



virus

BULLETIN

Fighting malware and spam

CONTENTS

- 2 **COMMENT**
MUTE: the rebirth of centralized sharing
- 3 **NEWS**
Tech firm to develop cyber weapon
Weakness in Wi-Fi routers exploited
- 3 **VIRUS PREVALENCE TABLE**
- MALWARE ANALYSES**
- 4 This Sig doesn't run
- 6 Dissecting the NGR bot framework: IRC botnets die hard
- 11 **FEATURE**
The top 10 spam, malware and cybersecurity stories of 2011
- 17 **SPOTLIGHT**
Challenges for the London Action Plan
- 19 **END NOTES & NEWS**

IN THIS ISSUE

NO BRAGGING RIGHTS

Some virus writers like to brag about themselves via their choice of virus name. It's rare that the content justifies the bragging though. The author of W64/Svafa named the virus 'Sigrún', which is Old Norse for 'victory rune'. However, there is little to be victorious about as the virus doesn't work. Peter Ferrie has the details.

page 4

TOP NEWSMAKERS

2011 was filled with plenty of security stories involving spam, malware, hacking and more. Terry Zink picks out his top ten security news stories of the year.

page 11

CHALLENGING TIMES

In 2004, the FTC and the UK's Office of Fair Trading organized a workshop in London in which 27 international organizations participated. They established an informal cooperation network: the London Action Plan (LAP). Wout de Natris describes some of LAP's early successes and the challenges it now faces.

page 17



'The ability to exchange URLs in real time is a particular advantage ... since malicious URLs are usually a time-critical issue.'

Philipp Wolf
Avira, Germany

MUTE: THE REBIRTH OF CENTRALIZED SHARING

When I first applied for a job in the IT security industry back in 1999, I set my sights on the virus lab because it sounded like the most interesting area of the business. When I got my lucky break, one of the first tasks assigned to me was to replicate file infectors. One morning my boss came in and gave me a new, undetected virus sample. He told me that, although virus samples were usually sent to us by customers, this one had come from a competitor. I was confused. 'From a competitor?', I thought, 'Why on earth would they send us viruses? Won't they lose their competitive edge?'

I kept my questions to myself and analysed the virus, but eventually my curiosity got the better of me. I asked my boss what this 'competitor sharing' was all about. He explained: 'You can't always be the first one to find a new virus; that's why we share them. Ultimately, all of us in this industry have the same goal: to protect people who don't know as much about viruses as we do.' I was fascinated by this idea, and looking back I think this collaborative spirit was one of the main reasons I was drawn to the anti-malware industry.

Editor: Helen Martin

Technical Editor: Morton Swimmer

Test Team Director: John Hawes

Anti-Spam Test Director: Martijn Grooten

Security Test Engineer: Simon Bates

Sales Executive: Allison Sketchley

Web Developer: Paul Hettler

Consulting Editors:

Nick FitzGerald, *Independent consultant, NZ*

Ian Whalley, *IBM Research, USA*

Richard Ford, *Florida Institute of Technology, USA*

As time went by, I was put in charge of collection sharing. In other words, I decided which malware files Avira would share with other security vendors and who those third parties would be. During that time, there were countless presentations and discussions at various conferences and meetings about the sharing of malware and about the possibility of creating a centralized point for doing so. Unfortunately, the industry never came up with a workable solution. Not only were there political issues, but the sharing of malicious files via one centralized point also posed technical challenges. In particular, accommodating the ever-growing volume of malicious files would involve huge hardware costs.

But as time went on, the threat landscape changed. Traditional viruses all but disappeared, and more and more threats began to lurk on the web. This development introduced a new sharing vector: malicious URLs. Soon many companies were sharing malicious URLs in addition to malicious files.

It was after a conversation about our companies sharing URLs at a *Virus Bulletin* conference that Costin Raiu, Tony Lee, Jong Purisima, Nick Bilogorskiy and I decided to revisit the idea of a centralized point for sharing. Creative technicians that we are, we named the project MUTE (Malware URL Tracking and Exchange). MUTE allows a company to consolidate a 1:N relationship (the company shares with many others) into a 1:1 relationship (the company shares with MUTE and MUTE shares with it). While collecting the same data from the other companies, each member additionally benefits from statistics, search functionality, real-time sharing, unified data and so on. The ability to exchange URLs in real time is a particular advantage of the system compared to the way they are shared now, since malicious URLs are usually a time-critical issue.

Since URLs are very small pieces of data, the technical challenges involved in sharing them are minor compared with those of sharing files. We kicked off the MUTE back-end in October at the VB2011 conference in Barcelona. After our presentation, many vendors expressed an interest in joining MUTE – a promising sign for the project!

Right now the MUTE system is in beta. Various companies are testing it for bugs and other problems. We have been pleasantly surprised by the absence of political obstacles – at least from the feedback we have received so far. The members of MUTE are looking forward to seeing the system evolve, welcoming new members and spreading the great spirit of sharing within the anti-malware industry.

NEWS

TECH FIRM TO DEVELOP CYBER WEAPON

Tech firm *Fujitsu* has reportedly been commissioned by the Japanese government to develop malware that will track, identify and disable the sources of cyber attacks.

According to *The Daily Yomiuri*, a three-year project was launched in 2008 to develop the 'cyber weapon' as well as to research and test security tools and network monitoring equipment. The newspaper reports that the new virus – which has undergone testing in a closed network environment – can identify not only the immediate source of attack, but also all 'springboard' computers used in the attack, and that it also has the ability to disable the malicious program and harvest relevant information.

According to officials, the Defense Ministry intends to use the virus for defensive purposes, such as identifying intrusions and tracing the source of attacks against Japanese Self-Defense Force systems.

The development of malware tools for beneficial purposes has long been a controversial topic within the anti-malware industry, there being immense scope for problems – whether due to countermeasures developed by the attackers, or the possibility of the tool falling into the wrong hands. However, several governments (including France and Germany) are already believed to be using specially developed spyware tools to assist in tracking criminals and terrorists, while cyber weapons are said to be in use in countries including the United States and China.

WEAKNESS IN WI-FI ROUTERS EXPLOITED

Two security researchers have demonstrated tools that exploit a weakness in the configuration of most consumer Wi-Fi routers.

The vulnerability, which was first reported by independent researcher Stefan Viehbock, exists in the Wi-Fi Protected Setup (WPS) – a protocol that makes it easier for non-technical users to set up a secure home Wi-Fi network. WPS is enabled by default on the vast majority of consumer Wi-Fi access points.

A few days after reporting the vulnerability, Viehbock demonstrated a tool that can crack a home Wi-Fi network within two hours, while Craig Heffner of *Tactical Network Solutions* (who had been working independently on the same vulnerability), has also developed a tool that will allow access to secure Wi-Fi networks within four to 10 hours.

Currently, most owners of reasonably modern Wi-Fi routers are at risk – it is hoped that the Wi-Fi Alliance and the manufacturers of Wi-Fi access points will work quickly to resolve the issue.

Prevalence Table – November 2011 ^[1]

Malware	Type	%
Autorun	Worm	8.08%
Encrypted/Obfuscated	Misc	5.46%
Heuristic/generic	Virus/worm	5.13%
Salicy	Virus	3.98%
Zbot	Trojan	3.77%
Adware-misc	Adware	3.54%
Crack/Keygen	PU	3.46%
Heuristic/generic	Trojan	3.04%
LNK-Exploit	Exploit	2.95%
Heuristic/generic	Misc	2.91%
Conficker/Downadup	Worm	2.85%
Virut	Virus	2.57%
Cycbot	Trojan	2.57%
Agent	Trojan	2.46%
Downloader-misc	Trojan	2.40%
Iframe-Exploit	Exploit	2.35%
Potentially Unwanted-misc	PU	2.31%
VB	Worm	2.12%
Autolt	Trojan	1.92%
BHO/Toolbar-misc	Adware	1.90%
FakeAlert/Renos	Rogue	1.76%
OnlineGames	Trojan	1.70%
Virtumonde/Vundo	Trojan	1.59%
Dorkbot	Worm	1.46%
FakeAV-Misc	Rogue	1.44%
Kryptik	Trojan	1.40%
Sirefef	Trojan	1.40%
Vobfus	Trojan	1.35%
Delf	Trojan	1.33%
Ramnit	Trojan	1.21%
PDF-Exploit	Exploit	1.17%
Crypt	Trojan	1.11%
Others ^[2]		17.36%
Total		100.00%

^[1]Figures compiled from desktop-level detections.

^[2]Readers are reminded that a complete listing is posted at <http://www.virusbtn.com/Prevalence/>.

MALWARE ANALYSIS 1

THIS SIG DOESN'T RUN

Peter Ferrie

Microsoft, USA

Some virus writers like to brag about themselves or their creations. Sometimes the bragging is done via the virus author's choice of name for the virus. Of course, it's rare that the content justifies the bragging, since lots of viruses contain bugs. Here we have the ultimate combination of bragging and bugs. The author of the virus gave it the name 'Sigrún', which is Old Norse for 'victory rune'. However, there is no victory because the virus does not work (the reason why will not be described here). Just in case the bug is fixed, let's call it W64/Svafa, because 'Sváfa' is the previous incarnation of Sigrún, and the name is thought to derive from the word for 'sleep-maker', which seems appropriate.

