ENPM687 CY01
AAKASH RAMAN
UID: 119211663
THE FINAL PROJECT (Comprehensive Forensic Investigation)

Brief Summary of Information:

In this Forensic Investigation, I am the best Forensic Analyst for the Imperial army and was tasked with finding out the original, final form of the malware created by the Rebels along with the messages it was sending to the Rebels across the galaxy. Somehow the Imperial army got a hand of the Rebel Malware Writer's original Hard Drive's disk image that I analyzed using my Forensic knowledge and tactics and answered the questions to determine my conclusions about what the Rebels were planning and what information they obtained. Overall, the Rebels used multiple executable files to cover their tracks and the VeraCrypt disk/file encryption/decryption software to mask their activities. However, all their activities were found as explained in the below sections.

Tools & Assumptions used during the Investigation:

Multiple tools helped me solve this investigation. The main tools I used were **Autopsy**, for analyzing the disk image of the Rebel Malware Writer. The "***File Types***", "***Deleted Files***", "***Encryption Suspected***" and "***Data Sources***" sections of Autopsy helped me greatly in my analysis from start to finish. It also had an "Extraction" feature which helped me extract any interesting files (executables, MP3 files) to my local machine for further analysis.

Another tool that helped me greatly was **Wireshark**. Being a Packet Analyzer tool, Wireshark's ability to capture the packets once an executable ran helped me decipher what messages the Rebel's were trying to send. This helped me get the original messages the Rebels were trying to send after finding out the Rebel Malware Writer's original Malware. Another tool that helped me was **CyberChef**. CyberChef is a web application for encryption, encoding/decoding to multiple formats, compression, and data analysis. It helped me decode a message that the Rebels sent which was in Base64 back to ASCII/English which was crucial to get the key to decrypt the encrypted file. Another small but important tool I used was **Windows Media Player**. This was used to find out if the audio file I suspected which was encrypted was actually an MP3 file. It correctly pointed out that it was not an MP3 file which led me in the right direction.

Finally, the last tool that helped me crack this investigation was **VeraCrypt**. After finding out the encrypted file was encrypted using VeraCrypt, I used VeraCrypt with the key I found to decrypt the suspected encrypted file. Without VeraCrypt, I would not have been able to find out the original Malware executables and plans of the Rebel Malware Writer.

The main assumption I used in this investigation was that of a traditional Forensic Analyst. I suspected typical attacker behavior and hence analyzed the Deleted Files section first which

gave me an idea of the possible encryption/decryption software the Rebel Malware Writer might have used. Next, I analyzed all the possible files in the “File Types” section specifically “exe” files which led me to the suspicious executables. Furthermore, using the knowledge I previously gained I related the evidence to eventually correlate the executables found, the messages they sent, and the VeraCrypt software which gave me the final form of the Rebel Writer’s Malware.

Repository #1 (Forensic Artifacts in the Rebel Malware Writer’s Hard Drive Image):

a. Summary of Evidence Found in the Rebel Malware Writer’s Hard Drive Image

The Rebel Malware Writer Hard Drive disk image had a lot of interesting findings. It largely consisted of multiple Python scripts, Image files in various formats such as “gif”, “png”, “jpg” and a large number of suspicious “exe” and “dll” files. Some web browser artifacts of Mozilla Firefox were also present in a few places. After a thorough analysis of the disk, I found many references to the **VeraCrypt** encryption/decryption software. There was a Beginner Tutorial Guide on VeraCrypt in the form of “jpg” images. In the Deleted Files section, there were some pictures related to VeraCrypt too. Moreover, while analyzing the executable files, the setup file for VeraCrypt was also found. While analyzing HTML files, I found the licenses for VeraCrypt too. Other than VeraCrypt, the relevant suspicious executables “**obiwan.exe**” and “**obiwan2.exe**” were also found in the “Administrator -> My Documents -> Code -> Dist” folder. Both these executables would send messages to the Rebels by communicating with a foreign IP Address. It was these messages that led me to a suspicious encrypted MP3 Audio file.

The suspicious encrypted MP3 audio file, “**not-the-droids-youre-looking-for.mp3**” was found both in the Encryption Suspected folder and also in the audio files section of Autopsy. This was the file into which the Rebel Malware Author hid the true Malware and Death Star Plans the Rebels found out. By executing the final Malware, I got to find out the final decrypted messages the Rebels sent out.

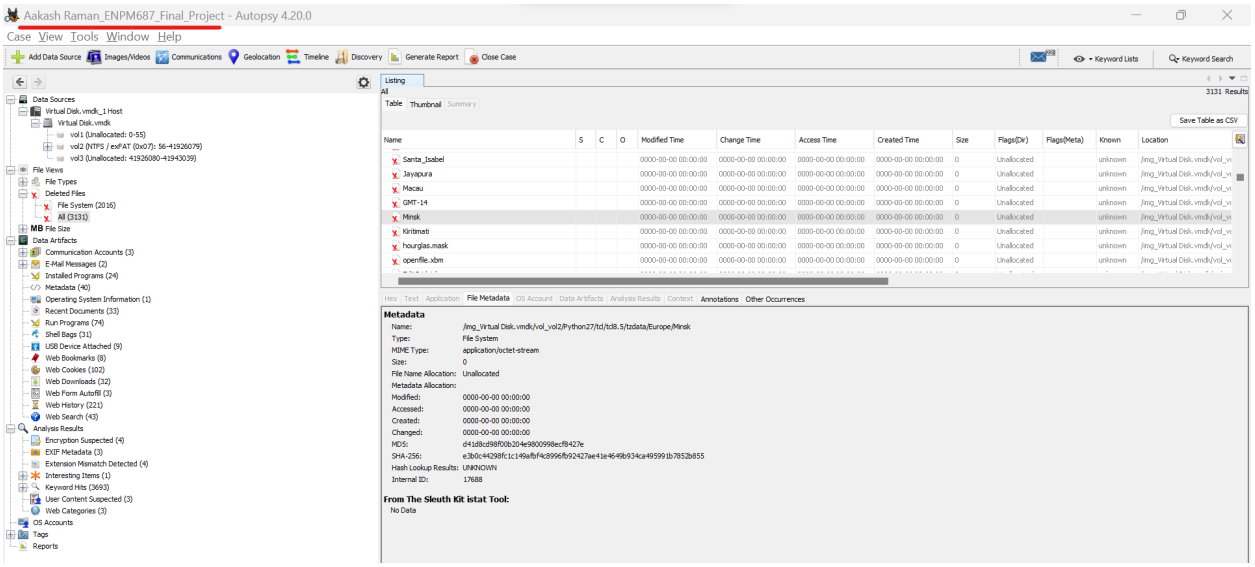
b. Analysis of Relevant Portions of Rebel Malware Writer’s Hard Drive Image

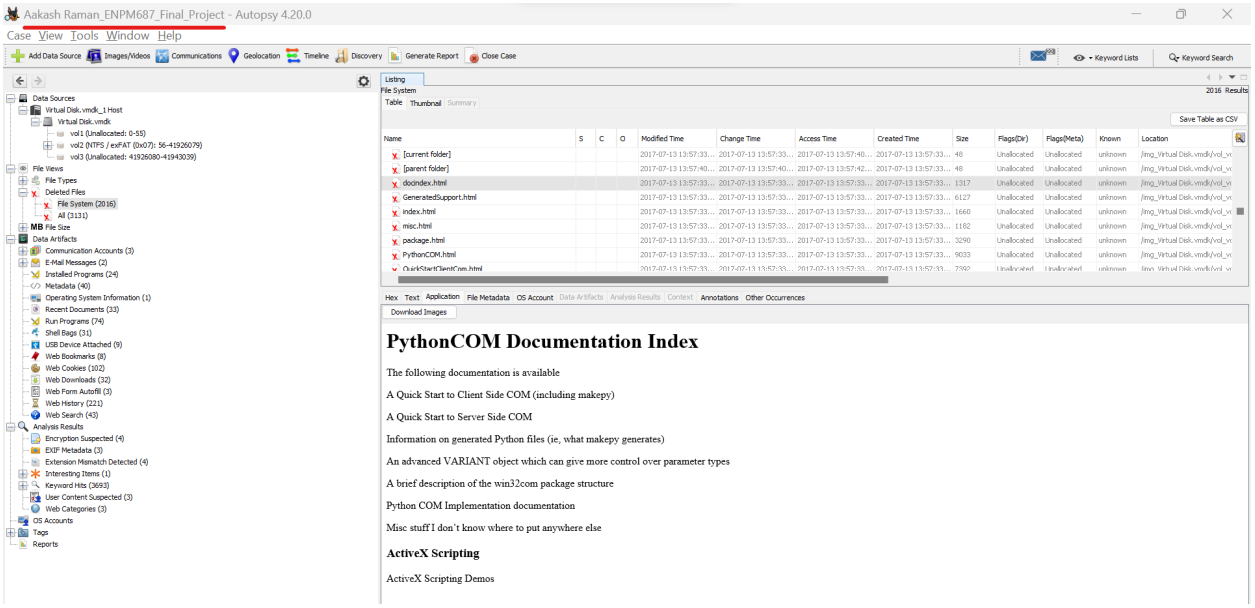
Initial Analysis:

After loading the disk image of the Rebel Malware’s Hard Drive onto Autopsy, I wanted to first check out the “Deleted Files” section of the Rebel Malware Writer’s Hard Drive first because attackers normally delete any malicious they write and that may leave some traces for Forensic Analysts to analyze. As this section consisted of multiple “exe”/Python scripts (and malware can

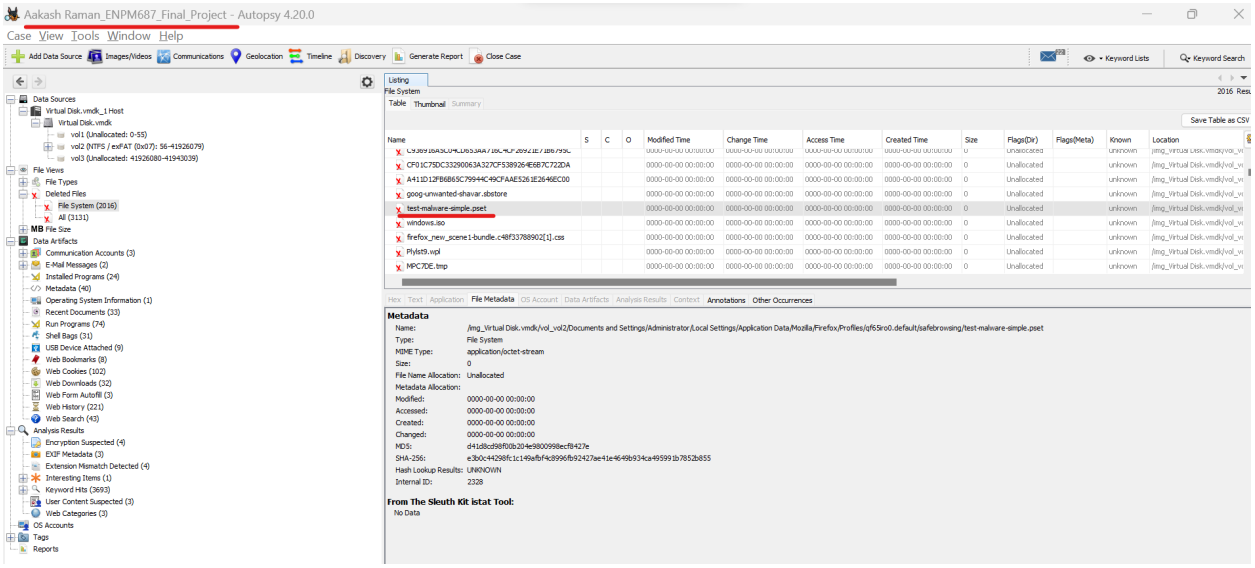
be written in Python), I had a strong suspicion of finding some form of malware traces in this section. However, nothing interesting was found. Given below are the details of my findings in this section.

The "***Deleted Files***" section consisted of details about countries like "***Minsk***, ***Macau***, ***Jayapura***", etc and a lot of Python scripts and documentation like, "***register.py***", "***policy.py***", "***testdynamic.py***", "***PythonCOM Integration*** (***GeneratedSupport.html***, ***index.html***,***misc.html***)", etc. Many "exe" files, "png" files, "dll" files, and "gif" files were found too, however, these findings did not point me toward any malicious activities that could help me find out the Rebel Malware Writer's final malware. Some of my findings are shown below.



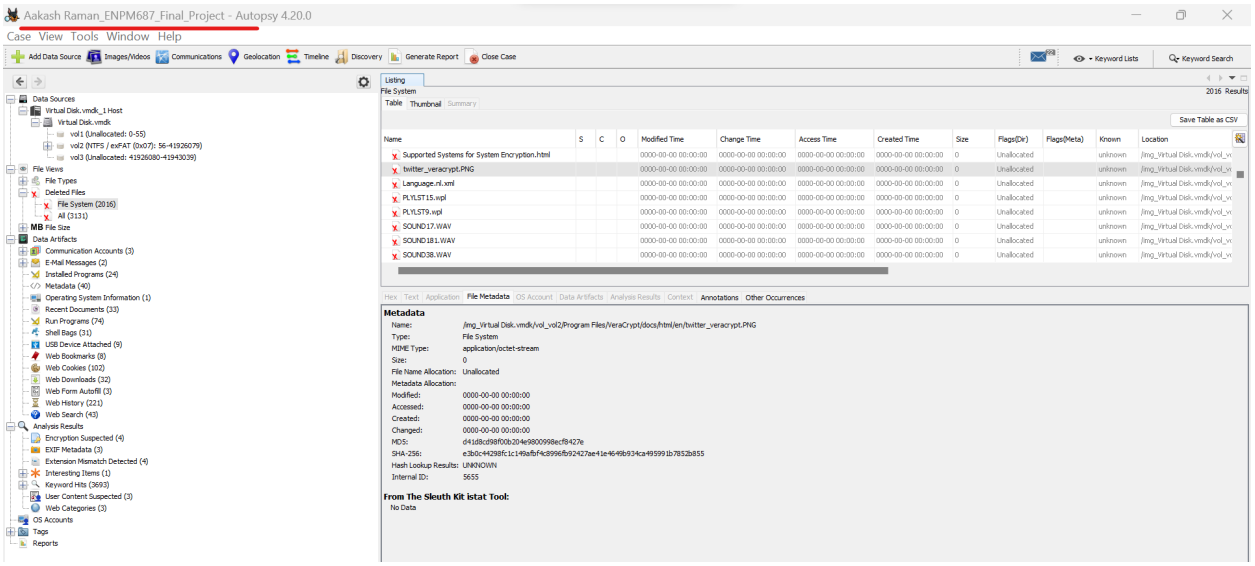


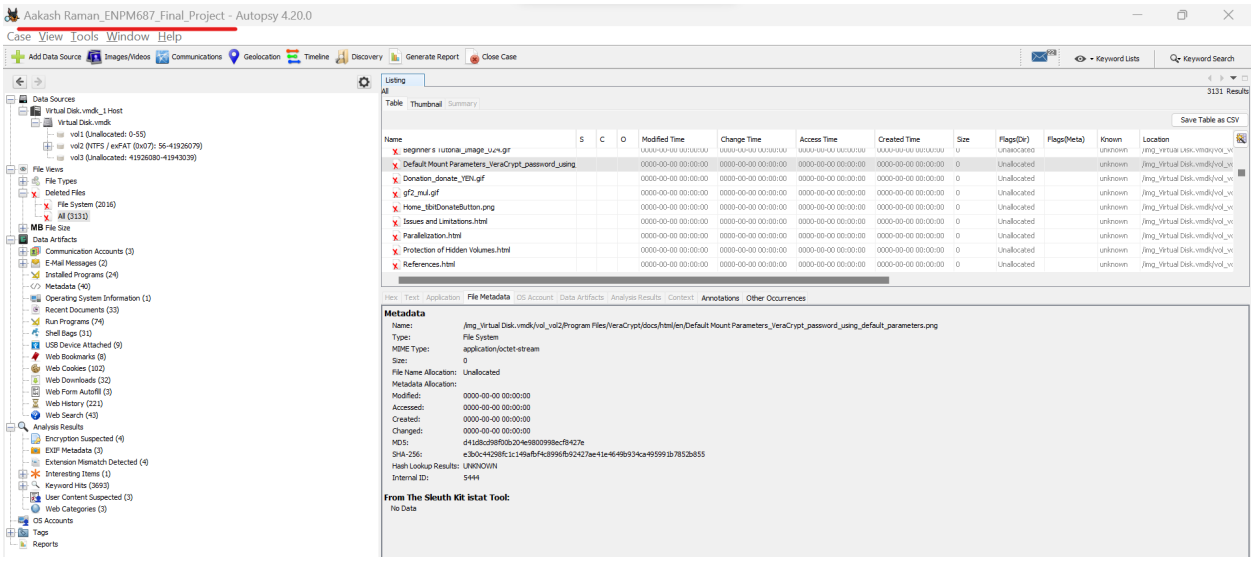
Another initially slightly suspicious finding I found in the Deleted Files section was a file called "***test-malware-simple.pset***" found in Mozilla Firefox's "safebrowsing" folder. I further did some research to find out if this was a malicious script. After some research [1], I found out that this file is not malicious and is by default in the Cache folder of Mozilla Firefox when it's installed.



