

## Unit 2: Network Forensics Report

CMP416-Ethical Hacking 4

BSc Ethical Hacking Year 4

2021/22

## Contents



# Abertay University

|  |    |
|--|----|
| .....  | 1  |
| 1 - Investigation of Capture1.pcap:.....                                   | 4  |
| 1.1 - Verify integrity of evidence and make copies of capture files: ..... | 5  |
| 1.2 – Statistical Flow Analysis: .....                                     | 5  |
| 1.3 - Wireshark initial analysis: .....                                    | 6  |
| 1.4 - Searching HTTP traffic for evidence of bribery:.....                 | 6  |
| 1.5 - Searching SMB traffic for downloaded files: .....                    | 7  |
| 2 – Investigation of Capture 2.pcap: .....                                 | 8  |
| 2.1 – Filter Wireshark for IRC traffic: .....                              | 8  |
| 2.2 – Search for encrypted conversation:.....                              | 9  |
| 2.3 – Decrypted conversation:.....   | 9  |
| 2.4: Decrypted conversation with Razor:.....                               | 9  |
| .....  | 10 |
| 2.5: Decrypted conversation with Method:.....                              | 10 |
| 2.6: Decrypted conversation with Killah: .....                             | 10 |
| .....  | 11 |
| 2.7: Decrypted conversation with Raekwon:.....                             | 11 |
| 3 – Investigation of Capture 3.pcap: .....                                 | 11 |
| 3.1 Filtering Wireshark by FTP traffic:.....                               | 11 |
| .....  | 12 |
| .....  | 12 |
| 3.2 Save ftp traffic as raw data: .....                                    | 12 |
| 3.3 Reconstructing image using cat: .....                                  | 12 |
| 3.4 Filtering traffic by mime multipart:.....                              | 13 |
| 3.5 Export the packet bytes: .....   | 13 |
| .....  | 13 |
| 3.6 Using cat to reconstruct the image:.....                               | 13 |
| 4 - Investigation of Capture 4.pcap:.....                                  | 14 |
| 4.1: Finding conversation in Wireshark .....                               | 14 |
| 4.2 find information in SMS conversation .....                             | 15 |
| 4.3 find geolocation data .....  | 16 |
| 4.4 find location in google earth .....                                    | 17 |

|  |    |
|--|----|
| 5 Critical Analysis & Evaluation:..... | 17 |
| 5.1 Capture 1:.....                    | 17 |
| 5.2 Capture 2:.....                    | 18 |
| 5.3 Capture 3:.....                    | 18 |
| 5.4 Capture 4:.....                    | 19 |
| Appendix .....                         | 20 |
| Appendix 1: .....                      | 20 |
| Appendix 2: .....                      | 21 |
| Appendix 3: .....                      | 22 |
| Appendix 4: .....                      | 22 |
| Appendix 5: .....                      | 25 |
| .....                                  | 25 |
| Appendix 6: .....                      | 25 |
| Appendix 7: .....                      | 26 |
| Appendix 8: .....                      | 27 |
| Appendix 9: .....                      | 27 |
| Appendix 10: .....                     | 27 |
| Appendix 11: .....                     | 28 |
| Appendix 12: .....                     | 28 |
| Appendix 13: .....                     | 29 |
| Appendix 14: .....                     | 33 |

## Introduction

Network forensics has become an essential part of combating cyber-crime and corruption in recent years. Network forensics usually involves an investigator analysing packets of data transmitted over a network in real time or with packet captures that they can analyse at a later point in time. Nowadays, network forensic investigators have a wide array of tools at their disposal such as Wireshark. These tools if successfully implemented and used, can efficiently locate, download, and save evidence from captured packets to convict a suspect of a cybercrime. When an investigator conducts a forensic investigation it is vitally important that they handle the evidence with care and stick to a strict methodology in order to keep the investigation as forensically sound as possible. If a methodology is not followed, then there is a high risk of compromising the data packets contained within the packet capture thus harming the validity of a forensic investigation.

Tools used:

- Wireshark
- Tshark
- CyberChef
- Grep
- Python
- SiLk
- Yaf
- Kali Linux
- Unzip
- Cat
- Google Earth
- Excel
- <https://www.convertcsv.com>
- crackstation

## 1 - Investigation of Capture1.pcap:

### 1.1 - Verify integrity of evidence and make copies of capture files:

Before each investigating each case in the forensic investigation, the integrity of the captured evidence was verified. This was done using the commands **md5sum capture1.pcap** and **sha256sum capture1.pcap**. Figure 1 and Figure 2 below show the command and results for case 1.

```
(kali@kali)~[/Desktop]
$ md5sum Capture1.pcap
bae9aade7f29f88494a985cea8ff350f Capture1.pcap
```

Figure 1: md5sum for capture 1

```
(kali@kali)~[/Desktop]
$ sha256sum Capture1.pcap
52a9a89500b9aeb22edb0845c3e394b4c42f15b60d9ac3cb6cf77b6a4f3ab1e1 Capture1.pcap
```

Figure 2: sha256sum for capture 1

After verifying the integrity of the evidence, the next step was to make copies of the capture files as this is the best practice in a forensic investigation. Searching or modifying the original capture files should be avoided because the integrity of the evidence can be compromised. After making copies of the capture files, the checksum hashes were verified again as seen in Figure 4.

|                      |          |                       |            |
|----------------------|----------|-----------------------|------------|
| Capture1.pcap        | 20.9 MiB | Packet Capture (PCAP) | 07/11/2014 |
| Capture1(copy1).pcap | 20.9 MiB | Packet Capture (PCAP) | 07/11/2014 |

Figure 3

```
(kali@kali)~[/Desktop/1]
$ sha256sum -t Capture1(copy 1).pcap
52a9a89500b9aeb22edb0845c3e394b4c42f15b60d9ac3cb6cf77b6a4f3ab1e1 Capture1(copy 1).pcap
```

Figure 4

### 1.2 – Statistical Flow Analysis:

To get an understanding of the captured packets, statistical flow analysis had to be used. Statistical flow analysis was done with the tool SiLK. Before statistical analysis can be done the pcap must file be converted to IPFIX format using YAF. This was done with the command **yaf -in Capture1.pcap -out Capture1.yaf**. The converted file then had to be converted to IPFIX format with the command **rwipfix2silk Capture1.yaf -silk-output=capture1.rw**. The command **rwstats capture1.rw --fields=1,2 --values=packets -count=20** was used next. The results showed significant traffic originating from the IP addresses: 172.29.1.23, 64.12.132.55, 93.184.215.248, and 172.29.1.20. The results also showed that 22.2% of packets were exchanged between the 172.29.1.23 and 64.12.132.55. The results of the command can be found in Figure 5 below. The command **rwstats capture1.rw -fields=3 -values=packets -count=10** was the last command used in the statistical analysis stage. The results

showed 33% of packets used port 80 (HTTP) and it also showed that 3% of packets in the capture used port 445 (SMB). The results can be seen in Figure 6 below.

```
(kali@kali)~[~/Desktop/CMP416PCAP/Investigation1]
$ rwsstats capture1.rw --fields=1,2 --values=packets --count=20
INPUT: 1368 Records for 322 Bins and 30295 Total Packets
OUTPUT: Top 20 Bins by Packets
```

| sIP             | dIP             | Packets | %Packets  | cumul_%   |
|-----------------|-----------------|---------|-----------|-----------|
| 172.29.1.23     | 64.12.132.55    | 6735    | 22.231391 | 22.231391 |
| 64.12.132.55    | 172.29.1.23     | 3636    | 12.001981 | 34.233372 |
| 93.184.215.248  | 172.29.1.20     | 2684    | 8.859548  | 43.092920 |
| 172.29.1.20     | 93.184.215.248  | 1571    | 5.185674  | 48.278594 |
| 172.29.1.23     | 172.29.1.20     | 1362    | 4.495791  | 52.774385 |
| 172.29.1.20     | 172.29.1.23     | 965     | 3.185344  | 55.959729 |
| 184.28.16.25    | 172.29.1.20     | 485     | 1.600924  | 57.560654 |
| 173.194.79.103  | 172.29.1.20     | 463     | 1.528305  | 59.088959 |
| 172.29.1.20     | 184.28.16.25    | 388     | 1.280739  | 60.369698 |
| 172.29.1.23     | 74.125.239.60   | 324     | 1.069483  | 61.439181 |
| 172.29.1.20     | 173.194.79.103  | 312     | 1.029873  | 62.469054 |
| 74.125.239.60   | 172.29.1.23     | 285     | 0.940749  | 63.409804 |
| 23.216.11.91    | 172.29.1.20     | 269     | 0.887935  | 64.297739 |
| 171.161.199.100 | 172.29.1.20     | 262     | 0.864829  | 65.162568 |
| 172.29.1.23     | 74.125.239.50   | 241     | 0.795511  | 65.958079 |
| 172.29.1.20     | 205.188.16.197  | 233     | 0.769104  | 66.727183 |
| 74.125.239.50   | 172.29.1.23     | 222     | 0.732794  | 67.459977 |
| 172.29.1.23     | 69.172.216.55   | 215     | 0.709688  | 68.169665 |
| 172.29.1.20     | 23.216.11.91    | 214     | 0.706387  | 68.876052 |
| 172.29.1.20     | 171.161.199.100 | 205     | 0.676679  | 69.552731 |

Figure 5: Filter top-20 sIP/ dIP pairs based on the amount of packets

```
(kali@kali)~[~/Desktop/CMP416PCAP/Investigation1]
$ rwsstats capture1.rw --fields=3 --values=packets --count=10
INPUT: 1368 Records for 686 Bins and 30295 Total Packets
OUTPUT: Top 10 Bins by Packets
```

| sPort | Packets | %Packets  | cumul_%   |
|-------|---------|-----------|-----------|
| 80    | 10051   | 33.177092 | 33.177092 |
| 50180 | 6593    | 21.762667 | 54.939759 |
| 443   | 2395    | 7.905595  | 62.845354 |
| 50291 | 1336    | 4.409969  | 67.255323 |
| 445   | 942     | 3.109424  | 70.364747 |
| 1784  | 308     | 1.016669  | 71.381416 |
| 1315  | 204     | 0.673378  | 72.054795 |
| 50039 | 182     | 0.600759  | 72.655554 |
| 1769  | 181     | 0.597458  | 73.253012 |
| 1696  | 147     | 0.485229  | 73.738241 |

Figure 6: port 445 (SMB) traffic detected

### 1.3 - Wireshark initial analysis:

The first step was to verify if the results from the statistical analysis stage were accurate by filtering the Capture1.pcap for evidence of HTTP and SMB traffic. The results can be found in appendix 1. The results confirmed the existence of HTTP and SMB traffic, especially on the IP address 172.29.1.23.

### 1.4 - Searching HTTP traffic for evidence of bribery:

The next step done was to search HTTP traffic for any packets that might reveal incriminating evidence. This was done by using Wireshark's 'export objects' feature, which makes it easier to filter/ find packets based on their byte size and file type. The export objects feature can be found under **File > Export Objects > HTTP** as seen in Figure 7 below. After this was done, a from the sIP 172.29.1.23 was found to contain 8654kb of data, and after using the feature 'follow tcpstream' a conversation between the aliases [snowedinedward@aol.com](mailto:snowedinedward@aol.com) and [wikiofleaks@aol.com](mailto:wikiofleaks@aol.com) was found. The packet also revealed evidence of a .zip and .pcap file transfer between 172.29.1.23 and 64.12.132.55 which can be seen Figure 8. Upon further inspection the docs.pcap file didn't contain anything incriminating. The steps taken and the full conversation can be found in appendix 2.