Í UPPHAFI (INITIALLY)

The first generation of the virus begins by saving the relative address of the original entry point on the stack. However, depending on the imagebase value that was used when building it, this value might be completely wrong. The virus applies the current imagebase value from the ImageBaseAddress field in the Process Environment Block, in order to account for Address Space Layout Randomization (ASLR). This is an interesting way to deal with ASLR. It is more common simply to calculate the difference between a branch instruction and the host entry point.

The virus also saves the current stack pointer to a field in its body. Using this value the virus can undo any changes to the stack at any point during the execution of the code. This is particularly important during API resolution, since the virus cannot easily know how many APIs have been saved before something goes wrong.

The virus begins by retrieving the base address of ntdll.dll. It does this by walking the InMemoryOrderModuleList from the PEB_LDR_DATA structure in the Process Environment Block. This is compatible with the changes that were made in *Windows 7*. The virus also saves the pointer to the current position in the list so that it can resume the parsing later to find the base address of kernel32.dll. If the virus finds the PE header for ntdll.dll, it resolves the two required APIs: RtlAddVectoredExceptionHandler and RtlRemoveVectoredExceptionHandler.

The virus uses hashes instead of names, but the hashes are sorted alphabetically according to the strings they represent. This means that the export table needs to be parsed only

once for all of the APIs. Each API address is placed on the stack for easy access, but because stacks move downwards in memory, the addresses end up in reverse order in memory. The virus also checks that the exports really exist by limiting the parsing to the number of exports in the table.

The hash table is terminated with a single byte whose value is 0x2a (the '*' character). This is a convenience that allows the file mask to follow immediately in the form of '*.exe'. The virus retrieves the base address of kernel32.dll by fetching the next entry in the list, using the pointer that was saved earlier. The same routine is used to retrieve the addresses of the API functions that it requires, and which is the absolute minimum set of APIs that it needs for replication – find first/next, open, map, unmap, close.

As with previous viruses by the same author, this virus only uses ANSI APIs. The result is that some files cannot be opened because of the characters in their names, and thus cannot be infected. The virus searches in the current directory (only), for objects whose names end in '.exe'. This is intended to be restricted to files, but can also include any directories that have such a name, and there is no filtering to distinguish between the two cases.

For each such file that is found, the virus attempts to open it and map a view of the contents. There is no attempt to remove the read-only attribute, so files that have that attribute set cannot be infected. In the case of a directory, the open will fail, and the map will be empty. The virus registers an exception handler at this point, and then checks whether the file can be infected.

RELOCATION ALLOWANCE

The virus is interested in Portable Executable files for the x64 platform. Renamed DLL files are not excluded, nor are files that are digitally signed. The subsystem value is checked, but incorrectly. The check is supposed to limit the types to GUI or CUI but only the low byte is checked. Thus, if a file uses a (currently non-existent) subsystem with a value in the high byte, then it could potentially be infected too.

The virus checks the Base Relocation Table data directory to see if the relocation table begins at the start of the last section. If so, then the virus assumes that the entire section is devoted to relocation information. This could be considered to be a bug. The virus checks that the physical size of the section is large enough to hold the virus code. There are multiple bugs with this check alone. The first bug is that the size of the relocation table could be much smaller than the size of the section, and other data might follow it. The data will be overwritten when the virus infects the file.

Further, the value in the Size field of the Base Relocation Table data directory cannot be less than the size of the

relocation information, and it cannot be larger than the size of the section. This is because the value in the Size field is used as the input to a loop that applies the relocation information. It must be at least as large as the sum of the sizes of the relocation data structures. However, if the value were larger than the size of the relocation information, then the loop would access data after the relocation table, and that data would be interpreted as relocation data. If the relocation type were not a valid value, then the file would not load. If the value in the Size field were less than the size of the relocation information, then it would eventually become negative and the loop would parse data until it hit the end of the image and caused an exception.

The second bug is that by checking only the physical size and not the virtual size, whatever the virus places in the file might be truncated in memory if the virtual size of the section is smaller than the physical size of the section. Both of these bugs are also present in the W64/Holey virus [1], by the same virus author.

However, it is the third bug that is very serious, very silly, and should have been very obvious. It is that the size of the virus code is less than one third of the total size that is needed to hold the data that the virus adds to a file. The true size of the virus is the size of the virus code multiplied by three, plus the size of the decoder. Thus, the section might be nowhere near large enough for the file to be infected correctly, but the virus won't notice the problem.

I'M LOOKING FOR... MY MASK!

If the section appears to be large enough, then its attributes are marked as executable and writable. The virus 'encrypts' its code using a byte-mask technique. There are two tables involved here, both of which are the same size as the virus code. One table contains the byte-mask, and the other contains a selection of host bytes. For each byte in an eight-byte set, the virus chooses randomly if it will be encoded or not.

If the byte is to be encoded, then the byte-mask will have the top bit set in the mask table, and the other seven bits will be set to a random value. The corresponding entry in the host table will contain the host byte, and the value at the original location in the host will be set to a random value.

If the byte is not to be encoded, then the byte mask will have the top bit clear in the mask table, and the other seven bits will be set to a random value. The corresponding entry in the host table will contain a random value, and the value at the original location in the host will maintain its original value. This process is repeated over the entire host body.

Once the encoding is complete, the virus zeroes the values in the Offset and Size fields of the Base Relocation Table

data directory, saves the original entry point in the virus body, and then sets the host entry point to point directly to the virus code.

TOUCH AND GO

The virus code ends with an instruction to force an exception to occur. This is used as a common exit condition. However, the virus does not recalculate the file checksum, even though it might have changed as a result of infection. It also does not restore the file's date and timestamps, making it very easy to see which files have been infected, even though the file size does not change.

I LIKE TO 'MOV' IT

When an infected file is executed, the virus decodes itself. The special thing here is that the decoding is done using three MMX instructions, one of which might be considered to be a bit obscure: MASKMOVQ. The MASKMOVQ instruction accepts two parameters which correspond to the two tables that the virus constructed. For each byte in the mask table whose high bit is set, the corresponding byte in the host table is copied into the host body at the location to which the EDI register points at that moment. If the high bit is clear, then no copy occurs. Thus, prior to decoding, the virus body is a mixture of real values and random values, and so is the host table.

There have been suggestions that MMX is not safe to use on the 64-bit platform, because the floating-point state (and thus the MMX state, which shares the same memory region) is not saved, but this is not the case. In user mode, the floating-point state is always saved during a context switch. Therefore, there is no problem at all for user-mode processes. In kernel mode, the context is not saved, but it was not saved on the 32-bit platform either, so there is no new behaviour here.

CONCLUSION

The MASKMOVQ technique is another surprise from the MMX instruction set, and one which makes static analysis a bit inconvenient. However, anti-malware emulators will see just another instruction, and shortly afterwards, just another virus.

REFERENCES

- [1] Ferrie, P. 'Holey' virus, Batman! Virus Bulletin, September 2011, p.4. <http://www.virusbtn.com/pdf/magazine/2011/201109.pdf>.

MALWARE ANALYSIS 2

DISSECTING THE NGR BOT FRAMEWORK: IRC BOTNETS DIE HARD

Aditya K. Sood, Richard J. Enbody
Michigan State University, USA

Rohit Bansal
SecNiche Security, USA

IRC-based botnets [1, 2] have become the preferred choice of bot herders for remotely managing bots. IRC networks provide anonymity during communication, which makes tracking their activity more difficult. The IRC network is used for sharing files, controlling network activity and sending distributed commands to networks of infected machines. The IRC network is comprised of dedicated servers that use specific communication channels. As a result, it is possible to control a large number of infected machines through a centralized space (IRC channel) to create a complete botnet. Basically, the bot is compiled with a configuration which has a predefined IRC channel name. Once the bot is installed, it connects back to the IRC channel and the bot

herder is able to send commands through that channel to operate the bot remotely. IRC-based botnets are popular for conducting Distributed Denial of Service (DDoS) attacks [3, 4]. However, the latest variants of IRC-based botnets, such as NGR, are designed to steal sensitive information by exploiting browser processes and acting as a backdoor. In this paper, we discuss the framework of the NGR bot version 1.1.0.0 which is growing in prominence in the malware world. The workings of the IRC bot are presented in Figure 1.

UNDERSTANDING THE FRAMEWORK

In this section, we present the design and analysis of the NGR bot framework. The framework consists of the bot executable with built-in modules. The design of the framework is discussed next.

