



# Abertay University

## **Network Forensics Investigation**

*National Security Agency International Sporting Competition  
Corruption Case*

**Marc Geggan**

1902548

CMP416: Advanced Digital Forensics

2022/2023

*\*Note that Information contained in this document is for educational purposes.*

# +Contents

---

1	Introduction .....	1
1.1	Aim of Investigation .....	1
1.2	Tools .....	1
2	Investigation.....	2
2.1	Capture 1 .....	2
2.1.1	Requirements.....	2
2.1.2	Packet Capture Investigation .....	2
2.1.3	Downloaded File Investigation.....	3
2.2	Capture 2 .....	6
2.2.1	Requirements.....	6
2.2.2	Packet Capture Investigation .....	6
2.2.3	Downloaded File Investigation.....	6
2.2.4	Image Investigation .....	7
2.2.5	Suspect Details .....	9
2.2.6	Steganography Investigation .....	9
2.3	Capture 3 .....	10
2.3.1	Requirements.....	10
2.3.2	Packet Capture Investigation .....	10
2.3.3	Meeting Details Investigation .....	11
3	Discussion.....	13
3.1	Critical Evaluation .....	13
3.2	Reflection.....	13
	References .....	14
	Appendices.....	15
	Appendix 1 – Actual Documents .....	15
	Appendix 2 – Enter The WuTang.....	18
	Appendix 3 – More Documents.....	20
	Appendix 4 – Recovered Images .....	22
	Appendix 5 – Fixed.py.....	24
	Appendix 6 – Conversation .....	26



# 1 INTRODUCTION

## 1.1 AIM OF INVESTIGATION

---

The National Security Agency has requested a digital forensic investigation into an international sporting competition corruption case. They require an investigation into three network capture files with the aim of recovering evidence and other relevant information that will aid in the inquiry. The investigator has been provided with three captures of interest, and a document must be created to provide the recovered evidence. This deliverable will be presented in such a way that the investigation may be recreated by other parties. Forensic integrity techniques such as file hashing and read-only evidence folders will also be considered to ensure the data is handled securely.

## 1.2 TOOLS

---

Name of Tool	Use of Tool
Wireshark	Used to analyse the captured traffic pcap files and filter through all important evidence.
CyberChef	Decode messages within obfuscated files.
Google Translate	Translate decoded Russian text.
Binwalk	Examine steganography techniques used to obfuscate data and files.
Find	Produce tree view of folders.
Tshark	Carve data from pcap file and output it to a separate file for examination.
CSV to KML Converter	Convert CSV coordinate data into KML coordinate data.
Google Maps	Read and output KML coordinate data.
SilentEye	Decode hidden messages.

# 2 INVESTIGATION

## 2.1 CAPTURE 1

### 2.1.1 Requirements

An investigation must be carried out to recover names and aliases of actors within the case suspected of downloading files of interest. This includes usernames, obfuscation techniques, and how they were recovered.

### 2.1.2 Packet Capture Investigation

After an investigation into the protocols used it was discovered that files of interest had been downloaded using the SMB protocol (See Figure 1).

The SMB protocol starts at packet **5857**, and within this packet a source IP address of **172.29.1.23** is visible. Further analysis of the SMB session reveals the full username within packet **23844** as **'fox-ws'** along with the host IP and MAC address information (See figure 2).

No.	Time	Source	Destination	Protocol	Length	Info
5857	23.8192929	172.29.1.23	172.29.1.20	BROWSER	418	1011 Announcement For: fox-ws, Workstation, Server, Print Queue Server, NT Workstation, Potential Browser, Backup Browser
5898	243.765960	172.29.1.23	172.29.1.20	SMB	213	213 Negotiate Protocol Request
5899	243.766459	172.29.1.20	172.29.1.23	SMB	143	143 Negotiate Protocol Response
5900	243.934327	172.29.1.23	172.29.1.20	SMB	162	162 Session Setup AndX Request, NTLMSSP_NEGOTIATE
5901	243.934826	172.29.1.20	172.29.1.23	SMB	319	319 Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
5903	244.118929	172.29.1.23	172.29.1.20	SMB	238	238 Session Setup AndX Request, NTLMSSP_AUTH, User: \
5904	244.119929	172.29.1.20	172.29.1.23	SMB	175	175 Session Setup AndX Response
5906	244.279556	172.29.1.23	172.29.1.20	SMB	136	136 Tree Connect AndX Request, Path: \\D0G-WS\IPCS
5907	244.279811	172.29.1.20	172.29.1.23	SMB	114	114 Tree Connect AndX Response
5908	244.336758	172.29.1.23	172.29.1.20	LANMAN	172	172 NetServerEnum2 Request, Domain Enum
5909	244.337256	172.29.1.20	172.29.1.23	LANMAN	138	138 NetServerEnum2 Response
5910	244.340753	172.29.1.23	172.29.1.20	LANMAN	186	186 NetServerEnum2 Request, Workstation, Server, SQL Server, Domain Controller, Backup Controller, Time Source, Apple Server, Novell
5911	244.341003	172.29.1.20	172.29.1.23	LANMAN	193	193 NetServerEnum2 Response

Figure 1 – SMB protocol packets

No.	Time	Source	Destination	Protocol	Length	Info
23844	642.004506	172.29.1.23	172.29.1.20	SMB	564	564 Session Setup AndX Request, NTLMSSP_AUTH, User: fox-ws\test
23845	642.006721	172.29.1.20	172.29.1.23	SMB	175	175 Session Setup AndX Response
23848	642.075412	172.29.1.23	172.29.1.20	SMB	136	136 Tree Connect AndX Request, Path: \\D0G-WS\IPCS

Frame 23844: 564 bytes on wire (4512 bits), 564 bytes captured (4512 bits) on interface 0  
Ethernet II, Src: Dell\_EA:80:35:60, Dst: HewlettP\_00:0C:29:00:00:00 (00:0C:29:00:00:00)  
Internet Protocol Version 4, Src: 172.29.1.23, Dst: 172.29.1.20  
Transmission Control Protocol, Src Port: 58291, Dst Port: 445, Seq: 268, Ack: 355, Len: 450  
NetBIOS Session Service  
SMB (Server Message Block Protocol)

Figure 2 – Details of SMB session creator

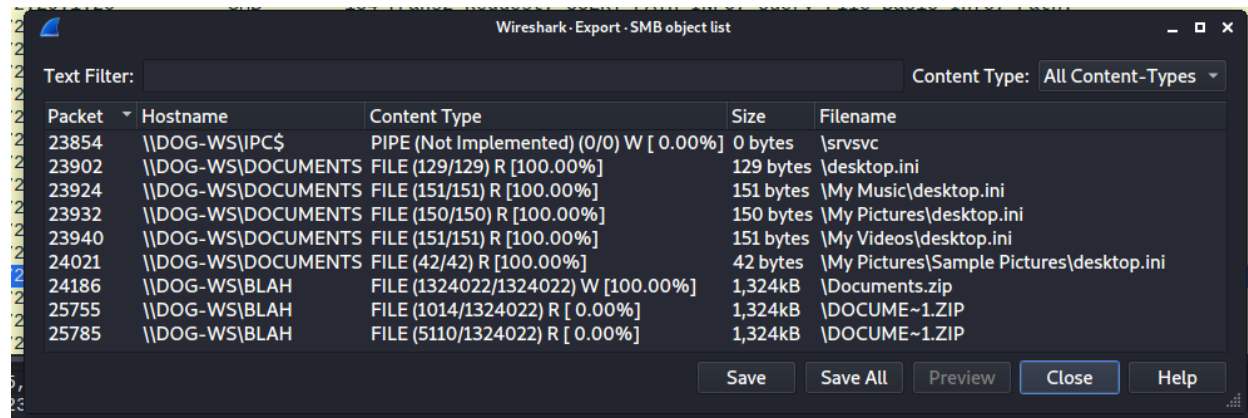
The **'FIND\_FIRST 2'** data within packet **23917** reveals that the suspect had accessed multiple files within the target's directory (See figure 3) and further evidence of accessed files within packet **24008**, however there is no evidence supporting that all files were downloaded.

No.	Time	Source	Destination	Protocol	Length	Info
23917	650.620640	172.29.1.20	172.29.1.23	SMB	462	462 Trans2 Response, FIND_FIRST2, Files: Customer Accounts, desktop.ini, drop folder, larryeatswrt-with-secretsauce.jpg, larryeatswrt-with-secretsauce.bmp
23919	651.001860	172.29.1.23	172.29.1.20	SMB	180	180 NT Create AndX Request, FID: 0x4004, Path: \\My Music\desktop.ini

Data Offset: 68  
Data Displacement: 0  
Setup Count: 0  
Reserved: 00  
Byte Count (BCC): 1749  
Padding: 00  
FIND\_FIRST2 Parameters  
Padding: 0000  
FIND\_FIRST2 Data  
Find File Both Directory Info File: ..  
Find File Both Directory Info File: ..  
Find File Both Directory Info File: Customer Accounts  
Find File Both Directory Info File: desktop.ini  
Find File Both Directory Info File: drop folder  
Find File Both Directory Info File: larryeatswrt-with-secretsauce.jpg  
Find File Both Directory Info File: larryeatswrt-with-secretsauce.bmp  
Find File Both Directory Info File: My Music  
Find File Both Directory Info File: My Pictures  
Find File Both Directory Info File: My Videos  
Find File Both Directory Info File: premium-customer-billing-list.xls  
Find File Both Directory Info File: production  
Find File Both Directory Info File: R and D  
Find File Both Directory Info File: Thumbs.db

Figure 3 – List of files accessible by suspect in packet 23917

With Wireshark's exporting tools, it is possible to recover the files that were downloaded by the suspect over the SMB protocol (**File > Export Objects > SMB**). The 9 files recovered have been downloaded from a host name of '**dog-ws**'. Multiple .ini files were downloaded, however these can be disregarded from investigation. A file of interest named **Documents.zip** has been identified (See figure 4).



