NETWORK FORENSICS INVESTIGATION

CMP416 – Unit 2 Assignment

Maximilian Ring
1905726

2174 Words

Contents

1. Introduction

This report covers a forensic investigation that was conducted into an international sporting competition corruption case in cooperation with the NSA. It will go over three different network captures that where forensically analysed by a forensic investigator to fulfil different tasks and aims for each capture. The report will cover each step in the forensic process and explain in depth, for each finding, what was found and how it was found.

1.1 Tools Used

The following tools were used throughout the forensic investigation:

- Wireshark & Tshark (Network analysis tools)
- Sublime Text 3 (Text editor and IDE)
- CyberChef (Online "Cyber Swiss Army Knife")
- Obsidian (Note Taking Software)
- Imgcat (MacOS software package to display images in terminal)
- Maps.co (Map Maker that allows plotting of coordinates)
- SilentEye (Stenography Tool that is used to hide and recover text in/from images)
- Unzip (Software to unzip archives within the terminal)
- Strings (Software to find strings to ASCII characters within a file)

2. Procedure & Findings

2.1 Capture 1

The forensic investigator loaded the given capture into WireShark and firstly had a brief overlook over the packets that where captured. They found two things of interest in this capture, firstly on the 11/07/2014 at 20:22 there was a SMB transfer of a file named *Documents.zip* that was downloaded from a SMB Share at the IP address 172.29.1.20 by a user that was using a machine with the IP address 172.29.1.23 and using the user *fox-ws\test*.

Using Wireshark's export object feature the forensic investigator exported the downloaded zip file to their machine and extracted it in a safe environment.

The zip file had the following architecture:

- Documents.zip (Un-zipped)
  - Enter the WuTang
    - track6.docx
    - track10.docx
  - Actual Documents
    - GoT Spoilers.docx
    - NorthKorea.docx
    - PiD.docx
  - Chess Boxing
    - NK.jpg
    - Rules 1..docx
    - Rules 2.docx
    - Rules 3.docx
    - Rules 4.docx
    - Rules 5.docx
    - Rules 6.docx
    - Rules 7.docx
  - More Documents
    - BillOfRights.txt
    - NorthKorea.jpeg
  - untitled folder.zip (Un-zipped)
    - untitled folder
      - untitled folder 2
        - untitled folder
          - untitled folder
            - SilentEye (Empty Folder)

The investigator then went though each file and folder to understand the contents, first in the Enter the WuTang folder the document titled *track6.docx* contained a list of usernames that were encoded using Base64 the investigator knew this due to the equals signs at the end of the text, this being a common sign of a string being Base64 encoded, the researcher copy & pasted the encoded text into CyberChef and when selecting Base64 decode the list of usernames was retrieved. It can be found in Appendix 1.

Two names in particular stood out *Kim Ill-Song* and *Matt Cassel* as they did not follow the pattern of the other names based on the other evidence within the documents zip file linking more to North

Korea the first name however is most likely the one of greater importance due to its North Korean origin while the second name sounds more like it comes from the western part of the world.

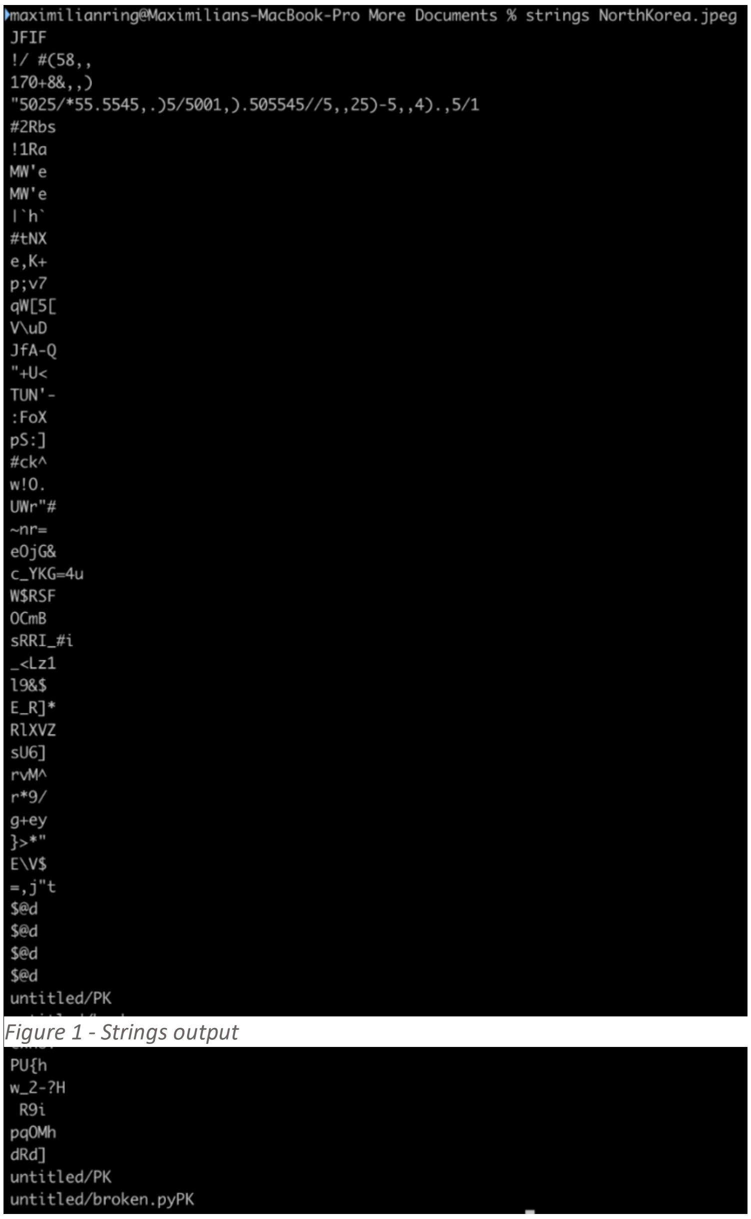
The other document in the folder *track10.docx* was a Base64 encoded song lyrics from what the investigator could tell.

