

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.

This screenshot shows the Autopsy 4.20.0 interface with the following details:

- Case:** Akash Raman_ENPM687_Final_Project
- Data Sources:** Virtual Disk.vmdk_1 Host, Virtual Disk.vmdk
 - File System: v01 (Unlocated: 0-55)
 - File System: v02 (NTFS / eFAT (0x07): 56-41926079)
 - File System: v03 (Unlocated: 41926080-41943039)
- File Views:** Deleted Files (10)
 - Santa_Isabel
 - Jayapura
 - Macau
 - GMT-14
 - Minsk
 - Kirtmati
 - hourglass.mask
 - openfile.xbm
- Data Artifacts:** Communication Accounts (3), E-Mail Messages (2), Installed Programs (24), Metadatas (49), Recent Documents (33), Run Programs (74), Shell Bags (31), USB Device Attached (9), Web Bookmarks (8), Web Cookies (102), Web Downloads (32), Web Form Autofill (2), Web History (22), Web Search (43).
- Analysis Results:** Encryption Suspected (4), EXP Metadata (3), Extension mismatch Detected (4), Interesting Items (1), Keyword Hits (3693), User Content Suspected (3), OS Account Categories (3).
- From The Sleuth Kit istat Tool:** No Data.
- File Metadata:** Name: /img/_VirtualDisk.vmdk/vol_v02/Python27/tb/tb8.3/tdata/Europe/Minsk, Type: File System, MIME Type: application/octet-stream, Size: 0, File Name Allocations: Unallocated, Metadata Allocations: Unallocated, Modified: 0000-00-00 00:00:00, Accessed: 0000-00-00 00:00:00, Created: 0000-00-00 00:00:00, Changed: 0000-00-00 00:00:00, MD5: d41d8cd98f002049800998ecfb42fe, SHA-256: e3b0c44298fc1c149a8fb4e3995fb9247ae41e4e49b934ca49591b7652b855, Hash Lookup Results: UNKNOWN, Internal ID: 1788.

This screenshot shows the Autopsy 4.20.0 interface with the following details:

- Case:** Akash Raman_ENPM687_Final_Project
- Data Sources:** Virtual Disk.vmdk_1 Host, Virtual Disk.vmdk
 - File System: v01 (Unlocated: 0-55)
 - File System: v02 (Unlocated: 41926080-41943039)
- File Views:** Deleted Files (10)
 - [current folder]
 - [parent folder]
 - donindex.html
 - GeneratedSupport.html
 - index.html
 - misc.html
 - package.html
 - PythonCOM.html
 - QuarkScriptClientVm.html
- Data Artifacts:** Communication Accounts (3), E-Mail Messages (2), Installed Programs (24), Metadatas (49), Operating System Information (1), Recent Documents (33), Run Programs (74), Shell Bags (31), USB Device Attached (9), Web Bookmarks (8), Web Cookies (102), Web Downloads (32), Web Form Autofill (2), Web History (22), Web Search (43).
- Analysis Results:** Encryption Suspected (4), EXP Metadata (3), Extension mismatch Detected (4), Interesting Items (1), Keyword Hits (3693), User Content Suspected (3), OS Account Categories (3).
- PythonCOM Documentation Index**
- The following documentation is available:
 - A Quick Start to Client Side COM (including makekey)
 - A Quick Start to Server Side COM
 - Information on generated Python files (i.e., what makekey generates)
 - An advanced VARIANT object which can give more control over parameter types
 - A brief description of the win32com package structure
 - Python COM Implementation documentation
 - Misc stuff I don't know where to put anywhere else
- ActiveX Scripting**
- ActiveX Scripting Demos

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.

The screenshot shows the Autopsy Forensic Browser interface. The left sidebar contains a tree view of data sources, file types, deleted files, and analysis results. The main pane displays a table of deleted files with columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, Flag(Dir), Flag(Meta), Known, and Location. One file, 'test-malware-simple.pset', is highlighted in red. The table includes several other entries like 'C:\Windows\Temp\...', 'CF01C7D...', 'A11D12...', 'goog-unwanted-shar...', 'windows.ao', 'firefox_newScene1...', 'Plyt8t.wpl', and 'MPC7DE.bmp'. At the bottom, there's a detailed view of the selected file 'test-malware-simple.pset' with sections for Metadata, File Contents, and Hex View.

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_vo12/Program Files/VeraCrypt/docs/html/en/twitter_veracrypt.png*” and “*/img_Virtual Disk.vmdk/vol_vo12/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.

The screenshot shows the Autopsy 4.20.0 interface with the following details:

- File System Listing:** A table view of files from a mounted volume (Virtual Disk.vmdk). The columns include Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, Flags(Dir), Flags(Meta), Known Location, and a preview icon.
- Selected File:** `twitter_veracrypt.html` is selected, showing its detailed metadata. The metadata includes:
 - Name: `twitter_veracrypt.html`
 - Type: File System
 - MIME Type: application/octet-stream
 - Size: 0
 - File Name Allocation: Unallocated
 - Metadata Allocation:
 - Kept By: 0000-00-00-00-00-00
 - Accessed: 0000-00-00-00-00-00
 - Created: 0000-00-00-00-00-00
 - Changed: 0000-00-00-00-00-00
 - MD5: d41d8cd98f002d4e9998ecbf9427e
 - SHA-256: e3b0c44298fc1c19aef5fc4d99fb9247ee41e4649b934ca499991b7852b855
 - Hash Lookup Results: UNKNOWN
 - Internal ID: 9655
- Annotations:** Annotations tab is visible in the header.
- Bottom Navigation:** From The Sleuth Kit istat Took: No Data

The screenshot shows the Aakash Raman ENPM687.Final Project - Autopsy 4.2.0 interface. The top menu bar includes Case, View, Tools, Window, Help, and several icons for file types like Images/Videos, Communications, Geolocation, Timeline, Discovery, Generate Report, and Close Case. The main window has a left sidebar with a tree view of data sources, file views, file types, and data artifacts. The central area displays a table of file metadata with columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, Flags(Dr), Flags(Meta), Known, and Location. A 'Save Table as CSV' button is at the top right of the table. Below the table is a navigation bar with links for Hex, Text, Application, File Metadata, OS Account, Data Artifacts, Analyze Results, Context, Annotations, and Other Occurrences. The bottom left shows a detailed view of the 'File Metadata' section for a specific file, and the bottom right shows the 'From The Sleuth Kit istat Tool' section with 'No Data'.

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.

Autopsy 4.20.0 File Analysis Results

File Types:

- By Extension:
 - Image (692)
 - Video (14)
 - Audio (155)
 - Archives (55)
 - Databases (15)
 - Documents (476)
 - HTML (476)
 - Office (15)
 - PDF (2)
 - Plain Text (495)
 - Rich Text (0)
 - Text (1)
 - XML (1)
- Deleted Files: All (3131)
- File System (2016)
- MB File Size
- OS Artifacts
- System Artifacts
- Communication Accounts (3)
- Email Messages (2)
- Installed Programs (24)
- Metadata (40)
 - Operating System Information (1)
 - Processor Components (3)
 - Run Programs (24)
 - Shell Beeps (31)
 - USB Device Attached (9)
 - Web Bookmarks (8)
 - Web Cookies (102)
 - Web Form Autofill (2)
 - Web Form Autofill (3)
 - Web History (22)
 - Web Search (43)
- Analysis Results
 - EDP-Metadate (4)
 - EDP-Metadate (3)
 - Extension Mismatch Detected (4)
 - Interesting Items (1)

VeraCrypt License

Software distributed under this license is distributed on an "AS IS" BASIS WITHOUT WARRANTIES OF ANY KIND. THE AUTHORS AND DISTRIBUTORS OF THE SOFTWARE DISCLAIM ALL LIABILITY. ANYONE WHO USES, COPIES, MODIFIES, OR (RE)DISTRIBUTES ANY PART OF THE SOFTWARE IS, BY SUCH ACTION(S), ACCEPTING AND AGREEING TO BE BOUND BY ALL TERMS AND CONDITIONS OF THIS LICENSE. IF YOU DO NOT ACCEPT THEM, DO NOT USE, COPY, MODIFY, NOR ANY PART(S) THEREOF.

This license does not grant you rights to use any contributors' name, logo, or trademarks, including IDRAC, VeraCrypt and all derivative names. For example, the following names are not allowed: VeraCrypt, VeraCrypt+, VeraCrypt Professional, iVeraCrypt, etc. Nor any other names confusingly similar to the name VeraCrypt (e.g., Vera-Crypt, Vera Crypt, Verkrypt, etc.).

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

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_vo12/Documents and Settings/Administrator/My Documents/code/dist/obowan.exe**” as shown below. Similarly, “*obiwan2.exe*”’s file path is “**/img_Virtual Disk.vmdk/vol_vo12/Documents and Settings/Administrator/My Documents/code/dist/obowan2.exe**” also 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

Data Sources

- Virtual Disk.vmdk, 1 Host
 - Virtual Disk (0-53)
 - vol1 (NTFS / exFAT (0x07): 56-41926079)
 - vol2 (Unallocated: 41926080-41943039)

File Types

- By Extension
 - Images (692)
 - Videos (14)
 - Office (15)
 - PDF (2)
 - Plain Text (495)
 - Rich Text (0)
- Executable
 - Batch (984)
 - .bat (723)
 - .cmd (4)
 - .com (15)
 - By MIME Type
 - File System (20)
 - AI (113)
- MB File Size
- Data Artifacts

Listing

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
microsoft.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 07:00:35 EDT	2017-07-12 11:04:30 EDT	2008-04-14 08:00:00 EDT	126464	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obwain.exe	0	2017-07-13 14:44:27 EDT	2017-07-13 15:51:21 EDT	2017-07-13 14:44:27 EDT	2017-07-13 15:50:59 EDT	1491789	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obwain2.exe	0	2017-07-13 15:30:44 EDT	2017-07-13 15:51:21 EDT	2017-07-13 15:30:44 EDT	2017-07-13 15:50:59 EDT	1491789	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcad32.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	32768	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcad32.exe	4	2008-04-14 08:00:00 EDT	2017-07-13 13:42:59 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	32768	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcconf.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	69632	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcconf.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	69632	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
oemig30.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:02:50 EDT	2017-07-12 11:04:30 EDT	2017-07-12 11:02:50 EDT	60416	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Metadata

Name: /img_Virtual Disk.vmdk/vol_vo2/Documents and Settings/Administrator/My Documents/code/obwain.exe

Type: File System

MIME Type: application/x-msdos-compressed

Size: 4206347

File Name Allocation: Allocated

Metadata Allocation: Allocated

Modified: 2017-07-13 14:44:27 EDT

Accessed: 2017-07-13 14:44:27 EDT

Created: 2017-07-13 14:44:27 EDT

Changed: 2017-07-13 15:31:21 EDT

MDS: 9709c03d80e00c22549e9b5f943

SHA-256: 90441659379e29d29c821dafecc7bdc10e695de2e36605e6d4519ac887d7

Hash Lookup Result: UNKNOWN

Internal ID: 3005

From The Sleuth Kit Istat Took

MFT Entry Header Values:

Entry: 17777 Sequence: 4

\$LogFile Sequence Number: 94953742

Allocated File

Links: 1

STANDARD_INFORMATION Attribute Values:

Flags: Archive

Owner ID: 0

Security ID: 387 (S-1-5-21-57989841-1972179041-1801674831-500)

Created: 2017-07-13 14:44:27 444883800 (EDT)

File Modified: 2017-07-13 14:44:27 4604644100 (EDT)

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

Data Sources

- Virtual Disk.vmdk, 1 Host
 - Virtual Disk (0-53)
 - vol1 (NTFS / exFAT (0x07): 56-41926079)
 - vol2 (Unallocated: 41926080-41943039)

File Types

- By Extension
 - Images (692)
 - Videos (14)
 - Office (15)
 - PDF (2)
 - Plain Text (495)
 - Rich Text (0)
- Executable
 - Batch (984)
 - .bat (723)
 - .cmd (4)
 - .com (15)
 - Deleted Files
 - File System (20)
 - AI (113)
- MB File Size
- Data Artifacts

Listing

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
microsoft.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 07:00:35 EDT	2017-07-12 11:04:30 EDT	2008-04-14 08:00:00 EDT	126464	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obwain.exe	0	2017-07-13 14:44:27 EDT	2017-07-13 15:51:21 EDT	2017-07-13 14:44:27 EDT	2017-07-13 15:50:59 EDT	1491789	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obwain2.exe	0	2017-07-13 15:30:44 EDT	2017-07-13 15:51:21 EDT	2017-07-13 15:30:44 EDT	2017-07-13 15:50:59 EDT	1491789	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcad32.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	32768	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcad32.exe	4	2008-04-14 08:00:00 EDT	2017-07-13 13:42:59 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	32768	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcconf.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	69632	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
obcconf.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	2008-04-14 08:00:00 EDT	2017-07-12 11:03:56 EDT	69632	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk
oemig30.exe	4	2008-04-14 08:00:00 EDT	2017-07-12 11:02:50 EDT	2017-07-12 11:04:30 EDT	2017-07-12 11:02:50 EDT	60416	Allocated	Allocated	Unknown			/img_Virtual Disk.vmdk

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Metadata

Name: /img_Virtual Disk.vmdk/vol_vo2/Documents and Settings/Administrator/My Documents/code/obwain2.exe

Type: File System

MIME Type: application/x-msdos-compressed

Size: 491789

File Name Allocation: Allocated

Metadata Allocation: Allocated

Modified: 2017-07-13 15:30:44 EDT

Accessed: 2017-07-13 15:30:44 EDT

Created: 2017-07-13 15:10:58 EDT

Changed: 2017-07-13 15:31:21 EDT

MDS: d183c181b160e00c22549e9b5f943

SHA-256: 49963c29dd9d45cc2899133c7a7e077be8a0020d79921c7e88845

Hash Lookup Result: UNKNOWN

Internal ID: 3007

From The Sleuth Kit Istat Took

MFT Entry Header Values:

Entry: 17777 Sequence: 4

\$LogFile Sequence Number: 94953742

Allocated File

Links: 1

STANDARD_INFORMATION Attribute Values:

Flags: Archive

Owner ID: 0

Security ID: 387 (S-1-5-21-57989841-1972179041-1801674831-500)