Wireshark - Export - SMB object list

Text Filter: Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
23854	\\DOG-WS\IPC\$	PIPE (Not Implemented) (0/0) W [ 0.00%]	0 bytes	\srvsvc
23902	\\DOG-WS\DOCUMENTS	FILE (129/129) R [100.00%]	129 bytes	\desktop.ini
23924	\\DOG-WS\DOCUMENTS	FILE (151/151) R [100.00%]	151 bytes	\My Music\desktop.ini
23932	\\DOG-WS\DOCUMENTS	FILE (150/150) R [100.00%]	150 bytes	\My Pictures\desktop.ini
23940	\\DOG-WS\DOCUMENTS	FILE (151/151) R [100.00%]	151 bytes	\My Videos\desktop.ini
24021	\\DOG-WS\DOCUMENTS	FILE (42/42) R [100.00%]	42 bytes	\My Pictures\Sample Pictures\desktop.ini
24186	\\DOG-WS\BLAH	FILE (1324022/1324022) W [100.00%]	1,324kB	\Documents.zip
25755	\\DOG-WS\BLAH	FILE (1014/1324022) R [ 0.00%]	1,324kB	\DOCUME~1.ZIP
25785	\\DOG-WS\BLAH	FILE (5110/1324022) R [ 0.00%]	1,324kB	\DOCUME~1.ZIP

Buttons: Save Save All Preview Close Help

Figure 4 – Files transferred over SMB protocol and host name.

The files were saved to the investigators machine in a read-only folder and hashes were generated to ensure no data is modified at any stage of the investigation.

### 2.1.3 Downloaded File Investigation

Using the find command, a full tree of files within the Documents folder can be seen. There is multiple .docx files, one .jpeg file, one .jpg file, one .txt file, and another zipped folder named 'untitled folder.zip' (See figure 5).

```
(kali@kali)-[~/Desktop/Investigation/Capture1]
$ find Documents
Documents
Documents/More Documents
Documents/More Documents/NorthKorea.jpeg
Documents/More Documents/BillofRights.txt
Documents/Enter the WuTang
Documents/Enter the WuTang/track6.docx
Documents/Enter the WuTang/track10.docx
Documents/untitled folder.zip
Documents/Actual Documents
Documents/Actual Documents/PiD.docx
Documents/Actual Documents/NorthKorea.docx
Documents/Actual Documents/GoT Spoilers.docx
Documents/Chess Boxing
Documents/Chess Boxing/Rules 2.docx
Documents/Chess Boxing/Rules 5.docx
Documents/Chess Boxing/Rules 3.docx
Documents/Chess Boxing/Rules 1..docx
Documents/Chess Boxing/Rules 6.docx
Documents/Chess Boxing/Rules 4.docx
Documents/Chess Boxing/NK.jpg
Documents/Chess Boxing/Rules 7.docx
```

Figure 5 – Contents of Documents.zip folder

Investigation into '**untitled folder.zip**' revealed file path of **/untitled folder/untitled folder/untitled folder 2/untitled folder/untitled folder/SilentEye**. The file within the folder path appears to be empty, however SilentEye refers to a tool used to hide images or audio within image files using steganography techniques (Alejandro, 2011).

#### 2.1.3.1 Folder - Actual Documents

Within the 'Actual Documents' folder there are three .docx files named **GoT Spoilers**, **NorthKorea**, and **PiD**. All three files contained base64 encoded text. **CyberChef** was used to decode the messages within each file, and the encoded and decoded versions can be seen in *Appendix 1 – Actual Documents*.

The first file, **GoT Spoilers.docx** contains information relating to a TV series. The second file, **NorthKorea.docx** contained encoded Russian text which was ran through google translate. The message explained a conspiracy that North Korea had developed time travel equipment. The final file named **PiD.docx** is about the death of Paul McCartney in 1966 and a man named William Campbell had replaced him.

It can be concluded that these files are **not relevant** to the investigation and can be **disregarded**.

#### 2.1.3.2 Folder – Chess Boxing

When inspecting the 'Chess Boxing' folder there are 7 .docx files named **Rules 1 - Rules 7**, and a file called **NK.jpg**. All rules(1-7).docx contained base64 encoded text that when decoded revealed information relating to 'chess boxing' games. This information is deemed **not relevant** to the investigation.

The file named **NK.jpg** contained an image of the North Korean flag. The tool named 'Binwalk' was used to analyse the file for obfuscated evidence however nothing was recovered except details describing it as a **PNG Image** (See figure 6).

```
(kali㉿kali)-[~/../Investigation/Capture1/Documents/Chess Boxing]
$ binwalk NK.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 1600 x 800, 8-bit/color RGBA, non-interlaced

Figure 6 – Binwalk of NK.jpg

#### 2.1.3.3 Folder – Enter The WuTang

There are 2 files within the folder named 'Enter The WuTang'. These files have .docx extensions and are named **track6** and **track10**. Both files contain base64 encoded text. When **track6** was decoded, it revealed a list of 'usernames' under the title 'The Mystery of Chess Boxing'. These names **may be of interest** within the investigation as other suspects. The en/decoded text can be found in *Appendix 2 – Enter The WuTang*.

The file named **track10**, contained the lyrics to Wu-Tang Clan's 'Protect Ya Neck'. This was deemed **not relevant** to the investigation.

#### 2.1.3.4 Folder – More Documents

There are 2 files within the 'More Documents' folder named **BillOfRights.txt** and **NorthKorea.jpeg**. The text file named **BillOfRights.txt** contained an English plain-text version of the American bill of rights. This file was deemed **not relevant** to the investigation and disregarded.

As there was evidence referring to SilentEye, a steganography tool, this image was also run through the tool 'Binwalk' for any hidden data. The tool revealed a zipped folder named **untitled.zip**, and within this folder a file named **broken.py** (See figure 7). The .py extension suggests that the file is a python script and can be seen in *Appendix 3 – More Documents*.

```
(kali@kali)-[~/Investigation/Capture1/Documents/More Documents]
$ binwalk NorthKorea.jpeg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
3453	0xD7D	Zip archive data, at least v2.0 to extract, name: untitled/
3492	0xDA4	Zip archive data, at least v2.0 to extract, compressed size: 604, uncompressed s
ize: 1397, name: untitled/broken.py		
4263	0x10A7	End of Zip archive, footer length: 22

Figure 7 – Binwalk of NorthKorea.jpeg



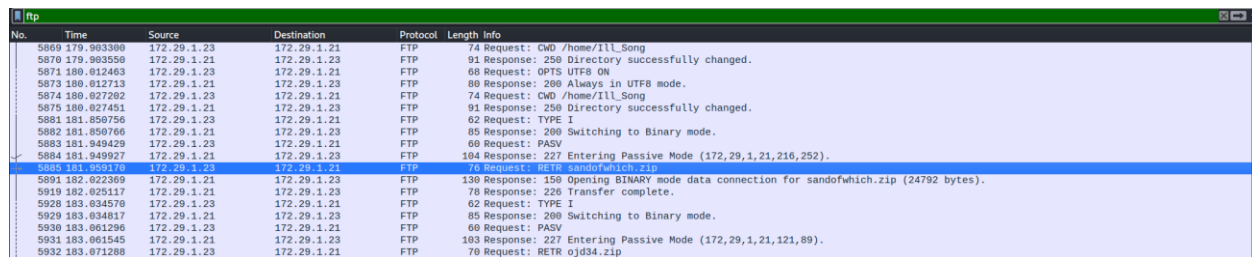
## 2.2 CAPTURE 2

### 2.2.1 Requirements

FTP and other traffic between a suspected actor and foreign national must be investigated, decoded, and evidence of data transferred must be recovered. Evidence must include information on what item was received by the foreign national.

### 2.2.2 Packet Capture Investigation

Information provided to the investigator suggests that files have been transferred through the File Transfer Protocol (FTP) between a suspect and foreign national. Due to this information, the network capture was filtered for all FTP traffic. This revealed two files of interest named **sandofwich.zip** at **packet 5885**, and **ojd34.zip** at **packet 5932** (See figure 8).



No.	Time	Source	Destination	Protocol	Length	Info
5869	179.903390	172.29.1.23	172.29.1.21	FTP	74	Request: CWD /home/ill_Song
5870	179.903550	172.29.1.21	172.29.1.23	FTP	91	Response: 250 Directory successfully changed.
5871	180.012463	172.29.1.23	172.29.1.21	FTP	68	Request: OPTS UTF8 ON
5873	180.012713	172.29.1.21	172.29.1.23	FTP	80	Response: 200 Always in UTF8 mode.
5874	180.027202	172.29.1.23	172.29.1.21	FTP	74	Request: CWD /home/ill_Song
5875	180.027451	172.29.1.21	172.29.1.23	FTP	91	Response: 250 Directory successfully changed.
5881	181.850756	172.29.1.23	172.29.1.21	FTP	62	Request: TYPE I
5882	181.850766	172.29.1.21	172.29.1.23	FTP	85	Response: 200 Switching to Binary mode.
5883	181.849429	172.29.1.23	172.29.1.21	FTP	60	Request: PASV
5884	181.849927	172.29.1.21	172.29.1.23	FTP	104	Response: 227 Entering Passive Mode (172,29,1,21,216,252).
5885	181.959178	172.29.1.23	172.29.1.21	FTP	70	Request: RETR sandofwich.zip
5891	182.022369	172.29.1.21	172.29.1.23	FTP	130	Response: 150 Opening BINARY mode data connection for sandofwich.zip (24792 bytes).
5919	182.025117	172.29.1.21	172.29.1.23	FTP	70	Response: 226 Transfer complete.
5928	183.034570	172.29.1.23	172.29.1.21	FTP	62	Request: TYPE I
5929	183.034817	172.29.1.21	172.29.1.23	FTP	85	Response: 200 Switching to Binary mode.
5930	183.061296	172.29.1.23	172.29.1.21	FTP	60	Request: PASV
5931	183.061545	172.29.1.21	172.29.1.23	FTP	103	Response: 227 Entering Passive Mode (172,29,1,21,121,89).
5932	183.071288	172.29.1.23	172.29.1.21	FTP	70	Request: RETR ojd34.zip

Figure 8 – FTP filtered capture revealing two files of interest

The zipped files were downloaded by **filtering for FTP-data > Right click file > Follow > TCP stream > Data: Raw > Save as**. This was done for both files, and they were saved in a read only folder on the investigators device.

