they overwrite instructions with opcode 0xCC to interrupt a running program, set the trace bit of the EFLAG Register and modify a DRx breakpoint register. This means that the malicious code may behave differently under debug control than in reality.

A much better solution is to use a virtual environment, such as an emulator, which makes it a lot harder for the malicious code to determine whether it is running on a real machine or under a debugger.

Consider the following scenario: imagine a piece of Trojan code that tries to determine whether it is connected to the Internet or running within an emulated network. The code checks for the existence of two files: 'http://abcdefg.com/

MUST_EXIST_FILE' and 'http://ijklmno.com/{RANDOMSTRING}_MUST_NOT_EXIST_FILE' (where {RANDOMSTRING} represents a random alphabetical string). When the Trojan code discovers that it is running within an emulated network, it stops working completely, even if the executable is restarted.

What happens when you attempt to study this code on a real machine?

- At first, you see two queries in the Bind log: 'abcdefg.com' and 'ijklmno.com'. You reconfigure Bind so that queries to the two sites are redirected to a local server and restore the client machine by overwriting the whole hard disk from the original image.
- On your second attempt you see two queries in the *Apache* log of your local machine: 'abcdefg.com/MUST_EXIST_FILE' and 'ijklmno.com/DFSDFDS_MUST_NOT_EXIST_FILE'. You reconfigure the web server to host these two pages and restore the client machine again.
- After many failed attempts you find yourself reconfiguring the web server, making sure that the file 'ijklmno.com/WERWER_MUST_NOT_EXIST_FILE' exists and the file abcdefg.com/MUST_EXIST_FILE' does not exist. You keep waiting in the hope of finding out whether this Trojan does anything other than making web server queries.

If you had been able to restart the program so that the GetTickCount() calls returned the same values each time, the random number generator would produce the same output: the download queries are always 'abcdefg.com/MUST_EXIST_FILE' and 'ijklmno.com/WERWER_MUST_NOT_EXIST_FILE'. A human operator might not able to do this, but a virtual user can. In this case, you only have four combinations to try: the two files multiplied by the two states (exists or does not exist).

In the virtual world you must have the ability to step into the same river twice or more. If you can feed all GetTickCount() values to a polymorphic engine, it will produce all possible outputs.

Virtual machines may be controlled both by real users and by virtual users. The virtual user may be a script such as the following that allows UNEXEPACK to run only while it is not in our own process or while it is in UNEXEPACK's own code:

```
while (computer[0].Motherboard.CPU[0].CR3 != OUR_PROCESS_CR3
   || Computer[0].Motherboard.CPU[0].EIP >= 0x00600000U ) {
  Computer[0].EmulateNextInstruction();
}
```

Skipping the emulation of OS functions can also help speed things up. Consider a Win32 executable loaded at 0x00400000...0x00403FFF. It uses DLL images loaded at 0x60000000...0xBFFFFFFF. The virtual user (script) can recognize Win32 function calls easily, depending on which address space the register instruction pointer, EIP, belongs to. Instead of having to run all necessary *Windows* processes you can run only one process, and instead of stepping through *Windows* functions, you can use faster replacement code.

Recent PE EXE packers/protectors can cause a headache when replacing Win32 calls. Consider in the unpacked code:

```
FF 15 xx xx xx xx  CALL [USER32DLL_RegQueryValueExA]
```

The packer will overwrite it with:

```
E8 xx xx xx xx     CALL equivalent
xx DB TRASH
```

The first time I came across such a substitution, I thought it would be very easy to handle: all I had to do was to compare the CALLed byte sequence with all byte sequences at DLL export entries. Unfortunately, the replaced function entry code can be obfuscated. There are many Win32 exported entries that look like this:

```
EXPORTED_ENTRY_1:
      MOV    CL, CONST_VALUE_1
      JMP    SAME_ENTRY_POINT
EXPORTED_ENTRY_2:
      MOV    CL, CONST_VALUE_2
      JMP    SAME_ENTRY_PONT
      etc ...
```

It is very common for routines to begin with a JMP to a common address. When I did an experiment to find the equivalent Win32 address of a replaced call, I ran the replaced call for cases where the EIP lay inside the user address space, stored the EIP and the general registers, then ran all Win32 exported entries, to a maximum of 1,000 instructions deep where the EIP and general registers were not equal to the stored values. In approximately 80 per cent of the cases the answer was one exported entry. In the other cases the answer was more than two entries or even no entries.

RESOURCE-MONITORING

When performing malware analysis we need a good resource-monitoring system. Generally we want to watch the disk I/O at file read and write level, but sometimes we want to monitor at the sector read/write level. We would also like to view the network traffic as TCP/IP packets (for email, FTP and HTTP) and as Ethernet packets.

This is another headache. Making a hardware emulator is far cheaper than emulating an operating system. The hardware is completely documented, and the scope is much smaller than an OS.

In a hardware emulator, Ethernet traffic, commands sent to the disk, etc. can be observed, but usually we do not want to do this – we want to monitor file I/O, email I/O, etc. Observation of these things is possible only if the resource-monitoring system knows something about the file system and network protocols.

We have to be able to build our virtual world for components, computers, CPUs, mainboards, Internet servers and so on. Portability is another crucial factor for emulators. Currently we are in transition between the '32-bit age' and the '64-bit age', so any hurriedly-written, non-portable code will quickly end up in the trash.

Instead of constructing an instruction emulator, there is a more complicated solution:

1. Make a global descriptor table when there is no accessible segment for a Ring3 code.
2. Make a local descriptor table with all segments marked as non-readable, non-writeable and non-executable, with the exception of six descriptors for the code you want to study (ES, CS, SS, DS, FS, GS).
3. Make an I/O privilege table, with access to any I/O ports disabled.
4. Run the code you want to study at Ring3. What happens when the OS is called? A small routine, such as GetLastError(), begins and returns. What about a KERNEL32::WriteFile call? The Ring3 DLL code calls a Ring0 KERNEL routine (INT 0x2E under Win2K, INT 0x80 under Linux x86), which causes an exception. This is the way to study the Ring0 calls. Of course, you cannot see the GetLastError() calls.

DECOMPILATION

Reading disassembled code is time-consuming and it would be much better to use decompilers for code that originated as a high-level language. The most important languages are: *Microsoft Visual C++*, *Borland Builder/ Delphi* and *Microsoft Visual Basic*, which has a special compiled format.