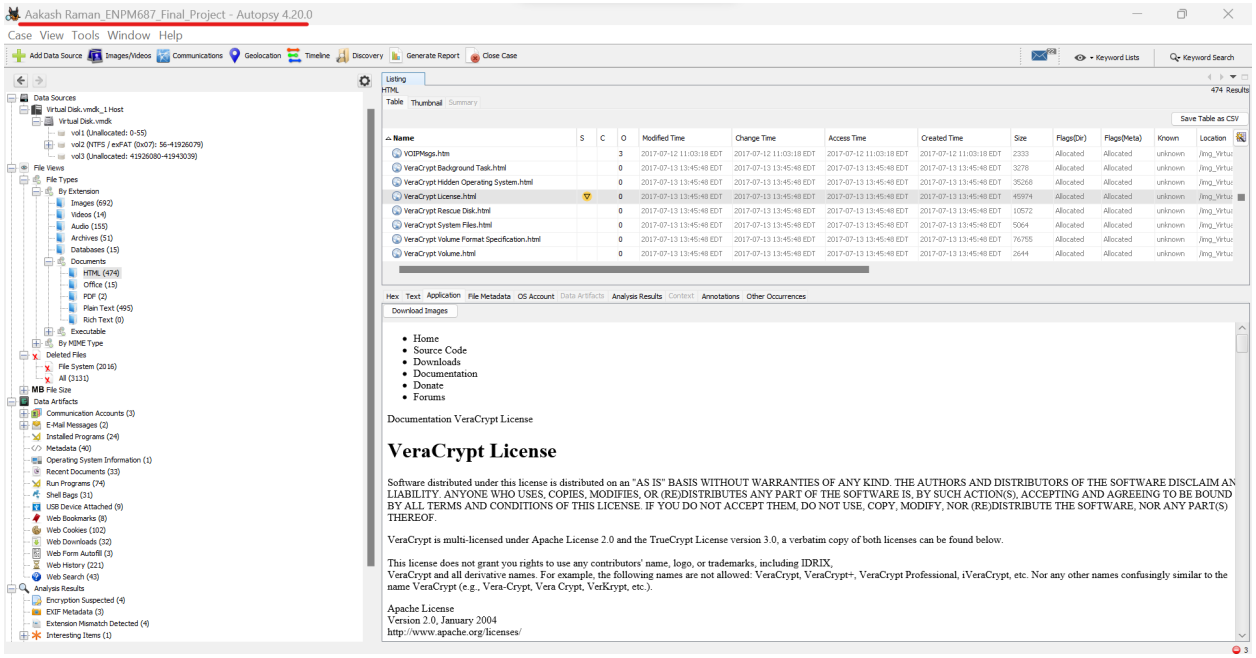
Discovery of VeraCrypt Software:

Although the "Deleted Files" section did not help me directly, it did help me figure out that somewhere on the disk is the "***VeraCrypt***" software, a popular encryption and decryption disk encryption software. I found the paths "***/img_Virtual Disk.vmdk/vol_vol2/Program Files/VeraCrypt/docs/html/en/twitter_veracrypt.png***" and "***/img_Virtual Disk.vmdk/vol_vol2/Program Files/VeraCrypt/docs/html/en/Default Mount Parameters_VeraCrypt_password_using_default_parameters.png***" as shown below. It is not common for such images to be deleted, hence the Malware Writer tried to hide the presence of VeraCrypt.



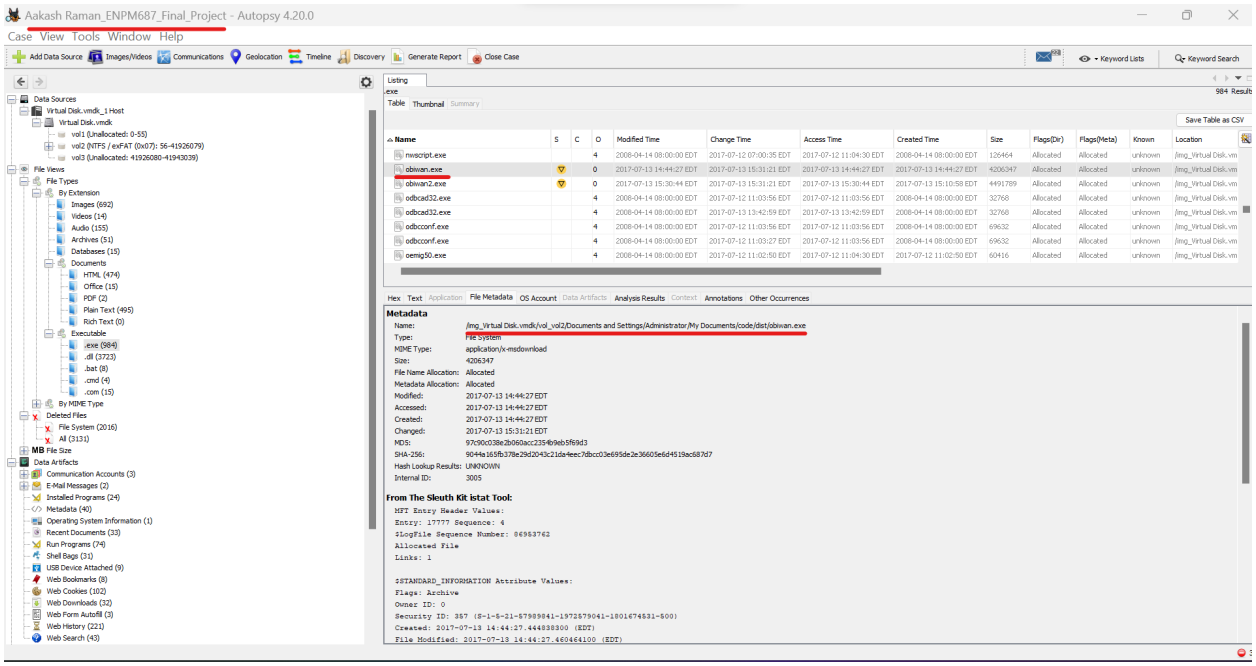


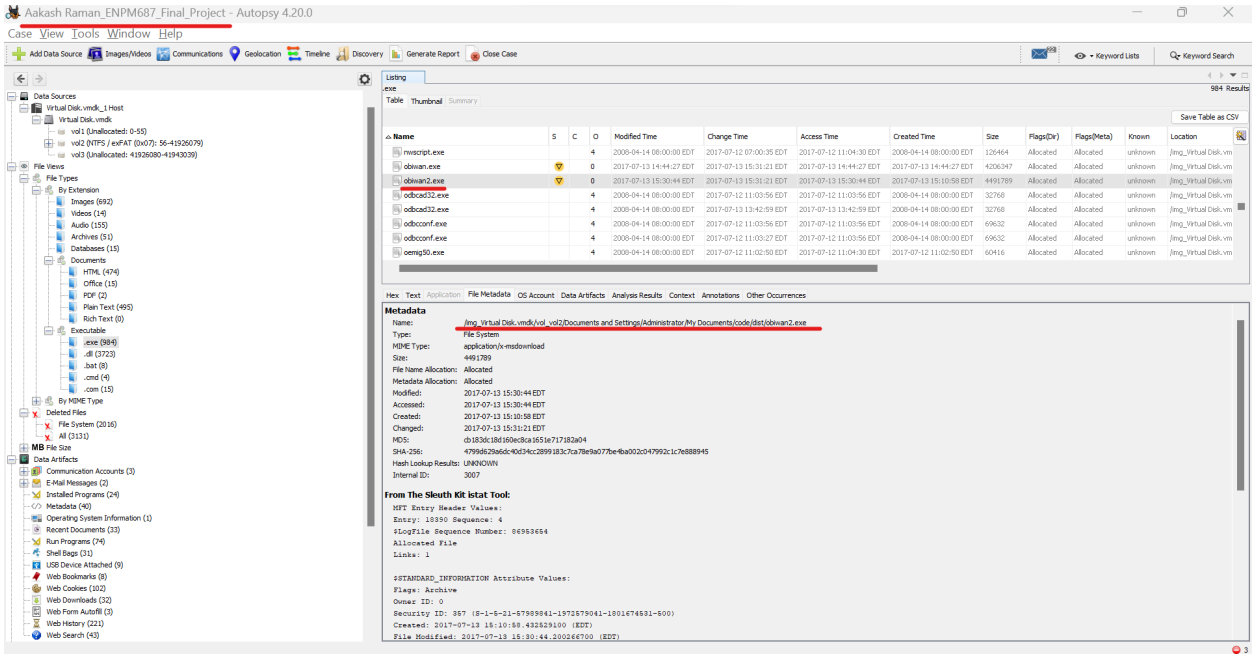
When I was searching some HTML documents, in an attempt to find out any documents that might point me in the right direction, I found the license files for VeraCrypt ("***VeraCrypt License.html, VeraCrypt Rescue Disk.html, VeraCrypt System Files.html***", and many more shown below), which further adds evidence that VeraCrypt might be used by the Rebel Malware Writer for encrypting something.



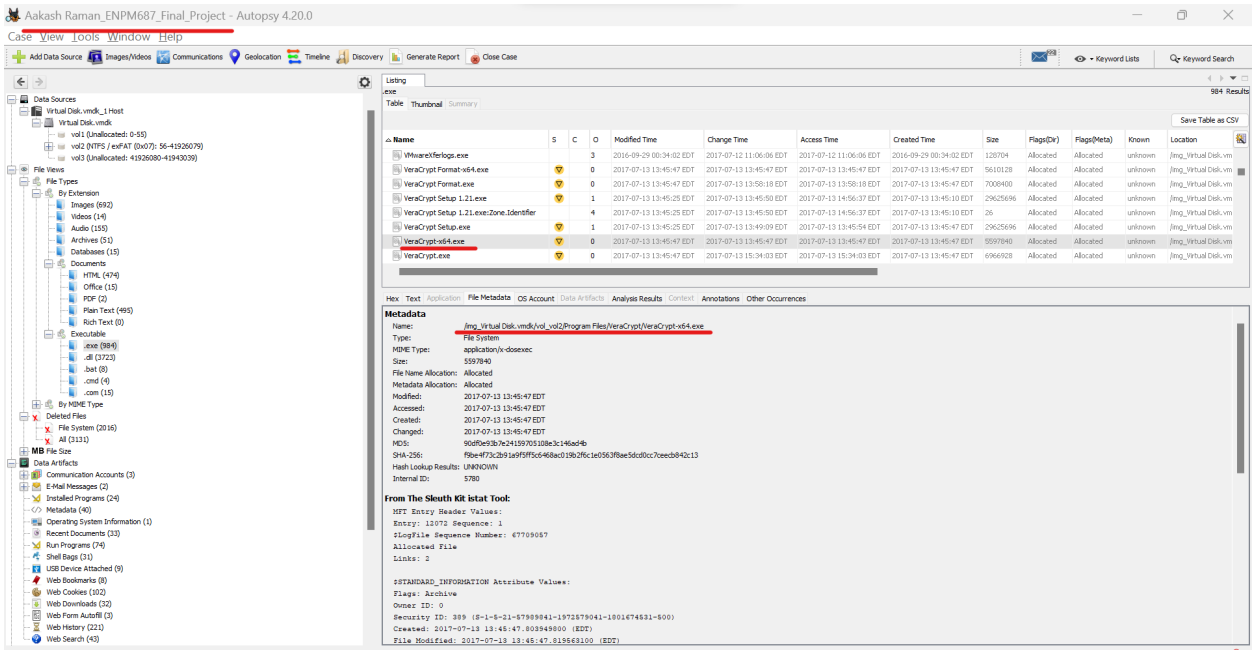
Discovery of Obiwan.exe and Obiwan2.exe:

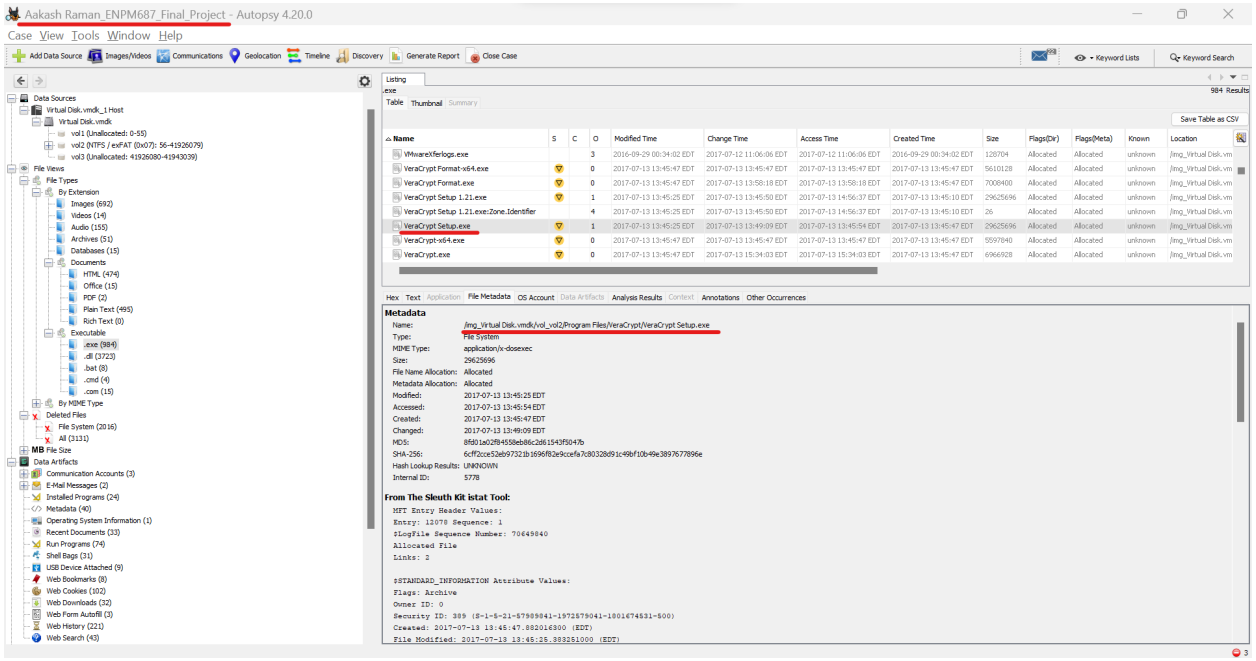
Now, without much success in the “Deleted Files” section, I thought that the malware could be an executable, hence I analyzed all the executables under the File Type section and finally found an executable file that is interesting, “***obiwan.exe***” and “***obiwan2.exe***”, that are 2 files related to the Rebels. Hence, somewhere on the disk are these 2 files. Now, my objective was to find them. Thankfully, Autopsy’s File Metadata feature gives the location of the file. Using this knowledge, I found out that “obiwan.exe”’s file path is “***/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code/dist/obiwan.exe***” as shown below. Similarly, “obiwan2.exe”’s file path is “***/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code/dist/obiwan2.exe***” also shown below.



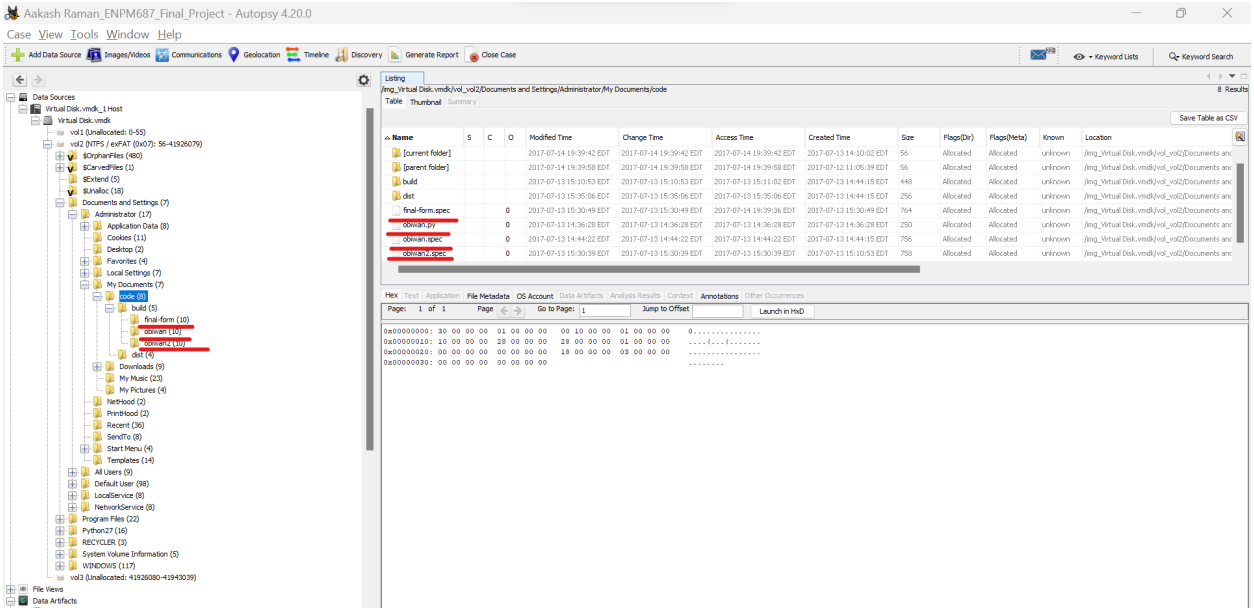


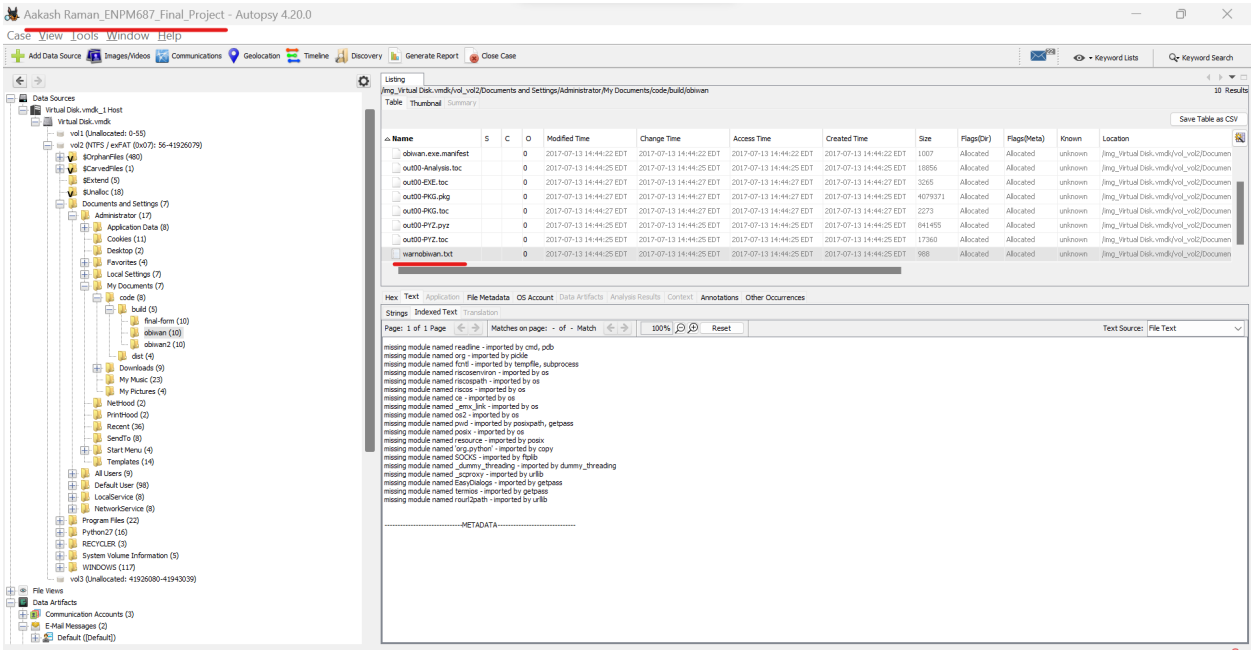
Also, while checking out any interesting “exe” files under the File Types section, I found the VeraCrypt executable file, “***VeraCrypt-x64.exe”*** and its setup file, “***VeraCrypt Setup.exe***”**.** The location of the executable can be found using the File Metadata feature and I found out the location of VeraCrypt 64-bit executable file to be “***/img_Virtual Disk.vmdk/vol_vol2/Program Files/VeraCrypt/VeraCrypt-x64.exe***” and the VeraCrypt Setup executable file to be “***/img_Virtual Disk.vmdk/vol_vol2/Program Files/VeraCrypt/VeraCrypt Setup.exe***”. These are shown below.

