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 Dercover' – 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

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:

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:

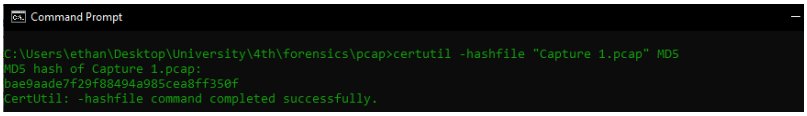


Figure 2 Hash Checksum recorded using certutil

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:

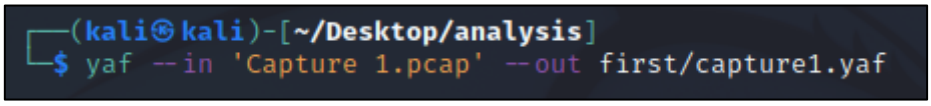


Figure 3 Converting First Capture File into YAF Format

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:

Figure 4 Viewing Entries in First Capture

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

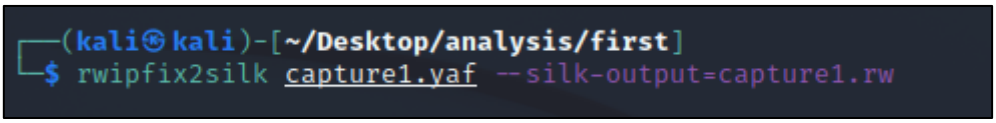


Figure 5 Converting YAF File to SiLK Format

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:

```
┌──(kali㉿kali)-[~/Desktop/analysis/first]
└─$ 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:

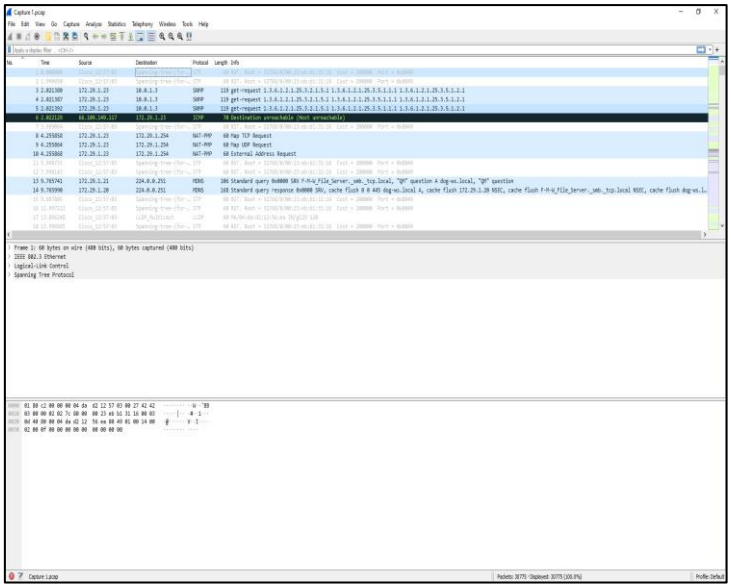
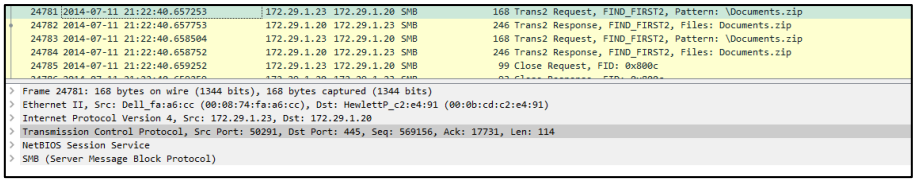


Figure 9 Example Output of Capture 1 in Wireshark

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':

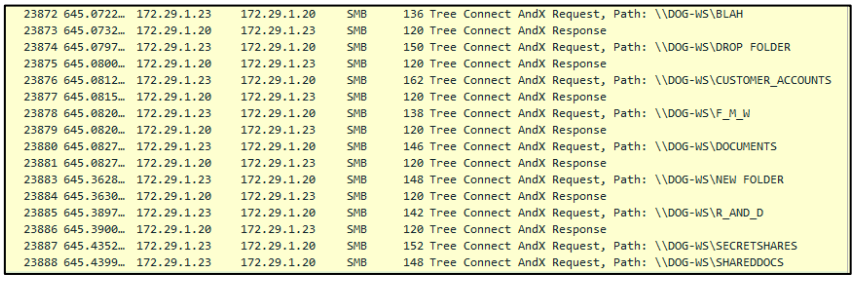


```
24781 2014-07-11 21:22:40.657253  172.29.1.23 172.29.1.20 SMB  168 Trans2 Request, FIND_FIRST2, Pattern: \Documents.zip
24782 2014-07-11 21:22:40.657753  172.29.1.20 172.29.1.23 SMB  246 Trans2 Response, FIND_FIRST2, Files: Documents.zip
24783 2014-07-11 21:22:40.658504  172.29.1.23 172.29.1.20 SMB  168 Trans2 Request, FIND_FIRST2, Pattern: \Documents.zip
24784 2014-07-11 21:22:40.658752  172.29.1.20 172.29.1.23 SMB  246 Trans2 Response, FIND_FIRST2, Files: Documents.zip
24785 2014-07-11 21:22:40.659252  172.29.1.23 172.29.1.20 SMB   99 Close Request, FID: 0x800c

> Frame 24781: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits)
> Ethernet II, Src: Dell_fa:a6:cc (00:08:74:fa:a6:cc), Dst: HewlettP_c2:e4:91 (00:0b:cd:c2:e4:91)
> Internet Protocol Version 4, Src: 172.29.1.23, Dst: 172.29.1.20
> Transmission Control Protocol, Src Port: 50291, Dst Port: 445, Seq: 569156, Ack: 17731, Len: 114
> NetBIOS Session Service
> SMB (Server Message Block Protocol)
```

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:



```
23872 645.0722…  172.29.1.23   172.29.1.20   SMB   136 Tree Connect AndX Request, Path: \\DOG-WS\BLAH
23873 645.0732…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23874 645.0797…  172.29.1.23   172.29.1.20   SMB   150 Tree Connect AndX Request, Path: \\DOG-WS\DROP FOLDER
23875 645.0800…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23876 645.0812…  172.29.1.23   172.29.1.20   SMB   162 Tree Connect AndX Request, Path: \\DOG-WS\CUSTOMER_ACCOUNTS
23877 645.0815…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23878 645.0820…  172.29.1.23   172.29.1.20   SMB   138 Tree Connect AndX Request, Path: \\DOG-WS\F_M_W
23879 645.0820…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23880 645.0827…  172.29.1.23   172.29.1.20   SMB   146 Tree Connect AndX Request, Path: \\DOG-WS\DOCUMENTS
23881 645.0827…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23883 645.3628…  172.29.1.23   172.29.1.20   SMB   148 Tree Connect AndX Request, Path: \\DOG-WS\NEW FOLDER
23884 645.3630…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23885 645.3897…  172.29.1.23   172.29.1.20   SMB   142 Tree Connect AndX Request, Path: \\DOG-WS\R_AND_D
23886 645.3900…  172.29.1.20   172.29.1.23   SMB   120 Tree Connect AndX Response
23887 645.4352…  172.29.1.23   172.29.1.20   SMB   152 Tree Connect AndX Request, Path: \\DOG-WS\SECRETSHARES
23888 645.4399…  172.29.1.23   172.29.1.20   SMB   148 Tree Connect AndX Request, Path: \\DOG-WS\SHAREDDOCS
```

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':

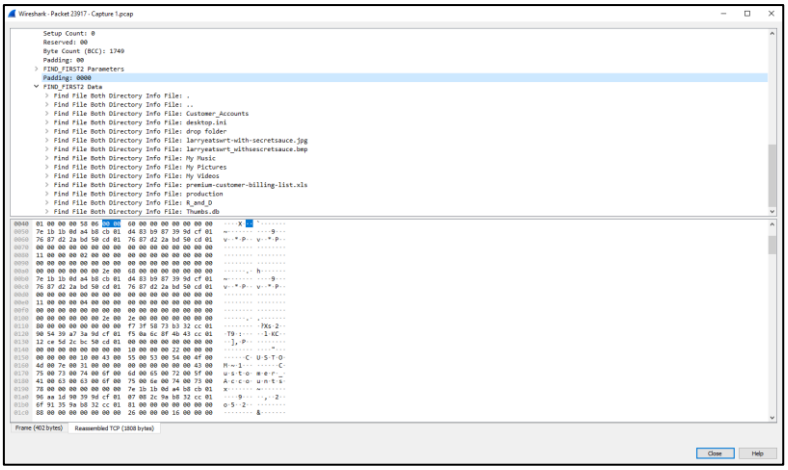


Figure 12 Other Files and Folders seen via SMB

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:

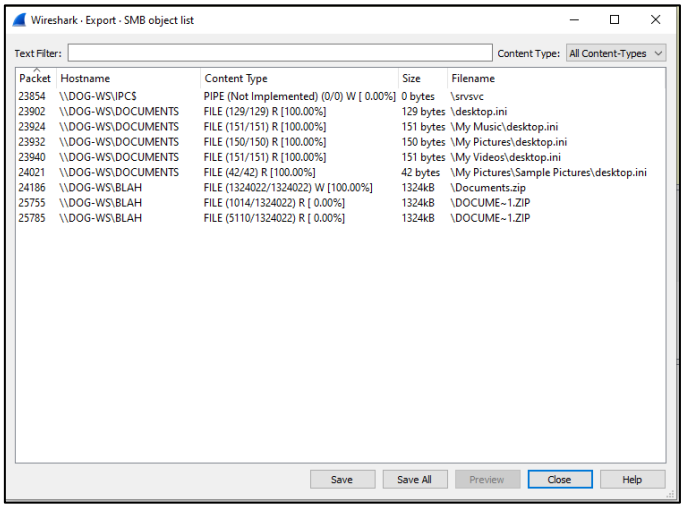


Figure 13 Using Wireshark's Export Feature to Obtain Some of the Recorded Files – Documents.zip, DOCUME~1.zip

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

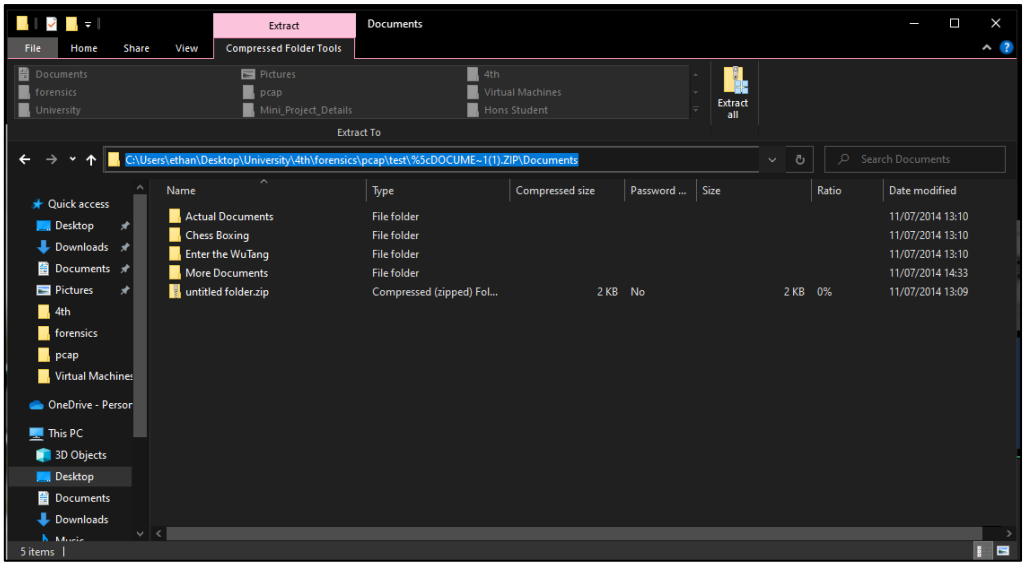


Figure 14 Contents of 'Documents'

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

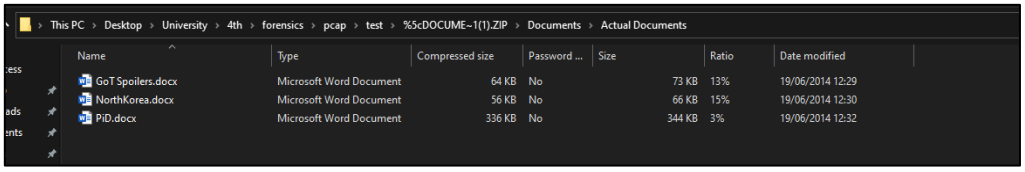


Figure 15 Contents of 'Actual Documents'

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

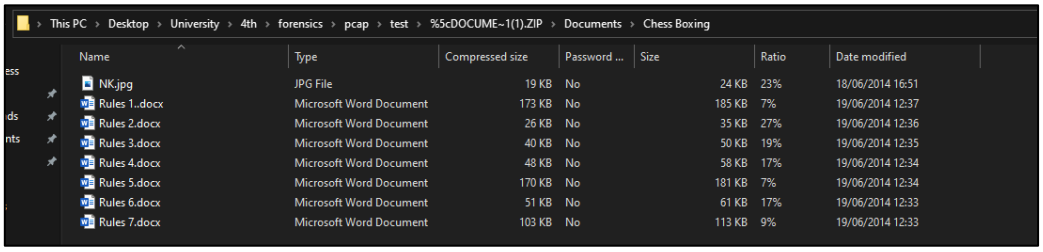


Figure 16 Contents of 'Chess Boxing'

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

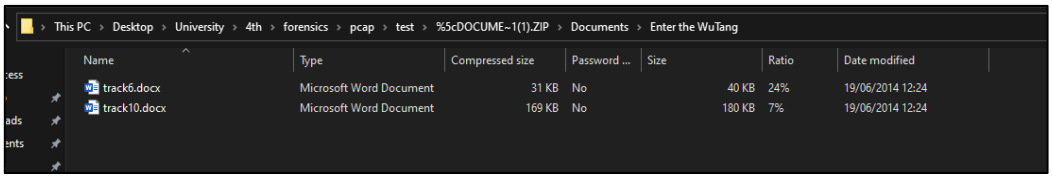


Figure 17 Contents of Other 'Documents'

Within the same ZIP, under 'More Documents', two files:

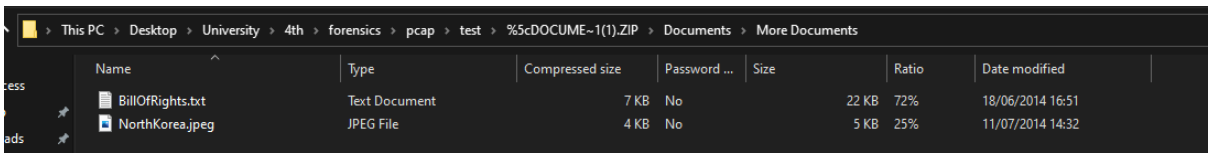


Figure 18 Contents of 'More Documents'

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

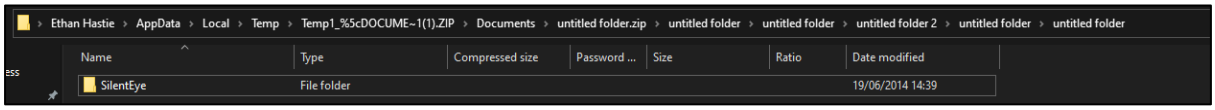


Figure 19 'SilentEye' Folder which Contained no Data

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:

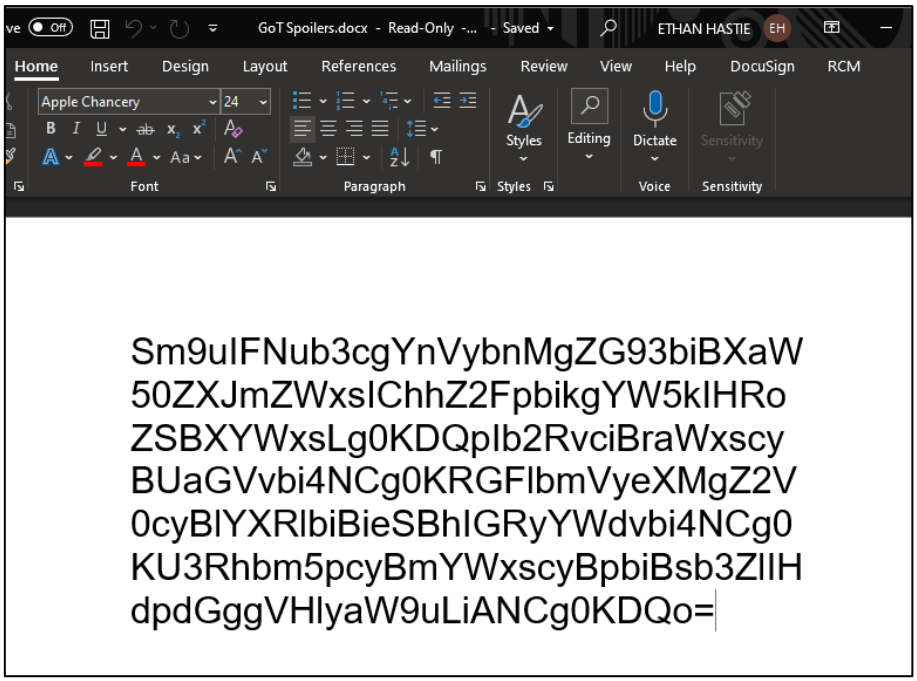


Figure 20 Base64 Encoded Message inside GoT Spoilers.docx

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

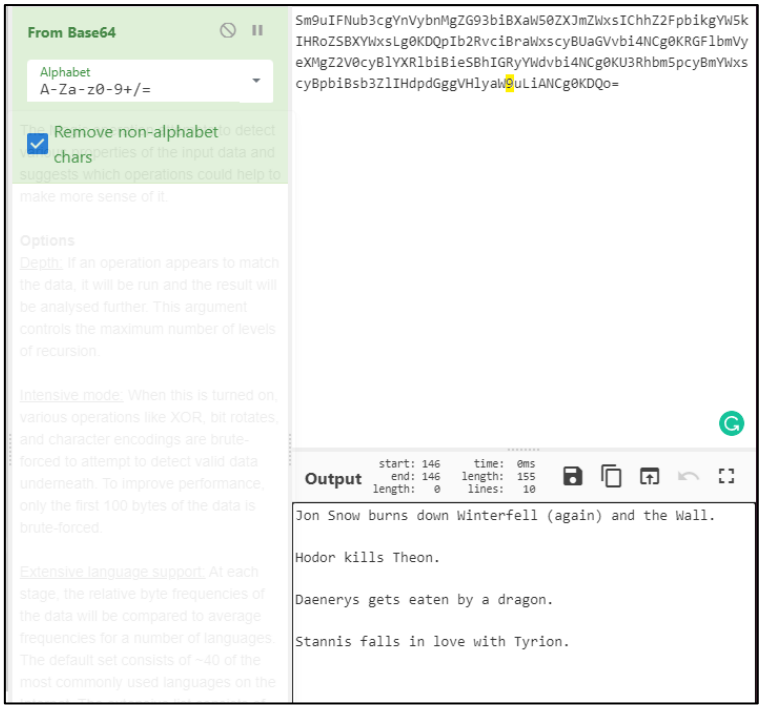


Figure 21 Translation of GoT Spoilers.docx

1.2.2.2 NorthKorea.docx

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

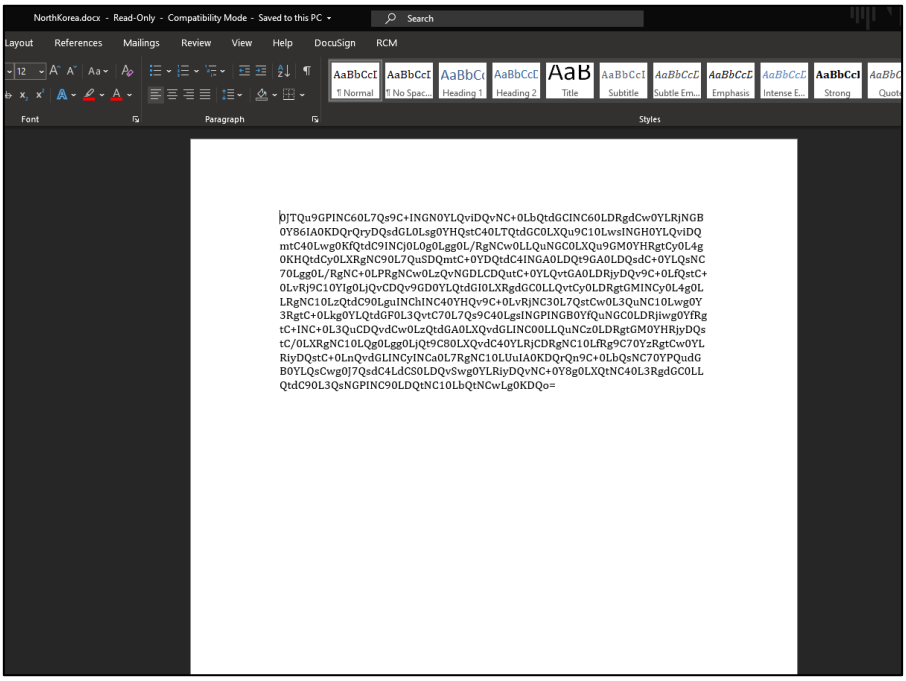


Figure 22 NorthKorea.docx

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

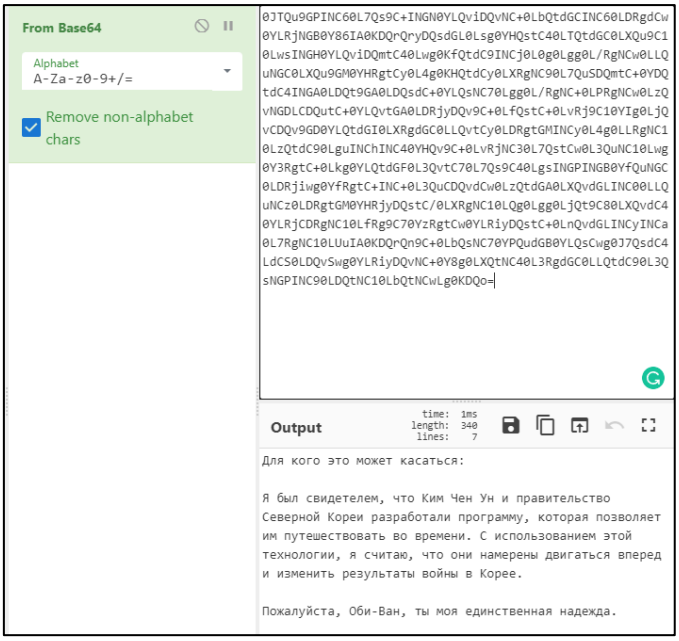


Figure 23 Decoded Message

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

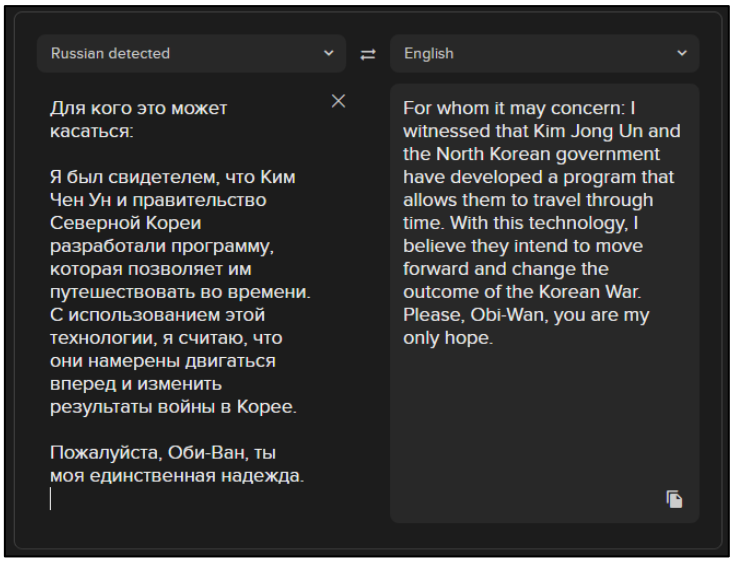


Figure 24 Message Translated from Russian to English

1.2.2.3 PiD.docx

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

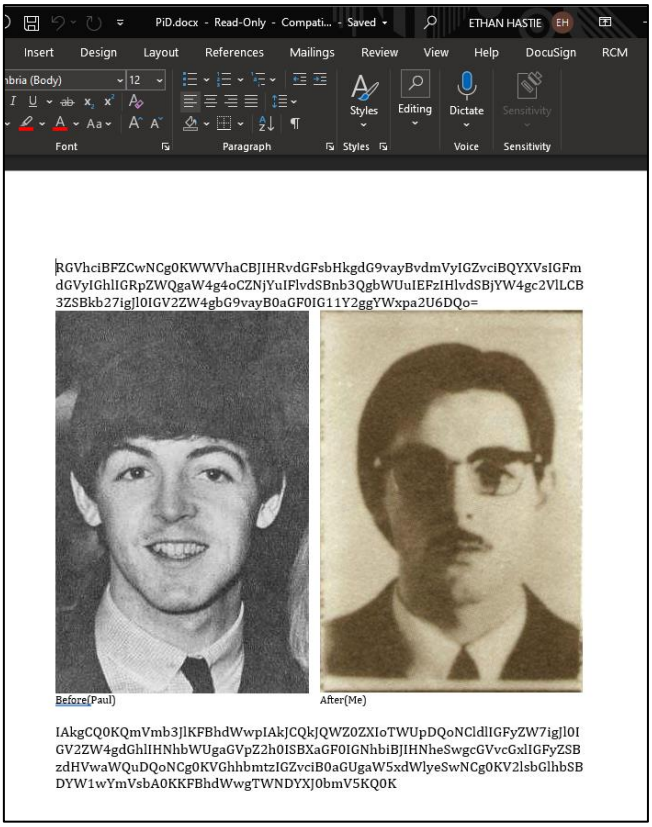


Figure 25 Contents of PiD.docx

This, translated reveals a message to ‘Ed’:

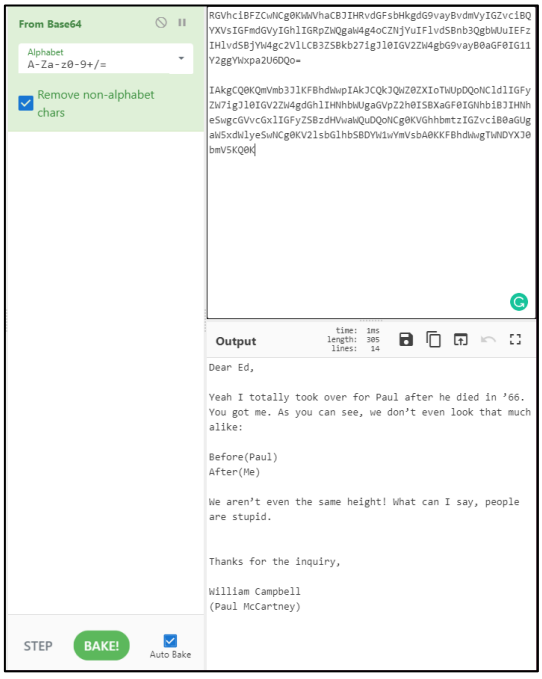
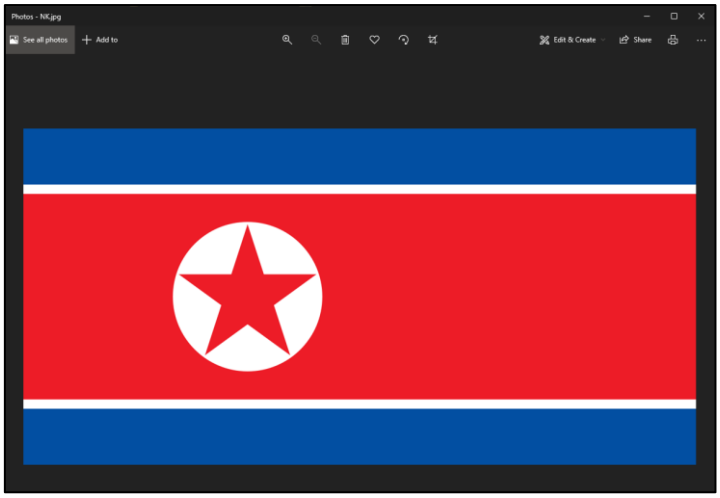


Figure 26 A Message to Ed from William Campbell

1.2.2.4 NK.jpg

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



1.2.2.5 Rules1.docx

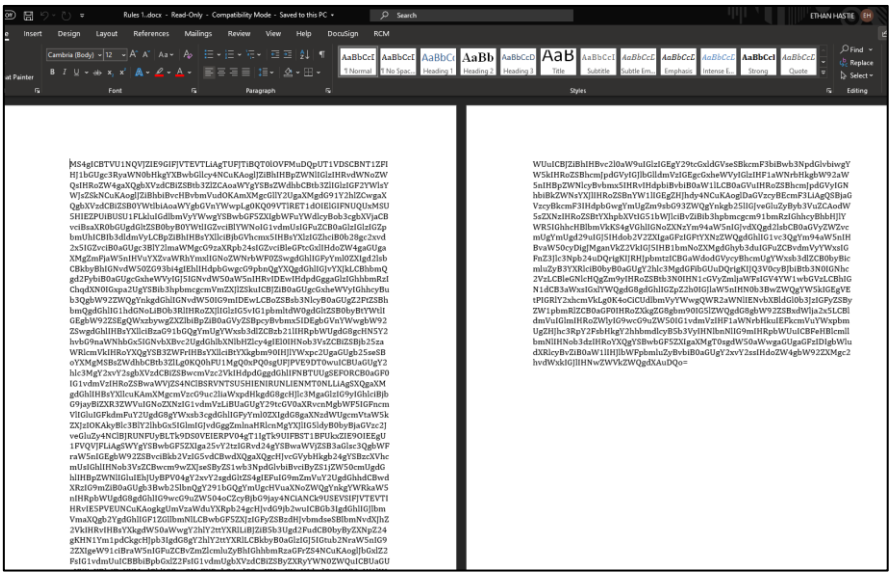


Figure 27 Rules1.docx

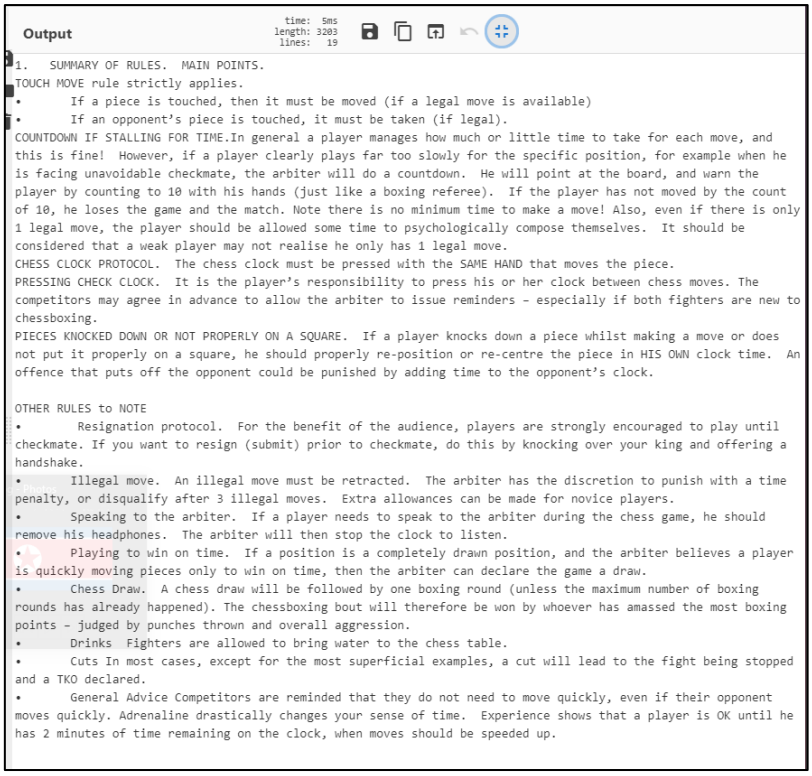


Figure 28 Rules1.docx Translated from Base64

Figure 28 reveals rules from the chess competition.

1.2.2.6 track6.docx

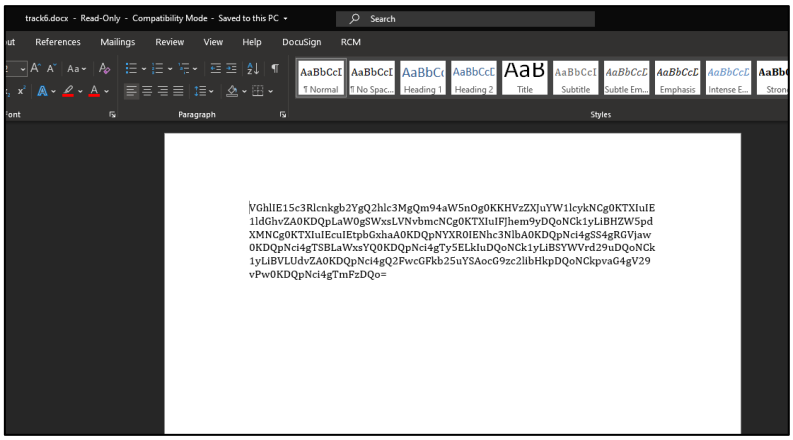


Figure 29 Contents of track6.docx

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

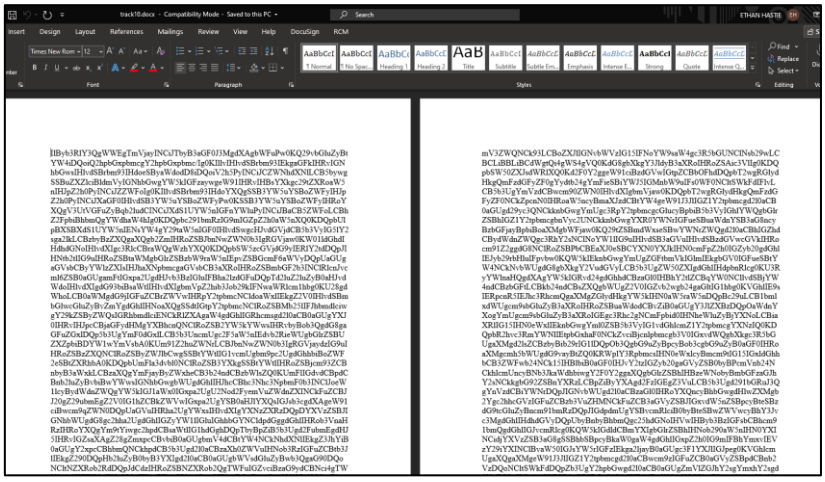


Figure 31 Contents of track10.docx

```
"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 uhh, 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:

Wireshark · Protocol Hierarchy Statistics · Capture 1.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	1070	100.0	662511	7488	0	0	0
Ethernet	100.0	1070	2.3	14980	169	0	0	0
Internet Protocol Version 4	100.0	1070	3.2	21400	241	0	0	0
Transmission Control Protocol	100.0	1070	94.5	626063	7076	0	0	0
Hypertext Transfer Protocol	100.5	1075	2069.8	13712323	154k	616	634363	7170
Portable Network Graphics	1.4	15	23.5	155633	1759	15	158433	1790
Online Certificate Status Protocol	3.6	38	6.6	43543	492	38	51493	582
MIME Multipart Media Encapsulation	0.1	1	1306.3	8654056	97k	1	8658225	97k
Media Type	1.5	16	26.4	174849	1976	16	159761	1805
Line-based text data	15.8	169	113.1	749162	8468	169	341201	3856
JPEG File Interchange Format	1.3	14	51.3	340192	3845	14	343343	3880
JavaScript Object Notation	5.1	55	5.9	39200	443	55	50627	572
HTML Form URL Encoded	4.3	46	4.3	28759	325	46	218446	2469
Data	0.9	10	452.8	3000000	33k	10	3005170	33k
Compuserve GIF	8.4	90	1.0	6680	75	90	7042	79