A decompiler may be able to resolve Win32 complex data types. Consider C code using Win32 SYSTEMTIME datatype. When EBX points to such a structure, WORD [EBX] is the field dwYear, WORD [EBX + 2] is dwMonth. However, if the data type cannot be recomposed, the SYSTEMTIME structure is an 'unsigned short int Array[8]', an access to dwYear is 'Array[0]', an access to dwMonth is 'Array[1]', and the resulting C code will be less readable than the output of a very intelligent decompiler.

Code comparison is also a very important goal. Imagine that one analysed program contains routines A, B, C and D. When another researcher analysing a different program finds A, B, C and D routines, they should be able to refer to the former analysis.

SUMMARY

For efficient malware analysis we need a virtual world, decompilers and code comparators. Emulation is the easiest solution from an algorithmic point of view. A lot of free software is available to help us to build virtual worlds, such as *Bochs* (emulator), *Windows Emulator*, *Samba*, *Apache*, and so on. They know file formats, network protocols and hardware specifications.

Code comparison is algorithmically simple at assembly level, but it is very difficult at high-level language level. When code comparison tools are being developed, it is important to retain backward-compatibility with routines that have been analysed previously.

Decompilation is also very difficult. No commercial or open source solutions are available for the very complex tasks, but I have heard of some in-house solutions. For example, Ero Carrera and Gergely Erdélyi introduced a code comparison tool for malware-naming in their VB2004 conference paper 'Digital genome mapping', and Lubos Vrtik introduced a VBA6 decompiler in his VB2003 conference paper 'Inside VBA6 decompiler'.

CONCLUSION

This article contains my own views on the tools that are needed by a very intelligent malware analysis lab, and I would welcome the opinions of others.

One day I will be sitting in the ideal virus lab, studying software that looks like this:

```
if ( MD5SUM ( _1st_Input() ) = _CONST_1 ) {
       if ( MD5SUM ( _2nd_Input() ) = _CONST_2 ) {
            if ( MD5SUM ( _3nd_Input() ) = _CONST_3) {
                     _1_st_Output()
...
```

As I study I will be considering how I will explain to an average, less-than-gifted user about the circumstances in which the software produces the results of '_1st_Output()'. If only that was my biggest problem.

Q & A

MICROSOFT'S DOG-AND-BONE OS – SMART AND SAFE?

Juha Saarinen
Independent technology writer, New Zealand

So, I got myself a smartphone with a 400 MHz *Intel Xscale* CPU, 128 MB of memory and a fast, EV-DO Internet connection that hits 7-800kbit/s at times. It's brilliant. With it, you can load and edit *Word* and *Excel* files, run a fair few executables and read your email, as well as wonder why website designers don't take into account us poor sods with 320 x 240 screens (yes, I know about Bitstream Thunderhawk and its 800 x 600 virtual resolution display).

Sites do load fast though, even if I can't actually make out a great deal of many. On top of that, the music, video, *MSN* IM, and *Skype* capabilities mean that I hardly ever use the smartphone for making mobile phone calls.

The phone is, however, a device that runs *Windows* as its operating system. More specifically, *Windows Mobile 2003 Second Edition* build 14132. I'm cool with that, but where's that Windows Update function to keep it safe and sound? After all, it does run *Internet Explorer* (albeit sans active content) and *Outlook*.

Before the current smartphone, I had another one that ran (and still runs) *Windows for Pocket PC 2002*. However, I can't upgrade that to *Windows Mobile 2003 SE*.

Does this lack of updates mean that *Microsoft*'s programmers have created an impenetrable device with which I can stumble around the Internet? I understand that this is an entirely different hardware platform from *Intel IA32*, but still, it has a powerful processor and fast Internet access, so surely it must be a juicy target for malware writers.

VOICING CONCERNS

I decided to put my concerns to *Microsoft*. My first port of call was *Microsoft*'s New Zealand office (*MSNZ*), where I was told that it is the vendor and/or the device manufacturer's responsibility to furnish customers with updates. Next, I tried contacting the vendor and the manufacturer of the smartphone with my concerns, but drew a blank with both, so I went back to *Microsoft* with my doubts.

This time around, Brett Roberts, *MSNZ*'s manager of platform strategy and security, took some time to explain how it all works and to allay my fears of a hacked smartphone with thousand dollar bills (the monopoly telco in New Zealand charges an arm and a leg for mobile data).

First, Brett told me: 'the first thing to consider is the difference between *Windows Mobile* and our operating system on the desktop, and in turn the resulting difference in upgrade venues and frequency.'

He added: '*Microsoft*'s relationship with PC owners is direct in many ways – we update software through *Windows Update*. In the mobile world the operator or device maker owns the device image and is responsible for updating the software.'

He was keen to stress *Microsoft*'s dedication to improving software and security though, saying, 'I stress that *Microsoft* is continually updating its software and providing updates to operators and OEMs which they, in turn, can use for new devices and those already in the hands of customers. The updates we provide are mostly based upon direct customer (OEM, Operator) feedback in terms of 'fixes'.'

Q & A

I didn't feel that the answers I had been given had really told me very much, so I decided to fire some more specific questions at Brett regarding smartphone updating. In the following dialogue I am Mr Q, and Brett is Mr A.

Q: Is it safe to use my *Windows Mobile* device without updates?

A: 'Strategically, there are three key areas with respect to security on the *Windows Mobile* platform:

i) Protecting access to the device.

ii) Securing data on the device.

iii) Securing the connections used to exchange data.

'Across each of these areas are secure software development processes, training and testing as part of the larger *Microsoft* Trustworthy Computing Initiative to help ensure that *Windows Mobile* software is secure. Additionally, *Microsoft* works to ensure that we enable a rich ecosystem of third-party security providers to provide additional security that may be required beyond what is provided natively in the platform.

'As an end-user, two of the most important things you can do to protect your device are:

i) Ensure perimeter security.

ii) Know the source of the applications you are downloading.

'Perimeter defence can be as simple as taking advantage of the strong password support provided natively in the platform or employing third-party solutions that enable you to wipe the device data if too many incorrect passwords are entered. Similarly, just as you exercise caution in the type of applications you download on to your PC, you should exercise the same care with your mobile device. If you don't trust the source of the application, you should not download it.'

Q. What is different about the *Pocket PC* that makes it 'invulnerable' so that it does not require regular updates?

A. '*Microsoft* believes that no software is invulnerable to security threats. As mentioned above, employing the functionality that exists natively is the primary line of defence. Using a device PIN for perimeter security and knowing the source of the applications you download and run are the most important things that an end user can control.

'Depending on the nature of security required, third-party security solutions may provide an additional level of security – for example, data encryption. *Microsoft* does provide periodic updates to the *Windows Mobile* software to our device-maker partners. Depending on where device makers are in their development cycle and commercialisation process, they may or may not elect to make these updates available.'