Created: 2017-07-13 15:10:58 432529100 (EDT)

File Modified: 2017-07-13 15:10:58 420266700 (EDT)

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_vo2/Program Files/VeraCrypt/VeraCrypt-x64.exe”** and the VeraCrypt Setup executable file to be **“/img_Virtual Disk.vmdk/vol_vo2/Program Files/VeraCrypt/VeraCrypt Setup.exe”**. These are 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

Data Sources Virtual Disk.vmdk_1 Host Virtual Disk.vmdk_1 Host Virtual Disk.vmdk_1 Host

File View File Types By Extension Images (692) Videos (14) Audio (155) Archives (15) Databases (15) Documents HTML (474) Office (15) PDF (2) Plain Text (495) Rich Text (0) Executable Java (894) DLL (372) Bat (8) Cmd (4) Com (15) By MIME Type Deleted Files File System (2016) All (3131)

MB File Size Data Artifacts Communication Accounts (3) E-Mail Messages (2) Installed Programs (24) Metadata (40) Operating System Information (1) Recent Documents (33) Run Programs (74) Shell Bags (1) USB Device Attached (9) Web Bookmarks (0) Web Cookies (102) Web Downloads (32) Web Form Audit (3) Web History (221) Web Search (43)

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Listing

Table: **Thumbnail** Summary

Name S C O Modified Time Change Time Access Time Created Time Size Flags(Dr) Flags(Meta) Known Location

VMwareFerlogs.exe 3 2016-09-29 00:34:02 EDT 2017-07-12 11:06:06 EDT 2017-07-12 11:06:06 EDT 2016-09-29 00:34:02 EDT 128704 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptFormat-x64.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 5610128 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptFormat.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 7008100 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.1.21.exe 1 2017-07-13 13:45:25 EDT 2017-07-13 13:45:50 EDT 2017-07-13 13:45:50 EDT 2017-07-13 14:56:37 EDT 2017-07-13 13:45:10 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.1.21.exe.Zone.Identifier 4 2017-07-13 13:45:25 EDT 2017-07-13 13:45:50 EDT 2017-07-13 13:45:50 EDT 2017-07-13 14:56:37 EDT 2017-07-13 13:45:10 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.exe 1 2017-07-13 13:45:25 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup-x64.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 5597040 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 6966928 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCrypt.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 6966928 Allocated Allocated unknown /img_Virtual Disk.vmdk

Save Table as CSV Keyword Search

964 Results

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

Data Sources Virtual Disk.vmdk_1 Host Virtual Disk.vmdk_1 Host Virtual Disk.vmdk_1 Host

File View File Types By Extension Images (692) Videos (14) Audio (155) Archives (15) Databases (15) Documents HTML (474) Office (15) PDF (2) Plain Text (495) Rich Text (0) Executable Java (894) DLL (372) Bat (8) Cmd (4) Com (15) By MIME Type Deleted Files File System (2016) All (3131)

MB File Size Data Artifacts Communication Accounts (3) E-Mail Messages (2) Installed Programs (24) Metadata (40) Operating System Information (1) Recent Documents (33) Run Programs (74) Shell Bags (1) USB Device Attached (9) Web Bookmarks (0) Web Cookies (102) Web Downloads (32) Web Form Audit (3) Web History (221) Web Search (43)

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Listing

Table: **Thumbnail** Summary

Name S C O Modified Time Change Time Access Time Created Time Size Flags(Dr) Flags(Meta) Known Location

VMwareFerlogs.exe 3 2016-09-29 00:34:02 EDT 2017-07-12 11:06:06 EDT 2017-07-12 11:06:06 EDT 2016-09-29 00:34:02 EDT 128704 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptFormat-x64.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 5610128 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptFormat.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 7008100 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.1.21.exe 1 2017-07-13 13:45:25 EDT 2017-07-13 13:45:50 EDT 2017-07-13 13:45:50 EDT 2017-07-13 14:56:37 EDT 2017-07-13 13:45:10 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.1.21.exe.Zone.Identifier 4 2017-07-13 13:45:25 EDT 2017-07-13 13:45:50 EDT 2017-07-13 13:45:50 EDT 2017-07-13 14:56:37 EDT 2017-07-13 13:45:10 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.exe 1 2017-07-13 13:45:25 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2965696 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup-x64.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 5597040 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCryptSetup.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 6966928 Allocated Allocated unknown /img_Virtual Disk.vmdk

VeraCrypt.exe 0 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 2017-07-13 13:45:47 EDT 6966928 Allocated Allocated unknown /img_Virtual Disk.vmdk

Save Table as CSV Keyword Search

964 Results

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_vo1/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 “obian” and “obian2” folders as shown below. The files “warnobiwan.txt” and “warnobiwan2.txt” inside the “obian” and “obian2” folders are very interesting. This is shown below.

Data Sources

- Virtual Disk.vmdk Host
 - vol1 (Uncached: 0-5\$)
 - vol2 (Uncached: 0x07): 56-41926079)
 - [Orphaned] (480)
 - [Deleted] (1)
 - [Recovered] (5)
 - [Valid] (18)
 - Documents and Settings (7)
 - Administrator (17)
 - Applikation (6)
 - Desktop (1)
 - Desktop (2)
 - Favorites (4)
 - Local Settings (7)
 - My Documents (7)
 - code (8)
 - build (5)
 - final-form (12)
 - obwarz (1)
 - obwarz (10)
 - obwarz2 (10)
 - dat (4)
 - Downloads (9)
 - My Music (23)
 - My Pictures (4)
 - Network (2)
 - PrintProof (2)
 - Recent (36)
 - SendTo (8)
 - Start Menu (4)
 - Terminals (14)
 - All Files (19)
 - Default User (98)
 - LocationService (8)
 - NetworkService (8)
 - Program Files (22)
 - Python27 (16)
 - RECYCLER (3)
 - System Volume Information (5)
 - WINDOWS (17)
 - vol3 (Uncached: 4192608-41943039)

Listing
/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flag(Dir)	Flag(Meta)	Known	Location
current folder				2017-07-14 19:39:42 EDT	2017-07-14 19:39:42 EDT	2017-07-14 19:39:42 EDT	2017-07-13 14:44:15 EDT	56	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/current folder
[parent folder]				2017-07-14 19:39:58 EDT	2017-07-14 19:39:58 EDT	2017-07-14 19:39:58 EDT	2017-07-12 02:11:05 EDT	56	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/[parent folder]
build				2017-07-13 15:10:53 EDT	2017-07-13 15:10:53 EDT	2017-07-13 15:11:02 EDT	2017-07-13 14:44:15 EDT	448	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/build
dist				2017-07-13 15:35:06 EDT	2017-07-13 15:35:06 EDT	2017-07-13 15:35:06 EDT	2017-07-13 14:44:15 EDT	256	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/dist
final-form.spec	0			2017-07-13 15:30:49 EDT	2017-07-13 15:30:49 EDT	2017-07-14 19:39:36 EDT	2017-07-13 15:30:49 EDT	764	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/final-form.spec
obwarz.py	0			2017-07-13 14:36:28 EDT	2017-07-13 14:36:28 EDT	2017-07-13 14:36:28 EDT	2017-07-13 14:44:15 EDT	250	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/obwarz.py
obwarz.spec	0			2017-07-13 14:44:22 EDT	2017-07-13 14:44:22 EDT	2017-07-13 14:44:22 EDT	2017-07-13 14:44:15 EDT	756	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/obwarz.spec
obwarz2.spec	0			2017-07-13 15:30:59 EDT	2017-07-13 15:30:59 EDT	2017-07-13 15:30:59 EDT	2017-07-13 15:30:49 EDT	758	Allocated	Allocated	unknown	/img_Virtual_Disk.vmdk/vol.vol2/Documents and Settings/Administrator/My Documents/code/obwarz2.spec