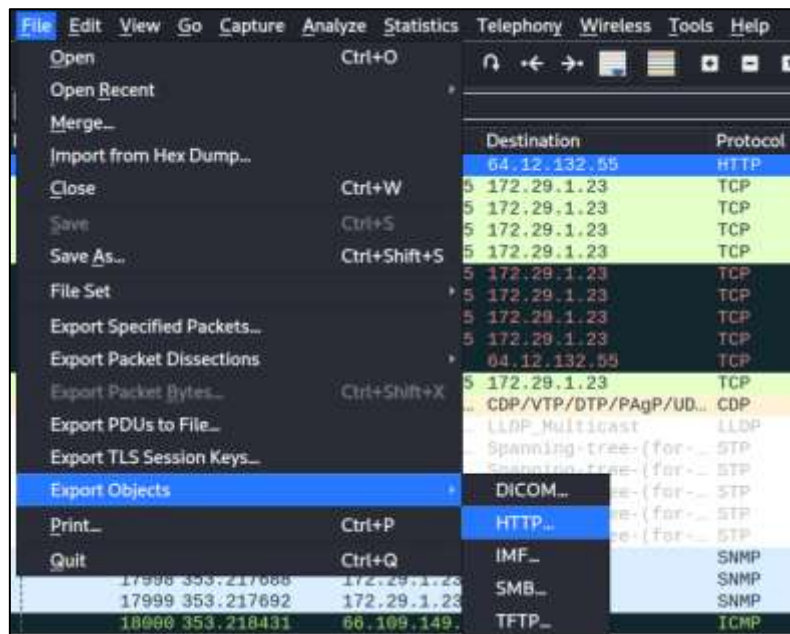


Figure 7: find HTTP objects

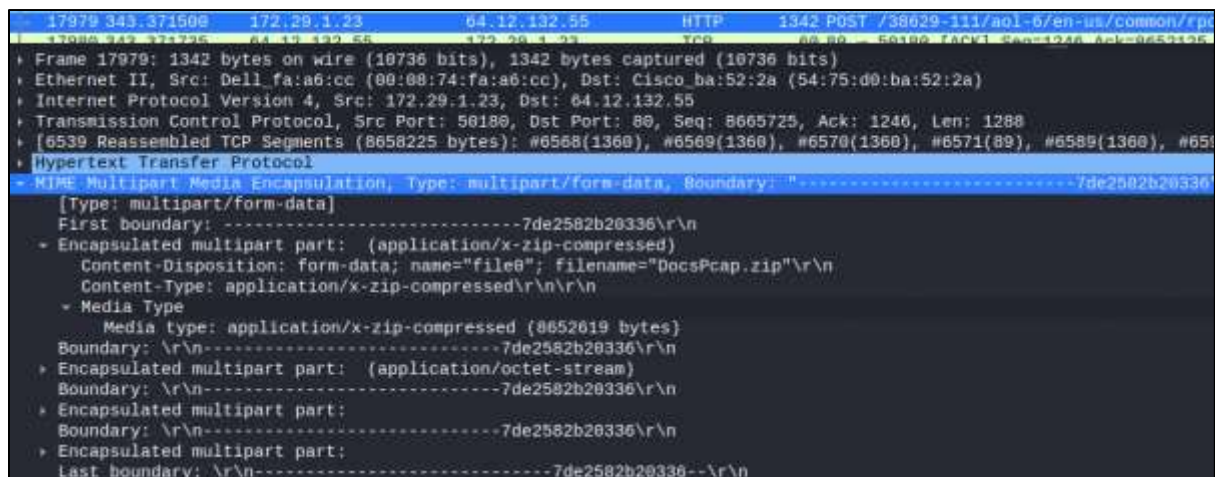


Figure 8: zip file

### 1.5 - Searching SMB traffic for downloaded files:

According to SiLK, there was SMB traffic in capture1.pcap so the next step was to repeat the steps used previously but for SMB traffic instead of HTTP traffic. This was done using the following steps: **File > Export Objects > SMB** which can be seen in **Error! Reference source not found.** below. After this was done, a zip file called 'documents.zip' was found. This zip file was sent by the SIP 172.29.1.23 to the DIP 172.29.1.20 as seen in appendix 3. Downloading and unzipping the zip file revealed five files called **Actual Documents, Chess Boxing, Enter the Wu-Tang Clan, and More Documents** which can be seen in Figure 10 below. The files contained files such as images of North Korean flags, spoilers from Game of Thrones encoded with base64, and aliases/ names inside track6.docx. These names were encoded with base64 so they may be related later in the investigation. In addition, using the unzip command on North Korea.jpeg revealed a hidden python script called broken.py. The full results of this stage can be found in appendix 4 including the names found in the track6 document. Overall, after much investigation, no evidence of bribery was found in capture 1.



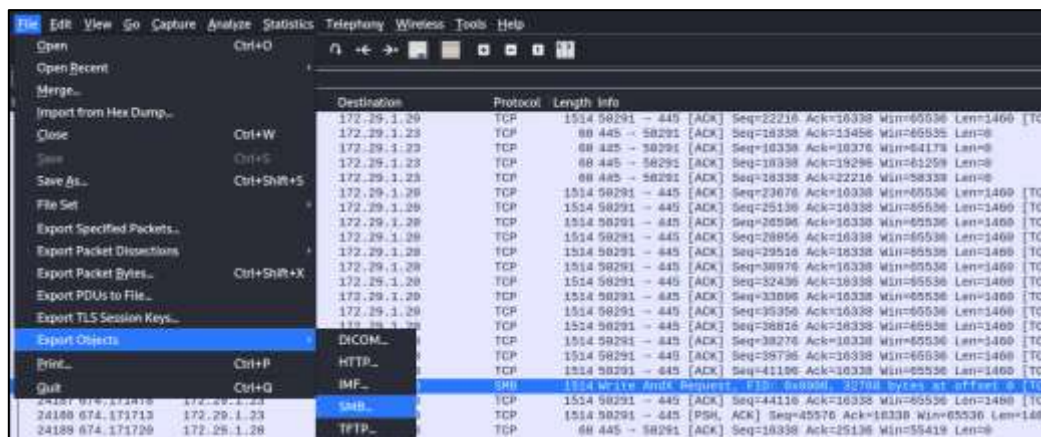


Figure 9: export SMB packets

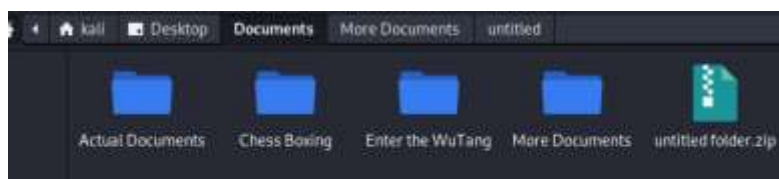


Figure 10: downloaded files

## 2 – Investigation of Capture 2.pcap:

### 2.1 – Filter Wireshark for IRC traffic:

According to intelligence, encrypted traffic between suspected corrupt officials was detected using IRC monitors. This means statistical analysis was not necessary, because the protocol in use (ICR) was already known. Therefore, the first step was to verify if the results from the IRC monitors were correct, which was done by filtering traffic in Wireshark by IRC as seen in **Error! Reference source not found. b** below. The results shown in appendix 5 confirm the findings by the IRC monitors, however they also show the existence of known aliases such as Genius, Raekwon, and Killah.

| No. | Time       | Source        | Destination   | Protocol | Length | Info               |
|-----|------------|---------------|---------------|----------|--------|--------------------|
| 16  | 13.268894  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 17  | 13.454498  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 22  | 15.058733  | 172.29.1.17   | 185.38.166.35 | IRC      | 276    | Request (PRIVMSG)  |
| 28  | 22.548315  | 172.29.1.17   | 185.38.166.35 | IRC      | 74     | Request (PING)     |
| 30  | 22.733678  | 185.38.166.35 | 172.29.1.17   | IRC      | 114    | Response (PONG)    |
| 35  | 28.205806  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 37  | 28.451405  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 43  | 34.232842  | 185.38.166.35 | 172.29.1.17   | IRC      | 220    | Response (PRIVMSG) |
| 53  | 43.262962  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 54  | 43.448567  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 65  | 52.573610  | 172.29.1.17   | 185.38.166.35 | IRC      | 74     | Request (PING)     |
| 74  | 58.260120  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 77  | 58.445729  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 95  | 70.677820  | 185.38.166.35 | 172.29.1.17   | IRC      | 448    | Response (PRIVMSG) |
| 100 | 73.257281  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 101 | 73.442885  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 108 | 82.567928  | 172.29.1.17   | 185.38.166.35 | IRC      | 74     | Request (PING)     |
| 109 | 82.753533  | 185.38.166.35 | 172.29.1.17   | IRC      | 114    | Response (PONG)    |
| 118 | 88.254188  | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 120 | 88.439793  | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 122 | 88.671858  | 172.29.1.17   | 185.38.166.35 | IRC      | 340    | Request (PRIVMSG)  |
| 134 | 103.251348 | 172.29.1.17   | 185.38.166.35 | IRC      | 96     | Request (ISON)     |
| 136 | 103.438865 | 185.38.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 148 | 112.311445 | 185.38.166.35 | 172.29.1.17   | IRC      | 204    | Response (PRIVMSG) |
| 151 | 112.577735 | 172.29.1.17   | 185.38.166.35 | IRC      | 74     | Request (PING)     |

Figure 11



## 2.2 – Search for encrypted conversation:

After finding the IRC traffic, the next step was to follow the TCP stream in order to view the whole encrypted conversation in an easier to read format. This was done right clicking on the first packet, hovering over follow, and then clicking TCP which can be seen in Figure 12 below. Once this was done, the entire encrypted conversation was displayed in ASCII format which can then be decrypted using online tools like Cyberchef. The results can be seen in appendix 6.

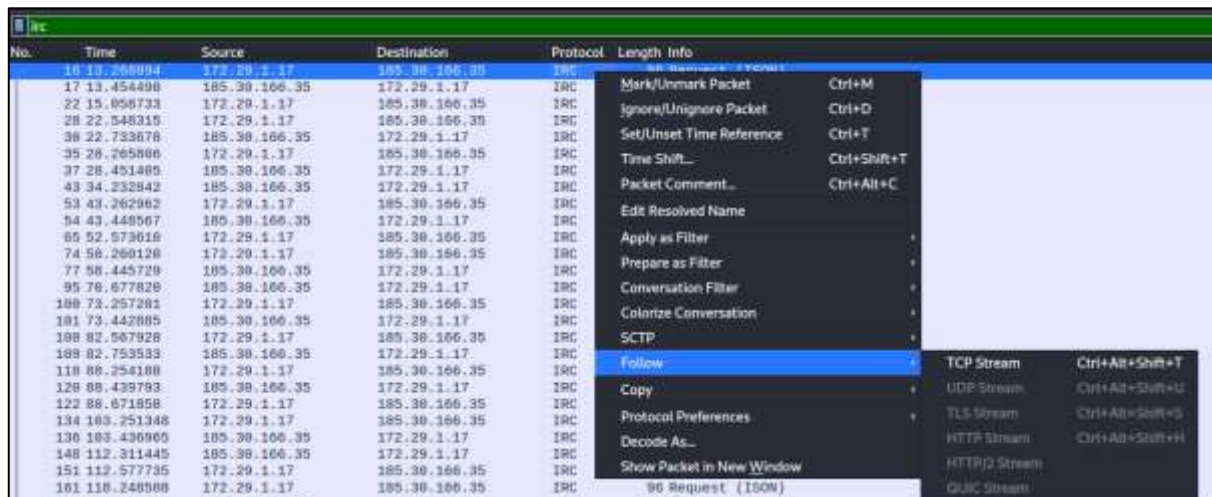


Figure 12: follow TCP stream

## 2.3 – Decrypted conversation:

The conversation was then decrypted using Cyberchef and after doing this, evidence of bribery was discovered between foreign officials as seen in Figure 13 below. It was found that a user going by the name “Ill Song” was trying to bribe foreign officials with large sums of money. The corrupt officials were encrypting their conversations with Base 64, Base 32, Octal, MD5, and Hexadecimal. According to the decrypted conversation, Ill Song was attempting to bribe officials going by the aliases Killah, Raekwon, Method and Razer.

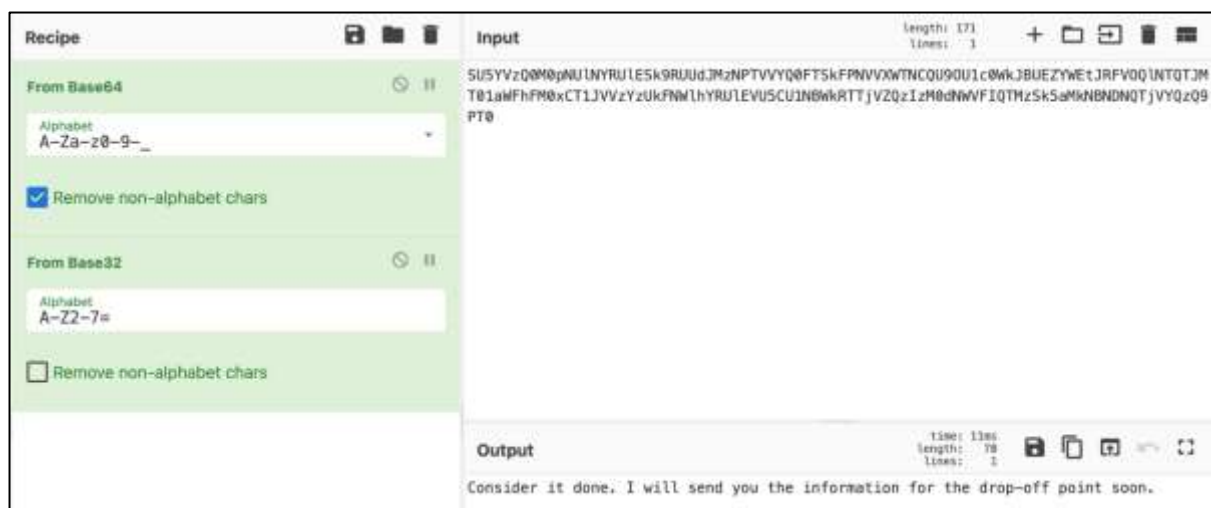


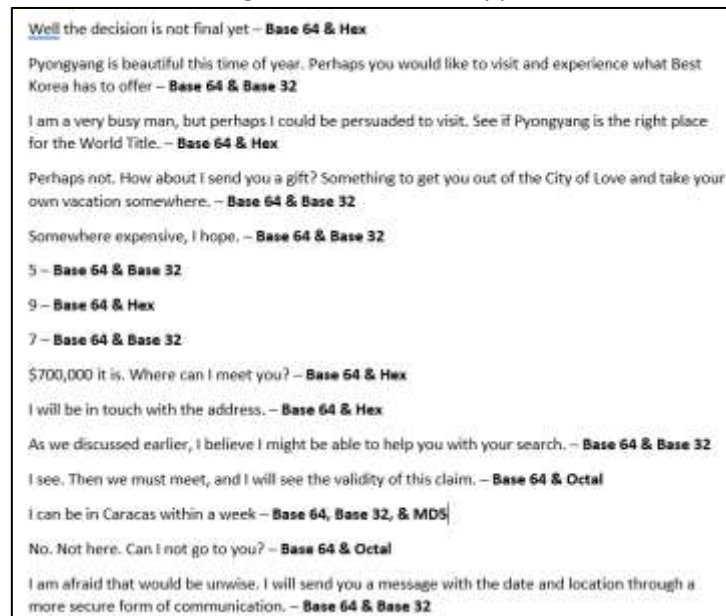
Figure 13: cyberchef decrypted conversation

## 2.4: Decrypted conversation with Razor:

In the first conversation Ill Song was trying to convince Razor to move the chess boxing world title to Pyongyang by bribing him with an expensive gift. Razor initially said he was busy but Ill Song and Razor then argued over large sums of money before finally settling on \$700,000. They could not settle on a

meeting point in the conversation, they initially talked about meeting in Caracas Venezuela. However, Ill Song then said he would send a data and location with a more secure form of communication later. This means Ill Song and Razor are guilty of bribery.