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':

```
MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----------------------------7de2582b20336"
    [Type: multipart/form-data]
    First boundary: ----------------------------7de2582b20336\r\n
  Encapsulated multipart part:  (application/x-zip-compressed)
      Content-Disposition: form-data; name="file0"; filename="DocsPcap.zip"\r\n
      Content-Type: application/x-zip-compressed\r\n\r\n
    Media Type
        Media type: application/x-zip-compressed (8652619 bytes)
    Boundary: \r\n----------------------------7de2582b20336\r\n
  Encapsulated multipart part:  (application/octet-stream)
      Content-Disposition: form-data; name="file1"; filename=""\r\n
      Content-Type: application/octet-stream\r\n\r\n
    Boundary: \r\n----------------------------7de2582b20336\r\n
  Encapsulated multipart part:
      Content-Disposition: form-data; name="requests"\r\n\r\n
    Data (881 bytes)
        Data: 5b7b2246726f6d223a22736e6f776564696e6564776172644061 6f6c2e636f6d222c2254…
        [Length: 881]
    Boundary: \r\n----------------------------7de2582b20336\r\n
```

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:

Figure 35 MD5 Hash on Second PCAP File

The same was performed on the Windows machine using certutil:

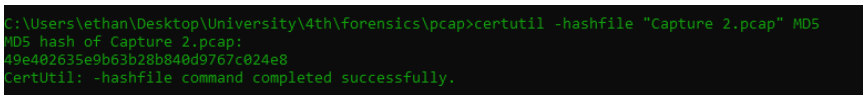


Figure 36 MD5 Hash on Second PCAP File Using Windows Tool

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	23.73.203.32	443	1	40	1
40	1	2014/06/17T21:10:09	2014/06/17T21:10:09	172.29.1.21	58305	184.31.18.110	443	1	40	1
40	1	2014/06/17T21:10:16	2014/06/17T21:10:16	172.29.1.21	32873	31.13.70.40	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:

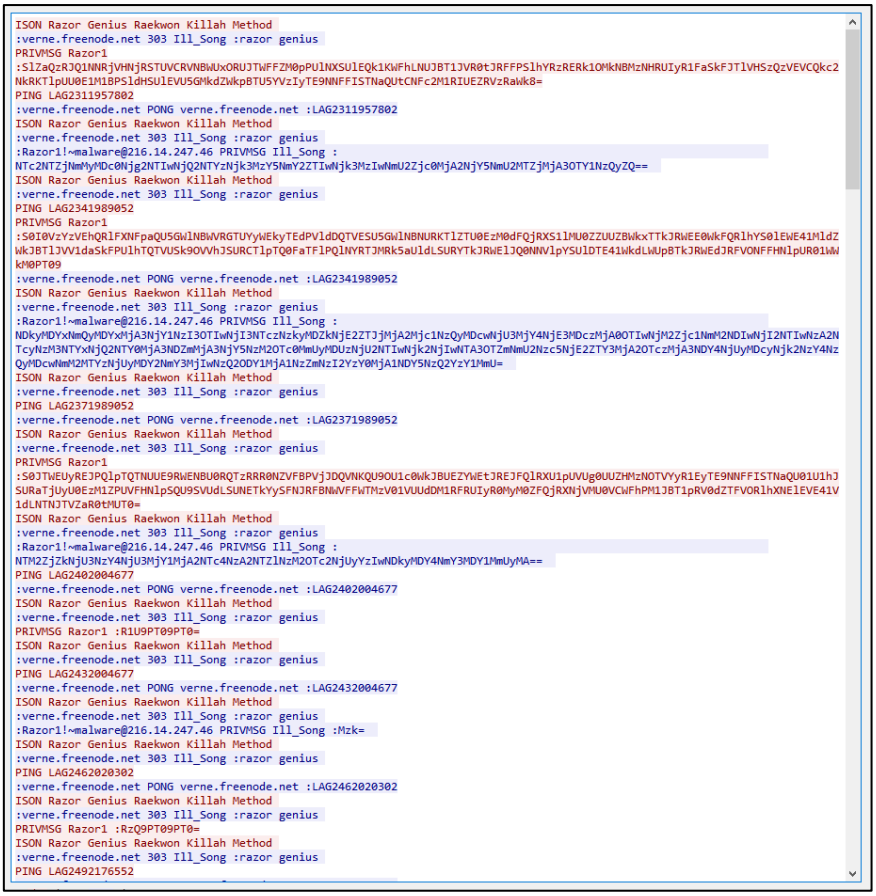


Figure 40 Sample Encoded Conversations of IRC Traffic

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	SlZaQzRJQ1NNRjVHNjRSTUVCRVNBWUxORUJTWFFZM0pPUlNXSUlEQk1KWFhLNUJBT1JVR0tJRFFPSlhYRzRERk1OMkNBMzNHRUIyR1FaSkFJTlVHSzQzVEVCQkc2NkRKTlpUU0E1	Mr. Razor, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang.	Base64, Base32

		M1BPSldHSUlEVU5GMkdZWkpBTU5YVzIyTE9NNFFISTNaQUtCNFc2M1RIUEZRVzRaWk8=		
Razor1	Ill_Song	NTc2NTZjNmMyMDc0Njg2NTIwNjQ2NTYzNjk3MzY5NmY2ZTIwNjk3MzIwNmU2Zjc0MjA2NjY5NmU2MTZjMjA3OTY1NzQyZQ==	Well the decision is not final yet.	Base64
Ill_Song	Razor1	S0I0VzYzVEhQRlFXNFpaQU5GWlNBWVRGTUYyWEkyTEdPVldDQTVESU5GWlNBNURKTlZTU0EzM0dFQjRXS1lMU0ZZUUZBWkxTTkJRWEE0WkFQRlhYS0lEWE41MldZWkJBTlJVV1daSkFPUlhTQTVUSk9OVVhJSURCTlpTQ0FaTFlPQlNYRTJMRk5aUldLSURYTkJRWElJQ0NNVlpYSUlDTE41WkdLWUpBTkJRWEdJRFVONFFHNlpUR01WWkM0PT09	Pyongyang is beautiful this time of year. Perhaps you would like to visit and experience what Best Korea has to offer.	Base64, Base32
Razor1	Ill_Song	NDkyMDYxNmQyMDYxMjA3NjY1NzI3OTIwNjI3NTczNzkyMDZkNjE2ZTJjMjA2Mjc1NzQyMDcwNjU3MjY4NjE3MDczMjA0OTIwNjM2Zjc1NmM2NDIwNjI2NTIwNzA2NTcyNzM3NTYxNjQ2NTY0MjA3NDZmMjA3NjY5NzM2OTc0MmUyMDUzNjU2NTIwNjk2NjIwNTA3OTZmNmU2Nzc5NjE2ZTY3MjA2OTczMjA3NDY4NjUyMDcyNjk2NzY4NzQyMDcwNmM2MTYzNjUyMDY2NmY3MjIwNzQ2ODY1MjA1NzZmNzI2YzY0MjA1NDY5NzQ2YzY1MmU=	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
Ill_Song	Razor1	S0JTWEUyREJPQlpTQTNUUE9RWENBU0RQTzRRR0NZVFBPVjJDQVNKQU9OU1c0WkJBUEZYWEtJREJFQlRXU1pUVUg0UUZHMzNOTVYyR1EyTE9NNFFISTNaQU01U1hJSURaTjUyU0EzM1ZPUVFHNlpSQU9SVUdLSUNETkYySFNJRFBNWVFFWTMzV01VUUdDM1RFRUIyR0MyM0ZFQjRXNjVMU0VCWFhPM1JBT1pRV0dZTFVORlhXNElEVE41V1dLNTNJTVZaR0tMUT0=	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
Ill_Song	Razor1	R1U9PT09PT0=	5	Base64, Base32
Razor1	Ill_Song	Mzk=	9	Base64, Hex
Ill_Song	Razor1	RzQ9PT09PT0=	7	Base64, Base32

Razor1	Ill_Song	NTM2ZjZkNjU3NzY4NjU3MjY1MjA2NTc4NzA2NTZlNzM2OTc2NjUyYzIwNDkyMDY4NmY3MDY1MmUyMA==	Somewhere expensive, I hope.	Base64
Razor1	Ill_Song	MjQzNzMwMzAyYzMwMzAzMDIwNjk3NDIwNjk3MzJlMjA1NzY4NjU3MjY1MjA2MzYxNmUyMDQ5MjA2ZDY1NjU3NDIwNzk2Zjc1M2Y=	$700,000 it is. Where can I meet you?	Base64
Ill_Song	Razor1	SkVRSE8yTE1OUVFHRVpKQU5GWENBNURQT1ZSV1FJRFhORjJHUUlEVU5CU1NBWUxFTVJaR0s0M1RGWT09PT09PQ==	I will be in touch with the address.	Base64, Base32
Ill_Song	Genuis1	SUZaU0E1M0ZFQlNHUzQzRE9WWlhWkxFRUJTV0M0VE1ORlNYRUxCQUpFUUdFWkxNTkZTWE1aSkFKRVFHMjJMSE5CMkNBWVRGRUJRV0UzREZFQjJHNklESU1WV0hBSURaTjUyU0E1M0pPUlVDQTZMUE9WWkNBNDNGTUZaR0cyQk8=	As we discussed earlier, I believe I might be able to help you with your search.	Base64, Base32
Genuis1	Ill_Song	MTExIDA0MCAxNjMgMTQ1IDE0NSAwNTYgMDQwIDEyNCAxNTAgMTQ1IDE1NiAwNDAgMTY3IDE0NSAwNDAgMTU1IDE2NSAxNjMgMTY0IDA0MCAxNTUgMTQ1IDE0NSAxNjQgMDU0IDA0MCAxNDEgMTU2IDE0NCAwNDAgMTExIDA0MCAxNjcgMTUxIDE1NCAxNTQgMDQwID E2MyAxNDUgMTQ1IDA0MCAxNjQgMTUwIDE0NSAwNDAgMTY2IDE0MSAxNTQgMTUxIDE0NCAxNTEgMTY0IDE3MSAwNDAgMTU3IDE0NiAwNDAgMTY0IDE1MCAxNTEgMTYzIDA0MCAxNDMgMTU0IDE0MSAxNTEgMTU1IDA1Ng==	I see. Then we must meet, and I will see the validity of this claim.	Base64, Octal
Ill_Song	Genuis1	SkVRR0dZTE9FQlJHS0lESk5ZUUdHT0xHTUUyV0VPRERNSVpXRU1KWkc1UVdLTkxETVUyR0VZTEdIQTJEQ05MQkdNM1RLWVJBTzVVWEkyREpOWVFISTJERkVCM1dLWkxMRlk9PT09PT0=	I can be in Caracas within the week.	Base64, MD5
Genuis1	Ill_Song	MTE2IDE1NyAwNTYgMDQwIDExNiAxNTcgMTY0IDA0MCAxNTAgMTQ1IDE2MiAxNDUgMDU2IDA0MCAxMDMgMTQxIDE1NiAwNDAgMTExIDA0MCAxNTYg	No. Not here. Can I not go to you?	Base64, Octal

		MTU3IDE2NCAwNDAgMTQ3IDE1NyAwNDAgMTY0IDE1NyAwNDAgMTcxIDE1NyAxNjUgMDc3		
Ill_Song	Genuis1	SkVRR0MzSkFNRlRIRVlMSk1RUUhJMkRCT1FRSE8zM1ZOUlNDQVlURkVCMlc0NTNKT05TUzRJQ0pFQjNXUzNETUVCWldLM1RFRUI0VzY1SkFNRVFHMlpMVE9OUVdPWkpBTzVVWEkyQkFPUlVHS0lERU1GMkdLSURCTlpTQ0EzRFBNTlFYSTJMUE5ZUUhJMkRTTjUyV08yQkFNRVFHMjMzU01VUUhHWkxET1ZaR0tJREdONVpHMklEUE1ZUUdHMzNOTlYyVzQyTERNRjJHUzMzT0ZZPT09PT09	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
Ill_Song	Method	SlZaQzRJQ05NVjJHUTMzRUZRUUVTSURCTlVRR0s2RERORjJHS1pCQU1GUkc2NUxVRUIyR1FaSkFPQlpHNjQzUU1WUlhJSURQTVlRSEkyREZFQkJXUVpMVE9NUUVFMzNZTkZYR09JRFhONVpHWVpCQU9SVVhJM0RGRUJSVzYzTEpOWlRTQTVEUEVCSUhTMzNPTTU0V0MzVEhGWT09PT09PQ==	Mr. Method, I am excited about the prospect of the Chess Boxing world title coming to Pyongyang.	Base64, Base32
Method	Ill_Song	NDkyMDYxNmQyMDZlNmY3NDIwNzM3NTcyNjUyMDc3Njg2ZjIwNzk2Zjc1MjA2MTcyNjUyYzIwNjI3NTc0MjA0OTIwNjg2MTc2NjUyMDYxNmUyMDY5NjQ2NTYxMmUyMDQ1Njk3NDY4NjU3MjIwNzc2MTc5MmMyMDQ5MjA2MTZkMjA2ZTZmNzQyMDY5NmU3NDY1NzI2NTczNzQ2NTY0MmU=	I am not sure who you are, but I have an idea. Either way, I am not interested.	Base64
Ill_Song	Method	U1NCaGJTQnFkWE4wSUdodmNHVm1kV3d1SUVsMElIZHZkV3hrSUcxbFlXNGdjMjhnYlhWamFDQjBieUJvWVhabElIUiBKRVFHQzNKQU5KMlhHNUJBTkJYWEFaTEdPVldDNElDSk9RUUhPMzNWTlJTQ0EzTEZNRlhDQTQzUEVCV1hLWTNJRUIyRzZJRElNRjNHS0lEVU5CU1NBVkRKT1JXR0tJRElNVlpHS0xSQUtCV0dLWUxUTVVRR0czM09PTlVXSVpMU0VCVVhJTFE9	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
Ill_Song	Killah	SkJYWE9JREpPTVFISTJERkVCM1dLWU	How is the weather in Qatar, Mr. Killah?	Base64, Base32

		xVTkJTWEVJREpOWVFGQ1lMVU1GWkNZSUNOT0lYQ0FTM0pOUldHQzJCNw==		
Killah	Ill_Song	MTEwIDE1NyAxNjQgMDU0IDA0MCAxNDEgMTYzIDA0MCAxNDEgMTU0IDE2NyAxNDEgMTcxIDE2MyAwNTYgMDQwIDEyNyAxNTAgMTU3IDA0MCAxNTEgMTYzIDA0MCAxNjQgMTUwIDE1MSAxNjMgMDc3	Hot, as always. Who is this?	Base64, Octal
Ill_Song	Killah	SkVRR0MzSkFNRVFHTVlMT0VCWFdNSUNETkJTWEc0WkFJSlhYUTJMT000WENBU0pBTzVYWEszREVFQldHNjVURkVCMkc2SURUVZTU0E1RElNVVFGSTJMVU5SU1NBMkRGTlJTQ0EyTE9FQkZXNjRURk1FWEE9PT09	I am a fan of Chess Boxing. I would love to see the Title held in Korea.	Base64, Base32
Killah	Ill_Song	MTI3IDE0NSAwNDAgMTY3IDE1MSAxNTQgMTU0IDA0MCAxNTAgMTQxIDE2NiAxNDUgMDQwIDE2NCAxNTcgMDQwIDE2MyAxNDUgMTQ1IDA0MCAxNTAgMTU3IDE2NyAwNDAgMTY0IDE1MCAxNDUgMDQwIDE0MiAxNTEgMTQ0IDA0MCAxNjQgMTY1IDE2MiAxNTYgMTYzIDA0MCAxNTcgMTY1IDE2NCAwNTY=	We will have to see how the bid turns out.	Base64, Octal
Ill_Song	Killah	SkZaU0E1RElNVlpHS0lEQk5aNFhJMkRKTlpUU0E1RElNRjJDQVNKQU1OWFhLM0RFRUJUTzZJRFVONFFHUVpMTU9BUUcyWUxMTVVVRSFMzM1ZPSVFFHSVpMRE5GWldTMzNPRUJTTV0M0M0pNVlpEEjNj09PQ==	Is there anything that I could do to help make your decision easier?	Base64, Base32
Killah	Ill_Song	MTE2IDE1NyAwNDEgMDQwIDEyNCAxNTAgMTQ1IDA0MCAxNDcgMTYyIDE0NSAxNDEgMTY0IDA0MCAxNTYgMTQxIDE2NCAxNTEgMTU3IDE1NiAwNDAgMTU3IDE0NiAwNDAgMTIxIDE0MSAxNjQgMTQxIDE2MiAwNDAgMTY3IDE1NyAxNjUgMTU0IDE0NCAwNDAgMTU2IDE0NSAxNjYgMTQ1IDE2MiAwNDAgMTQyIDE0NSAwNDAgMTYzIDE2NyAxNDEgMTcxIDE0NSAxNDQgMDQwIDE2MyAxNTcgMDQwIDE0NSAxND	No! The great nation of Qatar would never be swayed so easily.	Base64, Octal