RING 3 BOT

The NGR bot is a ring 3 bot that works in user space. The bot is written using the standard Visual Studio development kit. It has the characteristics of user-land rootkits [5, 6] and follows a similar process of DLL injection and hooking

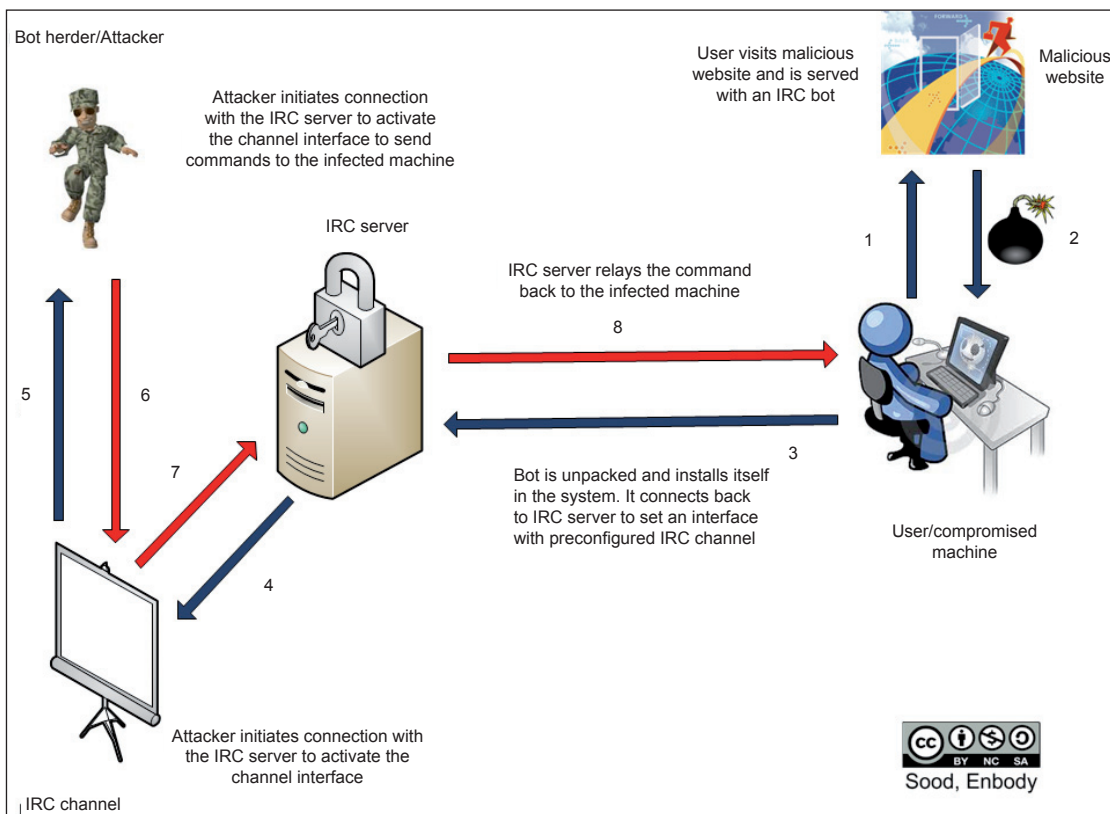


Figure 1: Working of IRC bot.

to infect the running processes in the system. The DLL injection is a system-wide operation and is not restricted to the browser process. That is, the NGR bot is capable of injecting content into any process in the system. The NGR bot exhibits some properties that are found in third generation botnets such as Zeus and SpyEye. The bot is designed to infect 32-bit processes and does not support 64-bit injections at this point. However, the bot can successfully be installed in versions of *Windows* including *XP*, *Vista*, *Windows 7* and *Windows Server*. From the design of the NGR bot we expect that upcoming versions will include full support for injecting into 64-bit processes. Figure 2 shows the layer model of the *Windows* operating system and where the bot infects it.

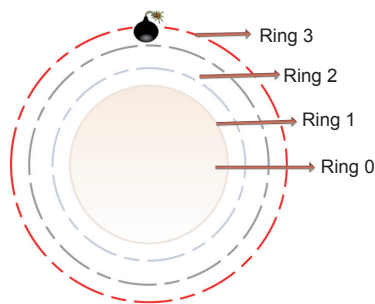


Figure 2: Ring 3 layer of OS and bot infection.

Like all bots, the NGR bot is designed to operate in a covert manner. It can be installed in *Windows* using any user account without administrative privileges. The bot is capable of sending back information about the access rights of the victim's account on the system through the IRC channel. The NGR bot's primary installation location is the user's application directory. The bot installs itself there under a randomly generated name so it varies from machine to machine. The bot is activated in the machine after a reboot. It ensures execution by creating a registry key with a path directly to the bot's binary (under its random name). As the system is restarted, the NGR bot connects back to the IRC channel through a relay server and then becomes ready to receive commands, as shown in Figure 3.

The bot sends information back to the server in the following format:

```
n{RU|XPa}kdsfkksd
```

The bot herder can decipher the information based on the string shown above. The 'n' parameter indicates that the bot is installed on a new machine. The 'RU' indicates that the victim machine is located in Russia. The 'XP' string indicates that the infected operating system is *Windows XP*. The 'a' parameter shows that the bot is installed using an account that has administrative rights. The string

'kdsfkksd' is the identity of the bot generated in a pseudo random manner. The following shows how the bot sends information back to the IRC channel:

```
<new>{<COUNTRY>|<OPERATING SYSTEM><user type>}<random letters>
```

Figure 4 shows the NGR bot in action. The '~version' command can be used to show the identity information from the NGR bot.

The bot is designed to communicate over SSL using an IRC channel. In order to set up the SSL communication, the IRC server must be configured to initiate an encrypted channel with the NGR bot. The bot binary is optimized and has a built-in module for testing the connection speed. The '+speed' command on the IRC channel can be used to measure the speed for exchanging data. All the Inter Process Communication (IPC) among processes is encrypted. The NGR dropper deletes itself once the bot is successfully unpacked and installed on the victim's machine. The bot also changes the extension of files to '.exe' so that other files such as *.vbs can be executed successfully to trigger infections.

We have described the NGR bot functionality above. In the next sections, we will present the rest of the NGR bot framework.

GRABBERS – BROWSERS, FTP & POP3

The NGR bot uses a form-grabbing module to extract sensitive information from the victim's machine. The latest version of the NGR bot is robust enough to execute hooking in both *Internet Explorer* and *Mozilla Firefox*. *IE* uses *wininet.dll*, whereas *Firefox* uses *nspr4.dll* for HTTP communication. The NGR bot hooks various functions in these libraries and captures the GET/POST requests to extract credentials in the forms. Since it uses form grabbing, the NGR bot does not have a keylogging module. We presented details of the form-grabbing technique in [7]. Additionally, the NGR bot has a built-in FTP grabber module that hooks the *ws2_32.lib* functions to extract the credentials for various FTP servers. Finally, a POP3 grabber module works in a similar way to the FTP grabber module.

Figure 5 shows how the NGR bot sends credentials back to the IRC server.

On the IRC channel, the '~logins' and '~stats' commands show the number of fetched credentials and related stats.

SPREADERS

Spreader modules are used to spread the botnet across a variety of interfaces of the victim machine. These interfaces include USB devices and Instant Messengers (IMs) such



Figure 3: NGR bot connecting back to IRC channel.



Figure 4: NGR bot sending identity information after successful installation

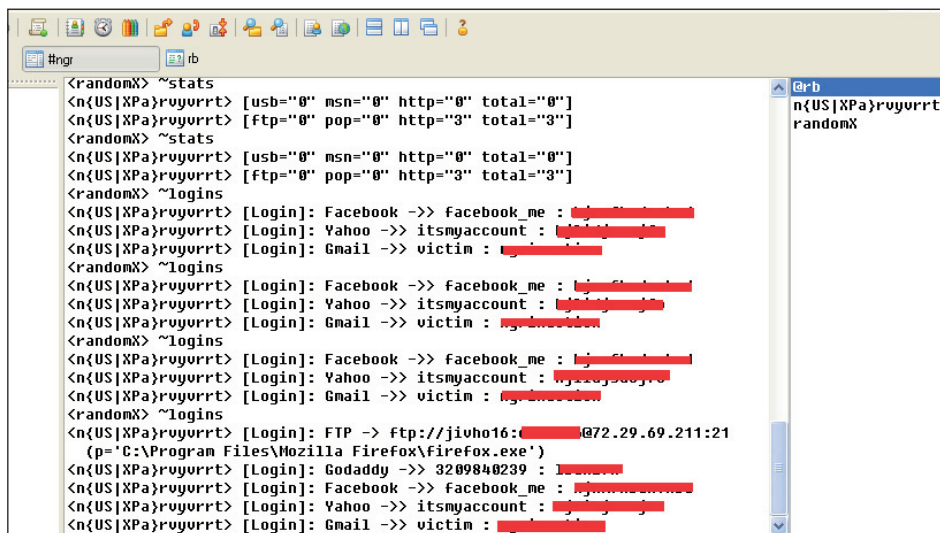


Figure 5: NGR bot sending form-grabbing credentials.

as *MSN*. The NGR bot has the following built-in spreaders.