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Page: 1 of 1 Page: < > Go to Page: 1 Jump to Offset: Launch in HD

```
0x00000000: 30 00 00 00 01 00 00 00 20 00 00 00 01 00 00 00 .....  
0x00000010: 10 00 00 20 00 00 20 00 00 00 01 00 00 00 .....  
0x00000020: 00 00 00 00 00 00 00 00 18 00 00 00 03 00 00 00 .....  
0x00000030: 00 00 00 00 00 00 00 00 .....  
0x00000040: .....  
0x00000050: .....  
0x00000060: .....  
0x00000070: .....  
0x00000080: .....  
0x00000090: .....  
0x000000A0: .....  
0x000000B0: .....  
0x000000C0: .....  
0x000000D0: .....  
0x000000E0: .....  
0x000000F0: .....  
0x000000G0: .....  
0x000000H0: .....  
0x000000I0: .....  
0x000000J0: .....  
0x000000K0: .....  
0x000000L0: .....  
0x000000M0: .....  
0x000000N0: .....  
0x000000O0: .....  
0x000000P0: .....  
0x000000Q0: .....  
0x000000R0: .....  
0x000000S0: .....  
0x000000T0: .....  
0x000000U0: .....  
0x000000V0: .....  
0x000000W0: .....  
0x000000X0: .....  
0x000000Y0: .....  
0x000000Z0: .....  
0x000000a0: .....  
0x000000b0: .....  
0x000000c0: .....  
0x000000d0: .....  
0x000000e0: .....  
0x000000f0: .....  
0x000000g0: .....  
0x000000h0: .....  
0x000000i0: .....  
0x000000j0: .....  
0x000000k0: .....  
0x000000l0: .....  
0x000000m0: .....  
0x000000n0: .....  
0x000000o0: .....  
0x000000p0: .....  
0x000000q0: .....  
0x000000r0: .....  
0x000000s0: .....  
0x000000t0: .....  
0x000000u0: .....  
0x000000v0: .....  
0x000000w0: .....  
0x000000x0: .....  
0x000000y0: .....  
0x000000z0: .....  
0x000000a1: .....  
0x000000b1: .....  
0x000000c1: .....  
0x000000d1: .....  
0x000000e1: .....  
0x000000f1: .....  
0x000000g1: .....  
0x000000h1: .....  
0x000000i1: .....  
0x000000j1: .....  
0x000000k1: .....  
0x000000l1: .....  
0x000000m1: .....  
0x000000n1: .....  
0x000000o1: .....  
0x000000p1: .....  
0x000000q1: .....  
0x000000r1: .....  
0x000000s1: .....  
0x000000t1: .....  
0x000000u1: .....  
0x000000v1: .....  
0x000000w1: .....  
0x000000x1: .....  
0x000000y1: .....  
0x000000z1: .....  
0x000000a2: .....  
0x000000b2: .....  
0x000000c2: .....  
0x000000d2: .....  
0x000000e2: .....  
0x000000f2: .....  
0x000000g2: .....  
0x000000h2: .....  
0x000000i2: .....  
0x000000j2: .....  
0x000000k2: .....  
0x000000l2: .....  
0x000000m2: .....  
0x000000n2: .....  
0x000000o2: .....  
0x000000p2: .....  
0x000000q2: .....  
0x000000r2: .....  
0x000000s2: .....  
0x000000t2: .....  
0x000000u2: .....  
0x000000v2: .....  
0x000000w2: .....  
0x000000x2: .....  
0x000000y2: .....  
0x000000z2: .....  
0x000000a3: .....  
0x000000b3: .....  
0x000000c3: .....  
0x000000d3: .....  
0x000000e3: .....  
0x000000f3: .....  
0x000000g3: .....  
0x000000h3: .....  
0x000000i3: .....  
0x000000j3: .....  
0x000000k3: .....  
0x000000l3: .....  
0x000000m3: .....  
0x000000n3: .....  
0x000000o3: .....  
0x000000p3: .....  
0x000000q3: .....  
0x000000r3: .....  
0x000000s3: .....  
0x000000t3: .....  
0x000000u3: .....  
0x000000v3: .....  
0x000000w3: .....  
0x000000x3: .....  
0x000000y3: .....  
0x000000z3: .....  
0x000000a4: .....  
0x000000b4: .....  
0x000000c4: .....  
0x000000d4: .....  
0x000000e4: .....  
0x000000f4: .....  
0x000000g4: .....  
0x000000h4: .....  
0x000000i4: .....  
0x000000j4: .....  
0x000000k4: .....  
0x000000l4: .....  
0x000000m4: .....  
0x000000n4: .....  
0x000000o4: .....  
0x000000p4: .....  
0x000000q4: .....  
0x000000r4: .....  
0x000000s4: .....  
0x000000t4: .....  
0x000000u4: .....  
0x000000v4: .....  
0x000000w4: .....  
0x000000x4: .....  
0x000000y4: .....  
0x000000z4: .....  
0x000000a5: .....  
0x000000b5: .....  
0x000000c5: .....  
0x000000d5: .....  
0x000000e5: .....  
0x000000f5: .....  
0x000000g5: .....  
0x000000h5: .....  
0x000000i5: .....  
0x000000j5: .....  
0x000000k5: .....  
0x000000l5: .....  
0x000000m5: .....  
0x000000n5: .....  
0x000000o5: .....  
0x000000p5: .....  
0x000000q5: .....  
0x000000r5: .....  
0x000000s5: .....  
0x000000t5: .....  
0x000000u5: .....  
0x000000v5: .....  
0x000000w5: .....  
0x000000x5: .....  
0x000000y5: .....  
0x000000z5: .....  
0x000000a6: .....  
0x000000b6: .....  
0x000000c6: .....  
0x000000d6: .....  
0x000000e6: .....  
0x000000f6: .....  
0x000000g6: .....  
0x000000h6: .....  
0x000000i6: .....  
0x000000j6: .....  
0x000000k6: .....  
0x000000l6: .....  
0x000000m6: .....  
0x000000n6: .....  
0x000000o6: .....  
0x000000p6: .....  
0x000000q6: .....  
0x000000r6: .....  
0x000000s6: .....  
0x000000t6: .....  
0x000000u6: .....  
0x000000v6: .....  
0x000000w6: .....  
0x000000x6: .....  
0x000000y6: .....  
0x000000z6: .....  
0x000000a7: .....  
0x000000b7: .....  
0x000000c7: .....  
0x000000d7: .....  
0x000000e7: .....  
0x000000f7: .....  
0x000000g7: .....  
0x000000h7: .....  
0x000000i7: .....  
0x000000j7: .....  
0x000000k7: .....  
0x000000l7: .....  
0x000000m7: .....  
0x000000n7: .....  
0x000000o7: .....  
0x000000p7: .....  
0x000000q7: .....  
0x000000r7: .....  
0x000000s7: .....  
0x000000t7: .....  
0x000000u7: .....  
0x000000v7: .....  
0x000000w7: .....  
0x000000x7: .....  
0x000000y7: .....  
0x000000z7: .....  
0x000000a8: .....  
0x000000b8: .....  
0x000000c8: .....  
0x000000d8: .....  
0x000000e8: .....  
0x000000f8: .....  
0x000000g8: .....  
0x000000h8: .....  
0x000000i8: .....  
0x000000j8: .....  
0x000000k8: .....  
0x000000l8: .....  
0x000000m8: .....  
0x000000n8: .....  
0x000000o8: .....  
0x000000p8: .....  
0x000000q8: .....  
0x000000r8: .....  
0x000000s8: .....  
0x000000t8: .....  
0x000000u8: .....  
0x000000v8: .....  
0x000000w8: .....  
0x000000x8: .....  
0x000000y8: .....  
0x000000z8: .....  
0x000000a9: .....  
0x000000b9: .....  
0x000000c9: .....  
0x000000d9: .....  
0x000000e9: .....  
0x000000f9: .....  
0x000000g9: .....  
0x000000h9: .....  
0x000000i9: .....  
0x000000j9: .....  
0x000000k9: .....  
0x000000l9: .....  
0x000000m9: .....  
0x000000n9: .....  
0x000000o9: .....  
0x000000p9: .....  
0x000000q9: .....  
0x000000r9: .....  
0x000000s9: .....  
0x000000t9: .....  
0x000000u9: .....  
0x000000v9: .....  
0x000000w9: .....  
0x000000x9: .....  
0x000000y9: .....  
0x000000z9: .....  
0x000000a10: .....  
0x000000b10: .....  
0x000000c10: .....  
0x000000d10: .....  
0x000000e10: .....  
0x000000f10: .....  
0x000000g10: .....  
0x000000h10: .....  
0x000000i10: .....  
0x000000j10: .....  
0x000000k10: .....  
0x000000l10: .....  
0x000000m10: .....  
0x000000n10: .....  
0x000000o10: .....  
0x000000p10: .....  
0x000000q10: .....  
0x000000r10: .....  
0x000000s10: .....  
0x000000t10: .....  
0x000000u10: .....  
0x000000v10: .....  
0x000000w10: .....  
0x000000x10: .....  
0x000000y10: .....  
0x000000z10: .....  
0x000000a11: .....  
0x000000b11: .....  
0x000000c11: .....  
0x000000d11: .....  
0x000000e11: .....  
0x000000f11: .....  
0x000000g11: .....  
0x000000h11: .....  
0x000000i11: .....  
0x000000j11: .....  
0x000000k11: .....  
0x000000l11: .....  
0x000000m11: .....  
0x000000n11: .....  
0x000000o11: .....  
0x000000p11: .....  
0x000000q11: .....  
0x000000r11: .....  
0x000000s11: .....  
0x000000t11: .....  
0x000000u11: .....  
0x000000v11: .....  
0x000000w11: .....  
0x000000x11: .....  
0x000000y11: .....  
0x000000z11: .....  
0x000000a12: .....  
0x000000b12: .....  
0x000000c12: .....  
0x000000d12: .....  
0x000000e12: .....  
0x000000f12: .....  
0x000000g12: .....  
0x000000h12: .....  
0x000000i12: .....  
0x000000j12: .....  
0x000000k12: .....  
0x000000l12: .....  
0x000000m12: .....  
0x000000n12: .....  
0x000000o12: .....  
0x000000p12: .....  
0x000000q12: .....  
0x000000r12: .....  
0x000000s12: .....  
0x000000t12: .....  
0x000000u12: .....  
0x000000v12: .....  
0x000000w12: .....  
0x000000x12: .....  
0x000000y12: .....  
0x000000z12: .....  
0x000000a13: .....  
0x000000b13: .....  
0x000000c13: .....  
0x000000d13: .....  
0x000000e13: .....  
0x000000f13: .....  
0x000000g13: .....  
0x000000h13: .....  
0x000000i13: .....  
0x000000j13: .....  
0x000000k13: .....  
0x000000l13: .....  
0x000000m13: .....  
0x000000n13: .....  
0x000000o13: .....  
0x000000p13: .....  
0x000000q13: .....  
0x000000r13: .....  
0x000000s13: .....  
0x000000t13: .....  
0x000000u13: .....  
0x000000v13: .....  
0x000000w13: .....  
0x000000x13: .....  
0x000000y13: .....  
0x000000z13: .....  
0x000000a14: .....  
0x000000b14: .....  
0x000000c14: .....  
0x000000d14: .....  
0x000000e14: .....  
0x000000f14: .....  
0x000000g14: .....  
0x000000h14: .....  
0x000000i14: .....  
0x000000j14: .....  
0x000000k14: .....  
0x000000l14: .....  
0x000000m14: .....  
0x000000n14: .....  
0x000000o14: .....  
0x000000p14: .....  
0x000000q14: .....  
0x000000r14: .....  
0x000000s14: .....  
0x000000t14: .....  
0x000000u14: .....  
0x000000v14: .....  
0x000000w14: .....  
0x000000x14: .....  
0x000000y14: .....  
0x000000z14: .....  
0x000000a15: .....  
0x000000b15: .....  
0x000000c15: .....  
0x000000d15: .....  
0x000000e15: .....  
0x000000f15: .....  
0x000000g15: .....  
0x000000h15: .....  
0x000000i15: .....  
0x000000j15: .....  
0x000000k15: .....  
0x000000l15: .....  
0x000000m15: .....  
0x000000n15: .....  
0x000000o15: .....  
0x000000p15: .....  
0x000000q15: .....  
0x000000r15: .....  
0x000000s15: .....  
0x000000t15: .....  
0x000000u15: .....  
0x000000v15: .....  
0x000000w15: .....  
0x000000x15: .....  
0x000000y15: .....  
0x000000z15: .....  
0x000000a16: .....  
0x000000b16: .....  
0x000000c16: .....  
0x000000d16: .....  
0x000000e16: .....  
0x000000f16: .....  
0x000000g16: .....  
0x000000h16: .....  
0x000000i16: .....  
0x000000j16: .....  
0x000000k16: .....  
0x000000l16: .....  
0x000000m16: .....  
0x000000n16: .....  
0x000000o16: .....  
0x000000p16: .....  
0x000000q16: .....  
0x000000r16: .....  
0x000000s16: .....  
0x000000t16: .....  
0x000000u16: .....  
0x000000v16: .....  
0x000000w16: .....  
0x000000x16: .....  
0x000000y16: .....  
0x000000z16: .....  
0x000000a17: .....  
0x000000b17: .....  
0x000000c17: .....  
0x000000d17: .....  
0x000000e17: .....  
0x000000f17: .....  
0x000000g17: .....  
0x000000h17: .....  
0x000000i17: .....  
0x000000j17: .....  
0x000000k17: .....  
0x000000l17: .....  
0x000000m17: .....  
0x000000n17: .....  
0x000000o17: .....  
0x000000p17: .....  
0x000000q17: .....  
0x000000r17: .....  
0x000000s17: .....  
0x000000t17: .....  
0x000000u17: .....  
0x000000v17: .....  
0x000000w17: .....  
0x000000x17: .....  
0x000000y17: .....  
0x000000z17: .....  
0x000000a18: .....  
0x000000b18: .....  
0x000000c18: .....  
0x000000d18: .....  
0x000000e18: .....  
0x000000f18: .....  
0x000000g18: .....  
0x000000h18: .....  
0x000000i18: .....  
0x000000j18: .....  
0x000000k18: .....  
0x000000l18: .....  
0x000000m18: .....  
0x000000n18: .....  
0x000000o18: .....  
0x000000p18: .....  
0x000000q18: .....  
0x000000r18: .....  
0x000000s18: .....  
0x000000t18: .....  
0x000000u18: .....  
0x000000v18: .....  
0x000000w18: .....  
0x000000x18: .....  
0x000000y18: .....  
0x000000z18: .....  
0x000000a19: .....  
0x000000b19: .....  
0x000000c19: .....  
0x000000d19: .....  
0x000000e19: .....  
0x000000f19: .....  
0x000000g19: .....  
0x000000h19: .....  
0x000000i19: .....  
0x000000j19: .....  
0x000000k19: .....  
0x000000l19: .....  
0x000000m19: .....  
0x000000n19: .....  
0x000000o19: .....  
0x000000p19: .....  
0x000000q19: .....  
0x000000r19: .....  
0x000000s19: .....  
0x000000t19: .....  
0x000000u19: .....  
0x000000v19: .....  
0x000000w19: .....  
0x000000x19: .....  
0x000000y19: .....  
0x000000z19: .....  
0x000000a20: .....  
0x000000b20: .....  
0x000000c20: .....  
0x000000d20: .....  
0x000000e20: .....  
0x000000f20: .....  
0x000000g20: .....  
0x000000h20: .....  
0x000000i20: .....  
0x000000j20: .....  
0x000000k20: .....  
0x000000l20: .....  
0x000000m20: .....  
0x000000n20: .....  
0x000000o20: .....  
0x000000p20: .....  
0x000000q20: .....  
0x000000r20: .....  
0x000000s20: .....  
0x000000t20: .....  
0x000000u20: .....  
0x000000v20: .....  
0x000000w20: .....  
0x000000x20: .....  
0x000000y20: .....  
0x000000z20: .....  
0x000000a21: .....  
0x000000b21: .....  
0x000000c21: .....  
0x000000d21: .....  
0x000000e21: .....  
0x000000f21: .....  
0x000000g21: .....  
0x000000h21: .....  
0x000000i21: .....  
0x000000j21: .....  
0x000000k21: .....  
0x000000l21: .....  
0x000000m21: .....  
0x000000n21: .....  
0x000000o21: .....  
0x000000p21: .....  
0x000000q21: .....  
0x000000r21: .....  
0x000000s21: .....  
0x000000t21: .....  
0x000000u21: .....  
0x000000v21: .....  
0x000000w21: .....  
0x000000x21: .....  
0x000000y21: .....  
0x000000z21: .....  
0x000000a22: .....  
0x000000b22: .....  
0x000000c22: .....  
0x000000d22: .....  
0x000000e22: .....  
0x000000f22: .....  
0x000000g22: .....  
0x000000h22: .....  
0x000000i22: .....  
0x000000j22: .....  
0x000000k22: .....  
0x000000l22: .....  
0x000000m22: .....  
0x000000n22: .....  
0x000000o22: .....  
0x000000p22: .....  
0x000000q22: .....  
0x000000r22: .....  
0x000000s22: .....  
0x000000t22: .....  
0x000000u22: .....  
0x000000v22: .....  
0x000000w22: .....  
0x000000x22: .....  
0x000000y22: .....  
0x000000z22: .....  
0x000000a23: .....  
0x000000b23: .....  
0x000000c23: .....  
0x000000d23: .....  
0x000000e23
```

The screenshot shows the Autopsy 4.20.0 interface with the following details:

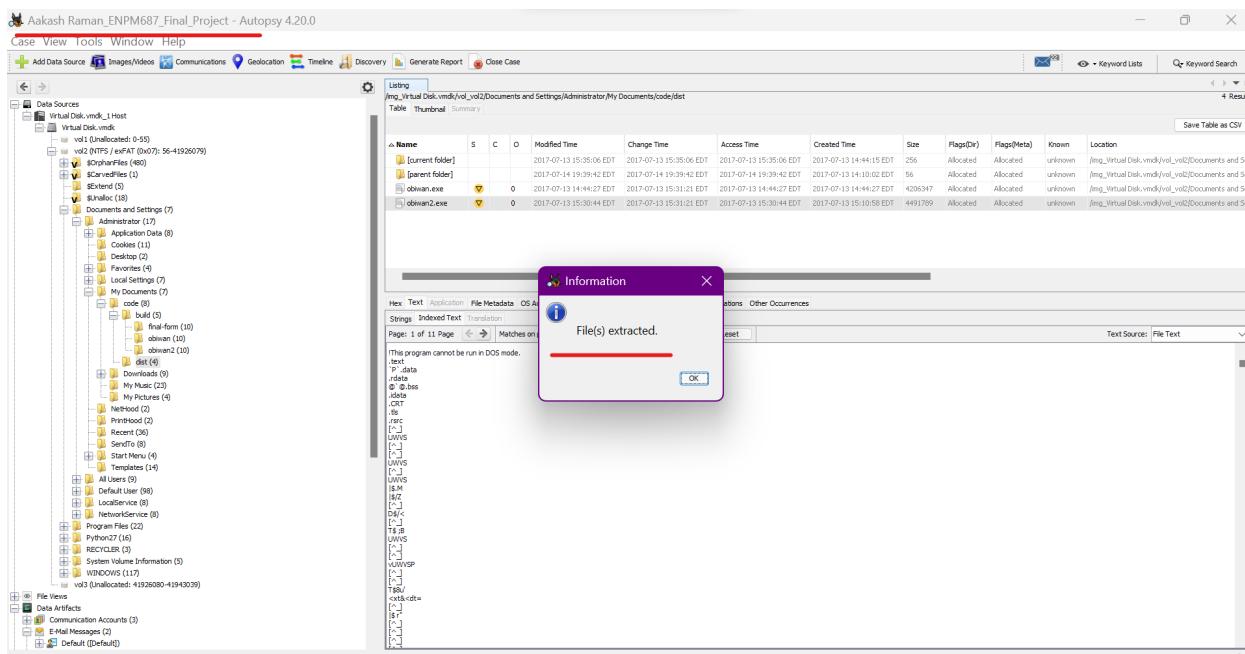
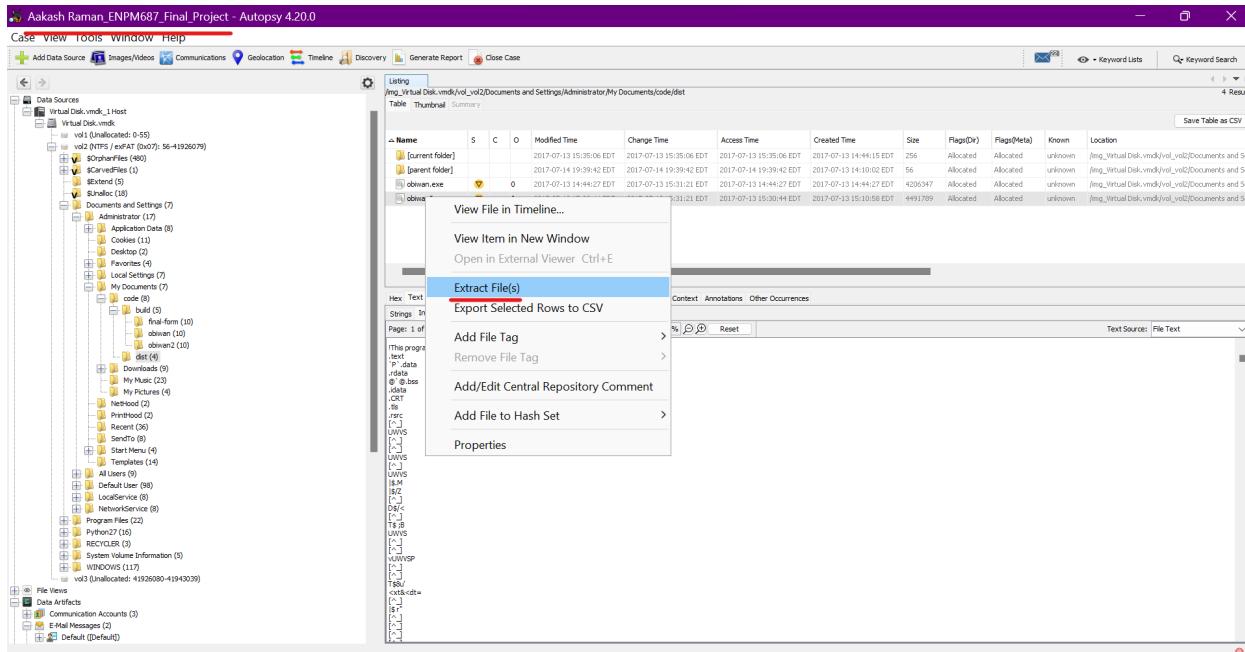
- Top Bar:** Aakash Raman ENPM687.Final Project - Autopsy 4.20.0
- Menu Bar:** Case, View, Tools, Window, Help
- Left Sidebar:** Data Sources (Virtual Disk, Local Disk, Host), File Views (Data Artifacts, Log Files, Configuration Acquires, Audit Messages, Default Forensics), Communications (IMAP, POP3, SMTP, SFTP, File Transfer, Network Service), Geolocation, Timeline, Discovery, Generate Report, Close Case.
- Central Area:**
 - File Listing:** Shows a list of files from a virtual disk. One file, "obwain.exe.manifest", is highlighted.
 - Search Results:** A search for "obwain" has been performed across the entire database. The results show various entries related to "obwain" across different files and timestamps.
- Bottom Bar:** Hex, Text, Application, File Metadata, OS Artifact, Data Artifacts, Analysis Results, Context, Annotations, Other Occurrences, Strings, Indexed Text, Translation, Page: 1 of 1 page, Matches on page: 0 - of 1 Match, 100%, Reset, Text Source: File Text.

The screenshot shows the Autopsy 4.20.0 interface. The main pane displays a file listing for the directory '/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/code/build/obian2'. The table includes columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, Flag(Dr), Flag(Meta), Known, and Location. Several files are listed, including 'obiwan.exe.manifest', 'out00-Analysis.toc', 'out00-EVE.toc', 'out00-PKG.pkg', 'out00-PKG.toc', 'out00-PIZ.pyz', and 'obiwan2.txt'. Below the table is a search results pane titled 'String' with the query 'obiwan'. It lists numerous matches across various files, such as 'obiwan.exe', 'obiwan2.exe', and 'obiwan2.txt'. The bottom of the search results pane shows a section titled '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.

The screenshot shows the Autopsy 4.20.0 interface with a context menu open over the 'obiwan2' folder in the file listing. The menu options include 'Extract File(s)', 'Export Selected Rows to CSV', 'Add File Tag', 'Remove File Tag', 'Add/Edit Central Repository Comment', 'Add File to Hash Set', and 'Properties'. The background shows the same file listing and search results as the previous screenshot.



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:

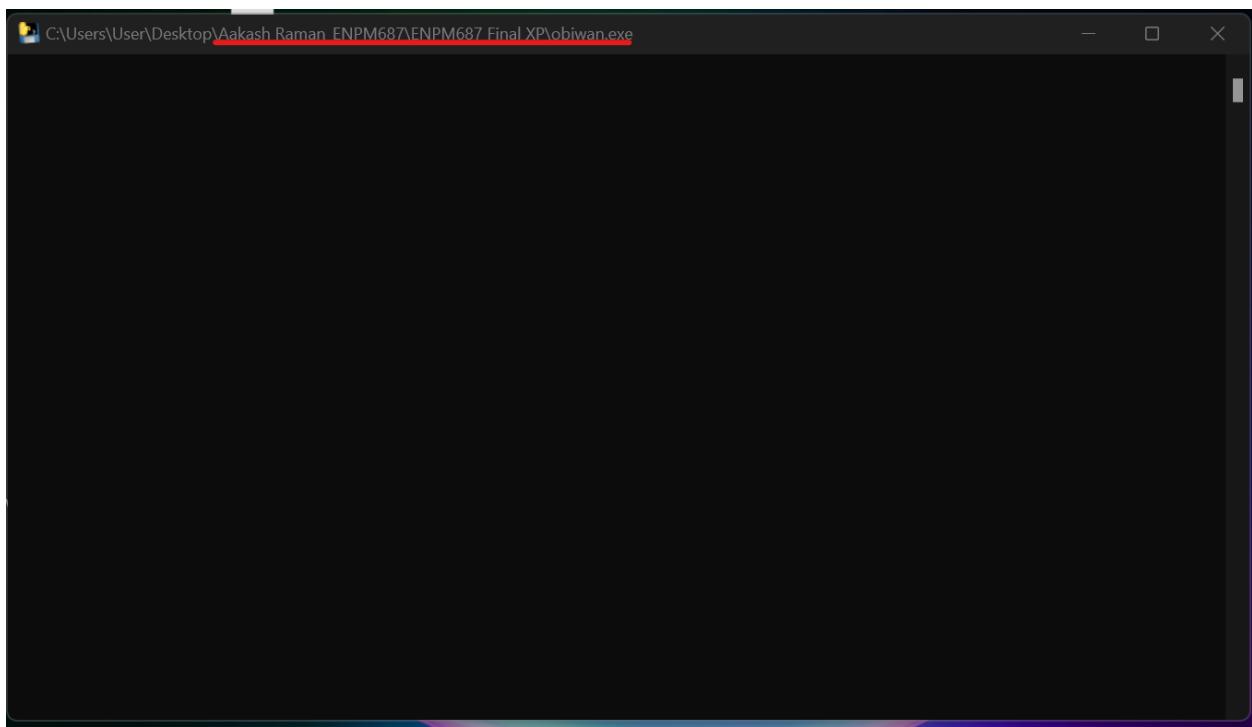
- <http://www.umd.edu/help-me-obiwan-kenobi>
- <http://www.umd.edu/youre-my-only-hope>

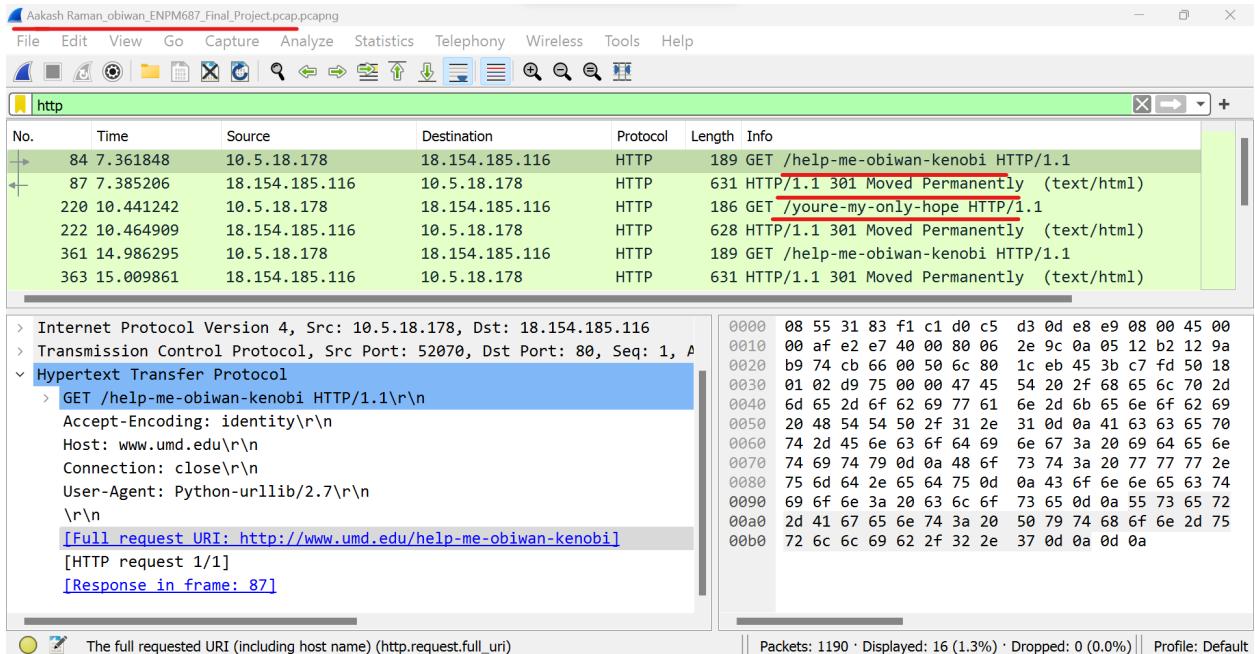
The decrypted messages from the “*obiwani.exe*” executable include:

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

The “*obiwani.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:

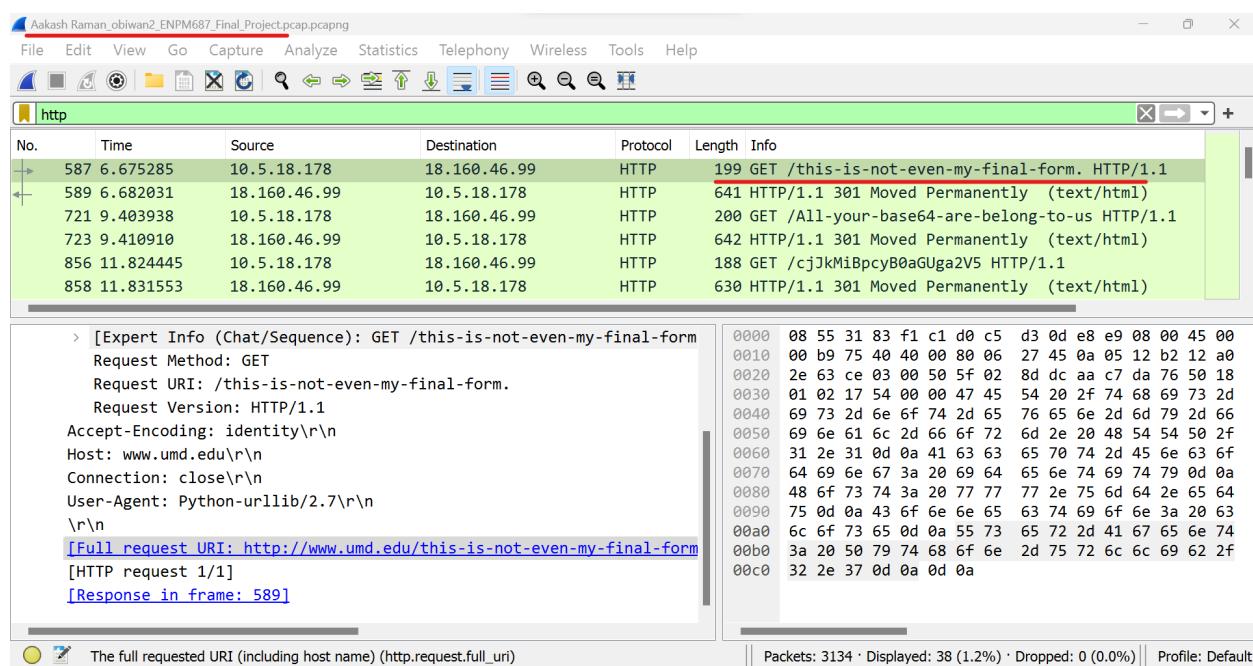
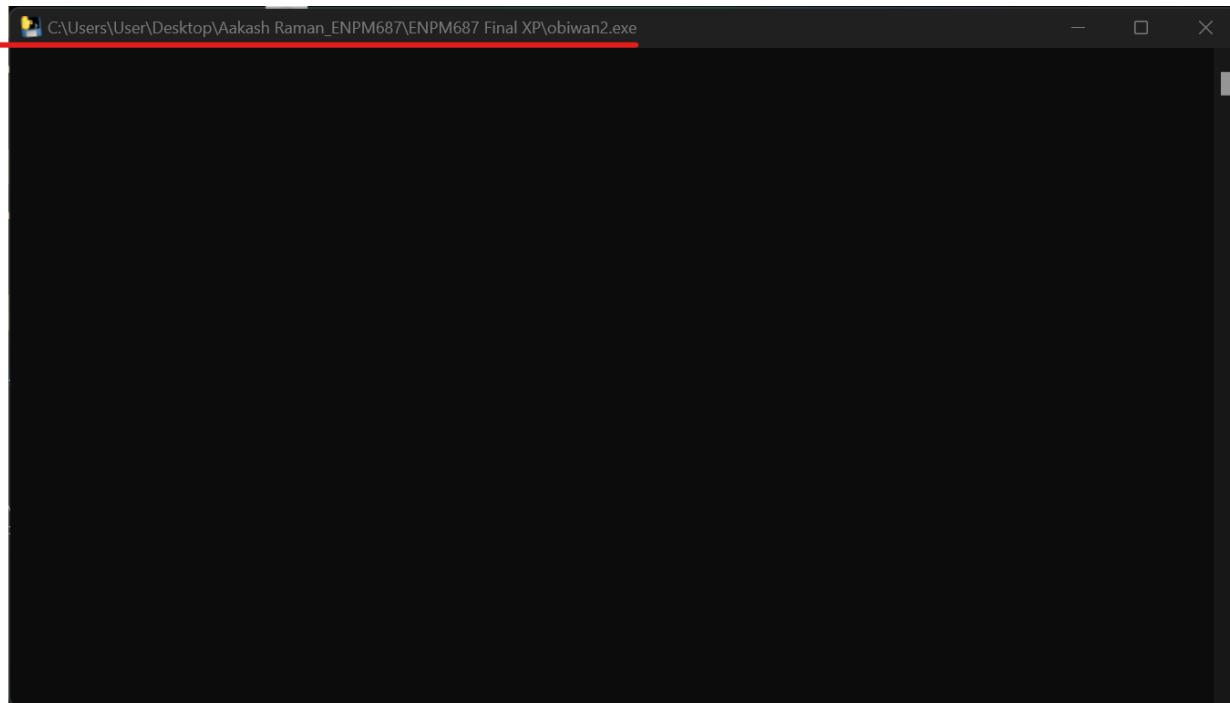
- <http://www.umd.edu/this-is-not-even-my-final-form>
- <http://www.umd.edu/All-your-base64-are-belong-to-us>
- <http://www.umd.edu/cjKMiBpcyB0aGUga2V5>

