Forensic Investigation and Analysis

Investigation for Imperial Forces

Version 1.0

12/11/2020

Author: Kevin Mukam

Email: kevinmukam@gmail.com

Course: Digital Forensics & Incident Response

Table of Contents

1. Introduction **6**

2. Summary of Information **7**

2.1 Functional Description 7
2.2 Tools Used in the Process 7
2.3 Goals 8

3. Procedure **9**

3.1 Virtual Disk Analysis 9
3.1.1 Web History 9
3.1.2 Web Downloads 10
3.1.3 Run Programs 11
3.1.4 Recent Documents 12
3.2 Network Analysis 13
3.3 Additional Research 19
3.3.1 Procexp.exe 19
3.3.2 Death Star Plans 20
3.4 Final Form 21
3.4.1 Extraction 21
3.4.2 Decryption 22
3.4.3 Network Analysis 23

4. Repository **25**

5. Recommendations **26**

Appendix A: Challenges **27**

Appendix B: Acronyms **28**

Appendix C: Glossary **29**

Appendix D: Referenced Documents **30**

Appendix E: Approvals **31**

List of Figures

Figure 1 – Malware Writer’s Disk Image 7
Figure 2 – Attacker’s Image Search 10
Figure 3 – VeraCrypt Download Location 10
Figure 4 – Some of the Programs Ran on July 13, 2017 11
Figure 5 - Virus Total Result for obiwan.exe 13
Figure 6 - Virus Total Result for obiwan2.exe 14
Figure 7 – Obtaining My IP Address 14
Figure 8 - Packets Captured in Wireshark 15
Figure 9 – Packet Details for the GET Request 16
Figure 10 – Packet Details for the HTTP Response 16
Figure 11 – Packet Details for the Second GET Request 17
Figure 12 – Response to the base64 Request 18
Figure 13 – Decoded Message from Packet Capture 18
Figure 14 – Process Explorer Folder 19
Figure 15 – Procexp.exe Analysis on Virus Total 20
Figure 16 – Suspected File 20
Figure 17 – Death Star Plans 21
Figure 18 - Encrypted File Hidden in Plain Sight 22
Figure 19 – Decrypted Files with VeraCrypt 22
Figure 20 - Network Analysis of final-form.exe 23
Figure 21 – HTTP Stream 24

List of Tables

Table 1 - Acronyms 28

Table 2 - Glossary 29

Table 3 - Referenced Documents 30

Table 4 - Approvals 31

1. Introduction

As we evolve in technology, computer systems become stronger, faster, more reliable, more secure and at the same time, attackers become more sophisticated and find techniques to go around the security measures put in place by Engineers. Attackers have developed methods for breaching defenses, exploiting weaknesses in computer systems, thereby causing terrible damage in various parts of the society, such as financial systems, institutional systems, healthcare, etc.

The Imperial Army has come into possession of an image of a hard drive belonging to a rebel scum malware writer. Their codes have plagued the army's computers in the past, and they are using these codes to send messages around the world. As the Imperial Forces' best forensics analyst, I have been tasked to analyze the image of the rebel malware writer's hard drive. The deliverables contain the final version of the malware writer's malware, the message contained inside the final malware, other artifacts/items relevant to the investigation that could help the Imperial Forces in future.

In order to successfully conduct my investigation and analysis, the Imperial Force has provided me with the most recent software such as Autopsy, Wireshark, Windows Virtual Machines and other tools which will be discussed in the document.