USB spreader: The NGR bot infects USB devices and replicates itself on them. Once the victim machine is infected with the bot, the built-in USB spreader module waits for USB devices to be inserted and tries to infect them. The USB spreader module uses a linking technique in which .lnk files are inserted into the USB drive with a path to the NGR bot. A desktop.ini file is also created to hide the folder in which the bot resides. In addition, the NGR bot is able to infect USB drives using an obfuscated autorun.inf method. This method can be activated using the '~mod usbi' command on the IRC channel. This module works on all versions of *Windows*.

MSN spreader: The NGR bot also has a built-in MSN spreader module that hooks the ws2_32! send function to detect *MSN* messages being sent. The spreader module monitors the *MSN* communication channel and waits for a certain set of messages so that it can start injecting illegitimate messages. The spreader can successfully inject processes such as msmmgr.exe, wlcomm.exe, pidgin.exe and msmsgs.exe using protocols msnp10 and msnp21. The '~msn.int' and '~msn.set' commands are defined in the NGR bot for this purpose.

DNS MODIFIER

DNS entry modifications are an important part of the NGR bot. Generally, DNS modification can be achieved in two ways:

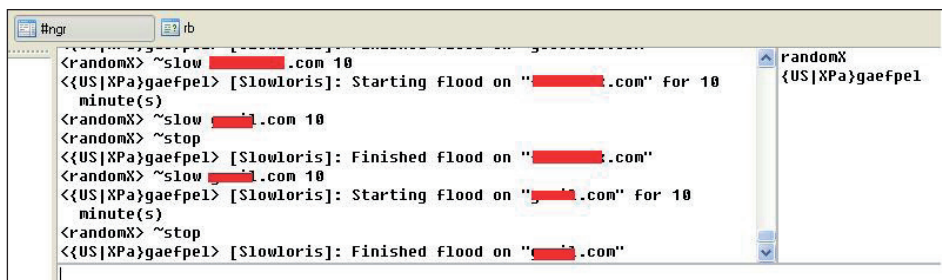
- **Updating the hosts file** – The bot can update the entries



```

#nrg  rb
<randomX> ~chdns http://www.example.com
<<US|XPa}gaeffe1> [DNS]: Blocked 0 domain(s) - Redirected 18 domain(s)
<randomX> +modns
<randomX> +modns mail.example.com
<randomX> ~chdns gooooooogle.com
<<US|XPa}gaeffe1> [DNS]: Blocked "gooooooogle.com"
<randomX> ~chdns yahoooooooooo.com
<<US|XPa}gaeffe1> [DNS]: Blocked "yahoooooooooo.com"
<randomX> ~chdns gooooooogle.com yahoooooo.com
<<US|XPa}gaeffe1> [DNS]: Redirecting "gooooooogle.com" to "yahoooooo.com"
<randomX> ~chdns *.asd *.jkl
<<US|XPa}gaeffe1> [DNS]: Redirecting "*.asd" to "*.jkl"
~chdns
  
```

Figure 6: DNS modifier in action.



```

#nrg  rb
<randomX> ~slow [redacted].com 10
<<US|XPa}gaeffe1> [Slowloris]: Starting flood on "[redacted].com" for 10
minute(s)
<randomX> ~slow [redacted].com 10
<randomX> ~stop
<<US|XPa}gaeffe1> [Slowloris]: Finished flood on "[redacted].com"
<randomX> ~slow [redacted].com 10
<<US|XPa}gaeffe1> [Slowloris]: Starting flood on "[redacted].com" for 10
minute(s)
<randomX> ~stop
<<US|XPa}gaeffe1> [Slowloris]: Finished flood on "[redacted].com"
  
```

Figure 7: Slowloris in action through IRC channel.

in the hosts file in order to manipulate the DNS resolution.

- **Hooking dnsapi.dll** – The bot can hook the required DNS DLL file and manipulate the entry present in the rule file.

The NGR bot hooks the dnsapi.dll file to modify the DNS entries on the victim's machine. This module is incorporated in the NGR bot so that virus detection websites such as *VirusTotal*, *Kaspersky*, and so on can be blocked on the host. The DNS modifier is also capable of setting a DNS redirection so that a legitimate website's address is mapped to an illegitimate one. This feature is used to serve malware. The '~chdns' command is used to perform those actions. Figure 6 shows the DNS modifier in action.

PROACTIVE DEFENCE (PDEF+) AND RUSKILL

Bot wars are on the rise as large botnets compete to infect the same computers. The NGR bot has a built-in module that kills other installed IRC bots in the system. PDEF+ is an active threat detection module that monitors and scrutinizes the various APIs and the file system to detect and remove infections. This module can detect and block malware that has been distributed using USB drives, IRC bots and browser exploits. The NGR bot has modified this module to detect and kill the butterfly bot, butterfly flooder, GBOT and all other IRC-based bots.

The built-in Ruskill module is designed to stealthily execute files. The NGR bot has the built-in command '~baja' which is used to download malicious executables from a remote website. The downloaded binary executes automatically and triggers infection. The Ruskill module monitors the downloaded binary and flags it, then deletes the binary on system reboot. This functionality is widely used by IRC bots to remove downloaded files after execution.

DENIAL OF SERVICE (DOS)

Denial of Service is a primary functionality of IRC bots so the NGR bot is well equipped

with DoS modules:

- **SYN flooder:** The bot sends a continuous flow of TCP packets with the SYN flag. The SYN flood can take down web servers that other flooders fail to.
- **UDP flooder:** The bot sends a continuous flow of UDP packets to take down the target. This module is designed to target small networks.
- **Slowloris:** The bot has a built-in Slowloris module [8]. This conducts DoS attacks against *Apache* web servers in which the module opens many connections to the web server and holds them open for a long period of time. As a result, the target website's connection pool becomes exhausted because the connection remains open and no new connections can be served by the web server.

Figure 7 shows the working of a Slowloris module.

CONCLUSION

In this paper, we have presented a detailed framework of the NGR bot and the different types of modules it supports. The NGR bot has been widely used to trigger infections and compromise machines. Our analysis has revealed that the bot is very effective and capable of running in a concealed manner. Looking at the development of the NGR bot, we can expect further advancement such as support for hooking 64-bit processes, *Facebook* IM spreaders and so on in the

near future. The sophisticated framework of the NGR bot indicates that IRC-based botnets will continue to be a hard nut to crack.

REFERENCES

- [1] Characterizing the IRC-based Botnet Phenomenon. <http://honeyblog.org/junkyard/reports/botnet-china-TR.pdf>.
- [2] A Review on IRC Botnet Detection and Defence. http://www.kaspersky.com/images/waldecker,_bernhard_-_a_review_on_irc_botnet_detection_and_defence-10-98487.pdf.
- [3] Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures. <http://palms.ee.princeton.edu/PALMSopen/DDoS%20Final%20PDCS%20Paper.pdf>.
- [4] Distributed Denial of Service Attacks. http://www.ensc.sfu.ca/~ljilja/papers/smc00_edited.pdf.
- [5] Introducing Ring-3 Rootkits. <http://invisiblethingslab.com/resources/bh09usa/Ring%20-3%20Rootkits.pdf>.
- [6] When Malware meet Rootkits. <http://www.symantec.com/avcenter/reference/when.malware.meets.rootkits.pdf>.
- [7] Sood, A. K.; Enbody, R. J.; Bansal, R. The art of stealing banking information – form grabbing on fire. Virus Bulletin November 2011, p.19. <http://www.virusbtn.com/pdf/magazine/2011/201111.pdf>.
- [8] Slowloris HTTP DoS. <http://ha.ckers.org/slowloris/>.

Appendix – NGR bot commands

Command	Options	Details
~baja	<url>,<md5>,<-r>,<-n>	Bot downloads and executes a file from the specified URL
~updt	<url>,<md5>,<-r>	Bot updates its file, but the update does not take effect until the system is restarted
~mata		Bot disconnects from the IRC server
~l1mpia		Bot removes itself from the system
~mudo	[state]	Enables/disables all output to IRC relating to commands and features
~version		Bot displays its version, customer name, the MD5 hash of its file, and its installed file path
~v1sit	[url][state]	Bot creates a browser instance and visits the specified link
~rc	<-n>,>-g>	Bot disconnects from the IRC server and waits 15 seconds before reconnecting
~move	<rule>,<options><channel>,<key>	Bot joins the specified channel
~p4rt	<rule>,<options><channel>	Bot leaves the specified channel
~pais	<rule>	Bot joins the channel for its country
~mix	<rule>	Bot leaves the channel for its country
~speed		Bot determines the average upload speed
~mod	[module], [state]	Enables/disables modules that use hooks
~stats	<-l>,<-s>	Retrieves statistics for spreading and/or login grabbing
~logins	<site>,<-c>	Retrieves all grabbed and cached logins
~stop		Bot ends all running flood tasks
~ssyn	[host],[port],seconds]	SYN flooder
~udp	[host],[port],seconds]	UDP flooder
~msn.int	[interval]	Sets the number of MSN messages in a conversation before one is changed with the spreading message
~msn.set	[message]	Sets the message that will be used for MSN spreading
~chdns	[url][domain1 <domain2lip2>]	Bot blocks access to or redirects the specified domain/IP address

