



virus

BULLETIN

Fighting malware and spam

CONTENTS

- 2 **COMMENT**
Living the meme
- 3 **NEWS**
New anti-phishing alliance formed
Spam affiliate programme to close
- 3 **VIRUS PREVALENCE TABLE**
- 4 **MALWARE ANALYSIS**
If Svar is the answer...
- 6 **TUTORIAL**
Static analysis of mobile malware
- 9 **FEATURE**
And the devil is six: the security
consequences of the switch to IPv6
- 13 **CONFERENCE REPORT**
Behind enemy lines: reporting from the CCC
28C3 Congress
- 16 **END NOTES & NEWS**

IN THIS ISSUE

SVAR SVAR AWAY...

It's not just graphic designers who can do interesting things with the Intel MMX instruction set. Virus writers are finding ways to (ab)use some of the instructions, too. Peter Ferrie has the details of W32/Svar.

page 4

STATICALLY MOBILE

Even in a mobile world, the principles of malware analysis remain the same. John Foremost takes us through the basic steps in the static analysis of mobile malware.

page 6

SIX SIX SIX

As the migration to IPv6 slowly begins to happen, Martijn Grooten takes a look at the potential security issues that could occur with the switch to IPv6, and encourages the security industry to ready itself for those challenges.

page 9

CCC 28C3

Morton Swimmer reports from Europe's premier hacker event.

page 13

‘Some security commentators suggest inventing answers to [security] questions rather than using real data.’ David Harley, ESET

LIVING THE MEME

One of my friends brought a pair of interesting *Facebook* memes¹ to my attention recently. They may not seem to have an immediate security connection, but I’ll come to that shortly.

Meme (1) involves the posting of a status update that reads something like ‘I’m going to live in Miami for 21 months’. Curiosity (or research, as it’s described in my job title) led me to discover that the meme relates to the poster’s birthday: 12 geographical locations represent each of the calendar months (e.g. Mexico = January, London = February, Miami = March), and the number of months for which the poster claims to be relocating represents the date of their birthday within that month. So in the example above, the poster’s birthday is 21st March. In another variant, the post reads ‘I’m [n] weeks in and craving [some kind of candy]’ where [n] represents the day and there is another list on which different types of candy represent different months of the year.

I gather that these games are played to raise awareness of breast cancer, though I don’t see how and if this kind of post fits usefully with other gender-oriented fund- and awareness-raising events such as Race For Life².

Meme (2) suggests that putting the last three digits of your cell phone number into a string like @[123:0] and adding it to a *Facebook* comment will return the name of the cell phone. In fact it has nothing to do with your cell phone, unless every device with a number ending in 123 (for example) is called Morgan Grice. The string format

¹ Meme: An idea, behaviour, style, or usage that spreads from person to person within a culture. <http://www.merriam-webster.com/dictionary/meme>.

² <http://raceforlife.cancerresearchuk.org/>.

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*

(sometimes) represents a numeric array associated with a *Facebook* account. It doesn’t even have to be a three-digit number: for example, @[4:0] returns ‘Mark Zuckerberg’ and @[21222:0] returns ‘DJ Vas Deferens’ (a shock jock, perhaps).

This is all very amusing, but I promised you some security content. Meme (1) is a pretty good way of letting those who are ‘in on the secret’ know when your birthday is – though your date of birth is likely to be of more use to an attacker when trying to access sensitive data via ‘secret questions’.

Meme (2) is less of an issue: only the most painstaking data aggregation attack will attempt to harvest cell phone numbers one triplet at a time. I’d be more concerned if the suggestion was to use a credit code or iGadget PIN. But nobody would fall for that, would they?

Well, the following is a meme flagged by Graham Cluley³ around the time of the royal wedding in the UK in 2011, highlighting a security issue with posting details like this:

What’s your royal wedding guest name? Start with Lord or Lady. Your first name is one of your grandparent’s names. Your surname is the name of your first pet double-barrelled with the name of the street you grew up on.

Secret answers to security questions posed by banking sites and the like as a supplement to passwords, or for people who *forget* their passwords, are pretty stereotyped. Names of relatives, names of pets, first school, childhood address and so on are highly characteristic, so some security commentators suggest inventing answers to such questions rather than using real data. That’s a logical alternative to inventing your own challenge/response – which is rarely an option – and I’m all in favour of it, as long as it doesn’t contravene some legal or quasi-legal restriction.

Do people lie in their social networking profiles, or when offered a candy bar in exchange for a password? I’m not in favour of dishonesty in general, but if this were general practice, it would suggest a healthily cynical attitude towards organizations who regard us not as customers, but as sources of commoditized data. However, experience with hoaxes shows that when ‘good causes’ like cancer awareness or missing children are involved, scepticism dissolves. I don’t know if *any* of these memes originate in an attempt at data harvesting, but such attacks would dovetail all too comfortably with the social network’s vested interest in data sharing, and work to an imaginative attacker’s advantage.

³ <http://nakedsecurity.sophos.com/2011/04/28/why-you-shouldnt-reveal-your-royal-wedding-guest-name/>.

NEWS

NEW ANTI-PHISHING ALLIANCE FORMED

A new alliance has been formed by the biggest players in the email service/technology industry to help combat spam and phishing. A total of 15 companies, including *Microsoft, PayPal, Google, Facebook, LinkedIn, AOL* and *Yahoo!*, have come together to develop new standards to help curb the problem of fraudulent mail. The alliance has been named 'Domain-based Message Authentication, Reporting and Conformance', or DMARC, and has already produced a draft specification that helps to formalize and automate message authentication processes.

Although a number of forms of email authentication exist – such as SPF, DKIM and SenderID, there is currently no common standard. The DMARC framework aims to provide a more comprehensive and integrated way for email senders to introduce email authentication technologies into their infrastructure. Once it has gathered some data from usage of the technology in the field, DMARC intends to submit the specification to the IETF for standardization.

SPAM AFFILIATE PROGRAMME TO CLOSE

By the time this issue of *Virus Bulletin* is published, there will be one less affiliate programme generating spam to clog up our inboxes.

GlavTorg.com was run by the people behind the prolific Glavmed/SpamIt operations which pushed out massive volumes of Canadian pharmacy spam until closing their doors in October 2010, claiming that increased attention on the business had made it impossible to continue. GlavTorg marketed sites selling cheap imitations of designer goods.

At the end of December 2011, GlavTorg affiliates were notified that the network was to be shut down and that they would not receive payment after 31 January. A message read:

'Dear partners, We would like to inform you that we have decided to close the trade direction replica handbags and clothing. The reasons for this decision and are associated with economic deterioration in the quality of products provided by our suppliers. We believe that any business should be to balance the interests of buyers and sellers, which has recently become disturbed.'

Researcher Brian Krebs suggests that the downfall of GlavTorg may partly have been due to brand owners taking action against those selling knock-offs of their products – in September, *Chanel* took legal action against several entities including one of GlavTorg's primary merchandising sites.