Upon further analysis the investigator discovered that every document in the folder structure had its contents encoded with Base64.

The folders *Actual Documents* and *Chess Boxing* had no files of further importance, but they can be viewed within the archive that will be submitted alongside the report.

Next the investigator inspected the *More Documents* folder. This contained the United States Bill of Rights in a TXT document, which was noted as an oddity since all other documents where docx files up until this point and the document's contents where not Base64 encoded, and a JPEG image of the North Korean flag which had a different reported file size by the file itself compared to the space it was actually taking up on the disk.

This prompted the investigator to check the file using the terminal utility *strings* to check for any ASCII strings that might be hidden within the file. The output (Figure 1) revealed that not only was there hidden strings within the image, but it appeared there was a file archive of some sort hidden within the image. The investigator made a copy of the JPEG file and changed its extension to .zip, they then used the *unzip* terminal utility to try and extract any files within the archive that was hidden. This extracted a file *broken.py* which appeared to be a python script with some minor syntax issues. The broken and fixed version can be found in Appendix 2 and Appendix 3.



```
maximilianring@Maximilians-MacBook-Pro More Documents % strings NorthKorea.jpeg
JFIF
!/ #(58,,
170+8&,,)
"5025/*55.5545,.)5/5001,).505545//5,,25)-5,,4).,5/1
#2Rbs
!1Ra
MW'e
MW'e
|`h`
#tNX
e,K+
p;v7
qW[5[
V\uD
JfA-Q
"+U<
TUN'-
:FoX
pS:]
#ck^
w!O.
UWr"#
~nr=
e0jG&
c_YKG=4u
W$RSF
OCmB
sRRI_#i
_<Lz1
l9&$
E_R]*
RlXVZ
sU6]
rvM^
r*9/
g+ey
}>*"
E\V$
=,j"t
$@d
$@d
$@d
$@d
untitled/PK
```

Figure 1 - Strings output

```
PU{h
w_2-?H
 R9i
pqOMh
dRd]
untitled/PK
untitled/broken.pyPK
```

After this the investigator checked all the other image files within the folder structure but found that none had hidden strings or archives within them.

The python script seems to be a way of encrypting a string using a text file as a sort of key, within the python script the word bill is used as a variable to store the documents contents as a string, this made the investigator believe the other file in the directory, *BillOfRights.txt*, was used as the key to encrypt a string using this script. This was noted down for later use.

Other than this the *Documents.zip* file also contained another zip file that was unzipped and just contained a number of untitled folders and a final empty folder named *SilentEye*. Upon some research the investigator found that this was an old stenography tool that was used to hide text within images and recover hidden text from images. The researcher downloaded this tool to a Windows Virtual Machine and attempted to use the tool on the different images found within the zipped file. None of them contained any hidden messages so the investigator moved on.

In addition to the zip file the investigator also discovered a email conversation between Edward Snowden (snowedinedward@aol.com) and Julian Assange (wikiofleaks@aol.com) on the 11/07/2014 that lasted from around 20:13 until 20:23. Mr Snowden sent a message stating that he had a pcap he wanted Mr Assange to look though, Mr Assange responded that he would do so and also asked if Mr Snowden had any further documents to send. Mr Snowden replied that he will send though more documents when able, and Mr Assange thanked him. This conversation can be seen in Appendix 4.