Ill Song and Razor encrypted the entire conversation with Base 64, Base 32, Octal, and Hexadecimal. The full conversation can be found in Figure 14 below and appendix 7.



Well the decision is not final yet – **Base 64 & Hex**

Pyongyang is beautiful this time of year. Perhaps you would like to visit and experience what Best Korea has to offer – **Base 64 & Base 32**

I am a very busy man, but perhaps I could be persuaded to visit. See if Pyongyang is the right place for the World Title. – **Base 64 & Hex**

Perhaps not. How about I send you a gift? Something to get you out of the City of Love and take your own vacation somewhere. – **Base 64 & Base 32**

Somewhere expensive, I hope. – **Base 64 & Base 32**

5 – **Base 64 & Base 32**

9 – **Base 64 & Hex**

7 – **Base 64 & Base 32**

\$700,000 it is. Where can I meet you? – **Base 64 & Hex**

I will be in touch with the address. – **Base 64 & Hex**

As we discussed earlier, I believe I might be able to help you with your search. – **Base 64 & Base 32**

I see. Then we must meet, and I will see the validity of this claim. – **Base 64 & Octal**

I can be in Caracas within a week – **Base 64, Base 32, & MD5**

No. Not here. Can I not go to you? – **Base 64 & Octal**

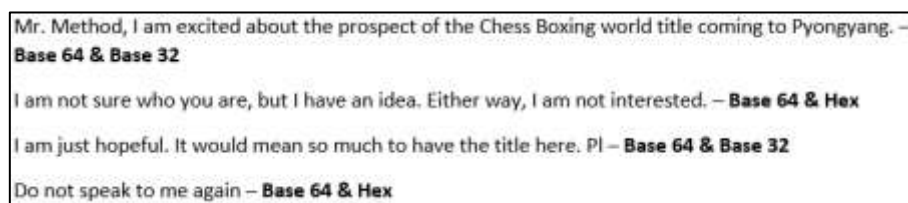
I am afraid that would be unwise. I will send you a message with the date and location through a more secure form of communication. – **Base 64 & Base 32**

Figure 14

## 2.5: Decrypted conversation with Method:

The second conversation, Ill Song again wanted to move the Chess Boxing World title to Pyongyang by bribing Method. However, unlike Razor, Method rejected Ill Song's offer and told him to never contact him again. Therefore, this means Method is not guilty of bribery.

The conversation between Method and Ill Song was encrypted with Base 64, Base 32, and Hexadecimal. The full conversation can be seen in Figure 15 below and appendix 7. The method of using Cyberchef to decrypt the conversation manually had its downsides. One disadvantage is that the process is very slow and arduous. The second is that it was unable to decrypt all of the conversation. This stage can be improved by using scripts to speed up the decryption process and using more than one tool.



Mr. Method, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang. – **Base 64 & Base 32**

I am not sure who you are, but I have an idea. Either way, I am not interested. – **Base 64 & Hex**

I am just hopeful. It would mean so much to have the title here. Pl – **Base 64 & Base 32**

Do not speak to me again – **Base 64 & Hex**

Figure 15

## 2.6: Decrypted conversation with Killah:

In the third conversation, Ill Song tries to bribe Killah, who resides in Qatar, into moving the chess boxing world title to Korea. Killah tells Ill Song to wait to see how the bid turns out. Upon Ill Song asking Killah if there is anything he could do to make his decision earlier, Killah refuses and tells Ill Song he, and Qatar would never accept bribes. Therefore, Killah and Qatar is not guilty of bribes.

The conversation between Killah and Ill Song was encrypted with Base 64, Base 32, Octal, and Hexadecimal. The full conversation can be seen in Figure 16 below and appendix 7.

How is the weather in Qatar, Mr Killah? – **Base 64 & Base 32**  
Hot as always. Who is this? – **Base 64 & Hex**  
I am a fan of Chess Boxing. I would love to see the title held in Korea – **Base 64 & Base 32**  
We will have to see how the bid turns out. – **Base 64 & Octal**  
Is there anything I could do to make your decision easier? – **Base 64 & Base 32**  
No! The great nation of Qatar would never be swayed so easily. – **Base 64 & Octal**  
Nor would I. We do not take kindly to this pathetic notion of bribery. – **Base 64 & Octal**

Figure 16

## 2.7: Decrypted conversation with Raekwon:

In the fourth conversation, Ill Song tries to contact Raekwon, a member of the ICBA. Raekwon says to Ill Song that he won't be bribed easily and that he would need at least 20 million Rubles. Ill Song accepts his price and tells Raekwon that he will drop the information at an undisclosed location. Therefore, Raekwon is guilty of bribes. Since Raekwon wants the money in Rubles it can be assumed that Raekwon is in Russia.

The conversation between Raekwon and Ill Song was encrypted with Base 64, Base 32, and Hexadecimal. The full conversation can be seen in Figure 17 below and appendix 7.

Mr Raekwon, have you spoken with Mr Razor – **Base 64 & Base 32**  
I have, but I won't be bought so easily. -  
Bought? Of course not. You are an official on the executive committee of the ICBA. I just want you to know that I am here to help make your decision as easy as possible. – **Base 64 & Base 32**  
I would need at least 20 million Rubles – **Base 64 & Hex**  
Consider it done. I will send you the information for the drop-off point soon. – **Base 32 & Base 64**

Figure 17

## 3 – Investigation of Capture 3.pcap:

### 3.1 Filtering Wireshark by FTP traffic:

According to intelligence, FTP traffic between a suspected corrupt official and a foreign national was picked up. Therefore, the first step in this stage was to filter traffic by FTP in order to find evidence of files being transferred. This was done by using the filter ftp-data in Wireshark. The results in Figure 18 show that two files called sandofwhich.zip and ojd34.zip were transferred between the sIP 172.29.1.21 and the dIP 172.29.1.23.

| No.  | ftp-data   | Source      | Destination | Protocol | Length | Info  |
|------|------------|-------------|-------------|----------|--------|---|
| 5880 | 182.022621 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5883 | 182.022634 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5884 | 182.022879 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5886 | 182.022908 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5887 | 182.023121 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5888 | 182.023304 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5889 | 182.023369 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5890 | 182.023381 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5891 | 182.023819 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5892 | 182.023838 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5893 | 182.023876 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5894 | 182.023881 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5895 | 182.024370 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5896 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5897 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5898 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5899 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5900 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5901 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5902 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5903 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5904 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5905 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5906 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5907 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5908 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5909 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5910 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5911 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5912 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5913 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5914 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5915 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5916 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5917 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5918 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5919 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5920 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5921 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5922 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5923 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5924 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5925 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5926 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5927 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5928 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5929 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5930 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5931 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5932 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5933 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5934 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5935 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5936 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5937 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5938 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5939 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5940 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5941 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5942 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5943 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5944 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |
| 5945 | 182.024631 | 172.29.1.21 | 172.29.1.23 | FTP-DA   | 1514   | FTP Data: 1488 bytes (PASV) (RETR sandwich.zip) |

Figure 18: traffic filtered by ftp-data

### 3.2 Save ftp traffic as raw data:

The next step was to follow TCP Stream (TCP Stream 158) and save the data in raw format as seen in Figure 19 in order to view the downloaded zip files. The second step was to unzip the file once the raw data has been downloaded. The command can be found in Figure 20 below. After the raw data was unzipped, two files called sandwich and ojd34 were found containing multiple jpeg files as seen in appendix 8. Upon further inspection, it appeared that the file names within the downloaded files were similar to a quote by Edward Snowden “I can’t in good conscience allow the US government to destroy privacy, internet freedom and basic liberties for people around the world with this massive surveillance machine they’re secretly building.”

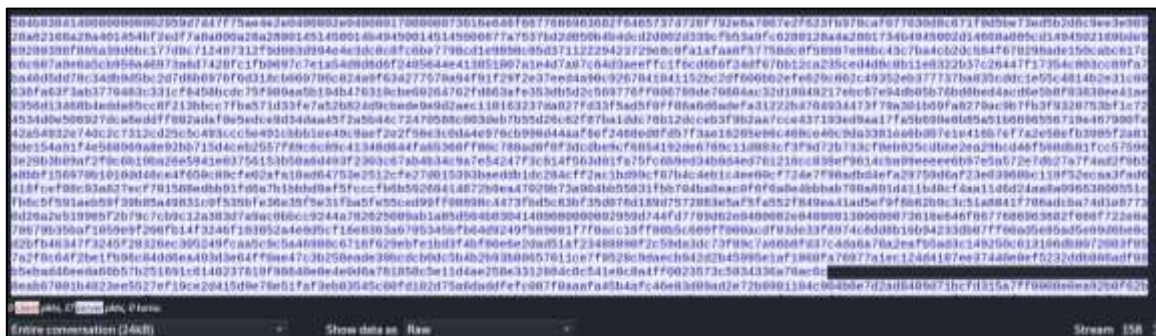


Figure 19

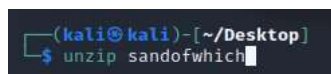


Figure 20

### 3.3 Reconstructing image using cat:

According to intelligence, some anti-forensic practises were used to hide the information. The fact that the names of the files are similar to an Edward Snowden quote, meant that the images were defragmented. Therefore, to stitch the images back together, the cat command was used. However, when the cat command was attempted, the file became corrupted. This meant that the image was incomplete, meaning there had not been any zip files in the capture that weren't in the filtered traffic.



### 3.4 Filtering traffic by mime multipart:

The next step was to search for additional files that might contain .jpg files. This was done by filtering Wireshark traffic by mime multipart (Multipurpose Internet Mail Extensions). This is used to transfer text and non-text attachments over email. When the traffic was filtered, two HTTP packets were found containing three zip files as seen in Figure 22 below. Upon further inspection, an AOL conversation between the aliases [da.genius36@aol.com](mailto:da.genius36@aol.com) and [kim.illsong@aol.com](mailto:kim.illsong@aol.com) was found and the content indicated the transfer of additional jpeg files as seen in Figure 21 below. The jpeg files names also appeared to be the missing words required to complete the Edward Snowden quote.

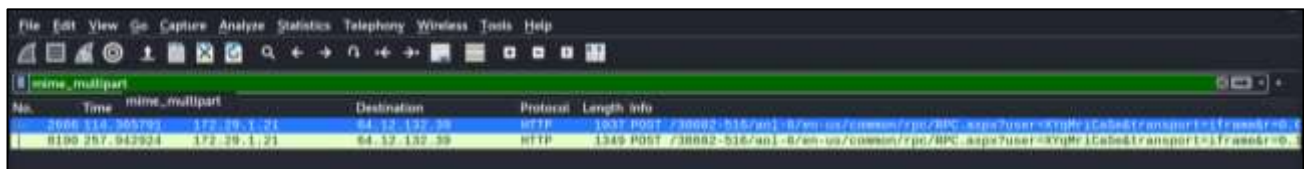


Figure 22: traffic filtered by mime\_multipart



Figure 21: discovered conversation

### 3.5 Export the packet bytes:

The next step was to download the three additional zip files called **34jdsioj**, **breaking\_bad\_season\_6**, and **canc3l** by exporting the packet bytes in Wireshark and saving them as zip files as seen in appendix 9. Once this was done, the exported bytes had to be unzipped using the Linux unzip command seen in Figure 20 above. Each of the downloaded zip files contained the necessary jpeg files to complete the Edward Snowden quote so they were moved into their own folder. The files in Figure 23 below show all the jpeg files that were required to complete the Edward Snowden quote.