Cisco and several other sources reported a decrease in global spam volumes immediately following SpamIt's closure in October 2010. It remains to be seen whether the closure of GlavTorg will have a similar effect.

Prevalence Table – December 2011 ^[1]

Malware	Type	%
Autorun	Worm	7.26%
Encrypted/Obfuscated	Misc	6.09%
Iframe-Exploit	Exploit	5.65%
LNK-Exploit	Exploit	5.30%
Heuristic/generic	Virus/worm	5.18%
Sality	Virus	5.01%
Zbot	Trojan	3.69%
Adware-misc	Adware	3.37%
Conficker/Downadup	Worm	2.89%
Crack/Keygen	PU	2.74%
BHO/Toolbar-misc	Adware	2.68%
Cycbot	Trojan	2.65%
Heuristic/generic	Trojan	2.59%
Autolt	Trojan	2.52%
Freeware-downloader	PU	2.37%
Agent	Trojan	2.35%
Slugin	Virus	2.26%
VB	Worm	1.89%
Sirefef	Trojan	1.87%
Virut	Virus	1.86%
Pameseg	Trojan	1.70%
Downloader-misc	Trojan	1.60%
PDF-Exploit	Exploit	1.51%
FakeAV-Misc	Rogue	1.40%
Crypt	Trojan	1.39%
Delf	Trojan	1.26%
FakeAlert/Renos	Rogue	1.25%
Virtumonde/Vundo	Trojan	1.23%
Exploit-misc	Exploit	1.18%
WinWebSec	Rogue	1.06%
Kryptik	Trojan	1.04%
OnlineGames	Trojan	1.03%
Others ^[2]		14.15%
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

IF SVAR IS THE ANSWER...

Peter Ferrie

Microsoft, USA

The *Intel* MMX instruction set is full of instructions whose usefulness might not be immediately clear to anyone who does not work with graphics. However, it's not just the graphic designers who can do interesting things with them. Virus writers are finding ways to (ab)use some of the instructions, too. This time, we have W32/Svar, and another way to encode.

IN THE BEGINNING

The first generation of the virus begins by saving the relative address of the original entrypoint on the stack. Unlike in W64/Svafa¹, this value is always correct. The virus applies the current imagebase value from the ImageBaseAddress field in the Process Environment Block, which would normally be required to account for Address Space Layout Randomization (ASLR). However, the virus disables ASLR for the file during infection, so the technique is not required.

The virus continues by setting up a Structured Exception Handler in order to intercept any errors that occur during infection. The virus retrieves the base address of kernel32.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 address of kernel32.dll is always the second entry in the list. The virus assumes that the entry is valid and that a PE header is present there. This is fine, though, because of the Structured Exception Handler that the virus has registered.

The virus resolves the addresses of the API functions that it requires, which is the bare minimum set of APIs that it needs for replication – find first/next, open, map, unmap, close. 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 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', however it does prevent the use of any API whose hash ends with that value.

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'. The search 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 an enlarged view of the contents. There is no attempt to remove the read-only attribute, so files that have the read-only attribute set cannot be infected. In the case of a directory, the open will fail, and the map will be empty. The map size is equal to the file size plus 8KB, to allow the file to be infected immediately if it is acceptable. This 8KB value is hard-coded in the virus, which could interfere with variants being made based on it, and which could lead to a crash during decryption. Using the post-infection size during the validation stage allows the virus to avoid having to close the file and re-open it with a larger map later. The virus assumes that the handle can be used, and then checks whether the file can be infected.

BITS AND PIECES

The virus is interested in Portable Executable files for the *Intel x86* platform that have no appended data. Renamed DLL files are not excluded, nor are files that are digitally signed (at least, not explicitly – most of them will be filtered implicitly, because it is common for the signature to be placed after the end of the last section, but this is not a requirement). The subsystem value is checked, but this is done 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 section attributes are marked as executable and writable. The virus encodes its body using a bit-mask technique. There is only one table involved this time, which is eight times the size of the virus code. The table contains the bit-mask. Each byte of the host is split into eight bits, and each bit is stored individually in the table after combining it with seven bits that are set to a random value. This process is repeated over the entire host body. Interestingly, most of the code is optimized for small size, but the encoding routine is not optimized at all. Instead of simply rotating the bit into the top of the random value in order to combine it, the virus performs the equivalent of an 'if...then...else' for each bit in the code, and ORs or ANDs the value as appropriate. Once the encoding is complete, the virus stores four bytes of zero, which are intended to mark the end of the virus body, but there are two problems with this. The first problem is that there is a small, but real chance that if the top four bits were zero in any byte in the virus code, and if the random number generator happened to return a zero in the low byte four times in a row, then the output would match exactly what the

¹ <http://www.virusbtn.com/pdf/magazine/2012/201201.pdf>.

virus uses to mark the end of the virus body. In that case, the decoder would stop too soon, and the virus would crash.

The virus zeroes the RVA of the Load Configuration Table in the data directory. This has the effect of disabling SafeSEH, but it affects the per-process GlobalFlags settings, among other things. The virus also zeroes the DLLCharacteristics field. This has the effect of disabling ASLR and 'No eXecute' for the process (allowing the virus to run in a section that does not have the executable flag set, but the virus sets it explicitly anyway, as noted above), and enabling Structured Exception Handling. The virus saves the original entrypoint in the virus body, and then sets the host entrypoint 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.

When an infected file is executed, the virus decodes itself. The decoding is done using two MMX instructions, one of which might be considered to be a bit obscure: PMOVMASKB. The PMOVMASKB instruction accepts two parameters which correspond to the table that the virus constructed, and the register to receive the result. The instruction works with eight bytes at a time, and combines a single bit from each byte into a single byte which the virus stores. The result is the decoded host byte. The decoder does not use a register to hold the number of bytes to decode. Instead, it checks if four bytes of zero were read at a particular alignment. However, there is a bug in this check, and this is the second problem: the alignment is incorrect for the purpose. As a result, the decoder interprets its own code as though it were also encoded, and 'decodes' this, too. Fortunately for the virus, this action is harmless because the table is so large that the decoder cannot be overwritten. However, there is a small potential problem which stems from the hard-coded 8KB value noted above: if the table and the decoder happened to end at exactly a multiple of 8KB, then the decoder bug would cause the decoder to access memory beyond the end of the image and crash.

CONCLUSION

The PMOVMASKB technique is yet another surprise from the MMX instruction set, but the entire body is encoded so it does not look like corrupted code. However, anti-malware emulators will have no problem emulating through the code and won't appreciate the technique.

'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



TUTORIAL

STATIC ANALYSIS OF MOBILE MALWARE

John Foremost

Independent researcher, USA

Even in a mobile world, the principles of malware analysis remain the same. Files can be captured through many different mediums, such as downloads from an application market or website, through a mobile device, through emulated lab environments, downloads from mobile malware repositories and more. Once captured, the study of the file begins with the age-old static analysis, with tools and tactics customized for mobile malware. The examples provided in this article are focused on *Android* threats, but the principles apply to all mobile malware analysis.