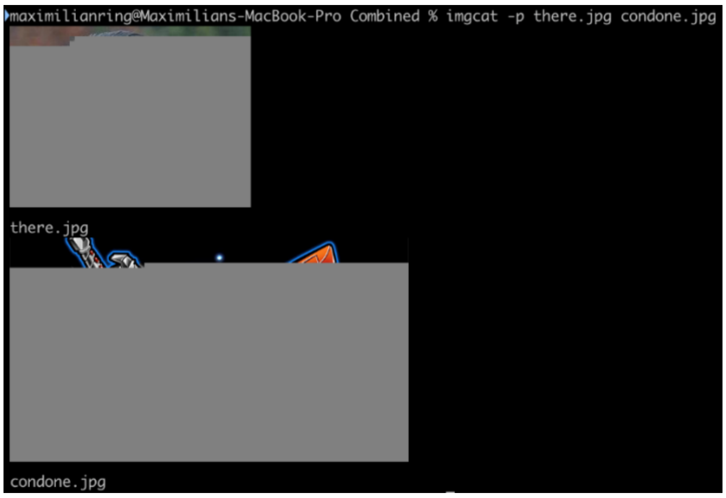
2.2 Capture 2

When analysing the capture file in WireShark the investigator filtered out all traffic but FTP related packets. They found that on the 03/07/2014 at 20:36 a user *Ill_Song* logged into the server *Super Secret Server* with the password *Ill_Song* and accessed the directory */home/Ill_Song*. The user then downloaded two zip files *sandofwhich.zip* and *ojd34.zip*. These zip files contained a number of JPG's all titled a different single word. Upon analysis and comparing all the words from the image files to famous Edward Snowden quotes the investigator was unable to find a matching quote. They expanded the analysis to other than FTP related packets and found two emails that were sent on the 03/07/2014 at 20:37 and 20:39 between kim.illsong@aol.com and da.genius36@aol.com. These emails can be found in Appendix 5. The investigator was able to recover the attachments by viewing the TCP streams data and exporting the bytes of the zip file as a file. This gave the other three zip files as *34jdsioj.zip*, *breaking_bad_season_6.zip* and *canc3l.zip*.

Once these were unzipped as well and contained, again, images all just titled with a single word the investigator found a matching Edward Snowden quote:

`I can't in good conscience allow the U.S. government to destroy privacy, internet freedom and basic liberties for people around the world with this massive surveillance machine they're secretly building.` - Edward Snowden (Mittal, 2017)

The investigator also found that three images had the start bytes of a JPG file these where *I.jpg*, *there.jpg* and *condone.jpg*. This was done by using *imgcat* on all the images within the zip files which allows to view the image contents within the terminal. These three images showed the start of an image while the rest displayed a corrupt image logo.



```
maximilianring@Maximilians-MacBook-Pro Combined % imgcat -p there.jpg condone.jpg
```

Figure 2 - imgcat showing the start of two images

The investigator than used the following command to reconstruct the image:

```
cat I.jpg cant.jpg in.jpg good.jpg conscience.jpg allow.jpg the.jpg
U.S..jpg government.jpg to.jpg destroy.jpg privacy.jpg internet.jpg
freedom.jpg and.jpg basic.jpg liberties.jpg for.jpg people.jpg
around.jpg world.jpg with.jpg this.jpg massive.jpg surveillance.jpg
machine.jpg theyre.jpg secretly.jpg building.jpg > Image1.jpg
```

Which produced the following image:

Figure 3 - Recovered Image 1

This is the image of a Chess Set which the investigator believes the corrupt official received.

The investigator also noticed some weird lines and artifacts within the image and knew this is a common sign of stenography being present within an image, they also remembered the presence of the SilentEye folder in the first capture and because of this ran the image though SilentEye (Screenshot can be found in Appendix 6) and recovered the following string of characters.

i2454 2497d2496n2502 2470 2500 2507o2436s2452 2500s2503n2502l2487e2456 2497 2500h2485l2487 2470b2490e2491a2501m2466 2483a2501a2501e2505 2497 2500a2486

This string of characters looked very alike the output generated by the fixed python script from capture 1 when the investigator was testing it after they had fixed it, so they added the following line to the python script:

```
print(indexInASCII("BillOfRights", "i2454 2497d2496n2502 2470 2500
2507o2436s2452 2500s2503n2502l2487e2456 2497 2500h2485l2487
2470b2490e2491a2501m2466 2483a2501a2501e2505 2497 2500a2486"))
```

And after they ran the scrip the string *DontTry2BruteForceThisPassword* was recovered, it is unknown to the investigator what the password could be used for, but it could be useful for further investigations.

2.3 Capture 3

The investigator used Wireshark to analyse the third capture and started by searching for the string “Ill-Song” and several messages between Ill-Song and Ann Dercover. They extracted all the messages from the capture and the full conversation can be found in Appendix 7. The method of communication for these messages was a mobile app called TextFree (https://textfree.us/) powered by the Pinger API (https://www.pinger.com/).

In the message packets the investigator was also able to recover phone numbers for both Ill-Song, +1 (406) 9243754, and Ann Dercover, +1 (406) 8522589. After the packets of the conversation there is several requests made to MapQuest each to a different by close to each other set of longitude and latitude. The investor used tshark and grep to filter out these requests and save them to a text file and then used the Sublime Text replace function with a regex expression to remove the extra packet details before and after the longitude and latitude info. This created a list of sets of latitude and longitudes, which can be found in Appendix 8.

The investigator then took the list of coordinates and used a MapMaker (https://maps.co/gis/) to place the locations on a map, this revealed the number 17 as seen in figure 4.