### 2.2.3 Downloaded File Investigation

The 'find' command was run against both folders to produce an overview of contents within both folders (See figure 9). Both folders contain **multiple .jpg** image files with **one-word filenames**.

```
(kali@kali)-[~/Desktop/Investigation/Capture2]
$ find ojd34/ sandofwich
ojd34/
ojd34/web-based.jpg
ojd34/and.jpg
ojd34/terrorism.jpg
ojd34/building.jpg
ojd34/around.jpg
ojd34/cant.jpg
ojd34/conscience.jpg
ojd34/basic.jpg
ojd34/allow.jpg
ojd34/Watergate.jpg
sandofwich
sandofwich/for.jpg
sandofwich/security.jpg
sandofwich/government.jpg
sandofwich/rights.jpg
sandofwich/destroy.jpg
sandofwich/in.jpg
sandofwich/freedom.jpg
sandofwich/NSA.jpg
sandofwich/I.jpg
sandofwich/good.jpg
```

Figure 9 – Overview of extracted zip files from FTP data

The investigator was given a suspected tip-off regarding obfuscation techniques stating that the suspect had used a famous Edward Snowden quote to disguise data. The one-word filenames fit the famous quote:

*'I can't in good conscience allow the U.S. government to destroy privacy, internet freedom and basic liberties for people around the world with this massive surveillance machine they're secretly building.'* (Snowden, 2013).

Every file within both folders appeared to be somewhat corrupted, and it was apparent when compared to the Edward Snowden quote that not all files were recovered during the investigation. This prompted further investigation into the packet capture. Carving the capture for zipped files located **packet 2666** and **packet 8190**. Within the **MIME data** of **packet 2666**, **2 zipped files** were extracted named **34jdsioj** and **breaking\_bad\_season\_6**. Within the **MIME data** of **packet 8190**, **1 zipped file** was extracted named **canc3l** (See Figure 10).



Figure 10 – 3 extracted missing files from packet capture

The Binwalk tool was used to extract the data from the zipped folders and a full list of extracted images can be found at *Appendix 4 – Recovered Images*. Comparing all filenames with Edward Snowden's quote, it can be confirmed there are enough words to match.

## 2.2.4 Image Investigation

Using the Cat command (See figure 11), the investigator was able to piece together the image fragments within the files according to the Snowden quote, and output to a .jpg file producing a fully viewable image of a chessboard (See figure 12).

```
(kali@kali)~/Desktop/Investigation/Capture2/Images
$ cat I.jpg cant.jpg in.jpg good.jpg conscience.jpg allow.jpg the.jpg U.S..jpg government.jpg to.jpg destroy.jpg privacy.jpg internet.jpg freedom.jpg
and.jpg basic.jpg liberties.jpg for.jpg people.jpg around.jpg world.jpg with.jpg this.jpg massive.jpg surveillance.jpg machine.jpg theyre.jpg secretly
.jpg building.jpg > ../reconstructed/snowdenQuote.jpg
(kali@kali)~/Desktop/Investigation/Capture2/Images
$
```

Figure 11 – Cat command to stitch images together

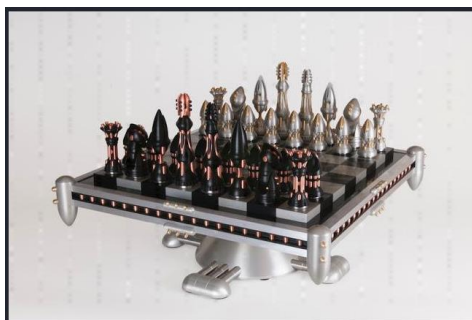


Figure 12 – Output image of Cat command

It was observed that not every file recovered by the investigator was used within the quote, leaving extra images to be investigated. **Condone.jpg** and **There.jpg** appeared to be two separate images unrelated to the Snowden quote. The investigator filtered out the previously used images and through a form of trial-and-error Bruteforcing, two additional images were recovered. The cat command was used on the first image (See figure 13) and after various attempts an image was produced (See figure 14).

```
(kali@kali)-[~/Desktop/Investigation/Capture2/Images]
$ cat condone.jpg American.jpg web-based.jpg rights.jpg constructing.jpg security.jpg terrorism.jpg NSA.jpg Watergate.jpg corrupt.jpg human.jpg behind.jpg closed.jpg doors.jpg > ../reconstructed/condoneReconstructed
(kali@kali)-[~/Desktop/Investigation/Capture2/Images]
```

Figure 13 – Cat command for condone.jpg image



Figure 14 – Output recovered image

The investigator moved onto the second partly visible image 'there.jpg'. Again, the same procedure was employed to bruteforce an image using the cat command (See figure 15 & 16).

```
(kali@kali)-[~/Desktop/Investigation/Capture2/Images]
$ cat there.jpg their.jpg a.jpg it.jpg but.jpg communism.jpg nor.jpg because.jpg unconstitutional.jpg secretive.jpg secret.jpg > ../reconstructed/thereReconstructed
(kali@kali)-[~/Desktop/Investigation/Capture2/Images]
```

Figure 15 – Cat command for there.jpg image



Figure 16 – Output recovered image

### 2.2.5 Suspect Details

Following **TCP stream 42** revealed communication over email between **da.genius36@aol.com** and **kim.illsong@aol.com**. The email sent from 'The Gza' has a subject of '*Urgent*' with text '*You have made a bold claim, but I'd like to see some proof.*' Kim IllSong replies with '*Ask and you shall receive. You know where to find it.*' (See figure 17).

```
-l:size: 9pt;\n"><tt>You have made a bold claim but i'd like to see some  
proof.\n</tt></pre>\n</div>\n\n <!-- end of AOLMsgPart_0_9d590ffb-d065-4b77-8bec-  
1baaf46137daa -->\n\n\n\n</div>\n\n\n<font>"PlainBody": "Ask and you shall receive  
You know where to find it.\n\n\n\n\n\n\n-----Original Message-----\nFrom: The Gza <a.genius36@aol.com>\nTo: kim_iilsong <kim.iilsong@aol.com>\nSent: Thu, Jul 3,  
2014 2:18 pm\nSubject: Urgent!\n\nYou have made a bold claim but i'd like to see  
some proof.\n\n\n","RichEdit":true,"ParentMessageID":"<D1651316342A3C-21DC-B0AB@webmail-  
d267.sysops.aol.com>", "AnswerUID": "272", "AnswerFolder": "Inbox", "SourceMsgUID": "272"  
}, {"SourceFolder": "Inbox", "SourceAttachmentIDs": [], "PreloadAttachmentIDs":  
[], "SendAttachAsLinks": false, "ComposeType": "reply", "action": "SendMessage"}]  
-----506390528859906812396841278  
Content-Disposition: form-data; name="automatic"
```

false

-----506390528859906812396841278--

Figure 17 – TCP Stream 42 email conversation recovered between Kim and ‘The Gza’

### 2.2.6 Steganography Investigation

When tidying up evidence the inspector referred to capture 1 and the SilentEye file. This hinted to steganography obfuscation techniques, therefore ran the images through SilentEye. Within the Snowden chessboard image there was an encrypted message (*See figure 18*).

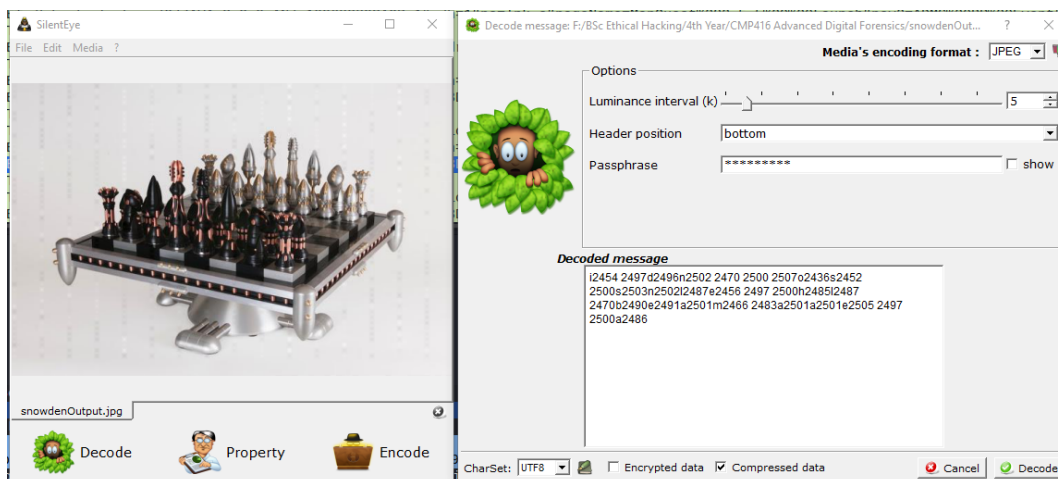


Figure 18 – Encrypted message output from Snowden chessboard image via SilentEye

Using a fixed version of the broken.py file that can be found in *Appendix 5 – Fixed.py*, the decoded message was ran through the python script and a message was recovered (*See figure 19*).

```
(kali㉿kali)-[~/Desktop/Investigation/Capture1/Documents]
└─$ python fixed.py -d "i2454 2497d2496n2502 2470 2500 2507o2436s2452 2500s2503n2502l2487e2456 2497 2500h2485l2487 2470b2490e2491a2501m2466 2483a2501a2501e2505 2497 2500a2486"
DontTry2BruteForceThisPassword
```

Figure 19 – Decoded message through fixed python script

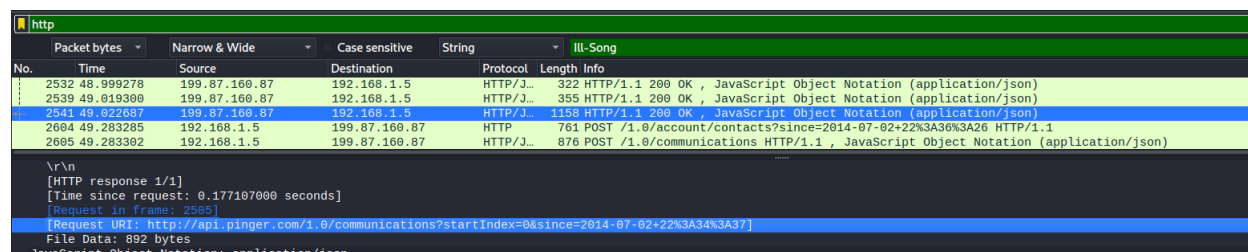
## 2.3 CAPTURE 3

### 2.3.1 Requirements

An investigation into the communication traffic between Ill-Song and Ann Dercover must be carried out. It is suspected that a secret meeting is being planned and the details of this meeting must be recovered.