STEP 1 – STATIC FILE METADATA

All files, malicious or not, have basic metadata details that are pertinent to an investigation. The basics that should be collected include (but are not limited to) hashes like MD5 and SHA1, file size, and other properties that may exist for the file. These may include: filename extension, header, file type (*Linux* command), packer details (scanning and

manual inspection methods), etc. Once collected, all file information needs to be organized into a research archive, as other similar samples or details may be discovered through the analysis process.

Internet searches should then be performed against all the data collected to look for related reports, abuse, or other information of relevance to the investigation. This may result in the discovery of anti-virus reports, related samples, dates and times of incidents and other data of interest. If the researcher is just trying to find out basic information – such as attempting to confirm the maliciousness of a file, a simple MD5 query can quickly provide the answer.

For example, e7584031896cb9485d487c355ba5e545 is the MD5 hash value of a known malicious mobile malware sample. A *Google* search on this value brings up three links which both name the sample and help to identify some functionality.

STEP 2 – ANTI-VIRUS SCANNING

Several web-based freeware scanners exist for processing mobile malware:

- *Avast*: <http://onlinescan.avast.com/>
- *Jotti's Scanner*: <http://virusscan.jotti.org/en>

```

com.droiddream.bowlingtime.apk->classes.dex (D4FA864EEDCF47FB7119E6B5317A4AC8->ADD472D8D4A39C602AD31E23ACE4F3BE)
Header:
  Magic:          "dex"
  Version: 035
  Checksum:      8F24DD46
  SHA-1: 00BC064674921016F23FCC0C92FAE51D8216C9A5
  FileSize:      303300
  HeaderSize:    00000070
  Endianness:    12345678
  LinkSize:      0
  LinkOffset:    00000000
  MapOffset:     00049FF4

[snip...]
5B 20 038B | iput-object v0, v2, field@038B; com.phonegap.AccelListener com.phonegap.DroidGap.accel
22 00 016C | new-instance v0, type@016C ; com.phonegap.CameraLauncher
70 30 06B1 0230 | invoke-direct {v0, v3, v2}, method@06B1 ; void com.phonegap.CameraLauncher.<init> (android.web-
kit.WebView, com.phonegap.DroidGap)
5B 20 0394 | iput-object v0, v2, field@0394; com.phonegap.CameraLauncher com.phonegap.DroidGap.launcher
22 00 016F | new-instance v0, type@016F ; com.phonegap.ContactManager
70 30 06BB 0320 | invoke-direct {v0, v2, v3}, method@06BB ; void com.phonegap.ContactManager.<init> (android.app.
Activity, android.webkit.WebView)
5B 20 0396 | iput-object v0, v2, field@0396; com.phonegap.ContactManager com.phonegap.DroidGap.mContacts
22 00 017A | new-instance v0, type@017A ; com.phonegap.FileUtils
70 20 0702 0030 | invoke-direct {v0, v3}, method@0702 ; void com.phonegap.FileUtils.<init> (android.webkit.
WebView)
[snip...]

```

Figure 1: Example of part of the output for a file analysed by DexID.

- *Metascan*: <http://www.metascan-online.com/>
- *NetQin*: <http://scan.netqin.com/en/>
- *VirusTotal*: <http://www.virustotal.com/>

The results returned by such scanners are not conclusive, but they do often help identify family and/or possible functionality. Also, metadata may exist on some sites such as *VirusTotal*, where users supply links, comments, or related data when uploading or analysing a sample of interest.

Application-based scanners may also be used to scan mobile malware. For example, a wealth of anti-virus applications exist for *Android*, ranging from *Zoner AntiVirus Free* to *AVG Mobilation Free* and *Kaspersky Mobile Security*. The downside of using such solutions for analysis is that the applications must be installed, configured and maintained on a lab device or in an emulated environment – a notable task that may be beyond the scope of the average department attempting to triage new samples.

DexID is a great freeware tool for identifying known *Android* malware. It can be obtained via [hxxp://dl.dropbox.com/u/34034939/dexid.zip](http://dl.dropbox.com/u/34034939/dexid.zip). (*Dexid.dat* is also required to obtain updated signature data associated with the tool.) The tool installs easily on a *Linux* system, requiring several Perl modules in order to run the tool as configured. Figure 1 shows an example of part of the verbose output for a file analysed by *DexID*.

STEP 3 – COMPARATIVE ANALYSIS

Fuzzy analysis using *ssdeep* can help to identify samples that may be similar to one another. This can be very useful

when trying to determine whether samples are closely related. For example, one variant may be discovered and another file may be suspected on the same network – perhaps a private update made to a device following infection. A fuzzy analysis helps to identify and/or locate related samples. Simply run a command such as:

```
ssdeep -rd DIRECTORY > results.txt
```

This command searches recursively through the specified directory to compare samples, writing the results into *results.txt*. A '-x' option can also be used to compare hashes in two or more files. The output is similar to the example shown below, revealing the degree of correlation as a percentage:

```
Computer1.data.txt:C:\tank.apk matches
Computer2.data.txt:C:\guns.apk (68)
```

STEP 4 – UNPACKING AND CONVERTING

Programs such as *7Z* or *WinZip* can be used to extract files including *Android* APK files. Extracted APK files may contain DEX script, XML and ARSC. Analysis begins with the manifest file, such as *AndroidManifest.xml*. This file contains a long list of strings that may reveal potential functionality for the code, such as SMS messaging, networking, phone interactions and more. A good place to look up the functionality of *Android*-based strings is <http://developer.android.com/reference/android/Manifest.permission.html>.

The example shown in Figure 2 contains extracts from the *Alsalah* malware's *AndroidManifest.xml* file. To avoid null

```
versionCode
versionName
label
icon
configChanges
theme
android*http://schemas.android.com/apk/res/android
package
manifestVersion
android.permission.INTERNET
android.permission.ACCESS_FINE_LOCATION
android.permission.ACCESS_NETWORK_STATE
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.READ_CONTACTS
android.permission.CHANGE_WIFI_MULTICAST_STATE
android.permission.CLEAR_APP_USER_DATA
android.permission.BIND_INPUT_METHOD
android.permission.WRITE_CONTACTS
android.permission.CLEAR_APP_CACHE
android.permission.AUTHENTICATE_ACCOUNTS
#android.permission.

(snip)

Alsalah.activity
Alsalah.intent-filter
Alsalah.intent.action.MAIN
Alsalah.intent.category.LAUNCHER
Alsalah.receiver
Alsalah.receiver.action
Alsalah.places
Alsalah.gps
Alsalah.help
Alsalah.about
Alsalah.settings
Alsalah.import
Alsalah.share
Alsalah.export
Alsalah.history
```

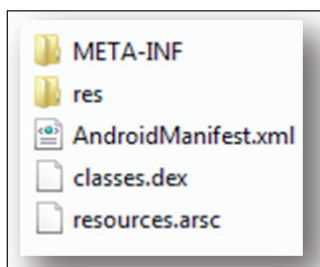
Figure 2: Extracts from the *Alsalah* malware's *AndroidManifest.xml* file.

character interpretations the file was opened in *Notepad* and then copied for primary ASCII strings of interest to review possible functionality leads. A few immediate leads are highlighted in bold text, and many more can be found by scanning through the output.