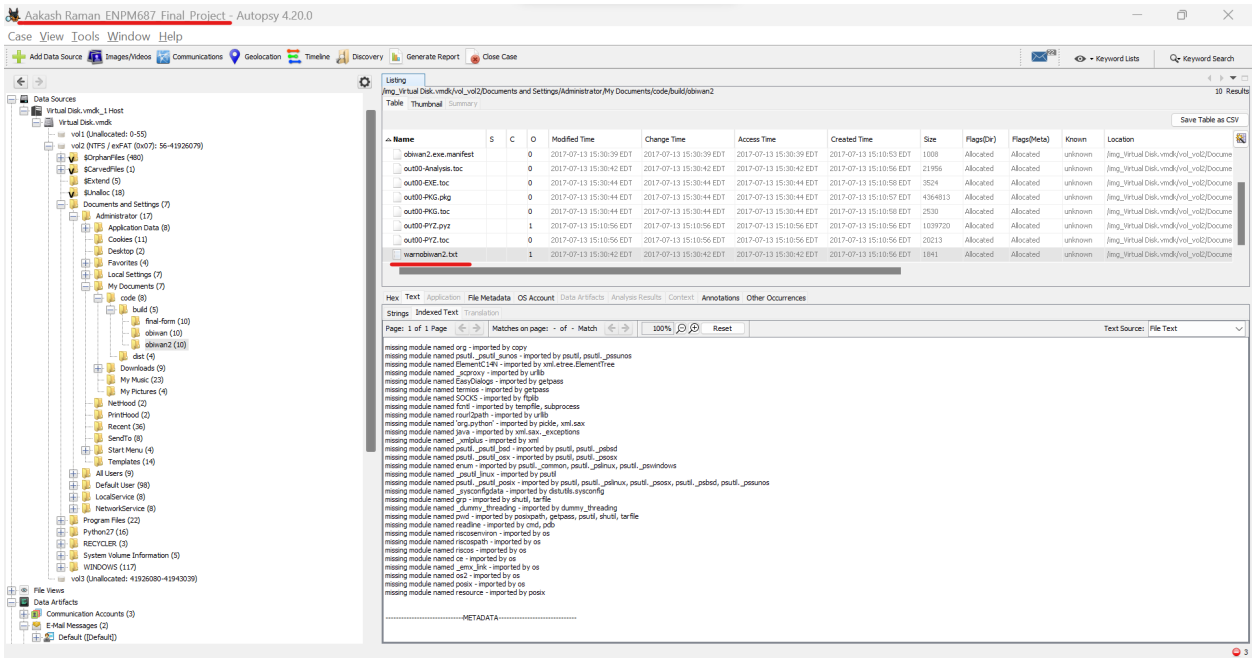




With this information, I wanted to check out the “obiwan.exe” and “obiwan2.exe” executable files so I went to the file path “***/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code***”, to analyze the Administrator folder. I found some very interesting folders as shown below, which indicate the “final form spec” of the malware, and special texts in the “obiwan” and “obiwan2” folders as shown below. The files “warnobiwan.txt” and “warnobiwan2.txt” inside the “obiwan” and “obiwan2” folders are very interesting. This is shown below.

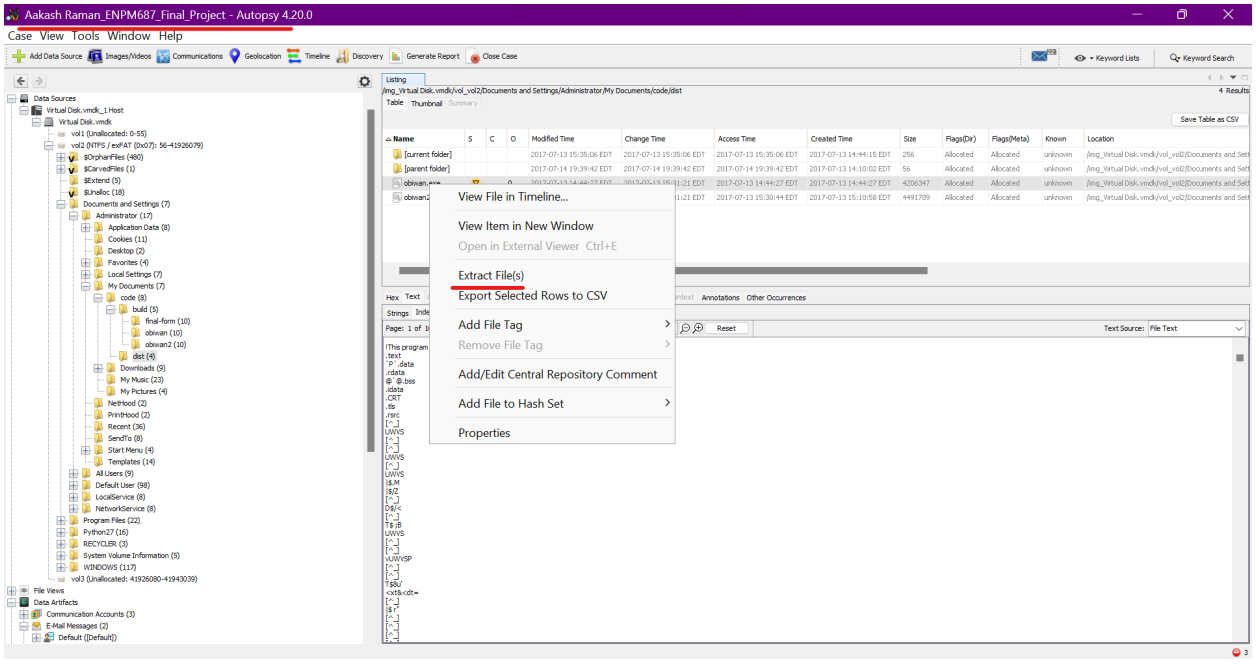

Aakash Raman_ENPM687_Final_Project - Autopsy 4.20.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
Keyword Lists
Keyword Search
Data Sources
Virtual Disk.vmdk_1 Host
Virtual Disk.vmdk
vol1 (Unallocated: 0-55)
vol2 (NTFS / exFAT (0x07): 56-41926079)
$OrphanFiles (480)
$CarvedFiles (1)
$Extend (5)
$Unalloc (18)
Documents and Settings (7)
Administrator (17)
Application Data (8)
Cookies (11)
Desktop (2)
Favorites (4)
Local Settings (7)
My Documents (7)
code (8)
build (5)
final-form (10)
obiwan (10)
obiwan2 (10)
dist (4)
Downloads (9)
My Music (23)
My Pictures (4)
NetHood (2)
PrintHood (2)
Recent (36)
SendTo (8)
Start Menu (4)
Templates (14)
All Users (9)
Default User (98)
LocalService (8)
NetworkService (8)
Program Files (22)
Python27 (16)
RECYCLER (3)
System Volume Information (5)
WINDOWS (117)
vol3 (Unallocated: 41926080-41943039)
File Views
Data Artifacts
Listing
/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code
8 Results
Table
Thumbnail
Summary
Save Table as CSV
Name
S
C
O
Modified Time
Change Time
Access Time
Created Time
Size
Flags(Dir)
Flags(Meta)
Known
Location
[current folder]
[parent folder]
build
dist
final-form.spec
obiwan.py
obiwan.spec
obiwan2.spec
Hex
Text
Application
File Metadata
OS Account
Data Artifacts
Analysis Results
Context
Annotations
Other Occurrences
Page: 1 of 1
Go to Page:
Jump to Offset
Launch in HxD

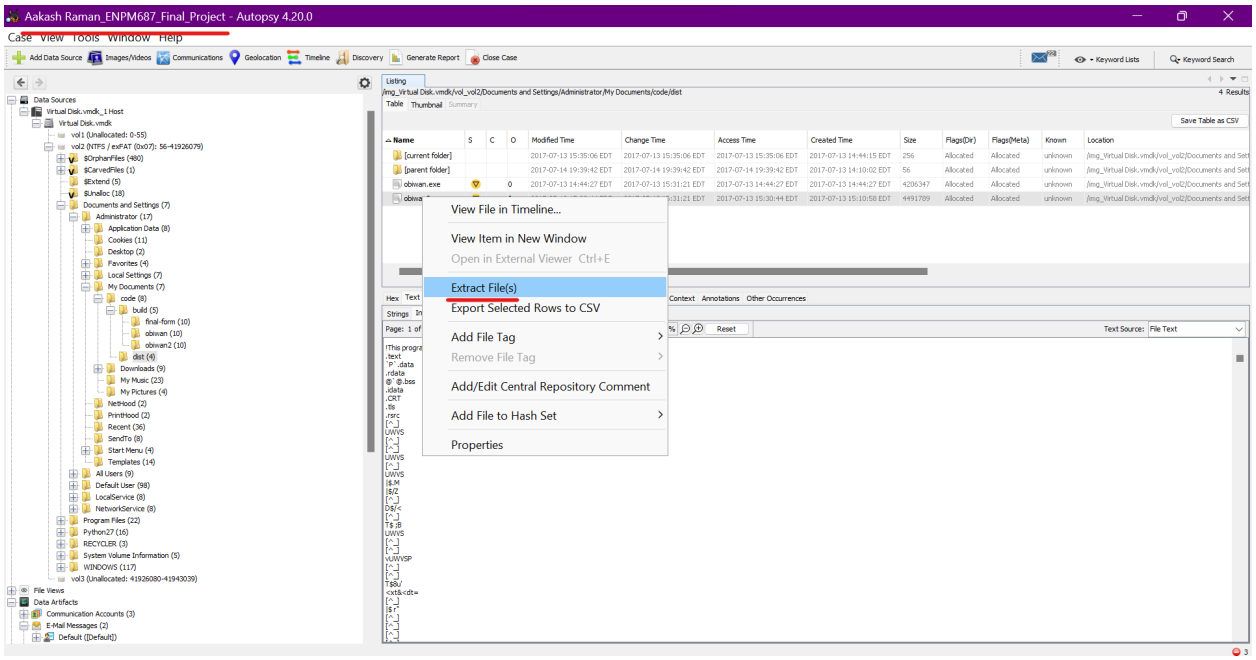

Aakash Raman_ENPM687_Final_Project - Autopsy 4.20.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Close Case
Keyword Lists
Keyword Search
Listing
/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code/build/obiwan
10 Results
Table
Thumbnail
Summary
Save Table as CSV
obiwan.exe.manifest
out00-Analysis.toc
out00-EXE.toc
out00-PKG.pkg
out00-PKG.toc
out00-PYZ.pyz
out00-PYZ.toc
warnobiwan.txt
Hex
Text
Application
File Metadata
OS Account
Data Artifacts
Analysis Results
Context
Annotations
Other Occurrences
Strings
Indexed Text
Translation
Page: 1 of 1 Page
Matches on page: - of - Match
100%
Reset
Text Source: File Text
missing module named readline - imported by cmd, pdb
missing module named org - imported by pickle
missing module named fcntl - imported by tempfile, subprocess
missing module named riscosenviron - imported by os
missing module named riscospath - imported by os
missing module named riscos - imported by os
missing module named ce - imported by os
missing module named _emx_link - imported by os
missing module named os2 - imported by os
missing module named pwd - imported by posixpath, getpass
missing module named posix - imported by os
missing module named resource - imported by posix
missing module named 'org.python' - imported by copy
missing module named SOCKS - imported by ftplib
missing module named _dummy_threading - imported by dummy_threading
missing module named _sgprovy - imported by urllib
missing module named EasyDialogs - imported by getpass
missing module named termios - imported by getpass
missing module named rourl2path - imported by urllib
----------METADATA----------

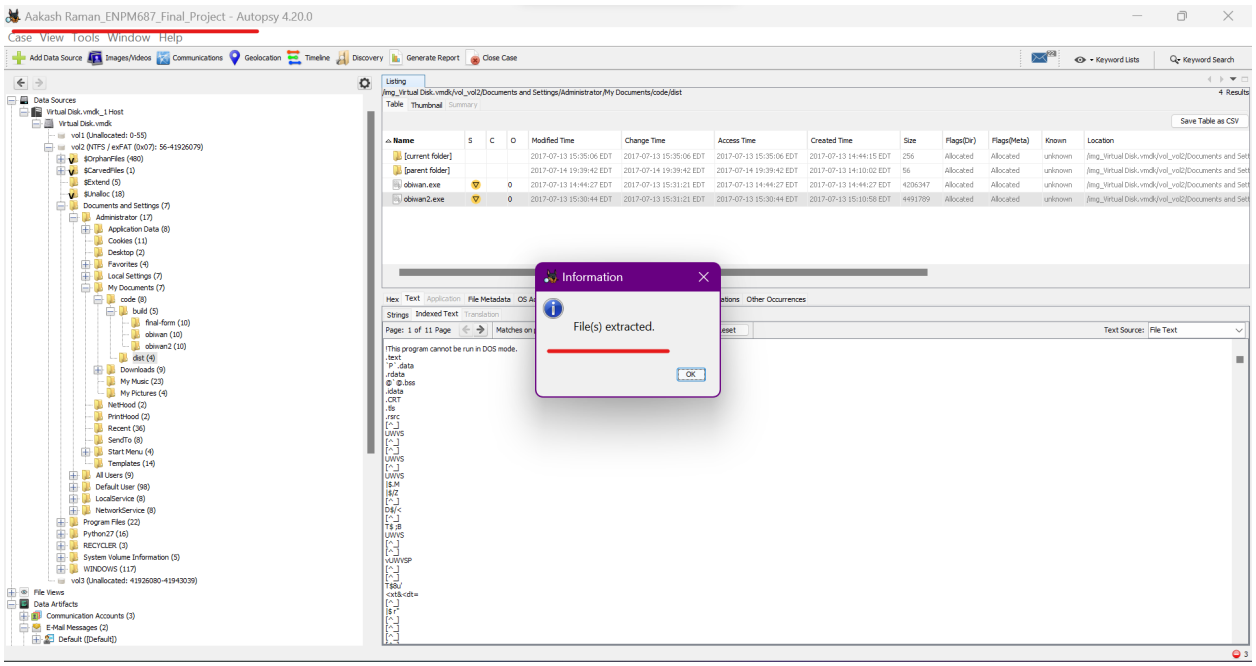


Extracting Obiwan.exe and Obiwan2.exe for further Forensic Analysis:

Maybe, these are the files that get generated once the executable is run. This piqued my curiosity to somehow download these executable files and capture the network traffic using a packet analysis tool like Wireshark. I know that Autopsy has the ability to extract files. I did exactly that by right-clicking and Extracting both “obiwan.exe” and “obiwan2.exe” as shown below.







Packet Analysis of each Executable File using Wireshark:

Now, I will run each of the executables and capture the packets using Wireshark. When I ran “obiwan.exe” and captured its packets using Wireshark, the output was similar to what I solved in Homework 6. The source IP Address of my computer, “*10.5.18.178*” reaches out to a different IP Address, “*18.154.185.116*”, not in the network, requesting some URLs. I filtered for only HTTP packets and realized that this executable reaches out to 2 URLs namely:

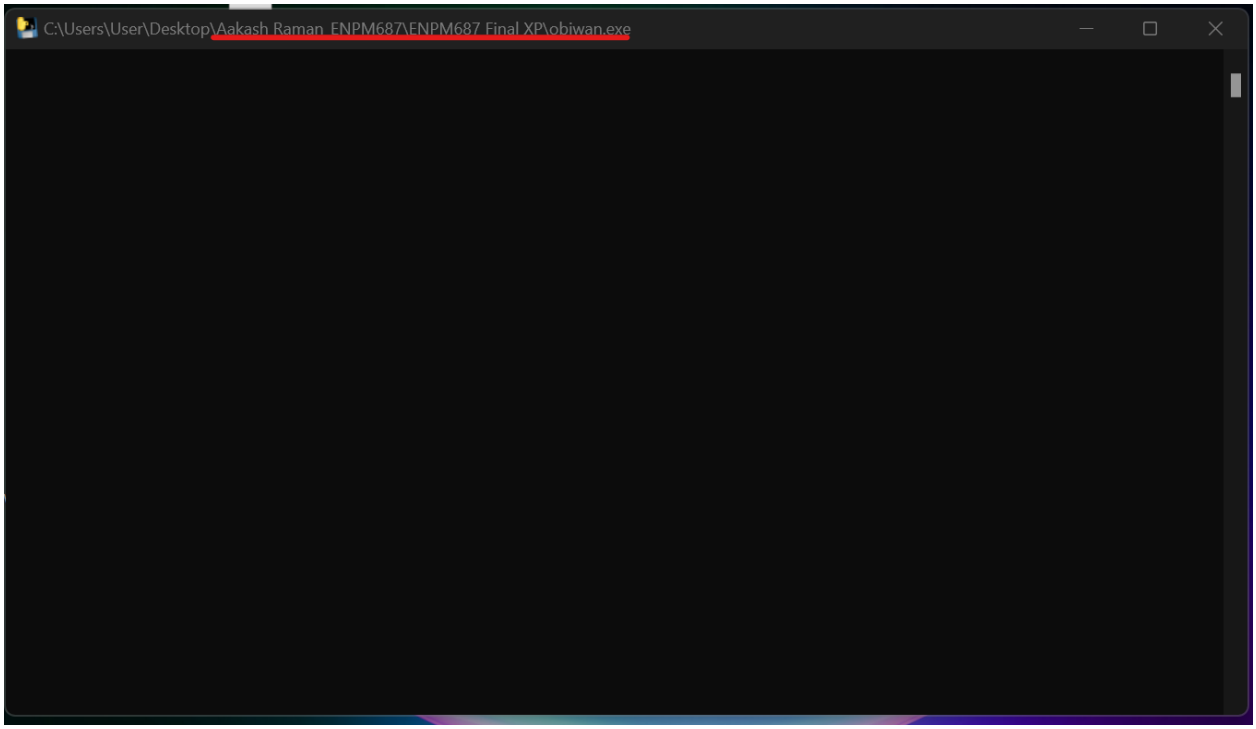
- http://www.umd.edu/help-me-obiwan-kenobi
- http://www.umd.edu/youre-my-only-hope

