

Unit 2: Network Forensics Investigation

CMP416 Digital Forensics 2

Name: Ethan Hastie

Student ID: 1801853

University of Abertay Dundee

Abstract

Following a reported crime of corruption on an international sporting competition, an investigation has been carried out using network captures captured from the parties of interest in this case. This report contains the findings of that investigation with details such as how they were carried out and the results found in this investigation. The findings of this report indicate a suspect by the name of ‘Kim Ill-Song’ who was found to have attempted to bribe several government officials. Files were seen to be downloaded which contained valuable information such as potential aliases involved in the corruption case. Anti-forensic methods were used to hide sensitive information, with the methodology of this investigation allowing some of these to be recovered. The method used included splitting the original files into more files that matched a quote spoken by Edward Snowden. Finally, this report also details the conversation between ‘Ann Dercov’ – a player within the competition - and ‘Kim Ill-Song’ and finds details of their next meeting using extracted information.

Tools

During this investigation, multiple tools were used to gather results required by the set briefs. These were used in a forensically sound manner to avoid tampering with the integrity of the data. Hash tools were used also to verify that files were not tampered with and at the end of the investigation of each network capture the resulting hash was compared to the original and found not to have been tampered with.

- Md5sum – calculates MD5 hashes of files.
- Certutil – Windows equivalent of md5sum used to calculate md5 hashes.
- YAF (CERT NetSA Security Suite, 2021) – performed on network captures to capture flow information and preserve for analysis.
- SiLK (CERT NetSA Security Suite, 2021) – for efficient analysis of network flow data and significant in statistical flow stages of the investigation.
 - Rwtotal
 - Rwstats
 - Rwuniq
- Wireshark (Wireshark, 2021) – GUI-based network protocol analyser.
- Tshark (Wireshark, 2021) – package included with Wireshark that is used to filter PCAP files within the command line.
- CyberChef (GCHQ, 2021) – SWISS Army knife used to decode information found in PCAP files.
- Cat – useful command to view files from the Linux command line.
- Google Translate (Google, 2021)
- Crackstation.net (Crackstation, 2019) – online rainbow table.
- Grep – pipe with command line tools to filter through data.
- Excel to KML (Earth Point, 2022) – converts CSV files that hold latitude and longitude values to KML Files that can be read by Google Earth.
- Google Earth (Google, 2022) – access satellite imagery everywhere in the world and in this case view KML files.

Table of Contents

Abstract	2
Tools	3
1. Investigation of ‘Capture1.pcap’ Network Capture	6
1.1 Brief	6
1.2 Method	6
1.2.1 Hash Checksum	6
1.2.2 Statistical Flow Analysis.....	6
1.2.2 Wireshark Analysis.....	7
1.2.2.1 GoTSpoilers.docx	11
1.2.2.2 NorthKorea.docx.....	12
1.2.2.3 PiD.docx.....	13
1.2.2.4 NK.jpg.....	14
1.2.2.5 Rules1.docx.....	15
1.2.2.6 track6.docx.....	16
1.2.2.7 track10.docx.....	17
1.2.2.8 MIME HTTP Traffic.....	18
2. Investigation of ‘Capture2.pcap’ Network Capture	20
2.1 Brief	20
2.2 Method	20
2.2.1 Hash Checksum	20
2.2.2 Statistical Flow Analysis.....	20
2.2.3 Wireshark Analysis.....	21
2.2.3.1 Decoded Conversations.....	22
2.2.3.2 Members Involved and Results of Bribery	28
3. Investigation of ‘Capture3.pcap’ Network Capture	29
3.1 Brief	29
3.2 Method	29
3.2.1 Hash Checksum	29
3.2.2 Statistical Flow Analysis.....	29
3.2.3 Wireshark Analysis.....	30
3.2.3.1 Image1.jpg	38
3.2.3.2 Image2.jpg	39
3.2.3.3 Image3.jpg	39
4. Investigation of ‘Capture4.pcap’ Network Capture	41
4.1 Brief	41

4.2	Method	41
4.2.1	Wireshark Analysis.....	41
4.2.1.1	Conversations between Ann Dercov and Ill_Song	42
4.2.2.2	Determining Location of the Meeting.....	43
5.	Appendix.....	48
5.1	Capture 1	48
5.1.1	Email Conversation.....	48
5.2	Capture 2.....	48
5.2.1	Encoded Conversations Between Ill Song and Other Members	48
5.3	Capture 4.....	52
5.3.1	locationData.csv	52
	References.....	55

1. Investigation of ‘Capture1.pcap’ Network Capture

1.1 Brief

“Our intelligence has warned us of a suspected bribery in an international competition. Our suspect has downloaded files and we must find out what these files contain. We need to know what files were sent, how you were able to recover them, and whether any of the files contain the potential names/aliases of actors in this corruption case.”

1.2 Method

1.2.1 Hash Checksum

Before the capture was analysed, an MD5 checksum was recorded on the file which would help preserve the integrity of the data analysed and to prove that it was not tampered with during investigations in Figure 1 as seen below:

```
(kali㉿kali)-[~/Desktop/analysis]
$ md5sum 'Capture 1.pcap'
bae9aae7f29f88494a985cea8ff350f  Capture 1.pcap
```

Figure 1 Hash Checksum recorded using md5sum

Since software was utilised on both Kali Linux and Windows, a hash was also recorded on the Windows machine as well:

```
Command Prompt

C:\Users\ethan\Desktop\University\4th\Forensics\pcap>certutil -hashfile "Capture 1.pcap" MD5
MD5 hash of Capture 1.pcap:
bae9aae7f29f88494a985cea8ff350f
CertUtil: -hashfile command completed successfully.
```

Figure 2 Hash Checksum recorded using certutil

1.2.2 Statistical Flow Analysis

The PCAP was then converted into a ‘.YAF’ format which would allow analysis to begin using the YAF command:

```
(kali㉿kali)-[~/Desktop/analysis]
$ yaf --in 'Capture 1.pcap' --out first/capture1.yaf
```

Figure 3 Converting First Capture File into YAF Format

Viewing this file using the command line is a simple task but the great amount of data makes it unreadable so it’s wise to view some only a few entries as seen in Figure 4 below:

```
(kali㉿kali)-[~/Desktop/analysis/first]
$ cat capture1.yaf.txt | less
```

Figure 4 Viewing Entries in First Capture

Then, using SiLK, it was converted into a ‘.RW’ format to allow analysis by SiLK:

```
(kali㉿kali)-[~/Desktop/analysis/first]
$ rwipfix2silk capture1.yaf --silk-output=capture1.rw
```

Figure 5 Converting YAF File to SiLK Format

A tool under SiLK called ‘rwtotal’ was used to analyse the traffic according to the number of bytes and packets each destination port received using the command ‘*rwtotal capture1.rw –skip-zero –dport | less*’. It is also of note that information captured by statistical flow analysis can be filtered for our needs using grep for more efficient analysis. Since the suspect downloaded files, this is usually accomplished over a network service such as File Transfer Protocol (FTP) or Server Message Block (SMB). These are some examples of standard network services that allow users to download and upload files to a server. By using ‘rwtotal’, we can immediately find if a network service such as these were used and what one was employed. Using the command, this revealed http/s web traffic (port 80 and 443) and SMB traffic, which is designated by port 445:

dPort	Records	Bytes	Packets
53	126	8202	127
67	1	4264	13
80	415	10180884	12448
137	2	402	5
138	2	710	3
139	2	2932	26
161	1	13234	126
443	127	330258	2431
445	1	1413105	1336

Figure 6 ‘rwtotal’ Highlighting Web and SMB traffic (rows 4-10)

Filtering the traffic once again using ‘rwtotal’ according to the first 24 bytes of IP addresses captured (for example: 169.72.43.*) revealed many bytes in the octal range of ‘172.29.1.*’ using the following command ‘*rwtotal capture1.rw –skip-zero –sip-first-24*’:

149.174. 97	13	4056	46
149.174. 98	21	12160	110
149.174.149	2	541	7
152.163. 13	40	78656	278
157. 56. 67	6	16422	36
157. 56.106	1	2740	20
157. 56.149	1	5338	13
171.161.199	8	309414	262
172. 29. 1	705	12084573	17623
172.230.165	4	38967	47

Figure 7 ‘rwtotal’ Highlighting Traffic from ‘172.29.1.*’ Subnet

‘rwstats’ was used alongside switches that showed source IP, destination IP, source and destination port numbers that revealed large web traffic between two IP addresses 172.29.1.23 and 64.12.132.55. In addition, it also revealed traffic via SMB between two hosts on the same subnet - 172.29.1.23 and 172.29.1.20:

rwstats capture1.rw --fields=1,2,3,4 --value=packets --count=20						
INPUT: 1368 Records for 1347 Bins and 30295 Total Packets						
OUTPUT: Top 20 Bins by Packets						
sIP	dIP	sPort	dPort	Packets	%Packets	cumul_%
172.29.1.23	64.12.132.55	50180	80	6593	21.762667	21.762667
64.12.132.55	172.29.1.23	80	50180	3547	11.708203	33.470870
172.29.1.23	172.29.1.20	50291	445	1336	4.409969	37.880838
172.29.1.20	172.29.1.23	445	50291	942	3.109424	40.990262
173.194.79.103	172.29.1.20	443	1784	460	1.518402	42.508665
172.29.1.20	173.194.79.103	1784	443	308	1.016669	43.525334

Figure 8 ‘rwstats’ used to Filter Top 20 Packets According to Source IP, Destination IP, Source Port and Destination Port

1.2.2 Wireshark Analysis

First, the capture file was loaded into Wireshark. The example output provided by Wireshark in this capture file can be seen in Figure 1:

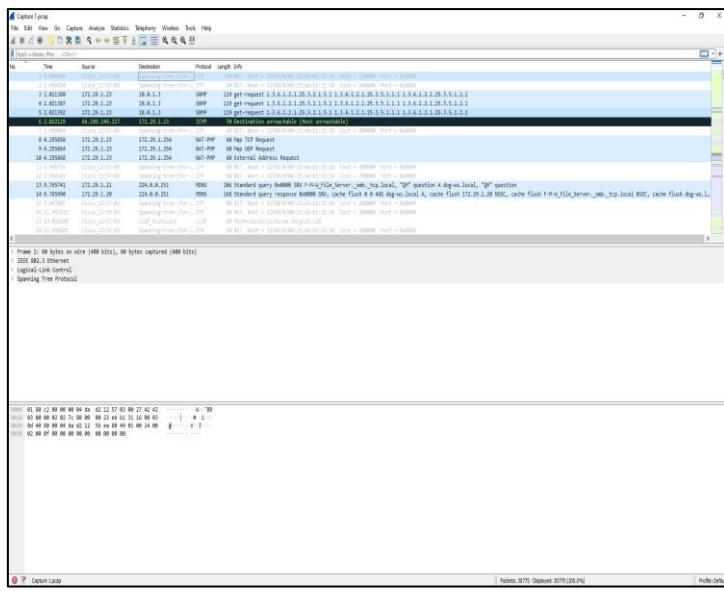


Figure 9 Example Output of Capture 1 in Wireshark

SMB was first investigated as this was the protocol used for the file transfer. Using Wireshark's filtering capabilities, the command '`ip.addr == 172.29.1.23 && smb`' was issued. This confirmed the presence of SMB traffic as highlighted in the flow analysis. Multiple files and folders in the transaction could be seen such as '\Documents.zip' and 'DOCUME~1.zip':

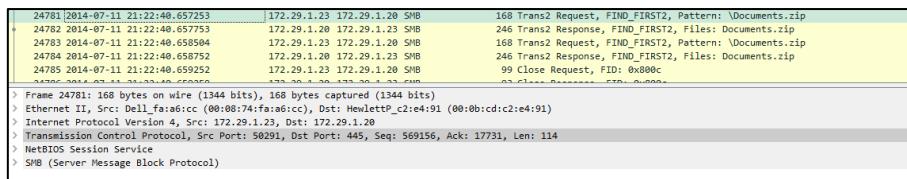


Figure 10 Presence of a 'Documents.zip'

In frame 23872 onwards, multiple folders can be seen that belong to the server hosting the SMB share:

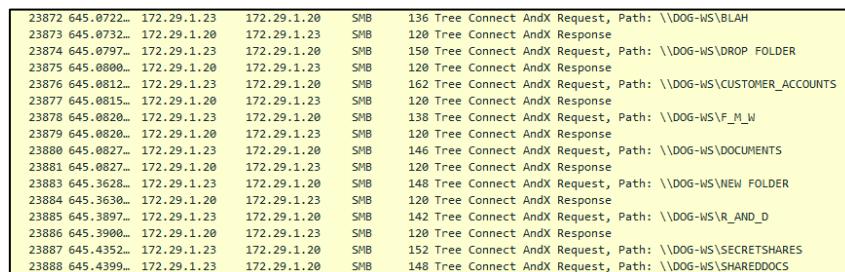


Figure 11 Folders seen on the SMB Share

Other files can be witnessed such as two JPEG files and a file titled 'premium-customer-billing-list.xls':

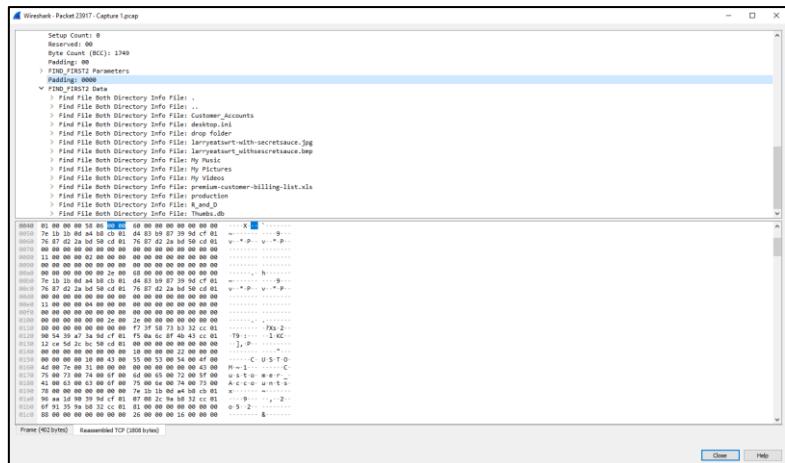


Figure 12 Other Files and Folders seen via SMB

The presence of these hosts meant that their MAC addresses could be identified as shown in the table below:

Table 1 IP Address and MAC Addresses involved in SMB Transaction

IP Address	MAC Address
172.29.1.23	00:08:74:fa:a6:cc
172.291.20	00:0b:cd:c2:e4:91

Using Wireshark's export objects feature, some files were recovered such as Documents.zip. This can be seen below:

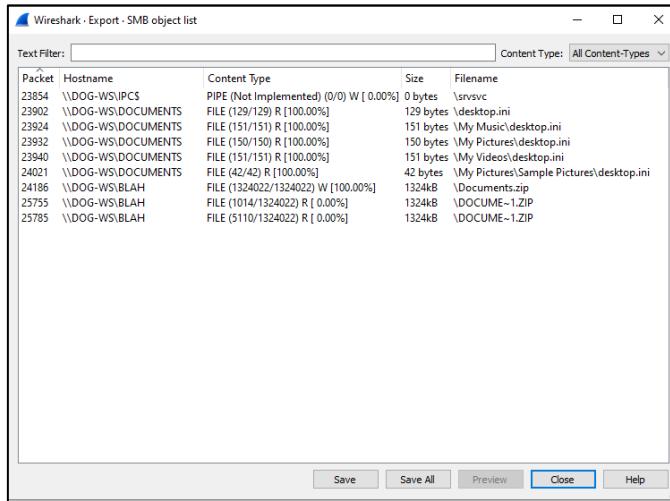


Figure 13 Using Wireshark's Export Feature to Obtain Some of the Recorded Files – Documents.zip, DOCUME~1.zip

Once extracted, these reveal folders such as ‘Actual Documents’, ‘Chess Boxing’, ‘Enter the WuTang’, ‘More Documents’ and a zipped folder called ‘untitled folder.zip’:

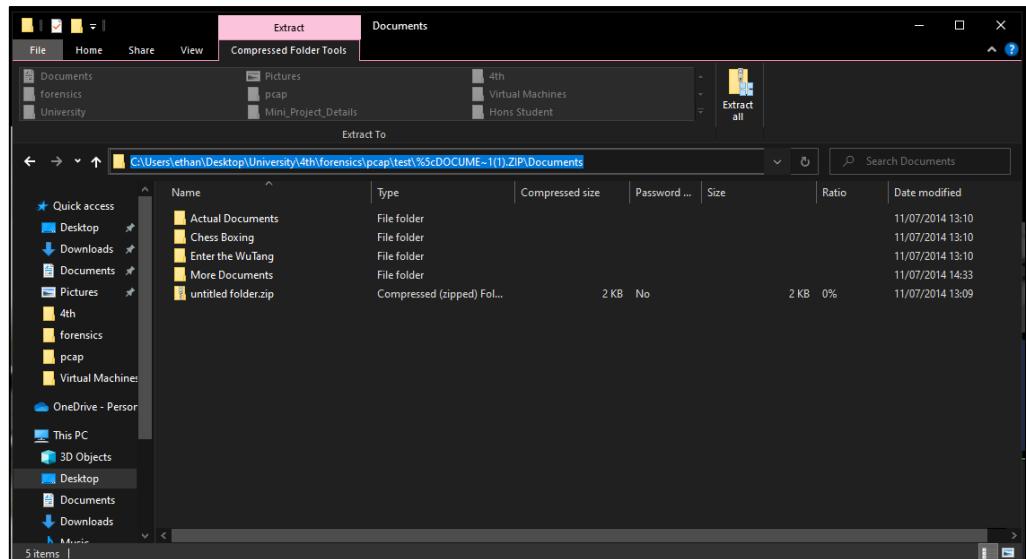


Figure 14 Contents of 'Documents'

Three files were found in ‘Actual Documents’ titled ‘GoTSpoilers.docx’, ‘NorthKorea.docx’ and ‘PiD.docx’:

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
GoTSpoilers.docx	Microsoft Word Document	64 KB	No	73 KB	13%	19/06/2014 12:29
NorthKorea.docx	Microsoft Word Document	56 KB	No	66 KB	15%	19/06/2014 12:30
PiD.docx	Microsoft Word Document	336 KB	No	344 KB	3%	19/06/2014 12:32

Figure 15 Contents of 'Actual Documents'

Eight files were found in ‘Chess Boxing’ which can be seen below:

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
NK.jpg	JPG File			24 KB	23%	18/06/2014 16:51
Rules 1.docx	Microsoft Word Document	173 KB	No	185 KB	7%	19/06/2014 12:37
Rules 2.docx	Microsoft Word Document	26 KB	No	35 KB	27%	19/06/2014 12:36
Rules 3.docx	Microsoft Word Document	40 KB	No	50 KB	19%	19/06/2014 12:35
Rules 4.docx	Microsoft Word Document	48 KB	No	58 KB	17%	19/06/2014 12:34
Rules 5.docx	Microsoft Word Document	170 KB	No	181 KB	7%	19/06/2014 12:34
Rules 6.docx	Microsoft Word Document	51 KB	No	61 KB	17%	19/06/2014 12:33
Rules 7.docx	Microsoft Word Document	103 KB	No	113 KB	9%	19/06/2014 12:33

Figure 16 Contents of 'Chess Boxing'

Within the DOCUME~1.zip, under ‘Documents’ two files were found:

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
track6.docx	Microsoft Word Document	31 KB	No	40 KB	24%	19/06/2014 12:24
track10.docx	Microsoft Word Document	169 KB	No	180 KB	7%	19/06/2014 12:24

Figure 17 Contents of Other 'Documents'

Within the same ZIP, under ‘More Documents’, two files:

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
BillOfRights.txt	Text Document	7 KB	No	22 KB	72%	18/06/2014 16:51
NorthKorea.jpeg	JPEG File	4 KB	No	5 KB	25%	11/07/2014 14:32

Figure 18 Contents of 'More Documents'

Silent Eye folder was contained in many untitled folders which contained no data inside:

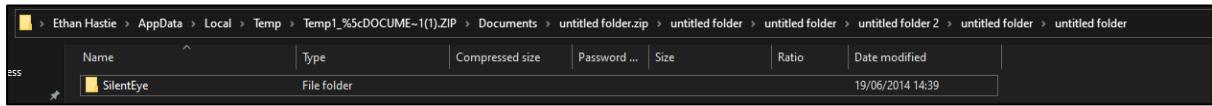


Figure 19 'SilentEye' Folder which Contained no Data

1.2.2.1 GoTSpoilers.docx

Opening this file revealed a Base64 encoded phrase within the word document. This was easily identified because Base64 encoded strings usually end in an equal's sign:

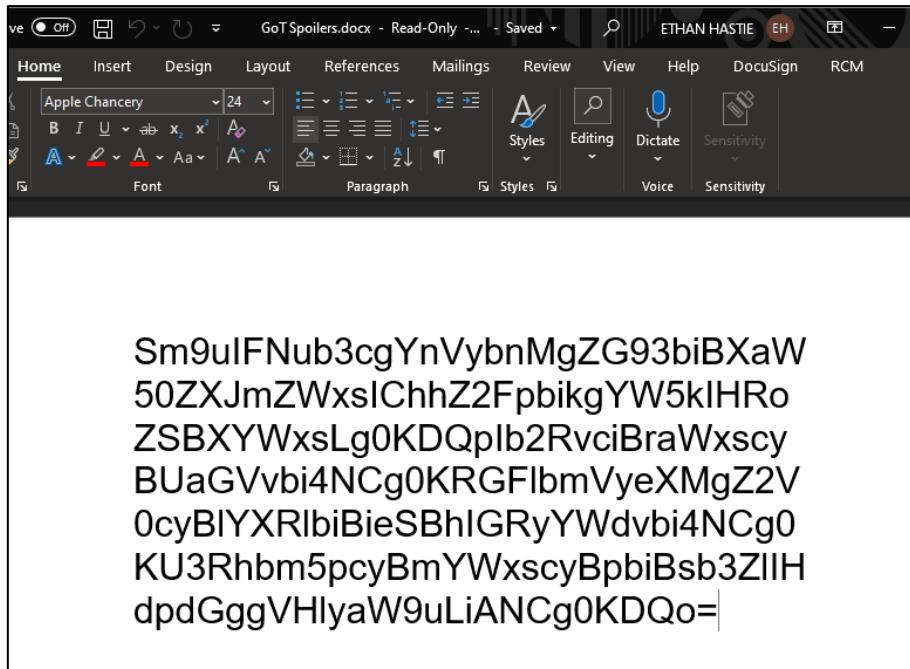


Figure 20 Base64 Encoded Message inside GoT Spoilers.docx

Using CyberChef, this was converted into ASCII readable text and revealed some horrific spoilers for Game of Thrones:

Figure 21 Translation of GoT Spoilers.docx

1.2.2.2 NorthKorea.docx

This was also Base64 encoded. Its original output can be seen below:

Figure 22 NorthKorea.docx

Using CyberChef, this was decoded which revealed a Russian message:

The screenshot shows a Base64 decoding interface. In the top left, it says "From Base64". Below that is a dropdown menu set to "Alphabet A-Za-zA-Z0-9+/=". There is also a checked checkbox labeled "Remove non-alphabet chars". The main text area contains a long Base64 encoded string. At the bottom, there is an "Output" section with a timestamp of "time: 1ms", a length of "348", and "lines: 7". Below this, there are two paragraphs of Russian text:

Для кого это может касаться:
Я был свидетелем, что Ким Чен Ун и правительство Северной Кореи разработали программу, которая позволяет им путешествовать во времени. С использованием этой технологии, я считаю, что они намерены двигаться вперед и изменить результаты войны в Корее.
Пожалуйста, Оби-Ван, ты моя единственная надежда.

Figure 23 Decoded Message

Using Google Translate, this was detected as Russian. Its message can be seen below:

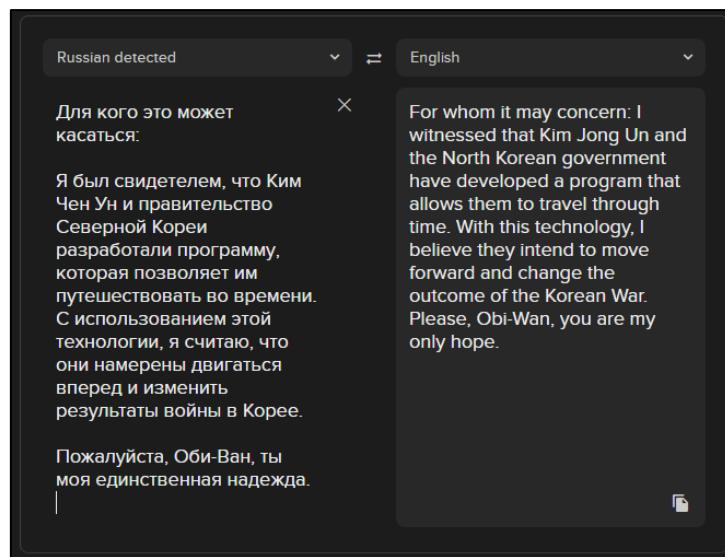


Figure 24 Message Translated from Russian to English

1.2.2.3 PiD.docx

Opening the file revealed an image of Paul McCartney along with a Base64 encoded message:

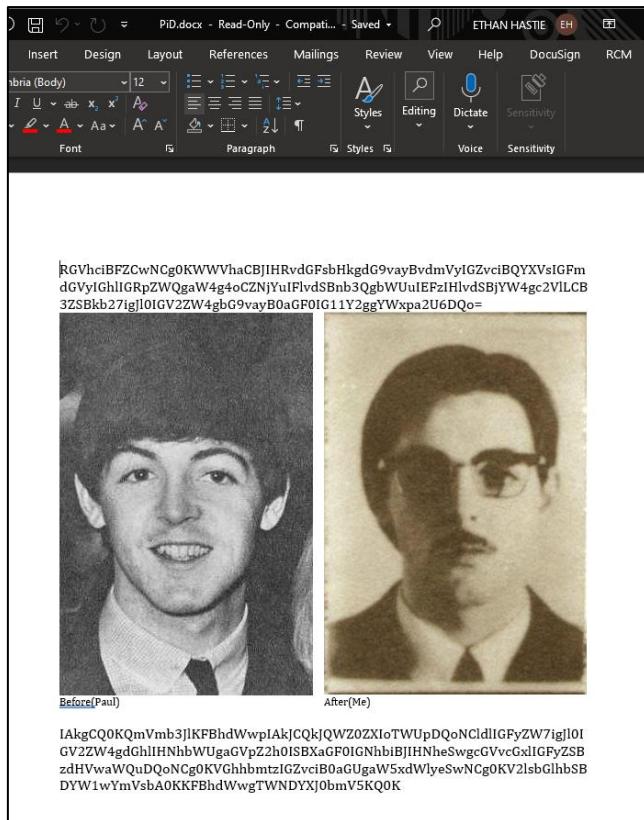


Figure 25 Contents of PiD.docx

This, translated reveals a message to ‘Ed’:

The image shows the BAKE! application interface. On the left, there is a 'From Base64' section with an 'Alphabet' dropdown set to 'A-Za-z0-9+='. A checkbox for 'Remove non-alphabet chars' is checked. The main area displays the decoded message. At the bottom, there is an 'Output' section with a text area containing the message, a file browser icon, and a 'Send' button. The message reads:

```

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

```

Below the message are two small images labeled 'Before(Paul)' and 'After(Me)'. The message continues:

```

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

Thanks for the inquiry,
William Campbell
(Paul McCartney)

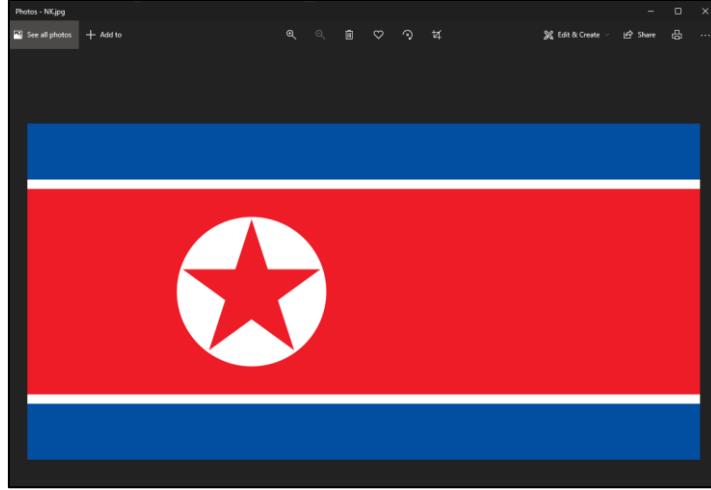
```

At the bottom left are buttons for 'STEP', 'BAKE!', and 'Auto Bake'.

Figure 26 A Message to Ed from William Campbell

1.2.2.4 NK.jpg

A picture of the North Korean flag within a JPEG file:



1.2.2.5 Rules1.docx

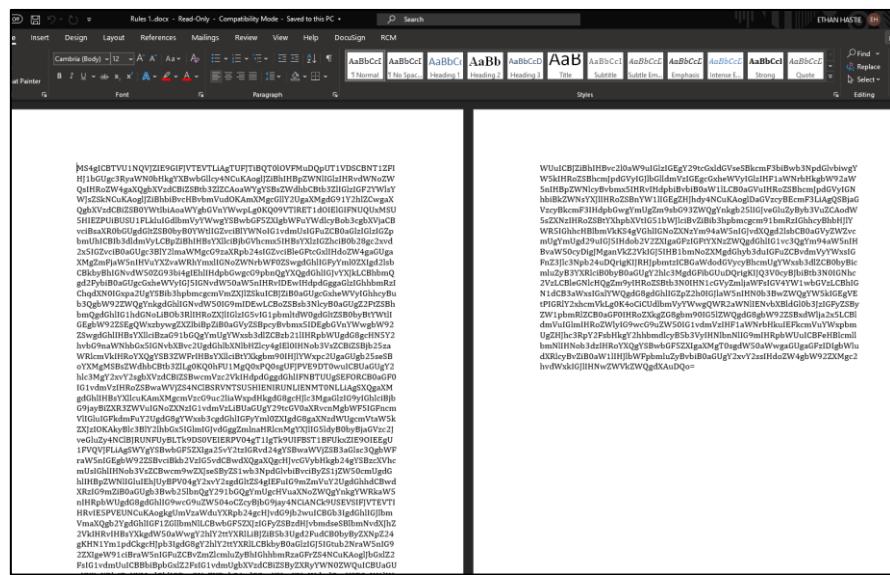


Figure 27 Rules1.docx

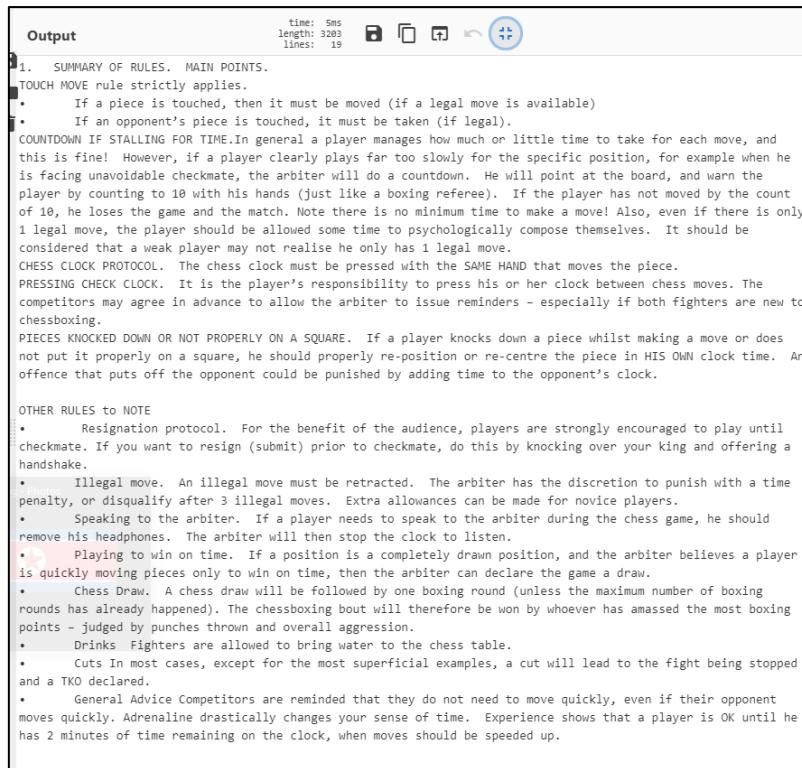


Figure 28 Rules1.docx Translated from Base64

Figure 28 reveals rules from the chess competition.

1.2.2.6 track6.docx

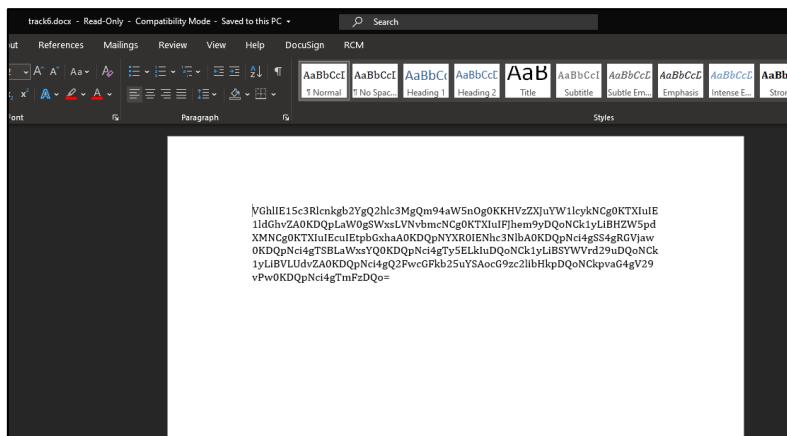


Figure 29 Contents of track6.docx

Output
The Mystery of Chess Boxing: (usernames)
Mr. Method
Kim Ill-Song
Mr. Razor
Mr. Genius
Mr. G. Killah
Matt Cassel
Mr. I. Deck
Mr. M Killa
Mr. O.D.B.
Mr. Raekwon
Mr. U-God
Mr. Cappadonna (possibly)
John Woo?
Mr. Nas

Figure 30 Potential Members Involved

This, translated from Rules6.docx, reveals several members including Mr. Method, Kim Ill-Song, and a possible Mr. Cappadonna and John Woo.