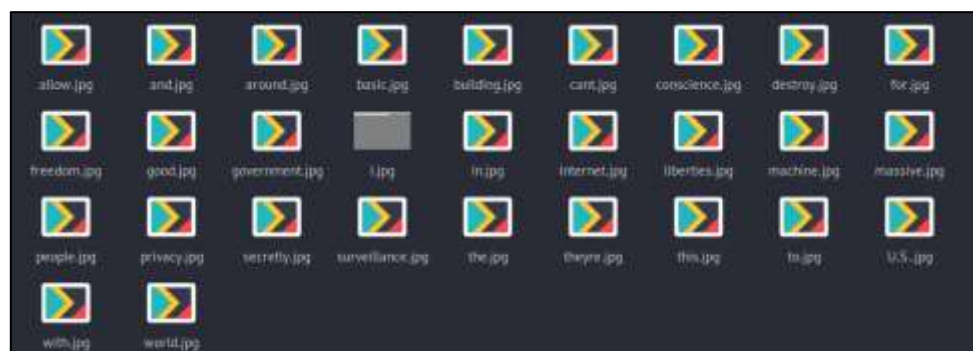


Figure 23

### 3.6 Using cat to reconstruct the image:

After all the images were collected, the images then had to be stitched together starting with I.jpg. This is because I.jpg was the only file with a valid magic number matching a jpeg file which is FF D8 FF. The files were reconstructed using the cat command in the Linux terminal as seen in Figure 24 and appendix 10. Once this was done, an image of a chessboard was reconstructed which can be seen in

Figure 25. The reconstructed image shows that the genius received a chessboard from ill song as a potential bribe.

1 ~\$ cat 1.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 host.jpg liberties.jpg for.jpg people.jpg around.jpg the.jpg world.jpg with.jpg this.jpg massive.jpg surveillance.jpg machine.jpg theyre.jpg secretly.jpg building.jpg + sonnen.jpg

Figure 24



Figure 25: reconstructed image of chessboard

## 4 - Investigation of Capture 4.pcap:

### 4.1: Finding conversation in Wireshark

According to intelligence, a suspect using the alias “Kim Ill-Song” was trying to set up a discreet meeting with another suspect using the alias “Ann Dercover”. Initial analysis of the protocol hierarchy in Wireshark showed that http (port 80) was the most used protocol, this is important because http is not encrypted so filtering by http traffic could reveal the messages in plaintext. Further analysis of the captured packet revealed that the suspects were using Apache Web Server and a URI called “api.pinger.com” to communicate which can be seen **Error! Reference source not found..** This was done by exporting all objects by HTTP and filtering applications by JSON. This revealed many objects with the obvious file name “communications” as seen in Appendix 11. Following the TCP stream on a packet using the URI <http://api.pinger.com/1.0/communications> like packet 3857 revealed a message from Kim Ill Song to Ann Dercover saying “Good Afternoon Ann” which can be seen in Figure 27. Further analysis showed that the message was sent using a Nexus 7 running Android 4.2.2 which most likely means this was an SMS message. They also appeared to be sent using the free application ‘Textfree’ by Pinger as seen in Figure 27.

```

+ Hypertext Transfer Protocol
+ HTTP/1.1 200 OK\r\n
  Date: Wed, 02 Jul 2014 22:39:41 GMT\r\n
  Server: Apache\r\n
  X-Host: sj2-web34\r\n
+ Content-Length: 573\r\n
  Keep-Alive: timeout=10, max=17\r\n
  Connection: Keep-Alive\r\n
  Content-Type: application/json\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.180204000 seconds]
  [Request in frame: 4075]
  [Request URI: http://api.pinger.com/1.0/communications?startIndex=0&since=2014-07-02+22%3A39%3A32]
  File Data: 573 bytes

```

Figure 26



```
POST /1.0/communications?startIndex=0&since=2014-07-02T22:33:33.333Z HTTP/1.1
x-rest-method: GET
Content-Type: application/json
X-Install-Id: 6965eedb59a7b202f94dc50e7a451474
x-client: textfree-android,2.3.2
x-os: android,4.2.2
x-uid: 586781709
x-gid: 0
Authorization: OAuth realm="http://api.pinger.com", oauth_consumer_key="586781709%3Btextfree-android-332781036009711-3404333778292", oauth_signature_method="H
oauth_nonce="bhneatwjpjdb", oauth_signature="ZnMircKHQwZyY4vtn7jzcsbXhnaA30"
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.2.2; Nexus 7 Build/JDQ39F)
Host: api.pinger.com
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 20

{"supportedMessages":["text"]} HTTP/1.1 200 OK
Date: Wed, 02 Jul 2014 22:38:05 GMT
Server: Apache
X-Host: sj2-web12
Content-Length: 585
Keep-Alive: timeout=10, max=36
Connection: Keep-Alive
Content-Type: application/json

{"success": "messages retrieved", "result": {"recMessages": [{"messageId": "45b537c51e5cf2f90f31779e9ec8fc46", "messageType": "normal", "messageText": "Good afternoon,
Ann", "recipientType": "phone", "recipientId": "14060522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02
22:38:55", "messageStatus": "read", "deliveryMethod": "onnet"}], "sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:38:5
5", "numTextsSent": 8, "numTextsRec": 0, "inviteCount": 0}]}
```

Figure 27: message from Kim Ill Song

#### 4.2 find information in SMS conversation

Since the suspects were using the URI “api.pinger.com”, it was then possible to use this as a filter in the command line tool ‘tshark’ in order to see the messages. This was done with the command **tshark -r Capture4.pcap -Y 'http.host' = “api.pinger.com” && http.request.uri contains “send” -T fields -e http.file\_data -e http.request.uri**. The tshark command revealed that Ann Dercover was wanting to meet at a discreet location on the 5<sup>th</sup> of September at 5pm as seen in Figure 28 below. However, the tshark command only revealed sent messages and not received messages which means the conversation was incomplete. The next step was to use grep to view received messages from Kim Ill Song. This was done with the command **grep --binary-file=text messageText Capture4.pcap**. The string “messageText” was used as a grep filter because it was a recurring string in the JSON file. The grep command revealed the SMS conversation between Ann Dercover and Kim Ill-Song that took place between 22:34 and 22:50 on 02/07/2014 as seen in Figure 29 below. According to the SMS conversation, Kim Ill Song wanted to meet Ann at their ‘old meetup’ spot on the 5<sup>th</sup> September. Ann says that Ill-Song should be careful but Ill Song is confident because ‘they’ will never know he is behind the bribes. Furthermore, when Ann asks ‘who is this’, Ill Song replies with ‘Castling’ which probably means he is using an alias to remain anonymous. Unfortunately, the SMS conversation does not reveal a specific location, which means there is another transferred file that reveals the meeting location.

```
$ tshark -r Capture4.pcap -Y 'http.host == "api.pinger.com" && http.request.uri contains "send" -T fields -e http.file_data -e http.request.uri >
messages.txt

{"senderId": "14060522589", "senderName": "Ann", "recipientId": "14069243754", "messageText": "who is this?", "senderType": "phone", "sendAsSms":
0, "recipientType": "phone"} /1.0/messages/text/send?lang=en-US
{"senderId": "14060522589", "senderName": "Ann", "recipientId": "14069243754", "messageText": "where are you?", "senderType": "phone", "sendAsSms":
0, "recipientType": "phone"} /1.0/messages/text/send?lang=en-US
{"senderId": "14060522589", "senderName": "Ann", "recipientId": "14069243754", "messageText": "Do you know that there are people investigating Kim Ill-
Song?", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"} /1.0/messages/text/send?lang=en-US
{"senderId": "14060522589", "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"} /1.0/messages/text/send?lang=en-US
{"senderId": "14060522589", "senderName": "Ann", "recipientId": "14069243754", "messageText": "yes", "senderType": "phone", "sendAsSms":
0, "recipientType": "phone"} /1.0/messages/text/send?lang=en-US
{"senderId": "14060522589", "senderName": "Ann", "recipientId": "14069243754", "messageText": "I told you to pay attention.", "senderType": "phone", "sendAsSms":
0, "recipientType": "phone"} /1.0/messages/text/send?lang=en-US
```

Figure 28



#### 4.4 find location in google earth

Once the coordinates were saved to a CSV file, they then had to be converted into a KML file in order to be compatible with Google Earth. This was done using an online tool called 'www.convertcsv.com' as seen in appendix 14. When the newly created KML file was loaded into Google Earth, over 100 pins were dropped on Missoula in Montana in a shape resembling the number '17'. This most likely means Kim Ill-Song and Ann Dercover are meeting on September 17th at 5pm. A screenshot of Google Earth shows the location of the meeting can be seen in **Error! Reference source not found.** below.

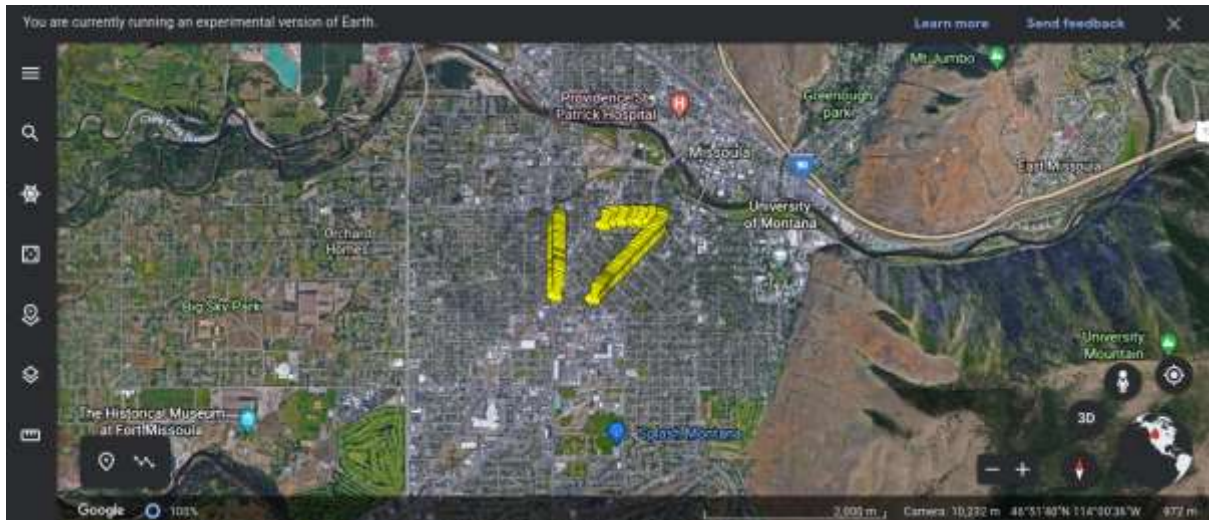


Figure 32

## 5 Critical Analysis & Evaluation:

### 5.1 Capture 1:

The first step before investigating the capture files was to verify the integrity of the evidence by checking the checksum hashes and to make copies of all the capture files. This was done because it is the most forensically sound approach to an investigation. As a result of doing this step, the investigation that followed can be considered reliable and trustworthy because the evidence has not been negatively affected.

Once the integrity of the evidence was verified and backups had been created, the investigation of capture 1 began. Since it was unclear where the evidence existed in the capture file and who was downloading/ sending the files, statistical analysis had to be implemented. Statistical analysis was important for narrowing down the large pool of packets that existed in the capture file and isolating activity relating to specific IP addresses. The tool command line utility 'Silk' was used to narrow down the volume of traffic in capture 1. This tool was useful for the investigation as it revealed the source and destination IP addresses, the protocols in use and the number of bytes sent. This approach was much more efficient and forensically sound than manually searching for clues within Wireshark.

Once Silk narrowed the search area, the next step was to locate and view the files that were downloaded by the suspect. This was done with the open-source packet analysis tool Wireshark. Wireshark was essential in the investigation because it features a vast array of built-in tools and features that greatly speeds up the forensic process. Wireshark made it possible to filter traffic by the IP addresses and protocols found in the statistical analysis stage, in this case the IP address 172.29.1.23 and port 445. This shows that Wireshark used in tandem with statistical analysis leads to an efficient and forensically sound investigation.

Wireshark also made it simple to find and download the evidence due to a built-in feature called 'Export Objects'. This feature allows investigators to filter by specific content types such as JSON, HTML, or plain-text files. However, the disadvantages to this method are that it is not possible to download individual packets, instead Wireshark downloads every packet inside the object list which can slow the investigation down.

Overall, case 1 was conducted in a forensically sound manner with no serious problems occurring that might have compromised the investigation. The integrity of the evidence was validated after the investigation concluded, which means the results of case 1 can be trusted. The tools involved in the investigation such as Silk and Wireshark were more than adequate to achieve the end result.

## 5.2 Capture 2:

Once all the evidence was successfully extracted from capture 1, the next step was to investigate capture 2. Since intelligence had IRC monitors, they knew that encrypted traffic being transmitted so it was not necessary to use statistical analysis. Therefore, it was relatively easy to use Wireshark's built-in packet filter to locate all the packets that were using the IRC protocol.

After locating the data packets that were suspected to contain encrypted conversations, the next step was to view packet contents in a more readable format. This stage is where one of Wireshark's built-in features greatly assisted the investigation. The feature 'follow TCP Stream' made it possible to view the entire encrypted conversation on one page instead of having to view each packet manually to find the conversations. This feature greatly improved overall efficiency and made the investigation more forensically sound.