Q. Why is it so easy to update *Windows* (and other OS) regularly, and why is it not the same for the *Pocket PC*?

A. 'The ease of updates for *Windows* on the desktop has been a convergence of two important factors:

i) The great work by the *Windows* division on *Windows Update* technology.

ii) The evolution and standardisation of PC hardware over the last 20 years. There is no question that the hardware standardisation has made it easier to create and deploy universal software updates to the existing PC base.

'*Windows Mobile* software is different in several ways from that of the desktop software with respect to updates. First, it is important to understand how the handheld device differs from the PC. *Windows Mobile* is a rich platform that provides an integrated telephony, PIM (Protocol Independent Multicast) experience. The robustness of the platform also enables unique solutions and applications to be built upon it. When a device manufacturer creates a *Windows Mobile*-based device, they integrate the *Windows Mobile* software with their own hardware, ensuring that the drivers (which interface between the software and the hardware) are optimised for their hardware and ensure integration with any applications or solutions experiences they might be adding.

'For connected devices (i.e. telephony-enabled devices), further optimisations may be required for each device model for each mobile operator network, in effect creating numerous similar, yet distinct products. Because the smartphone/connected-device space is so nascent, there is still a lack of standardisation of device drivers between hardware manufacturers. In addition, unlike PCs where the operating system is on a hard drive, *Windows Mobile* images are flashed to ROM, meaning that the update process requires a unique flashing mechanism on each hardware platform.

'This has significant implications on ubiquitous updates: mobile software updates must work seamlessly across a permutation of unique hardware and mobile operators' network optimisations. Accordingly, any software update must currently occur via tight integration of the software provider, the device maker and the mobile operator.

'Until the mobile device hardware reaches a sufficient level of maturity, the most effective and reliable way to deploy updates is for the platform provider to provide a software update to the device maker, who in turn creates a specific update for a specific device on a specific mobile operator, who then would roll out to the end customer.'

Q. Does *Microsoft* have any plans to provide an update service like *Windows Update* for *Windows Mobile*-based devices?

A. 'While we cannot comment on specific plans for future releases, *Microsoft* continues to investigate ways to provide a more seamless update experience of *Windows Mobile*-based devices for device makers, mobile operators and end users.'

Q. *Sony* provides regular updates to its mobile phone operating systems, incidentally. Should, perhaps, *Microsoft* work with vendors to ensure that they release regular updates and not just abandon year-old products?

A. 'While we cannot comment on competitors' practices, for *Windows Mobile* products, we certainly make upgrades available to OEMs who have in the past offered it to end users. However, the decision on whether or not to make an upgrade available to their customers is a decision on the part of the device maker who must create the upgrade and the mobile operator who will help roll it out.'

Q. Could you elaborate on *Microsoft*'s commitment to security on the *Windows Mobile* platform?

A. '*Microsoft* takes the security across all of its products seriously and *Windows Mobile* is no exception. *Windows Mobile* software is part of *Microsoft*'s larger Trustworthy Computing Initiative. As part of that initiative, later versions of *Windows Mobile* software have been undergoing rigorous security reviews during development and developers are given special training on secure software development practices. We also strive to provide a rich open platform that enables our valued partner ecosystem to develop additional software applications and solutions to help respond to security needs in the marketplace.

'*Microsoft* utilises a multi-pronged strategy to empower businesses and users with a more secure mobile computing experience.

- Threat modelling: *Microsoft* conducts threat modelling as a regular piece of our security program. This includes our own internal code review plus extensive testing by third parties.
- Two-tier access: *Windows Mobile for Smartphone* supports a two-tier access model (Privileged/Unprivileged) that is flexible in order to meet varying operator network requirements and provides strong security options to ensure end user satisfaction. This technology can be used to control which applications can install and execute on a device.
- Third-party solutions: a number of *Microsoft* partners offer a wide range of security solutions for *Windows Mobile*-based devices, including *Computer Associates*, *F-Secure*, *Symantec* and *McAfee*, *JP-Mobile Developer One*, *Information Security Corp.* and *Ilium Software*.
- End-user education: in addition to utilizing the technical security features in *Windows Mobile* software, *Microsoft* recommends that users employ the following safeguards to help protect the data stored on their devices:
  - Activate password protection on the device.
  - Install software and accept files only from reputable sources.'

BETWEEN THE LINES ...

Righto, no worries in other words, and no need to check for that ARM port of OpenBSD with telephony features.

[*In next month's issue of VB, Michael Moser takes a long hard look at the statements made here by Microsoft and gives us his interpretation.*]

PRODUCT REVIEW

SYMANTEC ANTIVIRUS 10

Matt Ham

I never know quite what to expect when a new product version number is announced by an anti-virus vendor. On the one hand, a vast number of changes may have been introduced – which can be interesting, yet confusing initially. At the other end of the spectrum are those products in which the new version remains all but identical to the old one, leaving the user (or reviewer) mystified as to exactly where the improvements lie. Where *Symantec*'s new version is positioned in this range should become more obvious as the review unfolds.

The brief thumbnail sketch of the vendor I usually include as part of the introduction to product reviews seems somewhat superfluous in this case. I have vague memories of a *Symantec* defragmentation application being bundled with some of the earlier versions of *Windows*. Ironically, back then I wondered whether the company would either be taken over by *Microsoft* or vanish as its products were assimilated into *Windows* standard features. It seems I need not have concerned myself. In the battle for the anti-virus vendor number one spot in terms of sales volume, *Symantec*'s position has been in the top two for many years now. Since the number one and two positions are very much dependent upon who is counting and what is counted, a stronger position in the market is not currently achievable.

As far as current product lines are concerned, *Symantec*'s offerings are many and varied. The anti-virus product has expanded, as is common nowadays, to cover a broader range of threats than before, with anti-spyware and anti-adware capabilities and increasing integration with the *Symantec* firewall products. The company's firewall offerings are part of a range of security and networking applications, including intrusion detection, vulnerability assessment and network status tools. Remote computer control is offered through *PC Anywhere* and imaging through *Ghost*. The anti-virus software is offered on a variety of platforms, though on this occasion *Windows* was used for testing.

DOCUMENTATION

At the time of writing this review, the first few boxed versions of *Symantec AntiVirus* (*SAV*) *10* were rolling off the production lines. Unfortunately, however, a slight delay in the supply chain between manufacturer and my desk resulted in my having only an electronic version of the software for review. This was supplied as the entire contents of the *SAV* CDs, however, so all that was missing was the hard copy documentation. Past versions of *SAV* have been supplied with a 'Getting Started' gatefold card and a comprehensive installation manual, and I would expect much the same with this release.