### 2.3.2 Packet Capture Investigation

Investigator filtered packet data by searching for 'Kim Ill-Song', revealing conversations over HTTP requests. The first match is **packet 2541**, revealing a text from Kim Ill-Song 'Good Afternoon, Ann', and a link to where the POST request was first made, at **packet 2505** (See figure 18).

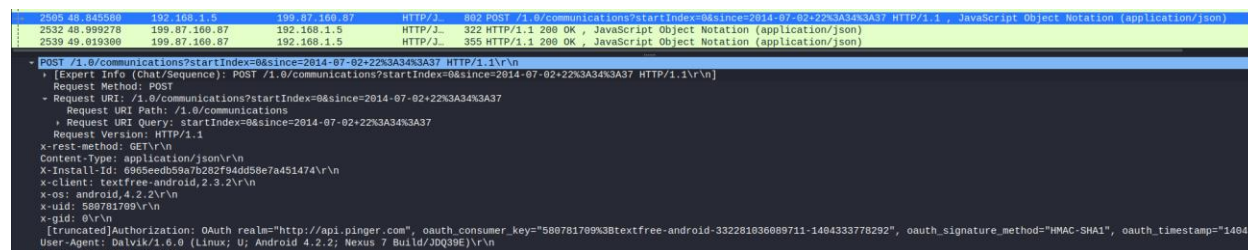


The image shows a Wireshark packet capture with a filter set to 'http'. A search for 'Ill-Song' has been performed, highlighting several packets. Packet 2541 is selected, showing its details in the bottom pane. The details pane shows the request URI: 'http://api.pinger.com/1.0/communications?startIndex=0&since=2014-07-02+22%3A34%3A37'.

No.	Time	Source	Destination	Protocol	Length	Info
2532	48.999278	199.87.160.87	192.168.1.5	HTTP/J...	322	HTTP/1.1 200 OK, JavaScript Object Notation (application/json)
2539	49.019309	199.87.160.87	192.168.1.5	HTTP/J...	355	HTTP/1.1 200 OK, JavaScript Object Notation (application/json)
2541	49.022337	199.87.160.87	192.168.1.5	HTTP/2...	1159	HTTP/1.1 200 OK, JavaScript Object Notation (application/json)
2604	49.283285	192.168.1.5	199.87.160.87	HTTP	761	POST /1.0/account/contacts?since=2014-07-02+22%3A36%3A26 HTTP/1.1
2605	49.283382	192.168.1.5	199.87.160.87	HTTP/J...	876	POST /1.0/communications HTTP/1.1, JavaScript Object Notation (application/json)

Figure 18 – Search of 'Ill-Song' revealing HTTP conversation

When visiting **packet 2505** and analyzing the HTTP data, information relating to the time and date, application used, and mobile device type can be found. It can be concluded by analyzing the **x headers**, that the device used was a **Nexus 7**, with **Android 4.2.2** operating system, and using a client named **'TextFree'** (See figure 19).

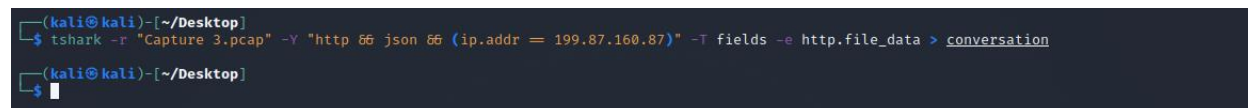


The image shows a Wireshark packet capture with a filter set to 'http'. A search for 'Ill-Song' has been performed, highlighting several packets. Packet 2505 is selected, showing its details in the bottom pane. The details pane shows the request URI: 'http://api.pinger.com/1.0/communications?startIndex=0&since=2014-07-02+22%3A34%3A37'. The x-headers section shows the following information: x-rest-method: GET, Content-Type: application/json, X-Install-Id: 0965ed859a7b282f9dd58e7a451474, x-client: textfree-android,2.3.2, x-os: android,4.2.2, x-uid: 580781799, x-gid: 0, [Truncated]Authorization: OAuth realm="http://api.pinger.com", oauth\_consumer\_key="58078179938textfree-android-332281036089711-1404333778292", oauth\_signature\_method="HMAC-SHA1", oauth\_timestamp="140434...

No.	Time	Source	Destination	Protocol	Length	Info
2505	49.045509	192.168.1.5	199.87.160.87	HTTP/1...	862	POST /1.0/communications?startIndex=0&since=2014-07-02+22%3A34%3A37 HTTP/1.1, JavaScript Object Notation (application/json)
2532	48.999278	199.87.160.87	192.168.1.5	HTTP/J...	322	HTTP/1.1 200 OK, JavaScript Object Notation (application/json)
2539	49.019309	199.87.160.87	192.168.1.5	HTTP/J...	355	HTTP/1.1 200 OK, JavaScript Object Notation (application/json)

Figure 19 – POST request displaying device information

Examining the data, it was clear they were all to and from an IP address of **199.87.160.87** and used **HTTP/JSON**. Entering this data into a Tshark command (See figure 20) extracted the whole conversation log that can be seen in *Appendix 6 – Conversation*. A table was produced by the investigator also to display the conversation in a more visibly appealing table (See table 1).



```
(kali@kali)~/Desktop
$ tshark -r "Capture.3.pcap" -Y "http && json && (ip.addr == 199.87.160.87)" -T fields -e http.file_data > conversation
(kali@kali)~/Desktop
$
```

Figure 20 – Tshark command to extract conversation data



### 2.3.3 Meeting Details Investigation

Name	Text
Kim Ill-Song	Good afternoon, Ann.
Ann Dercover	who is this?
Kim Ill-Song	Castling.
Ann Dercover	where are you?
Kim Ill-Song	I know I can't tell you that.
Ann Dercover	Do you know that there are people investigating Kim Ill-Song?
Kim Ill-Song	Of course. However, they will never know it is me behind the bribes.
Ann Dercover	<u>still</u> we should be careful. Pay attention. I want to meet in September at 5PM.
Kim Ill-Song	At our old meetup spot?
Ann Dercover	yes
Kim Ill-Song	What day?
Ann Dercover	I told you to pay attention.

Table 1 – Conversation between Ill-Song and Ann (Geggan, M. 2022)

It can be concluded from this conversation that Kim Ill-Song has another name ‘**Castling**’, and a **meeting** has been set up in **September at 5pm**, however the **exact date is unknown**. To locate this, the investigator focused back on the HTTP traffic within the pcap file. There were a high number of requests to a host named ‘**mob.mapquestapi.com**’. Research of this host revealed it is an online mapping program like Google Maps. The packet capture was filtered for this host and all HTTP packets were found to contain location coordinates (See figure 21).

No.	Time	Source	Destination	Protocol	Length	Info
19538	671.961896	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672 HTTP/1.1
19651	673.993840	192.168.1.5	207.200.102.1	HTTP	257	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857788085919735 HTTP/1.1
19735	675.963984	192.168.1.5	207.200.102.1	HTTP	257	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857658386219848 HTTP/1.1
19848	679.641093	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857513427719929 HTTP/1.1
19929	681.683639	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8574905395120821 HTTP/1.1
20821	683.642899	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857471466819538 HTTP/1.1

```

Frame Number: 19538
Frame Length: 258 bytes (2064 bits)
Capture Length: 258 bytes (2064 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:http]
[Coloring Rule Name: HTTP]
[Coloring Rule Strings: http || top.port == 80 || http2]
Ethernet II, Src: ASUSTekC:99:f4:d (00:a4:4c:99:f4:d), Dst: IntelCor_f0:ae:3e (00:26:c7:f9:ae:3e)
Internet Protocol Version 4, Src: 192.168.1.5, Dst: 207.200.102.1
Transmission Control Protocol, Src Port: 60358, Dst Port: 80, Seq: 1, Ack: 1, Len: 264
Hypertext Transfer Protocol
  GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672 HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672 HTTP/1.1\r\n]
    [GET /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672 HTTP/1.1\r\n]
    [Severity level: Chat]
    [Group: Sequence]
    Request Method: GET
    Request URI: /geocoding/v1/reverse?key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672
    Request URI Path: /geocoding/v1/reverse
    Request URI Query: key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857952117919922C-114.01103973388672
    Request URI Query Parameter: key=CmJtdn7Cluaa2qu2nd2Cb5k3Do5-gzb0
    Request URI Query Parameter: inFormat=kvp
    Request URI Query Parameter: outFormat=json
    Request URI Query Parameter: location=46.857952117919922C-114.01103973388672
    Request Version: HTTP/1.1
    Host: mob.mapquestapi.com\r\n
  
```

Figure 21 – Filtered packets with location coordinates

The filtered data was extracted to a CSV file for further data carving using **File > Export Packet Dissections > As CSV**. Two formulas were created within Excel to extract the coordinates from the lines of text, strip them of '%2C', and split them into X and Y coordinates (See figure 22 & 23)