		EgMTYzIDE1MSAxNTQgMTcxIDA1Ng==		
Killah	Ill_Song	MTE2IDE1NyAxNjIgMDQwIDE2NyAxNTcgMTY1IDE1NCAxNDQgMDQwIDExMSAwNTYgMDQwIDEyNyAxNDUgMDQwIDE0NCAxNTcgMDQwIDE1NiAxNTcgMTY0IDA0MCAxNjQgMTQxIDE1MyAxNDUgMDQwIDE1MyAxNTEgMTU2IDE0NCAxNTQgMTcxIDA0MCAxNjQgMTU3IDA0MCAxNjQgMTUwID E1MSAxNjMgMDQwIDE2MCAxNDEgMTY0IDE1MCAxNDUgMTY0IDE1MSAxNDMgMDQwIDE1NiAxNTcgMTY0IDE1MSAxNTcgMTU2IDA0MCAxNTcgMTQ2ID A0MCAxNDIgMTYy IDE1MSAxNDIgMTQ1IDE2MiAxNzEgMDU2	Nor would I. We do not take kindly to this pathetic notion of bribery.	Base64, Octal
Ill_Song	Raekwon	SlZaQzRJQ1NNRlNXVzUzUE5ZV0NBMkRCT1pTU0E2TFBPVVFIRzREUE5OU1c0SURYTkYyR1FJQ05PSVhDQVVUQlBKWFhFUFk9	Mr. Raekwon, have you spoken with Mr. Razor?	Base64, Base32
Raekwon	Ill_Song	NDkyMDY4NjE3NjY1MmMyMDYyNzU3NDIwNDkyMDc3NmY2ZTkyNzQyMDYyNjUyMDYyNmY3NTY3Njg3NDIwNzM2ZjIwNjU2MTczNjk2Yzc5MmU=	I have, but I won.t be bought so easily.	Base64
Ill_Song	Raekwon	SUpYWEtaM0lPUTdTQVQzR0VCUlc2NUxTT05TU0EzVFBPUVhDQVdMUE9VUUdDNFRGRUJRVzRJRFBNWlRHU1kzSk1GV0NBMzNPRUIyR1FaSkFNVjRHS1kzVk9SVVhNWkpBTU5YVzIzTEpPUjJHS1pKQU41VENBNURJTVVRRVNRMkNJRVhDQVNKQU5KMlhHNUJBTzVRVzQ1QkFQRlhYS0lEVU40UUdXM1RQTzRRSEkyREJPUVFFU0lEQk5VUUdRWkxTTVVRSEkzWkFOQlNXWTRCQU5WUVdXWkpBUEZYWEs0UkFNUlNXRzJMVE5GWFc0SURCT01RR0tZTFRQRVFHQzRaQU9CWFhHNDNKTUpXR0tMUkE=	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	NDkyMDc3NmY3NTZjNjQyMDZlNjU2NTY0MjA2MTc0MjA2YzY1NjE3Mzc0MjAzMjMwMjA2ZDY5NmM	I would need at least 20 million Rubles.	Base64

		2YzY5NmY2ZTIwNTI3NTYyNmM2NTczMmU=		
Ill_Song	Raekwon	SU5YVzQ0M0pNUlNYRUlESk9RUUdJMzNPTVVYQ0FTSkFPNVVXWTNCQU9OU1c0WkJBUEZYWEtJRFVOQlNTQTJMT01aWFhFM0xCT1JVVzYzUkFNWlhYRUlEVU5CU1NBWkRTTjVZQzIzM0dNWVFIQTMzSk5aMkNBNDNQTjVYQzQ9PT0=	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:

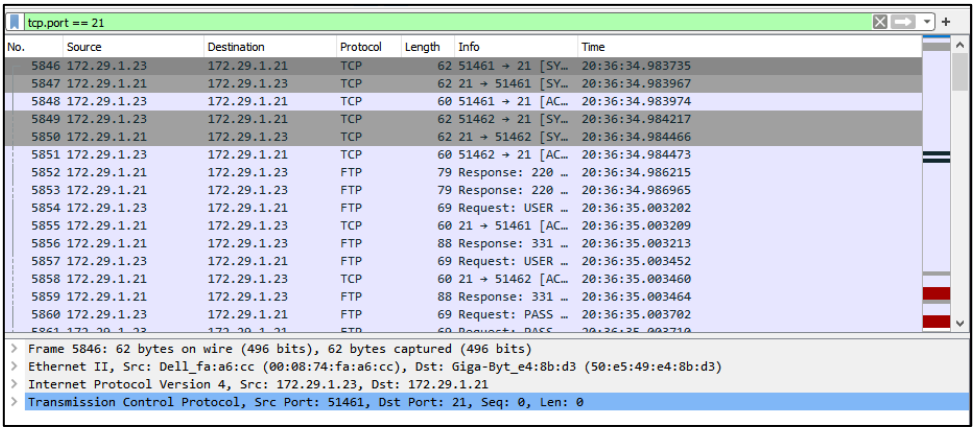


Figure 46 FTP Traffic within Wireshark

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’:

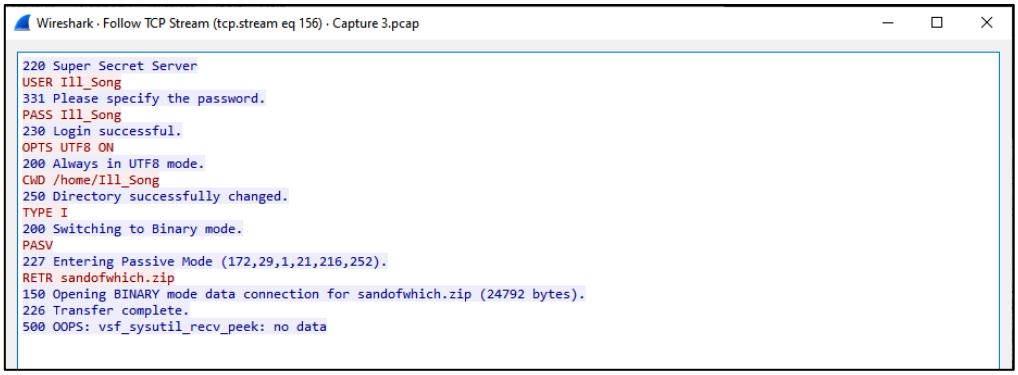


Figure 47 File Downloaded by Ill_Song

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’:

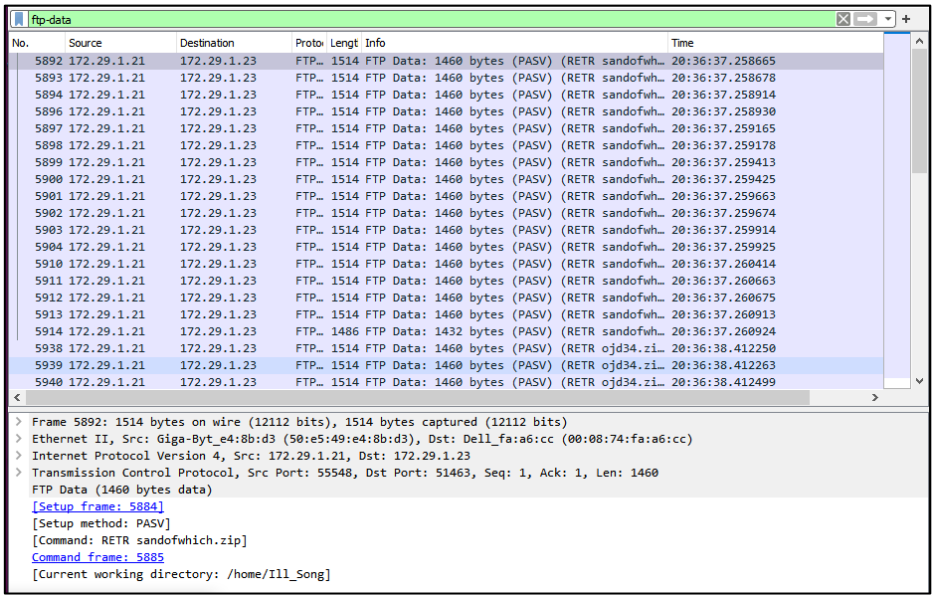


Figure 48 Transfer of ojd34.zip on Frame 5938 onwards

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:

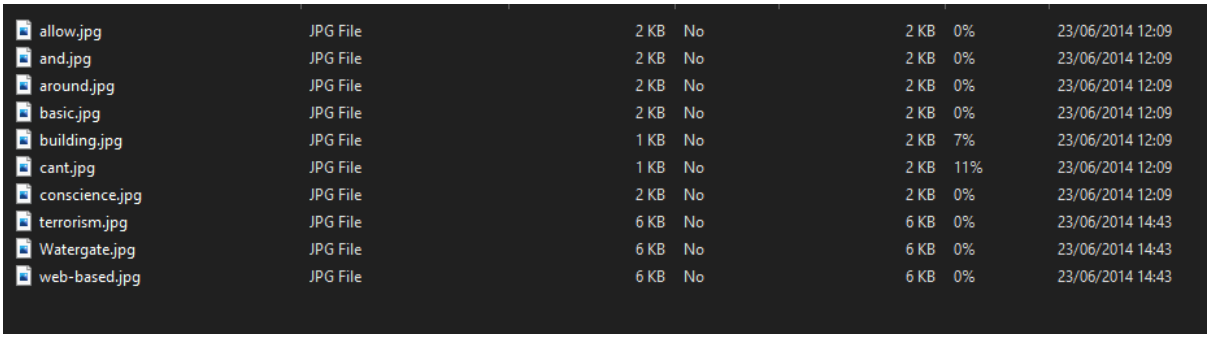


Figure 49 Contents of ‘ojd34.zip’

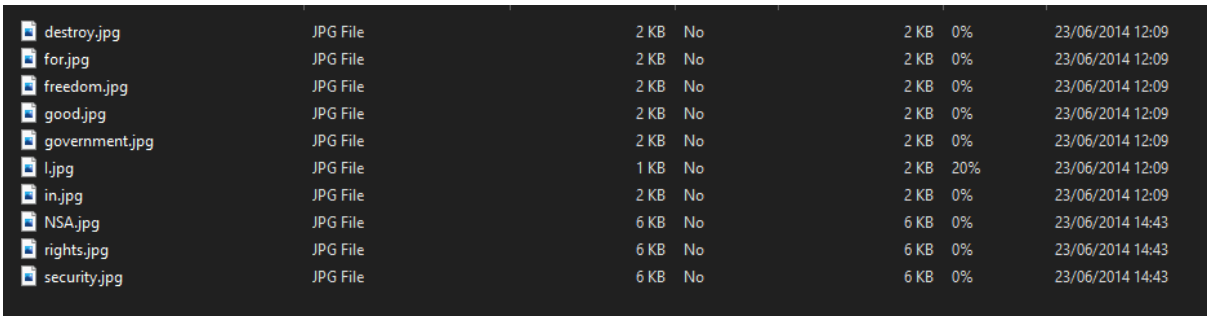


Figure 50 Contents of sandofwhich.zip

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:

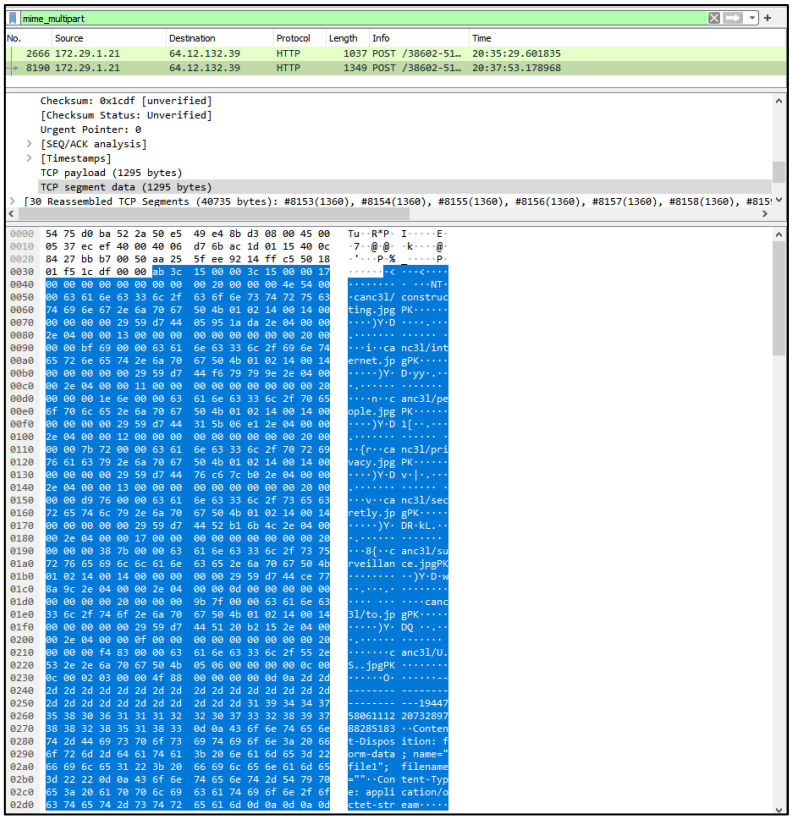


Figure 53 MIME Traffic Between 172.29.1.21 and 64.12.132.39

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:

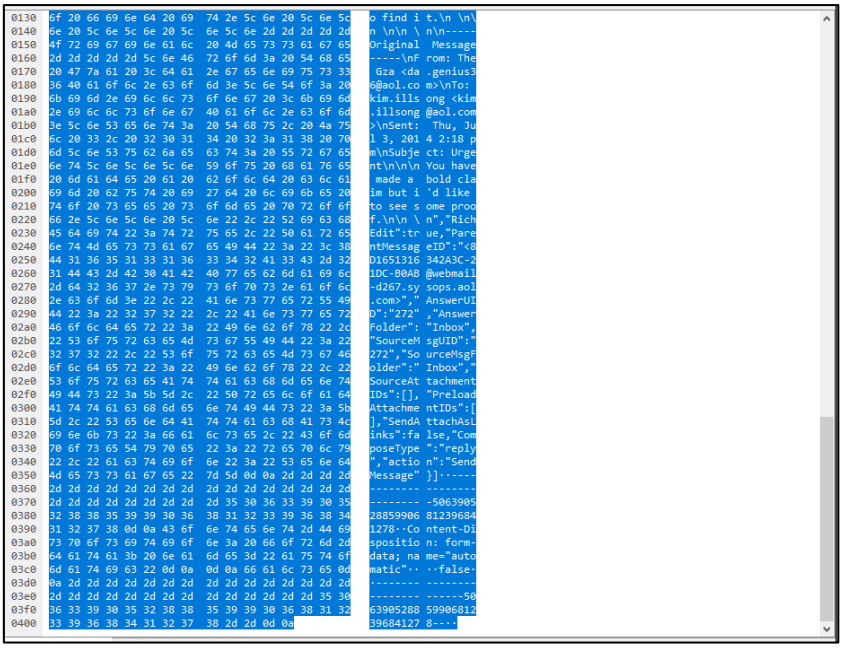


Figure 54 Email from da.genius36@aol.com to kim.illsong@aol.com

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

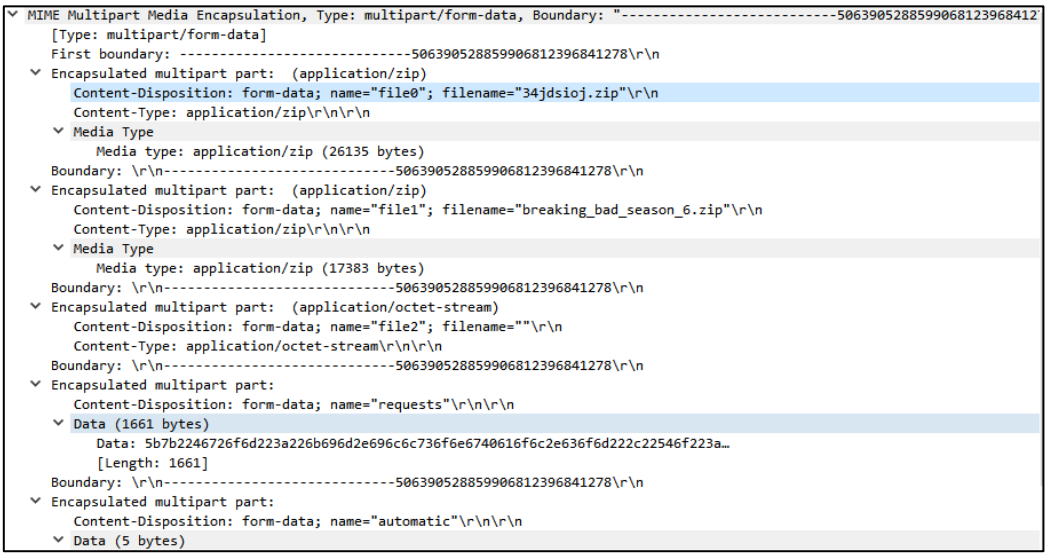


Figure 55 Metadata of Frame 2666 Showing Two ZIP Files: 34jdsioj.zip and breaking_bad_season_6.zip

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:

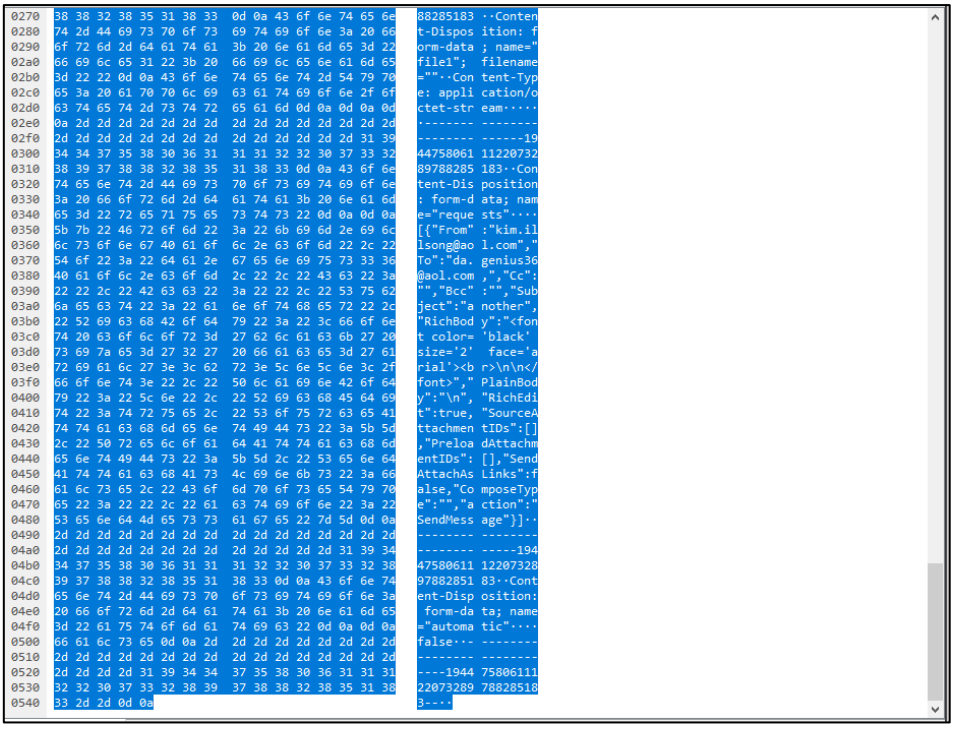


Figure 56 Email from kim.illsong@aol.com to da.genius36@aol.com

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

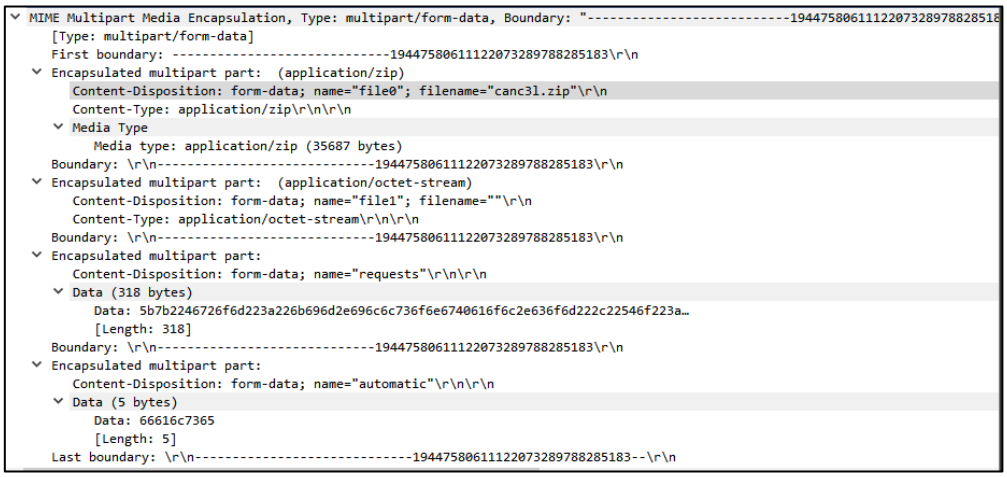


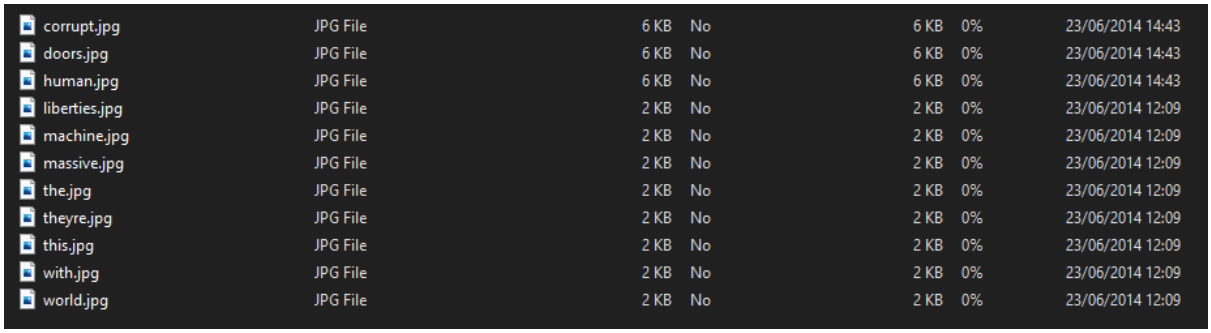
Figure 57 Metadata of Frame 8190 Revealing ‘canc3l.zip’

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

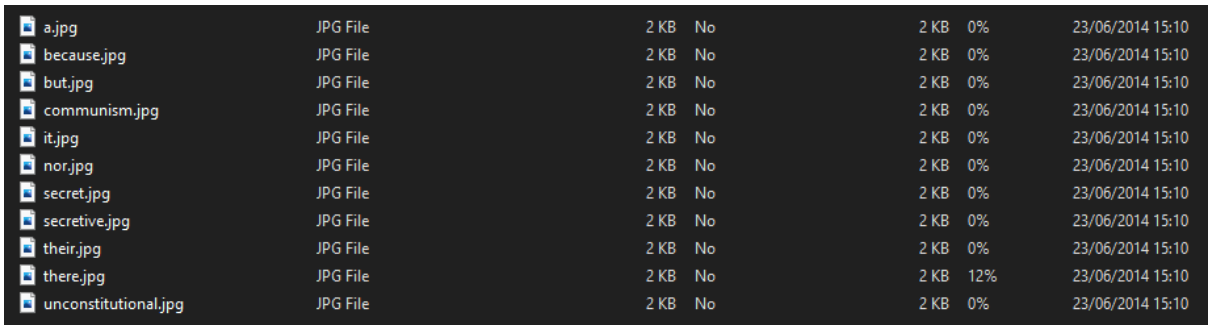
IP Address	MAC Address
172.29.1.21	50:e5:49:e4:8b:d3
64.12.132.39	54:75:d0:ba:52:2a
172.29.1.23	00:08:74:fa:a6:cc

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:



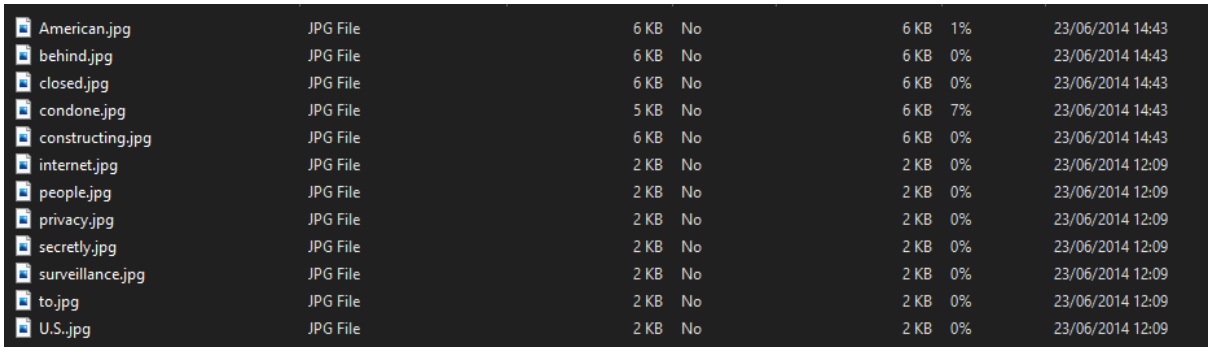
corrupt.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
doors.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
human.jpg	JPG File	6 KB	No	6 KB	0%	23/06/2014 14:43
liberties.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
machine.jpg	JPG File	2 KB	No	2 KB	0%	23/06/2014 12:09
massive.jpg	JPG File	2 KB	No	2 KB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

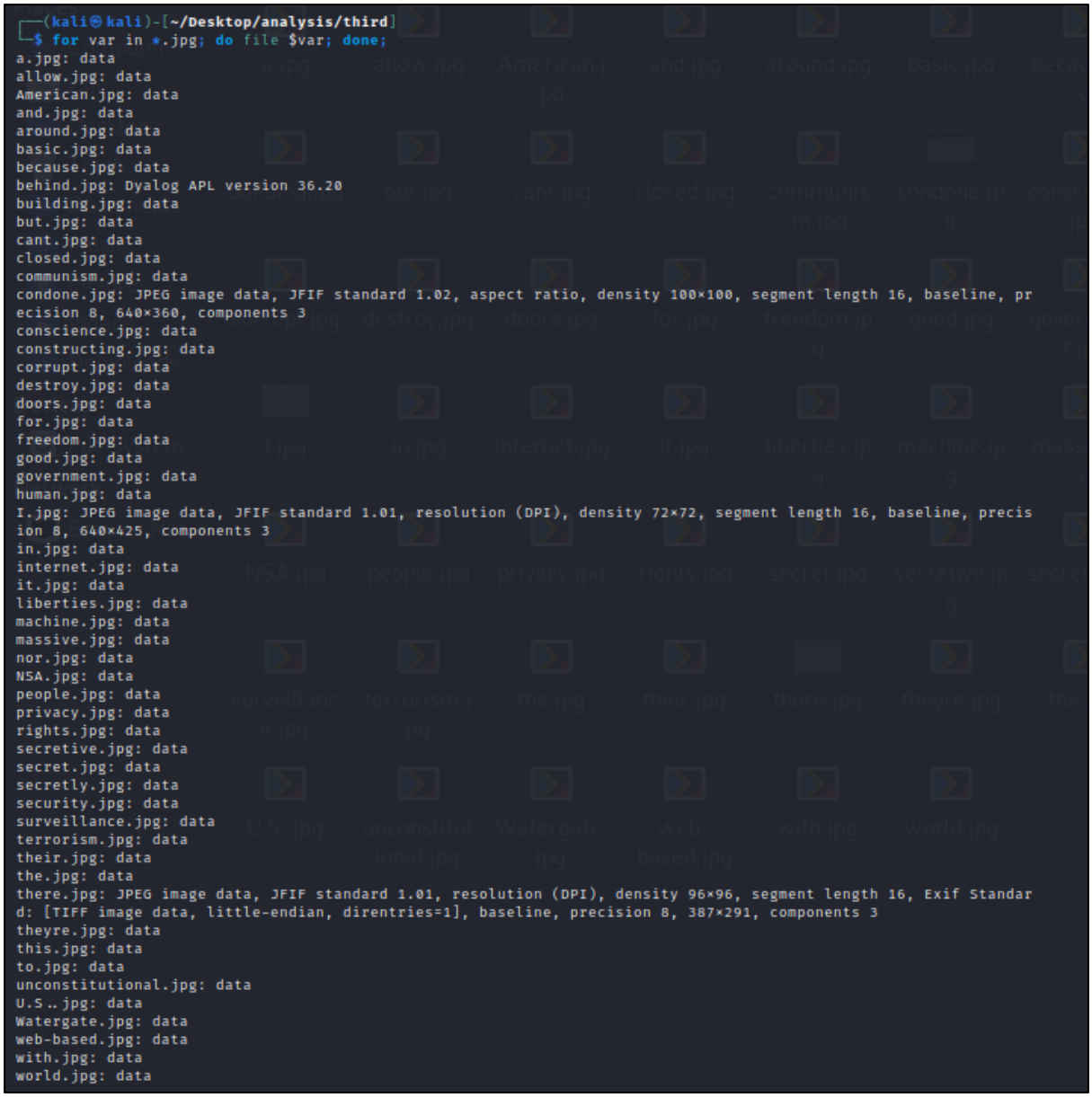
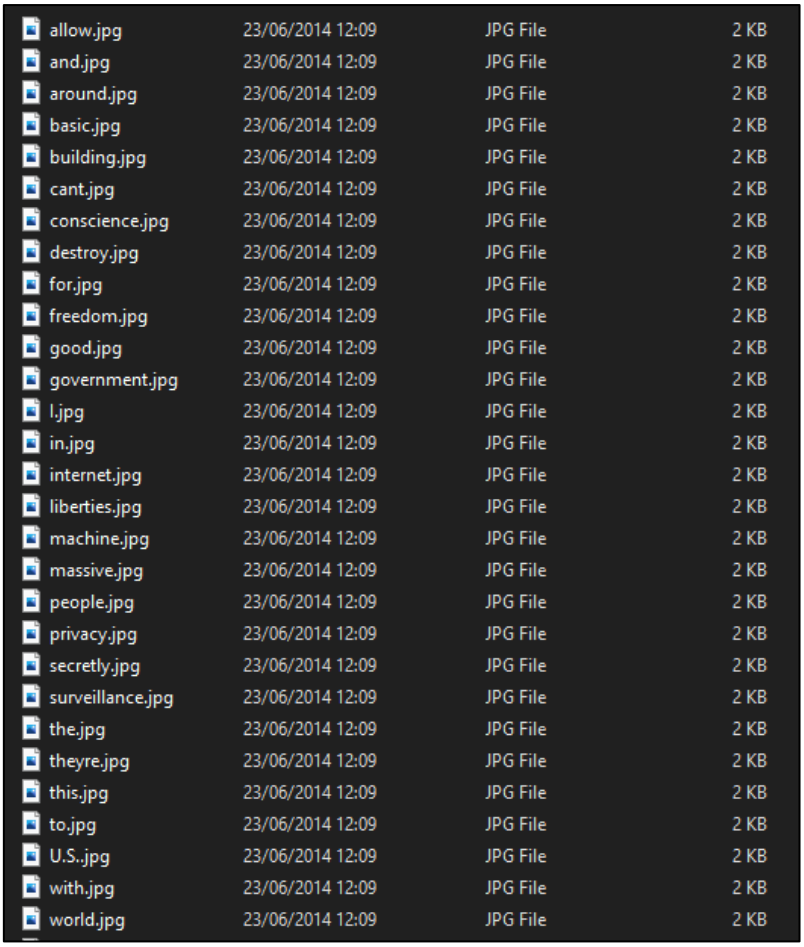


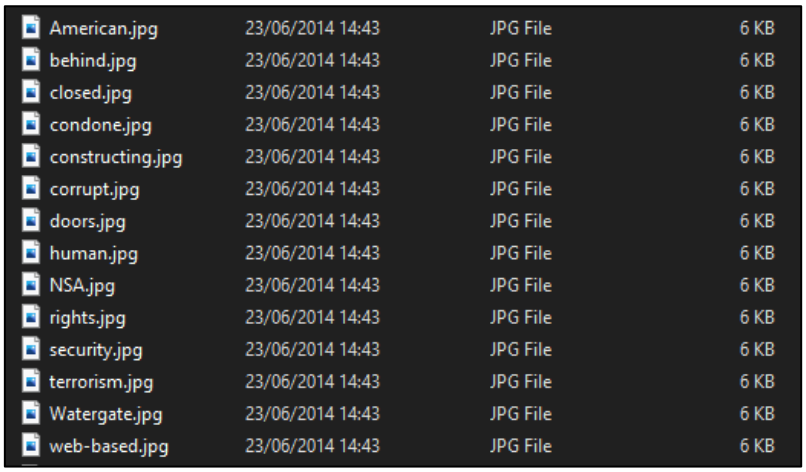
Figure 61 Bash Command to Reveal File Type of JPEG Files

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.jp
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:

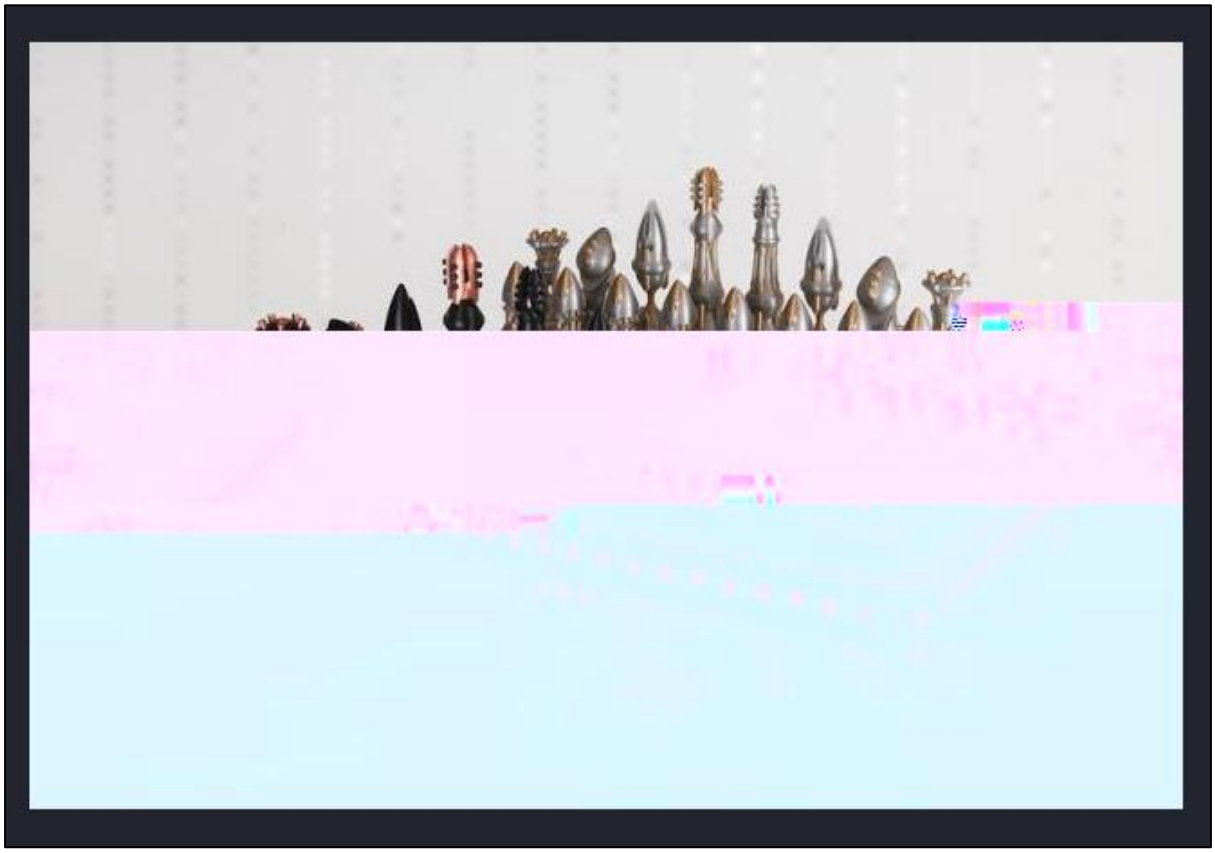


Figure 66 Image1.jpg (incomplete)

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 Watergate.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 secretive.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:

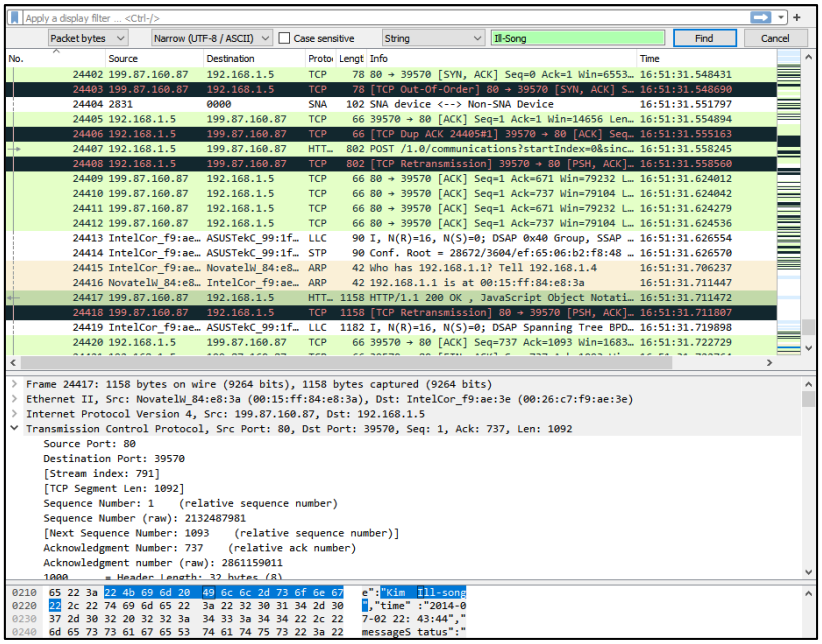


Figure 73 Mentions of Ill_Song

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 Dercover states 'I told you to pay attention.':

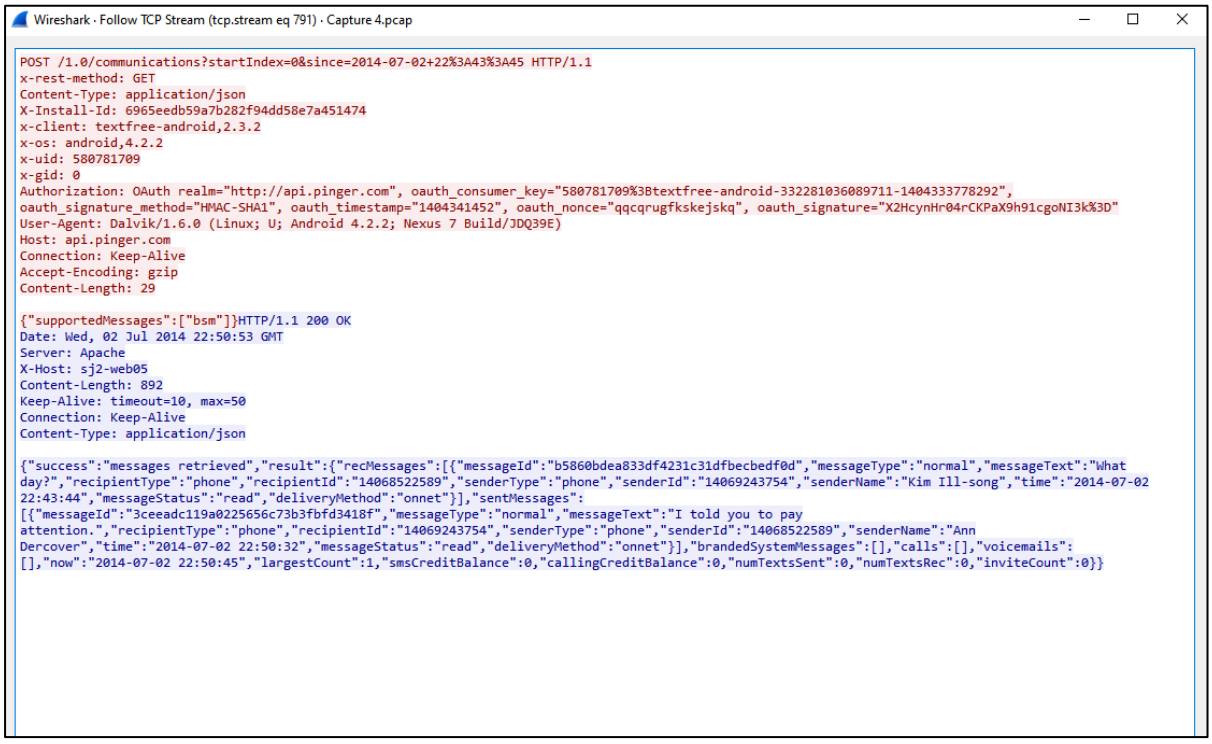


Figure 74 Metadata of Stream eq. 791 showing Message from Ann Dercover to Ill_Song

4.2.1.1 Conversations between Ann Dercover 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:

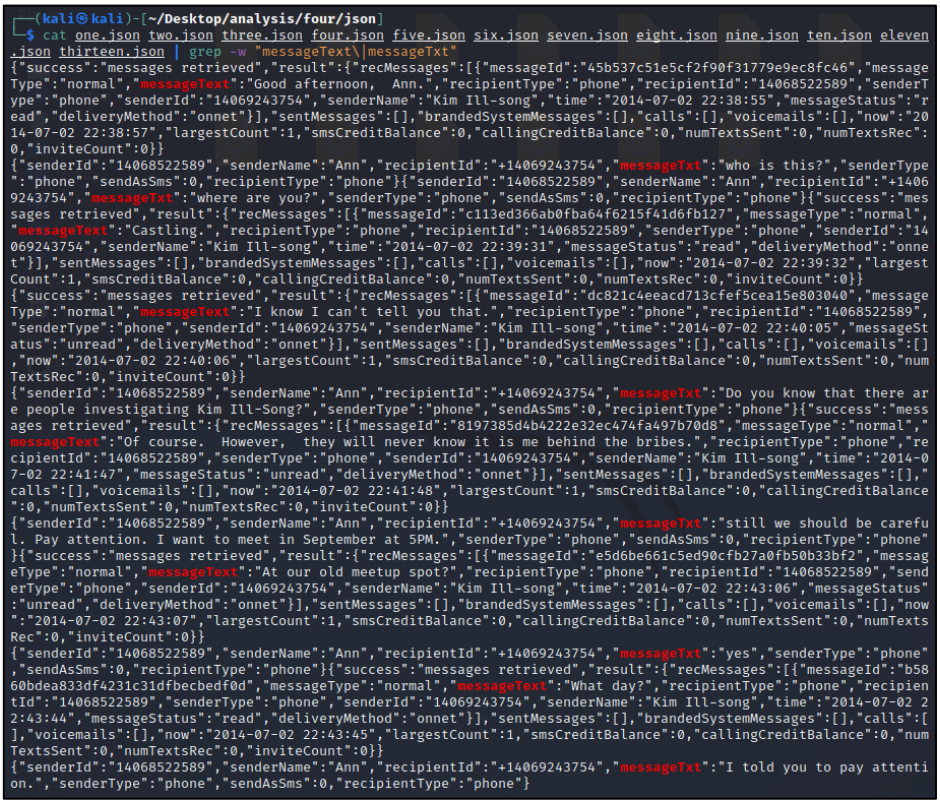


Figure 75 Conversation between Ann Dercover and Kim Ill-Song in Raw Format

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?':

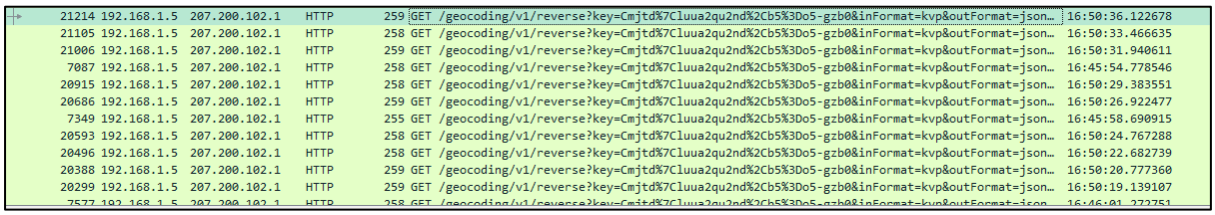
Table 7 Conversation between Ann Dercover and Kim Ill-Song

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

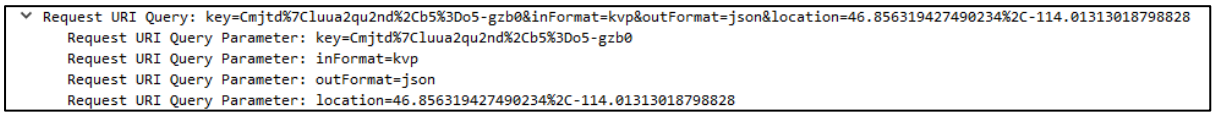
at this point that it would be in September at 5pm. When Kim asks Ann ‘What day?’ there is a pause until the final message. Examining each of the frames after the former message shows a series of requests made to a domain called ‘mob.mapquestapi.com’ from Ann’s device:



21214	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json…	16:50:36.122678
21105	192.168.1.5	207.200.102.1	HTTP	258	GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json…	16:50:33.466635
21006	192.168.1.5	207.200.102.1	HTTP	259	GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json…	16:50:19.139107

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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.856319427490234%2C-114.01313018798828
    Request URI Query Parameter: key=Cmjtd%7Cluua2qu2nd%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:

```
C:\Program Files\Wireshark>tshark.exe -r "C:\Users\ethan\Desktop\University\4th\forensics\pcap\Capture 4.pcap" -Y "http.host == mob.mapquestapi.com"
```