FEATURE

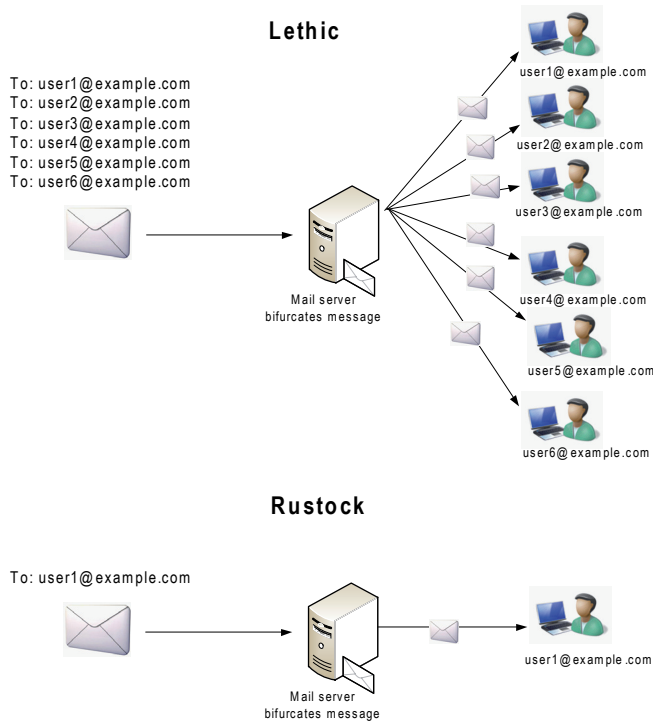
THE TOP 10 SPAM, MALWARE AND CYBERSECURITY STORIES OF 2011

Terry Zink
Microsoft, USA

2011 was filled with plenty of security stories. It wasn't just spam that made the news, but spam, malware, hacking and more¹. It was a jam-packed year, so let's take a look at the biggest newsmakers. (Please note that the views and opinions expressed in this article are my own and do not necessarily state or reflect those of *Microsoft*.)

1. MICROSOFT SHUTS DOWN RUSTOCK

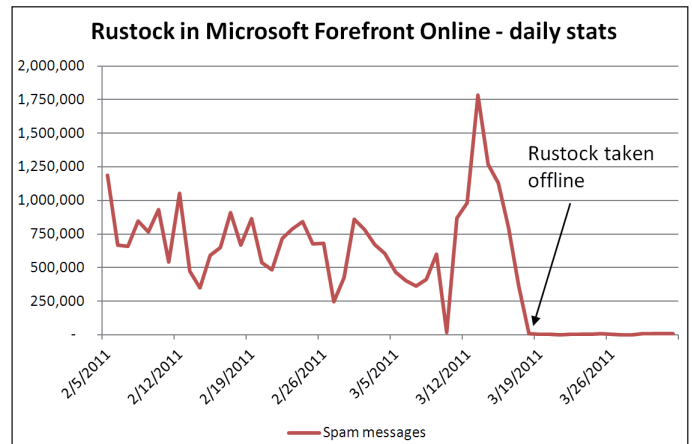
For several years, the spamming botnet with the biggest footprint was Rustock. Its characteristics were to 'wake up' at a specific time each time day, send hundreds of thousands of spam messages, and go back to sleep. Furthermore, Rustock sent lots of messages to lots of different email users from a lot of IP addresses. In other words, its footprint



¹ When I use the term 'spammers' in this article, I use it in a generic sense to refer to people who spam, distribute malware, perform black search engine optimization, etc.

was a mile wide and an inch deep. This distinguished it from some other botnets like Lethic that send a lot of email in one Internet connection (that is, an email with lots of recipients vs Rustock that sends to one recipient per email).

But on 16 March 2011, working with *Microsoft*, *Shadowserver* and some other partners, the US Department of Justice obtained a court order to seize command-and-control servers that were responsible for running the Rustock botnet in the United States. Virtually overnight, spam from Rustock plummeted and has never recovered:



It was a great takedown, federal authorities walked into server rooms and confiscated actual hard drives from command-and-control servers (a warrant was obtained and the action was coordinated across a couple of US states).

Microsoft had to make an effort to serve the operator of the botnet in court and give him a chance to explain himself. Unsurprisingly, he never showed up.

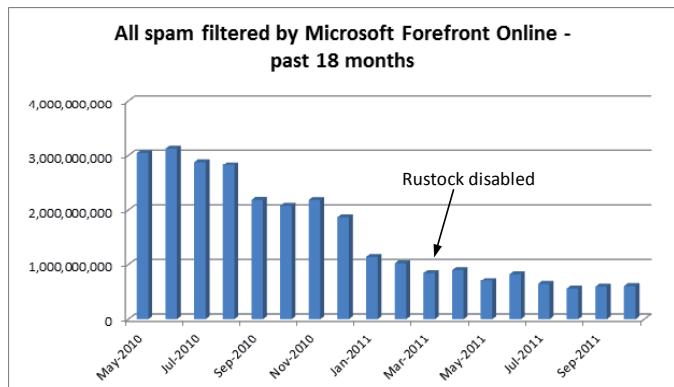
2. SPAM VOLUMES DROP

For years, we have heard about the scourge of spam:

- Spam volumes way up!
- 95% of all email is spam!
- Spam volumes *continue* to increase!

The trend was so bad for so long that everyone began to wonder whether spam would eventually become 99% of email, and then 100% (rounded up). The amount of processing resources it would take up would threaten to ruin email (which is why other technologies like RSS, instant messaging, and social networks always promise to replace email... but never do).

But, starting in late 2010 and continuing throughout 2011, spam levels started to decline. And they didn't just decline a little, they declined *a lot*.



What caused this steep decline? The answer is that nobody knows for certain, although there are theories:

- In late 2010, spam affiliate network SpamIt (also known as GlavMed) shut down and some of the revenue streams for spammers disappeared.
- In March 2011 the Rustock botnet was disabled. However, while this was significant, the decline in spam had already begun before the Rustock takedown.
- In response to botnet takedowns, spammers have adjusted their strategy – rather than sending a lot of email to a lot of users, they are keeping their botnets smaller and creating more direct spam campaigns.
- Pavel Vrublecky, the founder of Chronopay, which is allegedly responsible for online spammer payments, was arrested (on non-spam-related charges). Payment processing for spammers faced a new bottleneck.
- Spam from botnets still dominates the spam waves, but ‘grey mail’ spam – pseudo-legitimate bulk mail campaigns – have arisen and are more difficult to stop.

The battle against spam isn’t over, it has just shifted from one form to another. Now the problem of spam isn’t that it gets through because there’s so much of it that filters can’t keep up, but that filters can’t keep up because its smaller numbers are more difficult to detect.

3. RSA HACKED

In March, some disturbing information leaked out of *RSA*, the company that has long been associated with security. These are the guys that make the key fobs that many of us use to get onto our corporate networks using two-factor authentication.

Somebody, somewhere, sent an email to an *RSA* employee – not a high-level employee, just a regular ham-and-egger like you and me. The email came with an attachment. The subject line read something like ‘2011 Recruitment Plan’. It went into the employee’s junk mail folder.

The employee opened their junk folder, dug the email out, opened the attached *Excel* file, and their computer became infected with a piece of malware when it exploited a previously undocumented *Flash* vulnerability. The intruders were now inside the company’s systems.

It is unclear what the intruders were able to access because *RSA* representatives haven’t exactly been forthcoming with a full list of what was compromised and what wasn’t (not that I can blame them). But what we think we know is that it was a very sophisticated hack, and the hackers accessed the algorithm that *RSA* uses on its key fobs, as well as the seeds for the encryption algorithm.

RSA offered to replace the fobs of its customers and then issued the standard set of advice: use strong passwords, control access to the production environment, and re-educate employees about opening email messages with suspicious attachments.

RSA wasn’t the only big-name corporation to be hacked using a sophisticated attack this year. Large government contractors like *Lockheed Martin* and *Seagate Technology* were also hit, as well as the US Internal Revenue Service and *Freddie Mac*.

Who was behind all of these attacks? Well, that’s the subject of the next big story of 2011.

4. OPERATION SHADYRAT

In September 2011, *McAfee* released a report, in conjunction with *Vanity Fair* magazine [1], about Operation ShadyRAT.

In the report, *McAfee* studied several cyber intrusions where numerous victims had been targeted – government agencies in the United States, Taiwan, South Korea, Canada; large corporations in a number of countries; and non-profits such as the International Olympic Committee.

Why were these entities targeted?

There wasn’t a clear financial motive behind the attacks, no economic gain. Generally, if a cybercriminal hijacks your computer, he either wants your usernames and passwords so he can steal money from your back account, or he wants control of your computer to use it as part of a botnet.