The documentation in the electronic version is collected in one directory and is mostly in PDF format, with *Acrobat Reader* supplied for convenience.

There is one text file in the documentation, which is the reference guide for the use of *MSI* in the installation of *SAV*. This provides information on commonly-used command line switches and examples of their use. This was also where one of the new features in *SAV 10* was first revealed to me (although the changes are described at length within one of the PDFs). With *Windows XP Service Pack 2* installed the operating system can complain vociferously if a firewall is not installed locally. In many circumstances a LAN will be running numerous applications which either fulfil firewall functionality at one point, or would be hindered by the default settings of *Windows* firewall. *SAV* can be set up so as to suppress these warnings from the *Windows Security Center*. Other functionality is also available to interact with the *Windows Security Center*, most aspects of which can be controlled during *SAV* installation.

The documentation supplied includes PDF versions of both the installation manual and Getting Started card. The manual retains much the same style as the older one. For those unfamiliar with the *Symantec* documentation, it is one of the more easily used of those supplied by current vendors. The user can select from dedicated manuals for various aspects of functionality. As mentioned, one manual is dedicated to installation, while others cover central quarantines, LiveUpdate control, client software information and, of course, the more general Administrator's Guide. There is also a 50-page reference guide which covers the areas most likely to cause confusion.

With all the documentation available, the online help files should not require too much use, and are present in all those applications tested, with the notable exception of the LiveUpdate Administration utility. Where present, the help function information is comprehensive and is not simply a direct copy of the PDF documentation ported into help format.

THE COMPONENTS

With a product so obviously designed for centralised administration, the obvious test scenario was to install it on a server and then see how easy it would be to install to various attached workstations. In order to test this, several components needed to be installed. The first few times that I tested *SAV*'s administration tools the process proved to be very involved, with numerous installs required, in the right order, to obtain a centralised installation. With this release

the situation has improved significantly, although it is not totally intuitive yet.

When running the installation application from the CD, the user is presented with the option of installing *SAV* or installing Administrator Tools. There is also the option to 'Read This First', which is very much advisable. In fact, this consists of links to the most relevant parts of the PDF documentation, and somewhat deep within the information provided through these links is the correct order for installation, this being: 'Install the Symantec System Center. Install *Symantec AntiVirus* server software on the same computer as the Symantec System Center, and configure it as a primary management server. Install *Symantec AntiVirus* client software on other computers.' It would have been nice to have had an option available to perform each of these processes in turn.

The components inspected in this review were the Symantec System Center and *SAV* itself on client and server. In addition, the LiveUpdate Administration (LUA) utility was examined, since this can be used to control updates in a networked installation. The LUA utility seems to be the nearest thing to an afterthought supplied in the package – as mentioned already, it does not have the same level of integrated help available, and it was not noted as a standard part of a *SAV* installation. In fact, the utility covers all *Symantec* products which make use of the LiveUpdate mechanism for updating components.

INSTALLATION AND UPDATES

As suggested, the Symantec System Center was the first component to be installed. This requires *Internet Explorer* version 5.5 or higher to be available on the machine in question. For the purposes of testing, therefore, *Windows 2000 Server* with *Internet Explorer 6* was chosen as the installation environment.

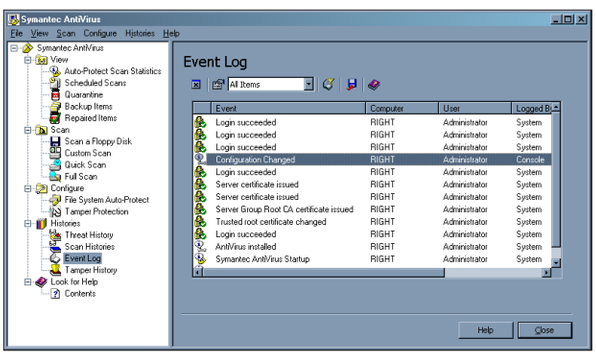
After the usual licence agreement has been accepted, the components to be installed can be selected. This area gives an idea as to the functionality offered within the System Center – *Symantec AntiVirus* and *FireWall* snap-ins and the server and client remote installation tools being selected by default. Alert Management System Console is not installed by default, but may be added to the selected packages here. For the purposes of the test all possible options were installed. Once the components have been selected, the installation completes with little further interaction required, although a reboot was required after installation.

Next on the to-do list was the installation of the *Symantec AntiVirus Server* software on the same machine. This is performed by selecting the action to install the software, rather than deploy it, from the CD setup menu. The choice appears as to whether a server or client installation is required, after which installation settings may be altered. Rather than consisting of major server-specific functionality, the options here are merely whether the GUI, help files and quarantine client files are to be installed. These are all installed by default, with there being few reasons likely to warrant a change from the settings provided.

Server-specific settings do occur in the second phase of configuration choices. Here, the password and username for the machine's Server Group must be selected. A Server Group in this context is a group of machines which can be configured *en masse*. Such groups are not limited to containing a single server machine, though one server must be designated as the primary server for the group. If the Server Group name chosen is not already in existence a check is provided that this is a new group, rather than a typo, after which general settings for the installation are also chosen. At this stage the general settings are simple: whether to use Auto-Protect, the on-access component, and whether to use LiveUpdate when the installation has terminated. With these choices made installation completes without further ado.

A reboot is not required after this part of the installation chain. Somewhat confusingly this server installation does, however, show up as having installed Symantec Client Security. Admittedly the security is provided for the clients, but seeing this as the result of a server installation led me to believe that I had installed the incorrect software at first.

At this stage the machine in question is ready to be configured as primary management server for the newly created Server Group. This is performed through the Symantec System Center (SSC). SSC is implemented as snap-ins for the *Microsoft Management Console* (*MMC*), thus offering a centralised area for configuration changes with any other *MMC*-integrated applications. Upon launching, this offers what amounts to a direct link to various FAQs – falling under the headings of 'Management Console', 'AntiVirus Protection Management' and 'Firewall



Protection Management'. At this stage, however, these are distractions from the task in hand, and the user is better served by selecting the Symantec System Center tree in the left-hand pane. Other functionality will be discussed later in the review.

Selecting the Symantec System Center tree brings up the System Hierarchy tree which, as a default at this stage, consists of just one server group, 'Symantec AntiVirus 1'. What is not immediately obvious is that the red diamond forming part of the System Hierarchy label is a decoration, rather than a sign that all is not well with the SSC settings. There is a problem nonetheless, in that there is no default primary management server. The server group is locked by default – unlocking requires the username and password provided during the installation. The username and password may be saved within *MMC* in order to automate the unlocking on future occasions.