Having made use of Wireshark's 'follow TCP Stream' feature, the next step was to decode the conversation because they were all encoded using unknown methods. Luckily, there are many open-source tools online that can detect encoding schemes and decode them automatically. One tool that was of great benefit to the investigation of capture 2 was CyberChef. This tool has a built-in feature that automatically detected the use of Base 64, Base 32, and Hex in the conversations, which meant there was less time spent trying to identify the encoding scheme. However, despite CyberChef ability to detect a lot of encryption schemes, it wasn't perfect. The tool was unable to detect the use of Octal and MD5, therefore a lot of valuable time was spent finding other tools and researching encoding patterns.

Overall, case 2 was conducted in a forensically sound manner with few problems occurring. The only problem that affected the efficiency of the investigation was CyberChef's inability to detect the use of Octal and MD5. However, enough evidence was collected to convict the individuals in the conversation. Furthermore, the integrity of the evidence was validated after the investigation concluded, which means the results of case 2 can be trusted. Tools such as Wireshark were more than adequate to achieve the end result.

## 5.3 Capture 3:

Once all the evidence was successfully extracted from capture 2, the next step was to investigate capture 3. Unlike capture 1, it was not necessary to undertake statistical flow analysis, because intelligence already suspected the use of FTP traffic between a corrupt official and a foreign national. Therefore, like capture 2, it was possible to make use of Wireshark's built-in filter, to filter packets by FTP traffic. This useful feature allowed the investigator to locate two zip files that were essential to the investigation.

After locating the zip files, the 'follow TCP stream' feature was successfully utilised again. Wireshark also has the useful ability to save the data into different formats, in this case the data was saved as

raw. Wireshark's 'follow TCP Stream' feature, the capability to filter by specific protocols, and the ability to save raw bytes of data for future analysis increased the overall efficiency of the investigation and removed the need for complicated command line tools.

Since there were not enough images to complete the Edward Snowden quote, Wireshark had to be used again to filter traffic by mime\_multipart. However, unlike the previous step it was not possible to download the data directly from the tcp stream. Instead, the investigator had to navigate to the decoded data within the packet and export the packet bytes to a zip file. The extra steps involved slowed the forensic process down, but it also shows that if one feature of Wireshark does not work for the methodology then there are still other methods and features in Wireshark to extract information.

After extracting the jpeg files from the zip files the next step was to join the images together as the suspects used anti forensic techniques to hide the real image. This was the stage in case 3 where making a python or bash script would have been very useful. The method selected in this investigation involved using the 'cat' command to join each fragmented image together into one single image. However, this process was very slow and arduous and often resulted in the image becoming blurred. This is why it is very important to make copies of the original jpeg images, so they don't get corrupted thus damaging the validity of the investigation. A more efficient method would be to make a python script that loops through the folder containing the files and automatically joins the fragmented jpeg files onto the first jpeg.

Overall, case 3 was conducted in a forensically sound manner. The only issue with the investigation was the overall efficiency of the forensic process. The integrity of the evidence was maintained throughout the investigation since copies were made of the original jpeg files.

#### 5.4 Capture 4:

Once all the evidence was successfully extracted from capture 3, the next step was to investigate capture 4. The methodology involved in this capture was probably one of the most efficient in the whole investigation because the investigator made use of python scripts to automate tasks that otherwise would have taken much longer. Wireshark's wide range of features were also put to good use in this investigation. The investigation also made use of command line tools such as tshark, and grep which made the investigation more forensically sound as it is important to not rely too much on a single tool. Using Wireshark's built in TCP Stream feature it was possible to view information about the suspect such as their mobile phone model, OS, and one crucial element a uri called 'api.pinger.com'. This was then used as a filter for the command line tool 'tshark' to filter all sent messages in the capture using api.pinger.com as a host. The tshark command had advantages in that it had a clean output, was fast, and it was very verbose in its findings. However, the disadvantages were that the commands in tshark were prone to failing due to its complexity. Since tshark only outputted sent messages from Ann Dercover, the grep command had to be used to get the received messages. The grep command was useful in that it provided all the messages required to convict the suspects, but the formatting made it very hard to read. A python script was attempted in order to automatically strip filter out the messages in the JSON files, but this was unsuccessful because the JSON output contained many non UTF-8 characters.

Overall, case 4 was conducted in a forensically sound manner. The approach was efficient due the inclusion of a python script to filter out the coordinates from the uri 'mob.maprequest.api'. The script could have been improved by automatically pasting the coordinates into a csv file instead of manually creating the csv file.



## Appendix 1:

[illegible]

Figure 33

Wireshark packet capture analysis showing a series of HTTP requests and responses. The packet list on the left shows packets 22487 through 22573. The packet details pane on the right shows the structure of an HTTP 200 OK response, including the status bar, headers (Content-Type, Content-Length, Content-Disposition), and the body (HTML). The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

| No.  | Time       | Source         | Destination    | Protocol | Length | Info  |
|--|------------|----------------|----------------|----------|--------|---|
| 22487  | 587.531818 | 172.29.1.20    | 206.188.18.187 | HTTP     | 1288   | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22488  | 587.531818 | 172.29.1.20    | 206.188.18.187 | HTTP     | 604    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22491  | 587.207133 | 213.212.52.58  | 172.29.1.20    | HTTP     | 202    | HTTP/1.1 204 No Content   |
| 22494  | 587.540513 | 172.29.1.20    | 206.188.14.146 | HTTP     | 685    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22496  | 587.550045 | 206.188.18.187 | 172.29.1.20    | HTTP     | 503    | HTTP/1.1 200 OK (text/html)   |
| 22497  | 587.642073 | 98.233.141.146 | 172.29.1.20    | HTTP     | 793    | HTTP/1.1 200 OK (text/html)   |
| 22498  | 587.782218 | 172.29.1.20    | 148.174.86.98  | HTTP     | 287    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22499  | 587.786455 | 172.29.1.20    | 172.29.1.20    | HTTP     | 1094   | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22501  | 587.801034 | 206.188.18.187 | 172.29.1.20    | HTTP     | 503    | HTTP/1.1 200 OK (text/html)   |
| 22502  | 587.835187 | 152.163.13.65  | 172.29.1.20    | HTTP     | 626    | HTTP/1.1 200 OK (application/javascript)  |
| 22506  | 587.873986 | 148.174.86.98  | 172.29.1.20    | HTTP     | 291    | HTTP/1.1 200 OK   |
| 22507  | 587.960390 | 172.29.1.20    | 255.143.12.68  | HTTP     | 1017   | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22508  | 587.979784 | 172.29.1.20    | 172.29.1.20    | HTTP     | 643    | HTTP/1.1 200 OK (application/javascript)  |
| 22511  | 588.235000 | 172.29.1.20    | 148.174.86.98  | HTTP     | 424    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22496  | 588.275617 | 148.174.86.98  | 172.29.1.20    | HTTP     | 138    | HTTP/1.1 200 OK (application/javascript)  |
| 22498  | 588.930648 | 172.29.1.20    | 54.86.10.73    | HTTP     | 693    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22498  | 588.931067 | 54.86.10.73    | 172.29.1.20    | HTTP     | 631    | HTTP/1.1 302 Moved Temporarily  |
| 22499  | 588.961461 | 172.29.1.20    | 94.84.238.238  | HTTP     | 688    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22507  | 588.960637 | 94.84.238.238  | 172.29.1.20    | HTTP     | 507    | HTTP/1.1 302 Moved Temporarily  |
| 22512  | 588.965029 | 172.29.1.20    | 94.84.238.238  | HTTP     | 746    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22526  | 588.975799 | 54.84.238.238  | 172.29.1.20    | HTTP     | 717    | HTTP/1.1 302 Moved Temporarily  |
| 22521  | 589.746550 | 172.29.1.20    | 54.86.10.73    | HTTP     | 133    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22525  | 589.765695 | 54.86.10.73    | 172.29.1.20    | HTTP     | 634    | HTTP/1.1 200 OK (text/html)   |
| 22526  | 589.871131 | 172.29.1.20    | 213.212.52.58  | HTTP     | 832    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22531  | 589.969074 | 213.212.52.58  | 172.29.1.20    | HTTP     | 362    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22498  | 589.974203 | 172.29.1.20    | 206.188.18.187 | HTTP     | 249    | POST /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1 |
| 22542  | 590.074185 | 206.188.18.187 | 172.29.1.20    | HTTP     | 559    | HTTP/1.1 200 OK, JavaScript Object Notation (application/json)                            |
| 22541  | 591.073093 | 172.29.1.20    | 206.188.18.187 | HTTP     | 486    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22542  | 591.080002 | 172.29.1.20    | 245.174.98.88  | HTTP     | 262    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22551  | 591.107133 | 206.188.18.187 | 172.29.1.20    | HTTP     | 832    | HTTP/1.1 200 OK (application/javascript)  |
| 22554  | 591.181283 | 148.174.86.98  | 172.29.1.20    | HTTP     | 291    | HTTP/1.1 200 OK   |
| 22561  | 591.875489 | 172.29.1.20    | 206.188.18.187 | HTTP     | 1221   | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22562  | 591.879992 | 172.29.1.20    | 213.212.52.58  | HTTP     | 701    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22564  | 591.880721 | 172.29.1.20    | 98.233.141.146 | HTTP     | 637    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| 22565  | 591.918148 | 213.212.52.58  | 172.29.1.20    | HTTP     | 742    | HTTP/1.1 200 OK (text/html)   |
| 22568  | 591.940914 | 98.233.141.146 | 172.29.1.20    | HTTP     | 782    | HTTP/1.1 200 OK (text/html)   |
| 22573  | 591.960392 | 206.188.18.187 | 172.29.1.20    | HTTP     | 503    | HTTP/1.1 200 OK (text/html)   |
| 22575  | 592.040693 | 206.188.18.187 | 172.29.1.20    | HTTP     | 503    | HTTP/1.1 200 OK (text/html)   |
| 22576  | 592.115693 | 172.29.1.20    | 64.12.132.55   | HTTP     | 117    | POST /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1 |
| 22577  | 592.180821 | 172.29.1.20    | 64.12.132.55   | HTTP     | 508    | HTTP/1.1 200 OK (application/javascript)  |
| 22586  | 592.315139 | 172.29.1.20    | 64.12.132.55   | HTTP     | 508    | POST /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1 |
| 22581  | 592.315139 | 64.12.132.55   | 172.29.1.20    | HTTP     | 841    | HTTP/1.1 200 OK, JavaScript Object Notation (application/json)                            |
| 22582  | 592.315139 | 172.29.1.20    | 172.29.1.20    | HTTP     | 688    | GET /Beacon(s)/all/en-us/BeaconReadMessage?view=live.htm?rnd=6.9374611556314055 HTTP/1.1  |
| Frame 22481: 941 bytes on wire (7528 bits) captured (7528 bits) on 0:0 |            |                |                |          |        |   |



## Appendix 2:

| Packet | Hostname     | Content Type        | Size    |
|--------|--------------|---------------------|---------|
| 17979  | mail.aol.com | multipart/form-data | 8,654kB |