1.2.2.7 track10.docx

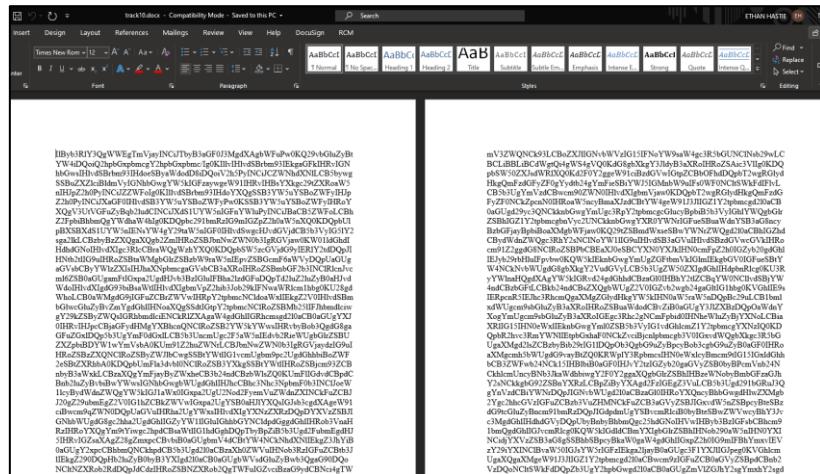


Figure 31 Contents of track10.docx

```

"Protect Ya Neck"
"So what's up man?
Cooling man"
"Chilling chilling?"
"Yo you know I had to call, you know why right?"
"Why?"
"Because, yo, I never ever call and ask, you to play something right?"
"Yeah"
"You know what I wanna hear right?"
"What you wanna hear?
I wanna hear that Wu-Tang joint"
"Wu-Tang again?"
"Ah yeah, again and again!"

[sounds of fighting]

[RZA] Wu-Tang Clan coming at you, protect your neck kid, so set it off the Inspector Deck
[Meth] watch your step kid [8X]

[Inspector Deck]
I smoke on the mic like smoking Joe Frazier
The hell raiser, raising hell with the flavor
Terrorize the jam like troops in Pakistan
Swinging through your town like your neighborhood Spiderman
So uh, tic toc and keep ticking
While I get you flipping off the shit I'm kicking
The Lone Ranger, code red, danger!
Deep in the dark with the art to rip charts apart
The vandal, too hot to handle
you battle, you're saying Goodbye like Tevin Campbell
Roughneck, Inspector Deck's on the set
The rebel, I make more noise than heavy metal

```

Figure 32 Base64 translated of track10.docx

1.2.2.8 MIME HTTP Traffic

Earlier, it was found there was heavy Multi-Purpose Internet Mail Extensions (MIME) traffic between 172.29.1.23 and 64.12.132.55. This kind of traffic allows users to exchange data over the Internet such as images. It is an extension of the internet email protocol (Jithin, 2016). This is confirmed in Wireshark using Protocol Hierarchy Statistics, which is only specifically displayed when filtering for HTTP traffic:

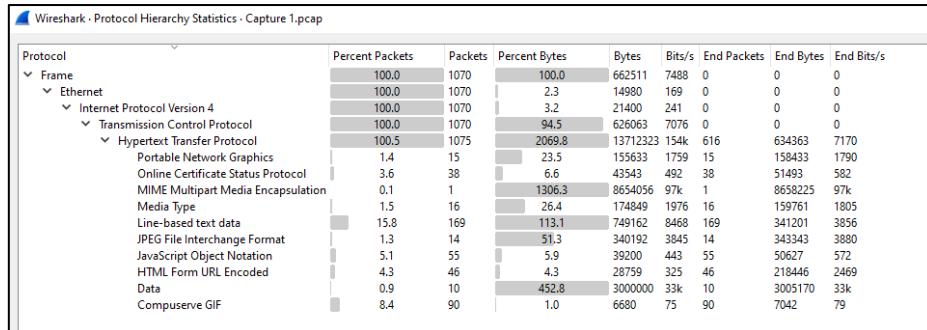


Figure 33 Protocol Hierarchy Statistics Revealing MIME Traffic

Filtering for MIME traffic reveals frame 17979 as MIME traffic. The email content includes a sender of ‘edward@aol.com’ and a recipient of ‘wikiofleaks@aol.com’. It seems to be discussing PCAP files, a singular file, that could be ‘useful’. In the attachment of the email is a single ZIP file called ‘DocsPcap.zip’:

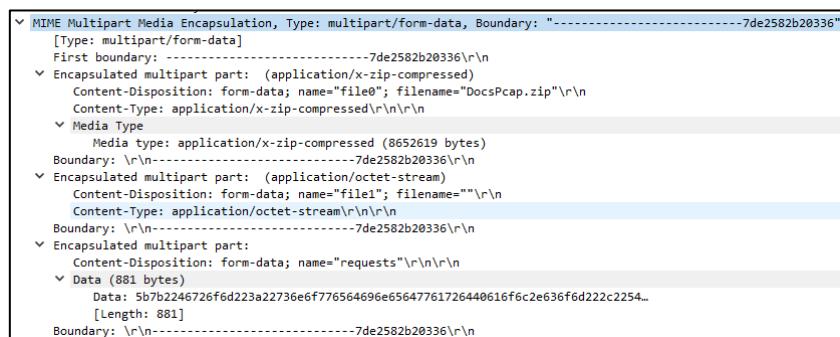


Figure 34 Metadata Showing ‘DocsPcap.zip’ In Attachment

The IP and MAC addresses of the sender and recipient were both captured which can be seen in the table below:

Table 2 IP and MAC Addresses

IP Address	MAC Address
172.29.1.23	00:08:74:fa:a6:cc
64.12.132.55	54:75:d0:ba:52:2a

2. Investigation of ‘Capture2.pcap’ Network Capture

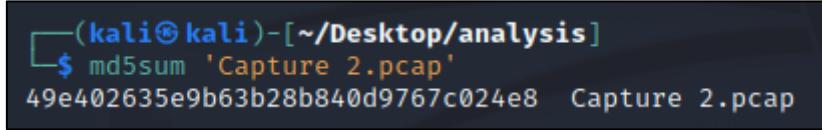
2.1 Brief

“IRC monitors have picked up encrypted traffic of suspected corrupt officials speaking with a foreign national that goes by the alias of “Ill-Song”. Decode the conversations in this capture and provide us with a summary of where the officials contacted by Ill-Song are located (countries) and whether the officials are innocent or guilty of bribery.”

2.2 Method

2.2.1 Hash Checksum

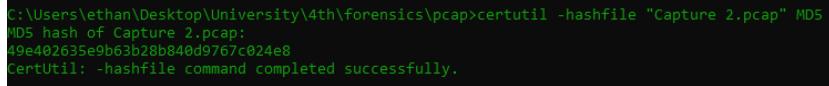
A hash checksum was recorded on the second capture like the previous file:



```
(kali㉿kali)-[~/Desktop/analysis]
$ md5sum 'Capture 2.pcap'
49e402635e9b63b28b840d9767c024e8 Capture 2.pcap
```

Figure 35 MD5 Hash on Second PCAP File

The same was performed on the Windows machine using certutil:

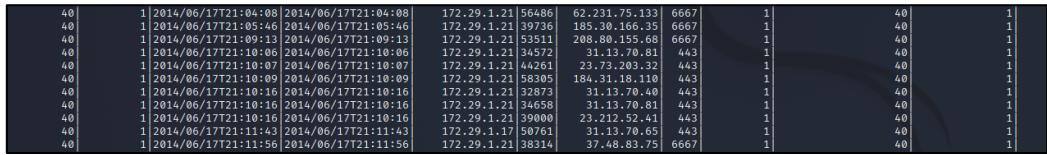


```
C:\Users\ethan\Desktop\University\4th\forensics\pcap>certutil -hashfile "Capture 2.pcap" MD5
MD5 hash of Capture 2.pcap:
49e402635e9b63b28b840d9767c024e8
CertUtil: -hashfile command completed successfully.
```

Figure 36 MD5 Hash on Second PCAP File Using Windows Tool

2.2.2 Statistical Flow Analysis

Using statistical flow analysis, the capture was filtered for Internet Relay Chat (IRC) traffic. Because of the nature of the brief, it was straight forward to find out what protocol was used, which also allowed for more efficient analysis. Users connect to a server using specialised software, such as an IRC client, that allow users to chat to each other. They communicate in channels, but private messaging is also a function too. Using ‘rwuniq’, the file was analysed for mentions of the main IRC port 6667 using the command ‘rwuniq capture2.rw –fields=byutes,packets,stime,etime,sip,sp,dip,dp –values=records,bytes,packets –sort-output’:



40	1	2014/06/17T21:04:08	2014/06/17T21:04:08	172.29.1.21	56486	62.231.75.133	6667	1	40	1
40	1	2014/06/17T21:05:46	2014/06/17T21:05:46	172.29.1.21	39736	185.30.166.35	6667	1	40	1
40	1	2014/06/17T21:09:13	2014/06/17T21:09:13	172.29.1.21	53511	208.80.155.68	6667	1	40	1
40	1	2014/06/17T21:10:06	2014/06/17T21:10:06	172.29.1.21	34572	31.13.70.81	443	1	40	1
40	1	2014/06/17T21:10:07	2014/06/17T21:10:07	172.29.1.21	44261	23.23.10.10	443	1	40	1
40	1	2014/06/17T21:10:09	2014/06/17T21:10:09	172.29.1.21	32005	184.51.15.110	443	1	40	1
40	1	2014/06/17T21:10:16	2014/06/17T21:10:16	172.29.1.21	32873	31.13.70.46	443	1	40	1
40	1	2014/06/17T21:10:16	2014/06/17T21:10:16	172.29.1.21	34658	31.13.70.81	443	1	40	1
40	1	2014/06/17T21:10:16	2014/06/17T21:10:16	172.29.1.21	39000	23.212.52.41	443	1	40	1
40	1	2014/06/17T21:11:43	2014/06/17T21:11:43	172.29.1.17	50761	31.13.70.65	443	1	40	1
40	1	2014/06/17T21:11:56	2014/06/17T21:11:56	172.29.1.21	38314	37.48.83.75	6667	1	40	1

Figure 37 Top Three Rows Shown by rwuniq that Capture IRC Traffic

Figure 37 confirmed the presence of IRC chat and the output here provided information about three hosts running IRC. For a more concise view, a different command was utilised: namely, rwstats. The following command was used to analyse the file, which revealed the top 30 highest network packets in communication - ‘rwstats capture2.rw –fields=sip,sp,dip,dp –values=packets,bytes –count=30’:

(kali㉿kali)-[~/Desktop/analysis/second]							
\$ rwstats capture2_rw --fields=sip,sp,dip,dp --values=packets,bytes --count=30							
INPUT: 2694 Records for 2629 Bins and 25561 Total Packets							
OUTPUT: Top 30 Bins by Packets							
sIP	sPort	dIP	dPort	packets	Bytes	%packets	cumul_%
74.125.239.101	80	172.29.1.17	50681	434	605352	1.697899	1.697899
198.41.247.149	80	172.29.1.17	50596	419	586626	1.639216	3.337115
23.67.247.106	80	172.29.1.21	51437	402	560535	1.572708	4.909824
172.29.1.21	51437	23.67.247.106	80	384	28093	1.502289	6.412112
198.41.247.149	80	172.29.1.17	50597	371	519954	1.451430	7.863542
172.29.1.17	50681	74.125.239.101	80	304	23624	1.189312	9.052854
74.125.28.147	443	172.29.1.21	47034	287	294728	1.122804	10.175658
96.17.148.18	80	172.29.1.21	34042	286	344119	1.118892	11.294550
172.29.1.21	47034	74.125.28.147	443	286	66675	1.118892	12.413442
162.159.242.98	80	172.29.1.21	57627	281	388172	1.099331	13.512773
74.125.239.46	80	172.29.1.21	38036	264	369567	1.032823	14.545597
172.29.1.17	50596	198.41.247.149	80	263	13352	1.028911	15.574508
172.29.1.17	50588	185.30.166.35	6667	256	17442	1.001526	16.576034
162.159.242.98	80	172.29.1.21	57625	254	351964	0.993701	17.569735
162.159.242.98	80	172.29.1.21	57604	241	320605	0.942843	18.512578
172.29.1.21	57627	162.159.242.98	80	240	15860	0.938930	19.451508
162.159.242.98	80	172.29.1.21	57603	237	295914	0.927194	20.378702
172.29.1.21	57625	162.159.242.98	80	236	13692	0.923282	21.301983
172.29.1.17	50597	198.41.247.149	80	232	11861	0.907633	22.209616
162.159.242.98	80	172.29.1.21	57630	228	313187	0.891984	23.101600
172.29.1.21	34042	96.17.148.18	80	225	62939	0.880247	23.981847
172.29.1.21	57604	162.159.242.98	80	223	13063	0.872423	24.854270
96.17.148.18	80	172.29.1.21	34039	222	278865	0.868511	25.722781
172.29.1.21	34169	96.17.148.18	80	220	29532	0.860686	26.583467
96.17.148.18	80	172.29.1.21	34169	214	288720	0.837213	27.420680
172.29.1.21	38036	74.125.239.46	80	213	13820	0.833301	28.253981
172.29.1.21	57603	162.159.242.98	80	208	17334	0.813740	29.067720
172.29.1.21	34039	96.17.148.18	80	190	29499	0.743320	29.811040
74.125.28.147	443	172.29.1.17	50685	181	239941	0.708110	30.519150
148.251.38.134	80	172.29.1.21	57366	179	246817	0.700286	31.219436

Figure 38 Output of rwstats to Identify IRC Traffic

Figure 38 shows one entry between 172.29.1.17 and 185.30.166.35 revealed IRC traffic. Using the ‘rwsort’ utility, the file was sorted by source and destination IP addresses, alongside source port and destination ports, to a more readable format. This revealed another transaction between 172.29.1.21 and 37.48.83.75:

(kali㉿kali)-[~/Desktop/analysis/second]						
\$ rwstats outfile.rw --fields=1,3,2,4 --count=20						
INPUT: 2694 Records for 2629 Bins and 2694 Total Records						
OUTPUT: Top 20 Bins by Records						
sIP	sPort	dIP	dPort	Records	%Records	cumul_%
172.29.1.21	135	172.29.1.17	50669	3	0.111359	0.111359
172.29.1.21	135	172.29.1.17	50670	3	0.111359	0.222717
172.29.1.17	50669	172.29.1.21	135	3	0.111359	0.334076
172.29.1.17	50670	172.29.1.21	135	3	0.111359	0.445434
172.29.1.17	50758	31.13.70.65	443	2	0.074239	0.519673
4.2.2.1	53	172.29.1.17	59360	2	0.074239	0.593912
172.29.1.17	50639	96.17.148.51	443	2	0.074239	0.668151
172.29.1.21	34658	31.13.70.81	443	2	0.074239	0.742390
4.2.2.1	53	172.29.1.17	64379	2	0.074239	0.816630
172.29.1.17	50633	31.13.70.81	443	2	0.074239	0.890869
172.29.1.17	52668	4.2.2.1	53	2	0.074239	0.965108
172.29.1.21	59720	93.184.216.146	443	2	0.074239	1.039347
172.29.1.17	50637	96.17.148.51	443	2	0.074239	1.113586
172.29.1.21	34572	31.13.70.81	443	2	0.074239	1.187825
31.13.70.65	443	172.29.1.17	50759	2	0.074239	1.262064
172.29.1.17	50642	96.17.148.51	443	2	0.074239	1.336303
172.29.1.21	32873	31.13.70.40	443	2	0.074239	1.410542
172.29.1.21	38314	37.48.83.75	6667	2	0.074239	1.484781
0.0.0.0	68	255.255.255.255	67	2	0.074239	1.559020
172.29.1.17	50641	96.17.148.51	443	2	0.074239	1.633259

Figure 39 Confirmation of IRC Traffic

2.2.3 Wireshark Analysis

Simply typing ‘irc’ in the Wireshark search filter allowed it to be identified. Example output is the conversations that took place using IRC. This is encoded in a Base64 format. A sample of the conversation can be seen below:

```

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Razor1
:Sl2zaQzRJQ1NNRjVHnJrSTUVCRVNBWOrUJTWFZM0pPU1NXSUIEQk1KWFhLNUBT1JVR0tJRFPSlhyRzRERk1OMkNBmzNHRUiYr1FaSkFJT1VHSzQzVEVCQkcz
NRKRT1pUUE1M1BPs1dSuLEvUSGMkd2WkpBTU5VvZyTE9NMF1STNaQuTCNCf2M1RIUEzRVzRaWk8=
PING LAG2311957802
:verne.freenode.net PONG verne.freenode.net :LAG2311957802
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1:~malware@216.14.247.46 PRIVMSG Ill_Song :
NTc2NTZjMyMDc0MjgNTIwMjQ2NTYzIjk3MzY5MjY2ZTiwNjk3MzIwNmU2Zj0MjA2NjY5NmU2MTzjMjA3OTY1NzQyZQ==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2341989052
PRIVMSG Razor1
:s010VzYzEHQR1FXNfpqAU5Gw1NBwVRGTUYyWEkyTeDpVldDQTVESU5Gw1BNBURNKT1ZTU0Ez0dFqjRXS11MU0ZZUUZBwkxTTkJRWE8kfQR1hYS01EW41M1dZ
WkjbT1jV1dskFPv1hQTvU5k90VhJSURCT1pTQoFAT1PQInYRT1Mjk5a1l1LSURYT1kRhe1lQmNIVlpSU10TE41lkdlUUbTkJRWEdJRFVONFFH1pUR01wW
kM0PT09
:verne.freenode.net PONG verne.freenode.net :LAG2341989052
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1:~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyhDyXmQyMDYxMjA3NjY1NzI30T1wMj13NtCnZckyMDzNjE2ZTjMjA2MjC1NzQyMDcwNjU3MjY4NjE3MDczMjA0OT1wNjM2ZjC1NmM2hD1wNjI2NT1wNzA2N
TcyNz13NTYxhjQ2NTY0MjA3NDzWmYjA3NjY5NzN20Tc0MlyDUd1zJu2NT1wNjk2NjIwNTA30TzNmU2NzC5NjE2ZTY3MjA20TczMjA3NDY4NjUyMDcyNjk2NzY4Nz
QyMDcwNlM2MhYzNjUyNDY2NmY3hjIwzQ2D0Y1MjA1NzZmNzI2YzY0MjA1NDY5NzQ2YzY1Mu=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2371989052
:verne.freenode.net PONG verne.freenode.net :LAG2371989052
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Razor1
:s01jHEuyREjPQ1pTQTNJUE9hWENBuRQzTRR0NZFBPvJJDQVNkQ9U01c0WkjBUEZYEtJREJFQ1RXU1pUVg0UZHmzNOTVyyR1EyTE9NNFF1STNaQU01u1h
SURATJuBEM1ZPUVfHN1psQ9SVsUdLSUNETKySFNjRFBmWFFWTMzV01vUuDD1mFRU1yR0MjM9ZfQjRNjVHU8VCfHfPM1jBT1pRV0dZTFVORlxNE1eve41v
1dLNTNJ7V2a0HUT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1:~malware@216.14.247.46 PRIVMSG Ill_Song :
NTM2ZjKuNjU3NzY4NjU3Mj1MjA2NtC4NzA2NTZInM20TcNjUyYzIwNDkyMDY4NjY3MDY1MmUyNa==_
PING LAG2402004677
:verne.freenode.net PONG verne.freenode.net :LAG2402004677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Razor1 :RIU9PT09PT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2432004677
:verne.freenode.net PONG verne.freenode.net :LAG2432004677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1:~malware@216.14.247.46 PRIVMSG Ill_Song :Mzk=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2462020302
:verne.freenode.net PONG verne.freenode.net :LAG2462020302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Razor1 :RzQ9PT09PT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2492176552

```

Figure 40 Sample Encoded Conversations of IRC Traffic

These conversations were then decoded, and IP Address and MAC Address data can be seen below:

Table 3 IP Address and MAC Address of Hosts Captured in IRC Traffic

IP Address	MAC Address
172.29.1.17	00:08:74:fa:a6:cc
172.29.1.21	50:e5:49:e4:8b:d3
62.231.75.133	54:75:d0:ba:52:2a
185.30.166.35	54:75:d0:ba:52:2a
208.80.155.68	54:75:d0:ba:52:2a
37:48:83.75	54:75:d0:ba:52:2a
62.231.75.133	54:75:d0:ba:52:2a

2.2.3.1 Decoded Conversations

The conversations were decoded, and the main encoding algorithms used included Base64, Base32 and Octal. There was one instance where Crackstation had to be used to crack and MD5 hash. This turned out to be ‘Caracas’. The member who goes by the alias of ‘Raekwon’ was contacted by ‘Ill_Song’ and said he would accept a bribe of at least 20 million Rubles:

Table 4 Decoded Conversations of IRC Traffic

Sender	Receiver	Message	Translated	Encoding Used
Ill_Song	Razor1	Sl2zaQzRJQ1NNRjVH NjRSTUVCRVNBW UxORUJTWFZM0p PUINXSUIEQk1KWF hLNUBT1JVR0tJRF FPSlhYRzRERk1OM kNBmzNHRUiYr1Fa SkFJTIVHSzQzVEVC Qkc2NkRKT1pUU0E1	Mr. Razor, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang.	Base64, Base32