B1			=IFERROR(MID(A1, SEARCH("location=", A1) + LEN("location="), SEARCH("%2C-", A1) - SEARCH("location=", A1) - LEN("location=")), "")		
A			B	C	
1	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85661315917969%2C-114.01860809326172 HTTP/1.1		46.85661315917969	-114.01860809326172	
2	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85693359375%2C-114.01863098144531 HTTP/1.1		46.85693359375	-114.01863098144531	
3	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85727310180664%2C-114.01868438720703 HTTP/1.1		46.85727310180664	-114.01868438720703	
4	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857601165771484%2C-114.01866912841797 HTTP/1.1		46.857601165771484	-114.01866912841797	
5	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858055114746094%2C-114.01866149902344 HTTP/1.1		46.858055114746094	-114.01866149902344	
6	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8582878112793%2C-114.01864624023438 HTTP/1.1		46.8582878112793	-114.01864624023438	
7	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858524322509766%2C-114.01863861083984 HTTP/1.1		46.858524322509766	-114.01863861083984	
8	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858734130859375%2C-114.01864624023438 HTTP/1.1		46.858734130859375	-114.01864624023438	

Figure 22 – Formula to extract X coordinate and sample of outputs

C1			=CONCATENATE("-", IFERROR(MID(A1, SEARCH("%2C-", A1) + LEN("%2C-"), SEARCH(" HTTP", A1) - SEARCH("%2C-", A1) - LEN("%2C-")), ""))		
A			B	C	
1	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85661315917969%2C-114.01860809326172 HTTP/1.1		46.85661315917969	-114.01860809326172	
2	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85693359375%2C-114.01863098144531 HTTP/1.1		46.85693359375	-114.01863098144531	
3	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85727310180664%2C-114.01868438720703 HTTP/1.1		46.85727310180664	-114.01868438720703	
4	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.857601165771484%2C-114.01866912841797 HTTP/1.1		46.857601165771484	-114.01866912841797	
5	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858055114746094%2C-114.01866149902344 HTTP/1.1		46.858055114746094	-114.01866149902344	
6	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8582878112793%2C-114.01864624023438 HTTP/1.1		46.8582878112793	-114.01864624023438	
7	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858524322509766%2C-114.01863861083984 HTTP/1.1		46.858524322509766	-114.01863861083984	
8	GET /geocoding/v1/reverse?key=Cmjtd%7Cluaa2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858734130859375%2C-114.01864624023438 HTTP/1.1		46.858734130859375	-114.01864624023438	

Figure 23 – Formula to extract Y coordinate and sample of outputs

The X and Y columns were extracted to a new CSV file and converted to a **KML file** so that the coordinates can be fed into Google Maps. Once uploaded to Google Maps, a map with multiple pins in the shape of a **17** can be seen. It can be concluded that it hints toward the date of the **meeting being the 17<sup>th</sup> day of the month** (See figure 24).

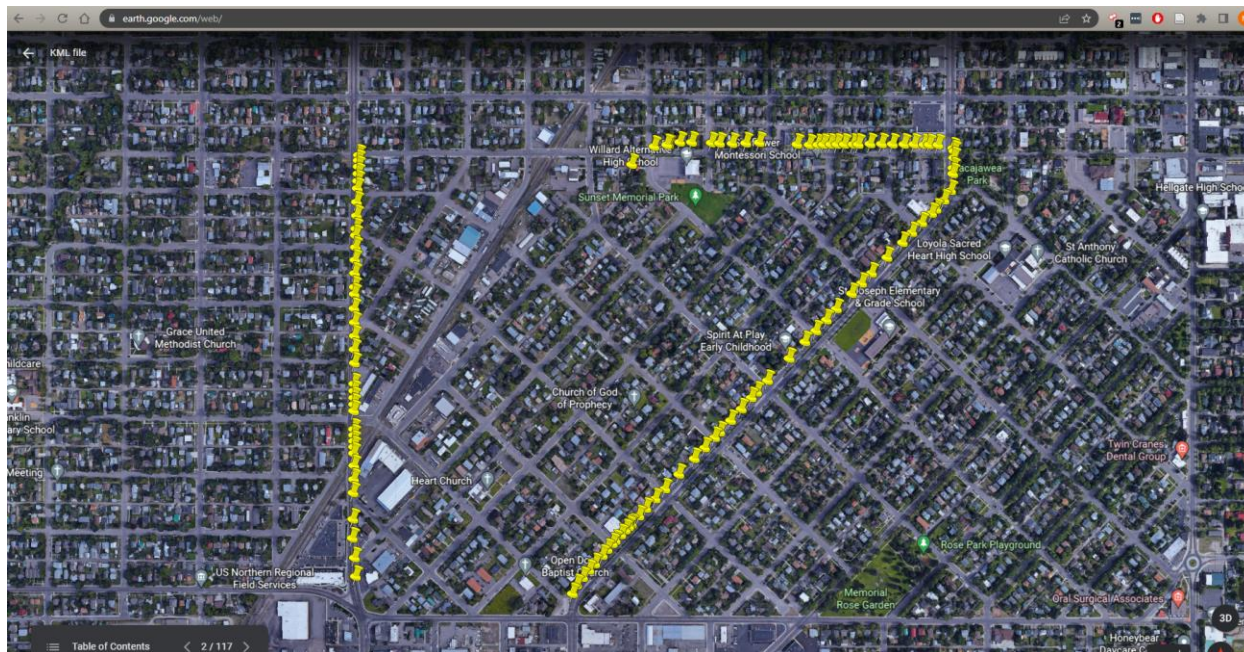


Figure 24 – KML file uploaded to Google Maps showing a number 17

## 3 DISCUSSION

### 3.1 CRITICAL EVALUATION

---

Within Capture 1.pcap a list of malicious actors' usernames was recovered, along with a python script for encoding, and two computer device names 'fox-ws' and 'dog-ws'. A reference to steganography tool 'SilentEye' was also discovered.

Moving onto Capture 2.pcap multiple folders had been transferred over FTP. These all contained corrupt images. Using hints, partial images were combined to create 3 full images. This was difficult as it took a lot of trial and error to get the right alignment. The image of the chess board was found to contain a hidden encoded message that could be decoded by fixing the previous python script. This again was difficult, and some code analysis was involved. These images along with the encoded message were sent between da.genius36@aol.com and kim.illsong@aol.com.

Finally, Capture 3.pcap revealed a secret meeting between Ann Dercover and Kim Ill-Song on the 17<sup>th</sup> of September at 5pm. This information was located through recovered message data and extensive requests to a MapQuest host, revealing a date obfuscated as geo-location data.

### 3.2 REFLECTION

---

A malicious hacker can use data obscuring techniques to make it difficult for investigators to access and analyze the data that they have stolen or compromised. For example, they may encrypt the data using a strong encryption algorithm, making it impossible to read without the decryption key. They may also use techniques like data fragmentation and data hiding to split the data into multiple pieces and conceal it within other files, making it difficult to locate and reconstruct. Additionally, hackers may use steganography to hide the data within other digital files, such as images or audio files, making it even harder to detect. All these techniques can make it difficult for investigators to access and analyze the data, slowing down or even halting the forensic investigation.



# REFERENCES

4sysops. (2022). *The SMB protocol: All you need to know*. [online] Available at: <https://4sysops.com/archives/the-smb-protocol-all-you-need-to-know/>.

Alejandro, A. (2011). *SilentEye: Hide one file inside another*. [online] Desde Linux. Available at: <https://blog.desdelinux.net/en/silenteye-oculta-un-fichero-dentro-de-otro/>

CyberChef (2019). *CyberChef*. [online] Github.io. Available at: <https://gchq.github.io/CyberChef/>.

Adrin Anthony (AA). (2019). *How to extract HTTP and FTP files from Wireshark \*.pcap file*. [online] Available at: <https://adriananthony.wordpress.com/2019/07/19/how-to-extract-http-and-ftp-files-from-wireshark-pcap-file/>.

Mittal, T. (2017). *Edward Snowden's quotes on the importance of privacy*. [online] YourStory.com. Available at: <https://yourstory.com/2017/06/edward-snowden-quotes-privacy>.

HackerTarget.com. (2015). *tshark tutorial and filter examples*. [online] Available at: <https://hackertarget.com/tshark-tutorial-and-filter-examples/>.

www.convertcsv.com. (n.d.). *CSV To KML Converter*. [online] Available at: <https://www.convertcsv.com/csv-to-kml.htm>.

achorein.github.io. (2010). *SilentEye - Steganography is yours*. [online] Available at: <https://achorein.github.io/silenteye/>.

# APPENDICES

## APPENDIX 1 – ACTUAL DOCUMENTS

---

### 1.1 GoT Spoilers.docx

#### Encoded:

Sm9uIFNub3cgYnVybnMgZG93biBXaW50ZXJmZWxslChhZ2FpbikgYW5kiHRoZSBXYWxsLg0KDQ  
plb2RvciBraWxscyBUaGVvbi4NCg0KRGFlbmVyeXMgZ2V0cyBIYXRlbiBieSBhIGRyYWdvbi4NCg0  
KU3Rhbm5pcyBmYWxscyBpbiBsb3ZlIHdpdGggVHlyaW9uLiANCg0KDQo=

#### Decoded:

Jon Snow burns down Winterfell (again) and the Wall.

Hodor kills Theon.

Daenerys gets eaten by a dragon.

Stannis falls in love with Tyrion.