Figure 35

|   |                         |                         |                   |      |      |                               |
|---|-------------------------|-------------------------|-------------------|------|------|-------------------------------|
| 17979   | 343.371300              | 172.29.1.23             | 64.12.132.55      | HTTP | 1342 | POST /38029-11/wml-6/en-w     |
| 17980   | 343.371305              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17981   | 343.372235              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17982   | 343.372733              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17983   | 343.372983              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17984   | 343.457420              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 [TCP Dup ACK 17983=1] 88 → |
| 17985   | 343.457426              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 [TCP Dup ACK 17985=2] 88 → |
| 17986   | 343.457666              | 64.12.132.55            | 172.29.1.23       | TCP  | 88   | 88 [TCP Dup ACK 17986=3] 88 → |
| MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----7de2582b26336/r/n" |                         |                         |                   |      |      |                               |
| [Type: multipart/form-data]   |                         |                         |                   |      |      |                               |
| First boundary: -----7de2582b26336/r/n  |                         |                         |                   |      |      |                               |
| Encapsulated multipart part: (application/x-zip-compressed)                                       |                         |                         |                   |      |      |                               |
| Content-Disposition: form-data; name="file0"; filename="DocsPcap.zip"/r/n                         |                         |                         |                   |      |      |                               |
| Content-Type: application/x-zip-compressed/r/n/r/n  |                         |                         |                   |      |      |                               |
| Media Type: application/x-zip-compressed (8652618 bytes)  |                         |                         |                   |      |      |                               |
| 8600110a0   | 72 65 73 72 65 64 68 6a | 8d 8a 5c 46 83 94 18 88 | r/essent. PK      |      |      |                               |
| 8600110f2   | 80 80 80 80 80 80 80 80 | 44 9f 65 73 2c 70 89 41 | 1..... Docs Doc   |      |      |                               |
| 860011090   | 4b 4b 4b 4b 4b 4b 4b 4b | 44 9f 65 73 2c 70 89 41 | ..... Doc Doc     |      |      |                               |
| 860011116   | 70 ec 8a 7a 3c 94 0f 7b | 0f 3c 55 27 84 24 c8 82 | p..... 70: 5: 2   |      |      |                               |
| 860011129   | 11 82 88 55 08 c5 52 83 | 83 8c 70 89 39 c9 c8 88 | wv7.83.....       |      |      |                               |
| 860011130   | 80 39 0c 42 84 c2 86 78 | 53 24 c5 23 45 43 84 43 | Y.D.E.x 00:7..... |      |      |                               |
| 8600111a0   | 88 23 84 38 1a 85 48 6d | 88 23 84 38 1a 85 48 6d | L00111E 82.....   |      |      |                               |
| 86001118a   | 53 44 8a 8f c7 6f 73 78 | c7 6f 73 78 57 7b 70 72 | w.....            |      |      |                               |
| 860011180   | 0f c5 3a c2 78 56 67 78 | 80 73 c7 80 73 35 d0 c5 | .....             |      |      |                               |
| 860011176   | 29 80 30 0f 9f 81 31 88 | 84 8a 7c 0f 30 08 40 40 | .....             |      |      |                               |
| 860011108   | 8e 40 00 00 8a 88 81 5d | 23 92 8a 18 c1 3d 28 28 | .....             |      |      |                               |
| 860011190   | c2 8f 8b 3f a5 40 c8 18 | c5 14 8f c2 38 38 2c 25 | .....             |      |      |                               |
| 8600111a0   | 0f c7 7d 4b 50 c0 39 8a | 80 c3 2e 84 43 c3 80 6a | g 3FP.8.....      |      |      |                               |
| 860011180   | 8a 23 84 38 1a 85 48 6d | 39 93 3a 2c 90 48 82 63 | 2a.....           |      |      |                               |
| 86001118a   | 23 22 38 70 14 18 83 43 | c0 22 18 86 30 2a 9f 83 | 10.....           |      |      |                               |
| 860011180   | 4b 80 07 c4 02 8a 44 78 | 7a 3c 94 0f c7 48 90 85 | .....             |      |      |                               |
| 860011168   | 14 7b e0 40 92 1a 55 24 | 46 99 29 13 31 25 2e 99 | .....             |      |      |                               |
| 860011176   | 81 3c 8a 43 5a 8a 88 8e | 03 19 90 89 3c 91 88 2f | .....             |      |      |                               |
| 860011180   | 14 8a 8a 70 38 73 82 83 | c8 c3 e8 82 97 c3 82 88 | .....             |      |      |                               |
| 860011118   | 70 1c 00 db 80 80 80 80 | 30 97 2c c5 83 8a 42 8a | .....             |      |      |                               |
| 860011220   | 0f c2 33 7d 88 82 3f 86 | 88 8a 88 83 8a 26 38 2a | .....             |      |      |                               |
| 86001128a   | 20 38 18 00 85 12 8a 84 | 8a 08 22 3c 20 2a 9f 83 | .....             |      |      |                               |
| 860011240   | 70 57 18 87 c3 52 49 10 | 38 89 8c 80 8d 88 8a    | .....             |      |      |                               |
| 860011268   | 28 c8 0f 80 19 c7 e4 01 | 26 74 7c 34 36 67 43 05 | .....             |      |      |                               |
| 860011200   | 67 40 c1 61 38 c3 89 3c | 86 07 2a 83 79 87 c9 12 | .....             |      |      |                               |
| 860011278   | 8c 10 4e 38 01 31 1e 83 | 82 83 55 c1 3c 00 7e 7c | .....             |      |      |                               |
| 860011280   | 7b 24 3c 61 87 74 13 38 | 77 66 80 c2 c3 a7 08 83 | .....             |      |      |                               |
| 860011280   | 8d 1a 6a 8a 3a 9a 29 08 | 48 89 00 29 82 7b 78 11 | .....             |      |      |                               |
| 8600112a0   | 0f c3 43 15 88 35 8a 38 | 8c c5 08 34 84 61 90 86 | .....             |      |      |                               |

Figure 37

|   |                         |                         |                     |      |      |                               |
|---|-------------------------|-------------------------|---------------------|------|------|-------------------------------|
| 17979   | 343.371300              | 172.29.1.23             | 64.12.132.55        | HTTP | 1342 | POST /38029-11/wml-6/en-w     |
| 17980   | 343.371305              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17981   | 343.372235              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17982   | 343.372733              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17983   | 343.372983              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 → 50100 [ACK] Seq=1246 A   |
| 17984   | 343.457412              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 [TCP Dup ACK 17983=1] 88 → |
| 17985   | 343.457426              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 [TCP Dup ACK 17985=2] 88 → |
| 17986   | 343.457666              | 64.12.132.55            | 172.29.1.23         | TCP  | 88   | 88 [TCP Dup ACK 17986=3] 88 → |
| MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----7de2582b26336/r/n" |                         |                         |                     |      |      |                               |
| [Type: multipart/form-data]   |                         |                         |                     |      |      |                               |
| First boundary: -----7de2582b26336/r/n  |                         |                         |                     |      |      |                               |
| Encapsulated multipart part: (application/x-zip-compressed)                                       |                         |                         |                     |      |      |                               |
| Content-Disposition: form-data; name="file0"; filename="DocsPcap.zip"/r/n                         |                         |                         |                     |      |      |                               |
| Content-Type: application/x-zip-compressed/r/n/r/n  |                         |                         |                     |      |      |                               |
| Media Type: application/x-zip-compressed (8652618 bytes)  |                         |                         |                     |      |      |                               |
| 8600110a0   | 54 75 d0 ba 52 2a 88 88 | 74 fa a6 cc 88 00 45 00 | Tn 6" t E           |      |      |                               |
| 860011012   | 05 30 26 50 4b 00 88 8a | 5a 00 ac 10 81 17 40 0c | 0000 A.....         |      |      |                               |
| 860011030   | 84 37 c4 84 80 88 32 2b | 86 6c 51 b2 8d 90 30 18 | 7... P2+ n m P      |      |      |                               |
| 860011050   | 80 7f 80 17 00 00 01 80 | 37 00 00 00 7e 66 84 00 | .....               |      |      |                               |
| 860011070   | 80 80 6d 8a 2d 2d 2d 2d | 2d 2d 2d 2d 2d 2d 2d 2d | -----7de2582 b26336 |      |      |                               |
| 860011090   | 2d 37 64 65 32 95 38 92 | 62 32 30 33 33 36 0d 0a | Content- Disposit   |      |      |                               |
| 8600110f0   | 43 6f 6e 74 65 6e 74 3b | 44 69 73 79 6f 73 69 74 | ion: for m-data     |      |      |                               |
| 860011110   | 09 6f 6e 3a 2b 66 6f 72 | 6d 2d 84 61 74 61 3b 2b | name="fi le1": fi   |      |      |                               |
| 860011130   | 6e 03 6d 05 3d 22 66 69 | dc 65 31 22 3b 2b 66 69 |                     |      |      |                               |

Figure 36

## Appendix 3:

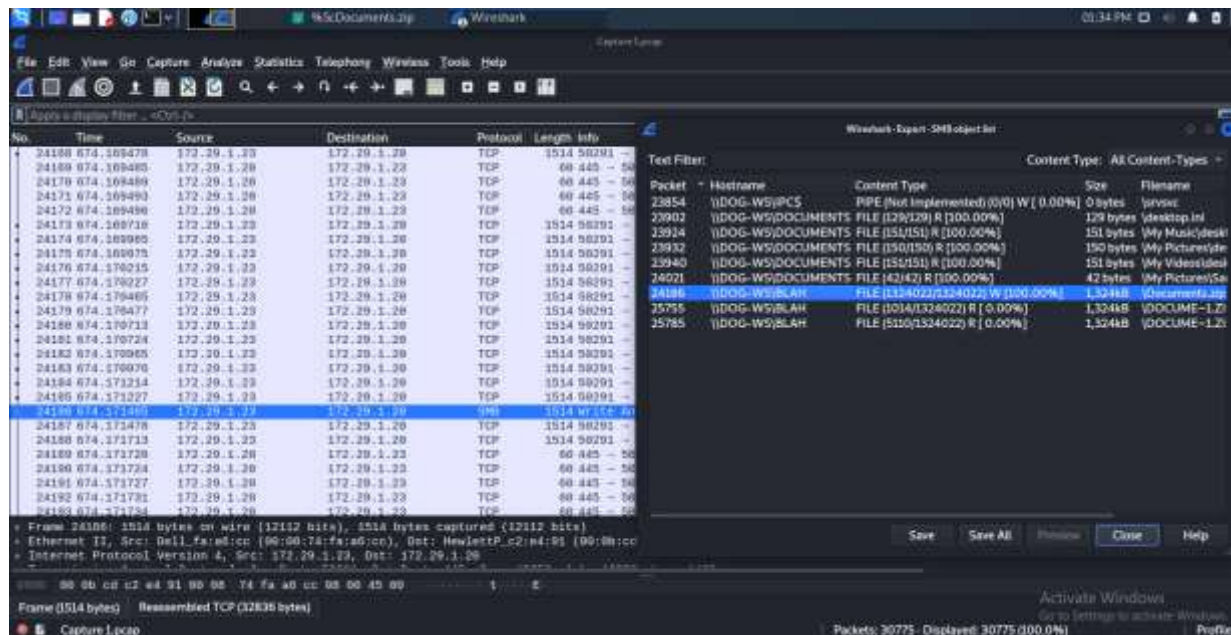


Figure 39

[illegible]

Figure 38

```

C:\Users\hasek> cd ~/Desktop/smbfiles.
> unzip KSCDocuments.zip
Archive: KSCDocuments.zip
  creating: Documents/Actual Documents/
  inflating: Documents/Actual Documents/Got Spoilers.docx
  inflating: Documents/Actual Documents/NorthKorea.docx
  inflating: Documents/Actual Documents/FID.docx
    creating: Documents/Cheess Boxing/
  inflating: Documents/Cheess Boxing/NK.jpg
  inflating: Documents/Cheess Boxing/Rules 1..docx
  inflating: Documents/Cheess Boxing/Rules 2.docx
  inflating: Documents/Cheess Boxing/Rules 3.docx
  inflating: Documents/Cheess Boxing/Rules 4.docx
  inflating: Documents/Cheess Boxing/Rules 5.docx
  inflating: Documents/Cheess Boxing/Rules 6.docx
  inflating: Documents/Cheess Boxing/Rules 7.docx
    creating: Documents/Enter the WuTang/
  inflating: Documents/Enter the WuTang/track10.docx
  inflating: Documents/Enter the WuTang/track10.docx
    creating: Documents/More Documents/
  inflating: Documents/More Documents/5110FRights.txt
  inflating: Documents/More Documents/NorthKorea.jpgx
  extracting: Documents/untilted folder.zip

```

Figure 40

## Appendix 4:

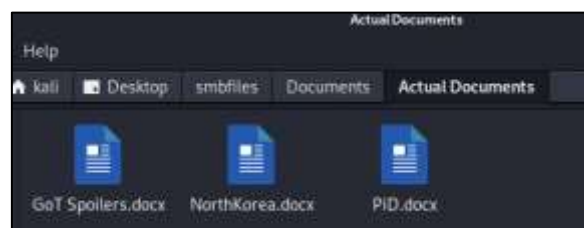


Figure 41

Sm9uTFNub3cgYnVybniMgZG93biBXaW  
5oZXJmZWxsIChhZ2ZpbiKgYWs5kTHRoZ  
SBXlWxsLgoKDQp1b2RvcjBraWxscyB  
UaGVvbi4NCgoKRGFlbmVyeXMgZ2Vo  
cyBlYXRlbiBieSBhIGRyYWdvbi4NCgoK  
U3Rhbm5pcyBmYWxscyBpbiBsb3ZlTHdp  
dGggVHlyaW9uLiANCgoKDQo=

Figure 43

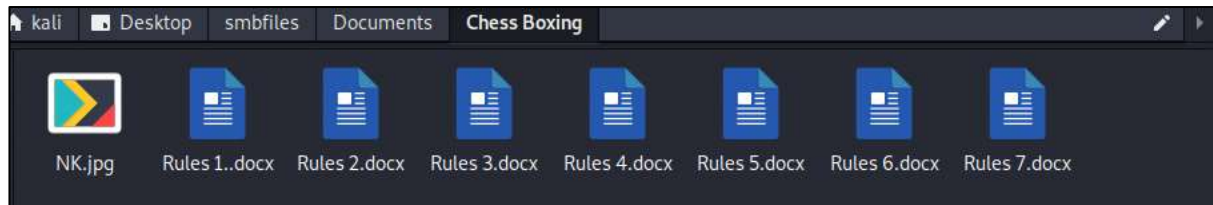


Figure 42

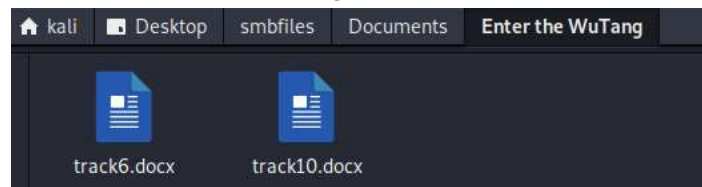


Figure 44

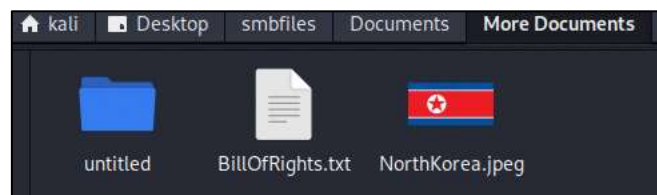


Figure 45

```
(kali@kali)-[~/Desktop/Documents/More Documents]
$ unzip NorthKorea.jpeg
Archive: NorthKorea.jpeg
warning [NorthKorea.jpeg]: 3453 extra bytes at beginning or within zipfile
(attempting to process anyway)
creating: untitled/
inflating: untitled/broken.py
```