		M1BPSldHSUIEVU5 GMkdZWkpBTU5YV zIyTE9NNFFISTNaQ UtCNFc2M1RIUEZR VzRaWk8=		
Razor1	III_Song	NTc2NTZjNmMyMD c0Njg2NTIwNjQ2NT YzNjk3MzY5NmY2Z TIwNjk3MzlwNmU2 Zjc0MjA2NjY5NmU2 MTZjMjA3OTY1NzQ yZQ==	Well the decision is not final yet.	Base64
III_Song	Razor1	S0I0VzYzVEhQRIFX NFpaQU5GWINBVW RGTUYyWEkyTEdP VldDQTVESU5GW1 NBNURKTIZTU0Ez ModFQjRXS1IMU0Z ZUUZBWkxTTkJRW EE0WkFQRlhYS0IE WE41MldZWkJBTIJ VV1daSkFPUIhTQTV USk9OVVhJSURCTI pTQ0FaTFIPQINYRT JMRk5aUlDSLURYTk JRWEIJQ0NNVlpYS UIDTE41WkdLWUp BTkJRWEdJRFVONF FHNlpUR01WWkM0 PT09	Pyongyang is beautiful this time of year. Perhaps you would like to visit and experience what Best Korea has to offer.	Base64, Base32
Razor1	III_Song	NDkyMDYxNmQyM DYxMjA3NjY1NzI3O TIwNji3NTczNzkyM DZkNjE2ZTJjMjA2M jc1NzQyMDcwNjU3 MjY4NjE3MDczMjA 00TIwNjm2Zjc1Nm M2NDIwNjI2NTIwNz A2NTcyNzM3NTYx NjQ2NTY0MjA3NDZ mMjA3NjY5NmZ2OT c0MmUyMDUzNjU2 NTIwNjk2NjIwNTA3 OTZmNmU2Nzc5NjE 2ZTY3MjA2OTczMj A3NDY4NjUyMDcy Njk2NzY4NzQyMDc wNmM2MTYzNjUy MDY2NmY3MjIwNz Q2ODY1MjA1NzZm NzI2YzY0MjA1NDY 5NzQ2YzY1MmU=	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.	Base64
III_Song	Razor1	S0JTWEUyREJPQlpT QTNUEU9RWENBU 0RQTzRRR0NZVFBB VjJDQVNKQU9OU1 c0WkJBUEZYWEtJR EJFQIRXU1pUVUg0 UUZHMzNOTVYyR1 EyTE9NNFFISTNaQ U01U1hJSURaTjUyU 0EzM1ZPUVFHNlpS QU9SVUdLSUNETk YySFNJRBFBNWVFF WTMzV01VUUdDM 1RFRUIyR0MyM0ZF QjRXNjVMU0VCWF hPM1JBT1pRV0dZTF VORlhXNEIEVE41V 1dLNTNJTVzaR0tM UT0=	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.	Base64, Base32
III_Song	Razor1	R1U9PT09PT0=	5	Base64, Base32
Razor1	III_Song	Mzk=	9	Base64, Hex
III_Song	Razor1	RzQ9PT09PT0=	7	Base64, Base32

Razor1	III_Song	NTM2ZjZkNjU3NzY 4NjU3MjY1MjA2NTc 4NzA2NTZINzM2OT c2NjUyYzIwNDkyM DY4NmY3MDY1Mm UyMA==	Somewhere expensive, I hope.	Base64
Razor1	III_Song	MjQzNzMwMzAyYz MwMzAzMDIwNjk3 NDIwNjk3MzJlMjA1 NzY4NjU3MjY1MjA 2MzYxNmUyMDQ5 MjA2ZDY1NjU3NDI wNzk2Zjc1M2Y=	\$700,000 it is. Where can I meet you?	Base64
III_Song	Razor1	SkVRSE8yTE1OUVF HRVpKQU5GWENB NURQT1ZSV1FJRFh ORjJHUUIEVU5CU1 NBWUxFTVJaR0s0M 1RGWT09PT09PQ==	I will be in touch with the address.	Base64, Base32
III_Song	Genuis1	SUZaU0E1M0ZFQ1N HUzQzRE9WWlhHW kxFRUJTV0M0VE1O RINYRUxCQUpFUU dFWkxNTkZTWE1aS kFKRVFHMjJMSE5C MkNBWVRGRUJRV 0UzREZFQjJHNklES U1WV0hBSURaTjUy U0E1M0pPU1VDQ TZ MUE9WWkNBNDN GTUZaR0cyQk8=	As we discussed earlier, I believe I might be able to help you with your search.	Base64, Base32
Genuis1	III_Song	MTEIDA0MCAxNj MgMTQ1IDE0NSAw NTYgMDQwIDEyNC AxNTAgMTQ1IDE1 NiAwNDAgMTY3ID E0NSAwNDAgMTU1 IDE2NSAxNjMgMTY 0IDA0MCAxNTUgM TQ1IDE0NSAxNjQg MDU0IDA0MCAxN DEgMTU2IDE0NCA wNDAgMTExIDA0M CAxNjcgMTUxIDE1 NCAxNTQgMDQwID E2MyAxNDUgMTQ1 IDA0MCAxNjQgMT UwIDE0NSAwNDAg MTY2IDE0MSAxNT QgMTUxIDE0NCAx NTEgMTY0IDE3MS AwNDAgMTU3IDE0 NiAwNDAgMTY0ID E1MCAxNTEgMTYz IDA0MCAxNDMgM TU0IDE0MSAxNTEg MTU1IDA1Ng==	I see. Then we must meet, and I will see the validity of this claim.	Base64, Octal
III_Song	Genuis1	SkVRR0dZTE9FQJH S01ESk5ZUUdHT0xH TUuV0VPRERN SV pXRU1KWkc1UVdL TkxETVUyR0VZTEd IQTJEQ05MQkdNM1 RLWVJBTzVVWEky REpOWVFISTJERkV CM1dLWkxMRlk9PT 09PT0=	I can be in Caracas within the week.	Base64, MD5
Genuis1	III_Song	MTE2IDE1NyAwNT YgMDQwIDEyNiAx NTcgMTY0IDA0MC AxNTAgMTQ1IDE2 MiAxNDUgMDU2ID A0MCAxMDMgMTQ xIDE1NiAwNDAgMT ExIDA0MC AxNTYg	No. Not here. Can I not go to you?	Base64, Octal

		MTU3IDE2NCAwND AgMTQ3IDE1NyAw NDAgMTY0IDE1Ny AwNDAgMTcxIDE1 NyAxNjUgMDc3		
III_Song	GenuisI	SkVR0RMzSkFNRIrI RVIMSk1RUUhJMkR CT1FRSE8zM1ZOU1 NDQVIURkVCMlc0N TNKT05TUzRJQ0pF QjNXUzNETUVCW1 dLM1RFRUI0VzY1S kFNRVFHMIpMVE9 OUVdPWkpBTzVVW EkyQkFPUIVHS0IER U1GMkdLSURCTlpT Q0EzRFBNTIFYSTJ MUE5ZUUhJMkRTTj UyV08yQkFNRVFH MjMzU01VUUhHWk xET1ZaR0JREdONV pHMkIEUE1ZUUdH MzNOTIYyVzQyTER NRjJHUzMzT0ZZPT0 9PT09	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.	Base64, Base32
III_Song	Method	SIZaQzRJQ05NVjJH UTMzRUZRUUVTS URCTIVRR0s2RERO RjJHS1pCQU1GUkc2 NUxVRUIyRIFaSKFP QlpHNjQzUU1WUlhJ SURQTVIRSEkyREZ FQkJXUVpMVE9NU UVFMzNZTkZYR09J RFhONVpHWVpCQ U9SVVhJM0RGRUJS VzYzTEpOWIRTQT VEUEVCSUhTMzNP TTU0V0MzVEhGWT 09PT09PQ==	Mr. Method, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang.	Base64, Base32
Method	III_Song	NDkyMDYxNmQyM DZINmY3NDIwNzM 3NTcyNjUyMDc3Njg 2ZjIwNzk2Zjc1MjA2 MTcyNjUyYzIwNjI3 NTc0MjA0OTIwNjg2 MTc2NjUyMDYxNm UyMDY5NjQ2NTYx MmUyMDQ1Njk3ND Y4NjU3MjIwNzc2MT c5MmMyMDQ5MjA2 MTZkMjA2ZTZmNz QyMDY5NmU3NDY 1NzI2NTczNzQ2NTY 0MmU=	I am not sure who you are, but I have an idea. Either way, I am not interested.	Base64
III_Song	Method	U1NCaGJTQnFkWE4 wSUoddmNHVm1kV 3d1SUVsMEIJZHkV 3hrSUcxbFIxNGdjMj hnYlhWamFDQjBieU JvWVhabEIJUibBKRV FHQzNKQU5KMIhH NUJBTkjYWEFaTED PVldDNEIDSk9RUUh PMzNWTJQTQ0EzTE ZNrlhDQTQzUEVC V1hLWTNJRUlyRzZ JREINRjNHS0IEVU5 CU1NBVkrKRT1JXR0 tjREINVipHS0xSQuT CV0dLWUxUTVVRR 0czM09PTIVXSvpM U0VCVvhJTFE9	I am just hopeful. It would mean so much to have t I am just hopeful. It would mean so much to have the Title here. Please consider it.	Base64, Base32
III_Song	Killah	SkJYWE9JREpPTVfI STJERkVCM1dLWU	How is the weather in Qatar, Mr. Killah?	Base64, Base32

		xVTkJTWEVJREpO WVFGQ1IMVU1GW kNZSUNOT0IYQ0FT M0pOUldHQzJCNw= =		
Killah	III_Song	MTEwIDE1NyAxNjQ gMDU0IDA0MCAXN DEgMTYzIDA0MCA xNDEgMTU0IDE2Ny AxNDEgMTcxIDE2M yAwNTYgMDQwIDE yNyAxNTAgMTU3ID A0MCAXNTEgMTYz IDA0MCAXNjQgMT UwIDE1MSAxNjMg MDc3	Hot, as always. Who is this?	Base64, Octal
III_Song	Killah	SkVRR0MzSkFNRVF HTVIMT0VCWFdNS UNETkJTWEc0WkFJ SlhYUTJMT000WEN BU0pBTzVYWEszRE VFQldHNjVURkVC Mkc2SURUTVZTU0 E1REINVVFGSTJMV U5SU1NBmRGTJIT Q0EyTE9FQkZXNjR URk1FWEE9PT09	I am a fan of Chess Boxing. I would love to see the Title held in Korea.	Base64, Base32
Killah	III_Song	MTI3IDE0NSAwND AgMTY3IDE1MSAx NTQgMTU0IDA0MC AxNTAgMTQxIDE2 NiAxNDUgMDQwID E2NCAXNTcgMDQw IDE2MyAxNDUgMT Q1IDA0MCAXNTAg MTU3IDE2NyAwND AgMTY0IDE1MCAX NDUgMDQwIDE0Mi AxNTEgMTQ0IDA0 MCAXNjQgMTY1ID E2MiAxNTYgMTYzI DA0MCAXNTcgMTY 1IDE2NCAXNtY=	We will have to see how the bid turns out.	Base64, Octal
III_Song	Killah	SkZaU0E1REINVlpH S0lEQk5aNfHJMkRK TlpUU0E1REINRjJD QVNKQU1OWFhLM 0RFRUJTrzZJRFVO NFFHUVpMTU9BU UcyWUxMTVVRSF MzM1ZPSVFHSVpM RE5GWldTMzNPRUJ TV0M0M0pNVlpENj 09PQ==	Is there anything that I could do to help make your decision easier?	Base64, Base32
Killah	III_Song	MTE2IDE1NyAwND EgMDQwIDEyNCAX NTAgMTQ1IDA0MC AxNDcgMTYyIDE0N SAxNDEgMTY0IDA 0MCAXNTYgMTQxI DE2NCAXNTEgMTU 3IDE1NiAwNDAGMT U3IDE0NiAwNDAG MTIxIDE0MSAxNjQ gMTQxIDE2MiAwN DAgMTY3IDE1NyAx NjUgMTU0IDE0NCA wNDAGMTU2IDE0N SAxNjYgMTQ1IDE2 MiAwNDAGMTQyID E0NSAwNDAGMTYz IDE2NyAxNDEgMTc xIDE0NSAxNDQgM DQwIDE2MyAxNTcg MDQwIDE0NSAxND	No! The great nation of Qatar would never be swayed so easily.	Base64, Octal

		EgMTYzIDE1MSAx NTQgMTcxIDA1Ng= =			
Killah	Ill_Song	MTE2IDE1NyAxNjlg MDQwIDE2NyAxNT cgMTY1IDE1NCAxN DQgMDQwIDEExMS AwNTYgMDQwIDEy NyAxNDUgMDQwID E0NCAxNTcgMDQw IDE1NiAxNTcgMTY 0IDA0MCAxNjQgMT QxIDE1MyAxNDUg MDQwIDE1MyAxNT EgMTU2IDE0NCAx NTQgMTcxIDA0MC AxNjQgMTU3IDA0 MCAxNjQgMTUwID E1MSAxNjMgMDQw IDE2MCAxNDEgMT Y0IDE1MCAxNDUg MTY0IDE1MSAxND MgMDQwIDE1NiAx NTcgMTY0IDE1MS AxNTcgMTU2IDA0 MCAxNTcgMTQ2ID A0MCAxNDIgMTYy IDE1MSAxNDIgMT Q1IDE2MiAxNzEgM DU2	Nor would I. We do not take kindly to this pathetic notion of bribery.	Base64, Octal	
	Ill_Song	Raekwon	SIZaQzRJQ1NNRINX VzUzUE5ZV0NBMK RCT1pTU0E2TFBPV VFIRzREUE5OU1cOS URYTkYyR1FJQ05P SVhDQVVUQ1BKWF hFUfk9	Mr. Raekwon, have you spoken with Mr. Razor?	Base64, Base32
Raekwon	Ill_Song	NDkyMDY4NjE3NjY 1MmMyMDYyNzU3 NDIwNDkyMDc3Nm Y2ZTkyNzQyMDYy NjUyMDYyNmY3NT Y3Njg3NDIwNzM2Zj IwNjU2MTczNjk2Yzc 5MmU=	I have, but I won't be bought so easily.	Base64	
	Ill_Song	Raekwon	SupYWEtaM0lPUTd TQVQzR0VCUlc2NU xTT05TU0EzVFBU VhDQVdMUE9VUud DNFRGRUJRVzRJRF BNWIRHU1kzSk1GV 0NBMzNPRIyR1Fa SkFNVjRHS1kzV9S VVhNWkpBTU5YVz IzTEpPUjHS1pKQU 41VENBNURJTVVR RVNRMkNJRvHdQ VNKQU5KMlhHNuj BTzVRVzQ1QkFQRI hYS0lEVU40UUdXM 1RQtzRRSEkyREJP UVFFU0lEQk5VUud RwKxTTVVRSEkzW kFOQInXWTRCQU5 WUVdXWkpBUEZY WEs0ukFNuINXRzJ MVE5GWFc0SURCT 01RR0tZTFRQRVFH QzRaQU9CWFhHND NKTUpXR0tMUkE=	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.	Base64, Base32
Raekwon	Ill_Song	NDkyMDc3NmY3NT ZjNjQyMDZINjU2NT Y0MjA2MTc0MjA2Y zY1NjE3Mzc0MjAzM jMwMjA2ZDY5NmM	I would need at least 20 million Rubles.	Base64	

		2YzY5NmY2ZTIwNT I3NTYyNmM2NTcz MmU=		
Ill_Song	Raekwon	SU5YVzQ0M0pNUlN YRUIESk9RUUDJMz NPTVVYQ0FTSkFP NVVXWTNCQU9OU 1c0WkJBUEZYWEtJ RFVOQINTQTJMT01 aWFhFM0xCT1JVvz YzUkfNWlhYRUlEV U5CU1NBWkRTTjV ZQzIzM0dNWVFIQT MzSk5aMkNBNDNQ TjVYQzQ9PT0=	Consider it done. I will send you the information for the drop-off point soon.	Base64, Base32

2.2.3.2 Members Involved and Results of Bribery

To determine the official's locations, the decoded conversations in Table 3 were analysed. The conversation between Razor1 and Ill_Song had revealed that the former had accepted the bribe of \$700,000. In addition, Ill_Song had asked them if they could send them a gift, specifically mentioning the 'City of Love'. This is Paris, just by googling it on the internet, which suggests that's where that official is now. Ill_Song messaged Genius1 offering to help them with something and saying to them they could be in Caracas within the week, which is where that official could be. However, there is no clue as to whether Genius1 has accepted a bribe. Method refused to speak to Ill_Song and questioned who they were. This would also suggest that they didn't accept the bribe. Killah is asked by Ill_Song 'How is the weather in Qatar, Mr. Killah?', which immediately means that official must be there. Killah rejects Ill_Song's request for a bribe. Finally, Raekwon is asked if he has spoken with Razor and initially doesn't want to accept the bribe. The conversation suggests he is a member of ICBA, an American Organisation, and he eventually takes the bribe and convinces Ill_Song to give him 20 million rubles. This is a form of currency used in the Russian Economy which could suggest they are in Russia. The results are shown below in Table 5:

Table 5 Results from IRC Conversations showing Username, Suspected Location and If Accepted the Bribe

Username	Suspected Location	Bribe – Yes/No
Razor1	Paris	Yes
Genius1	Caracas	Unsure
Method	Unsure	No
Killah	Qatar	No
Raekwon	Russia	Yes

3. Investigation of ‘Capture3.pcap’ Network Capture

3.1 Brief

“We have picked up FTP traffic between a suspected corrupt official and a foreign national. Decode this traffic capture and provide us with evidence on what item the corrupt official received. We suspect that some anti-forensic practices have been used to hide information sent—an Edward Snowden quote may help you decipher.”