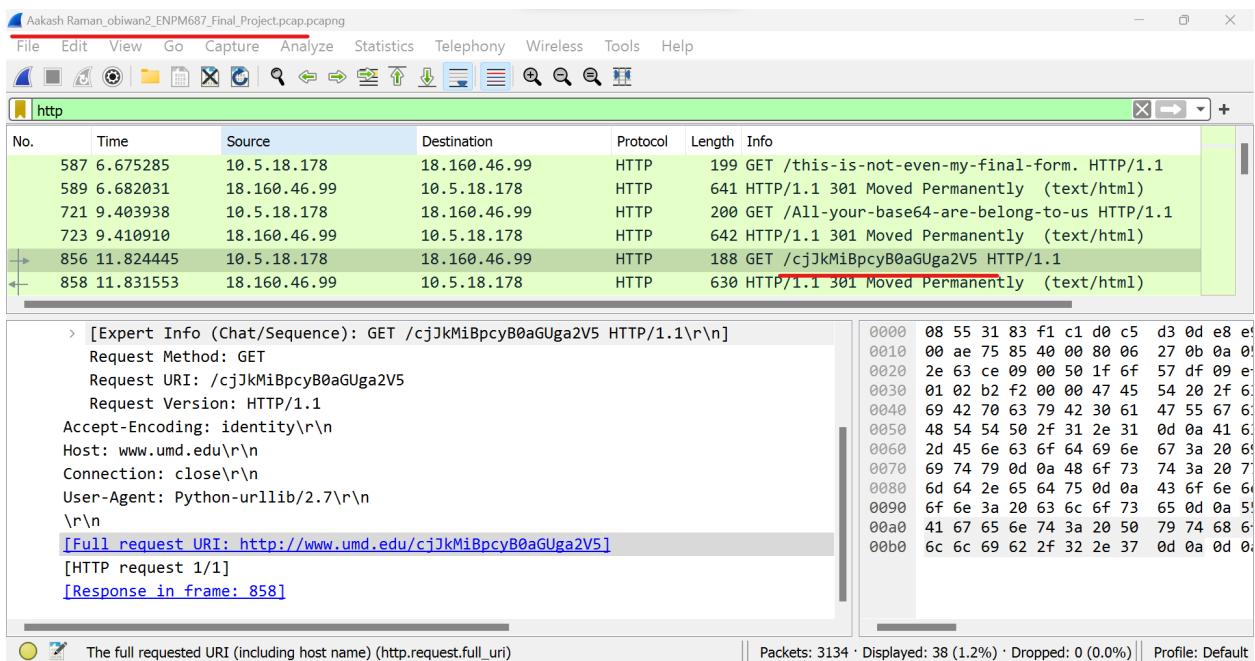
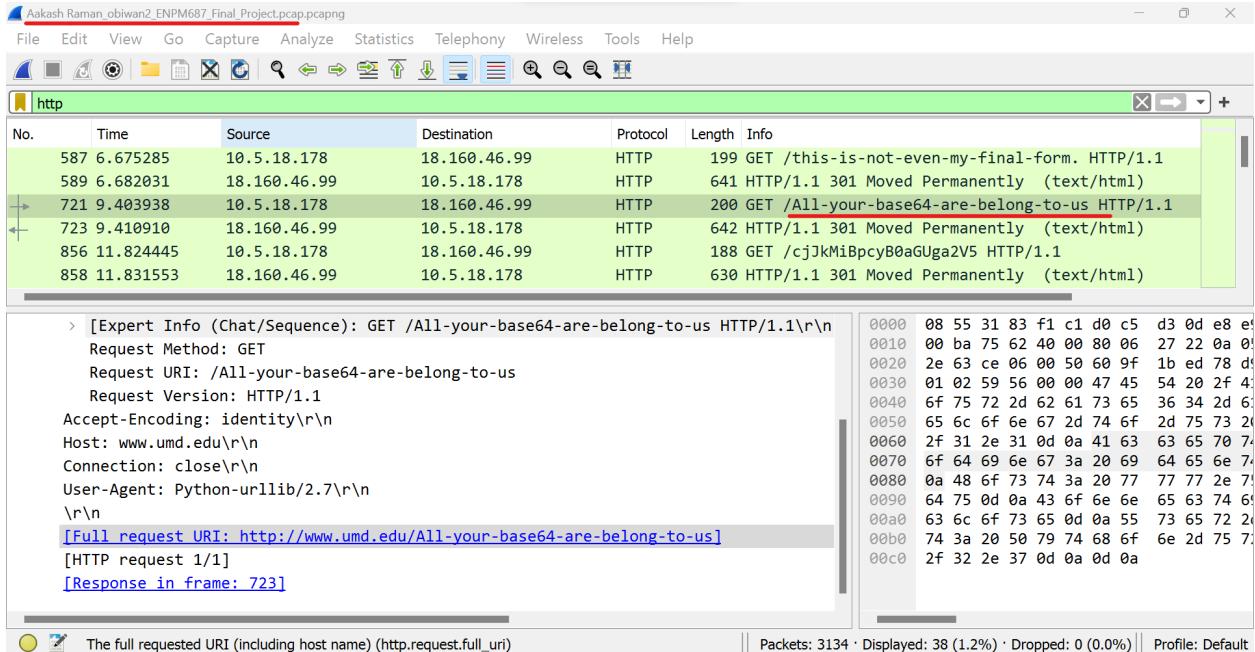
Similarly, the executable is also possibly written in Python 2.7 to query these 2 URLs because the User-Agent is a Python-urllib/2.7. Furthermore, the host it is querying is “www.umd.edu”.

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

- ***this-is-not-even-my-final-form***
- ***All-your-base64-are-belong-to-us***
- ***cjKMiBpcyB0aGUga2V5***

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.





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.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like Data format, Encryption / Encoding, Public Key, etc. The main area has a Recipe panel titled "From Base64" which includes settings for Alphabet (A-Za-z0-9+/=), Remove non-alphabet chars (checked), and Strict mode (unchecked). The Input field contains the Base64 string "cjJkMiBpcyB0aGUga2V5". The Output field shows the decoded result: "r2d2 is the key". Below the input and output fields are sections for Raw Bytes and LF.

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-you're-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.

The screenshot shows the Autopsy 4.20.0 interface. The left sidebar displays various data sources and artifacts. The main pane shows a table titled 'Encryption Suspected' with the following data:

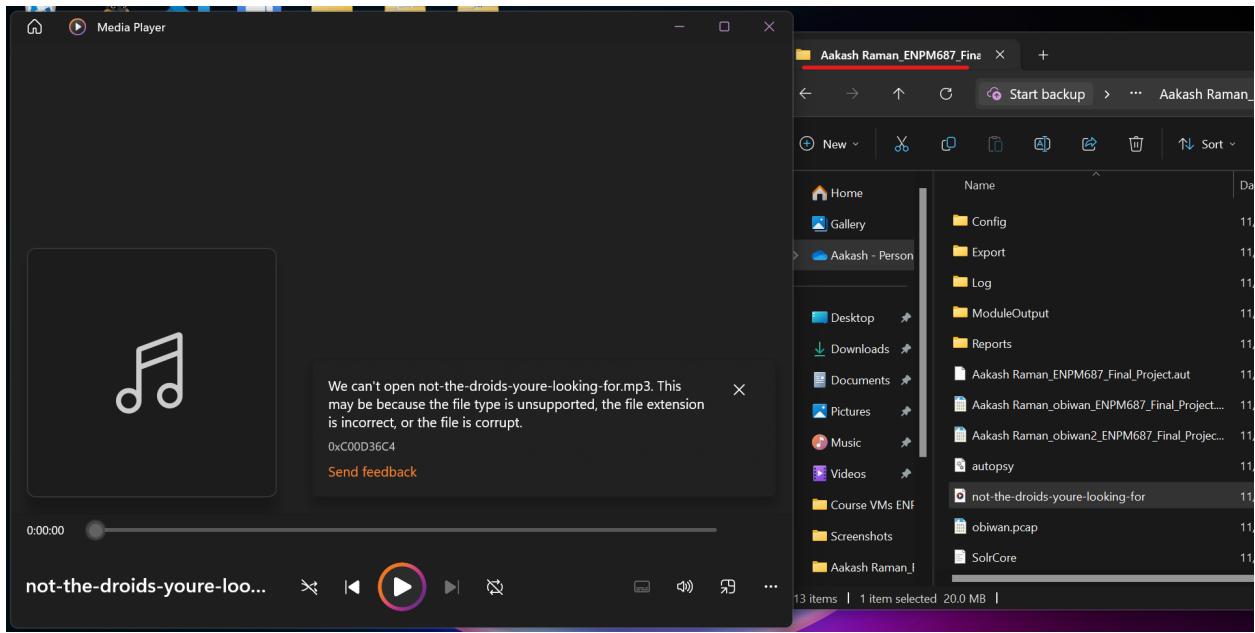
Source Name	S	C	O	Source Type	Score	Conclusion	Configuration	Justification	Comment
1bddef1fbcc270c29b040ab5dcb2bd319b8766eb8be2773edee	0			File	Likely Notable			Suspected encryption due to high entropy (7.999018).	Suspected encryption due to high entropy (7.999018).
not-the-drds-you're-looking-for.mp3	0			File	Likely Notable			Suspected encryption due to high entropy (7.999991).	Suspected encryption due to high entropy (7.999991).
oembos.bn	4			File	Likely Notable			Suspected encryption due to high entropy (7.999988).	Suspected encryption due to high entropy (7.999988).
oembos.bn	4			File	Likely Notable			Suspected encryption due to high entropy (7.999988).	Suspected encryption due to high entropy (7.999988).

The file 'not-the-drds-you're-looking-for.mp3' is highlighted in red, indicating it is likely encrypted.

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

The screenshot shows the Autopsy 4.20.0 interface with a confirmation dialog box in the foreground. The dialog says 'File(s) extracted.' and has an 'OK' button. The background shows the same 'Encryption Suspected' table as the previous screenshot.

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-you're-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.

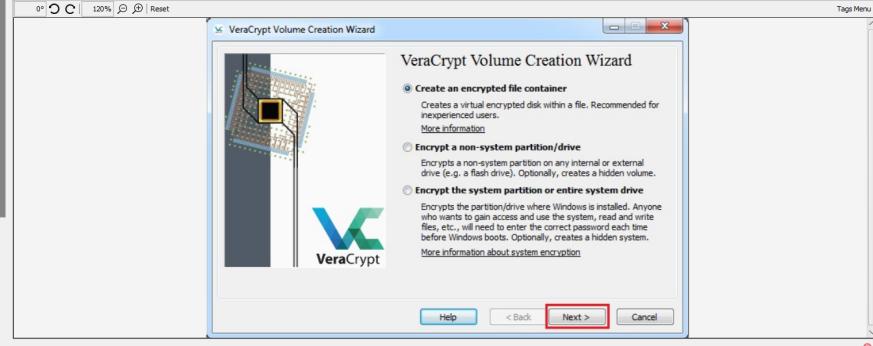
Aakash Raman ENPM687 Final Project - Autopsy 4.20.0

Case View Tools Window Help

Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case

Data Sources

- Virtual Disk (1 Host)
 - Virtual Disk (1)
 - vd2 (NTFS - eFAT (0x07) - 56-4192079)
 - vd3 (Unlocated: 41926980-41943039)
- File Views
- File Types
- By Extension
- Images (692)
 - Image (14)
 - Video (14)
 - Audio (155)
 - Archives (51)
 - Databases (15)
 - Documents (474)
 - HTML (474)
 - Office (15)
 - PDF (2)
 - Plain Text (495)
 - Rich Text (0)
 - Executable (95)
 - Exe (85)
 - .dll (3723)
 - .bst (0)
 - .cmd (4)
 - .com (15)
 - Script (1)
 - All (113)
 - Deleted Files
 - File System (2016)
 - All (313)
 - MB File Size
 - Data Artifacts
 - Communication Accounts (3)
 - E-Mail Messages (2)
 - Default (Default)
 - Installed Programs (24)
 - Metadata (46)
 - Operating System Information (1)
 - Recent Documents (33)
 - Run Programs (74)
 - Shell Bags (31)
 - USB Device Attached (5)
 - Web Bookmarks (8)
 - Web Cookies (102)
 - Web Downloads (32)
 - Web Form Autofill (3)
 - Web History (21)