DEX files (<http://code.google.com/p/dex2jar/>) can be converted to JAR files in order to view them using programs like *JD-GUI* (<http://java.decompiler.free.fr/?q=jdgui>), *Djdec39*, *Cavajdemo* or others. To convert files use the following options for *Windows* and *Linux*:

- *Windows*: dex2jar.bat file.apk
- *Linux*: sh dex2jar.sh file.apk

For example, *Alsalah.apk* unpacks to the following:



STEP 5 – ANALYSING EXTRACTED DATA

Once a DEX file has been converted into a JAR file, analysis can begin along the lines of a normal Java analysis, using the common aforementioned tools. A simple review of scripts often reveals functionality, URLs, or similar data of interest. The data can then be fed back into this process recursively so that all static data can be researched and analysed accordingly, until all avenues of static analysis have been exhausted.

Figure 3 shows an example of a DEX file converted to JAR and then viewed within *JD-GUI* to identify URLs associated with the mobile malware:

STEP 6 – FREE SANDBOX ANALYSIS

While not ‘static’, sandbox options often follow static analysis and do not require any specialized lab set-up to triage mobile malware. Sandbox analysis for mobile malware is still emergent and may not be as timely as desired, but it is available free of charge (but note that the following site is sometimes down for maintenance): <http://www.mobile-sandbox.com/upload.php>.

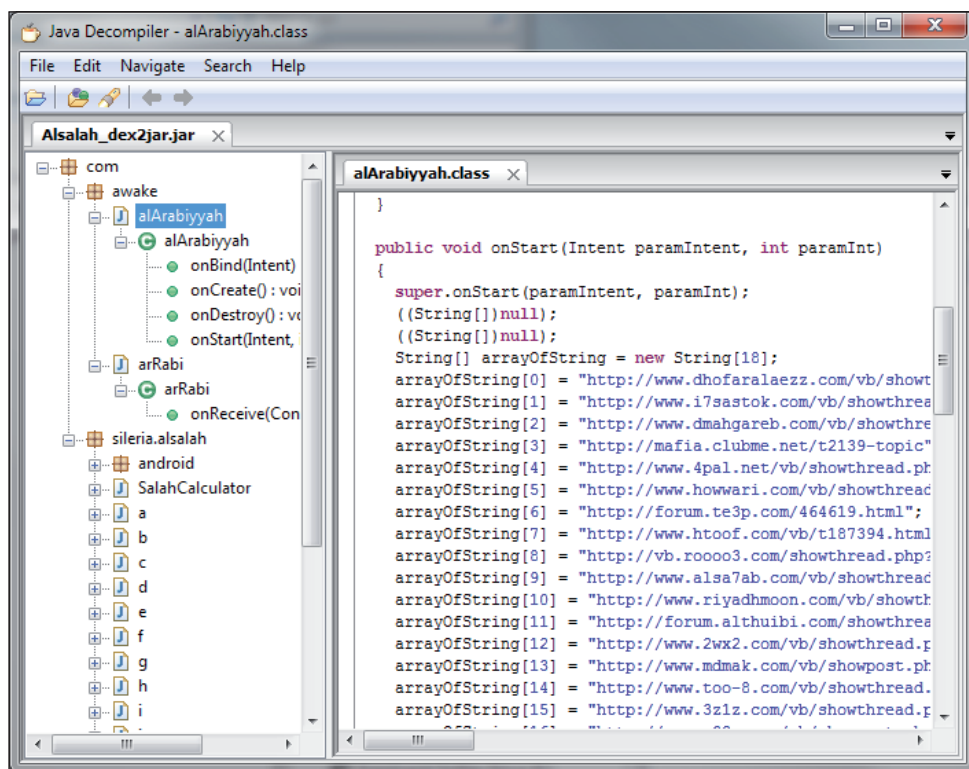


Figure 3: *JD-GUI* decompiles a JAR file to reveal URLs in the mobile malware.

SUMMARY

As illustrated in this article, static analysis alone can provide an excellent view into related abuse, related malware samples and code functionality. Many individuals globally are beginning to develop new skills for mobile malware analysis. Our community needs to further develop and automate static solutions so that static analysis becomes a well understood and standard component in the analysis of mobile malware. Over time, dynamic solutions will become more prevalent and more robust, adding further support to the quest to understand malicious mobile applications and artefacts.

FEATURE

AND THE DEVIL IS SIX: THE SECURITY CONSEQUENCES OF THE SWITCH TO IPv6

Martijn Grooten
Virus Bulletin, UK

Early in 2011, ICANN handed out its last available /8s (IPv4-speak for ‘blocks of 16,777,216 IP addresses’) to the five regional Internet registries (RIRs). One can picture ICANN as the IP address factory and the RIRs as the warehouses distributing them – with the important exception that ICANN can’t simply produce more addresses when it needs to: it is limited by their 32-bit length, which means that, in theory, just over four billion of them are available; in practice there are even fewer than that.

This has not come as a surprise to anyone. Ever since the Internet outgrew its original purpose of a network for computer researchers and defence experts, we have known that the total number of possible IP addresses is significantly smaller than the world’s population – and, in fact, there is a need for many more than one IP address per person.

Even though RIRs and ISPs still had enough IP addresses available for the near future, the message last year was clear: we were running out of IP addresses and we really had to start moving towards a new protocol that allowed for a much larger address space.

Thankfully, since the late 1990s such a new protocol has been available: IPv6 (the ‘standard’ IP protocol, version 4¹, is commonly and henceforth referred to as IPv4). Unfortunately, for a number of reasons migration to IPv6 has been rather slow, and it will probably continue to be slow for some time to come.

No matter how slowly, though, the migration to IPv6 *is* happening. And, with new protocols come new security issues and challenges. This article looks at some of these issues and hopes to encourage the security industry to ready itself for IPv6 – and all of the nasty side effects that may come with it.

This is not a warning against migration to IPv6. I believe this migration is a necessary step in order to keep the Internet usable in the future – and one that should probably have been taken some time ago.

This is also not an article that claims to give a complete overview of all the bad things that can happen or that are

¹ There have never really been IP versions other than 4 and 6 that have been used in practice; 4 and 6 are, however, not just version names: these numbers are used in the first four bits of every IP packet.

happening with IPv6. Rather, it points out some of the security issues that may occur with the switch to IPv6. Its purpose is to raise awareness of these issues as well as the many that aren’t covered.