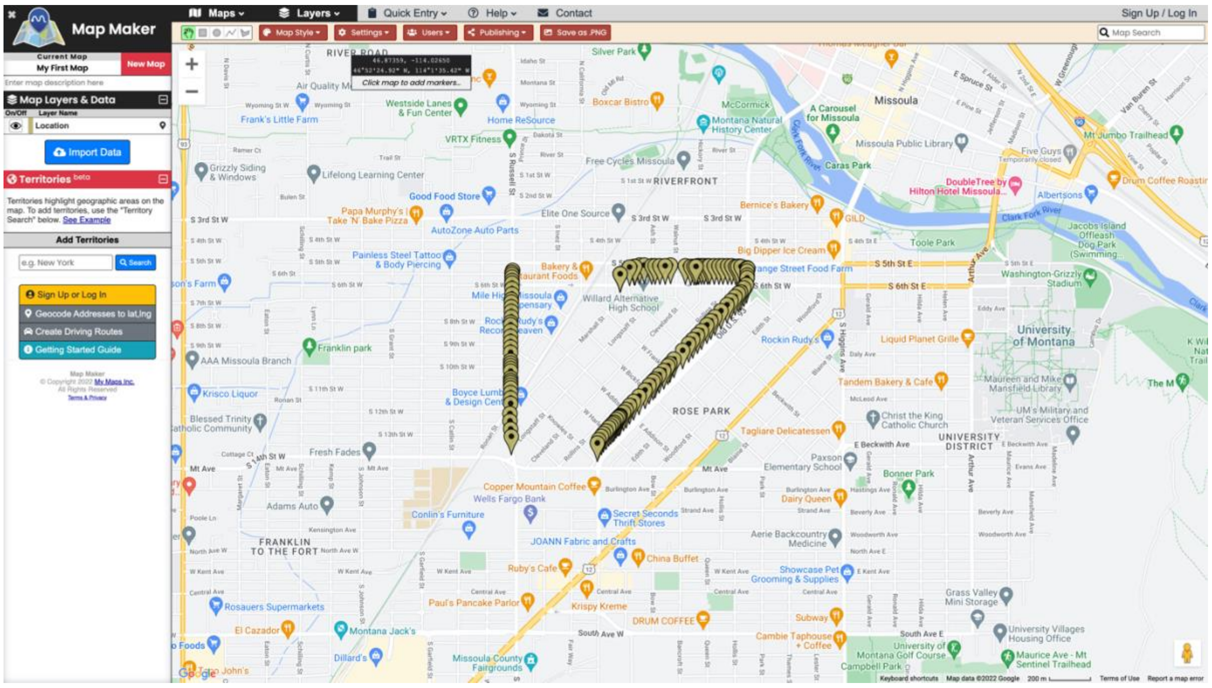


Figure 4 - List of Longitudes and Latitudes plotted on map

3. Discussion

3.1 Network Forensics Practice

Since the only evidence there was for this investigation, the methodology used was rather simple:

- Take the brief of the Capture file and turn it into aims
- Analyse the PCAP file for evidence
- Follow any leads that are found in the PCAP file
- Document any files/communications found in the process

This methodology was easy to follow and worked very efficiently at getting the required evidence.

3.2 Network Data

The most important information from the network data was the contents and dates for this investigation as those where the focus points given to the investigators by the brief. The contents allowed the investigators to reconstruct files and the timestamps allowed them to put a time and date on actions being taken as well as fulfil the aim of capture 3 fully as the year of the capture gave context to the date mentioned by the conversation only contained a month and day.

3.3 Critical Evaluation

For the most part the investigation was challenging but not overly difficult, the most time consuming and difficult was certainly the reconstructing of the images but when the investigator came up with the idea of using imgcat to display the image in the terminal instead of saving each attempt at a new image as a file and then opening that file, it massively increased the speed at which the investigator was able to finish that section of the investigation.

3.4 Reflective Component

There where multiple attempts to impede a forensic investigation including Base64 encoding, Stenography, custom encryption of a string and sending the day of the month of the meeting via a list of coordinates that would show the number if plotted on a map by the suspects. Most of these like the Base64 encoding where extremely basic and easy to spot methods of obfuscation while others like the mapping of the coordinates was a bit more advanced and took more thinking and problem solving. Nevertheless, these where clear attempts by the suspects to hide what they were doing and what they where communicating.

3.5 Evidence Submission

Due to the submission portal (MyLearningSpace) not accepting ZIP files all the evidence along with the investigative notes have been made available to review at the following URL: https://github.com/maxgestic/CMP416-Forensic-Investigation

References

Mittal, T., 2017. *Edward Snowden's quotes on the importance of privacy.* [Online] Available at: https://yourstory.com/2017/06/edward-snowden-quotes-privacy [Accessed 11 Dec 2022].

Appendices

Appendix 1 – Base64 Decoded Document (track6.docx) List of Usernames

The Mystery of Chess Boxing:

(usernames)

Mr. Method

Kim Ill-Song

Mr. Razor

Mr. Genius

Mr. G. Killah

Matt Cassel

Mr. I. Deck

Mr. M Killa

Mr. O.D.B.

Mr. Raekwon

Mr. U-God

Mr. Cappadonna (possibly)

John Woo?