Aakash Raman_ENPM687_Final Project - Autopsy 4.20.0

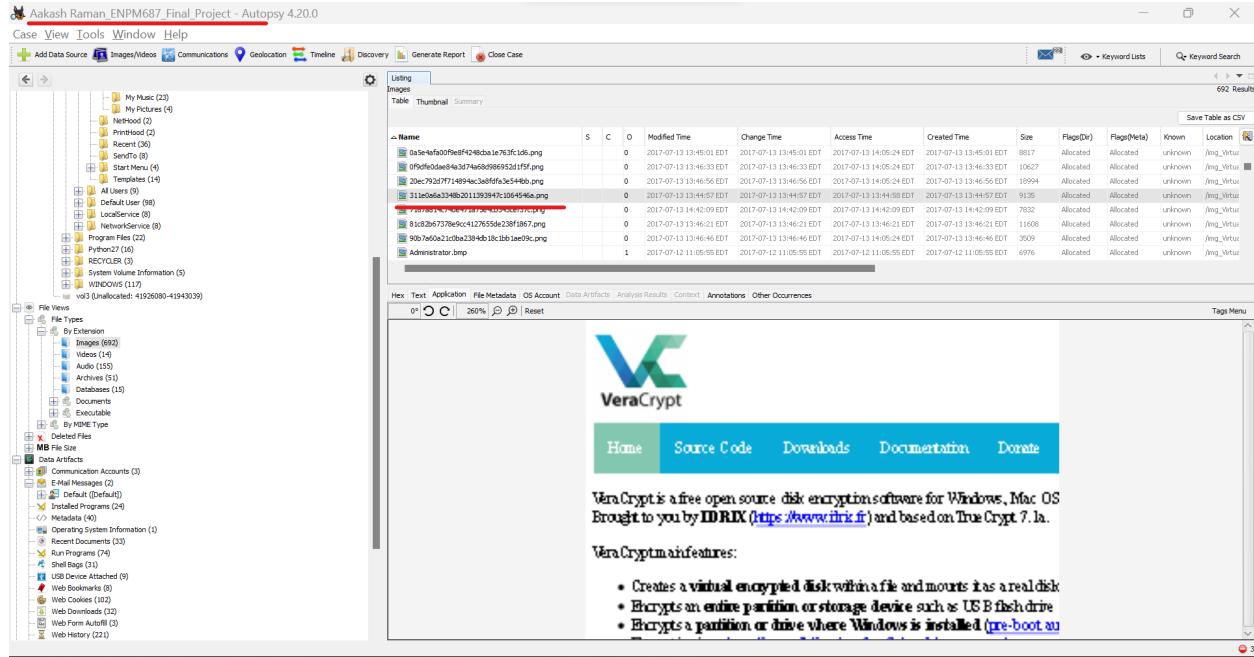
Case View Tools Window Help

Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case

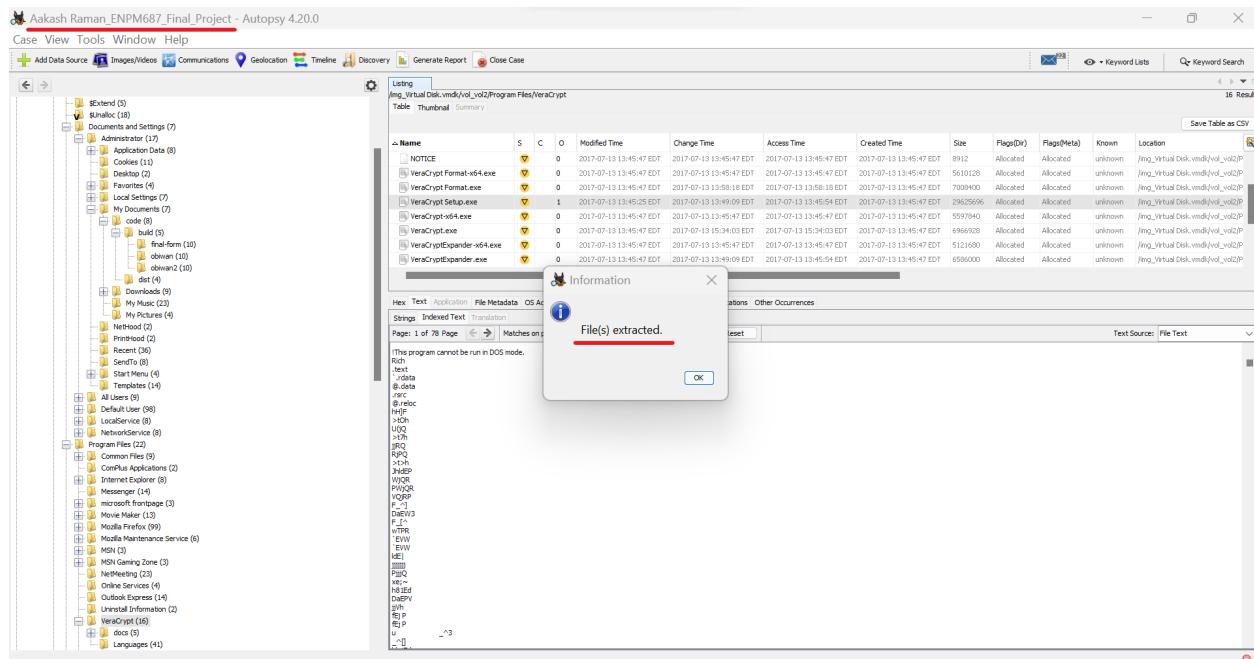
Data Sources

- Virtual Disk (1 Host)
 - Virtual Disk (1)
 - vd2 (NTFS - eFAT (0x07) - 56-4192079)
 - vd3 (Unlocated: 41926980-41943039)
- File Views
- File Types
- By Extension
- Images (692)
 - Image (14)
 - Video (14)
 - Audio (155)
 - Archives (51)
 - Databases (15)
 - Documents (474)
 - HTML (474)
 - Office (15)
 - PDF (2)
 - Plain Text (495)
 - Rich Text (0)
 - Executable (95)
 - Exe (85)
 - .dll (3723)
 - .bst (0)
 - .cmd (4)
 - .com (15)
 - Script (1)
 - All (113)
 - Deleted Files
 - File System (2016)
 - All (313)
 - MB File Size
 - Data Artifacts
 - Communication Accounts (3)
 - E-Mail Messages (2)
 - Default (Default)
 - Installed Programs (24)
 - Metadata (46)
 - Operating System Information (1)
 - Recent Documents (33)
 - Run Programs (74)
 - Shell Bags (31)
 - USB Device Attached (5)
 - Web Bookmarks (8)
 - Web Cookies (102)
 - Web Downloads (32)
 - Web Form Autofill (3)
 - Web History (21)

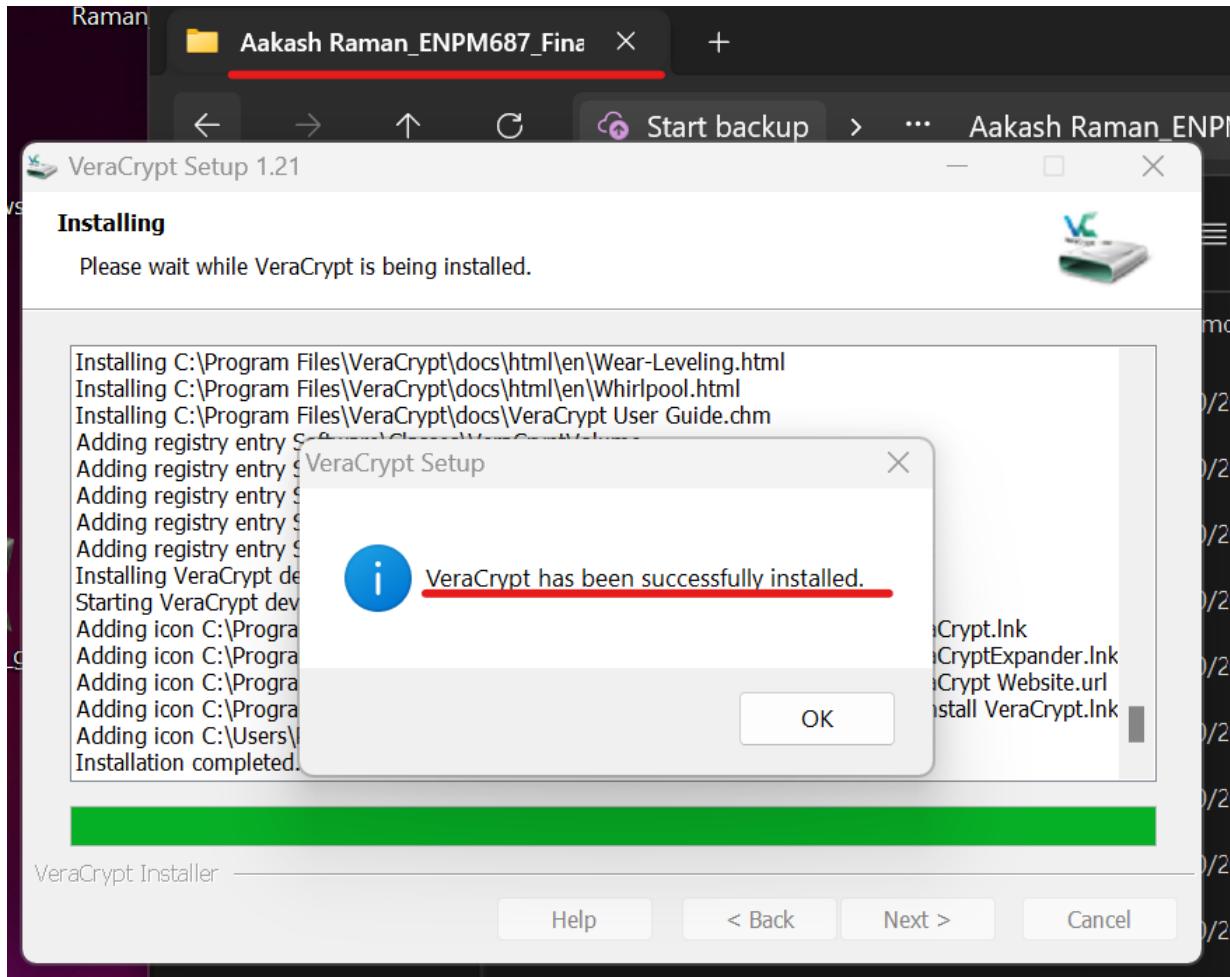




As I know where the VeraCrypt setup software is let me extract it and then try to decrypt the “***not-the-droids-you're-looking-for.mp3***” file. This is shown below.