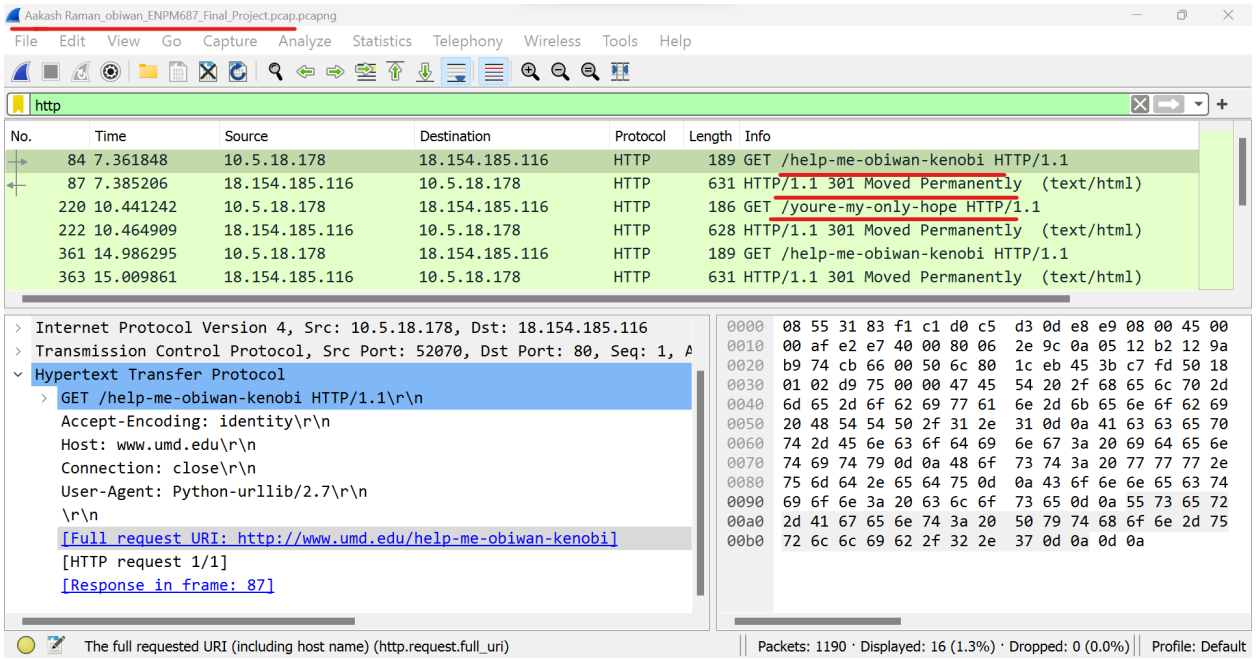
The decrypted messages from the “obiwan.exe” executable include:

- ***help-me-obiwan-kenobi***
- ***youre-my-only-hope***

The “obiwan.exe” executable only sends GET Requests between a short interval to these 2 URLs and gets a HTTP response of 301 indicating that the website has been moved permanently by the Web Server. The executable is also possibly written in Python 2.7 to query these 2 URLs because the User-Agent is a Python-urllib/2.7. Furthermore, the host it is querying is “www.umd.edu”.

Based on the directories requested these are clear messages by the Rebel’s to Obiwan Kenobi to help rescue them. These screenshots are shown below.





Now, let me run “obiwan2.exe” and see what it is trying to do. When I ran “obiwan2.exe” and captured its packets on Wireshark, I noticed some interesting findings. I again filtered for HTTP Packets thinking, that this executable would follow the same pattern as Obiwan.exe and my hunch was correct. My computer’s source IP Address, “*10.5.18.178*” reaches out to “*18.160.46.99*”, another IP Address not present in the network, and reaches out to 3 URLs:

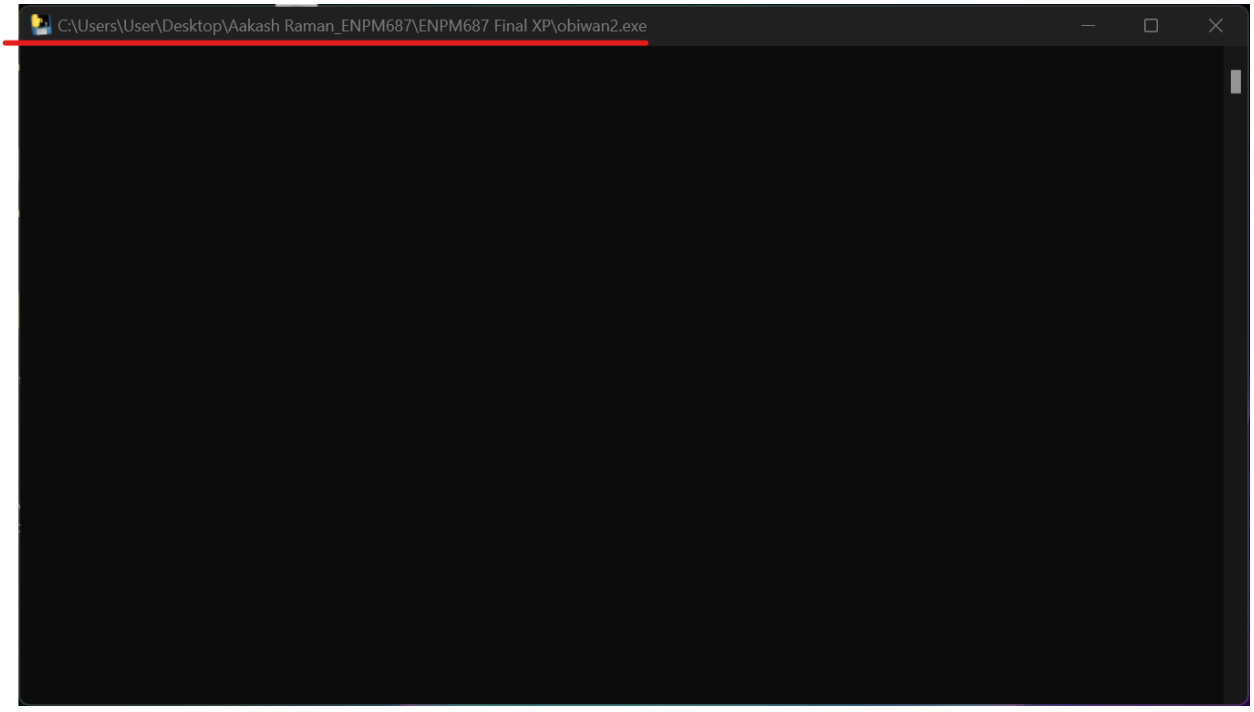
- http://www.umd.edu/this-is-not-even-my-final-form
- http://www.umd.edu/All-your-base64-are-belong-to-us
- http://www.umd.edu/cjJkMiBpcyB0aGUga2V5

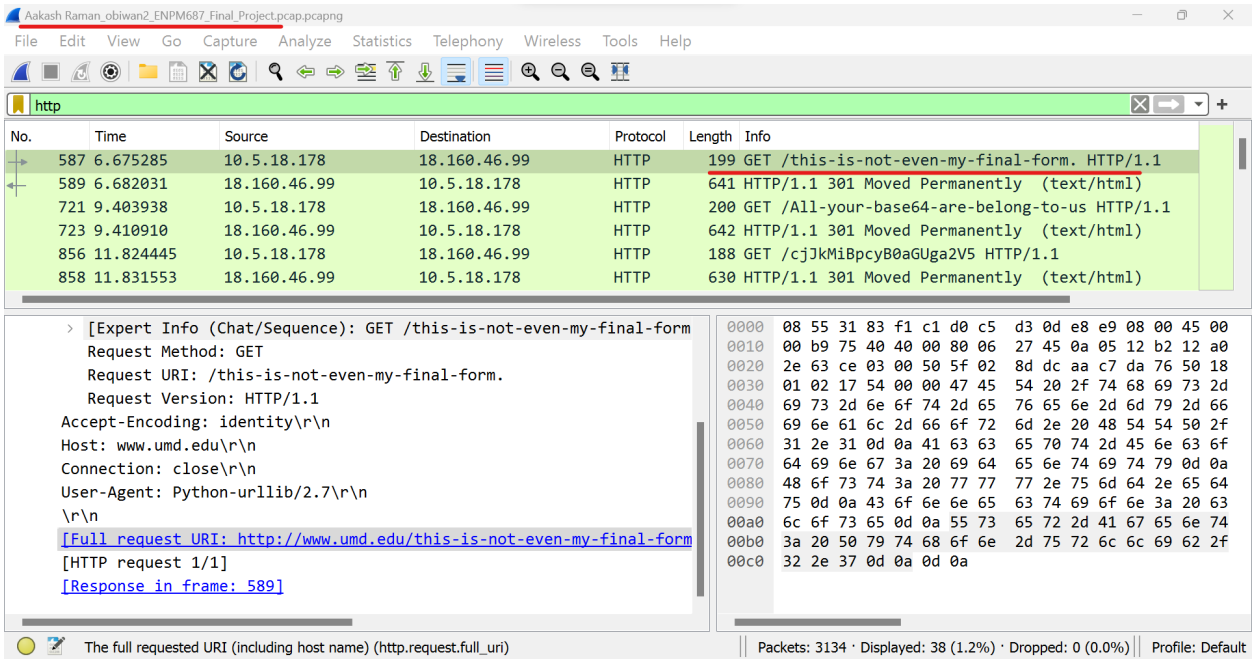
Similarly, the executable is also possibly written in Python 2.7 to query these 2 URLs because the User-Agent is a Python-urllib/2.7. Furthermore, the host it is querying is “www.umd.edu”.

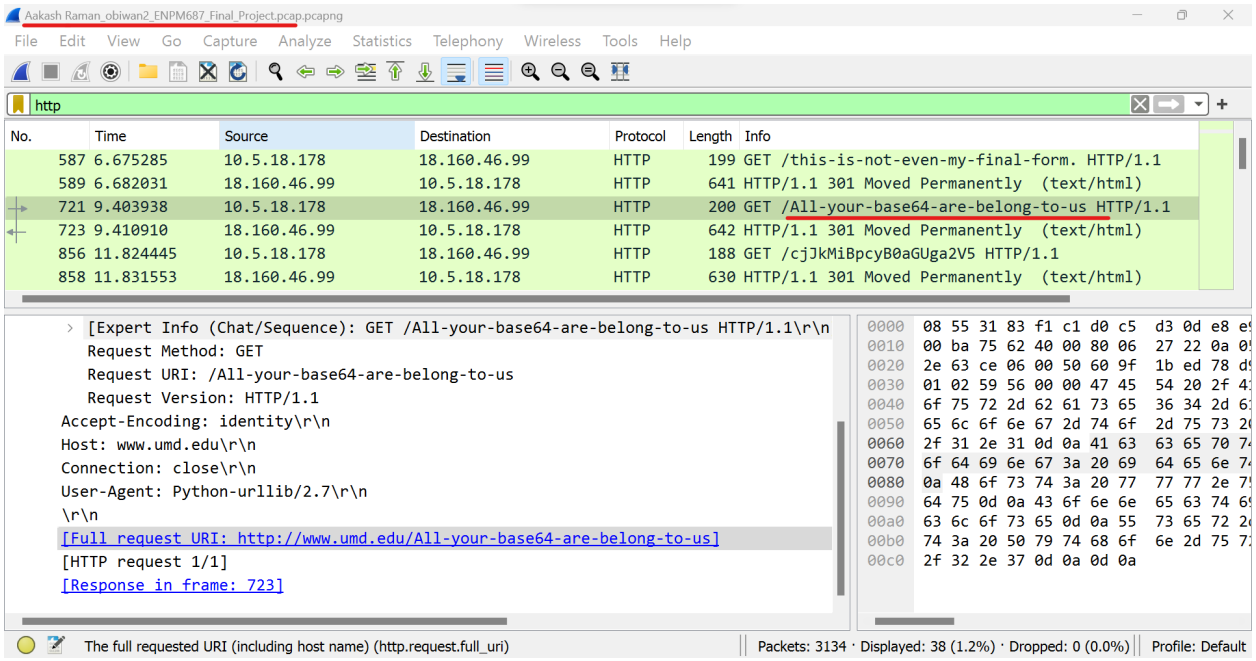
The decrypted messages from the “obiwan2.exe” executable include:

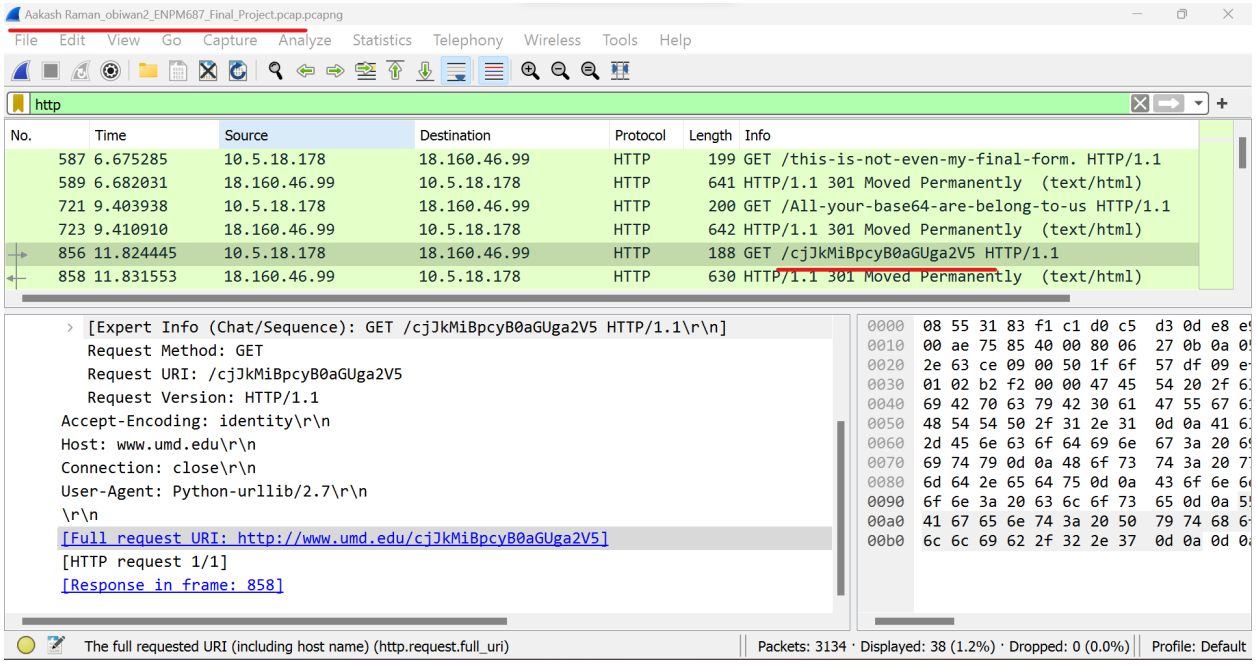
- ***this-is-not-even-my-final-form***
- ***All-your-base64-are-belong-to-us***
- ***cjJkMiBpcyB0aGUga2V5***

Similar to Obiwan.exe, Obiwan2.exe also sends GET Requests to these 3 URLs and gets a HTTP Response of 301, indicating that the website has been moved permanently by the Web Server. The time interval between the requests is also small just like in the case of “Obiwan.exe”. This executable was probably kept to decoy the Imperials into thinking its the final form of the Rebel Malware Writer’s Malware, but clearly, based on the messages, I have not found the final Malware. The screenshots are shown below.


C:\Users\User\Desktop\Aakash Raman_ENPM687\ENPM687 Final XP\obiwan2.exe


Aakash Raman_obiwan2_ENPM687_Final_Project.pcap.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
587 6.675285 10.5.18.178 18.160.46.99 HTTP 199 GET /this-is-not-even-my-final-form. HTTP/1.1
589 6.682031 18.160.46.99 10.5.18.178 HTTP 641 HTTP/1.1 301 Moved Permanently (text/html)
721 9.403938 10.5.18.178 18.160.46.99 HTTP 200 GET /All-your-base64-are-belong-to-us HTTP/1.1
723 9.410910 18.160.46.99 10.5.18.178 HTTP 642 HTTP/1.1 301 Moved Permanently (text/html)
856 11.824445 10.5.18.178 18.160.46.99 HTTP 188 GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1
858 11.831553 18.160.46.99 10.5.18.178 HTTP 630 HTTP/1.1 301 Moved Permanently (text/html)
[Expert Info (Chat/Sequence): GET /this-is-not-even-my-final-form
Request Method: GET
Request URI: /this-is-not-even-my-final-form.
Request Version: HTTP/1.1
Accept-Encoding: identity\r\n
Host: www.umd.edu\r\n
Connection: close\r\n
User-Agent: Python-urllib/2.7\r\n
\r\n
[Full request URI: http://www.umd.edu/this-is-not-even-my-final-form
[HTTP request 1/1]
[Response in frame: 589]
0000 08 55 31 83 f1 c1 d0 c5 d3 0d e8 e9 08 00 45 00
0010 00 b9 75 40 40 00 80 06 27 45 0a 05 12 b2 12 a0
0020 2e 63 ce 03 00 50 5f 02 8d dc aa c7 da 76 50 18
0030 01 02 17 54 00 00 47 45 54 20 2f 74 68 69 73 2d
0040 69 73 2d 6e 6f 74 2d 65 76 65 6e 2d 6d 79 2d 66
0050 69 6e 61 6c 2d 66 6f 72 6d 2e 20 48 54 54 50 2f
0060 31 2e 31 0d 0a 41 63 63 65 70 74 2d 45 6e 63 6f
0070 64 69 6e 67 3a 20 69 64 65 6e 74 69 74 79 0d 0a
0080 48 6f 73 74 3a 20 77 77 77 2e 75 6d 64 2e 65 64
0090 75 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 63
00a0 6c 6f 73 65 0d 0a 55 73 65 72 2d 41 67 65 6e 74
00b0 3a 20 50 79 74 68 6f 6e 2d 75 72 6c 6c 69 62 2f
00c0 32 2e 37 0d 0a 0d 0a
The full requested URI (including host name) (http.request.full_uri)
Packets: 3134 · Displayed: 38 (1.2%) · Dropped: 0 (0.0%)
Profile: Default

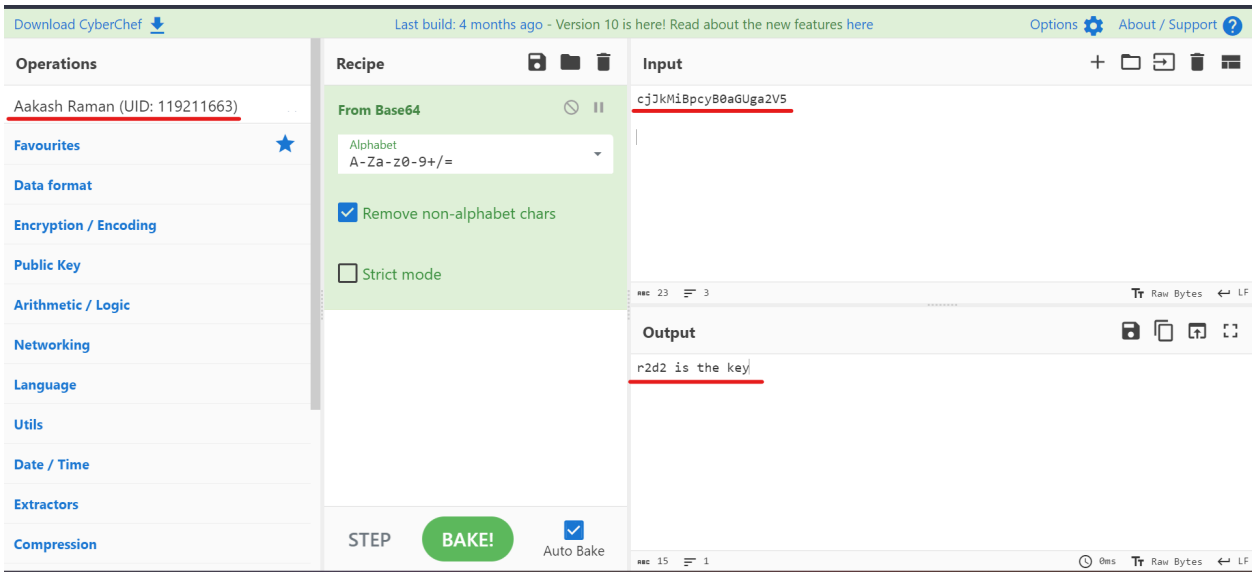




What struck me now is a couple of things. Firstly, this is **Not** the Final-Form of the malware written by the Rebel Malware Writer as mentioned by the 1st URL. Secondly, the 3rd URL that was beaconed to by my computer is not in ASCII/English. It is in Base64 and that is indicated by the 2nd URL ("All-your-base64-are-belong-to-us"). Hence, let me decode the 3rd URL from Base64 back to English/ASCII.