Figure 46

```

1 def fileToString(pathToFile):
2     f = open(pathToFile, "r")
3     str = ""
4     #adds each line of the file to the str string
5     for line in f.readlines():
6         str+=line
7     return str
8 def ASCII():
9     #number of ASCII characters
10    NumOfASCII = 0
11    #returns list of all ASCII characters
12    return "".join([chr(i) for i in range(NumOfASCII)])
13 def sumName(name):
14     sums=0
15     #sums the indices in ASCII of all the characters in name
16     for x in name:
17         sums+=ord(x)
18     return sums
19 def indexInFile(password):
20     indices = []
21     ASCIIArray = ASCII()
22     #populates an array of indices to be used by the encoder
23     for chr in password:
24         indices.append(ASCIIArray.index(chr)+sumName(name)*7)
25     return indices
26 def indexInASCII(name):
27     indices = []
28     ASCIIArray = ASCII()
29     #split on all non-numeric characters
30     #remove first index because it is blank
31     indexList = re.split("[^\d]", encoded)[1:]
32     #converts encoded characters to ASCII
33     for index in indexList:
34         indices.append(ASCIIArray[int(index) - (sumName(name)*7)])
35     #returns decoded message
36     return "".join(indices)
37 def encode(name):

```

Figure 47

|                   |
|-------------------|
| Method Man        |
| Kim Ill-Song      |
| Razor             |
| Mr Genius         |
| Ghost Face Killah |
| Matt Cassel       |
| Inspectah Deck    |
| Ol Dirty B*stard  |
| Raekwon           |
|                   |
| U God             |
| Cappadonna        |
| John Woo          |

Table 1



## Appendix 5:

| File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help |            |               |               |          |        |                    |
|--|------------|---------------|---------------|----------|--------|--------------------|
| irc  |            |               |               |          |        |                    |
| No.  | Time       | Source        | Destination   | Protocol | Length | Info               |
| 16   | 13.268894  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 17   | 13.454498  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 22   | 15.058733  | 172.29.1.17   | 185.30.166.35 | IRC      | 276    | Request (PRIVMSG)  |
| 28   | 22.548315  | 172.29.1.17   | 185.30.166.35 | IRC      | 74     | Request (PING)     |
| 30   | 22.733678  | 185.30.166.35 | 172.29.1.17   | IRC      | 114    | Response (PONG)    |
| 35   | 28.265886  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 37   | 28.451405  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 43   | 34.232842  | 185.30.166.35 | 172.29.1.17   | IRC      | 220    | Response (PRIVMSG) |
| 53   | 43.262962  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 54   | 43.448567  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 65   | 52.573610  | 172.29.1.17   | 185.30.166.35 | IRC      | 74     | Request (PING)     |
| 74   | 58.260120  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 77   | 58.445729  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 95   | 70.677820  | 185.30.166.35 | 172.29.1.17   | IRC      | 448    | Response (PRIVMSG) |
| 100  | 73.257281  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 101  | 73.442885  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 108  | 82.567928  | 172.29.1.17   | 185.30.166.35 | IRC      | 74     | Request (PING)     |
| 109  | 82.753533  | 185.30.166.35 | 172.29.1.17   | IRC      | 114    | Response (PONG)    |
| 118  | 88.254188  | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 120  | 88.439793  | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 122  | 88.671858  | 172.29.1.17   | 185.30.166.35 | IRC      | 340    | Request (PRIVMSG)  |
| 134  | 103.251348 | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 136  | 103.436965 | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 148  | 112.311445 | 185.30.166.35 | 172.29.1.17   | IRC      | 204    | Response (PRIVMSG) |
| 151  | 112.577735 | 172.29.1.17   | 185.30.166.35 | IRC      | 74     | Request (PING)     |
| 161  | 118.248508 | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |
| 164  | 118.434116 | 185.30.166.35 | 172.29.1.17   | IRC      | 103    | Response (303)     |
| 173  | 126.114049 | 172.29.1.17   | 185.30.166.35 | IRC      | 84     | Request (PRIVMSG)  |
| 185  | 133.245667 | 172.29.1.17   | 185.30.166.35 | IRC      | 96     | Request (ISON)     |

• Frame 16: 96 bytes on wire (768 bits), 96 bytes captured (768 bits)  
 • Ethernet II, Src: Dell fa:a6:cc (00:08:74:fa:a6:cc), Dst: Cisco ba:52:2a (54:75:d0:ba:52:2a)  
 • Internet Protocol Version 4, Src: 172.29.1.17, Dst: 185.30.166.35  
 • Transmission Control Protocol, Src Port: 50588, Dst Port: 6667, Seq: 1, Ack: 1, Len: 42  
 • Internet Relay Chat

```

0000  54 75 d0 ba 52 2a 00 00  74 fa a6 cc 08 00 45 00  Tu R' - t - - - E
0010  00 52 75 77 40 00 00 06  78 be ac 1d 01 11 b9 1e  Ruw@ - - x - - - -
0020  a0 23 c5 9c 1a 0b 06 40  48 40 34 b4 cb 52 50 18  # - - - F HF4 RP
0030  00 ff 71 d6 00 00 49 53  4f 4e 20 52 61 7a 6f 72  - q - - IS ON Razor
0040  20 47 65 6e 69 75 73 20  52 61 65 6b 77 6f 6e 20  Genius Raekwon
0050  4b 69 6c 6c 61 68 20 4d  65 74 68 6f 64 20 0d 0a  Killah M ethod - -
  
```

Figure 48

## Appendix 6:

| Wireshark: Follow TCP Stream (tcp.stream eq 0) - Capture 2.pcap  |                    |
|--|--------------------|
| <pre> [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius PRIVMSG Razor: :512aQz9JQ1W9Jv9hJ95TuhCv9w9u9RUJ7WFFZM9pPUj9X9UJ9Q1W9F9u9J9B1J9V96J9RFF9S19Y9C9R9K10M9B9M9Z9H9Uj9Y9F9a9k9J7J1V9S9Q9V9C9Q9C9N9K1J9U9W9E1M9P9J9H9U9EV9U9G9H9C :4K9p9T9U9V9Z1T99H9FF19T9a9Q9t9CN9C9M9R19U9Z9V9Z9a9k9k PING :LAG2311957802 :verne.freemove.net PONG verne.freemove.net :LAG2311957802 [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius Razor1:-malware@216.14.247.40 PRIVMSG ILL Song : WTCZNTZJW9H9D9C9J9ZNTJW9JQ2NTY2NjK3MZY9N9Y2ZTJW9Jk3MZY9W9U2ZJc9MjA2NjY5W9U2HTZJWjA30TY1N2QyZQ== [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius PING :LAG2341989902 PRIVMSG Razor: :5819V9Z9V9C9R1X9F9a9Q9S9G1N9W9V9C9T9Y9W9k9Y9Ed9P9V9DQ9Y9S9U9G1N9B9U9K1ZT9U9Z9M9F9Qj9X511P9U9Z9U9Z9W9K1T9J9W9E9W9K9Q9R19Y9D9E9W941M9J9Z9W9J9T1J9V9U9S9K9P9U9J9T9V9U9S9K9V9V9H :59URCT19T9Q9F9T9F9I9V9H1J9H9K9a9U9L9S9U9Y9T9J9W9L1J9999V9P9Y9U9J9D9T94Z9W9L9W9G9T9J9W9L9J9F9V9N9F9H19U9R9D9W9K9P9T9D :verne.freemove.net PONG verne.freemove.net :LAG2341989902 [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius Razor1:-malware@216.14.247.40 PRIVMSG ILL Song : NDk9MDY9N9Q9MDY9XjA2NjY1N2I3OTJW9JZ9WTCZ9K9Y9D9Z9NjE2ZTJjWjA2NjY1N2QyZQ9U9J9MjY4NjE3MDZ9MjA0GTJW9jR2ZjC1N9M9R9D1W9jI2NTJW9Z9Z9T9cy9Z9M9J9T9Y9HjQ2NTY0MjA3NDZ9NjA3 NjY5N2M2OTc9M9U9D9Z9U2NTJW9Jk2NjZW9TA2OTZ9W9U2N2c5NjE2ZTY3MjA2OTc9MjA3NDY4NjU9MDcy9Jk2NjY4R2Qy9MDcy9W9M2NTY2NjU9MDY2ZW9Y3MjW9Z9Q9D9Y1MjA3N2Z9N2I2Y9Y9MjA3NDY5 N2Q9Y2Y1M9U9 [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius PING :LAG2371989902 :verne.freemove.net PONG verne.freemove.net :LAG2371989902 [ISON Razor Genius Raekwon Killah Method :verne.freemove.net 383 ILL Song :razor genius :5819V9Z9V9C9R1X9F9a9Q9S9G1N9W9V9C9T9Y9W9k9Y9Ed9P9V9DQ9Y9S9U9G1N9B9U9K1ZT9U9Z9M9F9Qj9X511P9U9Z9U9Z9W9K1T9J9W9E9W9K9Q9R19Y9D9E9W941M9J9Z9W9J9T1J9V9U9S9K9P9U9J9T9V9U9S9K9V9V9H </pre> |                    |
| Entire conversation (17KB)   | Show data as ASCII |

Figure 49

## Appendix 7:

Mr. Razor, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang. – **Base 64 & Base 32**

Well, the decision is not final yet – **Base 64 & Hex**

Pyongyang is beautiful this time of year. Perhaps you would like to visit and experience what Best Korea has to offer – **Base 64 & Base 32**

I am a very busy man, but perhaps I could be persuaded to visit. See if Pyongyang is the right place for the World Title. – **Base 64 & Hex**

Perhaps not. How about I send you a gift? Something to get you out of the City of Love and take your own vacation somewhere. – **Base 64 & Base 32**

Somewhere expensive, I hope. – **Base 64 & Base 32**

5 – **Base 64 & Base 32**

9 – **Base 64 & Hex**

7 – **Base 64 & Base 32**

\$700,000 it is. Where can I meet you? – **Base 64 & Hex**

I will be in touch with the address. – **Base 64 & Hex**

As we discussed earlier, I believe I might be able to help you with your search. – **Base 64 & Base 32**

I see. Then we must meet, and I will see the validity of this claim. – **Base 64 & Octal**

I can be in Caracas within a week – **Base 64, Base 32, & MD5**

No. Not here. Can I not go to you? – **Base 64 & Octal**

I am afraid that would be unwise. I will send you a message with the date and location through a more secure form of communication. – **Base 64 & Base 32**

Mr. Method, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang. – **Base 64 & Base 32**

I am not sure who you are, but I have an idea. Either way, I am not interested. – **Base 64 & Hex**

I am just hopeful. It would mean so much to have the title here. PI – **Base 64 & Base 32**

Do not speak to me again – **Base 64 & Hex**

How is the weather in Qatar, Mr Killah? – **Base 64 & Base 32**

Hot as always. Who is this? – **Base 64 & Hex**

I am a fan of Chess Boxing. I would love to see the title held in Korea – **Base 64 & Base 32**

We will have to see how the bid turns out. – **Base 64 & Octal**

Is there anything I could do to make your decision easier? – **Base 64 & Base 32**

No! The great nation of Qatar would never be swayed so easily. – **Base 64 & Octal**



Nor would I. We do not take kindly to this pathetic notion of bribery. – **Base 64 & Octal**

Mr Raekwon, have you spoken with Mr Razor – **Base 64 & Base 32**

I have, but I won't be bought so easily. -

Bought? Of course not. You are an official on the executive committee of the ICBA. I just want you to know that I am here to help make your decision as easy as possible. – **Base 64 & Base 32**

I would need at least 20 million Rubles – **Base 64 & Hex**

Consider it done. I will send you the information for the drop-off point soon. – **Base 32 & Base 64**