Who was behind the attacks? *McAfee* wouldn’t point a finger at any particular culprit, the researchers just presented the evidence and then let the policy makers draw their own conclusions. However, at least one security expert believes the identity of the culprits is obvious: ‘All the signs point to China,’ says James A. Lewis, director and senior fellow of the Technology and Public Policy Program at the Center for Strategic and International Studies, adding, ‘Who else spies on Taiwan?’

Regardless of who is behind the attacks, 2011 saw a huge increase in the detection of APTs – Advanced Persistent

Threats. An APT is a hacking attack that is advanced because it contains sophistry that is beyond the abilities of most hackers. It is persistent because it occurs over and over again. APTs are usually attributed to other countries – first, because states are the only entities that have the resources to continually fund these types of threats, looking for vulnerabilities in software. Second, because the motive for APTs is unclear and looks more like espionage, and it is assumed that states have more use for it than private enterprise.

What Operation ShadyRAT exposed was the size and scope of this type of activity. And while it is fun to accuse China of being behind all these hack attacks, the fact is that there are many countries that all have varying degrees of skill when it comes to collecting intelligence. And that's what APTs are all about – espionage. The United States, China, and Russia are all very good at collecting intelligence. They have well-funded government departments dedicated to acquisition of information. Other countries – such as Ireland or New Zealand – do not have so many dedicated resources.

The arms race has been going on for years. What changed in 2011 was that we became very much more *aware* of what is going on.

Maybe we liked it better when we didn't know so much.

5. LULZSEC HACKS THE WORLD, SO DOES ANONYMOUS

Hacking isn't all fun and games, as illustrated by my previous two stories. But this next story kind of *is* fun and games. Unless you were one of the targets, that is.

Anonymous is an international hacking group, spread through the Internet, initiating active civil disobedience while attempting to maintain anonymity. *CNN* has referred to the group as one of the successors to *WikiLeaks*.

In April, *Sony* revealed that it had been the victim of a hack where a raft of user information was stolen from its servers and posted online. Anonymous had previously announced that it had planned to do this in response to *Sony's* lawsuit against George Hotz (a young hacker who had hacked into the *PlayStation*). Further attacks were carried out against the Spanish Police (DDoS), websites of the Malaysian government, and against the Orlando Chamber of Commerce.

In June, a suspected splinter group of Anonymous – LulzSec – launched its own hacking attacks. The group went after the websites of the CIA, *Sony*, *Fox News*, *PBS* and others. During the summer, it seemed like there was a new hack attack each week and what became newsworthy was not the fact that a company had been hacked, but that a week had gone by without an attack.

The hack attacks of LulzSec and Anonymous eventually quietened down. In contrast to the APTs that continue to occur to this day, LulzSec and Anonymous eventually wound down their campaigns (perhaps in response to generating too much heat, or perhaps in response to law enforcement catching up with some of the members of their group). But also unlike the APTs, these hacking incidents were very public. Both groups claimed responsibility for their break-ins and revelled in the attention they received.

The motives of these two attacks are very different from the APTs; whereas states desire the acquisition of information, Anonymous and LulzSec are driven by hacktivism. Their motives are political in nature and are designed to poke fun at public figures with whom they disagree. *Sony* acting as a bully by infringing on users' 'rights' to hack the *PlayStation*? Or the Arizona state government building a fence to keep out illegal immigrants? What better way to protest than breaking into their computer networks!

Not all of the hacks were politically motivated, though. By their own admission, some of the groups' activities were carried out through sheer bravado.

The problem with bravado is that if you generate too much publicity, eventually the law catches up with you. In stock trading, there's an old saying: 'There are old traders, and there are bold traders, but there are no old, bold traders.'

LulzSec and Anonymous would do well to take heed of that.

6. MALWARE APPEARS FOR MACS

For years, *Microsoft* has devoted resources to stamping out the problem of malware that infects its operating systems. Fair or not, *Windows* has earned the reputation of being insecure and susceptible to malware. In contrast, *Apple* has historically prided itself on the belief that it is *not* prone to malware. Television commercials have even been released implying exactly that.

But this year, something odd happened – malware started to appear for the *Mac*. And users were falling for it.

It's not that malware has never existed for the *Mac*; it has. It's just that this year it became really obvious that malware exists for the *Mac* because of its prevalence. *Reuters* announced [2] in May that *McAfee* had seen a steady stream of malware related to the *Mac*, and users called in asking for assistance on how to get rid of it. At first *Apple* denied that this was a problem [3]. For a company that is as image-conscious as *Apple*, I can see how it might be in denial that its computer wasn't as perfect as it thought it was. Eventually, the company released software that removed that malware and included it as an update to *MacOS*.

More malware variants did not keep popping up, leading some to think that this was just a temporary phenomenon. Those of us in the PC industry don't agree, though. In fact, some of us experienced just a little bit (i.e. a lot) of schadenfreude at this turn of events. Not because we're happy that the *Mac* has got malware, only that outspoken *Mac* users were put in their place just a little bit².

The good news in all of this is that *Apple* responded to the problem fairly quickly and is now starting to get into the rhythm of security releases. Like anything, once you become popular, you become a target for online criminals. And that brings me to the next story.

7. MOBILE MALWARE GAINS TRACTION

The post-PC era. That's the new buzzword these days. Personal computers are getting smaller and smaller. First we had huge machines with vacuum tubes. Then we invented transistors and the machines became smaller so that they could fit on a desk. Then they fitted onto our laps. Now, they fit into our hands.

Remember when a phone was just a phone? You used it to call your friends. Then they started doing things like browsing the web and playing music. Now they do a whole lot more, and you can *almost* put your entire life on your smartphone (of course, there's so much more to life than playing *Angry Birds* – but I digress).

And that's just it. Our phones are now doing more and more, which is why we call them 'smartphones'. They are like miniature computers that we carry around with us, but, like yesterday's computers, they are prone to the same type of problem – malware.

Mobile malware is still in its infancy and hasn't yet taken off fully, but it is growing. And just as the PC became a target for malware writers because of its ubiquity and open systems (anyone could write applications), smartphones are becoming targets for malware writers where the systems are ubiquitous and the platform is open. And the fastest growing target is *Google's Android*.

There are two main types of malware for *Android*:

1. SMS trojans that send background messages over your phone once it is infected.
2. Data theft trojans that steal your personal information and send it back to a remote server.

Does this mean that smartphones, particularly ones that run *Android*, are especially vulnerable? Not exactly. The smartphone space is still fairly new, and luckily, that

²I'm having fun with *Mac* owners, but that's okay because I am one, too.

still works in its favour. As long as you buy or install applications from reputable places, you are not going to have much of a problem (most of the time). Unlike the *iPhone*, *Android* doesn't have a central clearing house for apps and users are fooled into thinking that everything is secure, when it really isn't. Thus, while *Android's* openness is its strength, it is also its weakness. This is *exactly* like *Windows*. But the prevention is also exactly the same as for *Windows* – make sure you buy or install applications from reputable places.

There really is nothing new under the sun. And speaking of nothing new, that leads me to my next story.

8. THE THREAT OF ZERO-DAY MALWARE IS OVERSTATED

Twice per year, *Microsoft* releases its Security Intelligence Report [4]. Each time it releases one, it writes about a particular theme and the latest one, SIR volume 11, released in October, looked at zero-day malware.

In the security industry, zero-day malware gets a lot of hype. Users are afraid of it – they fear that one day they will be hit by a virus infection that will spread through their computer and erase all of their files or infect their computer and steal all of their data. Worse yet is that they can have all of the possible security solutions in place, but *nothing* will protect them because there are no anti-malware signatures for zero-days. There is no defence!

Microsoft decided to examine the threat of zero-days and assess whether the fears were justified, and to put this type of threat into context relative to all of the other threats that are out there.

The analysis revealed that zero-day threats account for far fewer than 1% of all malware threats out there. The fact is that for all of the hype that exists, for the vast majority of threats there are defences already in place. You can protect yourself if you just follow basic steps like keeping your software up to date, running a firewall, and running anti-virus software.

The report was not intended to play down the risks posed by zero-day malware or to encourage IT administrators to relax and let down their guard. Instead, the message is that we all have a limited set of resources and need to prioritize the tasks we do and where we allocate those resources. Having the right information lets us keep our priorities straight. Users are the most vulnerable part of any security system, so maybe resources are better spent educating users, implementing secure practices and following basic security steps. Many of these practices are pretty easy to implement. You can action them today.

And that's the point of the report.

9. THE UNIVERSITY OF CALIFORNIA: GO AFTER THE BANKS TO STOP ONLINE CRIME

My favourite story of the year is this one. Why? Because it offers a genuinely new method of fighting online crime.

For years, the anti-spam and anti-malware industry has focused on technical solutions to the problem of online abuse. Most of the top stories of 2011 are about that. Heck, it makes sense; we have to use the tools we have available, and technical solutions are how we solve technical problems.

These solutions are augmented with user education (teaching users not to fall for scams) and criminal prosecution of online criminals. The former recognizes that scams are a human problem, and so does the latter.