Converting Base64 URL back to ASCII/English:

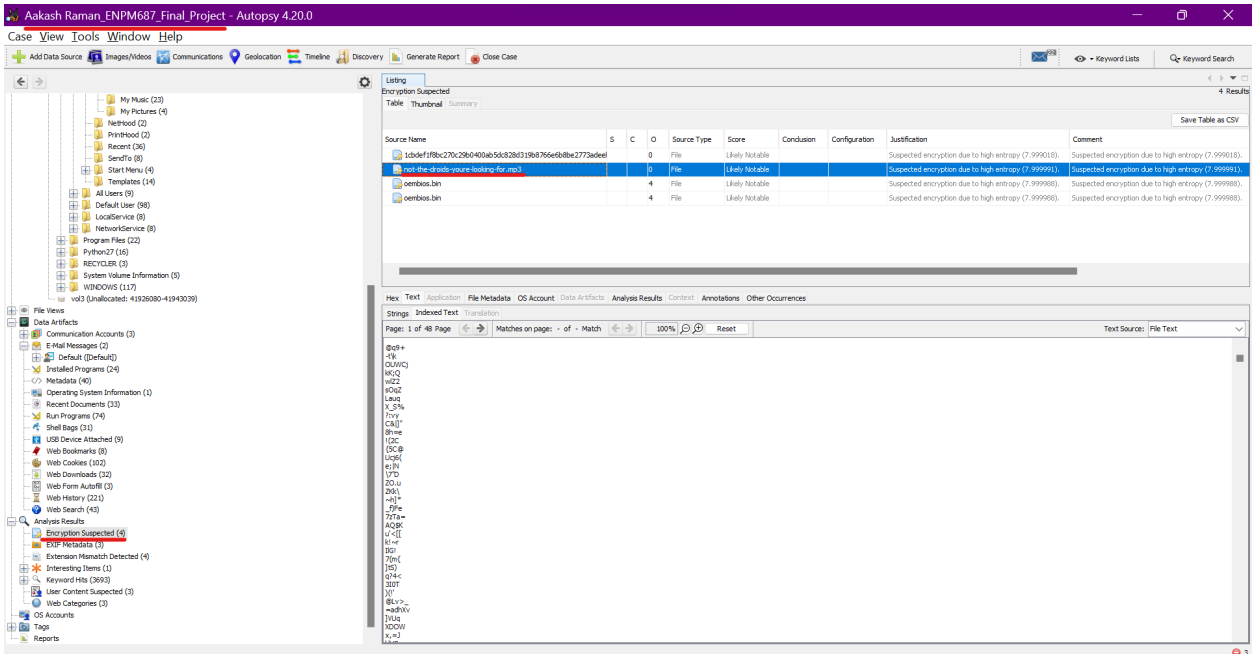
Using CyberChef [2], a popular platform for converting between encoding formats, I get the Base64 decoded version of "***cjJkMiBpcyB0aGUga2V5***" as "***r2d2 is the key***". This is shown below.



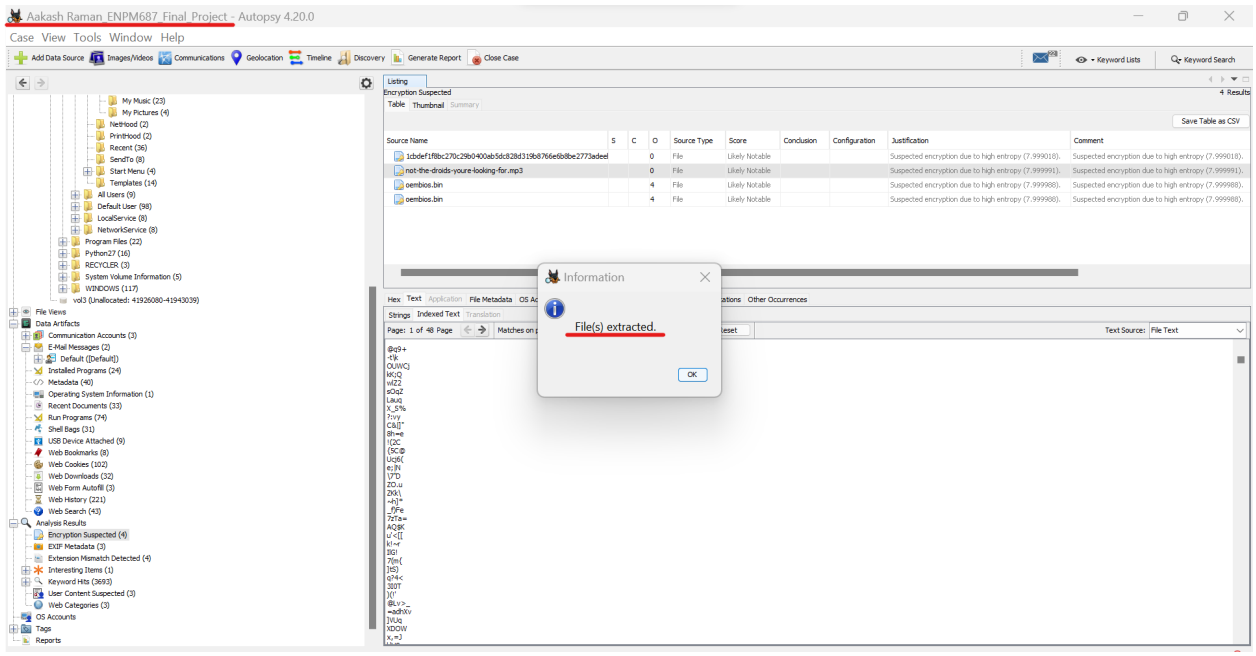
Great, I know that "***r2d2***" is the key to something, I know I have to find out what that is. From my knowledge of the course, Keys are normally used when something is encrypted, maybe this is the decryption key for something that is encrypted. Hence, I decided to find artifacts that are encrypted and try the key "r2d2" to decrypt them, this will lead me in the right direction.

Finding the Encrypted File that might have r2d2 as its Decryption Key:

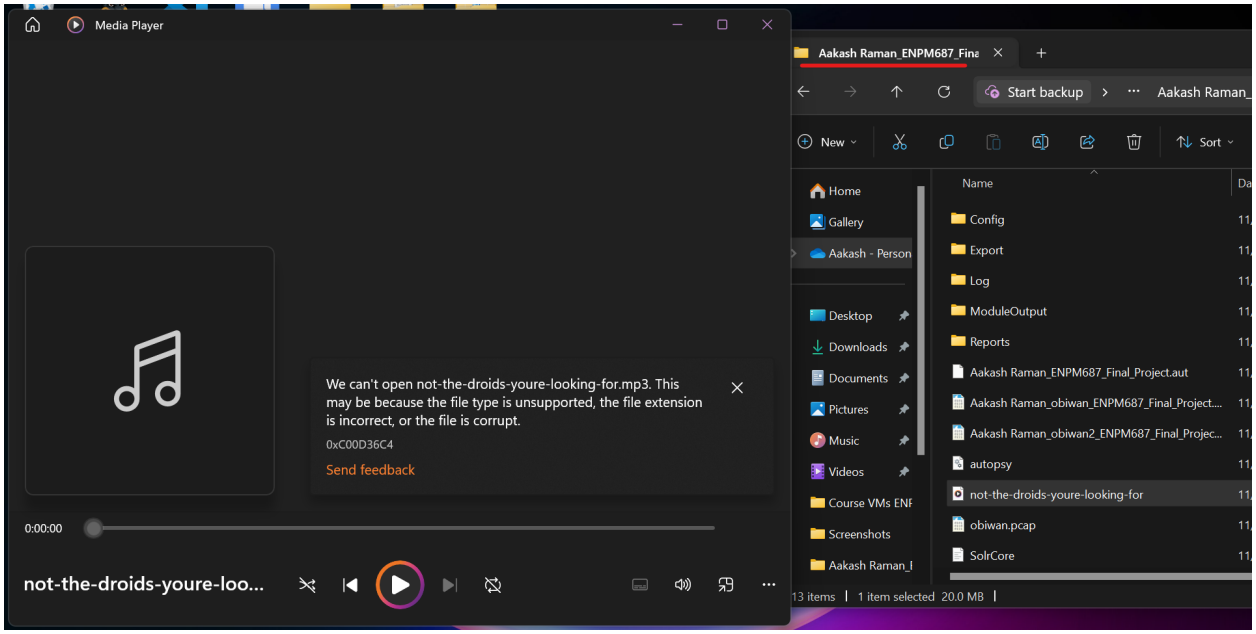
Going back to Autopsy, I know that Autopsy has a section called "**Encryption Suspected**" which lists all the files that Autopsy predicts with a high probability are encrypted due to high entropy within the files. Hence, I wanted to check out this section and when I opened it, my initial hunch was correct. There were only 4 files and there was a unique and interesting MP3 file named "***not-the-droids-youre-looking-for.mp3***" which has some sort of relation with "r2d2", as it is also a Droid. Hence, I wanted to extract this file and figure out the software that was used to encrypt this file. This is shown below.



Hence, I then extracted this file to determine the encryption software that encrypted this file as shown below.



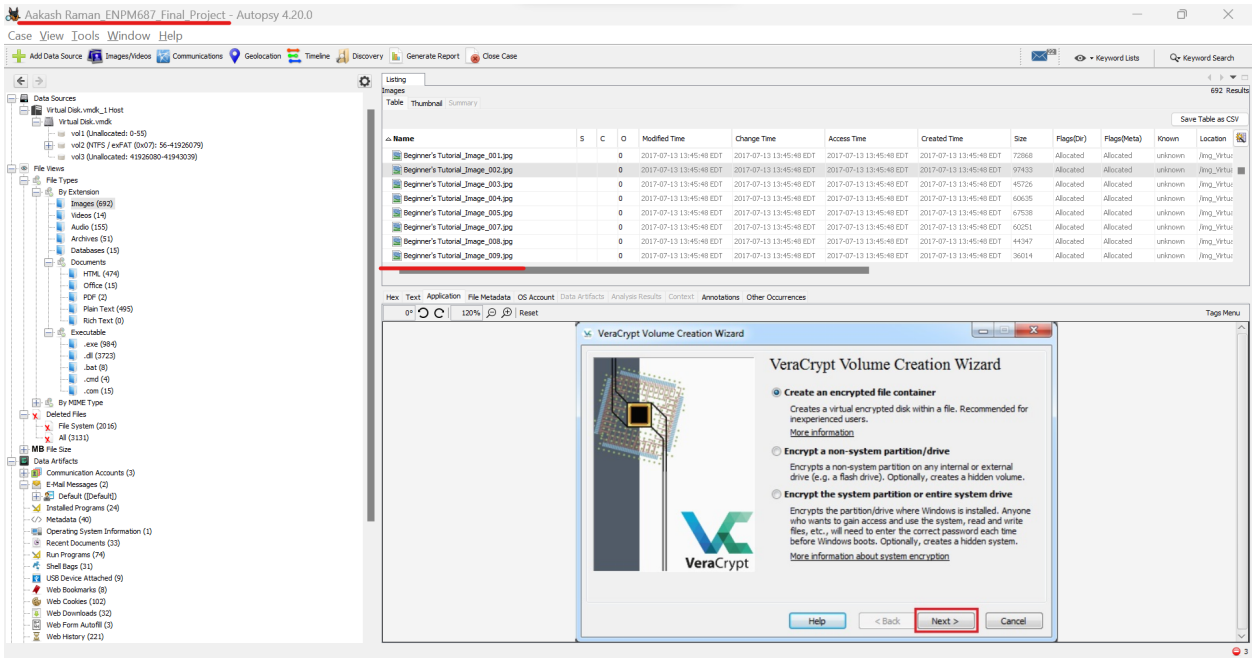
To test my suspicion if the file was encrypted and was not an “MP3” file, after extracting the file I tried opening it in Windows Media Player and got a notification from Windows Media Player that the file cannot be played because it is not an MP3 file or the file is corrupt as shown below.

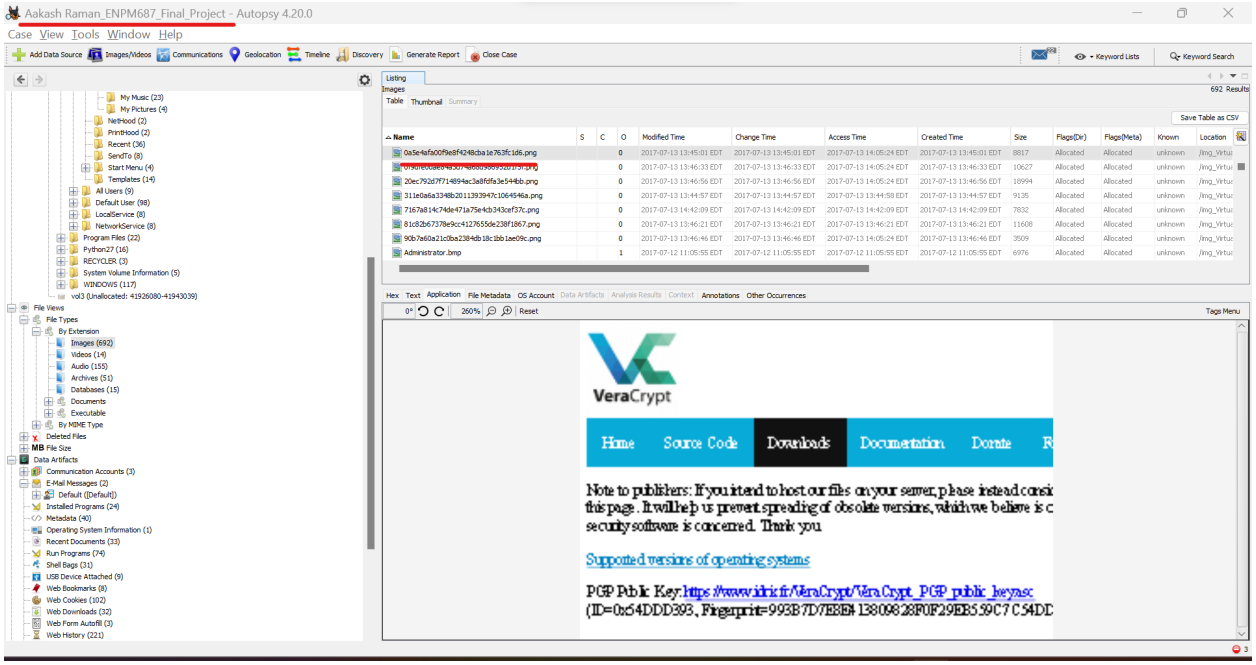


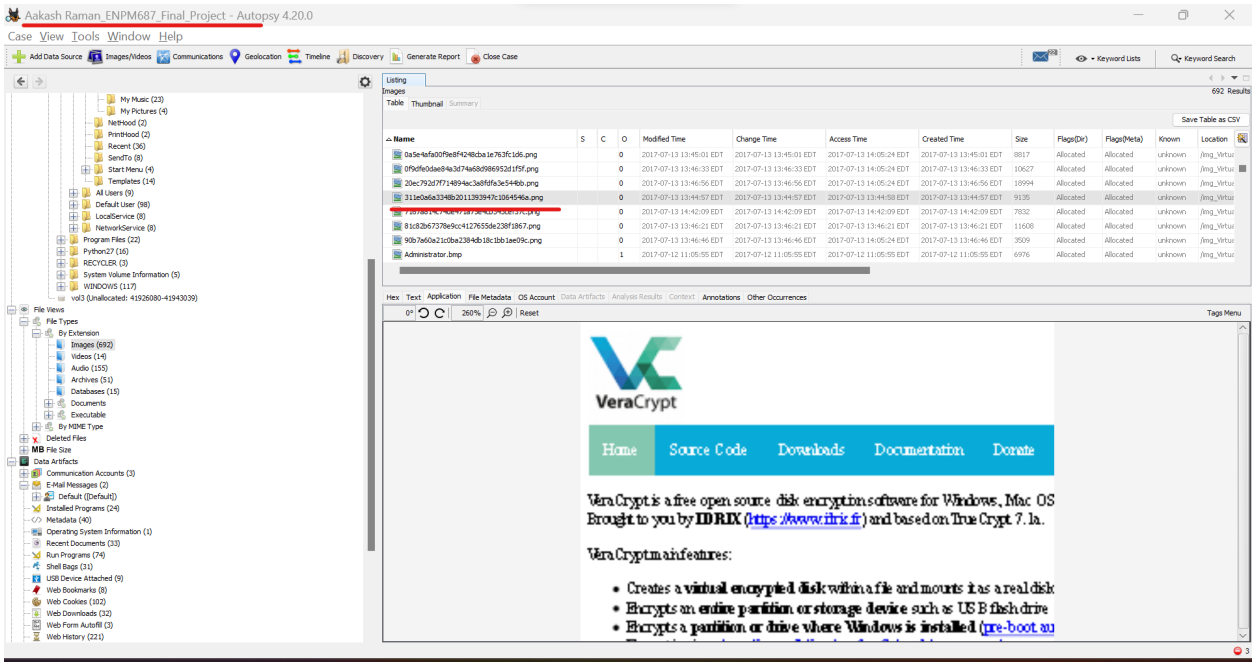
Determining the Encryption Software and Decrypting the not-the-droids-youre-looking-for.mp3 file using the r2d2 Key:

Based on the information I gathered so far, I have seen multiple instances of disk/file encryption and decryption software, VeraCrypt. Moreover, other than the occurrences of VeraCrypt I mentioned above, I saw its clear usage in the form of Image files. When I analyzed the ***Images*** folder under the "File Types" section, I found many references to VeraCrypt. Some images like "***Beginner's Tutorial_Image_001.jpg***" to "***Beginner's Tutorial_Image_022.jpg***" and "***Beginner's Tutorial_Image_034.jpg"*** depict the steps to encrypt a disk/file using VeraCrypt. This is the file that is asked by VeraCrypt to view if a user has never used it in the past. The presence of these files means that the Rebel Malware Writer had never used VeraCrypt before and had kept photos of the tutorial for reference on his device.