With the preamble completed the newly-installed server may be edited to become the primary management server. At this stage there is a change in the tree labels for the server itself, where the new labels for the server change from red to blue to indicate that there are no problems.

With the SSC settings tweaked as directed in the installation documentation, it is possible to return to the process of installing clients. The CD setup menu is where this operation is performed. Client machines were running *Windows XP*, both with and without *Service Pack 2*, and were located in a domain which included the server machine. At this stage the 'Deploy AntiVirus Client to 2000/XP/2003' option was selected, which launches the ClientRemote Installation application. As the first part of this application the path to the installation files must be supplied. This mentions several places which have not even been hinted at in the process so far, making it a happy event when the default location worked with no further thought required.

With the source location having been supplied, the management server and the destinations must now be selected. ClientRemote Installation detects machines reachable over the network which may be selected for installation via a GUI. All machines on the network were presented here, although if workgroup-based machines or machines without *Internet Explorer 5.5* or later were selected, the process of installation was noted as being impossible if attempted.

Having selected the machines towards which to roll out the software the remainder of the process simply involved activating the process and watching the progress bars. Somewhat ominously, the instructions after installation were to check whether machines required a reboot. Among the test machines examined, however, no reboots were required.

With the server and clients now in place, all that remains for basic functionality is to ensure that updates occur properly. For machines with a direct Internet connection LiveUpdate is the most obvious update facility provided, though not one that will be very popular in larger or more managed organisations. The LiveUpdate Administration utility is of help here. This enables a customised hosts file to be produced, pointing to a central repository within the organisation. This repository can be attached directly to LiveUpdate, or fed material from the LiveUpdate servers after it has undergone internal testing. The LiveUpdate Administration (LUA) utility, as mentioned before, is somewhat less elegant and less well documented than the other portions of the software, but it works nonetheless.

FEATURES

The Symantec System Center has, if anything, too many features to allow for a full discussion. Luckily the majority of these features exert control of the clients at a distance and a discussion of client functionality will thus cover a large proportion of what the SCC can be used for.

Such conveniences as forcing client updating, applying a degree of randomisation to scheduled client updates and locking down client-side configuration changes are supported of course. The latter has had additional care applied, as spyware has increasingly aimed to sabotage or subvert security software. Communication between client and server is now secured cryptographically, with the ability to configure key durability for the paranoid. Any attacks on *SAV* are also noted and can be pre-empted and logged, though the latter option is not turned on by default.

The default hierarchy is rather simpler and more monolithic than can be constructed using SCC. Within each server group, for example, can be defined client groups, each with their own set of rules. These client groups can be created arbitrarily based on machine requirement similarity, without reference to where in the network structure they are located logically. The SCC allows for different users to be allowed different levels of control over the settings of *SAV*.

All of this is much as expected, so we can return to the client software itself – all functions as noted here are also controllable and can be initiated from the SCC. The client is of the standard two-paned type beloved of *Windows* developers, the left pane containing the trees of View, Scan, Configure, Histories and Look for Help.

The View tree can be used to produce views of on-access file scanning details and the scheduled scans set up for the machine. In addition, the quarantine, backed-up items and repaired items can be viewed here. Obviously some of these areas will be very dull indeed, since it is not guaranteed that

any viruses will be discovered. For a general user, however, this may be their only clue as to the presence of any infections on their machine, since the administrator can set disinfection options to be invisible to the user.

The Scanning tree consists of three preset scans: Floppy, Full Scan and Quick Scan. Custom scans may also be set up here. These offer all the usual choices, though the options available are much more technically detailed than would have been the case a few years ago. I suspect that they may still be a little too advanced for the average user – but it is nice to have this level of detail as to what options actually mean when implemented. I, for one, would far rather see more details than some incomprehensible trademarked name which means the same thing when translated.

Perhaps more care than usual is devoted to the treatment of files migrated during backup procedures. This is an area where headaches look possible, though the attention will probably be ignored by most, and praised effusively by those who are directly affected. It is also of note that the default option is now for active scanning for spyware and adware. A different treatment may be set for each of these threats when detected, along with associated jokes, tools and an assortment of other categories. This differentiation is a good thing to see, though perhaps an area where the 'correct' action will be open to debate.

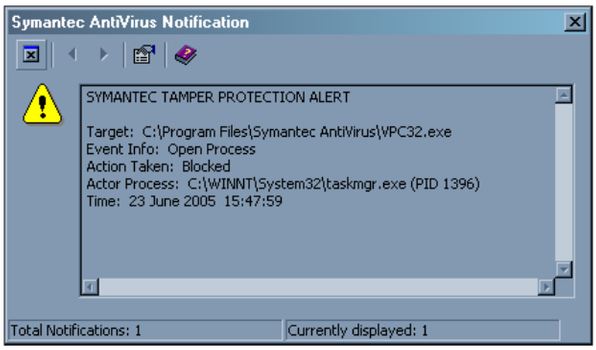
The Configure tree is devoted to the on-access functionality of the software. This is primarily the on-access virus scanning, though the Tamper Protection functionality also falls into this category. This allows alterations to *SAV* to be either blocked or logged. I did not, however, manage to trigger either function while using the simplest of attack methods: deleting *SAV* files randomly. However, it did block and notify when attempts were made to halt *Symantec*-related processes.

The deleting of files did have an unexpected side effect, in that the installation routine was triggered, replacing the affected files. However, this self-repair functionality was limited in the test environment since deleted virus definitions caused the process major problems. This should be less of an issue if LiveUpdate is attached to the Internet whether directly or indirectly.

The last two views, Histories and Help, are self-explanatory, showing the various log files and the online help. In addition to the standard Threat History, Scan History and the Event Log, this is the area where the Tamper History log can be inspected.

CONCLUSION

Rather than being a significantly different product from its previous incarnations, *SAV 10* is very much *SAV 9* with



additions. These additions are relatively small in number and reflect, by and large, the need for greater internal security within *SAV*'s infrastructure and the integrity of its operation.

Some might point out that the small number of changes involved are as a result of *Symantec* having put every possible feature into the software already – the client installations here came to 115 MB, while the server installations were a substantial 225 MB. There certainly seemed to be little lacking in the tasks I set out to perform. This completeness is aided by the documentation supplied, which is large in both quantity and quality.