## Appendix 8:

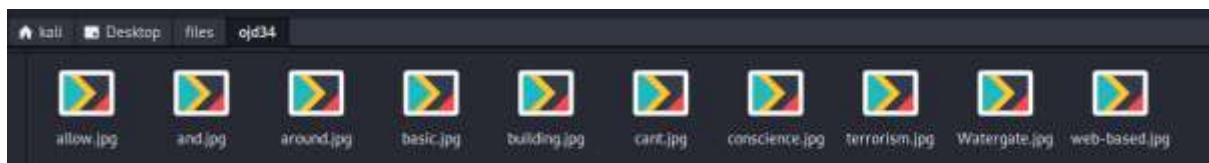


Figure 51

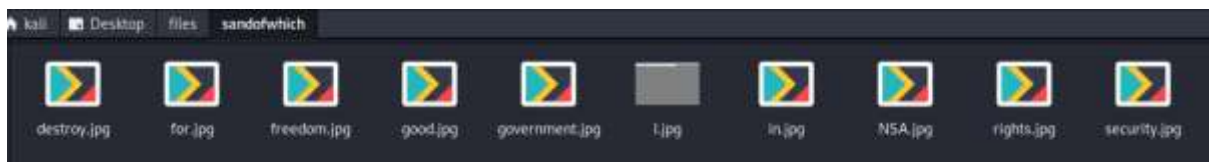


Figure 50

## Appendix 9:

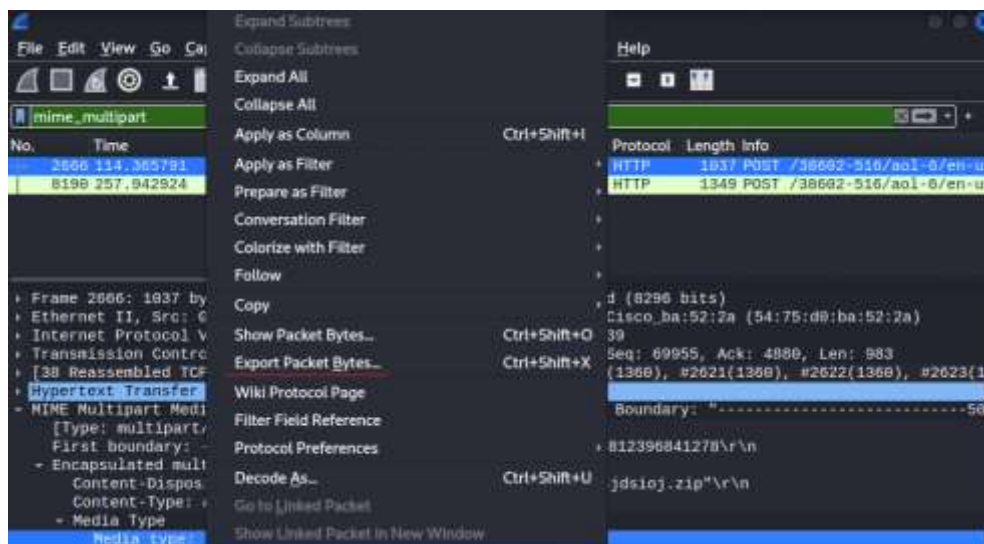


Figure 52

## Appendix 10:

cat l.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 the.jpg world.jpg with.jpg this.jpg massive.jpg surveillance.jpg machine.jpg theyre.jpg secretly.jpg building.jpg > snowden.jpg

## Appendix 11:

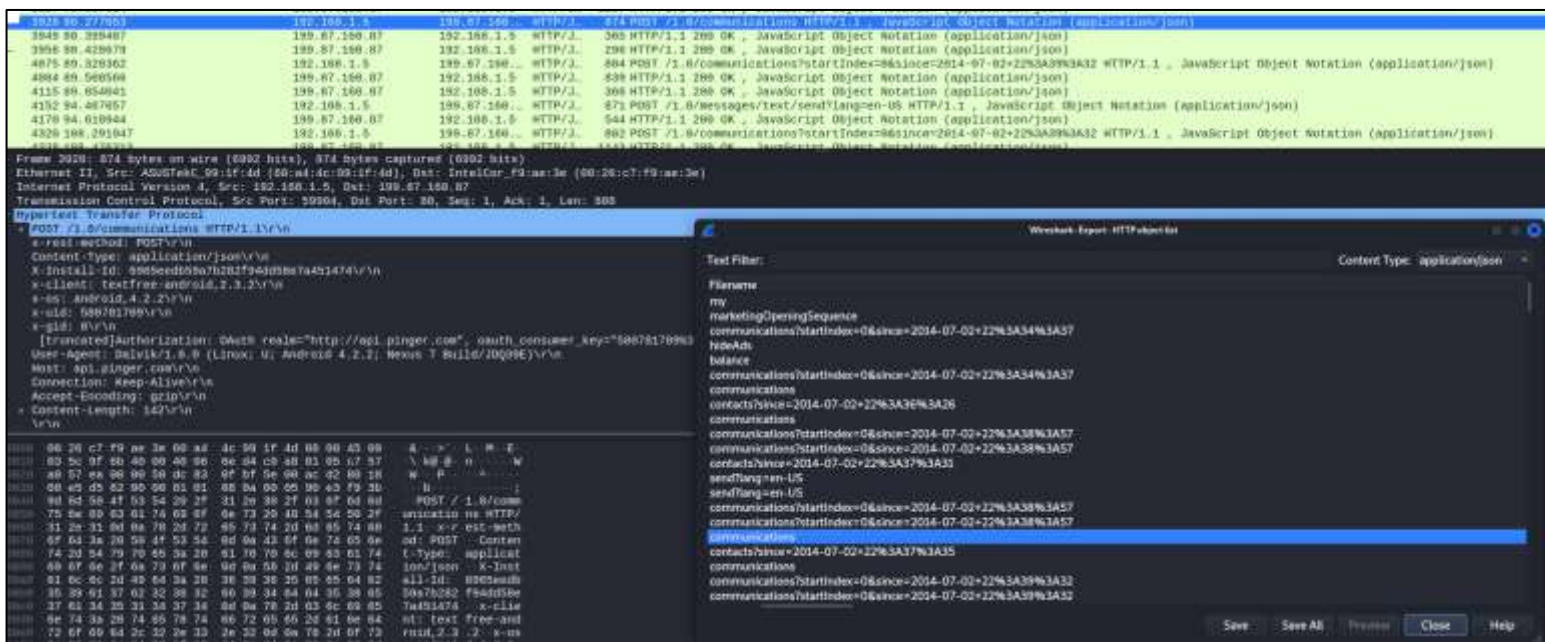


Figure 53

## Appendix 12:

Ann Dercover, 22:34:13: "this is a test"

Kim Ill-song, 22:38:55: "Good afternoon, Ann."

Ann Dercover, 22:39:15: "Who is this?"

Kim Ill-song, 22:39:31: "Castling."

Ann Dercover, 22:39:46: "where are you?"

Kim Ill-song, 22:40:05: "I know I can't tell you that"

Ann Dercover 22:41:25: "Do you know that there are people investigating Kim Ill-Song?"

"Kim Ill-son, 22:41:47: "Of course. However, they will never know it is me behind the bribes"

Ann Dercover, 22:42:54: "still we should be careful. Pay attention. I want to meet in September at 5PM."

Kim Ill-song, 22:43:06: "At our old meetup spot?"

Ann Dercover, 22:43:28: "Yes"

Kim Ill-song, 22:43:44: "What day?"

Ann Dercover, 22:50:32: "I told you to pay attention."

### Appendix 13:

| Latitude           | Longitude           |
|--------------------|---------------------|
| 46.86405563354492  | -114.00670623779297 |
| 46.864044189453125 | -114.01074981689453 |
| 46.863704681396484 | -114.01164245605469 |
| 46.856597900390625 | -114.01287078857422 |
| 46.861228942871094 | -114.01863861083984 |
| 46.864051818847656 | -114.00592803955078 |
| 46.85693359375     | -114.01863098144531 |
| 46.85727310180664  | -114.01868438720703 |
| 46.85708236694336  | -114.01225280761719 |
| 46.862701416015625 | -114.00432586669922 |
| 46.86325454711914  | -114.00360107421875 |
| 46.859046936035156 | -114.01864624023438 |
| 46.86147689819336  | -114.01863098144531 |
| 46.864051818847656 | -114.00646209716797 |
| 46.86404800415039  | -114.00680541992188 |
| 46.86103057861328  | -114.00672912597656 |
| 46.86052322387695  | -114.01863861083984 |
| 46.859466552734375 | -114.01864624023438 |
| 46.86122131347656  | -114.00647735595703 |
| 46.856319427490234 | -114.01313018798828 |
| 46.85969161987305  | -114.01864624023438 |
| 46.86370849609375  | -114.01163482666016 |
| 46.85697937011719  | -114.01237487792969 |
| 46.85979080200195  | -114.00848388671875 |
| 46.864044189453125 | -114.00694274902344 |
| 46.8577880859375   | -114.01127624511719 |
| 46.863643646240234 | -114.0035400390625  |

46.858943939208984 -114.01864624023438  
46.860843658447266 -114.00699615478516  
46.86098861694336 -114.01863098144531  
46.857234954833984 -114.01204681396484  
46.86166000366211 -114.00584411621094  
46.85914993286133 -114.01864624023438  
46.86355209350586 -114.01854705810547  
46.8637809753418 -114.01853942871094  
46.86210632324219 -114.00520324707031  
46.85980987548828 -114.01864624023438  
46.857513427734375 -114.01164245605469  
46.86037063598633 -114.0076675415039  
46.85733413696289 -114.01190948486328  
46.86405944824219 -114.00534057617188  
46.86148452758789 -114.00609588623047  
46.859500885009766 -114.00887298583984  
46.86407470703125 -114.00875854492188  
46.864078521728516 -114.00962829589844  
46.85672378540039 -114.01271057128906  
46.85884475708008 -114.01864624023438  
46.86253356933594 -114.00457763671875  
46.863983154296875 -114.00354766845703  
46.864051818847656 -114.00662231445313  
46.86408996582031 -114.01042175292969  
46.85661315917969 -114.01860809326172  
46.858829498291016 -114.00979614257813  
46.864051818847656 -114.0074691772461  
46.86387252807617 -114.0185317993164  
46.86309051513672 -114.00376892089844  
46.864051818847656 -114.00452423095703  
46.858646392822266 -114.01005554199219

46.86065673828125 -114.00727081298828  
46.858375549316406 -114.01044464111328  
46.86404037475586 -114.00392150878906  
46.864044189453125 -114.00716400146484  
46.86306381225586 -114.0185775756836  
46.863426208496094 -114.0185546875  
46.86393356323242 -114.0035171508789  
46.86408233642578 -114.0084228515625  
46.86381912231445 -114.00352478027344  
46.856834411621094 -114.01256561279297  
46.859989166259766 -114.00820922851563  
46.86405944824219 -114.00563049316406  
46.857418060302734 -114.01179504394531  
46.85795211791992 -114.01103973388672  
46.862361907958984 -114.00481414794922  
46.86405563354492 -114.00506591796875  
46.858524322509766 -114.01863861083984  
46.857181549072266 -114.01212310791016  
46.86029052734375 -114.01863098144531  
46.8590087890625 -114.0095443725586  
46.864070892333984 -114.0094223022461  
46.864051818847656 -114.00605773925781  
46.86248779296875 -114.01860046386719  
46.86406707763672 -114.00910186767578  
46.86183547973633 -114.0055923461914  
46.857601165771484 -114.01866912841797  
46.864017486572266 -114.01107025146484  
46.862281799316406 -114.01860046386719  
46.8582878112793 -114.01864624023438  
46.85749053955078 -114.01168823242188  
46.85910415649414 -114.00941467285156

46.860755920410156 -114.01863098144531  
46.86286163330078 -114.00408172607422  
46.862064361572266 -114.01861572265625  
46.864051818847656 -114.00627899169922  
46.86159896850586 -114.01863098144531  
46.85957717895508 -114.01864624023438  
46.85765838623047 -114.0114517211914  
46.864044189453125 -114.00414276123047  
46.85930252075195 -114.00914001464844  
46.86408996582031 -114.01012420654297  
46.858123779296875 -114.01079559326172  
46.864044189453125 -114.0042724609375  
46.86183547973633 -114.01862335205078  
46.85993194580078 -114.01864624023438  
46.85747146606445 -114.01171112060547  
46.86367416381836 -114.01853942871094  
46.86260223388672 -114.01859283447266  
46.858055114746094 -114.01866149902344  
46.86354446411133 -114.00354766845703  
46.85647201538086 -114.01302337646484  
46.86282730102539 -114.0185775756836  
46.85969161987305 -114.00862121582031  
46.864051818847656 -114.00477600097656  
46.86330032348633 -114.01856231689453  
46.86293411254883 -114.00396728515625  
46.86404800415039 -114.01071166992188  
46.858734130859375 -114.01864624023438



## Appendix 14:

# Convert CSV to KML

Use this tool to translate CSV into KML.

### From CSV/Excel

- CSV To Datedata
- CSV To Flat File
- CSV To GeoJSON
- CSV To HTML Table
- CSV To JSON
- CSV To KML
- CSV To Markdown
- CSV To Multi-line Data
- CSV To PDF
- CSV To SQL
- CSV To Word
- CSV To XML
- CSV To YAML
- Pivot CSV
- Transpose CSV
- Query CSV with SQL

This conversion is now available as an API at [ConvertCiv.io](#)

You must have a description and latitude and longitude information in your data and an optional altitude. You may use up to 2 fields for the Description. Planned enhancements of this tool will geocode an address field to supply the latitude and longitude.  
See also [CSV to GeoJSON](#) and [KML to CSV](#)

### Step 1: Select your input

Enter Data

Choose File

Enter URL

Enter or paste CSV here

Clear Input

Example

### To CSV/Excel

- Flat File to CSV
- GeoJSON To CSV
- HTML Links To CSV
- HTML Table To CSV
- JSON To CSV
- KML To CSV
- SQL To CSV
- XML To CSV
- YAML To CSV

### Data Tools

Figure 54