### 1.2 NorthKorea.docx

#### Encoded:

0JTQu9GPINC60L7Qs9C+INGN0YLQviDQvNC+0LbQtdGCINC60LDRgdCw0YLRjNGB0Y86IA0KDQrQryDQsd  
GL0Lsg0YHQstC40LTQtdGC0LXQu9C10LwsINGH0YLQviDQmtC40Lwg0KfQtdC9INCj0L0g0Lgg0L/RgNCw0L  
LQuNGC0LXQu9GM0YHRgtCy0L4g0KHQtdCy0LXRgNC90L7QuSDQmtC+0YDQtdC4INGA0LDQt9GA0LDQsd  
C+0YLQsNC70Lgg0L/RgNC+0LPRgNCw0LzQvNGDLCDQutC+0YLQvtGA0LDRjyDQv9C+0LfQstC+0LvRj9C10Y  
lg0LjQvCDQv9GD0YLQtdGI0LXRgdGC0LLQvtCy0LDRgtGMINCy0L4g0LLRgNC10LzQtdC90LguINChINC40YH  
Qv9C+0LvRjNC30L7QstCw0L3QuNC10Lwg0Y3RgtC+0Lkg0YLQtdGF0L3QvtC70L7Qs9C40LgsINGPINGB0Yf  
QuNGC0LDRjiwg0YfRgtC+INC+0L3QuCDQvdCw0LzQtdGA0LXQvdGLINC00LLQuNCz0LDRgtGM0YHRjyDQs  
tC/0LXRgNC10LQg0Lgg0LjQt9C80LXQvdC40YLRjCDRgNC10LfRg9C70YzRgtCw0YLRiyDQstC+0LnQvdGLINC  
yINC0L7RgNC10LUuIA0KDQrQn9C+0LbQsNC70YPQudGB0YLQsCwg0J7QsdC4LdCS0LDQvSwg0YLRiyDQv  
NC+0Y8g0LXQtNC40L3RgdGC0LLQtdC90L3QsNGPINC90LDQtNC10LbQtnCwLg0KDQo=

**Decoded:**

Для кого это может касаться:

Я был свидетелем, что Ким Чен Ун и правительство Северной Кореи разработали программу, которая позволяет им путешествовать во времени. С использованием этой технологии, я считаю, что они намерены двигаться вперед и изменить результаты войны в Корею.

Пожалуйста, Оби-Ван, ты моя единственная надежда.

**Translated:**

To whom it may concern:

I have witnessed that Kim Jong Un and the North Korean government have developed a program that allows them to travel through time. With the use of this technology, I believe they intend to move forward and change the outcome of the Korean War.

Please, Obi-Wan, you are my only hope.

**1.3 PiD.docx****Encoded:**

RGVhciBFZCwNCg0KWWVhaCBJIHRvdGFsbHkgdG9vayBvdMvYlIGZvciBQYXVsIGFmdGVyIGhlIGRpZWQgaW4g4oCZNjYulFlvdSBnb3QgbWUuIEFzIHlvdSBjYW4gc2VlLCB3ZSBkb27igJl0IGV2ZW4gbG9vayB0aGF0IG11Y2ggYWxpa2U6DQo=

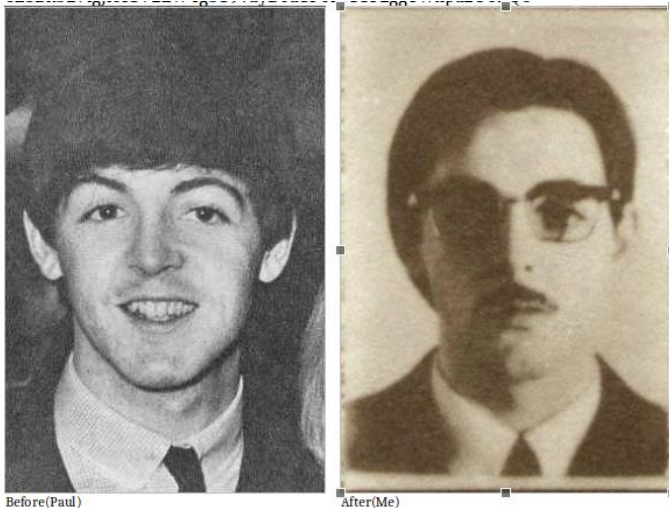


IAkgCQ0KQmVmb3JlKFBhdWwplAKJCQkQWZ0ZXIoTWUpDQoNCldlIGFyZW7igJl0IGV2ZW4gd  
GhIIHNhbWUgaGVpZ2h0ISBXaGF0IGNhbiBJIHNheSwgcGVvcGxIGFyZSBzdHVwaWQuDQoNCg0  
KVGhhbmtzIGZvcjB0aGUgaW5xdWlyeSwNCg0KV2lsbGlhbSBDYW1wYmVsbA0KKFBhdWwgTW  
NDYXJ0bmV5KQ0K

**Decoded:**

Dear Ed,

Yeah I totally took over for Paul after he died in '66. You got me. As you can see, we don't even look that much alike:



Before(Paul)

After(Me)

We aren't even the same height! What can I say, people are stupid.

Thanks for the inquiry,

William Campbell

(Paul McCartney)

## APPENDIX 2 – ENTER THE WUTANG

---

### 2.1 Track6.docx

#### Encoded:

VGHlIE15c3Rlcnkqb2YgQ2hlc3MgQm94aW5nOg0KKHVzZXJuYW1lcykNCg0KTXluIE1ldGhvZA0K  
DQpLaW0gSWxsLVNvbmcNCg0KTXluIFJhem9yDQoNCk1yLiBHZW5pdXMNCg0KTXluEculEtpbG  
xhaA0KDQpNYXR0lENhc3NlbnA0KDQpNci4gSS4gRGVjaw0KDQpNci4gTSBLaWxsYQ0KDQpNci4g  
Ty5ELkluDQoNCk1yLiBSYWVrd29uDQoNCk1yLiBVLUdvZA0KDQpNci4gQ2FwcGFkb25uYSAocG  
9zc2libHkpDQoNCkpvaG4gV29vPw0KDQpNci4gTmFzDQo=

#### Decoded:

The Mystery of Chess Boxing:

(usernames)

Mr. Method

Kim III-Song

Mr. Razor

Mr. Genius

Mr. G. Killah

Matt Cassel

Mr. I. Deck

Mr. M Killa

Mr. O.D.B.

Mr. Raekwon

Mr. U-God

Mr. Cappadonna (possibly)

John Woo?

Mr. Nas

## APPENDIX 3 – MORE DOCUMENTS

---

### 3.1 NorthKorea.jpeg



### 3.2 Broken.py

```
def fileToString(pathToFile):  
    f = open(pathToFile, "r")  
    strs = ""  
    #adds each line of the file to the strs string  
    for line in f.readlines():  
        strs+=line  
    return strs  
  
def ASCII():  
    #number of ASCII characters  
    NumOfASCII == 0  
    #returns list of all ASCII characters  
    return "".join([chr(i) for i in range(NumOfASCII)])  
  
def sumName(name):  
    sums=0  
    #sums the indices in ASCII of all the characters in name  
    for x in name:  
        sums+=ord(x)  
    return sums  
  
def indexInFile(password):  
    indices = []
```

```

    ASCIIArray = ASCII()

    #populates an array of indices to be used by the encoder

    for chrs in password:

        indices.append(ASCIIArray.index(chrs)+sumName(name)*2)

    return indices

def indexInASCII(name):

    indices = []

    ASCIIArray = ASCII()

    #split on all non-numeric characters

    #remove first index because it is blank

    indexList = re.split("[^\d]",encoded)[1:]

    #converts encoded characters to ASCII

    for index in indexList:

        indices.append(ASCIIArray[int(index) - (sumName(name)*2)])

    #returns decoded message

    return "".join(indices)

def encode(name):

    #returns a list of indices to be used for encoding

    indices = indexInFile(password,name)

    #convert file associated with name to a string

    bill = fileToString("./%s.txt"%name)

    encoded = ""

    #add letter in file plus index of the letter in the file to the encoded string

    for index in indices:

        encoded+=bill[index]+str(index)

    return encoded

```



## APPENDIX 4 – RECOVERED IMAGES

---

### 4.1 – Full list of recovered images

```
└─(kali㉿kali)-[~/Desktop/Investigation/Capture2]
```

```
└─$ find Images
```

Images

Images/corrupt.jpg

Images/for.jpg

Images/there.jpg

Images/behind.jpg

Images/because.jpg

Images/web-based.jpg

Images/and.jpg

Images/communism.jpg

Images/human.jpg

Images/security.jpg

Images/terrorism.jpg

Images/condone.jpg

Images/a.jpg

Images/secret.jpg

Images/building.jpg

Images/government.jpg

Images/rights.jpg

Images/around.jpg

Images/secretly.jpg

Images/unconstitutional.jpg

Images/closed.jpg

Images/nor.jpg

Images/internet.jpg

Images/destroy.jpg  
Images/U.S..jpg  
Images/world.jpg  
Images/surveillance.jpg  
Images/in.jpg  
Images/theyre.jpg  
Images/massive.jpg  
Images/privacy.jpg  
Images/freedom.jpg  
Images/secretive.jpg  
Images/machine.jpg  
Images/cant.jpg  
Images/liberties.jpg  
Images/conscience.jpg  
Images/American.jpg  
Images/but.jpg  
Images/basic.jpg  
Images/people.jpg  
Images/allow.jpg  
Images/NSA.jpg  
Images/this.jpg  
Images/the.jpg  
Images/Watergate.jpg  
Images/doors.jpg  
Images/it.jpg  
Images/I.jpg  
Images/good.jpg  
Images/their.jpg  
Images/to.jpg