LAYER UPON LAYER UPON LAYER

Communication on the Internet uses several hierarchical abstraction layers². At the very bottom is the *physical layer* (which may in fact be wireless), which is used to transport two different kinds of ‘things’ – commonly referred to as 0s and 1s – between two devices. On top of that is the *link layer*, which puts these 0s and 1s together in ‘frames’ to allow for a meaningful data exchange.

The *internet layer* comes next – this is where IPv4 and IPv6 come into play. This layer is used to send packets from one device to another over the Internet, despite there not being a direct connection between the two devices. For this reason, all devices are given an address; it is these addresses that, in the case of IPv4, we’re running out of.

On top of that, the *transport layer* is used to provide end-to-end communication between applications, with the most commonly used transport layer protocols being TCP and UDP. The former maintains a connection between the two devices and makes sure information spread over multiple packets can be assembled correctly; the latter is used to send small packets where speed is more important than guaranteed delivery and no connection state is maintained. Ports are part of the transport layer.

Finally, on top of the transport layer, there is the *application layer*, used by actual applications such as HTTP, SMTP, FTP (all of which use TCP) and DNS (which almost always uses UDP). These protocols describe the communication rules for particular applications that use the Internet.

A change in one layer does not usually affect the others: to use IPv6, one can use the same cables that are used for an IPv4 connection. In fact, IPv4 and IPv6 generally make use of the same infrastructure and as long as routers and computers are IPv6-enabled³, IPv4 and IPv6 can be used together on the same networks. IPv6 doesn’t require changes to upper-layer protocols either: there is no HTTP-for-IPv6 or DNS-for-IPv6 – the protocols themselves are IP-agnostic.

However, in practice it is hard to think of the upper-layer protocols without seeing the IP address as part of them.

² Different authors use different names for the various layers; the bottom two layers are commonly seen as a single layer. This section gives a brief introduction to the subject of IPv6; anyone wanting to know more should consult one of the numerous books on the subject – or the many well-written *Wikipedia* pages.

³ Many existing routers still don’t support IPv6, but all commonly used operating systems do.

While in principle, a web server will be able to serve IPv6 requests if the machine is able to make IPv6 connections, in practice without special arrangements for IPv6, at the very least the log files will start to look odd and it is possible that more serious problems will arise⁴. For mail servers (that use SMTP) the problem is much worse, as IP-based (spam)filtering is heavily embedded in most of them.

This is also something to keep in mind when designing a security application that sniffs transport or application layer traffic: such applications are likely to work for IPv6 traffic right away. However, if the application is not aware of IPv6, it is less likely to derive anything meaningful from the sniffing.

IPv6 differs in a number of ways from IPv4. Apart from the four-bit version number (which is 0110 for IPv6), there is more flexibility for using ‘extension headers’, while there is no checksum any more: it is believed that the link layer currently provides sufficient error detection. The most noticeable change, however, is the 128-bit length for IP addresses, compared to 32-bit for IPv4.

Because of the 128 bits, there are 2^{128} possible IPv6 addresses – about 340 undecillion, and 2^{96} times as many as there are IPv4 addresses. For comparison, the amount of spam sent every day is in the order of 2^{36} . For all practical purposes, the number of IPv6 addresses equals infinity.

While IPv4 addresses are commonly written in dot-decimal notation (e.g. 212.58.244.69), the much longer IPv6 addresses are written in hexadecimal notation, with four groups of eight digits, separated by colons – for instance, 2620:0000:1cfe:face:b00c:0000:0000:0003. To simplify notation, leading zeroes can be omitted in each group and two colons can replace one or more consecutive groups of zeros⁵, so that the aforementioned address becomes 2620:0:1cfe:face:b00c::3 – indeed this is the IPv6 address used by *Facebook*.

The x/n notation commonly used for IPv4 blocks is also used in IPv6 to denote the block of IP addresses that are equal to x in all but the first n bits. For instance, the block 2000::/3 consists of those addresses for which the first three bits are 001. In fact, this is the block that is currently allocated for the use of publicly routable addresses. Therefore, in practice one only has to take into account 2^{125} IPv6 addresses – although this is nothing but a smaller version of infinity.

FOUR AND SIX. AND FOUR-AND-SIX.

In an ideal world, all ISPs, software vendors and network experts would spend the next few months making sure

⁴That is not to say that most commonly used web servers aren’t IPv6-ready. *Apache*, for instance, has been ready since the release of its 2.0 version almost a decade ago.

⁵The double colon can occur only once in an IPv6 address.

we were all IPv6-ready, and by the end of the year IPv4 would be added to the list that already includes gopher and ARPANET – useful once, but no longer needed. Unfortunately, the Internet is not an ideal world.

So, while it is important for organizations to become IPv6-connected, for the foreseeable future it will continue to be much more important to remain IPv4-connected.

An organization that wants to increase its online presence should therefore ensure it stays IPv4-connected despite the possible lack of availability of IPv4 addresses. There are a number of ways in which an organization can do that and, from a security point of view, they can make the picture slightly more complicated.

The first is to use NAT (Network Address Translation) on a larger scale, commonly known as ‘carrier-grade NAT’ or CGN. NAT allows for multiple devices to be connected to the Internet using a single public IP address: using port-mapping, a router at the gateway makes sure that IPv4 packets received from the Internet are sent to the correct device. NAT is commonly used in households and small offices. In principle, it can be used for larger areas too: for instance, an ISP with a limited number of IPv4 addresses can put groups of customers on a NAT.

There are a number of drawbacks to being on a NAT, the lack of end-to-end connectivity probably being the most important, but for day-to-day Internet usage it generally suffices. However, from the outside world, it is usually not possible to discern from which particular device traffic from the NAT’s IP address was sent. This has important security implications.

Knowledge that a certain IP address has been used in malicious activity – commonly because it is part of a botnet – is useful in the fight against malware. Certain activity, such as sending email⁶, can be denied to the IP address until it has been proven to be clean. When the IP address is, in fact, the gateway to a wider area NAT, this means that innocent users will be blocked, despite being unable to influence the activity for which the block is in place.

Similarly, for law enforcement purposes it can be very useful to know which IP address has participated in a certain activity. If the IP address is shared by a large number of users, it will provide little information without further details from the ISP. The additional information that can be provided by the ISP will be dependent on the quality of the ISP’s log files – and keeping logs of ‘good quality’ may see the ISP run into storage issues and may also conflict with privacy laws.

⁶While sending email is the most obvious example, it may not be the best one: many ISPs, regardless of whether they use CGN, disallow the sending of direct-to-MX email; in many cases port 25 has proactively been closed. A better example is for access to a popular online game or website to be denied as a result of abusive activity from the IP address.

Other than using NAT, a company that is in need of IPv4 addresses may also find them on the ‘second-hand market’. Large chunks of IPv4 addresses (including /8s) were assigned in the 1980s and early 1990s to organizations that were large at the time. Many of these addresses have never been used and have now been given back – or, indeed, are being sold on IPv4 marketplaces.