3.2 Method

3.2.1 Hash Checksum

An MD5 Hash was calculated on the third capture file for Kali and Windows. Using md5sum, the hash was calculated:

```
(kali㉿kali)-[~/Desktop/analysis]
$ md5sum 'Capture 3.pcap'
b5b6c3f6edbb14c1806c62c06da119cd Capture 3.pcap
```

Figure 41 Hash Calculated on Third Capture File using md5sum

And, similarly, another hash was calculated like output from above:

```
C:\Users\ethan\Desktop\University\4th\forensics\pcap>certutil -hashfile "Capture 3.pcap" MD5
MD5 hash of Capture 3.pcap:
b5b6c3f6edbb14c1806c62c06da119cd
CertUtil: -hashfile command completed successfully.
```

Figure 42 Hash Calculated on Third Capture File using certutil

3.2.2 Statistical Flow Analysis

Since it was known that the traffic from the suspected corrupt official and the foreign national was conducted over File Transfer Protocol (FTP), statistical flow analysis tools were combined with the grep utility to allow more efficient analysis. This could have been performed easily in the second capture because it was known that IRC traffic was conducted (port 6667). FTP operates under TCP port 21 so using tools that allow us to specify source ports, in this case 21, can make data easier to find. Using rwuniq, the file was searched for any entries that had FTP traffic. In this case, two entries appeared between two hosts: 172.29.1.23 and 172.29.1.21. The command used to retrieve the output below was ‘rwuniq capture3.rw –fields=bytes, packets, stime, etime, sip, sp, dip, dp –values=records, bytes, packets –sort-output | grep 21’:

40	1	2014/07/03T20:37:37	2014/07/03T20:37:37	172.29.1.23	51461	172.29.1.21	21	2	80	2
40	1	2014/07/03T20:37:38	2014/07/03T20:37:38	172.29.1.23	51462	172.29.1.21	21	2	80	2

Figure 43 FTP Traffic Identified by ‘rwuniq’

FTP traffic is also confirmed if we search for source ports. Filtering by the top 200 packets reveals one entry of FTP using the tool ‘rwstats’:

51533	42	0.221053	71.331579
42029	41	0.215789	71.547368
21	41	0.215789	71.763158

Figure 44 Filtering by Source Port

Further analysis also further cemented that the two hosts had acted as a service to each other since both had been using port 21:

(kali㉿kali)-[~/Desktop/analysis/third]						
\$ rwstats outfile.rw --fields=1,2,3,4 --count=3000 grep -w 21						
172.29.1.21	208.71.121.1	59747	443	4	0.168279	0.168279
172.29.1.23	172.29.1.21	51462	21	3	0.126210	0.294489
208.71.121.1	172.29.1.21	443	59747	3	0.126210	0.420698
172.29.1.23	172.29.1.21	51461	21	3	0.126210	0.673117
85.214.32.27	172.29.1.21	80	54477	2	0.084140	0.757257
4.2.2.1	172.29.1.21	53	44486	1	0.042070	9.465713
85.214.32.27	172.29.1.21	80	54419	1	0.042070	9.507783
172.29.1.21	172.29.1.23	21	51461	1	0.042070	9.549853
172.29.1.21	72.21.215.147	59589	80	1	0.042070	9.676062
4.2.2.1	172.29.1.21	53	43740	1	0.042070	9.718132
172.29.1.21	4.2.2.1	60553	53	1	0.042070	9.844342
74.125.239.104	172.29.1.21	443	50407	1	0.042070	9.886411
4.2.2.1	172.29.1.21	53	37856	1	0.042070	10.012621
172.29.1.21	205.188.95.228	56602	80	1	0.042070	10.096761
23.7.199.108	172.29.1.21	80	36364	1	0.042070	10.138830

Figure 45 More FTP Traffic

3.2.3 Wireshark Analysis

FTP traffic was filtered for within Wireshark. This revealed several entries from frame 5846 onwards:

tcp.port == 21						
No.	Source	Destination	Protocol	Length	Info	Time
5846	172.29.1.23	172.29.1.21	TCP	62	51461 → 21 [SYN]	20:36:34.983735
5847	172.29.1.21	172.29.1.23	TCP	62	21 → 51461 [SYN]	20:36:34.983967
5848	172.29.1.23	172.29.1.21	TCP	60	51461 → 21 [ACK]	20:36:34.983974
5849	172.29.1.23	172.29.1.21	TCP	62	51462 → 21 [SYN]	20:36:34.984217
5850	172.29.1.21	172.29.1.23	TCP	62	21 → 51462 [SYN]	20:36:34.984466
5851	172.29.1.23	172.29.1.21	TCP	60	51462 → 21 [ACK]	20:36:34.984473
5852	172.29.1.21	172.29.1.23	FTP	79	Response: 220	20:36:34.986215
5853	172.29.1.21	172.29.1.23	FTP	79	Response: 220	20:36:34.986965
5854	172.29.1.23	172.29.1.21	FTP	69	Request: USER	20:36:35.003262
5855	172.29.1.21	172.29.1.23	TCP	60	21 → 51461 [ACK]	20:36:35.003269
5856	172.29.1.21	172.29.1.23	FTP	88	Response: 331	20:36:35.003213
5857	172.29.1.23	172.29.1.21	FTP	69	Request: USER	20:36:35.003452
5858	172.29.1.21	172.29.1.23	TCP	66	21 → 51462 [ACK]	20:36:35.003460
5859	172.29.1.21	172.29.1.23	FTP	88	Response: 331	20:36:35.003464
5860	172.29.1.23	172.29.1.21	FTP	69	Request: PASS	20:36:35.003702
5861	172.29.1.23	172.29.1.21	FTP	60	No Document Data	20:36:35.003710

> Frame 5846: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
> Ethernet II, Src: Dell_fax:6:cc (00:08:74:fa:6:cc), Dst: Giga-Byte_e4:8:b:d3 (50:e5:49:e4:8:b:d3)
> Internet Protocol Version 4, Src: 172.29.1.23, Dst: 172.29.1.21
> Transmission Control Protocol, Src Port: 51461, Dst Port: 21, Seq: 0, Len: 0

Figure 46 FTP Traffic within Wireshark

Following the TCP stream of these file revealed a file transfer had taken place. A ZIP file called ‘sandofwhich.zip’ was downloaded by a user the name of ‘Ill_Song’:

```
Wireshark - Follow TCP Stream (tcp.stream eq 156) · Capture 3.pcap

220 Super Secret Server
USER Ill_Song
333 Please specify the password.
PASS Ill_Song
230 Login successful.
OPTS UTF8 ON
200 Always in UTF8 mode.
CWD /home/Ill_Song
250 Directory successfully changed.
TYPE I
200 Switching to Binary mode.
PASV
227 Entering Passive Mode (172,29,1,21,216,252).
RETR sandofwhich.zip
150 Opening BINARY mode data connection for sandofwhich.zip (24792 bytes).
226 Transfer complete.
500 OOPS: vsf_sysutil_recv.Peek: no data
```

Figure 47 File Downloaded by Ill_Song

Using Wireshark’s built-in features to extract HTTP and FTP files from network frames (Anthony, 2022), FTP data was filtered for and TCP stream was followed to allow the data to be saved in a raw format. Whilst

filtering for FTP data, another file transfer was discovered which saw another ZIP file to be downloaded called ‘ojd34.zip’:

No.	Source	Destination	Proto	Length	Info	Time
5892	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.258665
5893	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.258678
5894	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.258914
5896	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.258930
5897	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259165
5898	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259178
5899	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259413
5900	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259425
5901	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259663
5902	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259674
5903	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259914
5904	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.259925
5910	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.260414
5911	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.260663
5912	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.260675
5913	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR sandofwh..)	20:36:37.260913
5914	172.29.1.21	172.29.1.23	FTP..	1486	FTP Data: 1432 bytes (PASV) (RETR sandofwh..)	20:36:37.260924
5938	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR ojd34..zil..)	20:36:38.412258
5939	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR ojd34..zil..)	20:36:38.412263
5940	172.29.1.21	172.29.1.23	FTP..	1514	FTP Data: 1460 bytes (PASV) (RETR ojd34..zil..)	20:36:38.412499

Figure 48 Transfer of ojd34.zip on Frame 5938 onwards

Both files were saved in a raw format and allowed data to be seen. Within both folders, they contained several JPEG images. The contents of sandofwhich.zip and ojd34.zip can be seen below:

allow.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
and.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
around.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
basic.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
building.jpg	JPG File	1 KB	No	2 KB	7%	23/06/2014 12:09
cant.jpg	JPG File	1 KB	No	2 KB	11%	23/06/2014 12:09
conscience.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
terrorism.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
Watergate.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
web-based.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43

Figure 49 Contents of ‘ojd34.zip’

destroy.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
for.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
freedom.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
good.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
government.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
l.jpg	JPG File	1 KB	No	2 KB	20%	23/06/2014 12:09
in.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
NSA.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
rights.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
security.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43

Figure 50 Contents of sandofwhich.zip

It was warned in the brief that anti-forensics methods had been used to conceal data and that an Edward Snowden quote would assist in this part of the investigation. The first approach was to determine if the JPEG files were holding any data, i.e., using ‘strings’ command, and research was conducted into steganalysis tools that would allow this data to be extracted. Unfortunately, without a passphrase that would unlock the files it would be useless. Furthermore, the files were not encrypted – with a passphrase – so that approach wasn’t necessary. Using the strings command, the file ‘I.jpg’ from Figure 50 was found to contain some data and was in fact a JPEG file. Looping through these files revealed that most of them had a file type of ‘data’ so at this point it wasn’t sure what method was used. The Edward Snowden quote mentioned in the brief was then

studied for any clues indicating another approach that could be used. It was then realised that within the contents of Figure 49 and 50 that each of the files had filenames that connected to a quote said by Edward Snowden that is shown below:

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

Figure 52 Edward Snowden Quote

However, files present in the two ZIP files captured did not match up entirely to the quote. ‘I.jpg’ was thought to be the start of the JPEG file while the other files that matched the quote may be a part of the original file. An example would be ‘cant.jpg’ which was hypothesised to be part of the original file because in the quote it is ‘I can't ...’. The method of steganography used here was that the original file was split up into smaller versions of itself, almost like the quote, to avoid detection. This is further reinforced when examining the timestamps of the files because files ending in ‘12:09’ match words said in the quote, although not entirely. There was another timestamp seen ending in ‘14:43’, which suggests that another attempt was made to hide another file. In addition, these ones do not match the quote. It was then realised that to complete the quote said by Snowden that further analysis must be undertaken to recover the other files. At first, it was thought that there were some other secret FTP transfers done that would be useful to complete the quote. So, FTP was investigated once again but no other file transfers could be seen. Statistical flow analysis was performed again which revealed heavy HTTP traffic, as in the other packet captures.

HTTP was investigated in Wireshark which was filtered using Hierarchy Statistics. It was here that it was discovered that MIME, as previously demonstrated in Capture 1, was in use at that time. This was then filtered for in Wireshark, which revealed two frames:

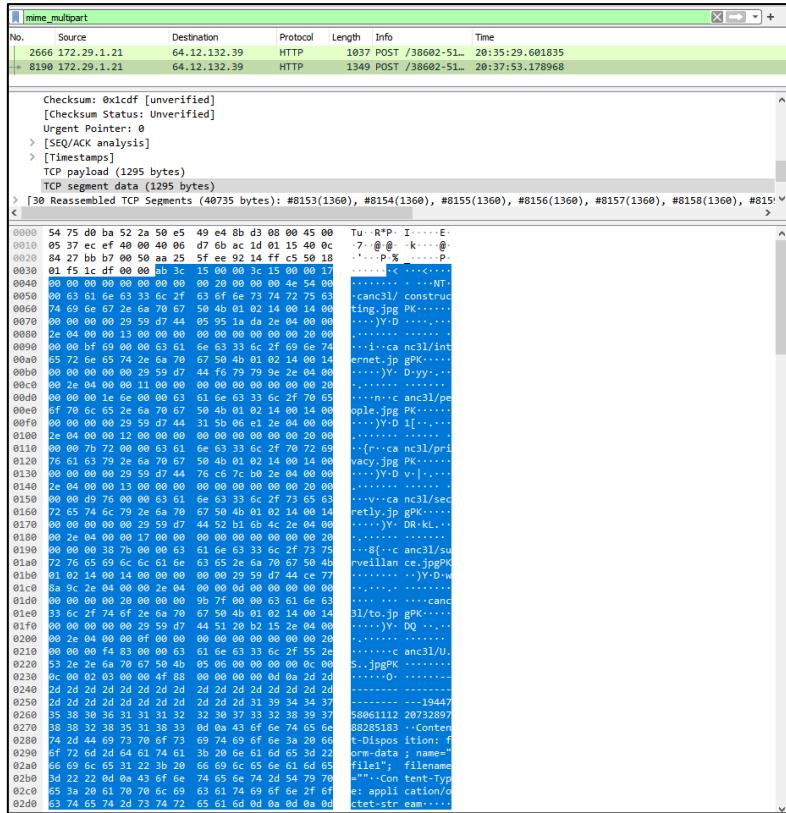


Figure 53 MIME Traffic Between 172.29.1.21 and 64.12.132.39

Upon seeing the results shown in Figure 53, this was analysed and revealed communications between a ‘da.genius36@aol.com’ to a known person of interest ‘kim.illsong@aol.com’. Here, ‘da.genius’ talks to ‘kim.illsong’ about a bold claim they made and they would like to see some proof:

```

0130 6f 20 66 69 6e 64 20 69 74 2e 5c 6e 20 5c 6e 5c o find i t.\n \n
0140 6e 20 5c 6e 5c 6e 20 5c 6e 5c 6e 2d 2d 2d 2d 2d o \n\n \n-----\nOriginal Message
0150 4f 72 69 67 69 6e 61 5c 20 4d 65 73 73 61 67 65 ----\nFrom: The
0160 2d 2d 2d 2d 2d 5c 6e 46 72 6f 6d 3a 20 54 68 65 Gza <da .genius3
0170 20 47 7a 61 20 3c 64 61 2e 67 65 6a 69 75 73 33 6baol.co m>\nTo:
0180 3e 40 61 6f 6c 2e 63 6f 6d 3e 5c 6c 54 6f 3a 20 kim.illsong @aol.com
0190 6b 69 6d 2e 69 6c 6c 73 6f 6e 67 20 3c 6b 69 6d .illsong @aol.com
01a0 3e 5c 6e 53 65 6e 74 3a 20 54 68 75 2c 20 4a 75 >\nSent: Thu, Ju
01b0 6c 20 33 2c 20 32 30 31 34 20 62 33 31 38 20 70 l 3, 2014 2:18 p
01c0 6d 5c 6e 53 75 62 6a 65 63 74 3a 20 55 72 67 65 \nSubject: ct: Urge
01d0 6d 5c 6e 53 75 62 6a 65 63 74 3a 20 55 72 67 65 nt\n\n\nYou have
01e0 6e 74 5c 6e 5c 6e 5c 59 6f 75 20 68 61 76 65 made a bold cla
01f0 20 6d 61 64 65 20 61 20 62 6f 6c 64 28 63 6c 61 im but i'd like
0200 69 6d 20 62 75 74 20 69 27 64 28 68 69 6b 65 20 to see some prod
0210 74 6f 20 73 65 65 20 73 6f 6d 65 20 78 72 6f 6f f.\n\n\nRich
0220 6e 20 5c 6e 53 6e 20 61 6f 6d 22 2c 52 69 63 68 0230 45 64 74 22 3a 74 72 75 66 6f 6d 61 67 65 Editor tr ue."Pre
0240 74 4d 65 73 61 67 65 49 44 22 3c 28 53 38 ntMessage eID":<
0250 44 31 36 21 31 31 36 63 34 35 61 33 33 3d 44 0260 0261 6f 64 43 24 42 30 41 42 40 77 65 62 6d 61 69 6d 1DC_B0AB @webmail
0270 2d 64 32 36 37 2e 73 79 73 6f 70 73 2e 61 6f 6d -j267.9 copi.aol
0280 2e 63 6f 6d 3e 22 2c 79 22 41 6e 73 77 65 72 55 49 .com">" AnswerU
0290 44 22 3a 22 32 37 32 32 22 22 41 6e 73 77 65 72 O": "272" "Answer
0300 46 6f 6c 64 65 72 22 3a 22 29 6e 62 6f 78 22 2c Folder", "Inbox"
0280 22 53 6f 75 72 63 65 4d 73 67 55 44 44 22 3a 22 0290 "SourceID": "SourceI msgID": "272", "So urceMsgF
02c0 32 37 32 22 2c 22 53 6f 73 72 63 65 4d 73 67 46 older": "1" Inbox",
02d0 6f 66 64 65 72 22 3a 22 49 6e 62 6f 78 22 2c 22 02e0 SourceAttachment
02e0 53 6f 75 72 63 65 41 74 74 61 63 68 6d 65 6e 74 IDs": [], "Preload
02f0 49 44 73 22 3a 5b 5d 2c 22 50 72 65 6c 6f 61 64 attachments": [
0300 41 74 74 61 63 68 6d 65 6e 74 49 44 73 22 3a 5b ], "SendAttachmentS
0310 5d 2c 22 53 65 6e 64 41 74 74 61 63 68 41 73 4c 0320 69 6e 6b 73 22 3a 66 61 6c 73 65 2c 22 43 6f 6d links": "fa lse", "Com
0330 70 6f 73 65 54 79 79 65 22 3a 22 72 65 79 79 poseType": "reply
0340 22 22 22 61 63 74 69 6f 66 22 3a 22 53 65 64 ", "action": "Send
0350 4d 65 73 73 61 67 65 22 7d 5d 0d 0d 2d 2d 2d Message" }-----\n0360 2d ----- -----
0370 3d 3c 38 35 39 30 36 31 32 33 39 36 38 34 ----- -5063905
0380 32 3c 38 35 39 30 36 31 32 33 39 36 38 34 28859906 81239684
0390 31 32 37 38 6d 0a 43 6f 66 74 65 6e 74 20 44 69 1278..Content-Di
03a0 73 70 6f 73 69 74 69 6f 66 74 20 69 6f 72 6d 26 spositio n: form-
03b0 64 6f 6c 61 63 65 66 51 60 62 67 69 75 6f 61 data; na me= auto
03c0 6d 61 74 69 63 22 0d 0a 0d 0b 6d 61 6c 73 65 60 matic" .."false"
03d0 6d 2d ----- -----
03e0 3d 50
03f0 36 33 39 30 35 32 38 38 35 39 30 36 38 31 32 6390528859906812396841278
0400 33 39 36 38 34 31 32 37 38 2d 2d 0d 0a 396841278 8--.