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

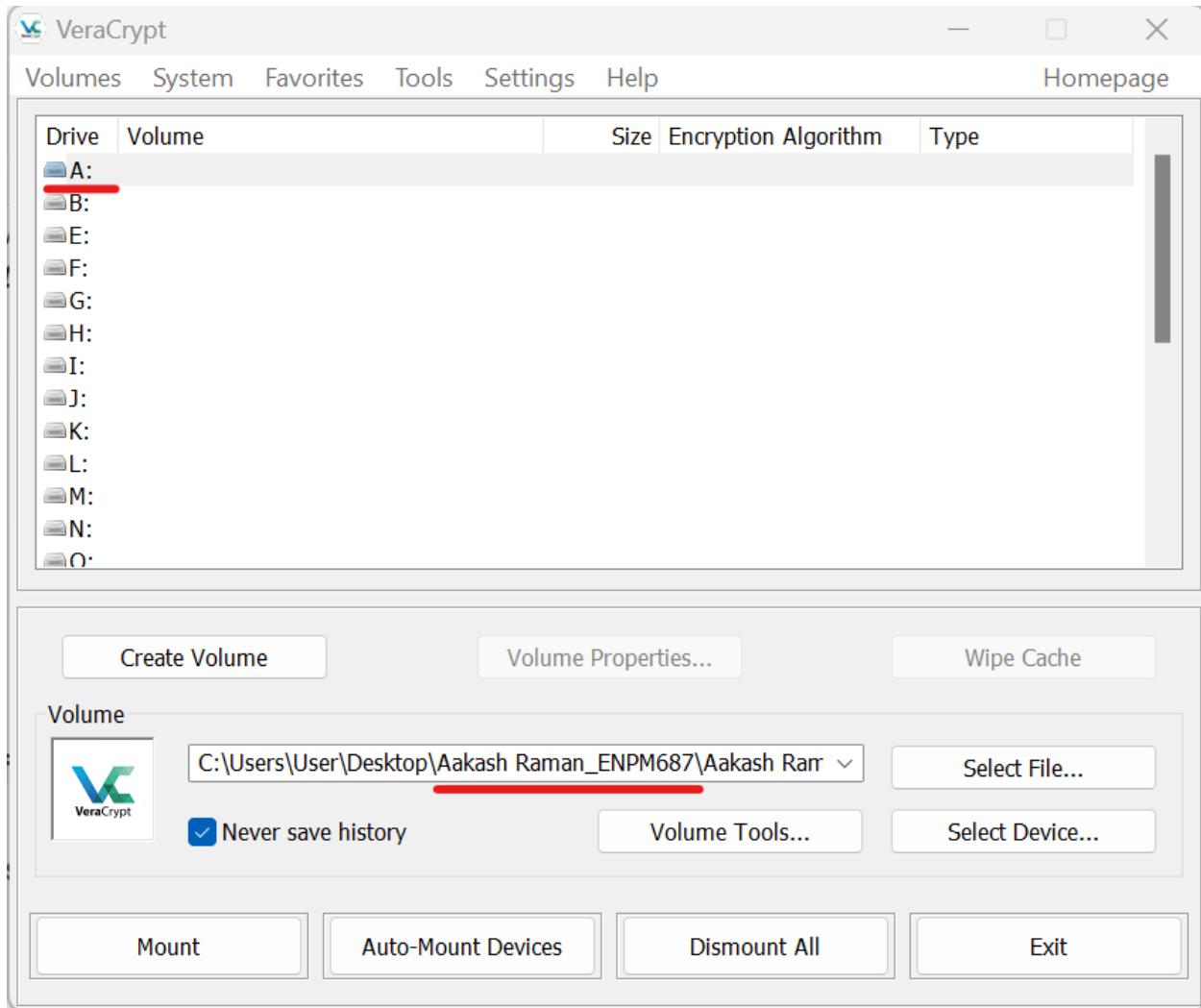


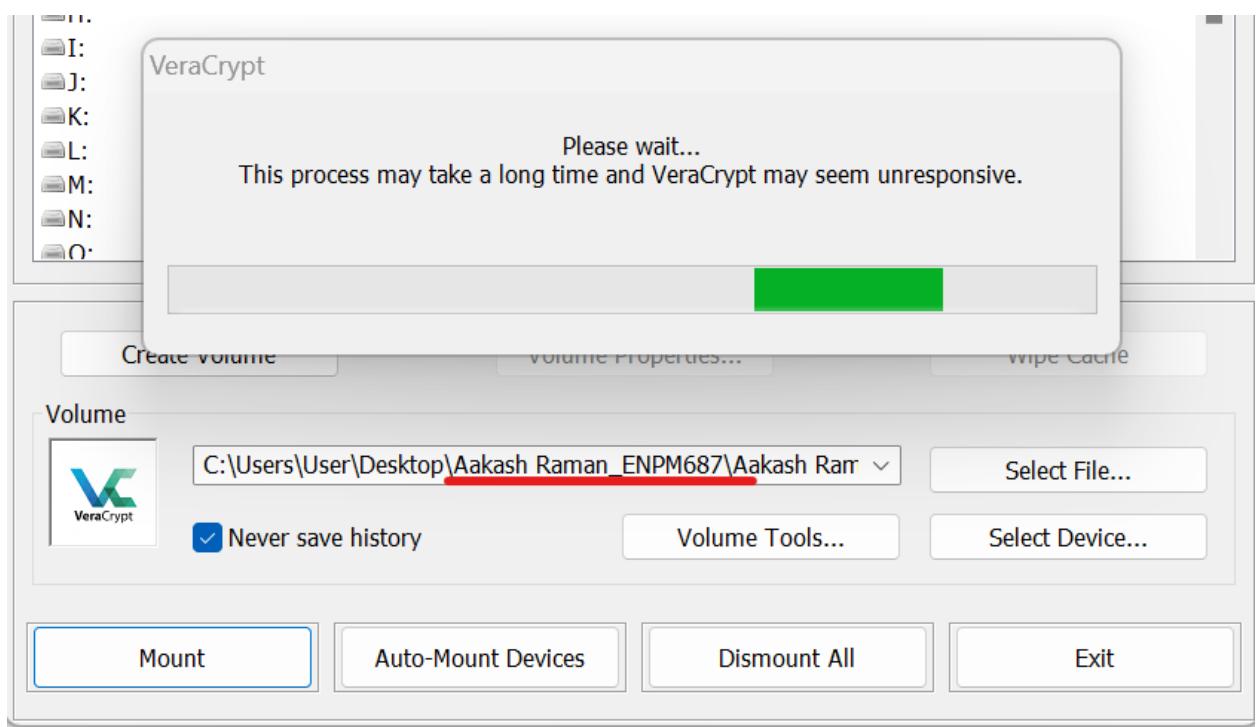
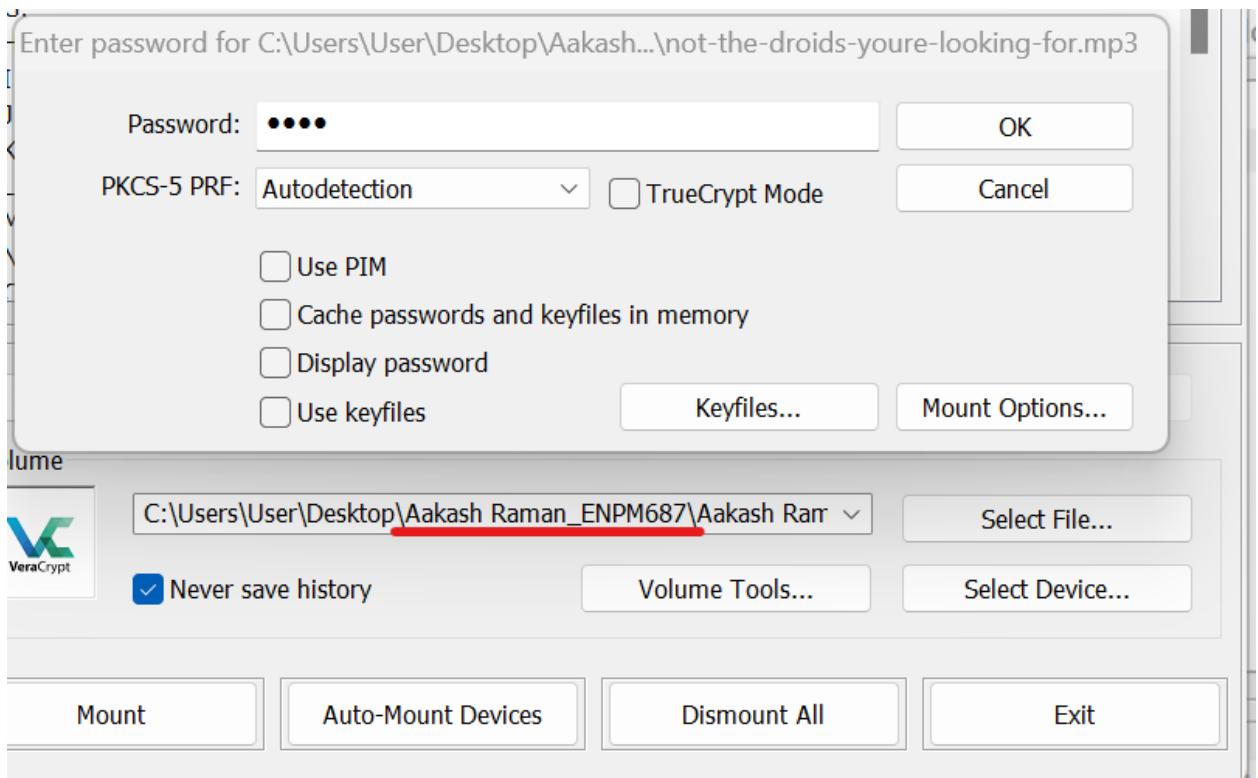
Decryption using VeraCrypt:

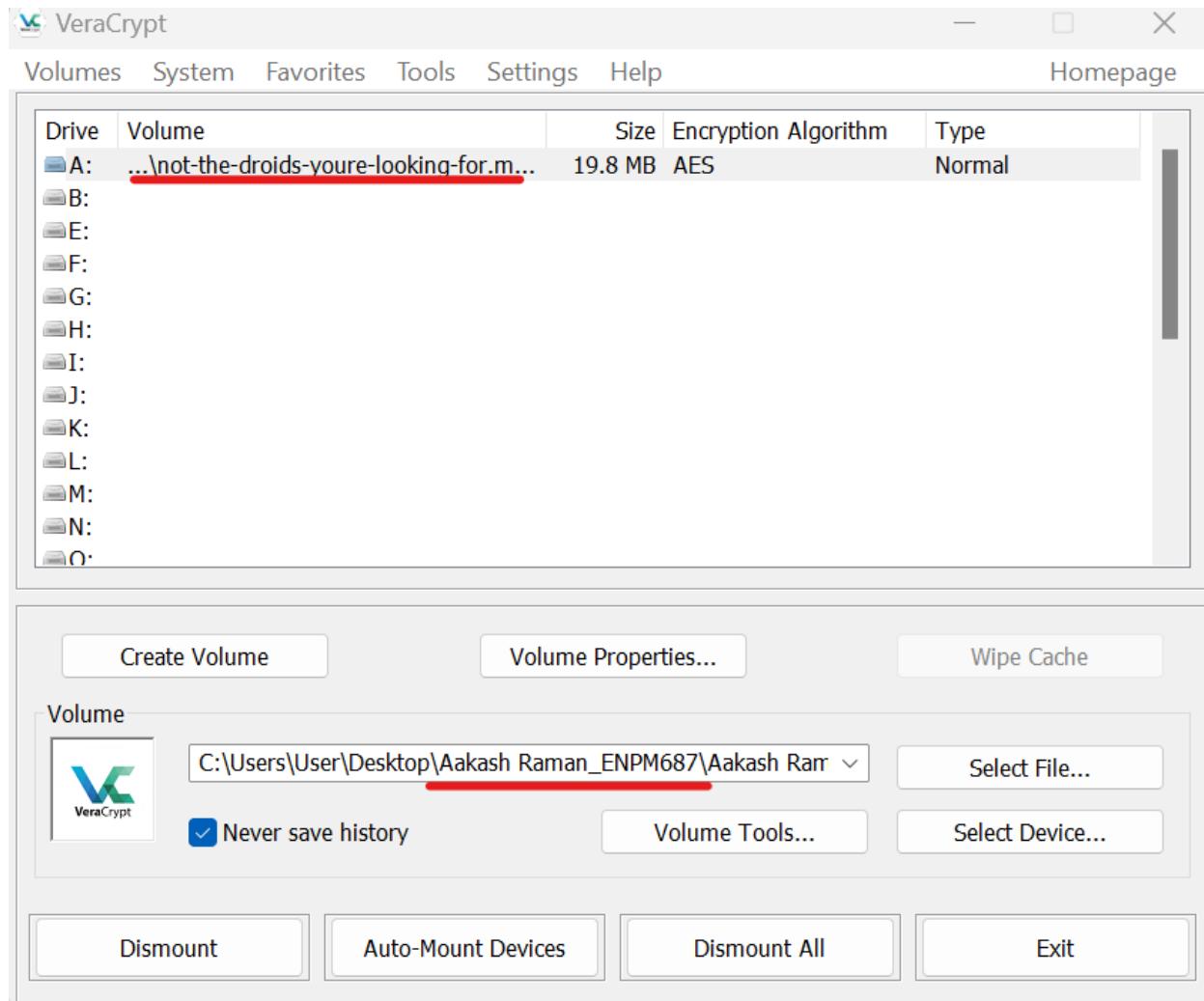
Now that VeraCrypt is installed, I will attempt to decrypt the "**not-the-droids-you're-looking-for.mp3**" file using the "**r2d2**" key. This is shown below.

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-you're-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-you're-looking-for.mp3**" file. All the steps I described are shown in the below screenshots.



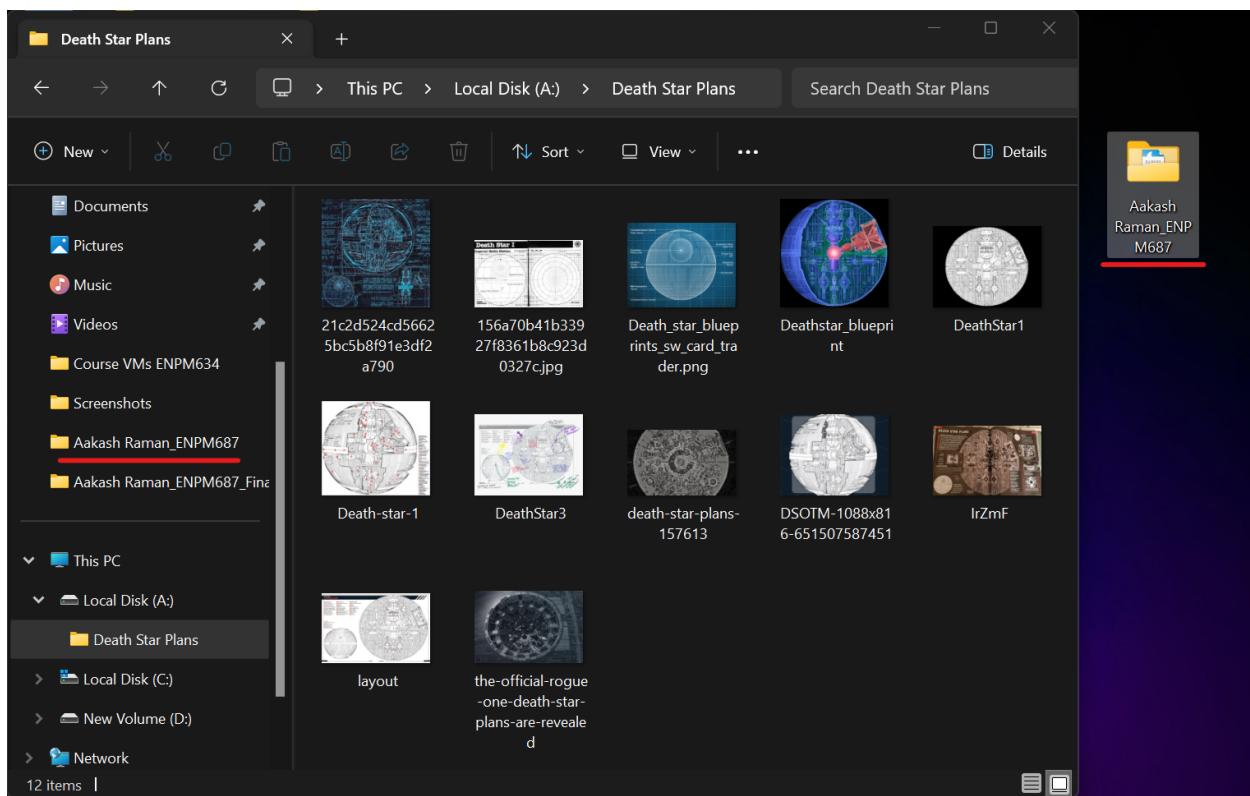
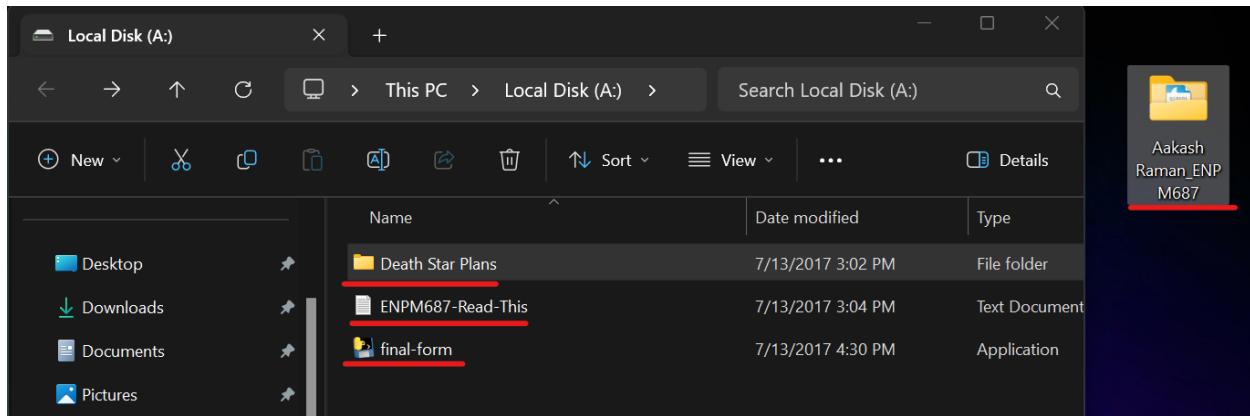