There is nothing inherently insecure about this practice, yet it is something we ought not to ignore. For instance, a lot of applications depend on determining the geographical location of a certain IP address. As a security measure, geolocation-based restrictions are easily evaded, but they are still in place and it is good to be aware of the fact that, with chunks of IP addresses being sold, *geoIP* databases are likely to become outdated. Denying an Internet user access to a certain application because their IP’s location doesn’t match their physical one is thus likely to result in many false positives.

As IPv4 blocks are being sold, the global routing table is becoming bigger too. It is not unimaginable that this will lead to routing issues, which could be abused by those with malicious intentions. In one reported case, a block of IP addresses was effectively ‘stolen’ [1]. Again, it is not unimaginable that this will happen more often in the future.

Thankfully, not everyone will cling to IPv4 for as long as possible. But switching to IPv6 might not be as easy as it sounds: doing so depends on both router(s) and provider to be IPv6-ready, and many are not. Thankfully, there are a number of ways to use IPv6 over an IPv4 connection.

In the 6to4 transition mechanism, IPv6 packets are encapsulated inside IPv4 packets⁷, allowing IPv6 packets to travel over an IPv4-only connection. The Teredo transition technology (and its *Linux* equivalent Miredo) works by encapsulating IPv6 packets inside IPv4-UDP packets: it can even be used by devices on a local network behind a NAT.

Both 6to4 and Teredo (as well as 4in6, which allows IPv6-only devices to send and receive IPv4 traffic) are useful protocols and there is nothing inherently wrong with them (certainly not from a security point of view). However, developers of network security applications ought to be aware of their existence and consider them as possibilities when sniffing network traffic. They also mean that using IPv6 is a more easily available option for botnet authors than it may at first seem. (There are many other ways to use IPv6 on an IPv4 network; I have singled these two out

⁷The *protocol number* inside the IPv4 header – normally used to define the transport layer protocol used (e.g. 6 for TCP, 17 for UDP) – is set to 41, denoting that the body of the IPv4 packet contains an IPv6 packet.

because they are probably the easiest to set up, which makes them more attractive for malware authors.)

BIGGER AND BETTER: IPv6 AND SPAM

It is hard to think of current spam filters without thinking of IP-based blacklists, whitelists and reputation systems. IPv4 is, in many ways, ideal for spam filters: a binary list containing a 0 or 1 for each possible IPv4 address is 0.5GB in size and fits on a small USB stick. The number of legitimate mail servers is relatively small and the vast majority of IP addresses should never send direct-to-MX emails⁸ (or have a history of sending spam), making it very useful to keep a list of the legitimate senders (an IP whitelist) or, more commonly, those that send spam (an IP blacklist).

Don’t even consider trying to do the same for the 2^{125} publicly routable IPv6 addresses. There will never be enough storage space for these, and there are enough addresses for every single piece of spam to be sent from a different IPv6 address.

The idea of IPv6 address assignment is for end-users and small offices to be assigned at least a /64 block of IPv6 addresses, so you could base a blacklist on the first 64 bits. Now, if indeed the Internet did behave in this ideal way (world peace is far more likely), you would ‘only’ have to consider 2^{61} ($=2^{125}/2^{64}$) possible IPv6 blocks. This number is still far too large to run a blacklist.

One solution to this problem would be to start by managing a whitelist of IPv6 addresses belonging to outbound mail servers: legitimate mailers, but possibly also spammers. Any email sent from addresses that are not on this whitelist is blocked anyway, and within the whitelist, spammers may be blacklisted or more advanced scanning techniques may be applied. This is the principle behind *ipv6whitelist.eu* [2], for instance.

Another possibility is to filter based on domains, rather than on IP address; DKIM, which cryptographically links a domain name to an email message, is ideal for this. As its many advocates will happily point out, DKIM is not only IP-agnostic, but it has a number of advantages over IP-based filtering that make it useful for IPv4 already.

It is, for instance, possible to have multiple DKIM signatures attached to a message (if it is sent by company A on behalf of company B). By using subdomains, DKIM also allows organizations to make a distinction between different kinds of emails they send. And organizations are less likely to change domains than they are to change IP addresses.

⁸Rather, they should connect to their ISP’s mail server using an authenticated SMTP connection.

DKIM thus has a lot of potential for IPv4 already; if it is more widely deployed by senders and filters alike, the switch to IPv6 will be a lot more seamless.

However, all of this may not be needed in the foreseeable future. As noted before, the number of mail servers is small – significantly smaller than the number of IPv4 addresses – and for the foreseeable future, despite exhausting the supply of IPv4 addresses, organizations will be able to find one or two IPv4 addresses for their mail servers. It may well be that email is the last part of the Internet to switch to IPv6, and this switch may not happen until the middle of the century or even later.

Indeed, while a number of mail servers and spam filter vendors have proudly announced that they are IPv6-ready, the amount of spam sent over IPv6 is extremely small. Those spam messages that have been sent have, without exception, been sent over IPv6 because this happened to be the default connection between the sender (usually a node on a botnet) and the recipient. There are no known examples of spam sent over IPv6 where the spammer deliberately used that connection.

Finally, it is good to look at one other way in which spammers can make use of IPv6: by including IP-based URLs. Such a URL is written as `http://[2a00:1450:400c:c01::6a]/` – the square brackets are to distinguish the colons from the ones used to denote the port number – and a number of email programs turn such URLs into clickable links. They may not be recognized as such by spam filters.

IPv6 AND MALWARE

As already mentioned, spammers have barely jumped onto the IPv6 bandwagon and the same can be said for malware authors. There is very limited evidence of malware that is either IPv6-aware (one rare example is a Zeus variant that is capable of sniffing IPv6 traffic [3]) or which uses IPv6 to communicate. However, that does not mean that IPv6 doesn't open up new possibilities for malware authors.

To begin with, IPv6 is new and, while it has been around for quite some time, it hasn't been used as extensively in the wild as IPv4 has. The protocol – or, more likely, implementations using it – may carry undiscovered vulnerabilities (in fact, it would be a miracle if they didn't exist⁹). Such vulnerabilities are not always discovered by hackers with a bright white hat and even if they are, slow patching means that there will be ample opportunity

⁹Both the Ping of Death and Teardrop denial-of-service attacks utilized incorrect handling of specifically crafted IPv4 packets. It is unlikely that these attacks will work against IPv6 implementations, but history has shown us time and again that new applications will have vulnerabilities inside them.

for the bad guys to take advantage of them. Of course it is impossible to predict the implications of these yet undiscovered vulnerabilities, but they are likely to be serious. The least the security industry can do is to make sure it has a good understanding of IPv6 and that it is ready to act when needed.

Because it is so new, merely using IPv6 may already mean that malicious traffic is leaving the network undetected, simply because security applications aren't aware of it. As we have seen, via Teredo, 6to4 and 4in6, IPv6 adds more possibilities to tunnel network traffic. By combining various kinds of tunnelling, a similar situation may occur as seen in the obfuscation of malicious files: it's easy to detect if you know what is happening, but if you don't, it might be hard to figure out what's really going on.