```

Figure 54 Email from da.genius36@aol.com to kim.illsong@aol.com

Within the attachment of the email shown in Figure 53, contained two ZIP files titled '34jdsioj.zip' and 'breaking_bad_season_6.zip':

```

▼ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----506390528859906812396841278\r\n
  [Type: multipart/form-data]
  First boundary: -----506390528859906812396841278\r\n
▼ Encapsulated multipart part: (application/zip)
  Content-Disposition: form-data; name="file0"; filename="34jdsioj.zip"\r\n
  Content-Type: application/zip\r\n\r\n
  ▼ Media Type
    Media type: application/zip (26135 bytes)
    Boundary: \r\n-----506390528859906812396841278\r\n
  ▼ Encapsulated multipart part: (application/zip)
    Content-Disposition: form-data; name="file1"; filename="breaking_bad_season_6.zip"\r\n
    Content-Type: application/zip\r\n\r\n
    ▼ Media Type
      Media type: application/zip (17383 bytes)
      Boundary: \r\n-----506390528859906812396841278\r\n
    ▼ Encapsulated multipart part: (application/octet-stream)
      Content-Disposition: form-data; name="file2"; filename=""\r\n
      Content-Type: application/octet-stream\r\n\r\n
      Boundary: \r\n-----506390528859906812396841278\r\n
    ▼ Encapsulated multipart part:
      Content-Disposition: form-data; name="requests"\r\n\r\n
      ▼ Data (1661 bytes)
        Data: 5b7b2246726f6d223a226b696d2e696c6c736f6e6740616f6c2e636f6d222c22546f223a...
        [Length: 1661]
      Boundary: \r\n-----506390528859906812396841278\r\n
    ▼ Encapsulated multipart part:
      Content-Disposition: form-data; name="automatic"\r\n\r\n
      ▼ Data (5 bytes)

```

Figure 55 Metadata of Frame 2666 Showing Two ZIP Files: 34jdsioj.zip and breaking_bad_season_6.zip

Also investigated was the second email sent, which revealed a message from Ill_Song with the subject of 'another'. A ZIP file was also sent with this transaction:

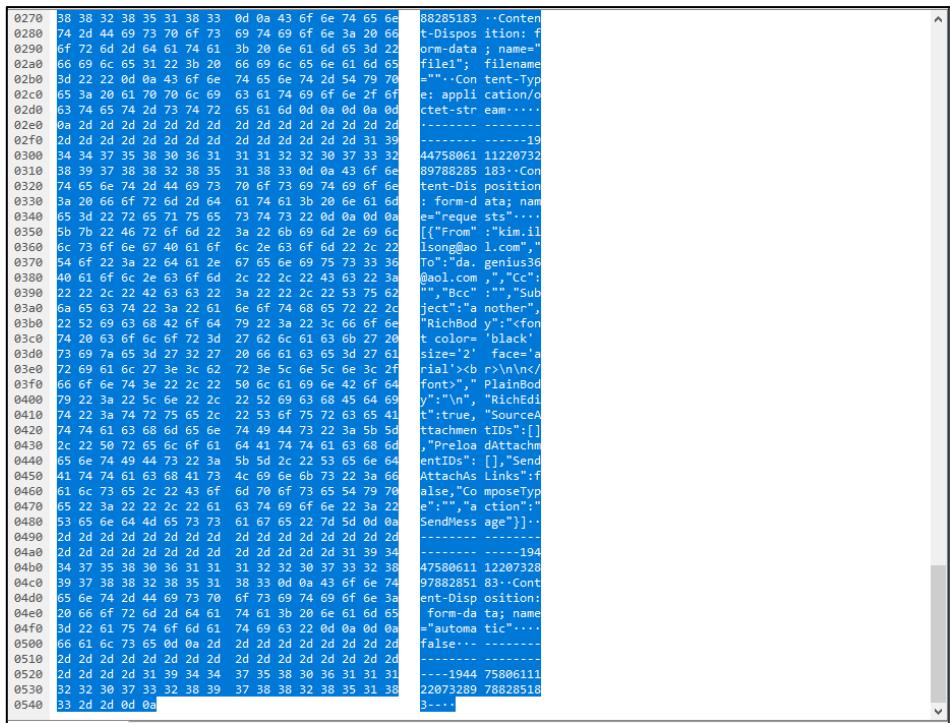


Figure 56 Email from kim.illsong@aol.com to da.genius36@aol.com

Metadata within Frame 8190 revealed another ZIP file that was sent titled ‘canc3l.zip’:



Figure 57 Metadata of Frame 8190 Revealing ‘canc3l.zip’

Analysis allowed the hosts to have their IP Addresses and MAC Addresses recorded. These can be seen below:

Table 6 IP Addresses along with MAC Addresses for Capture 3

IP Address	MAC Address
172.29.1.21	50:e5:49:e4:8b:d3
64.12.132.39	54:75:d0:ba:52:2a
172.29.1.23	00:08:74:fa:a6:cc

All ZIP files seen in Figures 54 and 56 were then extracted by right-clicking on ‘Media Type: application/zip’ and exporting the packet bytes. These were saved into files and the original filenames kept. These ZIP files were found to be storing JPEG files like the two found earlier. Their contents can be seen below:

corrupt.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
doors.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
human.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
liberties.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
machine.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
massive.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
the.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
theyre.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
this.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
with.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
world.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09

Figure 58 Contents of '34jdsioj.zip'

a.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
because.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
but.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
communism.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
it.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
nor.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
secret.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
secretive.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
their.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10
there.jpg	JPG File	2 KB	No	2 KB	12%	23/06/2014 15:10
unconstitutional.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 15:10

Figure 59 Contents of 'breaking_bad_season_6.zip'

American.jpg	JPG File	6 KB	No	6 KB	1%	23/06/2014 14:43
behind.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
closed.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
condone.jpg	JPG File	5 KB	No	6 KB	7%	23/06/2014 14:43
constructing.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
internet.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
people.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
privacy.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
secretly.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
surveillance.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
to.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
U.S.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09

Figure 60 Contents of 'canc3l.zip'

The files were then looped through to find any inconsistencies. This revealed two more real JPEGs called 'condone.jpg' and 'there.jpg', with other files possessing a raw data type:

```
(kali㉿kali)-[~/Desktop/analysis/third]
$ for var in *.jpg; do file $var; done;
a.jpg: data
allow.jpg: data
American.jpg: data
and.jpg: data
around.jpg: data
basic.jpg: data
because.jpg: data
behind.jpg: Dyalog APL version 36.20
building.jpg: data
but.jpg: data
cant.jpg: data
closed.jpg: data
communism.jpg: data
condone.jpg: JPEG image data, JFIF standard 1.02, aspect ratio, density 100x100, segment length 16, baseline, precision 8, 640x360, components 3
cession.jpg: data
destroy.jpg: data
doors.jpg: data
for.jpg: data
freedom.jpg: data
good.jpg: data
government.jpg: data
human.jpg: data
I.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, baseline, precision 8, 640x425, components 3
in.jpg: data
internet.jpg: data
it.jpg: data
liberties.jpg: data
machine.jpg: data
massive.jpg: data
nor.jpg: data
NSA.jpg: data
people.jpg: data
privacy.jpg: data
rights.jpg: data
secretive.jpg: data
secret.jpg: data
secretly.jpg: data
security.jpg: data
surveillance.jpg: data
terrorism.jpg: data
their.jpg: data
the.jpg: data
there.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 96x96, segment length 16, Exif Standard: [TIFF image data, little-endian, direntries=1], baseline, precision 8, 387x291, components 3
theyre.jpg: data
this.jpg: data
to.jpg: data
unconstitutional.jpg: data
U.S..jpg: data
Watergate.jpg: data
web-based.jpg: data
with.jpg: data
world.jpg: data
```

Figure 61 Bash Command to Reveal File Type of JPEG Files

It was thought that these were the start of the two other JPEGs that had been sent and had been split up into other chunks. This is confirmed by grouping each of the files according to timestamp:

allow.jpg	23/06/2014 12:09	JPG File	2 KB
and.jpg	23/06/2014 12:09	JPG File	2 KB
around.jpg	23/06/2014 12:09	JPG File	2 KB
basic.jpg	23/06/2014 12:09	JPG File	2 KB
building.jpg	23/06/2014 12:09	JPG File	2 KB
cant.jpg	23/06/2014 12:09	JPG File	2 KB
conscience.jpg	23/06/2014 12:09	JPG File	2 KB
destroy.jpg	23/06/2014 12:09	JPG File	2 KB
for.jpg	23/06/2014 12:09	JPG File	2 KB
freedom.jpg	23/06/2014 12:09	JPG File	2 KB
good.jpg	23/06/2014 12:09	JPG File	2 KB
government.jpg	23/06/2014 12:09	JPG File	2 KB
I.jpg	23/06/2014 12:09	JPG File	2 KB
in.jpg	23/06/2014 12:09	JPG File	2 KB
internet.jpg	23/06/2014 12:09	JPG File	2 KB
liberties.jpg	23/06/2014 12:09	JPG File	2 KB
machine.jpg	23/06/2014 12:09	JPG File	2 KB
massive.jpg	23/06/2014 12:09	JPG File	2 KB
people.jpg	23/06/2014 12:09	JPG File	2 KB
privacy.jpg	23/06/2014 12:09	JPG File	2 KB
secretly.jpg	23/06/2014 12:09	JPG File	2 KB
surveillance.jpg	23/06/2014 12:09	JPG File	2 KB
the.jpg	23/06/2014 12:09	JPG File	2 KB
theyre.jpg	23/06/2014 12:09	JPG File	2 KB
this.jpg	23/06/2014 12:09	JPG File	2 KB
to.jpg	23/06/2014 12:09	JPG File	2 KB
U.S.jpg	23/06/2014 12:09	JPG File	2 KB
with.jpg	23/06/2014 12:09	JPG File	2 KB
world.jpg	23/06/2014 12:09	JPG File	2 KB

Figure 62 Image1.jpg Deconstructed

American.jpg	23/06/2014 14:43	JPG File	6 KB
behind.jpg	23/06/2014 14:43	JPG File	6 KB
closed.jpg	23/06/2014 14:43	JPG File	6 KB
condone.jpg	23/06/2014 14:43	JPG File	6 KB
constructing.jpg	23/06/2014 14:43	JPG File	6 KB
corrupt.jpg	23/06/2014 14:43	JPG File	6 KB
doors.jpg	23/06/2014 14:43	JPG File	6 KB
human.jpg	23/06/2014 14:43	JPG File	6 KB
NSA.jpg	23/06/2014 14:43	JPG File	6 KB
rights.jpg	23/06/2014 14:43	JPG File	6 KB
security.jpg	23/06/2014 14:43	JPG File	6 KB
terrorism.jpg	23/06/2014 14:43	JPG File	6 KB
Watergate.jpg	23/06/2014 14:43	JPG File	6 KB
web-based.jpg	23/06/2014 14:43	JPG File	6 KB

Figure 63 Image2.jpg Deconstructed

a.jpg	23/06/2014 15:10	JPG File	2 KB
because.jpg	23/06/2014 15:10	JPG File	2 KB
but.jpg	23/06/2014 15:10	JPG File	2 KB
communism.jpg	23/06/2014 15:10	JPG File	2 KB
it.jpg	23/06/2014 15:10	JPG File	2 KB
nor.jpg	23/06/2014 15:10	JPG File	2 KB
secret.jpg	23/06/2014 15:10	JPG File	2 KB
secretive.jpg	23/06/2014 15:10	JPG File	2 KB
their.jpg	23/06/2014 15:10	JPG File	2 KB
there.jpg	23/06/2014 15:10	JPG File	2 KB
unconstitutional.jpg	23/06/2014 15:10	JPG File	2 KB

Figure 64 Image3.jpg Deconstructed

3.2.3.1 Image1.jpg

Using the cat command, the file was reconstructed into its original counterpart using the quote of Edward Snowden by starting with 'I.jpg' since this was the original start of the JPEG file:

```
(kali㉿kali)-[~/Desktop/analysis/third]
$ cat I.jpg cant.jpg in.jpg good.jpg conscience.jpg allow.jpg the.jpg U.S..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
g this.jpg massive.jpg surveillance.jpg machine.jpg theyre.jpg secretly.jpg building.jpg > IMAGE1.jpg
```

Figure 65 Cat command

The saved file was then accessed which revealed a JPEG file. However, it was not the full image even though it was reconstructed as shown in Figure 65. Solutions to reveal the full image were not successful:

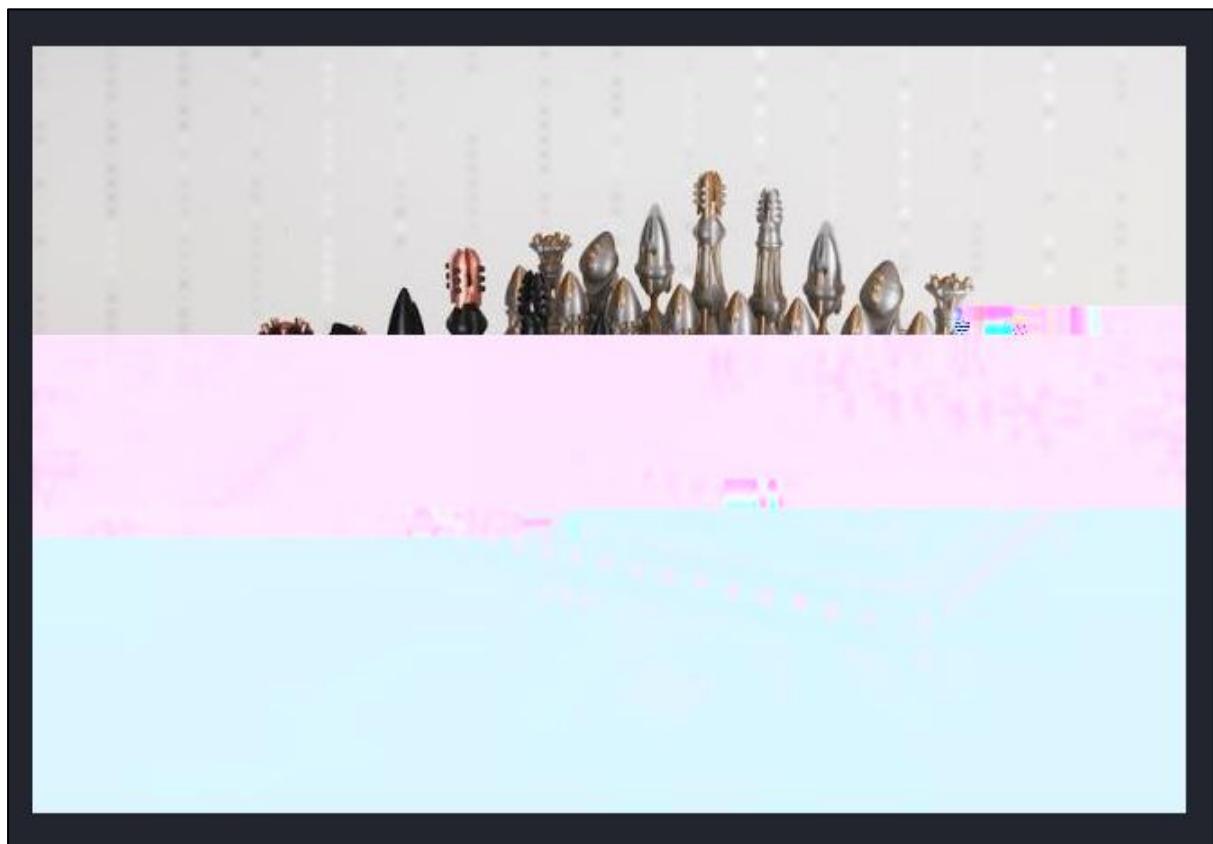


Figure 66 Image1.jpg (incomplete)

3.2.3.2 Image2.jpg

The image was reconstructed according to the timestamps seen in Figure 63. This was successful and revealed an image of a cartoon character.

```
(kali㉿kali)-[~/Desktop/analysis/third]
└─$ cat condone.jpg American.jpg web-based.jpg rights.jpg constructing.jpg security.jpg terrorism.jpg NSA.jpg Wa
tergate.jpg corrupt.jpg human.jpg behind.jpg closed.jpg doors.jpg > IMAGE2.jpg
```

Figure 67 Cat command to construct IMAGE2.jpg



Figure 68 Image2.jpg

3.2.3.3 Image3.jpg

Constructing this image from Figure 64 showed a picture of Kim Jong-un:

```
(kali㉿kali)-[~/Desktop/analysis/third]
└─$ cat there.jpg their.jpg a.jpg it.jpg but.jpg communism.jpg nor.jpg because.jpg unconstitutional.jpg secretiv
e.jpg secret.jpg > IMAGE3.jpg
```

Figure 69 Cat command to construct image



Figure 70 Picture of Kim Jong Un (Image3.jpg)

4. Investigation of ‘Capture4.pcap’ Network Capture

4.1 Brief

“We have uncovered communication traffic between Ill-Song and a known person of interest taking part in the international competition—Ann Dercover. We believe that they are trying to set up a meeting discreetly to avoid detection. We need to know details of the conversation that took place and the date and time that they are planning to meet.”