Where the University of California (UC) offers a unique insight is in looking at the motivation of criminals – for the money. What if instead of going after the criminals, we dried up their money supply? The security industry insists that if people stopped buying the stuff spammers are peddling, eventually they would go out of business (the same as any other enterprise). But if people won't stop buying the stuff, then maybe we can cut off the criminals' supply of money.

The University of California started buying from spam campaigns and then looked at the credit card transactions. Most people are only vaguely aware of how this process works. You go to a website, enter your credit card details, and like magic you receive your goods a few days or weeks later. You then happily pay your credit card bill where said transaction ends up on the statement.

What really happens when you initiate a credit card transaction is that the big companies like *Visa* or *MasterCard* act as a clearing house. The financial transactions really go through the banks. They are the ones that authorize or clear these transactions, and they are the ones with the authority to approve, or more importantly **block** these transactions. The UC researchers discovered that the majority of credit card payments for spammy products went through three banks – one in Azerbaijan, one in Latvia, and one in St. Kitts and Nevis. The banks are a major bottleneck in getting money to scammers.

Instead of going after users or using technology to keep users safer, the researchers proposed pressuring banks to clamp down on fraudulent banking transactions. After all, if three banks are responsible for three quarters of spam payments, then shutting those down would represent a real disruption to the spam business.

Moving the online abuse infrastructure is relatively easy. It's not difficult to register domains or set up botnets. However,

it is time consuming and costly to negotiate payments with banks. Spammers cannot simply pick up and move banks the way they can with domains. It is a human process that has checks and balances.

And that is why this is my favourite story. I had never thought of this before. If automation is the key to spamming, then making spammers go through manual steps is a great way to disrupt their cash flow patterns.

Maybe there's hope yet for solving the abuse problem.

10. FACEBOOK TOPPLES GOVERNMENTS... RIGHT?

One of the biggest geopolitical stories this year was the Arab Spring where popular uprisings led to changes in leadership in Tunisia and Egypt. Since that time, there has been debate as to the role that social media played in the toppling of the governments.

At the time, it certainly looked like *Facebook* and *Twitter* played a significant role since they were major organizing tools for the demonstrators. By creating user groups, fan pages and trending topics, and then using them to organize planned demonstrations, protesters created a decentralized movement that eventually forced change. I recall one article where an interviewee said 'Thank you, Mark Zuckerberg!'. The government of Egypt cut off Internet access for a period of time, but one study [5] by the University of Washington concluded that this only increased support for the movement. After all, an oppressive regime that suppresses freedom of expression deserves removal. This only fanned the flames of revolution.

The point was driven home: we live in a new age where the Internet and social networks have the power to topple oppressive regimes.

The problem is that the role of *Facebook* and *Twitter* was not as pronounced as some of us think. Numerous blog writers and editorials claim that *Facebook's* role in the Arab Spring was overstated. Even *Facebook's* VP for Advertising and Operations commented, saying 'When you see what is happening, you understand why changes are happening within the social media. However, I think *Facebook* gets too much credit for these things.' [6]

Behind the scenes, there were other political forces at work. In both Tunisia and Egypt, the military forced the respective presidents to step down because they either disapproved of their policies (in the case of Tunisia) or of their succession plans (in the case of Egypt). As evidenced by other protests across the region in other countries, protests do not always lead to regime change. Make no mistake, social networks

give visibility to social movements, but real action requires the strength of the military.

Thus, while *Facebook* did have a small role to play in the Arab Spring, it had a lot of help from the traditional arbiters of power.

11. OPERATION GHOST CLICK

Yes, I said that these were the top ten stories of 2011, and here I am at number 11. Consider this a bonus story.

In November, the FBI announced the arrest of six Estonian nationals in what some call the biggest cyber-heist arrest in history. Responsible for the DNSChanger malware, which redirected unsuspecting users to rogue Internet pages, the botnet that the suspects operated consisted of over four million computers.

As with Rustock, the rogue command-and-control servers were seized and infected requests to DNS servers were replaced with legitimate ones.

Not a bad Christmas present for law enforcement.

CONCLUSION

Well, that's the way I saw 2011. From APTs to hacking to malware and spam, there was a little something for everyone. I've now written a couple of these 'Top Ten' articles [7, 8], and while there is a lot of overlap, I'm always surprised by the new stories that appear and make the list.

Who knows what stories will make it onto my 2012 list.

REFERENCES

- [1] <http://www.vanityfair.com/culture/features/2011/09/operation-shady-rat-201109>.
- [2] <http://www.reuters.com/article/2011/05/17/us-apple-malware-idUSTRE74G60M20110517>.
- [3] <http://thenextweb.com/apple/2011/05/19/applecare-reps-told-not-to-help-users-remove-macdefender-malware/>.
- [4] <http://www.microsoft.com/SIR>.
- [5] <http://www.washington.edu/news/articles/new-study-quantifies-use-of-social-media-in-arab-spring>.
- [6] <http://www.fastcompany.com/1762972/facebook-vp-we-get-too-much-credit-for-the-arab-spring?partner=rss>.
- [7] <http://www.virusbtn.com/virusbulletin/archive/2010/01/vb201001-top-ten-stories>.
- [8] <http://www.virusbtn.com/virusbulletin/archive/2011/02/vb201102-top-ten>.

'Securing your Organization in the Age of Cybercrime'

A one-day seminar in association with the MCT Faculty of The Open University

- *Are your systems SECURE?*
- *Is your organization's data at RISK?*
- *Are your users your greatest THREAT?*
- *What's the real DANGER?*

Learn from top security experts about the latest threats, strategies and solutions for protecting your organization's data.

For more details:

www.virusbtn.com/seminar
or call 01235 555139



SEMINAR
19 April 2012
Milton Keynes, UK



The Open
University

SPOTLIGHT

CHALLENGES FOR THE LONDON ACTION PLAN

Wout de Natris

De Natris Consult, The Netherlands

In October 2011 the London Action Plan (LAP) held its annual workshop in Paris. Collaboration with the Messaging Anti-Abuse Working Group (MAAWG) meant that attendees were able to engage in more in-depth sessions with industry members and law enforcement representatives. However, with spam figures dropping while fraud and other forms of cybercrime continue to rise, the perceived significance of spam is in decline. LAP faces several challenges in 2012 that it must address in order to remain relevant. But before I present the challenges, an introduction is in order.

LONDON ACTION PLAN

The implementation of the 2002 EU ePrivacy and Electronic Communications directive¹, along with similar laws in other parts of the world, effectively dealt with the extreme nuisance of unsolicited electronic advertising, or spam. Anti-spam and malware enforcement agencies were established around the world and the need for cooperation became apparent. In 2004, the US Federal Trade Commission and the UK's Office of Fair Trading organized a workshop in London in which 27 organizations from around the world participated. They established an informal cooperation network: the London Action Plan. A mission statement was published: 'The purpose of this Action Plan is to promote international spam enforcement cooperation and address spam-related problems, such as online fraud and deception, phishing, and dissemination of viruses. The participants also open the Action Plan for participation by other interested government and public agencies, and by appropriate private sector representatives, as a way to expand the network of entities engaged in spam enforcement cooperation.' [1]

The Plan promoted cooperation and the sharing of data between different agencies, but it also promoted public-private cooperation at a time when it wasn't trending. Several early partners came from industry.

EARLY SUCCESSES

One of the group's early successes was information sharing. In the first set of cases involving cross-border

¹2002/58.

enforcement, New Zealand, Australian and US agencies² each took action against the prolific spammer Herbal King [2] and its mastermind, Lance Atkinson. Toni Demetriou, a senior investigator with the Anti-Spam Compliance Unit of New Zealand's Department of Internal Affairs, says: 'International cooperation was essential in getting a result in Operation Herbal King. The FTC was able to provide technical information, making it possible for us to identify the defendants and obtain evidence.' The various cases resulted in fines and strong injunctions.

Dollarrevenue [3] was another example of a LAP success. The case was brought by the Dutch OPTA³. By building its case based on data obtained from the FTC through the data-sharing provisions of the SAFE WEB law [4], OPTA was able to stop this source of malware, and levied a 600,000 euro fine.

The mere fact that there was a LAP membership list made contact much easier for enforcement officers. Other LAP initiatives also helped members achieve the shared goal of fighting spam. For example, LAP's data-sharing template helped standardize information requests and case referrals between agencies. Extensive training also led to the sharing of best practices and techniques for the participating agencies, e.g. on the lessons learned from cross-border cases or on potential cooperation with industry partners. LAP also promoted interaction with industry by co-organizing its annual workshop with the MAAWG meeting in 2007 and with Germany's annual eco anti-spam event – which included a *Microsoft* anti-fraud day – in Wiesbaden in 2008.

Hugh Stevenson, the FTC's Deputy Director for International Consumer Protection, sees a direct relationship between the LAP network and his agency's ability to prosecute spammers: 'Spam doesn't respect national borders, so law enforcers must find ways to work across them. LAP brings together the enforcers on the spam beat, as well as important private partners with a common interest in tackling the problem. Through training, information sharing, and ongoing contacts, we can all do far more together than we ever could on our own.'