We mentioned before how the increase in the use of CGNs to overcome the lack of available IPv4 addresses has security implications. However, so does doing away with NAT – which is exactly what IPv6 does.

This means that every IPv6-connected device is publicly routable and not protected by the implicit firewall present on a NAT. Any IPv6-connected device that is controlled by cybercriminals (for instance, because it is part of a botnet), and which is not protected by a properly configured firewall, can thus be used as a DNS server, a web server, a command-and-control server, etc. The fact that such devices automatically have end-to-end connectivity also makes the running of a peer-to-peer botnet a lot easier.

At the moment, it may not seem likely that this will happen. After all, the percentage of devices with IPv6 connectivity (those where the device has it enabled and the router and ISP support it) is small and the increase in 'quality' for such an IPv6 botnet is unlikely to weigh up against the significant decrease in quantity. However, we have seen how protocols such as Teredo/Miredo and 6to4 easily give IPv4-connected devices an IPv6 connection. It is not hard to imagine an advanced piece of malware doing exactly this.

We have seen above how the sheer size of the IPv6 address space has serious implications for spam filtering. It has consequences for malware and network filtering too.

Currently, most households and small offices use a /24 (e.g. 192.168.0.0/24) as a LAN, which gives (almost) 256 possible IPv4 addresses. Some organizations may use larger LANs or have been assigned a larger IPv4 block, but even on a /16 there are slightly less than 65,536 possible addresses. It is not difficult to run a script that checks them all.

In IPv6, most ISPs won't assign blocks smaller than a /64. There are over 18 quintillion (18,446,744,073,709,551,

616 to be precise) possible addresses in such a block. It is impossible to check them all once, let alone regularly. IPv6 addresses for hosts on a network are assigned based on both the network's IP block and, by default, the 48-bit MAC address of the device, but the MAC address does not necessarily have to be used and there is quite a bit of freedom here.

In fact, this freedom allows for devices to encode small amounts of information inside their IPv6 addresses. For some advanced pieces of malware it would be an interesting way to hide information in plain sight.

CONCLUSIONS

We have seen that the Internet is nowhere near as ready for IPv6 as it should be. Thankfully, at the moment it looks like this is the case for cybercriminals as well. As with every new protocol, IPv6 opens new possibilities for them: some can easily be identified, but many others are yet to be discovered. It is the security industry's task to protect Internet users against the former and to continue to look for the latter – and to respond quickly as new potential threats appear.

IPv6 is both necessary and exciting. Let's make sure it continues to be in the future.

ACKNOWLEDGEMENTS

Several people have been kind enough to answer my questions and to make suggestions for writing this article. I would like to thank Ben April (*Trend Micro*), Dreas van Donselaar (*SpamExperts*, ipv6whitelist.eu), Gunter Ollmann (*Damballa*), Wout de Natris (*De Natris Consult*), Ken Simpson (*Mailchannels*), Joe St Sauver (University of Oregon), Morton Swimmer (*Trend Micro*, *Virus Bulletin*) and Eric Vyncke (*Cisco*).

REFERENCES

- [1] Krebs, B. Spammers Hijack Internet Space Assigned to Egyptian President's Wife. <http://krebsonsecurity.com/2011/02/spammers-hijack-internet-space-assigned-to-egyptian-presidents-wife/>.
- [2] Van Donselaar, D. IPv6 mail server whitelist declaring war on botnets. *Virus Bulletin*, August 2011, p.15. <http://www.virusbtn.com/pdf/magazine/2011/201108.pdf>.
- [3] Ollmann, G. Botnet Feature Advancement and Zeus Tweaking. <http://blog.damballa.com/?p=438>.

CONFERENCE REPORT

BEHIND ENEMY LINES: REPORTING FROM THE CCC 28C3 CONGRESS

Morton Swimmer

Trend Micro & Virus Bulletin

For the past 28 years, the Chaos Computer Club has organized its Congress – covering 'technology, society and utopia' – between Christmas and New Year. For the past nine years, the Congress has been held in the Berlin Congress Center, right in the heart of Berlin on Alexanderplatz. The large, glass-walled building strikes one immediately as being rather inappropriate for a meeting of hackers – or perhaps appropriate for the transparency that many of the delegates wish to promote. (It is certainly a welcome departure from many conference venues that might as well be deep underground for all one could tell.)

The four-day Congress has grown over the years to attract far more international participants than the 3,000 that is the venue's capacity. For that reason, the last two years have seen the introduction of pre-paid tickets and an elaborate system that has been put in place to make it easier for would-be delegates to order tickets anonymously in advance (the system itself does not provide any anonymity, it just tries to facilitate it). As such, there were no four-day passes available at the door as these had sold out within minutes of the three separate ticket allotments coming online. Day passes could be obtained, with a bit of luck, for all days except the first.

One consequence of this new ticket regime was that the Congress had a different feel from previous years. While the rooms were not as ridiculously overpacked, it felt as if the usual spontaneity was lacking. In an attempt to accommodate the vast interest in the Congress, numerous parallel conferences and meetings were organized – for instance, BerlinSides_0x2, the cBase Sidebar and satellite events around the world – from which 'virtual' delegates could watch the live stream and pose questions via IRC.

The live streams themselves were excellent this year and the FeM team from the Technical University of Ilmenau was able to get most of the talks online for download within a day, allowing particularly obsessed delegates to watch parallel tracks (sometimes at the same time).

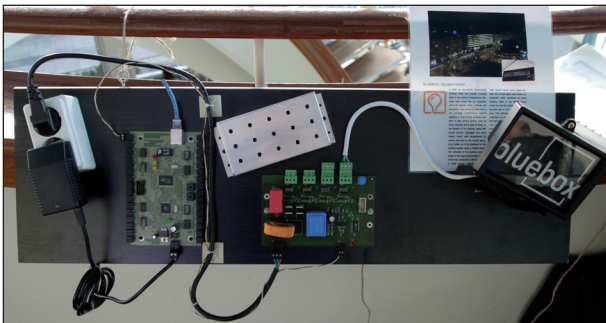
TALKS AND WORKSHOPS

The topics of the talks and workshops ranged from information society politics through technologies to arts and crafts. This year I (and others) lamented the lack of interesting arts projects and even the crafts (a.k.a. 'makers')

were not as prominent this year – though there were an alarming number of workshops on hacking food technology and Geiger counters. There were quite a few talks about German information society politics and a few more general political ones as well.

One of these ‘political’ talks was by Cory Doctorow, who departed a little from his core topic of copyright issues to remind us of copyright’s evil twin: the DRM (Digital Rights Management). technologies that are eroding our control over the devices we believe we own. Of course, DRM technologies are already in place on the *iPhone*, *iPad* and many *Android* devices, but Doctorow believes this will extend to laptops and PCs in the future and he decried the death of general computing that has benefited society so much. It was apt that, shortly after his talk, government-mandated backdoors were discovered in *iPhone* devices.