4.2 Method

Here is output captured by md5sum on the fourth capture file:

```
(kali㉿kali)-[~/Desktop/analysis]
$ md5sum 'Capture 4.pcap'
0487b1b8007e68b20e1f6a8c01467d07  Capture 4.pcap
```

Figure 71 MD5 Checksum on Capture File

Certutil was also used:

```
C:\Users\ethan\Desktop\University\4th\forensics\pcap>certutil -hashfile "Capture 4.pcap" MD5
MD5 hash of Capture 4.pcap:
0487b1b8007e68b20e1f6a8c01467d07
CertUtil: -hashfile command completed successfully.
```

Figure 72 MD5 Hash using certutil

4.2.1 Wireshark Analysis

Although flow analysis was used first to identify the service used for communication between ‘Ill_Song’ and ‘Ann Dercover’, a lot of HTTP traffic was found but it was not first thought to be the service used. UDP Port 5060 which is the port dedicated to Session Initiation Protocol (SIP) was found to have been in use but analysing these within Wireshark found no useful data. Since the name of two aliases involved were known, it was thought that filtering for one of the usernames would provide the results, which would lead to those frames being highlighted. This was successful as filtering for ‘Ill_Song’ immediately showed the contents of packet 24417:

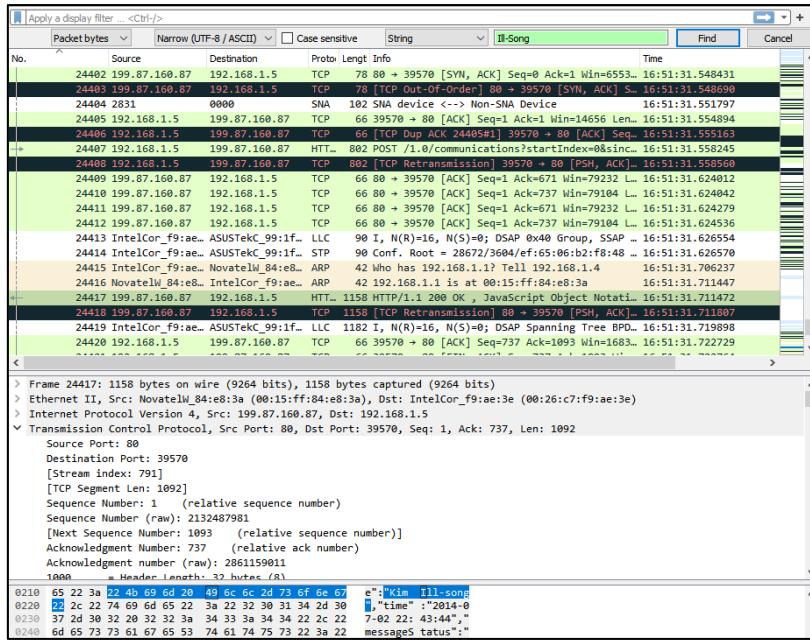


Figure 73 Mentions of Ill_Song

Following ‘tcp.stream eq 791’ reveals several key metadata that shows the application that was used for communications called Android Text Free. This offers SMS and calling services for Android phones. Other key details include the device type which is a Nexus 7. In this TCP stream, a message from Ann Dercov states ‘I told you to pay attention.’:

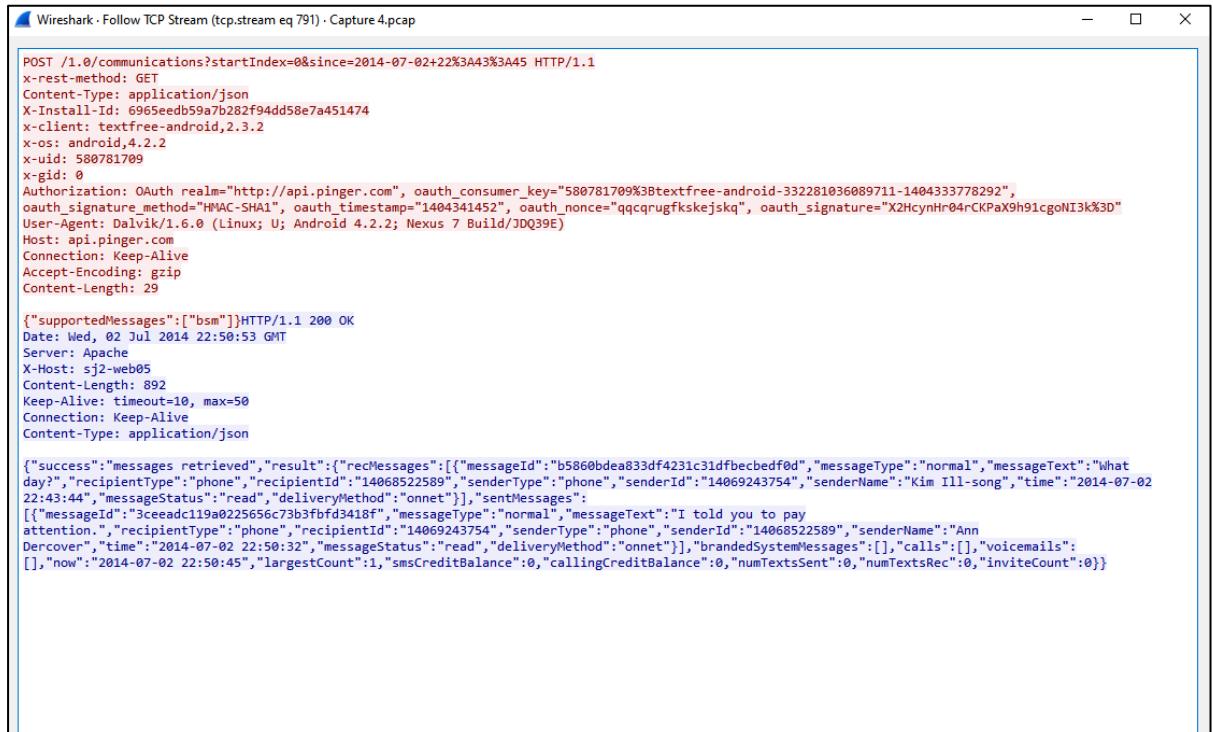


Figure 74 Metadata of Stream eq. 791 showing Message from Ann Dercov to Ill_Song

4.2.1.1 Conversations between Ann Dercov and Ill_Song

Since it was known that they were using TextFree, the rest of the messages between the two members were extracted from Wireshark. It was exported in a JSON format and copied to the Kali machine. Here, the JSON files were read from the command line using ‘cat’ and combined with grep to search for key values

(messageText, messageTxt) since JSON stores data in a key/value format. Grep highlights the filtered text allowing for easier reading:

```
(kali㉿kali)-[~/Desktop/analysis/four/json]
└─$ cat one.json two.json three.json four.json five.json six.json seven.json eight.json nine.json ten.json eleven.json thirteen.json | grep -w "messageText"\|messageTxt"
{"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "45b537c51e5cf2f90f31779e9ec8fc46", "messageType": "normal", "messageText": "Good afternoon, Ann.", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:38:55", "messageStatus": "read", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:38:57", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "who is this?", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "where are you?", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "c113ed366ab0fba64f6215f41d6fb127", "messageType": "normal", "messageText": "Castling.", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:39:31", "messageStatus": "read", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:39:32", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "I know I can't tell you that.", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "dc821c5eaeacd713cfef5cea15e803040", "messageType": "normal", "messageText": "I know I can't tell you that.", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:40:05", "messageStatus": "unread", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:40:06", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "Do you know that there are people investigating Kim Ill-Song?", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "8197385d4b4222e32ec474fa497b70d8", "messageType": "normal", "messageText": "Of course. However, they will never know it is me behind the bribes.", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:41:47", "messageStatus": "unread", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:41:48", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "At our old meetup spot?", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "e5d6be661c5ed90cfb27a0fb50b3bf2", "messageType": "normal", "messageText": "At our old meetup spot?", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:43:06", "messageStatus": "unread", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:43:07", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "still we should be careful. Pay attention. I want to meet in September at 5pm.", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}, {"success": "messages retrieved", "result": [{"recMessages": [{"messageId": "b5860bde833df4231c31dfbecbedf0d", "messageType": "normal", "messageText": "What day?", "recipientType": "phone", "recipientId": "14068522589", "senderType": "phone", "senderId": "14069243754", "senderName": "Kim Ill-song", "time": "2014-07-02 22:43:44", "messageStatus": "read", "deliveryMethod": "onnet"}, {"sentMessages": [], "brandedSystemMessages": [], "calls": [], "voicemails": [], "now": "2014-07-02 22:43:45", "largestCount": 1, "smsCreditBalance": 0, "callingCreditBalance": 0, "numTextsSent": 0, "numTextsRec": 0, "inviteCount": 0}], "senderId": "14068522589", "senderName": "Ann", "recipientId": "+14069243754", "messageTxt": "I told you to pay attention.", "senderType": "phone", "sendAsSms": 0, "recipientType": "phone"}]
```

Figure 75 Conversation between Ann Dercov and Kim Ill-Song in Raw Format

The conversation from Figure 75 was then converted into a more readable format, which can be seen in Table 7 below. This reveals details of the conversation including Kim Ill-Song's confession of the bribes, their meetup which will occur in September at 5pm which Kim asks is their 'old meetup spot?':

Table 7 Conversation between Ann Dercov and Kim Ill-Song

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

4.2.2.2 Determining Location of the Meeting

The conversation between Ann and Kim in Table 7 detailed interesting information but not all of it because the exact date of their meetup was unknown as well as the location. The only data that was known about their meeting

at this point that it would be in September at 5pm. When Kim asks Ann ‘What day?’ there is a pause until the final message. Examining each of the frames after the former message shows a series of requests made to a domain called ‘mob.mapquestapi.com’ from Ann’s device:

21214	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:36.122678
21108	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:33.466635
21008	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:31.940611
7087	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:45:54.778546
20915	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:29.383551
20686	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:26.922477
7349	192.168.1.5	207.200.102.1	HTTP	255	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:45:58.690915
20593	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:24.767288
20496	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:22.682739
20388	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:20.777360
20299	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:50:19.139107
7577	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json...	16:46:01.277751

Figure 76 HTTP Requests

Examining one of these frames reveals a URI parameter of location. The latitude and longitude values are separated by a ‘%2c’. Translated from HEX, this is a comma symbol:

```
▼ Request URI Query: key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.856319427490234%2C-114.01313018798828
Request URI Query Parameter: key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0
Request URI Query Parameter: inFormat=kvp
Request URI Query Parameter: outFormat=json
Request URI Query Parameter: location=46.856319427490234%2C-114.01313018798828
```

Figure 77 Query Parameters within Web Request

Each of the web requests made by ‘GET /geocoding/v1/...’ have similar parameters with latitude and longitude values. With this data, it was theorised that it would be able to give an idea of where the location was using coordinates. Using tshark, the capture was filtered and dumped to a file. This method would allow the data requested to be filtered and ready to be analysed from Python. Tshark is already available in Wireshark using filter capabilities but this is a much more efficient process in this case because the data that is needed is only specifically there, compared to other data available in Wireshark using the GUI:

Figure 78 ‘tshark’ Command to Filter the Capture for HTTP Requests that Have Coordinates

8140	192.168.1.5	+ 207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85824322597682C-114.01863861083984 HTTP/1.1	16:46:12.793430
8266	192.168.1.5	+ 207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8587341308593752C-114.01864624023438 HTTP/1.1	16:46:15.529776
8333	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858844757080082C-114.01864624023438 HTTP/1.1	16:46:15.812780
8425	192.168.1.5	+ 207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8589439392889842C-114.01864624023438 HTTP/1.1	16:46:16.218130
8516	192.168.1.5	+ 207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8590469360351582C-114.01864624023438 HTTP/1.1	16:46:17.051677
8610	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85914993286132C-114.01864624023438 HTTP/1.1	16:46:19.01753
8714	192.168.1.5	+ 207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8594665527343752C-114.01864624023438 HTTP/1.1	16:46:22.778213
8788	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.859577178955082C-114.01864624023438 HTTP/1.1	16:46:22.429991
8899	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.859691619873052C-114.01864624023438 HTTP/1.1	16:46:23.329981
9080	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.859809875488282C-114.01864624023438 HTTP/1.1	16:46:24.447993
9177	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.8599319455808782C-114.01864624023438 HTTP/1.1	16:46:25.518873
9286	192.168.1.5	+ 207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Clua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.860290527343752C-114.01863098144531 HTTP/1.1	16:46:28.827738

Figure 79 Example Entries Provided by Figure 78 tshark command

A Python file was then written that took the entries and extracted the coordinates. These were then formatted correctly within a CSV file:

```

1 file = open("locationData.txt","r")
2 file = file.read()
3 file = file.splitlines()
4 fields = ["Latitude", "Longitude"]
5 location = []
6 for i in file:
7     a = i[-17:]
8     a = a.replace(a, " ")
9     b = a.replace("%2C", ",")
10    c = b.replace("HTTP/1.1", " ")
11    split_data = c.split("location=")
12    location.append(split_data[1])
13
14 for i in range(len(location)):
15     print(location[i])

```

Figure 80 Python Code to Extract Location Data from URLs in Text File

The CSV file (seen in Appendix, Capture 4) was then supplied to a tool called Excel to KML available on Earth Point, which required a CSV file. This would then be ultimately for the purpose of viewing this data, via a Keyhole Markup Language (KML) file, on Google Earth which would allow the exact location of their meetup to be seen.

Excel To KML - Display Excel files on Google Earth.

A user account is recommended for the features on this web page. 

Import a spreadsheet of lat/long coordinates to Google Earth. Pop-up balloons, icons, paths, and polygons are easily created from the spreadsheet data.

Latitude and Longitude are all that is needed to create a basic display on Google Earth. Add a Name, Description, and an Icon for a professional presentation.

Advanced features support [GPS tracks](#), [Time Sliders](#), and [Grid Coordinates](#).

To get started, read the [Quick Start](#) instructions or download the sample data [ExcelToKmlDemo.zip](#).

Icon Text: Display text in the icons on Google Earth.

Polygons: Draw polygons onto Google Earth.

Select an Excel file (xls, xlsx, xlsm, xlsb, txt, or csv)

location_data.csv



	A	B	C	D	E
1	Latitude	Longitude	Name	Description	Icon
2	43°36'34.86"N	116°12'23.30"W	BAM	Art museum	12
3	43 36 33.22	-116 12 18.40	Roses	Nice garden	111
4	43.608879028	-116.20320277	Zoo	Great visit	186

Sample points plotted onto Google Earth.

Figure 81 Usage of Excel to KML

This gave a KML file which was imported onto the website Google Earth. Once imported, it revealed the location of the meetup:

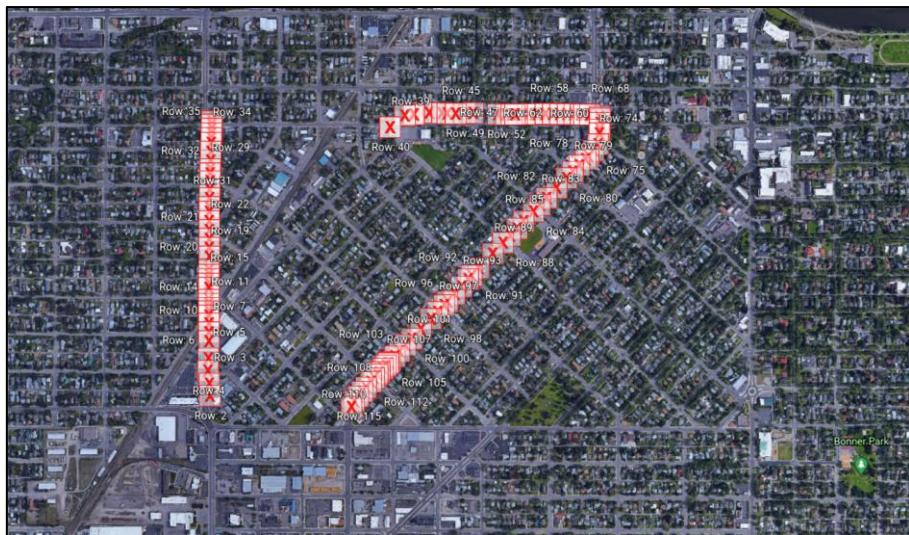


Figure 82 Location given by KML File

Zooming out reveals the location of the meetup will take place near Washington and Montana. Closely zooming in on the data reveals it being near the University of Montana.



Figure 83 Located Near Washington and Montana

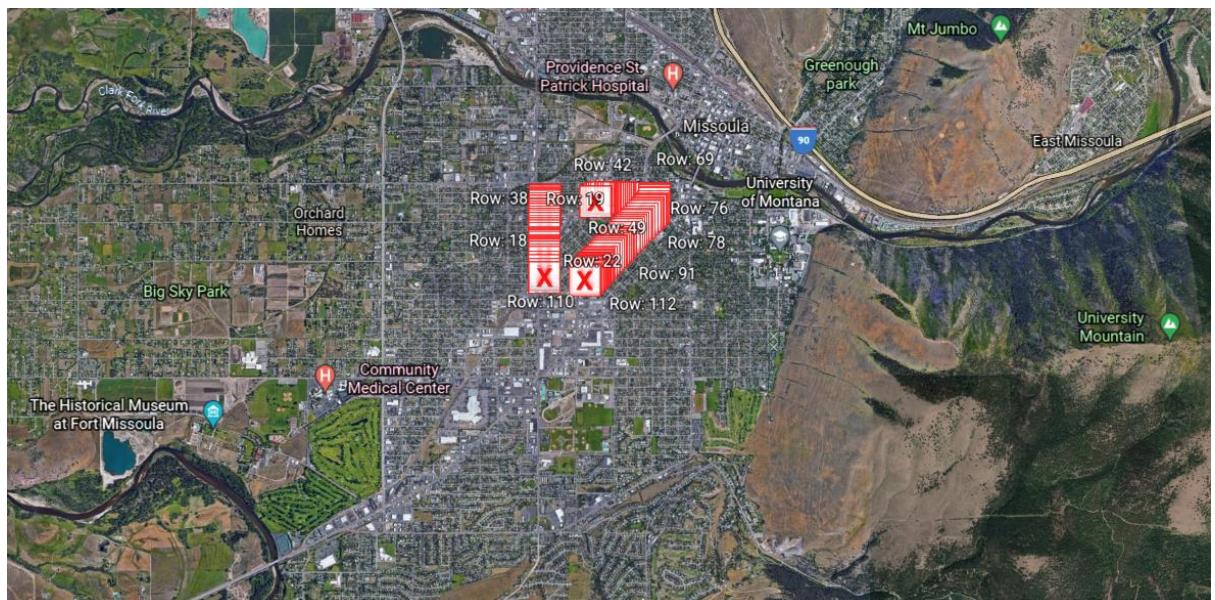


Figure 84 Location near University of Montana

One thing to note when importing the KML document was that while it seemed to have displayed the data and reveal the location, each of the coordinates are marked by a big red X. It wasn't sure why this was happening, so

it was ignored. When viewing Figure 82 it reveals a number '17'. This could suggest that the date of the meeting is the 17th September at 5pm between the area of the University of Montana and the Community Medical Center.

5. Appendix

5.1 Capture 1

5.1.1 Email Conversation

The screenshot shows three captured frames from a network analysis tool, likely NetworkMiner, displaying an email conversation. The frames are arranged vertically.

Frame 1 (Top): This frame shows a single message being sent. The hex dump shows the message content, which includes an HTML email body. The ASCII dump shows the raw text of the email, including the 'From' and 'To' fields. The notes section indicates the message is a multipart message with a boundary of '7de2 582b2033'. It also notes the presence of attachments.

Frame 2 (Middle): This frame shows another message being sent. The hex dump shows the continuation of the email body. The ASCII dump shows the raw text, including the 'Hello' greeting and the recipient's name. The notes section indicates the message is a plain text message.

Frame 3 (Bottom): This frame shows a third message being sent. The hex dump shows the final part of the email body. The ASCII dump shows the raw text, including the closing signature. The notes section indicates the message is a plain text message.