Mr. Nas

Appendix 2 – Broken Python Script (broken.py)

```
def fileToString(pathToFile):
	f = open(pathToFile, "r")
	strs = ""
	#adds each line of the file to the strs string
	for line in f.readlines():
		strs+=line
	return strs
def ASCII():
	#number of ASCII characters
	NumOfASCII == 0
	#returns list of all ASCII characters
	return "".join([chr(i) for i in range(NumOfASCII)])
def sumName(name):
	sums=0
	#sums the indices in ASCII of all the characters in name
	for x in name:
		sums+=ord(x
	return sums
def indexInFile(password):
	indices = []
	ASCIIArray = ASCII()
	#populates an array of indices to be used by the encoder
	for chrs in password:
		indices.append(ASCIIArray.index(chrs)+sumName(name)*2
	return indices
def indexInASCII(name):
	indices = []
	ASCIIArray = ASCII()
	#split on all non-numeric characters
	#remove first index because it is blank
	indexList = re.split("[^\d]",encoded)[1:]
	#converts encoded characters to ASCII
	for index in indexList:
		indices.append(ASCIIArray[int(index) -
(sumName(name)*2)])
	#returns decoded message
	return "".join(indices)
def encode(name):
	#returns a list of indices to be used for encoding
	indices = indexInFile(password,name)
	#convert file associated with name to a string
	bill = fileToString("./%s.txt"%name)
	encoded = ""
	#add letter in file plus index of the letter in the file to
the encoded string
	for index in indices:
		encoded+=bill[index]+str(index)

	return encoded
```

Appendix 3 – Fixed version of Python Script

```
import re

def fileToString(pathToFile):
      f = open(pathToFile, "r")
      strs = ""
      #adds each line of the file to the strs string
      for line in f.readlines():
            strs+=line
      return strs

def ASCII():
      #number of ASCII characters
      NumOfASCII = 128
      #returns list of all ASCII characters
      return "".join([chr(i) for i in range(NumOfASCII)])

def sumName(name):
      sums=0
      #sums the indices in ASCII of all the characters in name
      for x in name:
            sums+=ord(x)
      return sums

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

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

def encode(password,name): # function used to encode password with a
file
      #returns a list of indices to be used for encoding
      indices = indexInFile(password,name)
      #convert file associated with name to a string
```

```
	bill = fileToString("./%s.txt"%name)
	encoded = ""
	#add letter in file plus index of the letter in the file to 
the encoded string
	for index in indices:
		encoded+=bill[index]+str(index)

	return encoded
```

Appendix 4 – Email Conversation between Edward Snowden and Julian Assange

Many thanks.

-----Original Message-----

From: Edward Snowden snowedinedward@aol.com

To: wikiofleaks wikiofleaks@aol.com

Sent: Fri, Jul 11, 2014 2:07 pm

Subject: Re: Urgent

Julian,

I will do so as soon as I can.

-----Original Message-----

From: Julian Assange <wikiofleaks@aol.com>

To: snowedinedward <snowedinedward@aol.com>

Sent: Fri, Jul 11, 2014 2:05 pm

Subject: Re: Urgent

Snowedinedward,

I will look at that pcap as soon as I can. Can you also send me the documents you have collected during research of that recent scandal.

-----Original Message-----

From: Edward Snowden <snowedinedward@aol.com>

To: wikiofleaks <wikiofleaks@aol.com>

Sent: Fri, Jul 11, 2014 2:03 pm

Subject: Urgent

Hello friend,

I have a pcap I would like you to look through. Could be useful.