Once again, the Congress sported its own GSM base station that one could register with (sporting an SMS to Telex gateway), and there was a continuation of the talks given at previous Congresses about GSM security. GSM security is still largely broken, but many providers have at least pledged to upgrade their networks. Karten Nohl and Luca Melette introduced an instrumented phone that can determine whether the GSM provider has updated their security in a particular area. Using this, they were able to document the roll-out of the security updates. They also mentioned that conversation eavesdropping devices for GSM were far cheaper and more pervasive than expected.



Dan Kaminski gave his usual Black-Ops of TCP/IP talk, in which he broached many subjects that he has worked on over the past year. Bitcoin was one of these and, together with Travis Goodspeed, he demonstrated the use of Bitcoin as a form of permanent information storage. But he also explained that Bitcoin suffers from scalability and anonymity issues, though there are no known security issues in the core code, despite being very opaque indeed. The uPNP protocol is always a bundle of trouble on the LAN side, but now it turns out that some routers also listen to the uPNP protocol on the WAN side. Oh, joy. Dan also talked at BerlinSides_0x2, where he predicted that IPv6 is coming,

and will probably be important for P2P voice communication on smartphones. SOPA came up in the context of DNSSEC, which will break if arbitrary domains are blocked. On that subject he also believed that Certificate Authorities are on the way out, and a possible replacement may be the Electronic Frontier Foundation (EFF) Sovereign Keys proposal.

Back at the Congress, the EFF’s Peter Eckersley explained this alternative to the current hierarchical CA infrastructure. Users are now so used to clicking through certificate warnings that they are not much use. The EFF’s proposed alternative, Sovereign Keys, removes the need for CAs and allows domain owners to deploy their TLS keys directly. While this sounds promising, there are still many issues to be worked out.

There were a few talks about Tor and similar traffic anonymizing systems. Much was made of the fact that many governments are trying to block Tor nodes. Eric Filiol’s team claimed to have broken the Tor security by demonstrating that a Tor node could be compromised and then traffic directed through it. By engineering a weak key, that traffic could then be sniffed. The Tor team, who were present, countered that the vulnerability he described was already patched, and that the routing mechanisms would actively try to prevent such traffic redirection.

DC+ was presented as an alternative to Tor. This peer-to-peer system, where all participants receive all messages but can only decrypt the ones intended for them, turns out to be an incredibly slow anonymizing overlay network. Given that the performance of such a system is very poor, it is a fair way from being ready for prime time.

There was much hallway chatter about alternatives to the DNS system as a response to government censorship and initiatives like SOPA/PIPA. Tor, for instance, has the Tor2Web services for accessing anonymous sites. In general, it is far from clear whether any DNS alternative could possibly scale as well as DNS – a point that was confirmed by Dan Kaminsky.

Perhaps the best session (in my opinion) was Travis Goodspeed’s talk entitled ‘Packets in Packets’. He debunked the myth that the ISO network layers completely encapsulate each other by demonstrating a Layer 1 packet insertion from a Layer 7 protocol. In this way, data sent by HTTP could attack an unrelated machine on a local network. He first showed this on the Zigbee Layer 1 (802.15.4) and then on the more sophisticated 802.11B protocol. The trick is to realize that radio (or wire) signals look for certain patterns to mark the beginning of network frames and can be fooled by specially crafted contents being sent in a larger packet than they match on.

Artur Janc of *Google* demonstrated techniques that can be used to create backdoors in a browser session using

a method he calls 'resident XSS'. The premise is that an exploit can take up residence in the client-side storage or cache and persist over multiple sessions, creating a backdoor into the user's web client. Mitigating this is hard for the web application developer as it is really a web client problem. Also, HTML5 will make these attacks easier as it has more elaborate storage methods.

Mathias Payer, from ETH Zürich, introduced a framework for crafting format string attacks, which he calls string-oriented programming. While it's nice to see that DEP, ASLR and ProPolice have made code injection a lot harder, he showed that it is still possible to insert malicious code through other means. Given that there is a market for exploits, people will be motivated to create them, despite the complexity.

OTHER THINGS

Taking a cue from other hacking conferences, this year, the Congress organizers produced an electronic badge. It wasn't the official access badge: that still consisted of a wrist band crimped onto the wearer's wrist. The 'R0cket', as it was called, featured a little backlit LCD matrix display, a wireless mesh network transceiver, two buses (one for shields and one for lower-level hacking), a five-way mini joystick, a USB connector for programming and power, a light sensor and probably other features I missed. All this for EUR 30 if you were willing to stand hours in line for it.

As usual, there was also a hack centre in the basement where undisclosed stuff probably happened, but it had a less interesting feel than in previous years.

SUMMARY

As usual, there was far too much to report on and I've left a lot out here. The complete schedule is available at: <http://events.ccc.de/congress/2011/Fahrplan/> and this includes some links to slides and other material. The videos from some of the conference sessions are available at: <http://events.ccc.de/congress/2011/wiki/Documentation>. Next year, the organizers are debating moving to a different venue to accommodate the growing number of delegates. In any case, the CCC event will likely remain the premier hacker event in Europe for many years to come.



CALL FOR PAPERS

VB2012 DALLAS

Virus Bulletin is seeking submissions from those wishing to present papers at VB2012, which will take place 26–28 September 2012



at the Fairmont Dallas hotel, Dallas, TX, USA.

The conference will include a programme of 30-minute presentations running in two concurrent streams: Technical and Corporate.

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

A list of topics suggested by the attendees of VB2011 can be found at <http://www.virusbtn.com/conference/vb2012/call/>. However, please note that this list is not exhaustive, and the selection committee will consider papers on these and any other anti-malware and anti-spam related subjects.

SUBMITTING A PROPOSAL

The deadline for submission of proposals is **Friday 9 March 2012**. Abstracts should be submitted via our online abstract submission system. You will need to include:

- An abstract of approximately 200 words outlining the proposed paper and including five key points that you intend the paper to cover.
- Full contact details.
- An indication of whether the paper is intended for the Technical or Corporate stream.

The abstract submission form can be found at <http://www.virusbtn.com/conference/abstracts/>.

One presenter per selected paper will be offered a complimentary conference registration, while co-authors will be offered registration at a 50% reduced rate (up to a maximum of two co-authors). *VB* regrets that it is not able to assist with speakers' travel and accommodation costs.

Authors are advised that, should their paper be selected for the conference programme, they will be expected to provide a full paper for inclusion in the VB2012 Conference Proceedings as well as a 30-minute presentation at VB2012. The deadline for submission of the completed papers will be 6 June 2012, and potential speakers must be available to present their papers in Dallas between 26 and 28 September 2012.

Any queries should be addressed to editor@virusbtn.com.


END NOTES & NEWS

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>.

APWG eCrime Researchers Sync-Up takes place 7–8 March 2012 in Dublin, Ireland. For more information see <http://www.ecrimeresearch.org/2012syncup/cfp.html>.

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. 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?'. 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.