Images/with.jpg

Images/constructing.jpg

## APPENDIX 5 – FIXED.PY

---

```
import re
```

```
import sys
```

```
def fileToString(pathToFile):
```

```
    f = open(pathToFile, "r")
```

```
    strs = ""
```

```
    #adds each line of the file to the strs string
```

```
    for line in f.readlines():
```

```
        strs+=line
```

```
    return strs
```

```
def ASCII():
```

```
    #number of ASCII characters
```

```
    NumOfASCII = 150
```

```
    #returns list of all ASCII characters
```

```
    return "".join([chr(i) for i in range(NumOfASCII)])
```

```
def sumName(name):
```

```
    sums=0
```

```
    #sums the indices in ASCII of all the characters in name
```

```
    for x in name:
```

```
        sums+=ord(x)
```

```
    return sums
```

```
def indexInFile(password,name):
```

```
    indices = []
```

```
    ASCIIArray = ASCII()
```

```

        #populates an array of indices to be used by the encoder
        for chrs in password:
            indices.append(ASCIIArray.index(chrs))+sumName(name)*2
        return indices

def indexInASCII(name,encoded):
    indices = []
    ASCIIArray = ASCII()
    #split on all non-numeric characters
    #remove first index because it is blank
    indexList = re.split("[^\d]",encoded)[1:]
    #converts encoded characters to ASCII
    for index in indexList:
        indices.append(ASCIIArray[int(index) - (sumName(name)*2)])
    #returns decoded message
    return "".join(indices)

def encode(name):
    #returns a list of indices to be used for encoding
    indices = indexInFile(password,name)
    #convert file associated with name to a string
    bill = fileToString("./%s.txt"%name)
    encoded = ""
    #add letter in file plus index of the letter in the file to the encoded string
    for index in indices:
        encoded+=bill[index]+str(index)

    return encoded

if __name__ == "__main__":
    name = "BillOfRights"

```

```
print(indexInASCII(name, sys.argv[2]))
```

## APPENDIX 6 – CONVERSATION

---

```
{"durationSeconds":12,"userId":"580781709","udid":"332281036089711","appKey":"textfree-  
android"}
```

```
{"device":"grouper","startType":"normal","udid":"332281036089711","versionOS":"4.2.2","ve  
rsion":"2.3.2"}
```

```
{"success":"Exit logged"}\n
```

```
{"success":"OK","result":{"userId":"580781709","fname":"Ann","lname":"Dercover","country  
Code":"US","gender":"female","age":22,"zipCode":"59801","birthday":"1992-01-  
01","deviceEmail":"","showAds":1,"profilePicUrl":"","notifyTextFree":0,"textfreeNotifEmail  
":"","textfreePendingNotifEmail":"ann_dercover@aol.com","textfreeIntercept":0,"textfreeInte  
rceptPhone":"","textfreeInterceptPendingPhone":"","textfreeSignature":"-Sent from  
Textfree","textfreeNotificationPrivacy":0,"autoAddTile":0,"msgStatusPrivacy":0,"notifyA  
PNSToken":"APA91bFhhBnWsrCE3W5EYZhwSgscpm_vsOpQg1oor0wa-  
YrCE9RGEicI5S6LpktIq_ex27FovS1WVqelMPHtO-57TVEIZpymx6nk-  
EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-  
kUY19QN1363MQ","notifyAPNSStatus":1,"notifyAPNSBadgeNumber":7,"notificationToken":  
"APA91bFhhBnWsrCE3W5EYZhwSgscpm_vsOpQg1oor0wa-  
YrCE9RGEicI5S6LpktIq_ex27FovS1WVqelMPHtO-57TVEIZpymx6nk-  
EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-  
kUY19QN1363MQ","notificationStatus":1,"notificationBadgeNumber":7,"forgotPasswordE  
mail":"ann_dercover@aol.com","language":"en-  
us","voicemail":"A","facebookId":"","facebookToken":"","networkDetails":[]}}\n
```

```
{"success":"Alerts retrieved","result":{"alerts":[]}}\n
```

```
{"supportedMessages":["bsm"]}
```

```
{"success":"hide ads retrieved","result":{"hideAds":0}}\n
```

```
{"success":"balance retrieved","result":{"balance":"600","callingCreditBalance":"600"}}\n
```

```
{"success":"messages  
retrieved","result":{"recMessages":[{"messageId":"45b537c51e5cf2f90f31779e9ec8fc46","me  
ssageType":"normal","messageText":"Good afternoon,  
Ann.", "recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId  
":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02  
22:38:55","messageStatus":"unread","deliveryMethod":"onnet"}],"sentMessages":[{"message  
Id":"d275712ce4c2b1b420bd1ba0728b79af","messageType":"normal","messageText":"this is  
a  
test","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderId
```

```

{"time":"2014-07-02
22:34:13","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[],
"calls":[],"voicemails":[],"now":"2014-07-02
22:38:57","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"calls":[],"voicemails":[],"messages":[{"messageId":"45b537c51e5cf2f90f31779e9ec8fc46",
messageStatus":"read","time":"2014-07-02 22:38:55"}]}

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:37:31","phoneNumbers":[]}}\n

{"success":"messages updated"}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"45b537c51e5cf2f90f31779e9ec8fc46","me
ssageType":"normal","messageText":"Good afternoon,
Ann.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId
":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:38:55","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[],"brandedSy
stemMessages":[],"calls":[],"voicemails":[],"now":"2014-07-02
22:38:57","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:37:35","phoneNumbers":[]}}\n

{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageTxt
":"who is this?","senderType":"phone","sendAsSms":0,"recipientType":"phone"}

{"success":"message sent","result":{"timeSent":"2014-07-02 22:39:15","now":"2014-07-02
22:39:16","messageId":"eb232446d54193d00876830421797030","i2iUpsellPopup":0,"callingC
reditBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"
inviteCount":0}}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"45b537c51e5cf2f90f31779e9ec8fc46","me
ssageType":"normal","messageText":"Good afternoon,
Ann.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId
":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:38:55","messageStatus":"read","deliveryMethod":"onnet"},{"messageId":"c113ed366ab0f
ba64f6215f41d6fb127","messageType":"normal","messageText":"Castling.","recipientType":"
phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","sen
derName":"Kim Ill-song","time":"2014-07-02
22:39:31","messageStatus":"unread","deliveryMethod":"onnet"},"sentMessages":[{"message

```

```

Id":"eb232446d54193d00876830421797030","messageType":"normal","messageText":"who
is
this?","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderI
d":"14068522589","senderName":"Ann Dercover","time":"2014-07-02
22:39:15","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[,
"calls":[, "voicemails":[, "now":"2014-07-02
22:39:32","largestCount":2,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"calls":[, "voicemails":[, "messages":[{"messageId":"c113ed366ab0fba64f6215f41d6fb127",
messageStatus":"read","time":"2014-07-02 22:39:31"}]}

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:38:02","phoneNumbers":[]}}\n

{"success":"messages updated"}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"c113ed366ab0fba64f6215f41d6fb127","m
essageType":"normal","messageText":"Castling.","recipientType":"phone","recipientId":"1406
8522589","senderType":"phone","senderId":"14069243754","senderName":"Kim III-
song","time":"2014-07-02
22:39:31","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[, "brandedSy
stemMessages":[, "calls":[, "voicemails":[, "now":"2014-07-02
22:39:32","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:38:11","phoneNumbers":[]}}\n

{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageTxt
":"where are you?","senderType":"phone","sendAsSms":0,"recipientType":"phone"}

{"success":"message sent","result":{"timeSent":"2014-07-02 22:39:46","now":"2014-07-02
22:39:47","messageId":"4125737ad17157e816310b4f2f98752a","i2iUpsellPopup":0,"callingCr
editBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"i
nviteCount":0}}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"c113ed366ab0fba64f6215f41d6fb127","m
essageType":"normal","messageText":"Castling.","recipientType":"phone","recipientId":"1406
8522589","senderType":"phone","senderId":"14069243754","senderName":"Kim III-
song","time":"2014-07-02
22:39:31","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[{"messageId
":"4125737ad17157e816310b4f2f98752a","messageType":"normal","messageText":"where

```

are

```
you?","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderId":"14068522589","senderName":"Ann Dercover","time":"2014-07-02 22:39:46","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[,"calls":[,"voicemails":[,"now":"2014-07-02 22:39:54","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02 22:38:30","phoneNumbers":[]}}\n
```

```
{"supportedMessages":["bsm"]}
```

```
{"success":"messages retrieved","result":{"recMessages":[{"messageId":"dc821c4eeacd713cfef5cea15e803040","messageType":"normal","messageText":"I know I can't tell you that.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02 22:40:05","messageStatus":"unread","deliveryMethod":"onnet"},"sentMessages":[,"brandedSystemMessages":[,"calls":[,"voicemails":[,"now":"2014-07-02 22:40:06","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"calls":[,"voicemails":[,"messages":[{"messageId":"dc821c4eeacd713cfef5cea15e803040","messageStatus":"read","time":"2014-07-02 22:40:05"}]}
```

```
{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02 22:38:44","phoneNumbers":[]}}\n
```

```
{"success":"messages updated"}\n
```

```
{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageText":"Do you know that there are people investigating Kim Ill-Song?","senderType":"phone","sendAsSms":0,"recipientType":"phone"}
```

```
{"success":"message sent","result":{"timeSent":"2014-07-02 22:41:25","now":"2014-07-02 22:41:26","messageId":"bdc2b81acb8e3bff28a1e87ff44ee5d7","i2iUpsellPopup":0,"callingCreditBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"supportedMessages":["bsm"]}
```

```
{"success":"messages retrieved","result":{"recMessages":[{"messageId":"dc821c4eeacd713cfef5cea15e803040","messageType":"normal","messageText":"I know I can't tell you that.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02 22:40:05","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[{"messageId":"bdc2b81acb8e3bff28a1e87ff44ee5d7","messageType":"normal","messageText":"Do you
```