Snowedinedward

Appendix 5 – Emails between Kim Ill-Song and da.genius36

Email 1

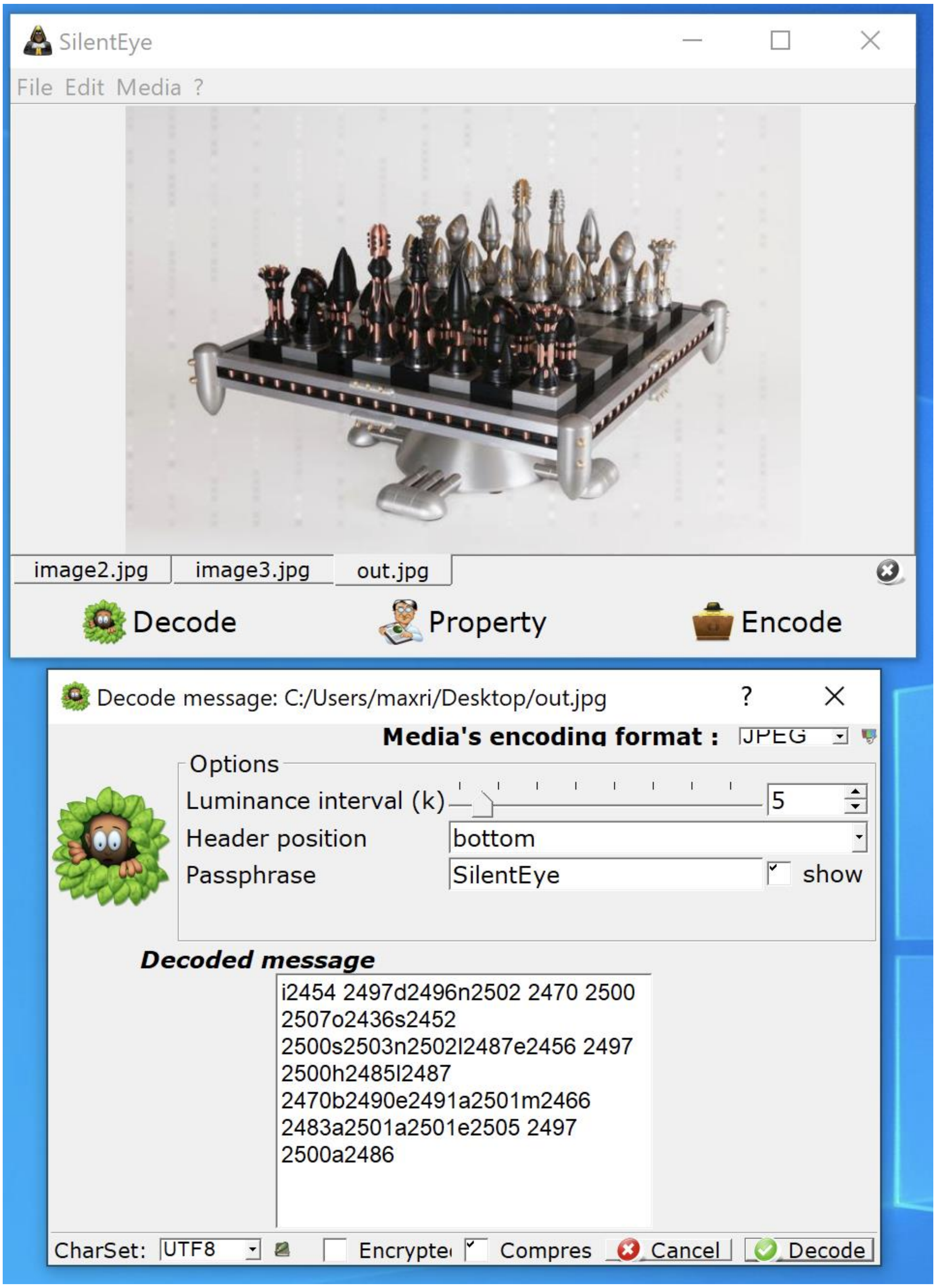
[{"isCertifiedMail":false,"inlineImgIds":[],"isVirusRepaired":false,"replyAllCcAddrs":"","isSuccess":tru
e,"baseSubject":"Urgent","spamReasonMsg":"","attachments":[{"size":35764,"subType":"zip","url":"
http://mail.aol.com/38602-516/aol-6/en-us/mail/get-
attachment.aspx?uid=431&folder=Inbox&partId=3&saveAs=34jdsioj.zip","name":"34jdsioj.zip","bas
eType":"application","id":3},{"size":23790,"subType":"zip","url":"http://mail.aol.com/38602-
516/aol-6/en-us/mail/get-
attachment.aspx?uid=431&folder=Inbox&partId=4&saveAs=breaking_bad_season_6.zip","name":"b
reaking_bad_season_6.zip","baseType":"application","id":4}],"folderType":"Inbox","hasLinksAndIma
ges":false,"displayTo":"da.genius36 <da.genius36@aol.com>","sentTime":"03-Jul-2014 16:19:42
+0000","isLikelyForward":false,"spamReason":"none","displayBcc":"","receivedTime":14044187820
00,"bcc":[],"hasImages":false,"isRead":true,"knownSender":true,"to":[{"shorterDisplayForm":"da.ge
nius36","displayForm":"da.genius36
<da.genius36@aol.com>","userInputForm":"da.genius36@aol.com"}],"isSeen":true,"isVoicemail":fal
se,"isVirusScanned":true,"uid":"431","hasAttachments":true,"replyAllAddrs":"kim.illsong@aol.com",
"isSpam":false,"messageID":"<8D1651341C79919-29BC-DD4A@webmail-
va010.sysops.aol.com>","hasFailedSpfCheck":false,"displayFrom":"Kim Illsong
<kim.illsong@aol.com>","subject":"Re:
Urgent","isFlagged":false,"replyToAddrs":"kim.illsong@aol.com","cc":[],"isDraft":false,"body":"<htm
l>\r\n<body class=\"AOLWebSuite\" style=\"background-color: white;font-family: Arial,sans-
serif;font-size: 10pt;border: 0px;\">\r\n\r\n<div id=\"AOLMsgPart_2_110a97b3-d220-4068-b43a-
c4a0d2f6f4e3\">\r\n<font color=\"black\" size=\"2\" face=\"arial\">Ask<font size=\"2\"> and<font
face=\"Arial, Helvetica, sans-serif\"> you shall receive. You know where to find
it.</font></font>\n\n<div> <br>\n\n</div>\n\n\n\n<div> <br>\n\n</div>\n\n\n\n<div>
<br>\n\n</div>\n\n\n\n<div style=\"font-family:arial,helvetica;font-size:10pt;color:black\">-----
Original Message-----<br>\n\nFrom: The Gza &lt;<a
href='mailto:da.genius36@aol.com'>da.genius36@aol.com</a>&gt;<br>\n\nTo: kim.illsong &lt;<a
href='mailto:kim.illsong@aol.com'>kim.illsong@aol.com</a>&gt;<br>\n\nSent: Thu, Jul 3, 2014 2:18
pm<br>\n\nSubject: Urgent<br>\n\n<br>\n\n\n\n\n\n\n<div id=\"AOLMsgPart_0_9d590ffb-d065-
4b77-86ec-1baf46137daa\" style=\"margin: 0px;font-family: Tahoma, Verdana, Arial, Sans-Serif;font-
size: 12px;color: #000;background-color: #fff;\">\n\n<pre style=\"font-size: 9pt;\"><tt>You have
made a bold claim but i&#39;d like to see some proof.\n</tt></pre>\n</div>\n\n
\n\n\n\n</div>\n\n</font>\r\n</div>\r\n\r\n</body>\r\n</html>\r\n","inputFrom":"kim.illsong@a
ol.com","canUnsubscribe":false,"references":"<8D1651316342A3C-21DC-B0AB@webmail-
d267.sysops.aol.com>","from":[{"shorterDisplayForm":"Kim Illsong","displayForm":"Kim Illsong
<kim.illsong@aol.com>","userInputForm":"kim.illsong@aol.com"}],"isOfficialMail":false,"displayCc":
"","folder":"Inbox"}]