Frame (1342 bytes) | Reassembled TCP (8658225 bytes)

5.2 Capture 2

5.2.1 Encoded Conversations Between Ill Song and Other Members

```

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1
:512a@QKJQ1NMRNjHNRSTUVCRVNBUx0RUJTWFZ@0pPULNXS11E0k1KwFhLNBUJ1T1JVR0@JRFPS1hYRzRERk10MkNBmzNRUIyR1Fa5kFJ1VHSzQzEVcQkc2
NRKRT1pJU0E1MLBPSLdhsU1eVU5GMkd2lkpBtUSYvzIyTe9WNFFISThaQutChFc2M1RIUEZRv=Rawk8=
PING LAG2311957802
:verne.freenode.net PONG verne.freenode.net :LAG2311957802
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG 111_Song :
NTC2NTZjMwMjNDc0Njg2NT1wNjQ2NTYzNjk3MzIwNmU2ZTjMja30TY1nZQyZQ==

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2341989052
PRIVMSG Razor1
:5010W/v1hQR01XWfpaQU5Gw1NBwVRGTVyMekyTdTpv1dDQTVES5Gw1lNBNURK1T1TU0Ez@M0dFQjRXS11M0ZUUZBkIxTTkJRWEE0hKFQrlhYS01EW41Mldz
Wk3BT1jV1dskPUjhTQTVUsk90VwhTSURCT1pTQFaTF1pQ1hVRTkJRW1E1JQ0W0W1pysu1DE41WkdLWdpBTk3RNEdJRfVONFFHN1pUR01W
kH0PPT09
:verne.freenode.net PONG verne.freenode.net :LAG2341989052
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG 111_Song :
NDkyMDYxMwQ/M0YxMjA3NjY1NzI3OT1wNjI3NTczNzkyMDzkjE2ZTjMja2Mjc1NzQyMdcwNjU3MjY4NjE3MdcMjA0OT1wNjM2Zjc1NmM2ND1wNjI2NT1wNzA2N
Tcyh3N3NTyhxjQ2NTY0hjA3MzD2Mj5Njy5NzH20tC0MyDUuNjU2NT1wNj2NjIwNTA0TzNmUzNz5NjE2ZTj3MjA2OTc2zjA3NDY4njUyDcyNjk2h2zY4Nz
QyMDcunM2MTYzNjUyM0YzNnyMjIwNzQ2ODY1MjA1NzZmNzI2YzY0MjA1NDy5NzQ2YzY1MmU=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2371989052
:verne.freenode.net PONG verne.freenode.net :LAG2371989052
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1
:5010W/v1hQR01XWfpaQU5Gw1NBwVRGTVyMekyTdTpv1dDQTVES5Gw1lNBNURK1T1TU0Ez@R0EJFQ1RXU1pUVUg0UUZMzNOTVyyR1EyTE5NMFISTNaQ01U1hj
SURtJvJu0EeM1ZPUVfHN1pQu95VUdLSUNETySfNFBNMVFNTM7wV1Uu0dM1RFRUiyR0MyM0ZFQjRXNjVmu0VwFhPM1JB1pRV0d2TfVOr1hXNE1EvE41V
1dLNTNj7zkaR0tMUT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG 111_Song :
NTM2ZjKzNjU3NjYANjU3MjY1MjA2Ntc4NzA2NzT1NzI2M0tC2NjUyYzIwNDkyM0Y4Nj3MDY1MmUyMa==

PING LAG2402004677
:verne.freenode.net PONG verne.freenode.net :LAG2402004677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG 111_Song :
MjOzNzWmTAyYzWtAzD1wNjk3NDiNjk3Mzj1MjA2MzYxNmUyIDQ5MjA2ZDY1NjU3NDiNzk2Zjc1M2Y=
PING LAG2462020302
:verne.freenode.net PONG verne.freenode.net :LAG2462020302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1 :RzQ9PT09PT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2462020302
:verne.freenode.net PONG verne.freenode.net :LAG2462020302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1 :RzQ9PT09PT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2492176552

```

```

:verne.freenode.net PONG verne.freenode.net :LAG2492176552
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG 111_Song :
MjOzNzWmTAyYzWtAzD1wNjk3NDiNjk3Mzj1MjA2MzYxNmUyIDQ5MjA2ZDY1NjU3NDiNzk2Zjc1M2Y=
PING LAG2522332802
:verne.freenode.net PONG verne.freenode.net :LAG2522332802
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1 :SkVRSE8yT1OUVfHRVpkQu5GwENBNURQT1Z5V1fJRFh0rJHUU1EVUSCU1NBwUxFTVJaR0s0M1RGWT09PT09PQ==

PING LAG2552379677
:verne.freenode.net PONG verne.freenode.net :LAG2552379677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Razor1 :SkVRSE8yT1OUVfHRVpkQu5GwENBNURQT1Z5V1fJRFh0rJHUU1EVUSCU1NBwUxFTVJaR0s0M1RGWT09PT09PQ==

PING LAG2582770302
:verne.freenode.net PONG verne.freenode.net :LAG2582770302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2613723247
:verne.freenode.net PONG verne.freenode.net :LAG2613723247
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2645629677
:verne.freenode.net PONG verne.freenode.net :LAG2645629677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Genius1
:SUZaU0E1w0ZfQ1NHhjQzRE5Wm1hHuKcFRUTVw0w0E10R1N1YRUxQCuUpFwKxNlTkZTwE1aSkFKRvFhMjJMSE5CkhbwVRGRUJRV0UzREZFQjJHnk1EsU1wV0h8
SURtJvJu0E01M0pPULvQ0TzNUE9MwKHENDNgTUZaR0cQk8=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2675785927
:verne.freenode.net PONG verne.freenode.net :LAG2675785927
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
:Genius1!~malware@216.14.247.46 PRIVMSG 111_Song :
MTEXidA0@CaxNjh@TQ1IDE0NSAwNTyM0Qw0IDExNCAxNTAgHTQ1IDE1NiAwNDAgHTY3IDE0NSAwNjMgMNTY0IDA0MCaxNTUgMTQ1IDE0NSAxN
j0gMDw0IDA0@CaxNjh@TQ1IDE0NSAwNTyM0Qw0IDExNCAxNTAgHTY3IDE0NSAwNjMgMNTY0IDA0MCaxNTUgMTQ1IDE0NSAxNj0gHTUw1DE0NSAwDAg@HTY2IDE0MS
AxTQg@HTUw1DE0@CaxNjh@HTY1DE0MSAwNDAgMTU3IDE0NiAwNDAgHTY0IDE1MCaxNTEgHTYzIDA0@CaxNjh@HTU0IDE0MSAxNTEg@HTU1IDA1Ng==

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PING LAG2705942177
:verne.freenode.net PONG verne.freenode.net :LAG2705942177
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 111_Song :razor genius
PRIVMSG Genius1
:SkVR0d2TE9fQ1JHs01Es5k5ZUuHt0@HTUuyw0PRERNSVpXRU1Kwkc1UVdLTkxETVUyR0VZTED1QJTEQ05M0QkdN1IRlwVJBTzVViEkyREp0wVFISTJERkVCM1dL
WikxMRLk9PT09PT0=
PING LAG2735043177

```

:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Genius1
:SkVRr0dTE9Pf0Hs01EsK5ZUdHT0&HTUuyV0/PRERNsVpXRU1Kvkc1UVdLTkxETVuyR0VZTEdIQTJEQ05M0kdNm1RLwVJBtZVViEkyREp0W/FISTJERkVCm1dL
Wd0R1k9PT09PT=0
PING LAG2735942177
:verne.freenode.net PONG verne.freenode.net :LAG2735942177
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:genius1+malware@216.14.247.46 PRIVMSG Ill_Song :
MTE2IDE1NyAwNTYgIDQwIDE1NxAnNTcgMTY0IDA0MCaxNTgAQt01IDE2MiAxNDUgMDU2IDA0MCaxMDMgMTQxIDE1NiAwNDAgMTExIDA0MCaxNTYgMTU3IDE2NCawN
DgAtHQ3IDE1NyAwMDAgTTy0IDE1NyAwMDAgTcxzDE1NyAwNjUgh0c3
PING LAG2765957882
:verne.freenode.net PONG verne.freenode.net :LAG2765957882
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2796176552
:verne.freenode.net PONG verne.freenode.net :LAG2796176552
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Genius1
:SkVRr0Mz2SkFNn1RIRvIMSk1RJuHjMkRCT1FRSEBzM1ZOU1NDQ0VlKvCm1c0NTNTK05UzRJQ0pFQjNXUzNETUvCwl1dLM1RFRU10VzY1skFNRVFM1pMV90UvD
WkpBT2VmEkyQfPUlVhs01ERu10MkdLSURCT1pTQ0EzRFBNtlyFyStMjE5ZUuhJmkRTTjUyV08yQkFNRVFM1jMz01VuvUhVhCxET1ZaR0tJRedONvphMk1EUE1zu
UdhHzNot1LyVqzTERNNrjJuHzT0ZzTP09PT09
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2826285927
:verne.freenode.net PONG verne.freenode.net :LAG2826285927
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2856348427
:verne.freenode.net PONG verne.freenode.net :LAG2856348427
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
PRIVMSG Method
:Sl2aQzRjQ5NwJhUTMzRUZUUVTSURCT1vRR0s2RER0RjJHS1pCQU1GUkc2NUxvXRUiYr1FaSkFPQ1pHnjQzUu1w1hJSURQTV1RSEkyREZFQkJXUVpMVE9NUUV
DmNZTzYR9RJRfHOnWphMvCQ95SVhJmBRGRUsVzJyTep0w1lRTQTVUeUECSuHtmNPTTU0V0MzVehGwt09PT09PQ==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
:Method+malware@216.14.247.46 PRIVMSG Ill_Song :
NDKyD0Yxh0mQyY3NDInWz3NtcyNjY0MDc3NjzGzJTwzK2zJc1MjA2MtcyNjUyZiWnj13NtC0MjA00TiWnjg2Mtc2NjUyMDYxNmUyMDY5NjQ2NTYxMmUyM
DQ1NjksD0N4j1NjU1jWmz5MwlyDQ5Mh2TzKjhA2ZTzNzQyMDySnmU3NDY1NzI2NtzCnzQ2NTY0Mmu=

:ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PING LAG13216520302
:verne.freenode.net PONG verne.freenode.net :LAG13216520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PING LAG13216520302
:verne.freenode.net PONG verne.freenode.net :LAG13216520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
Raekwon!-malware@216.14.247.46 PRIVMSG Ill_Song :
Dfry@DY4Nje3N1wYMdyNzU3NDy@NDy@23MmZ2TkyNy@NyDyNyjN@NyDyNmY3NTy3Njg3NDiWnZM2ZjIwNjU2MTczNjk2Yz5Mu=

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG13216520302
:verne.freenode.net PONG verne.freenode.net :LAG3216520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PRIVMSG Raekwon
:Sup#etaMUTDUTQVQ:R@0WuTT85TU8e:VFVBPUhDQdVmuE9VuUDnfNFRGJURVzRJRFBNw1lRHu1kzSk1GV0NBmzNPRIUiY1FaSkFNVjRH51kzVh9SVvhN
WkpBtUSVYz1zEpUjHs1pkQ41vENBNUr3T7VVRVNMvNKRJhWdqNQKuQ5M1hNuJU8TzVRzQ10kPQRlhys1leV40UUdX1lR0TzRRSEkyErJPVfFU01eQk5VU
DfRukxTtVRSERkzVfQ001NxTRCQ5WuDUkxpVZyNEs@0kFn1lnXrzJmVE5GwfC05URCT01R0tZtFtRQFVHQzRaQu9CwfHHDNOKTUPR0tMuKE=

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
Raekwon!-malware@216.14.247.46 PRIVMSG Ill_Song :
Dfry@DY4Nje3N1wYMdyNzU3NDy@NDy@23MmZ2TkyNy@NyDyNyjN@NyDyNmY3NTy3Njg3NDiWnZM2ZjIwNjU2MTczNjk2Yz5Mu=

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG3246520302
:verne.freenode.net PONG verne.freenode.net :LAG3246520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG3277410927
:verne.freenode.net PONG verne.freenode.net :LAG3277410927
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PRIVMSG Raekwon
:SUSV#Q0M#P#H#N#Y#R#L#S#R#U#D#J#2#P#T#V#Q#F#T#S#F#P#N#V#W#T#C#Q#U#0#1#C#W#K#B#U#Z#Y#H#J#T#R#F#V#Q#1#T#Q#T#J#H#T#0#1#A#W#F#M#0#C#T#1#J#V#Z#Y#U#K#F#N#1#h#Y#R#U#E#V#S#C#U#1#N#B#
W#K#R#T#J#T#Z#Q#Z#I#Z#W#D#N#F#Q#T#M#S#K#5#M#K#N#B#D#N#Q#T#J#V#Y#Q#P#T#=

ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG3309160927
:verne.freenode.net PONG verne.freenode.net :LAG3309160927
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG3339317177
:verne.freenode.net PONG verne.freenode.net :LAG3339317177
ISON Razor Genius Raekwon Killah Method

5.3 Capture 4

5.3.1 locationData.csv

Latitude	Longitude
46.85661316	-114.0186081
46.85693359	-114.018631
46.8572731	-114.0186844
46.85760117	-114.0186691
46.85805511	-114.0186462
46.85828781	-114.0186462
46.85852432	-114.0186386
46.85873413	-114.0186462
46.85884476	-114.0186462
46.85894394	-114.0186462
46.85904694	-114.0186462
46.85914993	-114.0186462
46.85946655	-114.0186462
46.85957718	-114.0186462
46.85969162	-114.0186462
46.85980988	-114.0186462
46.85993195	-114.0186462
46.86029053	-114.018631
46.86052322	-114.0186386
46.86075592	-114.018631
46.86098862	-114.018631
46.86122894	-114.0186386
46.8614769	-114.018631
46.86159897	-114.018631
46.86183548	-114.0186234
46.86206436	-114.0186157
46.8622818	-114.0186005
46.86248779	-114.0186005
46.86260223	-114.0185928
46.8628273	-114.0185776
46.86306381	-114.0185776
46.86330032	-114.0185623
46.86342621	-114.0185547
46.86355209	-114.0185471
46.86367416	-114.0185394
46.86378098	-114.0185394
46.86387253	-114.0185318
46.86370468	-114.0116425
46.8637085	-114.0116348
46.86401749	-114.0110703
46.86404419	-114.0107498

46.864048	-114.0107117
46.86408997	-114.0104218
46.86408997	-114.0101242
46.86407852	-114.0096283
46.86407089	-114.0094223
46.86406708	-114.0091019
46.86407471	-114.0087585
46.86408234	-114.0084229
46.86405182	-114.0074692
46.86404419	-114.007164
46.86404419	-114.0069427
46.864048	-114.0068054
46.86405563	-114.0067062
46.86405182	-114.0066223
46.86405182	-114.0064621
46.86405182	-114.006279
46.86405182	-114.0060577
46.86405182	-114.005928
46.86405945	-114.0056305
46.86405945	-114.0053406
46.86405563	-114.0050659
46.86405182	-114.004776
46.86405182	-114.0045242
46.86404419	-114.0042725
46.86404419	-114.0041428
46.86404037	-114.0039215
46.86398315	-114.0035477
46.86393356	-114.0035172
46.86381912	-114.0035248
46.86364365	-114.00354
46.86354446	-114.0035477
46.86325455	-114.0036011
46.86309052	-114.0037689
46.86293411	-114.0039673
46.86286163	-114.0040817
46.86270142	-114.0043259
46.86253357	-114.0045776
46.86236191	-114.0048141
46.86210632	-114.0052032
46.86183548	-114.0055923
46.86166	-114.0058441
46.86148453	-114.0060959
46.86122131	-114.0064774
46.86103058	-114.0067291
46.86084366	-114.0069962
46.86065674	-114.0072708

46.86037064	-114.0076675
46.85998917	-114.0082092
46.8597908	-114.0084839
46.85969162	-114.0086212
46.85950089	-114.008873
46.85930252	-114.00914
46.85910416	-114.0094147
46.85900879	-114.0095444
46.8588295	-114.0097961
46.85864639	-114.0100555
46.85837555	-114.0104446
46.85812378	-114.0107956
46.85795212	-114.0110397
46.85778809	-114.0112762
46.85765839	-114.0114517
46.85751343	-114.0116425
46.85749054	-114.0116882
46.85747147	-114.0117111
46.85741806	-114.011795
46.85733414	-114.0119095
46.85723495	-114.0120468
46.85718155	-114.0121231
46.85708237	-114.0122528
46.85697937	-114.0123749
46.85683441	-114.0125656
46.85672379	-114.0127106
46.8565979	-114.0128708
46.85647202	-114.0130234
46.85631943	-114.0131302

References

- Anthony, A., 2022. *How to extract HTTP and FTP files from Wireshark *.pcap file.* [Online] Available at: <https://adriananthony.wordpress.com/2019/07/19/how-to-extract-http-and-ftp-files-from-wireshark-pcap-file/>
- CERT NetSA Security Suite, 2021. *SiLK.* [Online] Available at: <https://tools.netsa.cert.org/silk/>
- CERT NetSA Security Suite, 2021. *YAF Core Library.* [Online] Available at: <https://tools.netsa.cert.org/yaf/libyaf/index.html>
- Chen, K., 2017. *5 Ways to Generate and Verify MD5 SHA Checksum of Any File in Windows 10.* [Online] Available at: <https://www.nextofwindows.com/5-ways-to-generate-and-verify-md5-sha-checksum-of-any-file-in-windows-10> [Accessed 22 12 2021].
- Crackstation, 2019. *Crackstation.* [Online] Available at: <https://crackstation.net/>
- Earth Point, 2022. *Excel To KML - Display Excel files on Google Earth.* [Online] Available at: <https://www.earthpoint.us/ExcelToKml.aspx>
- GCHQ, 2021. *Cyberchef.* [Online] Available at: <https://gchq.github.io/CyberChef/>
- Google, 2021. *Google Translate.* [Online] Available at: <https://translate.google.com/>
- Google, 2022. *Google Earth.* [Online] Available at: https://www.google.co.uk/intl/en_uk/earth/
- Jithin, 2016. *What is MIME (Multi-Purpose Internet Mail Extensions).* [Online] Available at: <https://www.interserver.net/tips/kb/mime-multi-purpose-internet-mail-extensions/>
- Software Engineering Institute, 2020. *SILK Documentation.* [Online] Available at: <https://tools.netsa.cert.org/silk/> [Accessed 22 12 2021].
- Suite, C. N. S., 2021. *SiLK.* [Online] Available at: <https://tools.netsa.cert.org/silk/rwuniq.html>
- Suite, C. N. S., 2021. *SiLK.* [Online] Available at: <https://tools.netsa.cert.org/silk/rwstats.html>
- Suite, C. N. S., 2021. *SiLK.* [Online] Available at: <https://tools.netsa.cert.org/silk/rwsort.html>
- Wireshark, 2021. *Wireshark.* [Online] Available at: <https://www.wireshark.org/> [Accessed 6 December 2021].