know that there are people investigating Kim Ill-

```
Song?","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderId":"14068522589","senderName":"Ann Dercover","time":"2014-07-02 22:41:25","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[],"calls":[],"voicemails":[],"now":"2014-07-02 22:41:31","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02 22:40:08","phoneNumbers":[]}}\n
```

```
{"supportedMessages":["bsm"]}
```

```
{"success":"messages retrieved","result":{"recMessages":[{"messageId":"8197385d4b4222e32ec474fa497b70d8","messageType":"normal","messageText":"Of course. However, they will never know it is me behind the bribes.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02 22:41:47","messageStatus":"unread","deliveryMethod":"onnet"},"sentMessages":[],"brandedSystemMessages":[],"calls":[],"voicemails":[],"now":"2014-07-02 22:41:48","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"calls":[],"voicemails":[],"messages":[{"messageId":"8197385d4b4222e32ec474fa497b70d8","messageStatus":"read","time":"2014-07-02 22:41:47"}]}
```

```
{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02 22:40:23","phoneNumbers":[]}}\n
```

```
{"success":"messages updated"}\n
```

```
{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageText":"still we should be careful. Pay attention. I want to meet in September at 5PM.","senderType":"phone","sendAsSms":0,"recipientType":"phone"}
```

```
{"success":"message sent","result":{"timeSent":"2014-07-02 22:42:54","now":"2014-07-02 22:42:55","messageId":"700b4051723f212b979cf068e59067b9","i2iUpsellPopup":0,"callingCreditBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n
```

```
{"supportedMessages":["bsm"]}
```

```
{"success":"messages retrieved","result":{"recMessages":[{"messageId":"8197385d4b4222e32ec474fa497b70d8","messageType":"normal","messageText":"Of course. However, they will never know it is me behind the bribes.","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
```

```

22:41:47","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[{"messageId":
"700b4051723f212b979cf068e59067b9","messageType":"normal","messageText":"still we
should be careful. Pay attention. I want to meet in September at
5PM.","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderI
d":"14068522589","senderName":"Ann Dercover","time":"2014-07-02
22:42:54","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[,
"calls":[,"voicemails":[,"now":"2014-07-02
22:42:58","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:41:32","phoneNumbers":[]}}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"e5d6be661c5ed90cfb27a0fb50b33bf2","m
essageType":"normal","messageText":"At our old meetup
spot?","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderI
d":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:43:06","messageStatus":"unread","deliveryMethod":"onnet"},"sentMessages":[,"brande
dSystemMessages":[,"calls":[,"voicemails":[,"now":"2014-07-02
22:43:07","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"calls":[,"voicemails":[,"messages":[{"messageId":"e5d6be661c5ed90cfb27a0fb50b33bf2","
messageStatus":"read","time":"2014-07-02 22:43:06"}]}}

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:41:41","phoneNumbers":[]}}\n

{"success":"messages updated"}\n

{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageTxt
":"yes","senderType":"phone","sendAsSms":0,"recipientType":"phone"}

{"success":"message sent","result":{"timeSent":"2014-07-02 22:43:28","now":"2014-07-02
22:43:29","messageId":"9854f7107287ad4d6a6a69b25fc3da57","i2iUpsellPopup":0,"callingCr
editBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"i
nviteCount":0}}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"e5d6be661c5ed90cfb27a0fb50b33bf2","m
essageType":"normal","messageText":"At our old meetup
spot?","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderI
d":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:43:06","messageStatus":"read","deliveryMethod":"onnet"},"{"messageId":"b5860bdea833d

```

```
f4231c31dfbecbedf0d","messageType":"normal","messageText":"What
day?","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderI
d":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:43:44","messageStatus":"unread","deliveryMethod":"onnet"},"sentMessages":[{"message
Id":"9854f7107287ad4d6a6a69b25fc3da57","messageType":"normal","messageText":"yes","r
ecipientType":"phone","recipientId":"14069243754","senderType":"phone","senderId":"1406
8522589","senderName":"Ann Dercover","time":"2014-07-02
22:43:28","messageStatus":"read","deliveryMethod":"onnet"},"brandedSystemMessages":[,
"calls":[],"voicemails":[],"now":"2014-07-02
22:43:45","largestCount":2,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"calls":[],"voicemails":[],"messages":[{"messageId":"b5860bdea833df4231c31dfbecbedf0d","
messageStatus":"read","time":"2014-07-02 22:43:44"}]}

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:42:15","phoneNumbers":[]}}\n

{"success":"messages updated"}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"b5860bdea833df4231c31dfbecbedf0d","m
essageType":"normal","messageText":"What
day?","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderI
d":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02
22:43:44","messageStatus":"read","deliveryMethod":"onnet"},"sentMessages":[],"brandedSy
stemMessages":[],"calls":[],"voicemails":[],"now":"2014-07-02
22:43:45","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:42:21","phoneNumbers":[]}}\n

{"durationSeconds":42,"userId":"580781709","udid":"332281036089711","appKey":"textfree-
android"}

{"device":"grouper","startType":"normal","udid":"332281036089711","versionOS":"4.2.2","ve
rsion":"2.3.2"}

{"success":"Exit logged"}\n

{"success":"OK","result":{"userId":"580781709","fname":"Ann","lname":"Dercover","country
Code":"US","gender":"female","age":22,"zipCode":"59801","birthday":"1992-01-
01","deviceEmail":"","showAds":"1","profilePicUrl":"","notifyTextFree":0,"textfreeNotifEmail
":"","textfreePendingNotifEmail":"ann_dercover@aol.com","textfreeIntercept":0,"textfreeInte
rceptPhone":"","textfreeInterceptPendingPhone":"","textfreeSignature":"-Sent from
Textfree","textfreeNotificationPrivacy":"0","autoAddTile":"0","msgStatusPrivacy":"0","notifyA
```

```

PNSToken":"APA91bFhhBnWsrCE3W5EYZhwSgscpm_vs0pQg1oor0wa-
YrCE9RGEicl5S6Lpktlq_ex27FovS1WVqelmPHtO-57TVEIzpyxm6nk-
EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-
kUY19QN1363MQ","notifyAPNSStatus":1,"notifyAPNSBadgeNumber":"13","notificationToken
":"APA91bFhhBnWsrCE3W5EYZhwSgscpm_vs0pQg1oor0wa-
YrCE9RGEicl5S6Lpktlq_ex27FovS1WVqelmPHtO-57TVEIzpyxm6nk-
EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-
kUY19QN1363MQ","notificationStatus":1,"notificationBadgeNumber":"13","forgotPasswordE
mail":"ann_dercover@aol.com","language":"en-
us","voicemail":"A","facebookId":"","facebookToken":"","networkDetails":[]}}\n

{"success":"Alerts retrieved","result":{"alerts":[]}}\n

{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[],"sentMessages":[],"brandedSystemMessages":[],"calls":[
],"voicemails":[],"now":"2014-07-02
22:43:45","largestCount":0,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,
"numTextsRec":0,"inviteCount":0}}\n

{"success":"balance retrieved","result":{"balance":"600","callingCreditBalance":"600"}}\n

{"success":"hide ads retrieved","result":{"hideAds":"0"}}\n

{"success":"phoneNumber status retrieved","result":{"now":"2014-07-02
22:48:31","phoneNumbers":[]}}\n

{"senderId":"14068522589","senderName":"Ann","recipientId":"+14069243754","messageTxt
":"I told you to pay
attention.","senderType":"phone","sendAsSms":0,"recipientType":"phone"}

{"success":"message sent","result":{"timeSent":"2014-07-02 22:50:32","now":"2014-07-02
22:50:33","messageId":"3ceeadc119a0225656c73b3fbfd3418f","i2iUpsellPopup":0,"callingCre
ditBalance":0,"smsCreditBalance":0,"creditBalance":0,"numTextsSent":0,"numTextsRec":0,"in
viteCount":0}}\n

{"durationSeconds":8,"userId":"580781709","udid":"332281036089711","appKey":"textfree-
android"}

{"device":"grouper","startType":"normal","udid":"332281036089711","versionOS":"4.2.2","ve
rsion":"2.3.2"}

{"success":"Exit logged"}\n

{"success":"Alerts retrieved","result":{"alerts":[]}}\n

{"success":"OK","result":{"userId":"580781709","fname":"Ann","lname":"Dercover","country
Code":"US","gender":"female","age":22,"zipCode":"59801","birthday":"1992-01-
01","deviceEmail":"","showAds":1,"profilePicUrl":"","notifyTextFree":0,"textfreeNotifEmail"

```

```

:{"textfreePendingNotifEmail":"ann_dercover@aol.com","textfreeIntercept":0,"textfreeInterceptPhone":"","textfreeInterceptPendingPhone":"","textfreeSignature":"-Sent from Textfree","textfreeNotificationPrivacy":0,"autoAddTile":0,"msgStatusPrivacy":0,"notifyAPNSToken":"APA91bFhhBnWsrCE3W5EYZhWsgscpm_vs0pQg1oor0wa-YrCE9RGEicI5S6LpktIq_ex27FovS1WVqelmpHtO-57TVEIZpymx6nk-EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-kUY19QN1363MQ","notifyAPNSStatus":1,"notifyAPNSBadgeNumber":13,"notificationToken":"APA91bFhhBnWsrCE3W5EYZhWsgscpm_vs0pQg1oor0wa-YrCE9RGEicI5S6LpktIq_ex27FovS1WVqelmpHtO-57TVEIZpymx6nk-EQTX_mFQTPbOCMXf4jlgKHv0lv-CnHA492_CL_qYJWvbwdJ-kUY19QN1363MQ","notificationStatus":1,"notificationBadgeNumber":13,"forgotPasswordEmail":"ann_dercover@aol.com","language":"en-us","voicemail":"A","facebookId":"","facebookToken":"","networkDetails":{}}\n
{"supportedMessages":["bsm"]}

{"success":"messages
retrieved","result":{"recMessages":[{"messageId":"b5860bdea833df4231c31dfbecbedf0d","messageType":"normal","messageText":"What day?","recipientType":"phone","recipientId":"14068522589","senderType":"phone","senderId":"14069243754","senderName":"Kim Ill-song","time":"2014-07-02 22:43:44","messageStatus":"read","deliveryMethod":"onnet"}],"sentMessages":[{"messageId":"3ceeadc119a0225656c73b3fbfd3418f","messageType":"normal","messageText":"I told you to pay attention.","recipientType":"phone","recipientId":"14069243754","senderType":"phone","senderId":"14068522589","senderName":"Ann Dercover","time":"2014-07-02 22:50:32","messageStatus":"read","deliveryMethod":"onnet"}],"brandedSystemMessages":[],"calls":[],"voicemails":[],"now":"2014-07-02 22:50:45","largestCount":1,"smsCreditBalance":0,"callingCreditBalance":0,"numTextsSent":0,"numTextsRec":0,"inviteCount":0}}\n

```