The major perceived weakness of the product still lies in the installation procedures, which remain far less simple than could be the case. Admittedly, when rolling out software to a large network this could be a good thing, since having to 'look before you leap' is likely to help some decisions to be made more carefully. Overall, however, the process could be more automated – a fact which is even more obvious when reading through the *SAV* documentation on this very subject. On the other hand, the situation is a lot clearer and less prone to problems than in the past.

SAV 10, therefore, can be seen as an evolved version of *SAV 9*. It will shock few and be easily usable by current *SAV* administrators. The changes within it reflect current malware trends, which cannot be a bad thing. With more proposed integration of security into *Windows* itself on the way in *Longhorn*, however, I have my suspicions that *SAV 11* will include a far greater set of changes than we have seen in *SAV 10*.

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*, *Windows XP Professional SP2* and *Windows 2000 Server SP6*.

Developer: *Symantec Corporation*, 20330 Stevens Creek Boulevard, Cupertino, CA 95014, USA; tel: +1 408 517 8000; email: sales@symantec.com; website: http://symantec.com/.

END NOTES & NEWS

A SRUTI 2005 workshop entitled 'Steps to Reducing Unwanted Traffic on the Internet' takes place 7–8 July 2005 in Cambridge, MA, USA. The workshop aims to bring academic and industrial research communities together with those who face the problems at the operational level. See http://www.research.att.com/~bala/sruti/.

Black Hat USA takes place 23–28 July 2005 in Las Vegas, NV, USA. Training will take place 23–26 July and the Briefings will take place 27–28 July. For details and online registration see http://www.blackhat.com/.

The 14th USENIX Security Symposium will be held 1–5 August 2005 in Baltimore, MD, USA. For more information see http://www.usenix.org/.

T2'05, the second annual T2 conference, will be held 15–16 September 2005 in Helsinki, Finland. The conference focuses on newly emerging information security research. All presentations are technically oriented, practical and include demonstrations. See http://www.t2.fi/english/.

COSAC 2005, the 12th International Computer Security Symposium, takes place 18–22 September 2005, running on a fully residential basis at the Killashee House Hotel, near Dublin, Ireland. A choice of more than 40 sessions and six full-day master classes and forums is available. Attendance numbers at the symposium are limited and the remaining places will be issued on a first-come, first-served basis. The full programme and details of how to register are available at http://www.cosac.net/.

The Network Security Conference takes place 19–21 September 2005 in Las Vegas, NV, USA. The conference is designed to meet the education and training needs of the seasoned IS professional as well as the newcomer. For details see http://www.isaca.org/.

The 5th Annual FinSec Conference takes place 20–23 September 2005 in London, UK. This year's conference will focus on the unique set of challenges afflicting information security professionals in the financial community. See http://www.mistieurope.com/.

e-Secure Malaysia 2005 takes place 28 September to 1 October 2004 in Kuala Lumpur, Malaysia. Organised by the Malaysian government, this inaugural exhibition and conference will cover pertinent issues such as computer emergency response, spam and viruses, hacking, cyber laws and terrorism, security management, access control, home computing and network security. For more information email conference@esecuremalaysia.org.my or see http://www.protemp.com.my/.

The 15th Virus Bulletin International Conference, VB2005, will take place 5–7 October 2005 in Dublin, Ireland. The programme for the three-day conference can be found on the *VB* website. For more information or to register online see http://www.virusbtn.com/.

Black Hat Japan (Briefings only) will be held 17–18 October 2005. Further details will be announced at the Black Hat USA event in July. See http://www.blackhat.com/.

RSA Europe 2005 will be held 17–19 October 2005 in Vienna, Austria. For more details see http://www.rsaconference.com/.

WORM 2005 (the 3rd Workshop on Rapid Malcode) will take place 11 November 2005 in Fairfax, VA, USA. The workshop will provide a forum to bring together ideas, understanding and experiences bearing on the worm problem from a wide range of communities, including academia, industry and the government. Full details can be found at http://www1.cs.columbia.edu/~angelos/worm05/.

The eighth Association of Anti-Virus Asia Researchers International Conference (AVAR 2005), takes place in Tianjin, China on 17 and 18 November 2005. The theme of this year's conference will be 'Wired to Wireless, Hacker to Cybercriminal'. For more details email avar2005@antivirus-china.org.cn or see http://aavar.org/.

Infosecurity USA will be held 6–8 December 2005 in New York, NY, USA. The conference will take place 6–8 December, with the accompanying exhibition running from 7–8 December. The full conference programme will be announced this month. For details see http://www.infosecurityevent.com/.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery: £195 (US$358)

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 Web: http://www.virusbtn.com/

vbSpam supplement

CONTENTS

NEWS & EVENTS

TREND RAISES ITS ANTI-SPAM PRESENCE

Trend Micro Inc. has acquired IP filtering and reputation services specialist *Kelkea* (formerly known as *Maps*). The acquisition of the non-profit company, which currently serves large-scale ISPs such as *AOL*, is part of *Trend*'s expansion into the anti-spam arena alongside the company's on-going alliance with *Postini*. *Trend* says that it plans to offer *Kelkea*'s services as an add-on to its service provider and enterprise customers.

MICROSOFT SUES AGAIN

Microsoft is back in court continuing its fight against spammers, this time in Germany. The software giant has filed a case against a company in the North Rhine-Westphalia region of Germany and its managing director.

According to *Microsoft*, customers of its *Hotmail* email service received thousands of emails advertising web design services, online casinos and pornography, which it says were traceable back to the unnamed Westphalia firm. *Microsoft* alleges that the German company is part of a wider spam group operating out of the Ukraine and the US and that its managing director has been running a business renting 'bulk mailer' servers to spam companies. Since there are no anti-spam laws in Germany, *Microsoft*'s case is based on the country's competition laws.

Meanwhile, *Microsoft* is upping the pressure on email senders to adopt Sender ID. Towards the end of the year, the company's *Hotmail* and *MSN* mail services will begin to flag as spam any incoming emails that do not carry a tag to verify the sender. Craig Spiezle, a director in the Technology Care and Safety group at *Microsoft* described the move as 'a call to action for domain holders and email senders to publish their SPF records'. It is estimated that around 30 per cent of email carries Sender ID information.

LIBERTY ALLIANCE TO SAFEGUARD MOBILE USERS AGAINST SPAM

Liberty Alliance, the global consortium for open identity standards and identity-based web services, has put together a team whose aim is to create service interface specifications for content messaging, including both SMS (Short Message Service) and MMS (Multimedia Messaging Service), in an attempt to safeguard the privacy of mobile device users.