However, the scene has changed over the past three years. The relationship between agencies has not intensified and several challenges for LAP have come to light.

CHALLENGES FOR 2012 AND BEYOND

With the rise of criminal activity on the Internet the focus has shifted away from spam, making spam enforcement

²The Department of Internal Affairs, Australian Communication and Media Authority, and the Federal Trade Commission.

³Onafhankelijke Post en Telecommunicatie Autoriteit (Independent Post and Telecommunications Authority).

a less essential topic and potentially leading to budget restraints as governments and agencies set different priorities.

Is this the correct way forward? To my mind it is not. LAP members can make a huge difference in fighting cybercrime, but they need to overcome several challenges. This can be done by capitalizing on what makes the LAP model of cooperation and knowledge and data sharing so unique.

Collecting high-quality data

Several spam and malware enforcement agencies have spam reporting centres. Inviting major ISPs and anti-virus companies to share their data with these centres leads to higher quality meta data. Evert Jan Hummelen, OPTA Deputy Head Consumers, Numbers and Chair's Office, who is responsible for the anti-spam and malware team, states: 'OPTA is constantly seeking information to improve its data position with respect to spam and malware. The first results from international cooperation and data sharing are now becoming visible.' By making the analysed data transparent, anonymity and hiding on the Internet becomes harder for spammers and attackers alike. For example, data on senders, infected computers, abused IP resources and hosting becomes available. By inviting selected industry partners and banks to share their data, and showing them the added value, more data will become available in 2012.

Cooperation with different enforcers and industry

As spam, fraud and malware have become virtually indistinguishable, different forms of enforcement have come into view. Toni Demetriou explains: 'Part of the challenge is realizing and understanding that each law enforcement agency works within a specific area. Police work within criminal law, and spam regulators/enforcers and consumer protection organizations work within civil or administrative law. Each has their own set of investigative tools and levels of proof that have to be provided to the legal system. Industry works with contracts and abuse clauses in those contracts. So the challenge is to overcome any legislative and jurisdictional barriers to legally and effectively share information and evidence in a timely and effective manner.' So who is best equipped to take on a specific case? All three entities have proven to be successful, for example, in taking down botnets. Coordination between them and the use of each one's unique powers will make a major difference where tackling cybercrime is concerned.

Coordination is not commonplace, so where do we start? My suggestion would be to look at sharing and analysing

data first. Then distribute the results, and from there work towards coordination. Also LAP could demonstrate the full potential of its members to other enforcement agencies through presentations at relevant events, e.g. at an eCrime meeting or at Europol and Interpol high-tech crime meetings.

The need for more countries to become actively involved

In order to be successful in fighting spam, fraud, malware and cybercrime, more countries need to become actively involved. In other words, more resources need to be put into enforcement agencies and the training of officers in this line of work. Within the EU this could be achieved by giving a form of coordinating power to ENISA, as OPTA suggested in 2009 [5], or by opening up the coordinative powers of the EU Cyber Crime Center (to be) to all agencies involved in enforcement on the Internet. On a worldwide scale this could be achieved through active involvement in the Council of Europe's Octopus programme and conference.

Whatever the challenge, it will be LAP's members that need to push for results at the aforementioned organizations. It will not be the other way around.

CONCLUSION

There are options available for LAP to prove its worth and make a difference, but it will take ambition, effort and resources. At the end of 2011 LAP faces a choice between obscurity and new successes. The comprehensiveness of the Plan puts LAP in a unique position to make a difference in the fight against spam, including all the harm that comes from the crime associated with it. The near future will show whether it is able to live up to this potential. If LAP is able to forge the necessary cooperation with old and new partners, I have no doubt that it will.

REFERENCES

- [1] <http://www.londonactionplan.org/>.
- [2] <http://www.spamhaus.org/news.lasso?article=649>.
- [3] <http://www.opta.nl/nl/actueel/alle-publicaties/publicatie/?id=2595> (in Dutch).
- [4] <http://www.ftc.gov/reports/ussafeweb/Summary%20of%20US%20SAFE%20WEB%20Act.pdf>.
- [5] http://ec.europa.eu/information_society/policy/nis/docs/pub_consult_nis_2009/public_bodies/OPTA.pdf.

END NOTES & NEWS

FloCon 2012 will be held 9–12 January 2012 in Austin, TX, USA. For more information see <http://www.flocon.org/>.

RSA Conference 2012 will be held 27 February to 2 March 2012 in San Francisco, CA, USA. Registration is now open. For full details see <http://www.rsaconference.com/events/2012/usa/index.htm>.

Black Hat Europe takes place 14–16 March 2012 in Amsterdam, The Netherlands. For details see <http://www.blackhat.com/>.

SOURCE Boston 2012 will be held 17–19 April 2012 in Boston, MA, USA. For further details see <http://www.sourceconference.com/boston/>.



The 3rd VB ‘Securing Your Organization in the Age of Cybercrime’ Seminar takes place 19 April 2012 in Milton Keynes, UK. Held in association with the MCT Faculty of The Open University, the seminar gives IT professionals an opportunity to learn from and interact with top security experts and take away invaluable advice and information on the latest threats, strategies and solutions for protecting their organizations. For details see <http://www.virusbtn.com/seminar/>.

Infosecurity Europe 2012 takes place 24–26 April 2012 in London, UK. See <http://www.infosec.co.uk/>.

The 21st EICAR Conference takes place 7–8 May 2012 in Lisbon, Portugal. The theme for this event will be ‘“Cyber attacks” – myths and reality in contemporary context’. For full details see <http://www.eicar.org/17-0-General-Info.html>.

The CARO 2012 Workshop will be held 14–15 May 2012 near Munich, Germany. The main theme of the conference will be ‘WWWTF – The Web: It’s broken, but can it be fixed?’. A call for papers has been issued, with a deadline for submissions of 15 January. For more information see <http://2012.caro.org/>.

NISC12 will be held 13–15 June 2012 in Cumbernauld, Scotland. The event will concentrate on ‘The Diminishing Network Perimeter’. For more information see <http://www.nisc.org.uk/>.

The 24th annual FIRST Conference takes place 17–22 June 2012 in Malta. The theme of this year’s event is ‘Security is not an island’. For details see <http://conference.first.org/>.

Black Hat USA will take place 21–26 July 2012 in Las Vegas, NV, USA. DEFCON 20 follows the Black Hat event, taking place 26–29 July, also in Las Vegas. For more information see <http://www.blackhat.com/> and <http://www.defcon.org/>.

The 21st USENIX Security Symposium will be held 8–10 August 2012 in Bellevue, WA, USA. For more information see <http://usenix.org/events/>.



VB2012 will take place 26–28 September 2012 in Dallas, TX, USA. VB is currently seeking submissions from those wishing to present at the conference. Full details of the call for papers are available at <http://www.virusbtn.com/conference/vb2012>. For details of sponsorship opportunities and any other queries please contact conference@virusbtn.com.



VB2013 will take place 2–4 October 2013 in Berlin, Germany. More details will be revealed in due course at <http://www.virusbtn.com/conference/vb2013/>. In the meantime, please address any queries to conference@virusbtn.com.

ADVISORY BOARD

Pavel Baudis, Alwil Software, Czech Republic
Dr Sarah Gordon, Independent research scientist, USA
Dr John Graham-Cumming, Causata, UK
Shimon Gruper, NovaSpark, Israel
Dmitry Gryaznov, McAfee, USA
Joe Hartmann, Microsoft, USA
Dr Jan Hruska, Sophos, UK
Jeannette Jarvis, McAfee, USA
Jakub Kaminski, Microsoft, Australia
Eugene Kaspersky, Kaspersky Lab, Russia
Jimmy Kuo, Microsoft, USA
Chris Lewis, Spamhaus Technology, Canada
Costin Raiu, Kaspersky Lab, Romania
Péter Ször, McAfee, USA
Roger Thompson, Independent researcher, USA
Joseph Wells, Independent research scientist, USA

SUBSCRIPTION RATES

Subscription price for Virus Bulletin magazine (including comparative reviews) for one year (12 issues):

- Single user: \$175
- Corporate (turnover < \$10 million): \$500
- Corporate (turnover < \$100 million): \$1,000
- Corporate (turnover > \$100 million): \$2,000
- *Bona fide* charities and educational institutions: \$175
- Public libraries and government organizations: \$500

Corporate rates include a licence for intranet publication.

Subscription price for Virus Bulletin comparative reviews only for one year (6 VBSpam and 6 VB100 reviews):

- Comparative subscription: \$100

See <http://www.virusbtn.com/virusbulletin/subscriptions/> for subscription terms and conditions.

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)1865 543153

Email: editorial@virusbtn.com Web: <http://www.virusbtn.com/>

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

This publication has been registered with the Copyright Clearance Centre Ltd. Consent is given for copying of articles for personal or internal use, or for personal use of specific clients. The consent is given on the condition that the copier pays through the Centre the per-copy fee stated below.

VIRUS BULLETIN © 2012 Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, England. Tel: +44 (0)1235 555139. /2012/\$0.00+2.50. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.