Email 2

[{"isCertifiedMail":false,"inlineImgIds":[],"isVirusRepaired":false,"replyAllCcAddrs":"","isSuccess":tru
e,"baseSubject":"another","spamReasonMsg":"","attachments":[{"size":48836,"subType":"zip","url"
:"http://mail.aol.com/38602-516/aol-6/en-us/mail/get-
attachment.aspx?uid=433&folder=Inbox&partId=3&saveAs=canc3l.zip","name":"canc3l.zip","baseTy
pe":"application","id":3}],"folderType":"Inbox","hasLinksAndImages":false,"displayTo":"da.genius36
<da.genius36@aol.com>","sentTime":"03-Jul-2014 16:22:05
+0000","isLikelyForward":false,"spamReason":"none","displayBcc":"","receivedTime":14044189250
00,"bcc":[],"hasImages":false,"isRead":true,"knownSender":true,"to":[{"shorterDisplayForm":"da.ge

nius36","displayForm":"da.genius36
<da.genius36@aol.com>","userInputForm":"da.genius36@aol.com"}],"isSeen":true,"isVoicemail":fal
se,"isVirusScanned":true,"uid":"433","hasAttachments":true,"replyAllAddrs":"kim.illsong@aol.com",
"isSpam":false,"messageID":"<8D16513974C02E6-29BC-DD83@webmail-
va010.sysops.aol.com>","hasFailedSpfCheck":false,"displayFrom":"Kim Illsong
<kim.illsong@aol.com>","subject":"another","isFlagged":false,"replyToAddrs":"kim.illsong@aol.com
","cc":[],"isDraft":false,"body":"<html>\r\n<body class=\"AOLWebSuite\" style=\"background-color:
white;font-family: Arial,sans-serif;font-size: 10pt;border: 0px;\">\r\n\r\n<div
id=\"AOLMsgPart_2_e260b467-c994-4550-a00a-455c1708a08d\">\r\n<font color=\"black\"
size=\"2\"
face=\"arial\"><br>\n\n</font>\r\n</div>\r\n\r\n</body>\r\n</html>\r\n","inputFrom":"kim.illson
g@aol.com","canUnsubscribe":false,"from":[{"shorterDisplayForm":"Kim
Illsong","displayForm":"Kim Illsong
<kim.illsong@aol.com>","userInputForm":"kim.illsong@aol.com"}],"isOfficialMail":false,"displayCc":
"","folder":"Inbox"}]

Appendix 6 – Screenshot of SilentEye Decoding



Decode message: C:/Users/maxri/Desktop/out.jpg

Media's encoding format : JPEG

Options

Luminance interval (k) 5

Header position bottom

Passphrase SilentEye [✓] show

Decoded message

```
i2454 2497d2496n2502 2470 2500
2507o2436s2452
2500s2503n2502l2487e2456 2497
2500h2485l2487
2470b2490e2491a2501m2466
2483a2501a2501e2505 2497
2500a2486
```

CharSet: UTF8 [ ] Encryptei [✓] Compres

Appendix 7 – Capture 3 Extracted Conversation

Kim Ill-Sung: Good afternoon, Ann.

Ann: who is this?

Kim Ill-Sung: Castling.

Ann: where are you?

Kim Ill-Sung: I know I can't tell you that.

Ann: Do you know that there are people investigating Kim Ill-Song?

Kim Ill-Sung: Of course. However, they will never know it is me behind the bribes.

Ann: still we should be careful. Pay attention. I want to meet in September at 5PM.

Kim Ill-Sung: At our old meetup spot?