Add-ons for mobile devices, such as new ring tones, skins and wallpapers, are becoming increasingly popular, particularly in Europe, and business is booming for the providers of this type of content (content messaging). The transaction between user and provider takes place via SMS or MMS text message: the user sends an SMS or MMS text message to the content provider to request the content. The provider then downloads the relevant content to the user via a series of SMS messages containing the encoded content. However, the Liberty Alliance, whose membership includes more than 150 international companies, non-profit and government organizations, has recognised that this practice leaves the user vulnerable to privacy intrusion and spam, since there are no regulations in place that prevent content providers from passing the user's information on to others.

The Content SMS and MMS (CSM) Service Interface Specifications will give content providers access to a mobile network, allowing the receipt and delivery of SMS and MMS messages both to and from users. As a result, users will be able to obtain content for their devices anonymously, thus eliminating the opportunity for their details to be passed or sold to third parties. The specifications will be deployable based on the Liberty's Identity Web Services Framework (IS-WSF). For more details see http://www.projectliberty.org/.

EVENTS

CEAS 2005, the Second Conference on Email and Anti-Spam, will be held 21–22 July 2005 at Stanford University, CA, USA. For more details see http://www.ceas.cc/.

TREC 2005, the Text Retrieval Conference, will be held 15–18 November 2005 at NIST in Gaithersburg, MD, USA. For more details see http://trec.nist.gov/.

FEATURE

BLOCKING UNWANTED MAIL WITH MAIL AVENGER

David Mazières
New York University, USA

Junk mail filters must evolve constantly to keep pace with increasingly clever spammers and virus writers. Mail Avenger, developed by the New York University Secure Computer Systems group, is an extensible SMTP server designed to facilitate mail filter innovation. It allows users to implement sophisticated filtering policies easily using Unix shell syntax, which is familiar to most administrators and many end users. Mail Avenger runs as a wrapper around existing mail transport agents (MTAs), permitting people to adopt new filters regardless of the underlying mail system in use.

SMTP-TIME FILTERING

The best time to filter mail is as early as possible, during the execution of SMTP, the Internet mail protocol. Refusing mail during an SMTP transaction saves the server from having to spool unwanted messages. Moreover, since legitimate clients notify senders of SMTP failures, inappropriately blocked mail will be brought to the attention of the sender.

A further advantage of filtering during the execution of SMTP (SMTP-time filtering) is that more information is available to filters while the client is still connected to the server. For example, filters can examine the network route to the client or check frequently updated real-time blacklists (RBLs) when deciding whether to accept a message.

Unfortunately, most MTAs make SMTP-time filtering difficult by requiring MTA-specific, trusted plug-in code that can affect all users if it malfunctions. To avoid this hassle, many people run mail filters at delivery time, through .forward, .procmailrc, or .qmail files, which allow one to hook in external filter programs. However, by the time such programs run, the server has already accepted mail from the client, leaving no satisfactory way to reject it.

Filters typically discard bad mail silently or place it in a dedicated junk folder, but either option allows legitimate mail to be overlooked if improperly categorized. Notifying senders of blocked mail by generating bounces is not a good solution, however, because most spam comes from forged sender addresses, meaning that innocent third parties receive unwanted bounces.

SMTP IMPLEMENTATION

Mail Avenger opens up the server-side SMTP implementation, allowing users to control SMTP responses with scripts and external programs. There are three principal commands issued by an SMTP client to send a message to the server. In order, these commands are:

```
MAIL FROM:<sender-address>
RCPT TO:<recipient-address>
DATA
message-body
.
```

The server responds to each command with a three-digit result code, followed by a more detailed explanation (and optionally an extended result code).

To accept mail, the server returns a 200-series result for each command – often just '250 ok'. To reject mail, the server returns a 500-series result for one of the commands (e.g. '550 unknown user'). Alternatively, the server can defer mail by returning a 400-series result (e.g. '451 temporary DNS error'), in which case a legitimate client will keep trying to send the message for a few days.

Mail Avenger performs most filtering in response to RCPT and DATA commands. Generally, it returns success to MAIL commands, unless the sender address is syntactically malformed or some transient error occurs (such as an overload condition or a name server failure).

The result of the RCPT command is determined by running a script depending on the recipient. At small sites, where each recipient corresponds to a Unix user, Mail Avenger runs the script ~/.avenger/rcpt in the user's home directory. For users without ~/.avenger directories, Mail Avenger runs a system-wide fallback script: /etc/avenger/default. Configuration files also allow administrators to map mail aliases and virtual domains to particular users.

~/.avenger/rcpt files are ordinary Unix shell scripts, sourced from a script that pre-defines a number of Mail-Avenger-specific environment variables and shell functions. For example, here are a few of the environment variables Mail Avenger sets:

`CLIENT_NAME, CLIENT_IP`	domain name and IP address of the client
`CLIENT_NETPATH`	the network route to the client
`CLIENT_SYNOS`	a guess of the client's operating system type
`RECIPIENT`	the recipient address of the message
`SENDER`	the sender address of the message
`SENDER_LOCAL, SENDER_HOST`	the user and hostname parts of SENDER

`SENDER_BOUNCERES`	SMTP error if SENDER cannot receive bounces
`SPF`	SPF disposition (whether CLIENT_IP is authorized for SENDER)

Many of these values are also included in a new X-Avenger: header field, which may help certain Bayesian spam filters.

Here are some of the shell functions available:

`accept [MESSAGE]`	This signifies that the server should accept the RCPT command with response '250 *MESSAGE*'.
`reject [MESSAGE]`	This signifies that the server should reject the RCPT command with response '550 *MESSAGE*'.
`defer [MESSAGE]`	This signifies that the server should defer the RCPT command with response '451 *MESSAGE*'.
`redirect user`	This redirects processing to the rcpt file corresponding to *user*.
`errcheck`	This rejects the mail if some simple default checks fail (for instance, if SPF indicates the mail is a forgery, or SENDER cannot receive bounces).
`greylist`	This defers mail the first time a SENDER uses a particular CLIENT_IP, but accepts if the client tries again at least 30 minutes later from the same CLIENT_IP. This technique has been known to defeat certain automated spambots.
`spf VARIABLE QUERY`	
`setvars`	This assigns *VARIABLE* to be the result of a query about CLIENT_IP using the SPF sender-specification language. Note the assignment to *VARIABLE* doesn't happen until the setvars function is called. To reduce latency, one can issue multiple concurrent spf commands (as well as other DNS-related commands that are not mentioned here) and wait for them with a single setvars.
`bodytest COMMAND`	This makes the RCPT command succeed, but then runs *COMMAND* on the body of the message to determine the result of the DATA command.