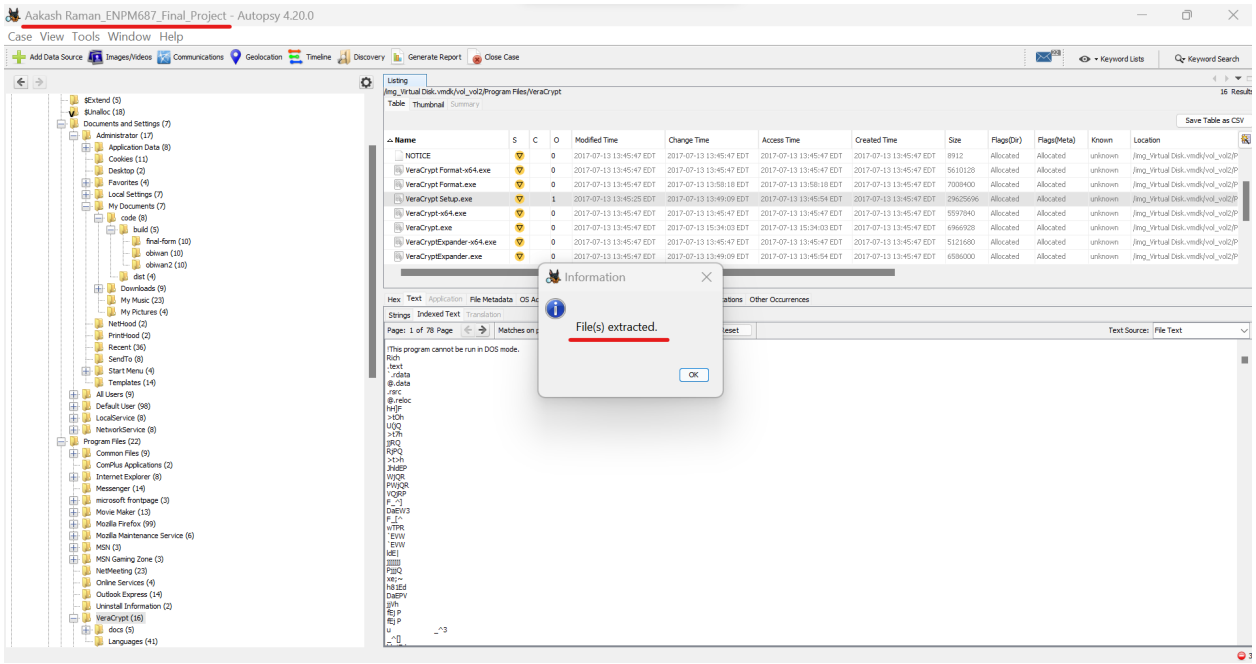
Other files such as, "***0a5e4afa00f9e8f4248cba1e763fc1d6.png***" and "***311e0a6a3348b2011393947c1064546a.png***" are blurred images but they show the VeraCrypt software being downloaded. The other files I found above also make me believe that VeraCrypt is the encryption/decryption software used by the Rebel Malware Writer. Some of these screenshots are shown below.


Aakash Raman_ENPM687_Final_Project - Autopsy 4.20.0
Case View Tools Window Help
VeraCrypt Volume Creation Wizard
Create an encrypted file container
Encrypt a non-system partition/drive
Encrypt the system partition or entire system drive
Help
< Back
Next >
Cancel

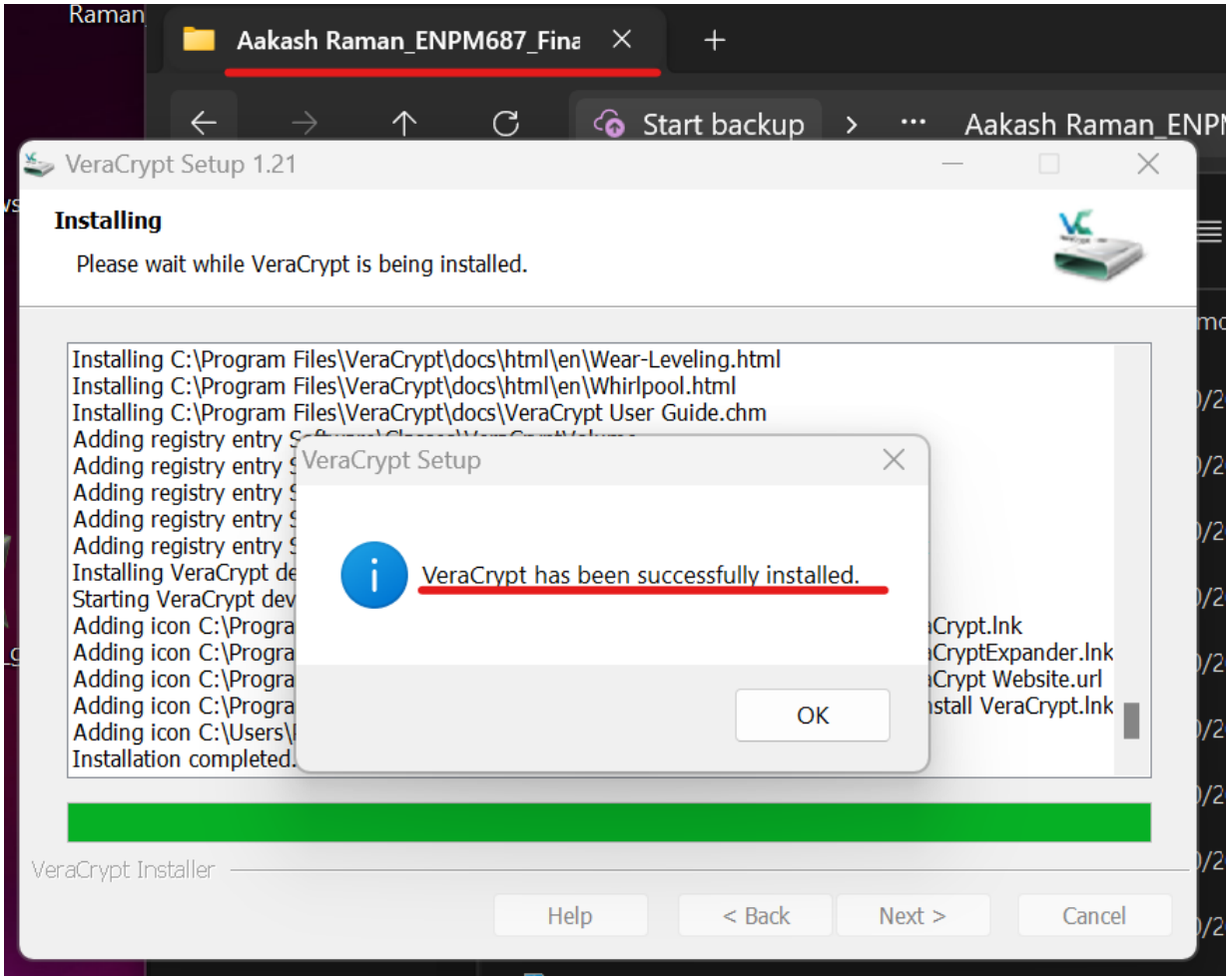

Aakash Raman_ENPM687_Final_Project - Autopsy 4.20.0
Case View Tools Window Help
VeraCrypt
Home
Source Code
Downloads
Documentation
Donate
Supported versions of operating systems



As I know where the VeraCrypt setup software is let me extract it and then try to decrypt the "***not-the-droids-youre-looking-for.mp3***" file. This is shown below.



Now, I will run the setup file to install and open VeraCrypt on my host system. This is shown below.

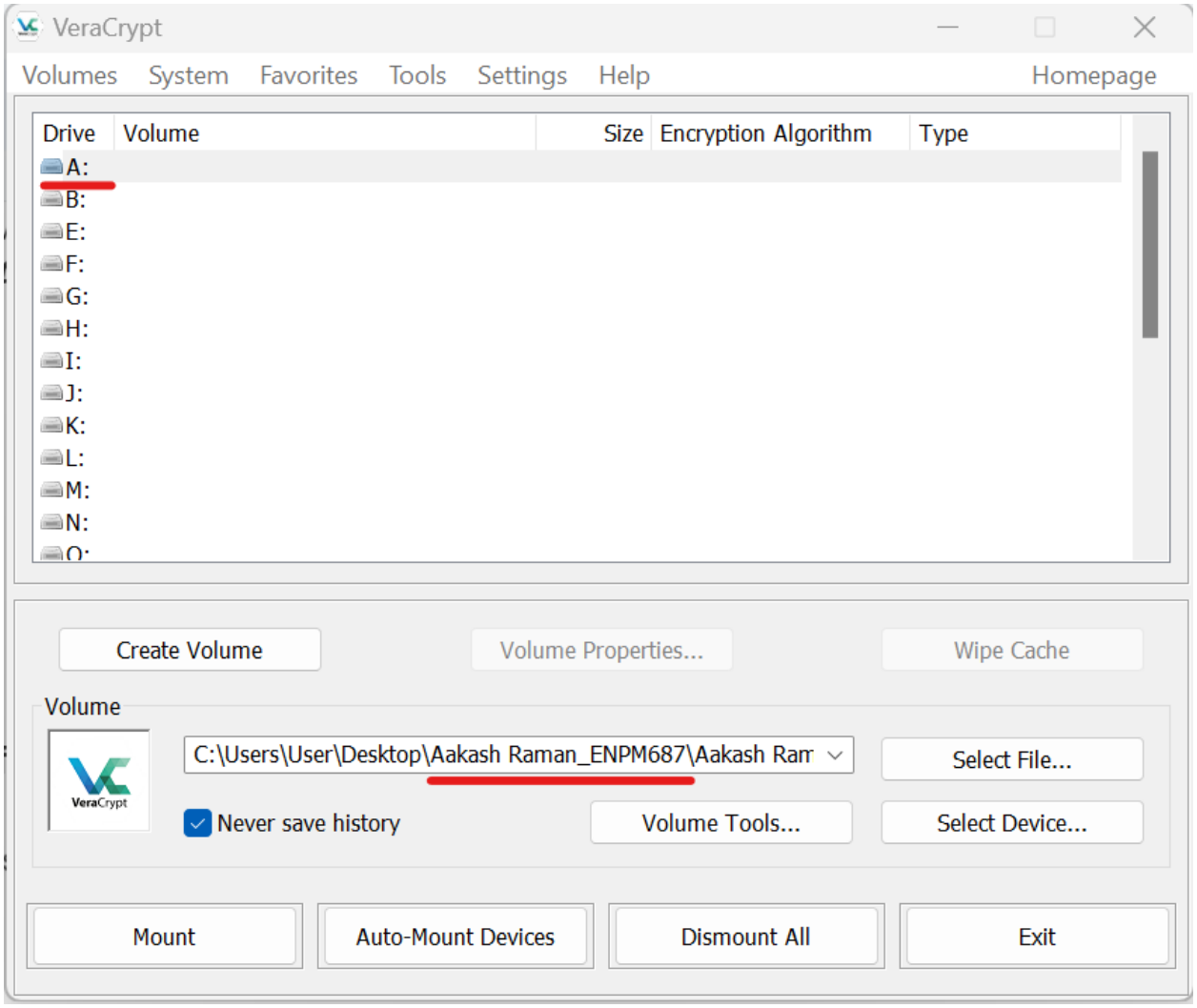


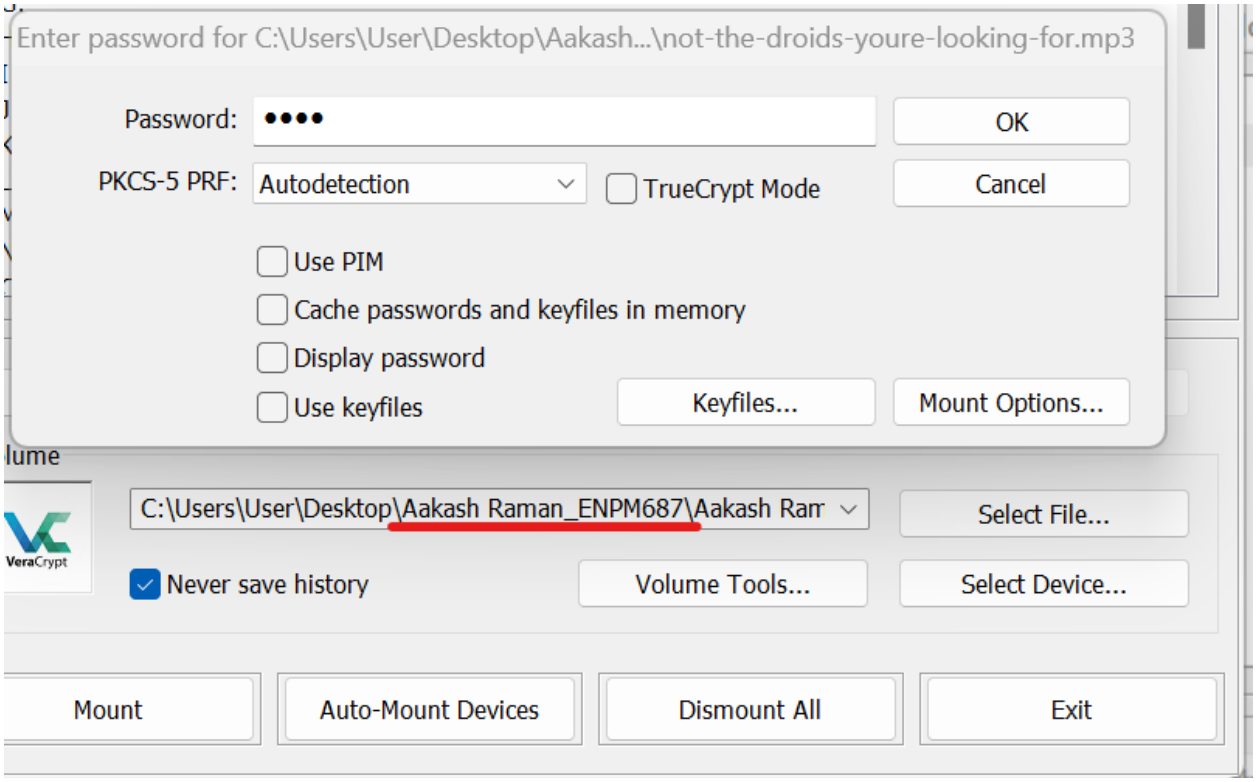
Decryption using VeraCrypt:

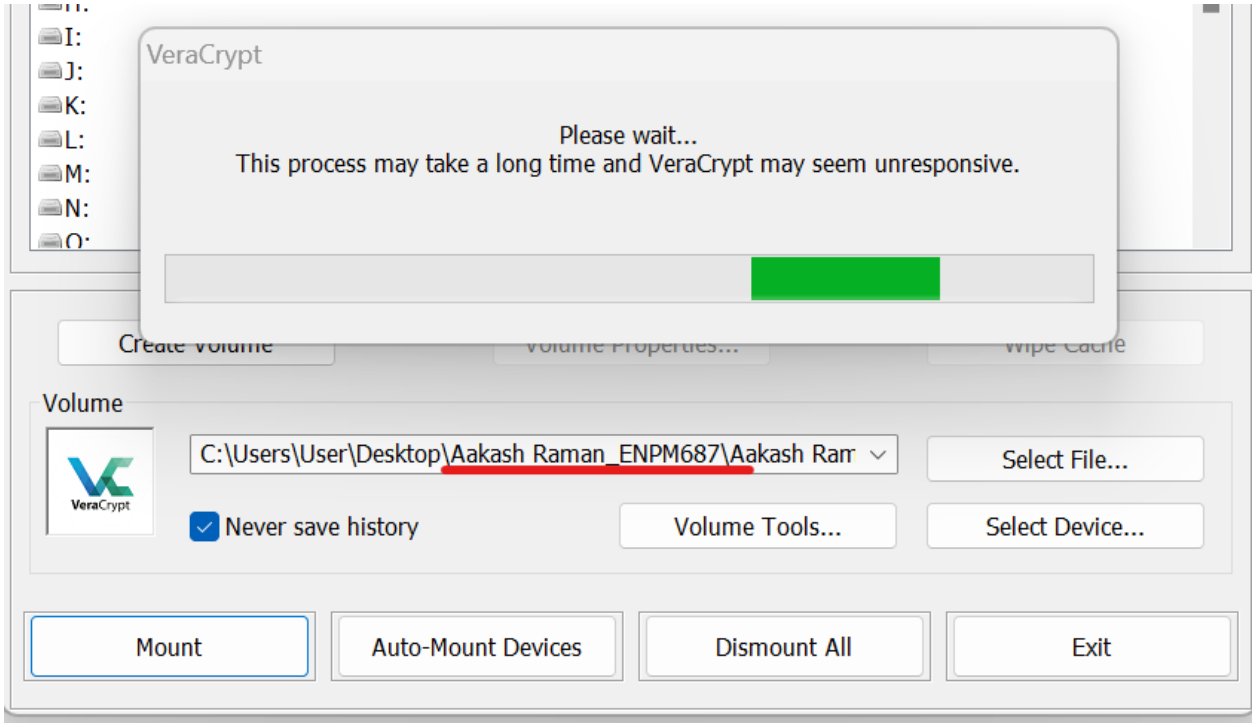
Now that VeraCrypt is installed, I will attempt to decrypt the "***not-the-droids-youre-looking-for.mp3***" file using the "***r2d2***" key. This is shown below.