Ann: yes

Kim Ill-Sung: What day?

Ann: I told you to pay attention.

Appendix 8 – List of Longitude and Latitudes

46.85661315917969, -114.01860809326172
46.85693359375, -114.01863098144531
46.85727310180664, -114.01868438720703
46.857601165771484, -114.01866912841797
46.858055114746094, -114.01866149902344
46.8582878112793, -114.01864624023438
46.858524322509766, -114.01863861083984
46.858734130859375, -114.01864624023438
46.85884475708008, -114.01864624023438
46.858943939208984, -114.01864624023438
46.859046936035156, -114.01864624023438
46.85914993286133, -114.01864624023438
46.859466552734375, -114.01864624023438
46.85957717895508, -114.01864624023438
46.85969161987305, -114.01864624023438
46.85980987548828, -114.01864624023438
46.85993194580078, -114.01864624023438
46.86029052734375, -114.01863098144531
46.86052322387695, -114.01863861083984
46.860755920410156, -114.01863098144531
46.86098861694336, -114.01863098144531
46.861228942871094, -114.01863861083984
46.86147689819336, -114.01863098144531
46.86159896850586, -114.01863098144531
46.86183547973633, -114.01862335205078
46.862064361572266, -114.01861572265625
46.862281799316406, -114.01860046386719
46.86248779296875, -114.01860046386719
46.86260223388672, -114.01859283447266
46.86282730102539, -114.0185775756836
46.86306381225586, -114.0185775756836
46.86330032348633, -114.01856231689453
46.863426208496094, -114.0185546875
46.86355209350586, -114.01854705810547
46.86367416381836, -114.01853942871094
46.8637809753418, -114.01853942871094
46.86387252807617, -114.0185317993164
46.863704681396484, -114.01164245605469
46.86370849609375, -114.01163482666016
46.864017486572266, -114.01107025146484
46.864044189453125, -114.01074981689453
46.86404800415039, -114.01071166992188
46.86408996582031, -114.01042175292969
46.86408996582031, -114.01012420654297
46.864078521728516, -114.00962829589844
46.864070892333984, -114.0094223022461
46.86406707763672, -114.00910186767578
46.86407470703125, -114.00875854492188
46.86408233642578, -114.0084228515625
46.864051818847656, -114.0074691772461

46.864044189453125, -114.00716400146484
46.864044189453125, -114.00694274902344
46.86404800415039, -114.00680541992188
46.86405563354492, -114.00670623779297
46.864051818847656, -114.00662231445313
46.864051818847656, -114.00646209716797
46.864051818847656, -114.00627899169922
46.864051818847656, -114.00605773925781
46.864051818847656, -114.00592803955078
46.86405944824219, -114.00563049316406
46.86405944824219, -114.00534057617188
46.86405563354492, -114.00506591796875
46.864051818847656, -114.00477600097656
46.864051818847656, -114.00452423095703
46.864044189453125, -114.0042724609375
46.864044189453125, -114.00414276123047
46.86404037475586, -114.00392150878906
46.863983154296875, -114.00354766845703
46.86393356323242, -114.0035171508789
46.86381912231445, -114.00352478027344
46.863643646240234, -114.0035400390625
46.86354446411133, -114.00354766845703
46.86325454711914, -114.00360107421875
46.86309051513672, -114.00376892089844
46.86293411254883, -114.00396728515625
46.86286163330078, -114.00408172607422
46.862701416015625, -114.00432586669922
46.86253356933594, -114.00457763671875
46.862361907958984, -114.00481414794922
46.86210632324219, -114.00520324707031
46.86183547973633, -114.0055923461914
46.86166000366211, -114.00584411621094
46.86148452758789, -114.00609588623047
046.86122131347656, -114.00647735595703
046.86103057861328, -114.00672912597656
46.860843658447266, -114.00699615478516
46.86065673828125, -114.00727081298828
46.86037063598633, -114.0076675415039
46.859989166259766, -114.00820922851563
46.85979080200195, -114.00848388671875
46.85969161987305, -114.00862121582031
046.859500885009766, -114.00887298583984
46.85930252075195, -114.00914001464844
46.85910415649414, -114.00941467285156
46.8590087890625, -114.0095443725586
46.858829498291016, -114.00979614257813
46.858646392822266, -114.01005554199219
046.858375549316406, -114.01044464111328
46.858123779296875, -114.01079559326172
46.85795211791992, -114.01103973388672
46.8577880859375, -114.01127624511719

46.85765838623047, -114.0114517211914
46.857513427734375, -114.01164245605469
46.85749053955078, -114.01168823242188
046.85747146606445, -114.01171112060547
46.857418060302734, -114.01179504394531
46.85733413696289, -114.01190948486328
46.857234954833984, -114.01204681396484
46.857181549072266, -114.01212310791016
46.85708236694336, -114.01225280761719
046.85697937011719, -114.01237487792969
46.856834411621094, -114.01256561279297
46.85672378540039, -114.01271057128906
46.856597900390625, -114.01287078857422
46.85647201538086, -114.01302337646484
46.856319427490234, -114.01313018798828