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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858524322509766%2C-114.01863861083984 HTTP/1.1  16:46:12.793430
8260  192.168.1.5 → 207.200.102.1 HTTP 259 GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.858734130859375%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85884475708008%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85894393920898%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85904693603515%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85914993286133%2C-114.01864624023438 HTTP/1.1  16:46:19.011753
8714  192.168.1.5 → 207.200.102.1 HTTP 259 GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85946552734375%2C-114.01864624023438 HTTP/1.1  16:46:22.178213
8788  192.168.1.5 → 207.200.102.1 HTTP 258 GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85957717895508%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85969161987305%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85980987548828%2C-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%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.85993194580078%2C-114.01864624023438 HTTP/1.1  16:46:25.518872
9286  192.168.1.5 → 207.200.102.1 HTTP 258 GET /geocoding/v1/reverse?key=Cmjtd%7Cluua2qu2nd%2Cb5%3Do5-gzb0&inFormat=kvp&outFormat=json&location=46.86029052734375%2C-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:

```
file = open("locationData.txt","r")
file = file.read()
file = file.splitlines()
fields = ["Latitude", "Longitude"]
location = []
for i in file:
    a = i[-17:]
    a = i.replace(a, " ")
    b = a.replace("%2C",",")
    c = b.replace("HTTP/1.1"," ")
    split_data = c.split("location=")
    location.append(split_data[1])

for i in range(len(location)):
    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.

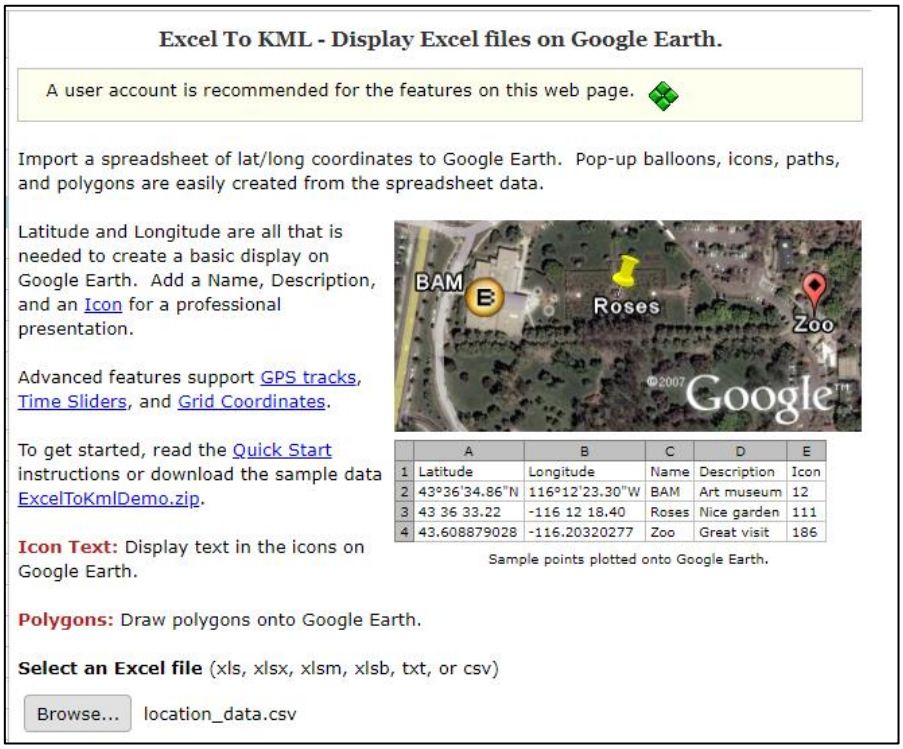


Figure 81 Usage of Excel to KML

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

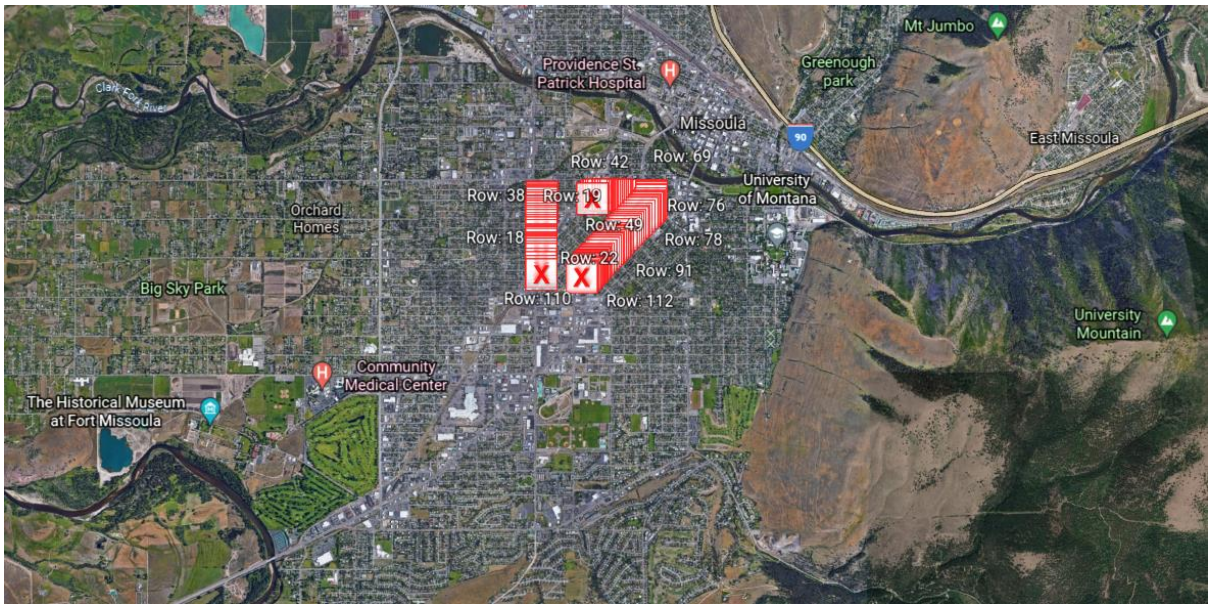


Figure 84 Location near University of Montana

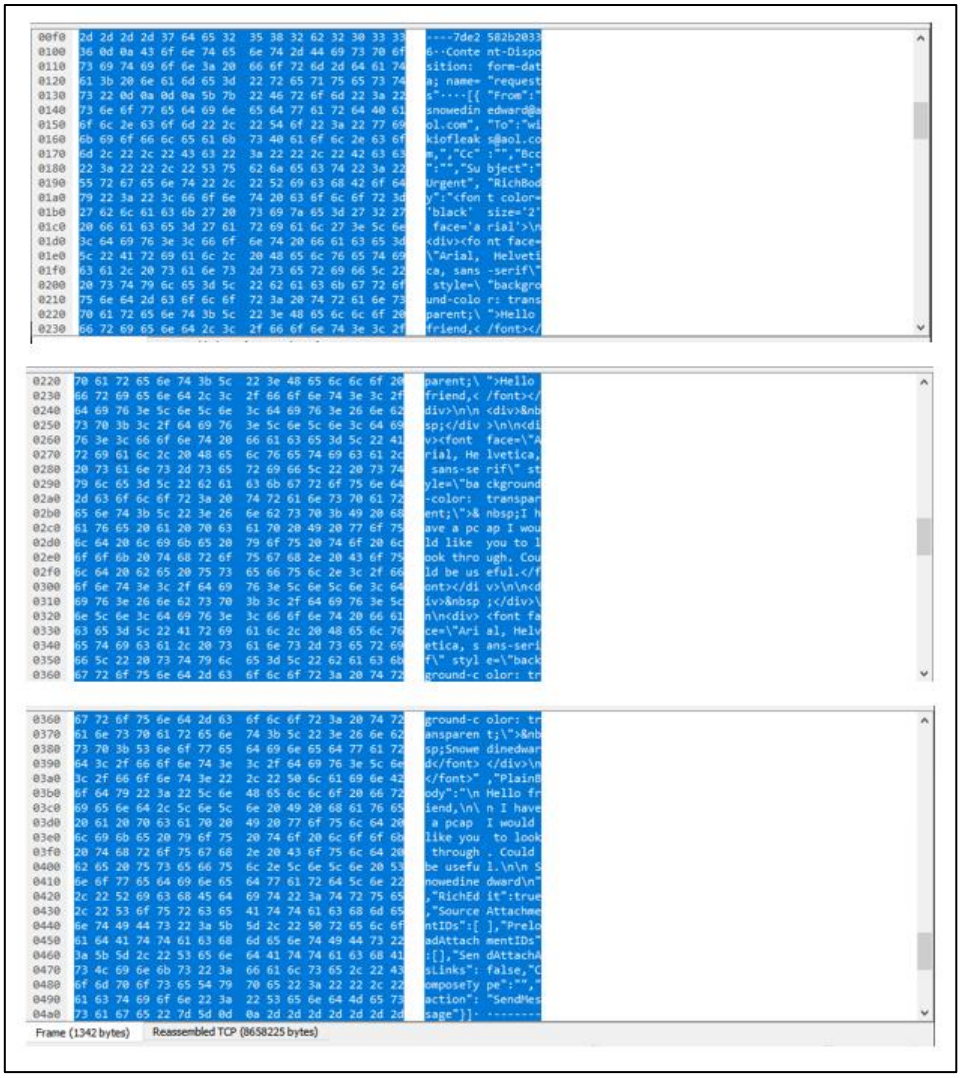
One thing to note when importing the KML document was that while it seemed to have displayed the data and reveal the location, each of the coordinates are marked by a big red X. It wasn't sure why this was happening, so

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

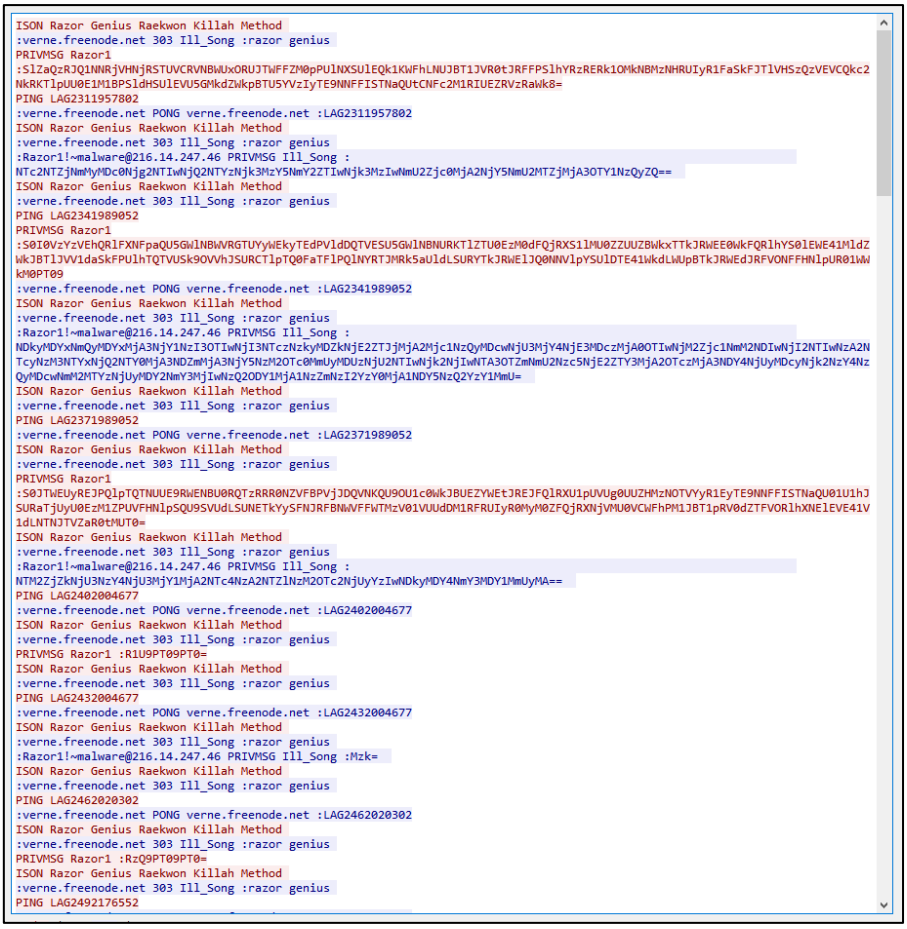
5.1 Capture 1

5.1.1 Email Conversation

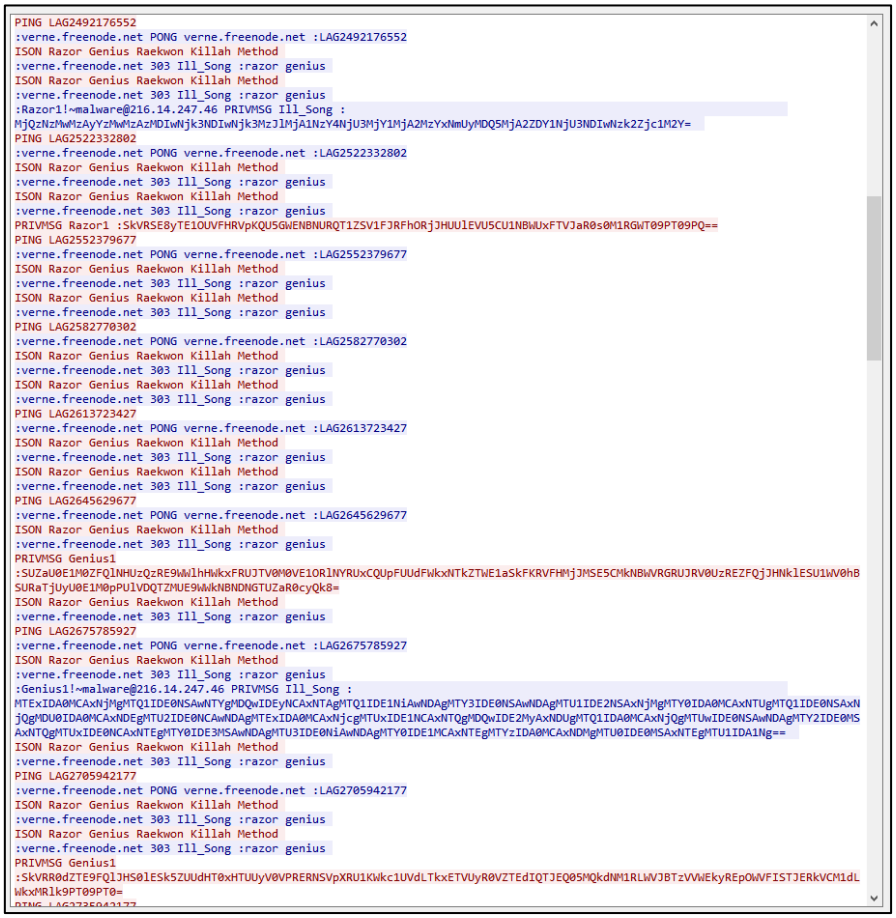


5.2 Capture 2

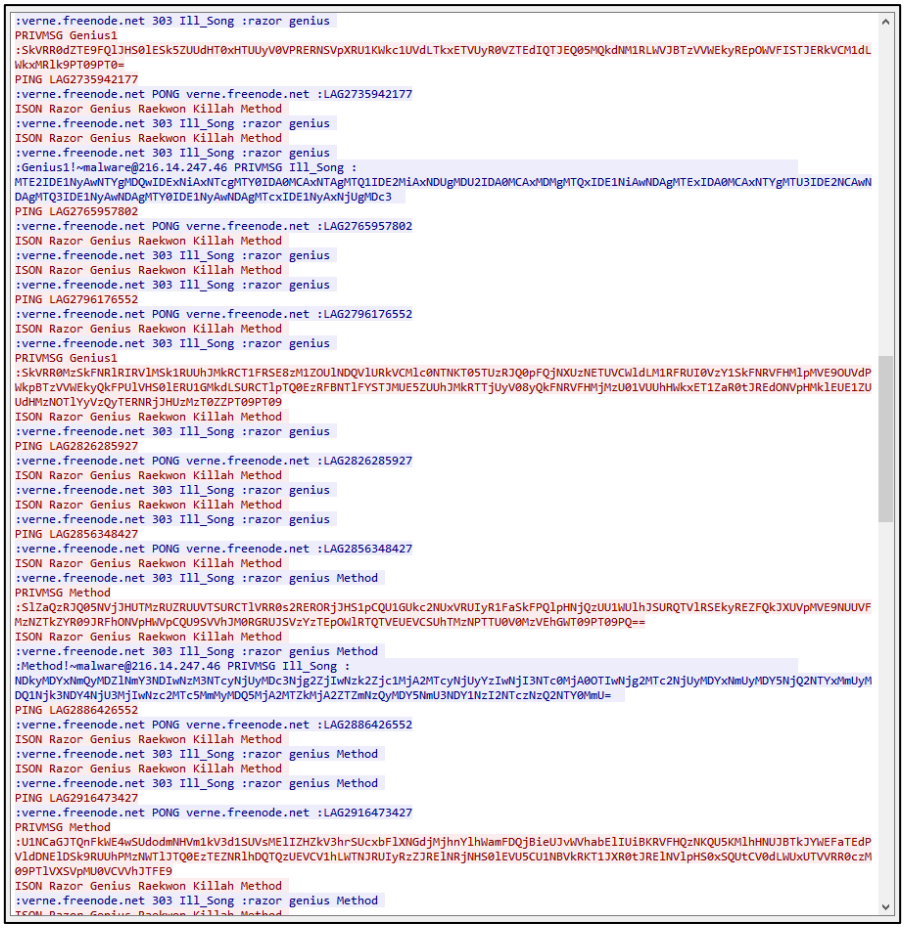
5.2.1 Encoded Conversations Between Ill Song and Other Members



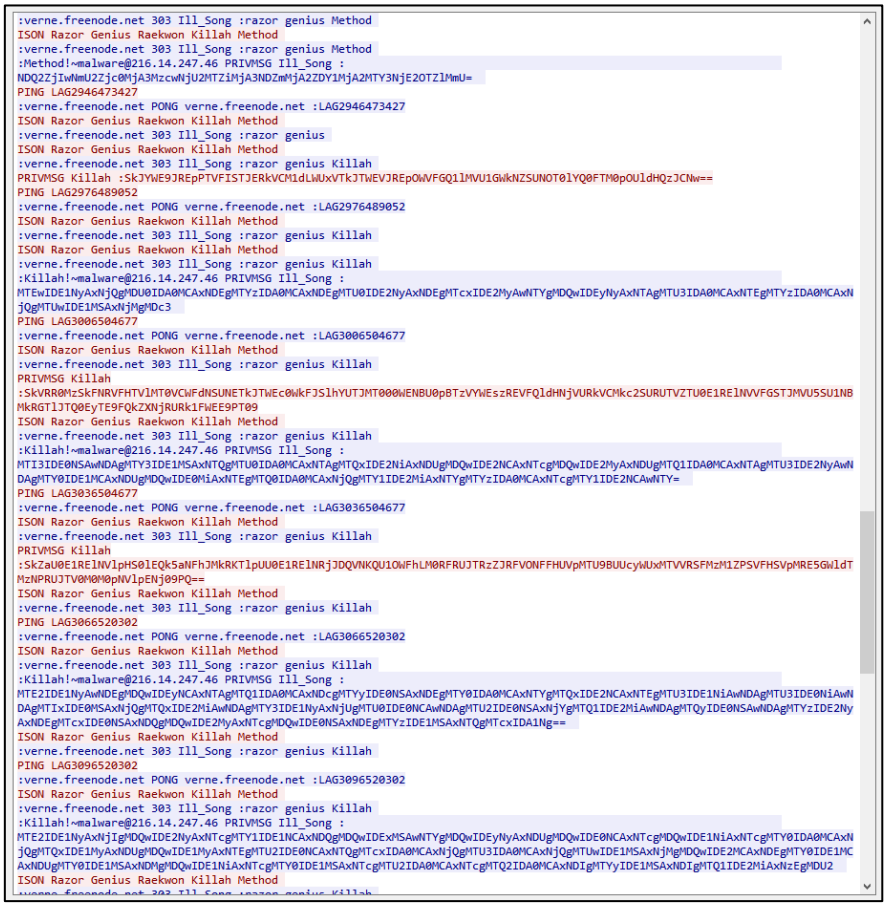
```
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Razor1
:SlZaQzRJQ1NNRjVHNjRSTUVCRVNBWUxORUJTWFFZM0pPUlNXSUlEQk1KWFhLNUJBT1JVR0tJRFFPSlhYRzRERk1OMkNBMzNHRUIyR1FaSkFJTlVHSzQzVEVCQkc2
NkRKTlpUU0E1M1BPSldHSUlEVU5GMkdZWkpBTU5YVzIyTE9NNFFISTNaQUtCNFc2M1RIUEZRVzRaWk8=
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 :
NTc2NTZjNmMyMDc0Njg2NTIwNjQ2NTYzNjk3MzY5NmY2ZTIwNjk3MzIwNmU2Zjc0MjA2NjY5NmU2MTZjMjA3OTY1NzQyZQ==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2341989052
PRIVMSG Razor1
:S0I0VzYzVEhQRlFXNFpaQU5GWlNBWVRGTUYyWEkyTEdPVldDQTVESU5GWlNBNURKTlZTU0EzM0dFQjRXS1lMU0ZZUUZBWkxTTkJRWEE0WkFQRlhYS0lEWE41MldZ
WkJBTlJVV1daSkFPUlhTQTVUSk9OVVhJSURCTlpTQ0FaTFlPQlNYRTJMRk5aUldLSURYTkJRWElJQ0NNVlpYSUlDTE41WkdLWUpBTkJRWEdJRFVONFFHNlpUR01WW
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 :
NDkyMDYxNmQyMDYxMjA3NjY1NzI3OTIwNjI3NTczNzkyMDZkNjE2ZTJjMjA2Mjc1NzQyMDcwNjU3MjY4NjE3MDczMjA0OTIwNjM2Zjc1NmM2NDIwNjI2NTIwNzA2N
TcyNzM3NTYxNjQ2NTY0MjA3NDZmMjA3NjY5NzM2OTc0MmUyMDUzNjU2NTIwNjk2NjIwNTA3OTZmNmU2Nzc5NjE2ZTY3MjA2OTczMjA3NDY4NjUyMDcyNjk2NzY4Nz
QyMDcwNmM2MTYzNjUyMDY2NmY3MjIwNzQ2ODY1MjA1NzZmNzI2YzY0MjA1NDY5NzQ2YzY1MmU=
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
:S0JTWEUyREJPQlpTQTNUUE9RWENBU0RQTzRRR0NZVFBPVjJDQVNKQU9OU1c0WkJBUEZYWEtJREJFQlRXU1pUVUg0UUZHMzNOTVYyR1EyTE9NNFFISTNaQU01U1hJ
SURaTjUyU0EzM1ZPUVFHNlpSQU9SVUdLSUNETkYySFNJRFBNWVFFWTMzV01VUUdDM1RFRUIyR0MyM0ZFQjRXNjVMU0VCWFhPM1JBT1pRV0dZTFVORlhXNElEVE41V
1dLNTNJTVZaR0tMUT0=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG Ill_Song :
NTM2ZjZkNjU3NzY4NjU3MjY1MjA2NTc4NzA2NTZlNzM2OTc2NjUyYzIwNDkyMDY4NmY3MDY1MmUyMA==
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 :R1U9PT09PT0=
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
```



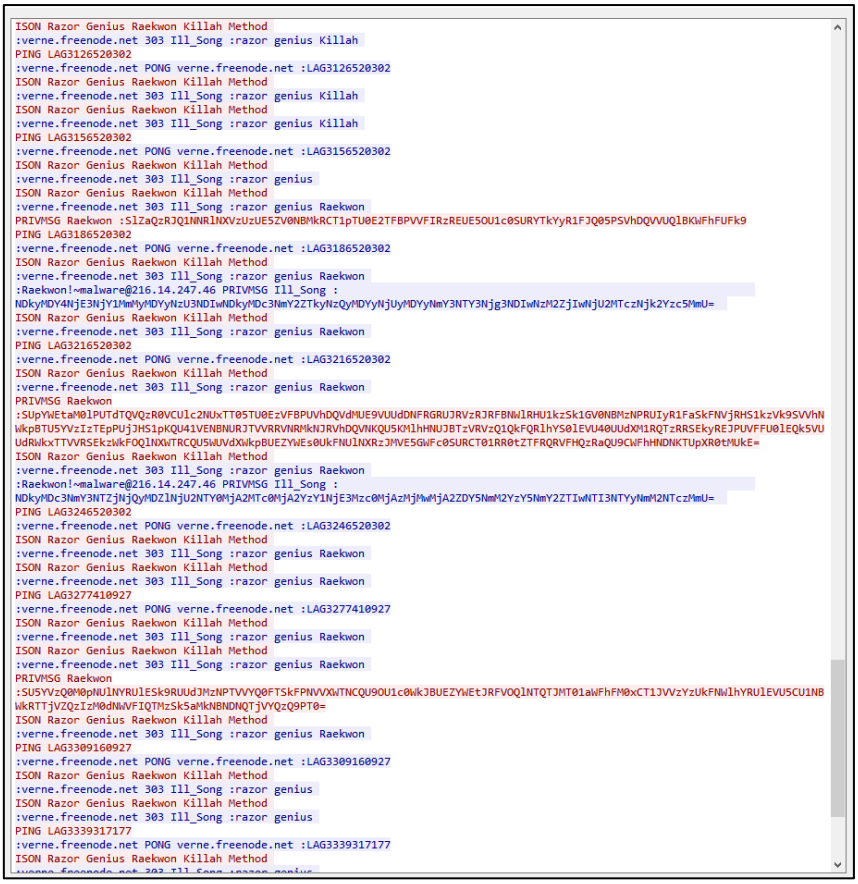
```
PING LAG2492176552
:verne.freenode.net PONG verne.freenode.net :LAG2492176552
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Razor1!~malware@216.14.247.46 PRIVMSG Ill_Song :
MjQzNzMwMzAyYzMwMzAzMDIwNjk3NDIwNjk3MzJlMjA1NzY4NjU3MjY1MjA2MzYxNmUyMDQ5MjA2ZDY1NjU3NDIwNzk2Zjc1M2Y=
PING LAG2522332802
:verne.freenode.net PONG verne.freenode.net :LAG2522332802
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
PRIVMSG Razor1 :SkVRSE8yTE1OUVFHRVpKQU5GWENBNURQT1ZSV1FJRFhORjJHUUlEVU5CU1NBWUxFTVJaR0s0M1RGWT09PT09PQ==
PING LAG2552379677
:verne.freenode.net PONG verne.freenode.net :LAG2552379677
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 LAG2582770302
:verne.freenode.net PONG verne.freenode.net :LAG2582770302
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 LAG2613723427
:verne.freenode.net PONG verne.freenode.net :LAG2613723427
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 LAG2645629677
:verne.freenode.net PONG verne.freenode.net :LAG2645629677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Genius1
:SUZaU0E1M0ZFQlNHUzQzRE9WWlhHWkxFRUJTV0M0VE1ORlNYRUxCQUpFUUdFWkxNTkZTWE1aSkFKRVFHMjJMSE5CMkNBWVRGRUJRV0UzREZFQjJHNklESU1WV0hB
SURaTjUyU0E1M0pPUlVDQTZMUE9WWkN8NDNGTUZaR0cyQk8=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2675785927
:verne.freenode.net PONG verne.freenode.net :LAG2675785927
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
:Genius1!~malware@216.14.247.46 PRIVMSG Ill_Song :
MTExIDA0MCAxNjMgMTQ1IDE0NSAwNTYgMDQwIDEyNCAxNTAgMTQ1IDE1NiAwNDAgMTY3IDE0NSAwNDAgMTU1IDE2NSAxNjMgMTY0IDA0MCAxNTUgMTQ1IDE0NSAxN
jQgMDU0IDA0MCAxNDEgMTU2IDE0NCAwNDAgMTExIDA0MCAxNjcgMTUxIDE1NCAxNTQgMDQwIDE2MyAxNDUgMTQ1IDA0MCAxNjQgMTUwIDE0NSAwNDAgMTY2IDE0MS
AxNTQgMTUxIDE0NCAxNTEgMTY0IDE3MSAwNDAgMTU3IDE0NiAwNDAgMTY0IDE1MCAxNTEgMTY1IDA0MCAxNDMgMTU0IDE0MSAxNTEgMTU1IDA1Ng==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG2705942177
:verne.freenode.net PONG verne.freenode.net :LAG2705942177
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
PRIVMSG Genius1
:SkVRR0dZTE9FQlJHS0lESk5ZUUdHT0xHTUUyV0VPRERNSVpXRU1KWkc1UVdLTkxETVUyR0VZTEdIQTJEQ05MQkdNM1RLWVJBTzVVWEkyREpOWVFISTJERkVCM1dL
WkxMRlk9PT09PT0=
PING LAG2735943177
```



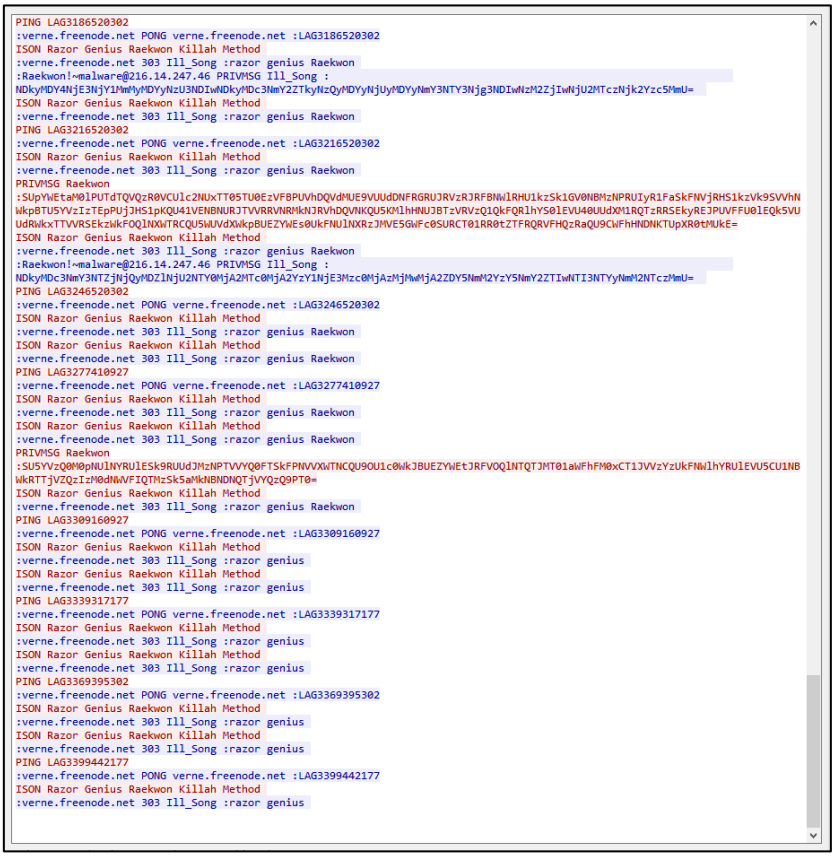
```
:verne.freenode.net 303 Ill_Song :razor genius
PRIVMSG Genius1
:SkVRR0dZTE9FQlJHS0lESk5ZUUdHT0xHTUUyV0VPRERNSVpXRU1KWkc1UVdLTkxETVUyR0VZTEdIQTJEQ05MQkdNM1RLWVJBTzVVWEkyREpOWVFISTJERkVCM1dL
WkxMRlk9PT09PT0=
PING LAG2735942177
:verne.freenode.net PONG verne.freenode.net :LAG2735942177
ISON Razor Genius 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 :
MTE2IDE1NyAwNTYgMDQwIDExNiAxNTcgMTY0IDA0MCAxNTAgMTQ1IDE2MiAxNDUgMDU2IDA0MCAxMDMgMTQxIDE1NiAwNDAgMTExIDA0MCAxNTYgMTU3IDE2NCAwN
DAgMTQ3IDE1NyAwNDAgMTY0IDE1NyAwNDAgMTcxIDE1NyAxNjUgMDc3
PING LAG2765957802
:verne.freenode.net PONG verne.freenode.net :LAG2765957802
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
:SkVRR0MzSkFNRlRIRVlMSk1RUUhJMkRCT1FRSE8zM1ZOUlNDQVlURkVCMlc0NTNKT05TUzRJQ0pFQjNXUzNETUVCWldLM1RFRUI0VzY1SkFNRVFHMlpMVE9OUVdP
WkpBTzVVWEkyQkFPUlVHS0lERU1GMkdLSURCT1pTQ0EzRFBNTlFYSTJMUE5ZUUhJMkRTTjUyV08yQkFNRVFHMjMzU01VUUhHWkxET1ZaR0tJREdONVpHMklEUE1ZU
UdHMzNOTlYyVzQyTERNRjJHUzMzT0ZZPT09PT09
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
:SlZaQzRJQ05NVjJHUTMzRUZRUUVTSURCTlVRR0s2RERORjJHS1pCQU1GUkc2NUxVRUIyR1FaSkFPQlpHNjQzUU1WUlhJSURQTVlRSEkyREZFQkJXUVpMVE9NUUVF
MzNZTkZYR09JRFhONVpHWVpCQU9SVVhJM0RGRUJSVzYzTEpOWlRTQTVEUEVCSUhTMzNPTTU0V0MzVEhGWT09PT09PQ==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
:Method!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyMDYxNmQyMDZlNmY3NDIwNzM3NTcyNjUyMDc3Njg2ZjIwNzk2Zjc1MjA2MTcyNjUyYzIwNjI3NTc0MjA0OTIwNjg2MTc2NjUyMDYxNmUyMDY5NjQ2NTYxMmUyM
DQ1Njk3NDY4NjU3MjIwNzc2MTc5MmMyMDQ5MjA2MTZkMjA2ZTZmNzQyMDY5NmU3NDY1NzI2NTczNzQ2NTY0MmU=
PING LAG2886426552
:verne.freenode.net PONG verne.freenode.net :LAG2886426552
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
PING LAG2916473427
:verne.freenode.net PONG verne.freenode.net :LAG2916473427
PRIVMSG Method
:U1NCaGJTQnFkWE4wSUdodmNHVm1kV3d1SUVsMElIZHZkV3hrSUcxbFlXNGdjMjhnYlhWamFDQjBieUJvWVhabElIUiBKRVFHQzNKQU5KMlhHNUJBTkJYWEFaTEdP
VldDNElDSk9RUUhPMzNWTlJTQ0EzTEZNRlhDQTQzUEVCV1hLWTNJRUIyRzZJRElNRjNHS0lEVU5CU1NBVkRKT1JXR0tJRElNV1pHS0xSQUtCV0dLWUxUTVVRR0czM
09PTlVXSVpMU0VCVVhJTFE9
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
```