Examining the True Contents of the not-the-droids-youre-looking-for.mp3 file in the A: Drive:

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.



```
ENPM687-Read-This
File Edit View
ENPM687 Final Project
To complete the last part of this project
you will need to determine what the message
sent by final-form.exe is.

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

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 “**obian.exe**” and “**obian2.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.](http://www.umd.edu/We-will-defeat-Darth-Vader)

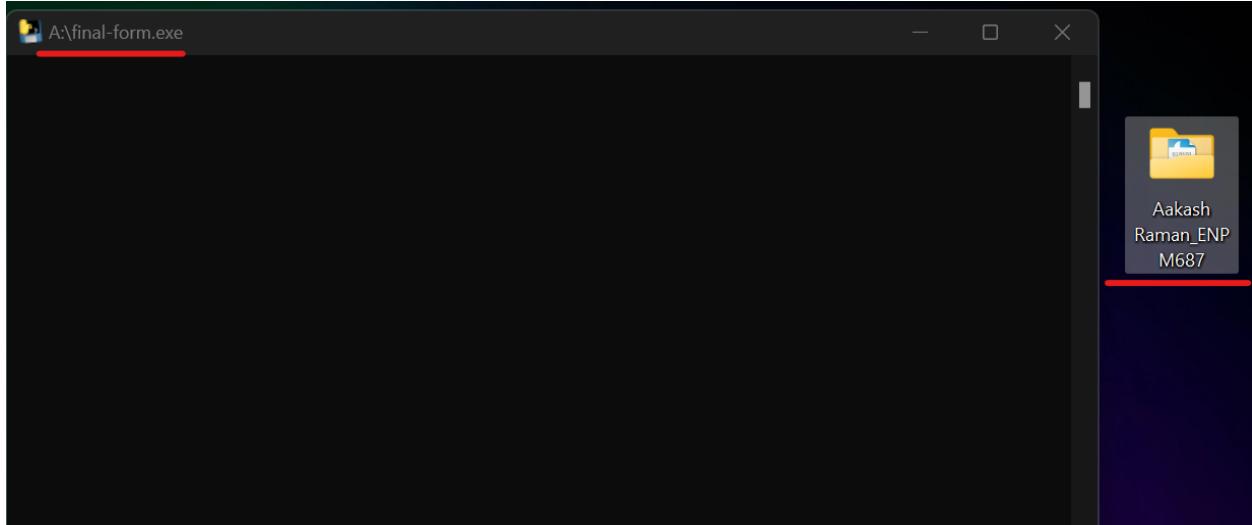
Hence, the final decrypted messages sent by the “final-form.exe” executable are:

- **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”.

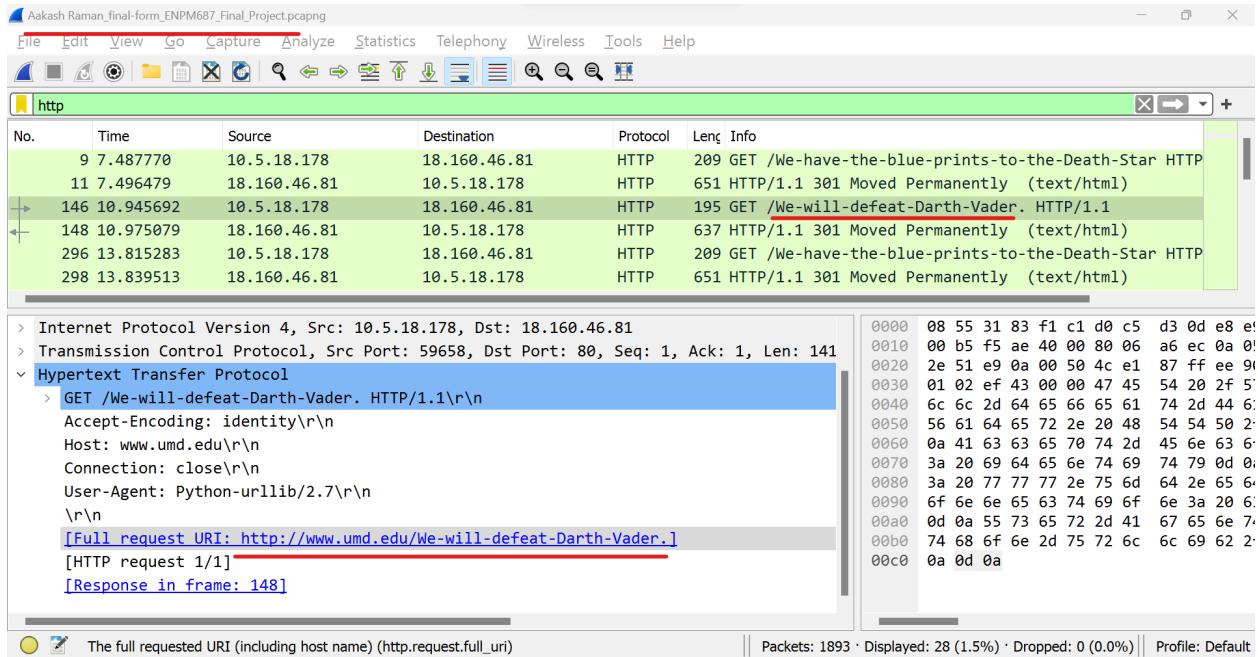
Overall, I believe that the Rebel Malware Writer wants to inform the Rebels that as the Rebels got the blueprints/plans of the Death Star from the previous evidence I found, they will defeat Darth Vader, the strongest Imperial Dark Jedi across the galaxy!

Given below are the screenshots depicting this.



The screenshot shows a Wireshark interface with the following details:

- Packets:** 1893 · Displayed: 28 (1.5%) · Dropped: 0 (0.0%)
- Selected Packet:** No. 9 7.487770 10.5.18.178 → 18.160.46.81 HTTP 209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP
- HTTP Headers:**
 - Accept-Encoding: identity\r\n
 - Host: www.umd.edu\r\n
 - Connection: close\r\n
 - User-Agent: Python-urllib/2.7\r\n
- Full Request URI:** http://www.umd.edu/We-have-the-blue-prints-to-the-Death-Star
- Response in frame:** 11



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.

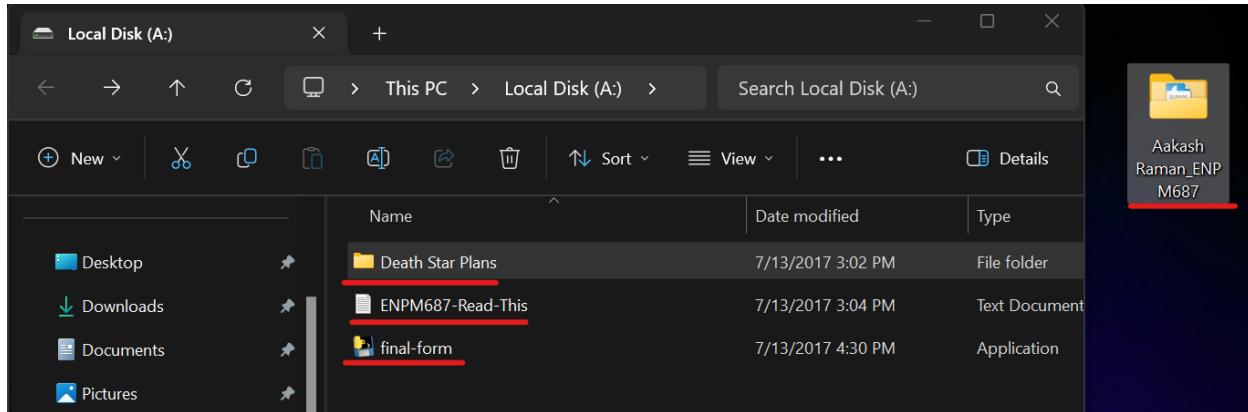
- Never keep traces of the software you use in the form of pictures. One must try to lower the chances of figuring out the exact manner and use of any software. The Images folder in Autopsy told me exactly that the Rebel Malware Writer used VeraCrypt because there were pictures of the tutorial given 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-you're-looking-for.mp3**" file using the "**r2d2**" key in the VeraCrypt software.



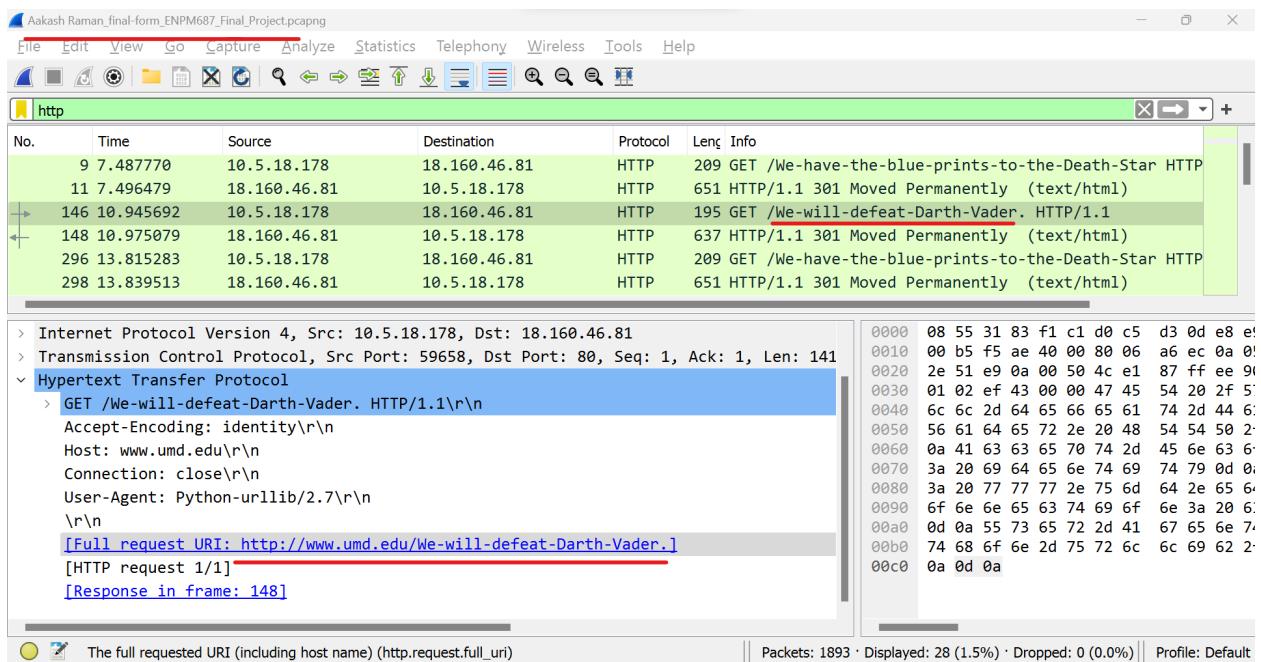
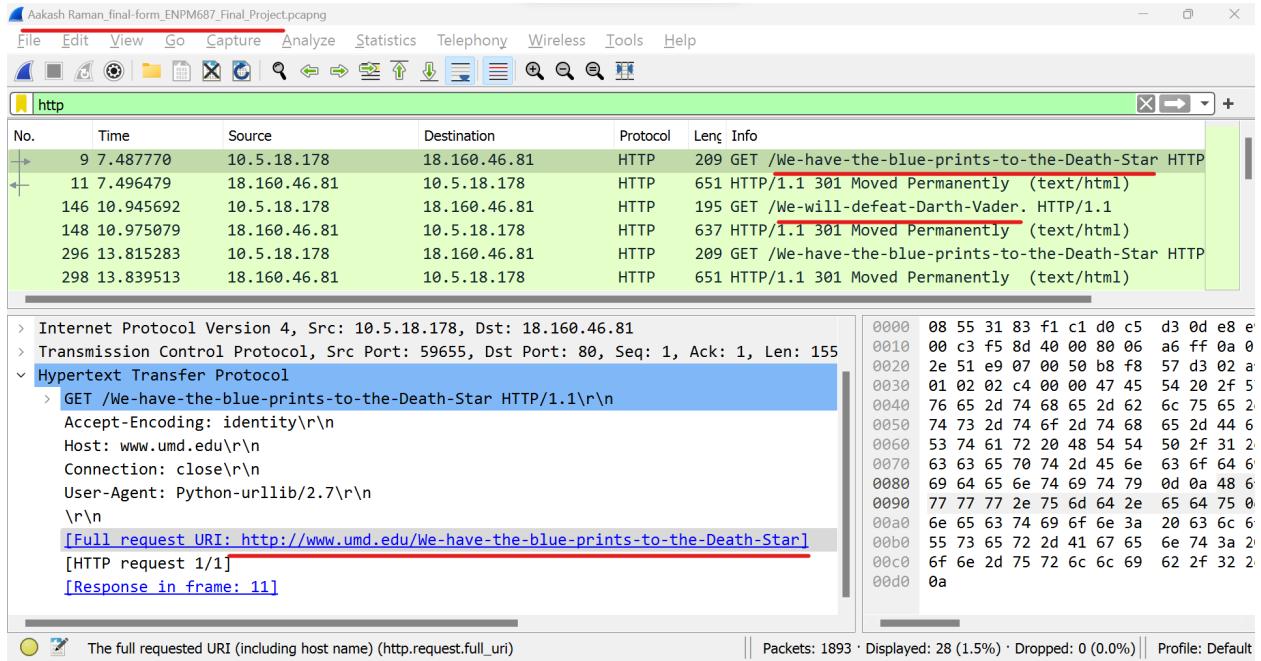
2. Determine what the message contained inside of the final malware is.

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 “**cJkMiBpcyB0aGUga2V5**” 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**”, “**cJkMiBpcyB0aGUga2V5**” 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 “**cJkMiBpcyB0aGUga2V5**” 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.mozilla.org/viewtopic.php?p=12536687>

[2] <https://gchq.github.io/CyberChef/>