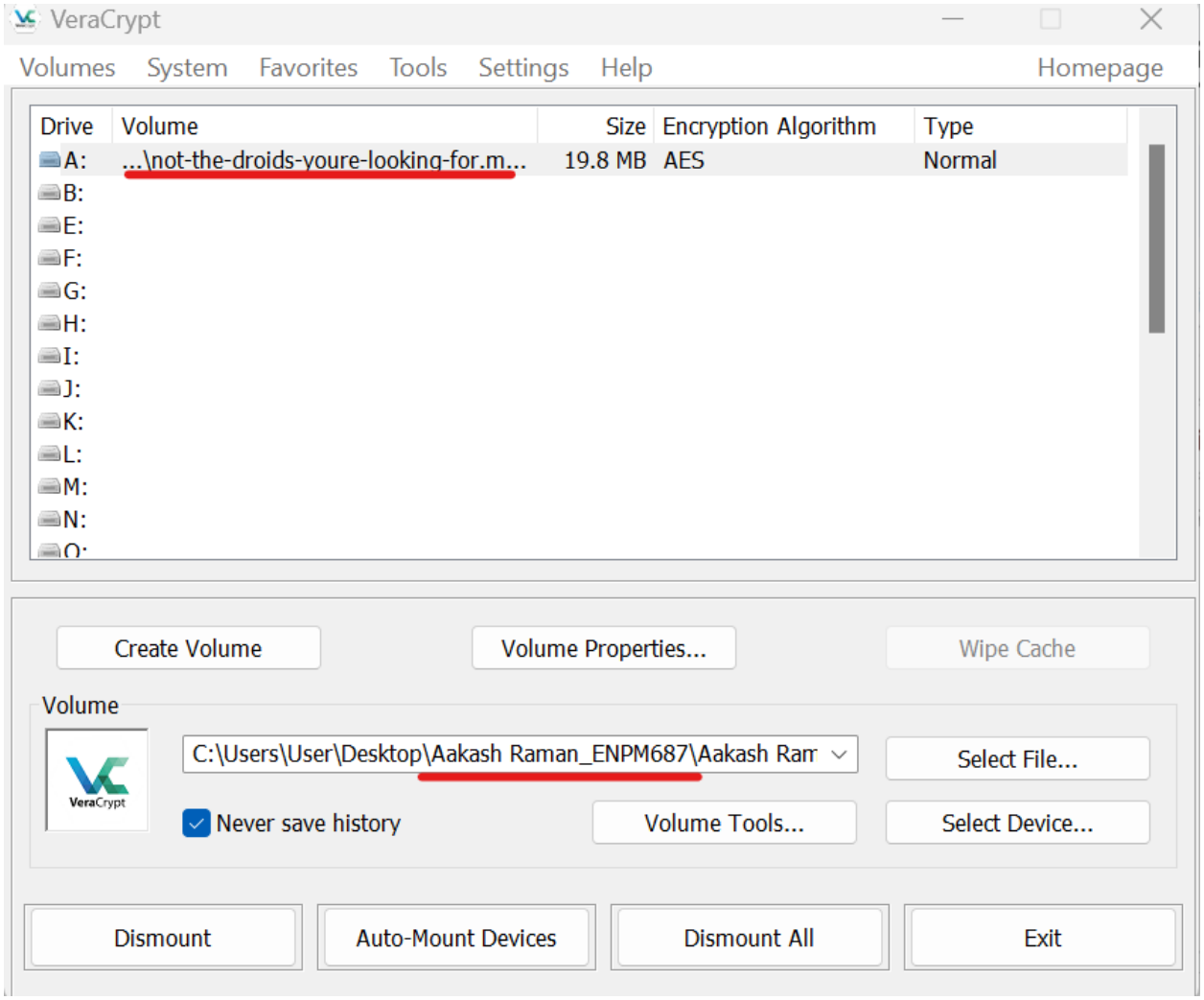
First I chose one of the free drives on my device to mount the data onto, I chose the A: drive. Then I selected the "***not-the-droids-youre-looking-for.mp3***" file and attempted to mount it. Then, VeraCrypt asked me for the password for decrypting the file for which I entered, "**r2d2**" and then VeraCrypt tried to decrypt the file.

As the decryption key was correct, it populated my system's A: Drive with the actual contents present in the "***not-the-droids-youre-looking-for.mp3***" file. All the steps I described are shown in the below screenshots.


VeraCrypt
Volumes
System
Favorites
Tools
Settings
Help
Homepage
Drive
Volume
Size
Encryption Algorithm
Type
A:
B:
E:
F:
G:
H:
I:
J:
K:
L:
M:
N:
O:
Create Volume
Volume Properties...
Wipe Cache
Volume
C:\Users\User\Desktop\Aakash Raman_ENPM687\Aakash Ram
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit


Enter password for C:\Users\User\Desktop\Aakash...\not-the-droids-youre-looking-for.mp3
Password:
OK
PKCS-5 PRF:
Autodetection
TrueCrypt Mode
Cancel
Use PIM
Cache passwords and keyfiles in memory
Display password
Use keyfiles
Keyfiles...
Mount Options...
lume
C:\Users\User\Desktop\Aakash Raman_ENPM687\Aakash Ram
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

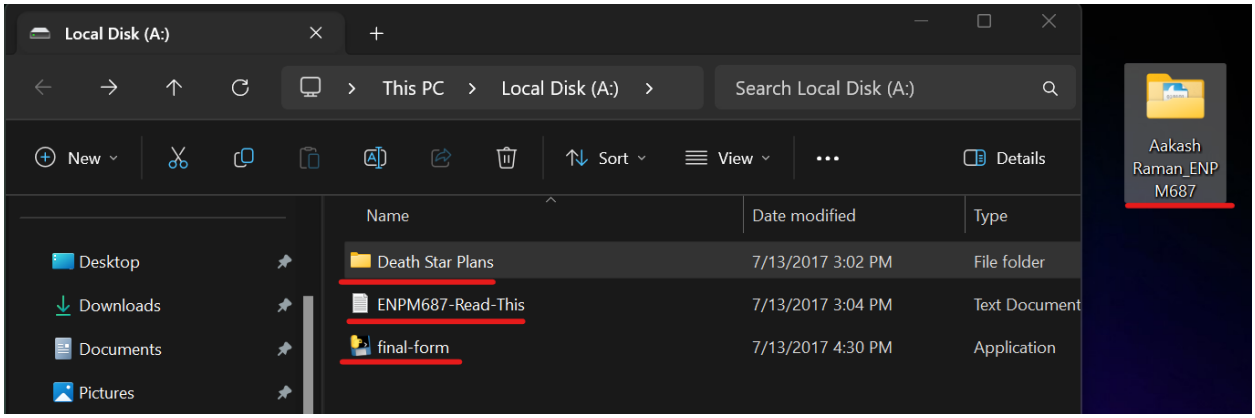

I:
J:
K:
L:
M:
N:
VeraCrypt
Please wait...
This process may take a long time and VeraCrypt may seem unresponsive.
Volume
C:\Users\User\Desktop\Aakash Raman_ENPM687\Aakash Ram
Select File...
Never save history
Volume Tools...
Select Device...
Mount
Auto-Mount Devices
Dismount All
Exit

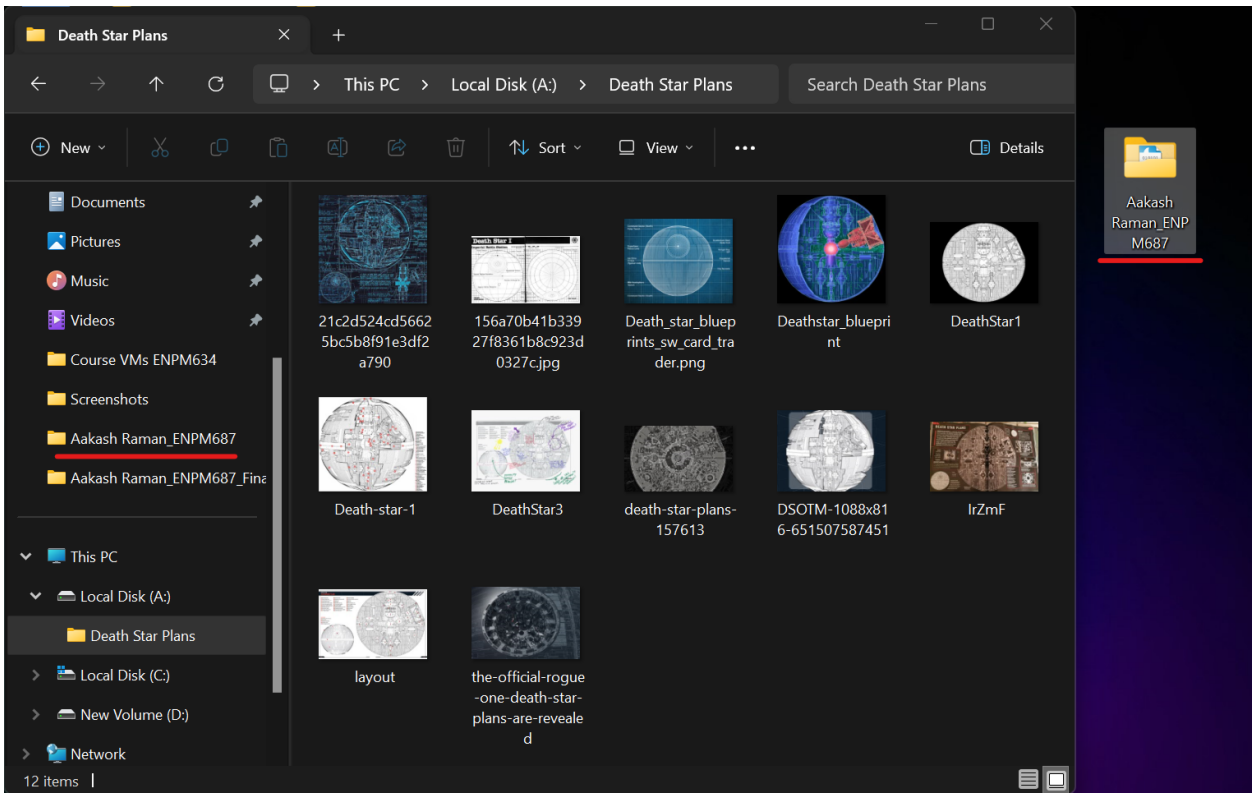


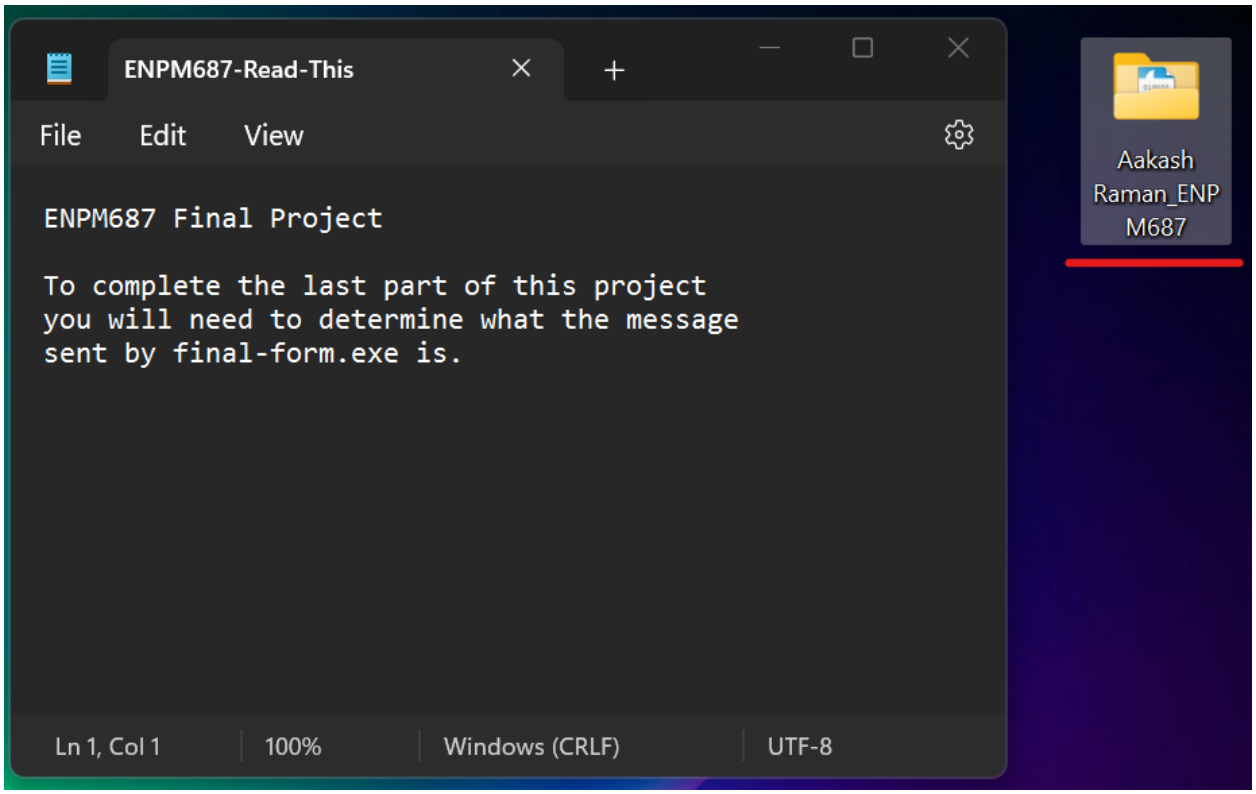
Examining the True Contents of the not-the-droids-youre-looking-for.mp3 file in the A: Drive:

When I went into the A: Drive of my system, a lot of interesting things were found. A folder labeled "***Death Star Plans***", a text file called "***ENPM687-Read This***" and an executable, "***final-form.exe***", which is the real malware written by the Rebel Malware Writer were found. Let me examine the folder and text file first. It looks like the Death Star Plans consist of 12 plans which are all the plans the Imperials have for the Death Star. These are confidential and it looks like the Rebels have got hold of them somehow. These screenshots are shown below.

By looking at the text file, I know that I need to run the "***final-form.exe***" and analyze its packet capture using Wireshark to determine the message it is sending. This is the message the Malware Rebel Writer wants to send to the Rebels. Hence, I will do this last part next.


Local Disk (A:)
This PC
Local Disk (A:)
Search Local Disk (A:)
New
Sort
View
Details
Name
Date modified
Type
Desktop
Downloads
Documents
Pictures
Death Star Plans
7/13/2017 3:02 PM
File folder
ENPM687-Read-This
7/13/2017 3:04 PM
Text Document
final-form
7/13/2017 4:30 PM
Application
Aakash Raman_ENPM687


Death Star Plans
This PC
Local Disk (A:)
Death Star Plans
Search Death Star Plans
New
Sort
View
Details
Documents
Pictures
Music
Videos
Course VMs ENPM634
Screenshots
Aakash Raman_ENPM687
Aakash Raman_ENPM687_Fina
This PC
Local Disk (A:)
Death Star Plans
Local Disk (C:)
New Volume (D:)
Network
12 items
21c2d524cd56625bc5b8f91e3df2a790
156a70b41b33927f8361b8c923d0327c.jpg
Death_star_blueprints_sw_card_trader.png
Deathstar_blueprint
DeathStar1
Death-star-1
DeathStar3
death-star-plans-157613
DSOTM-1088x816-651507587451
IrZmF
layout
the-official-rogue-one-death-star-plans-are-revealed
Aakash Raman_ENPM687



Analyzing the Packet Capture while running Final-Form.exe using Wireshark

I ran the "***final-form.exe***" executable and then captured its packets using Wireshark. From past experience, with the "obiwan.exe" and "obiwan2.exe" executables, I know that if I filter HTTP traffic, I can get the message the executable was trying to send. So, I will do that as shown below. My source IP Address "*10.5.18.178*" is sending 2 unique GET Requests to an IP Address "*18.160.46.81*", which is not part of the network. Again, it receives the HTTP 301 Moved Permanently response from the Web Server.