```
:verne.freenode.net 303 Ill_Song :razor genius Method
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Method
:Method!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDQ2ZjIwNmU2Zjc0MjA3MzcwNjU2MTZiMjA3NDZmMjA2ZDY1MjA2MTY3NjE2OTZlMmU=
PING LAG2946473427
:verne.freenode.net PONG verne.freenode.net :LAG2946473427
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 Killah
PRIVMSG Killah :SkJYWE9JREpPTVFISTJERkVCM1dLWUxVTkJTWEVJREpOWVFGQ1lMVU1GWkNZSUNOT0lYQ0FTM0pOUldHQzJCNw==
PING LAG2976489052
:verne.freenode.net PONG verne.freenode.net :LAG2976489052
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
:Killah!~malware@216.14.247.46 PRIVMSG Ill_Song :
MTEwIDE1NyAxNjQgMDU0IDA0MCAxNDEgMTYzIDA0MCAxNDEgMTUwIDE2NyAxNDEgMTcxIDE2MyAwNTYgMDQwIDEyNyAxNTAgMTU3IDA0MCAxNTEgMTYzIDA0MCAxN
jQgMTUwIDE1MSAxNjMgMDc3
PING LAG3006504677
:verne.freenode.net PONG verne.freenode.net :LAG3006504677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PRIVMSG Killah
:SkVRR0MzSkFNRVFHTVlMT0VCWFdNSUNETkJTWEc0WkFJSlhYUTJMT000WENBU0pBTzVYWEszREVFQldHNjVURkVCMkc2SURUTVZTU0E1RElNVVFGSTJMVU5SU1NB
MkRGTlJTQ0EyTE9FQkZXNjRURk1FWEE9PT09
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
:Killah!~malware@216.14.247.46 PRIVMSG Ill_Song :
MTI3IDE0NSAwNDAgMTY3IDE1MSAxNTQgMTU0IDA0MCAxNTAgMTQxIDE2NiAxNDUgMDQwIDE2MyAxNDUgMTQ1IDA0MCAxNTAgMTU3IDE2NyAwN
DAgMTY0IDE1MCAxNDUgMDQwIDE0MiAxNTEgMTQ0IDA0MCAxNjQgMTY1IDE2MiAxNTYgMTY1IDA0MCAxNTcgMTY1IDE2NCAwNTY=
PING LAG3036504677
:verne.freenode.net PONG verne.freenode.net :LAG3036504677
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PRIVMSG Killah
:SkZaU0E1RElNVlpHS0lEQk5aNFhJMkRKTlpUU0E1RElNRjJDQVNKQU1OWFhLM0RFRUJTRzZJRFVONFFHUVpMTU9BUUcyWUxMTVVRSFMzM1ZPSVFHSVpMRE5GWldT
MzNPRUJTV0M0M0pNVlpENj09PQ==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PING LAG3066520302
:verne.freenode.net PONG verne.freenode.net :LAG3066520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
:Killah!~malware@216.14.247.46 PRIVMSG Ill_Song :
MTE2IDE1NyAwNDEgMDQwIDEyNCAxNTAgMTQ1IDA0MCAxNDcgMTYyIDE0NSAxNDEgMTY0IDA0MCAxNTYgMTQxIDE2NiAwNDAgMTU3IDE0NiAwN
DAgMTIxIDE0MSAxNjQgMTQxIDE2MiAwNDAgMTY3IDE1NyAxNjUgMTU0IDE0NCAwNDAgMTU2IDE0NSAxNjYgMTQ1IDE2MiAwNDAgMTQyIDE0NSAxNjQgMTYzIDE2Ny
AxNDEgMTcxIDE0NSAxNDQgMDQwIDE2MyAxNTcgMDQwIDE0NSAxNDEgMTYzIDE1MSAxNTQgMTcxIDA1Ng==
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PING LAG3096520302
:verne.freenode.net PONG verne.freenode.net :LAG3096520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
:Killah!~malware@216.14.247.46 PRIVMSG Ill_Song :
MTE2IDE1NyAxNjIgMDQwIDE2NyAxNTcgMTY1IDE1NCAxNDQgMDQwIDExMSAwNTYgMDQwIDEyNyAxNDUgMDQwIDE0NCAxNTcgMTQwIDE1NiAxNTcgMTY0IDA0MCAxN
jQgMTQxIDE1MyAxNDUgMDQwIDE1MyAxNTEgMTU2IDE0NCAxNTQgMTcxIDA0MCAxNjQgMTU3IDA0MCAxNjQgMTUwIDE1MSAxNjMgMDQwIDE2MCAxNDEgMTY0IDE1MC
AxNDUgMTY0IDE1MSAxNDMgMDQwIDE1NiAxNTcgMTY0IDE1MSAxNTcgMTU2IDA0MCAxNTcgMTQ2IDA0MCAxNDIgMTY1IDE2MiAxNzEgMDU2
ISON Razor Genius Raekwon Killah Method
```



```
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Killah
PING LAG3126520302
:verne.freenode.net PONG verne.freenode.net :LAG3126520302
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 LAG3156520302
:verne.freenode.net PONG verne.freenode.net :LAG3156520302
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 Raekwon
PRIVMSG Raekwon :SlZaQzRJQ1NNRlNXVzUzUE5ZV0NBMkRCT1pTU0E2TFBPVVFIRzREUE5OU1c0SURYTkYyR1FJQ05PSVhDQVVUQlBKWFhFUFk9
PING LAG3186520302
:verne.freenode.net PONG verne.freenode.net :LAG3186520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
:Raekwon!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyMDY4NjE3NjY1MmMyMDYyNzU3NDIwNDkyMDc3NmY2ZTkyNzQyMDYyNjUyMDYyNmY3NTY3Njg3NDIwNzM2ZjIwNjU2MTczNjk2Yzc5MmU=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG3216520302
: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
:SUpYWEtaM0lPUTdTQVQzR0VCUlc2NUxTT05TU0EzVFBPUVhDQVdMUE9VUUdDNFRGRUJRVzRJRFBNWlRHU1kzSk1GV0NBMzNPRUIyR1FaSkFNVjRHS1kzVk9SVVhN
WkpBTU5YVzIzTEpPUjJHS1pKQU41VENBNURJTVVRRVNRMkNJRVhDQVNKQU5KMlhHNUJBTzVRVzQ1QkFQRlhYS0lEVU40UUdXM1RQTzRRSEkyREJPUVFFU0lEQk5VU
UdRWkxTTVVRSEkzWkFOQlNXWTRCQU5WUVdXWkpBUEZYWEs0UkFNUlNXRzJMVE5GWFc0SURCT01RR0tZTFRQRVFHQzRaQU9CWFhHNDNKTUpXR0tMUkE=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
:Raekwon!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyMDc3NmY3NTZjNjQyMDZlNjU2NTY0MjA2MTc0MjA2YzY1NjE3Mzc0MjAzMjMwMjA2ZDY5NmM2YzY5NmY2ZTIwNTI3NTYyNmM2NTczMmU=
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
:SU5YVzQ0M0pNUlNYRUlESk9RUUdJMzNPTVVYQ0FTSkFPNVVXWTNCQU9OU1c0WkJBUEZYWEtJRFVOQlNTQTJMT01aWFhFM0xCT1JVVzYzUkFNWlhYRUlEVU5CU1NB
WkRTTjVZQzIzM0dNWVFIQTMzSk5aMkNBNDNQTjVYQzQ9PT0=
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
```



```
PING LAG3186520302
:verne.freenode.net PONG verne.freenode.net :LAG3186520302
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
:Raekwon!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyMDY4NjE3NjY1MmMyMDYyNzU3NDIwNDkyMDc3NmY2ZTkyNzQyMDYyNjUyMDYyNmY3NTY3Njg3NDIwNzM2ZjIwNjU2MTczNjk2Yzc5MmU=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
PING LAG3216520302
: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
:SUpYWEtaM0lPUTdTQVQzR0VCUlc2NUxTT05TU0EzVFBPUVhDQVdMUE9VUUdDNFRGRUJRVzRJRFBNWlRHU1kzSk1GV0NBMzNPRUIyR1FaSkFNVjRHS1kzVk9SVVhN
WkpBTU5YVzIzTEpPUjJHS1pKQU41VENBNURJTVVRRVNRMkNJRVhDQVNKQU5KMlhHNUJBTzVRVzQ1QkFQRlhYS0lEVU40UUdXM1RQTzRRSEkyREJPUVFFU0lEQk5VU
UdRWkxTTVVRSEkzWkFOQlNXWTRCQU5WUVdXWkpBUEZYWEs0UkFNUlNXRzJMVE5GWFc0SURCT01RR0tZTFRQRVFHQzRaQU9CWFhHNDNKTUpXR0tMUkE=
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius Raekwon
:Raekwon!~malware@216.14.247.46 PRIVMSG Ill_Song :
NDkyMDc3NmY3NTZjNjQyMDZlNjU2NTY0MjA2MTc0MjA2YzY1NjE3Mzc0MjAzMjMwMjA2ZDY5NmM2YzY5NmY2ZTIwNTI3NTYyNmM2NTczMmU=
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
:SU5YVzQ0M0pNUlNYRUlESk9RUUdJMzNPTVVYQ0FTSkFPNVVXWTNCQU9OU1c0WkJBUEZYWEtJRFVOQlNTQTJMT01aWFhFM0xCT1JVVzYzUkFNWlhYRUlEVU5CU1NB
WkRTTjVZQzIzM0dNWVFIQTMzSk5aMkNBNDNQTjVYQzQ9PT0=
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
:verne.freenode.net 303 Ill_Song :razor genius
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
PING LAG3369395302
:verne.freenode.net PONG verne.freenode.net :LAG3369395302
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 LAG3399442177
:verne.freenode.net PONG verne.freenode.net :LAG3399442177
ISON Razor Genius Raekwon Killah Method
:verne.freenode.net 303 Ill_Song :razor genius
```

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://adrinanthony.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].