UTILITY PROGAMS

In addition to pre-defined shell functions, Mail Avenger comes with a suite of utility programs that help construct concise filtering policies. Some examples follow.

Suppose you have a mailing list that is never used as a sender address, and you wish to refuse bounce messages to the list. Because bounce messages come from an empty SENDER address, you can use the following line in an rcpt file:

```
test -z "$SENDER" && reject "no bounces, please"
```

The following line greylists all mail from *Windows* clients (the most likely to be infected by spam-sending malware), using match, a simple string-matching utility that comes with mail avenger:

```
match -q "*Windows*" "$CLIENT_SYNOS" && greylist
```

To run the spamassassin mail filter on the body of an email message, you can use the following commands:

```
errcheck
bodytest "spamassassin -e 100 > /dev/null"
```

errcheck rejects the mail immediately if it is obviously forged, to avoid wasting time with spamassassin. The bodytest command says to run 'spamassassin -e 100' on the message contents. '-e 100' instructs spamassassin to exit with status 100 if it considers the message to be spam. Exit status 100 tells Mail Avenger to reject the DATA command. (Exit status 0 means accept, while most other values result in deferral.)

A limitation of the previous script is that spamassassin annotates messages to indicate what spam tests were triggered by the message, yet the example discards those annotations. Fortunately, Mail Avenger lets bodytest commands edit messages. Mail Avenger even comes with a utility called edinplace that runs a program, replacing its input file with the program's output. Thus, to preserve spamassassin's annotations, use:

```
errcheck
bodytest edinplace spamassassin -e 100
```

Another powerful feature of Mail Avenger is its support for extension addresses, originally popularized by the qmail MTA.

EXTENSION ADDRESSES

Extension addresses allow users to receive mail at multiple addresses. For example, with a default sendmail installation at site 'example.net', Unix user 'user' receives mail addressed not just to <user@example.net>, but also to <user+ANYTHING@example.net>. Qmail uses the '-' character by default, so that user can receive mail to <user-ANYTHING@example.net>. To determine the result

of RCPT commands for <user+EXTENSION@server.com>, user must create a file ~user/.avenger/rcpt+EXTENSION (where EXTENSION is the actual extension in lower case).

One application of extension addresses is to create restricted addresses under which mailing lists can be subscribed to. Suppose you subscribe to mailing lists under the address <user+lists@example.net>. The lists to which you subscribe are all hosted either at New York University (NYU) or Stanford. You want to ensure that spammers cannot send you mail, even if they get hold of the subscriber list. You can achieve this by specifying a policy in ~/.avenger/rcpt+lists that accepts mail only from clients at NYU or Stanford. For example:

```
spf EDU_OK ptr:nyu.edu \
    ptr:stanford.edu mx:cs.nyu.edu/24
setvars
test "$EDU_OK" = pass && accept
test "$EDU_OK" = error && defer "Temporary DNS error"
reject "Address for NYU/Stanford clients only"
```

The spf command formulates a query about CLIENT_IP. Specifically, 'ptr:nyu.edu' asks whether the client's name ends in 'nyu.edu'. Similarly, 'ptr:stanford.edu' checks whether the client's name ends in 'stanford.edu'. Finally, 'mx:cs.nyu.edu/24' checks whether the first 24 bits of CLIENT_IP are the same as any of the mail exchangers for cs.nyu.edu. If any of the tests are positive, EDU_OK is set to 'pass' and the mail is accepted. If there is a temporary error, EDU_OK is set to 'error' and the mail is deferred. Otherwise, the mail is rejected.

Another feature of extension addresses is the ability to write catch-all rules for all suffixes, as with qmail. For example, the file ~user/.avenger/rcpt+bounce+default in user's home directory will match mail sent to <user+bounce+ANYTHING@example.net>. (As with qmail, the word 'bounce' here is an arbitrary string to embed in email addresses, while 'default' is the literal string 'default'.)

One application is to authenticate bounce messages using temporary codes. Doing so solves the problem of viruses and spammers forging your email address and causing you to receive bounces for mail you have not sent. Mail Avenger comes with a utility called macutil that generates and checks cryptographically-protected expiration dates. By setting the environment variable

```
MACUTIL_SENDER="user+bounce+*@example.net"
```

and then sending mail with the command 'macutil --sendmail' (which takes remaining arguments identical to sendmail), you can send outgoing mail from bounce addresses that resemble the following:

```
<user+bounce+tjmutvvdy6qfws4aztwuhsg6we@example.net>
```

Here, 'tjmutvvdy6qfws4aztwuhsg6we' is an encoded expiration date (cryptographically-protected with a password stored in file ~/.avenger/.macpass). You can then reject any bounces sent to your primary email address, by placing the following in your ~/.avenger/rcpt file:

```
test -z "$SENDER" && reject "no bounces, please"
```

Finally, you can check the validity of codes in the addresses at which you receive bounces. Taking advantage of the SUFFIX environment variable, which is set to the portion of the recipient address matching the trailing 'default' in the rcpt file name, you can place the following in your ~/.avenger/rcpt+bounce+default file:

```
macutil --check "$SUFFIX" \
     || reject "<$RECIPIENT>.. user unknown"
```

Because rcpt files are just shell scripts, it is easy to run external programs as mail filters. Moreover, because these programs run as the users in whose directories the rcpt files reside, a buggy rcpt script affects only recipient addresses that use the script. This makes it easy to develop and test new mail filters on a production mail server by deploying them initially only for certain recipients.

At large sites, system administrators can offer non-technical users a menu of filtering options. Default filtering can be implemented in the system-wide /etc/avenger/default file, while other scripts can be configured in the reserved avenger user's home directory, for example, ~avenger/.avenger/rcpt+strict, ~avenger/.avenger/rcpt+experimental. Users who wish to employ a particular level of filtering can simply place a line like the following in their ~/.avenger/rpct files:

```
redirect avenger+experimental
```

CONCLUDING REMARKS

Mail Avenger is MTA-independent. To spool accepted mail, it runs a configurable program (by default sendmail), and therefore it should be compatible with most existing Unix mail servers.

Mail Avenger has been tested with sendmail, qmail, and postfix on a variety of Unix variants including *Linux*, *OpenBSD*, *FreeBSD*, and MacOS X. Mail Avenger is free software, available from http://www.mailavenger.org/.

RELATED LINKS

[1] Mail Avenger: http://www.mailavenger.org/.

[2] SPF (Sender Policy Framework): http://spf.pobox.com/.

[3] SMTP (Simple Mail Transfer Protocol): ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt.

[4] Spamassassin: http://spamassassin.apache.org/.