After capturing the traffic, the executable sends GET Requests to 2 unique URLs namely:

- ***http://www.umd.edu/We-have-the-blue-prints-to-the-Death-Star***
- ***http://www.umd.edu/We-will-defeat-Darth-Vader*.**

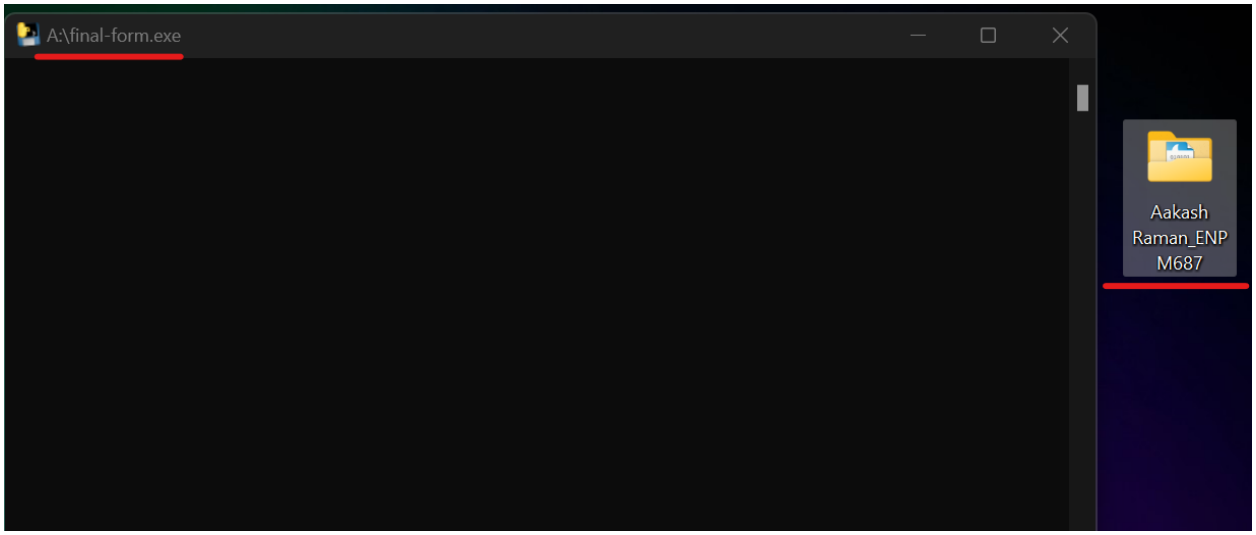
Hence, the final decrypted messages sent by the "final-form.exe" executable are:

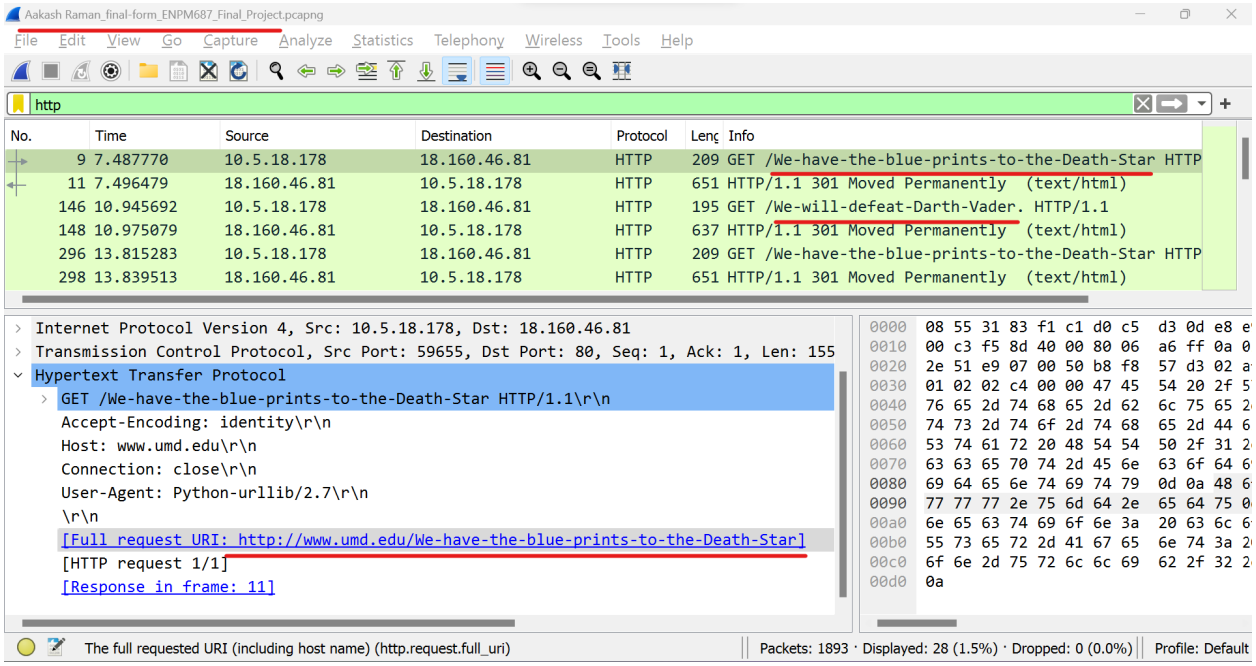
- ***We-have-the-blue-prints-to-the-Death-Star***
- ***We-will-defeat-Darth-Vader.***

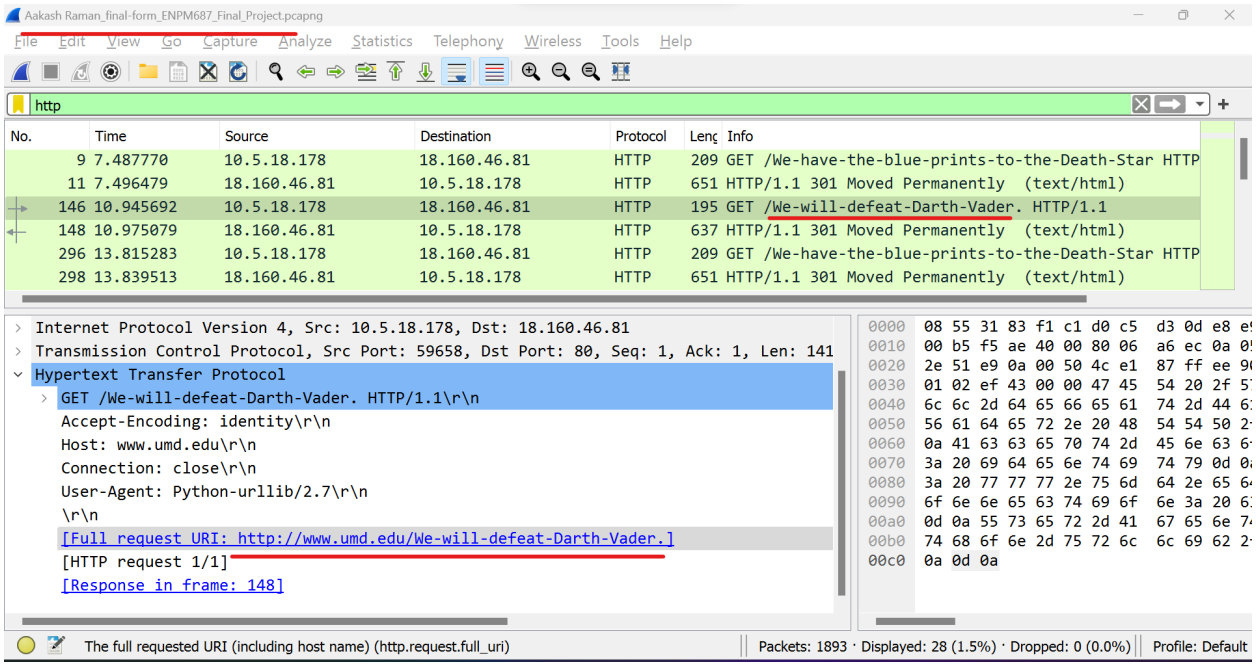
Similarly, the executable is also possibly written in Python 2.7 to query these 2 URLs because the User-Agent is a Python-urllib/2.7. Furthermore, the host it is querying is "www.umd.edu".

Overall, I believe that the Rebel Malware Writer wants to inform the Rebels that as the Rebels got the blueprints/plans of the Death Star from the previous evidence I found, they will defeat Darth Vader, the strongest Imperial Dark Jedi across the galaxy!

Given below are the screenshots depicting this.







c. Repetition of the above steps

As only 1 disk image of the Rebel Malware Writer's Hard Drive was provided, the above steps are the only steps to follow. However, if more disk images were provided then a similar approach would be followed to uncover the plans of the Rebels.

Recommendations & Future Plans

Overall, although the Rebel Malware Writer did a good job of concealing his plans, it was possible to get all the Rebel plans through the approach I followed. However, the same process could have been tougher if the Rebels followed the below suggestions:

- Deleted Files always leave some traces behind, hence, proper care must be taken to ensure that a file is permanently deleted. If initially, I had not found traces of VeraCrypt in the Deleted Files section of Autopsy, I would have not known that the Rebel Malware Writer used VeraCrypt.

- Always use HTTPS while communicating and share the SSL/TLS decryption key to decrypt the traffic with the Rebels only. This could have been a common SSL/TLS decryption key used only by the Rebels so that the Rebels could decrypt the messages using Wireshark with the decryption key they have (Wireshark can decrypt HTTPS provided you have the correct SSL/TLS decryption key). This would largely be difficult for an Imperial Forensic Analyst because the HTTPS traffic would need to be decrypted first

to then get analyzed. This would have bought enough time for the Rebels to execute their plans.

- Never keep traces of the software you use in the form of pictures. One must try to lower the chances of figuring out the exact manner and use of any software. The Images folder in Autopsy told me exactly that the Rebel Malware Writer used VeraCrypt because there were pictures of the tutorial given by VeraCrypt for people who have never used the software to learn how to use the tool. The Rebel Malware Writer might have taken photos of this for reference and stored it but that led me to believe that they did directly use VeraCrypt for encrypting some file/files.

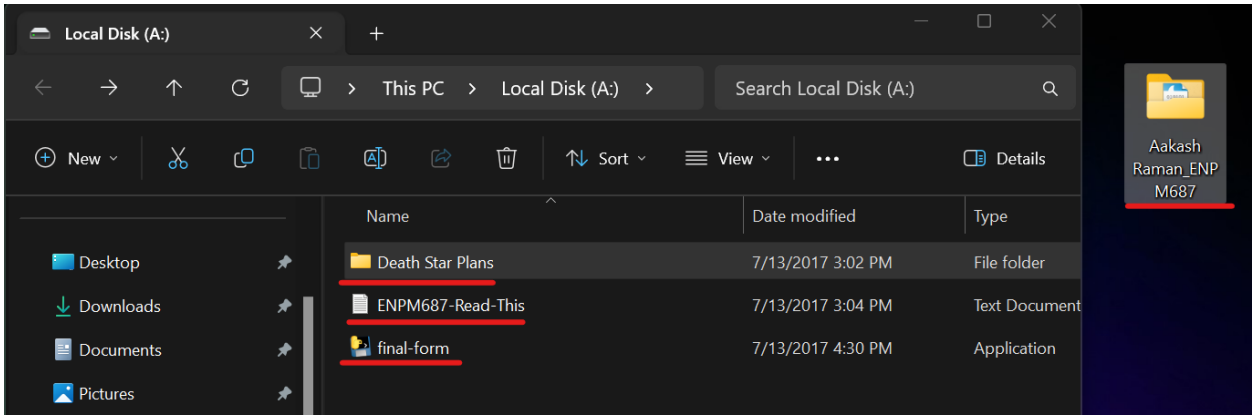
- The key was easily found out because Base64 encoding was used on it. If the Rebels used some other uncommon/hard-reverse encoding format like Unicode or invented their own encoding format, it would have been impossible to decipher the message and get the key even if only the HTTP protocol was being used.

Overall, in the future, the Rebels should take care of these points and if they do it will greatly increase the complexity of the Imperial Forensic Analysts from deciphering their actions.

Questions

1. **Find the final version of the malware writer's malware.**

Answer: The final version of the Malware Writer's Malware is "***final-form.exe***". It was found after decrypting the "***not-the-droids-youre-looking-for.mp3***" file using the "***r2d2***" key in the VeraCrypt software.



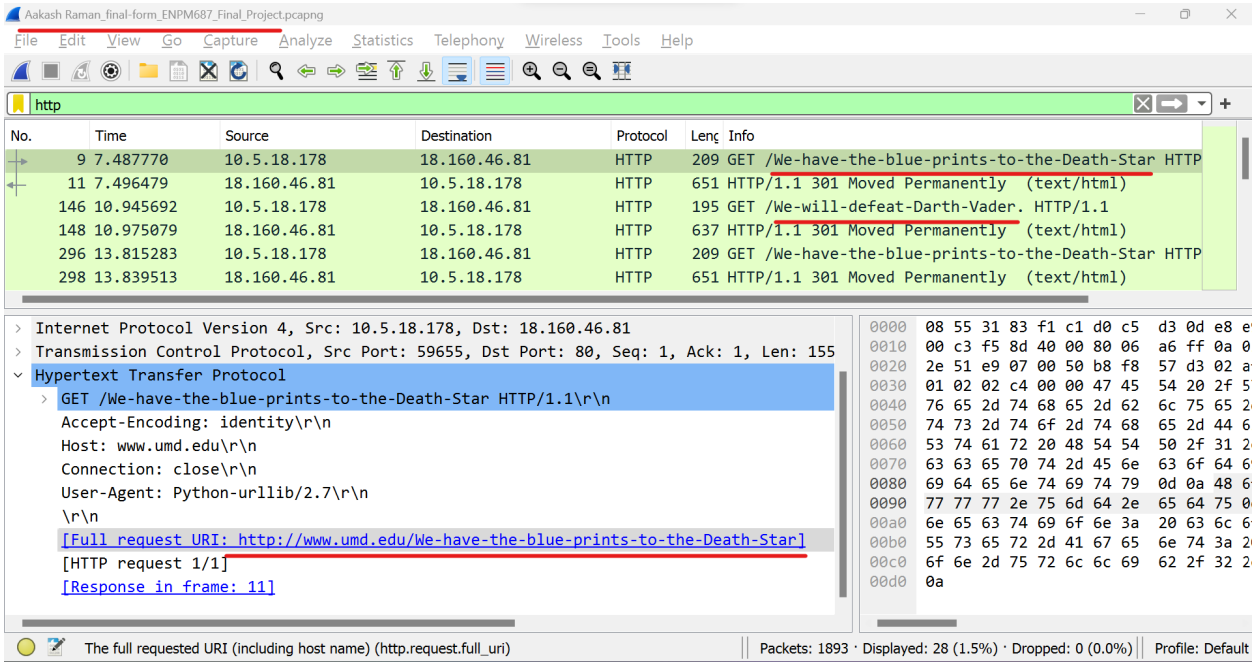
2. **Determine what the message contained inside of the final malware is.**

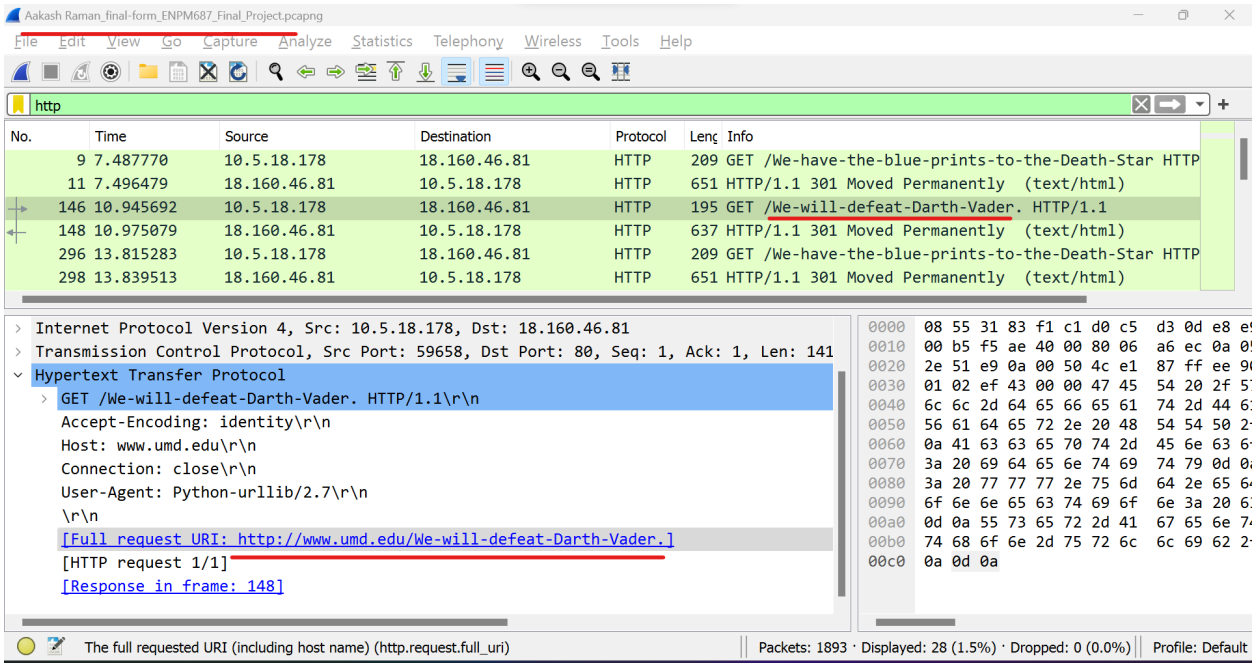
Answer: After running the "final-form.exe" malware, and capturing its packets using Wireshark, the malware reached out to 2 URLs:

- ***http://www.umd.edu/We-have-the-blue-prints-to-the-Death-Star***
- ***http://www.umd.edu/We-will-defeat-Darth-Vader*.**

Based on the URLs, the messages they contained were:

- ***We-have-the-blue-prints-to-the-Death-Star***
- ***We-will-defeat-Darth-Vader.***





3. **Find some other interesting items/artifacts/clues that are definitely 'relevant' to the investigation that will aid the Imperial Forces. Include these in your report in order to get full credit.**

Answer: The other interesting artifacts and clues that were definitely "relevant" to the investigation were the files, "**obiwan.exe**", "**obiwan2.exe**", "**not-the-droids-youre-looking-for.mp3**" and the "**cjJkMiBpcyB0aGUga2V5**" message sent by the "obiwan2.exe" executable. Both the executable files gave me an idea of how the Rebels were sending their messages in the form of HTTP GET Requests. The Base64 message from "**obiwan2.exe**", "**cjJkMiBpcyB0aGUga2V5**" translates to "**r2d2 is the key**" when Base64 decoded, which helped me know what the encryption key the Rebel Malware Writer used.

The "**not-the-droids-youre-looking-for.mp3**" file was the encrypted file inside which I found the "final-form.exe", the true form of the Rebel's Malware. Although the logical meaning was that as the "**not-the-droids-youre-looking-for.mp3**" file was in the Encryption Suspected folder in Autopsy, it might be encrypted, even if you think commonly, R2D2 is a popular droid famous in the Star Wars universe and is loved by the Rebels hence it makes the most sense that it is the key for decrypting this file. The screenshots of all these files are above in the Analysis section.

4. **Describe two challenges or difficulties you had to overcome to complete the final project.**

Answer: 2 challenges that I had to overcome to complete the Final Project were:

- It was difficult to find the "**not-the-droids-youre-looking-for.mp3**" file and be certain that this was the file that the Rebel Malware Writer had encrypted with the "r2d2" key because I thought normally the contents that the Malware author might have encrypted would have been a text or document file, but as this was an MP3 Audio file, it was a bit hard to find and correlate.

- Realizing that the "**cjJkMiBpcyB0aGUga2V5**" GET Request message obtained from "obiwan2.exe" was Base64 encoded was a bit confusing at first too, and I had to go through the entire packet capture to realize that it was Base64 encoded.

References

These are the references I used for the Final Project.

[1] https://forums.mozillazine.org/viewtopic.php?p=12536687
[2] https://gchq.github.io/CyberChef/