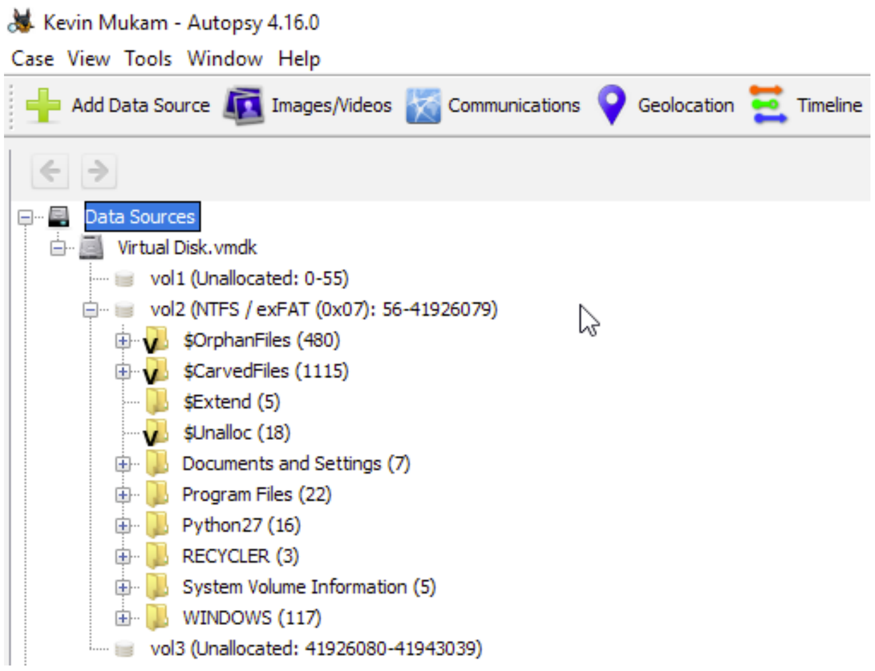
2. Summary of Information

The captured system is a disk image. This is a computer file containing the contents and structure of the attacker's hard drive. The name of the disk image captured is "Virtual Disk.vmdk" (where vmdk is short for Virtual Machine Disk).

Somewhere in the disk image is the malware writer's malware, with other relevant information that could lead me to determine what the attacker was doing and what tools he was using.

2.1 Functional Description

The data in the disk image is stored in an NTFS (New Technology File System) type. NTFS is the file system that Windows operating systems uses for storing and retrieving files on a hard disk. It uses a tree directory scheme to keep track of files. The hard disk is divided into partitions of the total disk space. Every partition will be analyzed to try to detect the malware writer's actions. Each file in the hard disk holds, along with its data content, a metadata section which contains a description of its attributes.



```
Data Sources
  Virtual Disk.vmdk
    vol1 (Unallocated: 0-55)
    vol2 (NTFS / exFAT (0x07): 56-41926079)
      $OrphanFiles (480)
      $CarvedFiles (1115)
      $Extend (5)
      $Unalloc (18)
      Documents and Settings (7)
      Program Files (22)
      Python27 (16)
      RECYCLER (3)
      System Volume Information (5)
      WINDOWS (117)
    vol3 (Unallocated: 41926080-41943039)
```

Figure 1 – Malware Writer's Disk Image

2.2 Tools Used in the Process

VMware Fusion: It is a line of Desktop Hypervisor products designed for modern developers, IT professionals and everyone that wants to run virtual machines. VMware Fusion supports

Windows operating system, which is the system where other software such as Autopsy and Wireshark are used to track the attacker's moves.

Autopsy: Autopsy is a computer software that makes it simpler to deploy many open source programs and plugins used in The Sleuth Kit (a library and collection of Unix and Windows-based utilities for extracting data from disk drives to facilitate the forensic analysis of computer systems). The graphical user interface displays the results from the forensic search of the underlying volume making it easier for investigators to flag pertinent sections of data.

Wireshark: Wireshark is the world's most widely used network protocol analyzer. It lets you see what is happening on your network at a microscopic level and it is the standard across many enterprises, government agencies, and educational institutions. It offers features such as live packet capture, offline analysis, decryption support, deep inspection of protocols, and so many important tools for network forensics.

Online Tool: I used the website http://virustotal.com to analyze files enabling the detection of viruses, worms, trojans and other kinds of malicious content using antivirus engines and web scanners. VirusTotal inspects items with over 70 antivirus scanners and a myriad of tools to extract signals from studied content.

VeraCrypt: It is a free open source disk encryption software for Windows, Mac OSX and Linux. It creates a virtual encrypted disk within a file and mounts it as a real disk. It provides hidden volume (steganography) and hidden operating system. It is possible to encrypt an entire partition or storage device with VeraCrypt.

Command Line (cmd): text interface for the computer. It is a program that takes in commands, which it passes on to the computer's operating system to run.

2.3 Goals

- Finding the final version of the malware writer's malware.
- Determining the message contained inside the final malware.
- Finding artifacts relevant to the investigation that will aid the Imperial Force.
- Develop challenges and difficulties encountered during the investigation.

3. Procedure

Autopsy has a section called “Results” which provides several information that are used in the investigation process. There is an “Extracted Content” subsection where many ingest modules place EXIF data, Web History, Running Programs and other interesting tabs that can be used for analysis.

After going through the Extracted Content section, I came up with a timeline of the attacker’s actions. I analyzed his Web history searches, Web downloads, Running programs, and Recent documents.

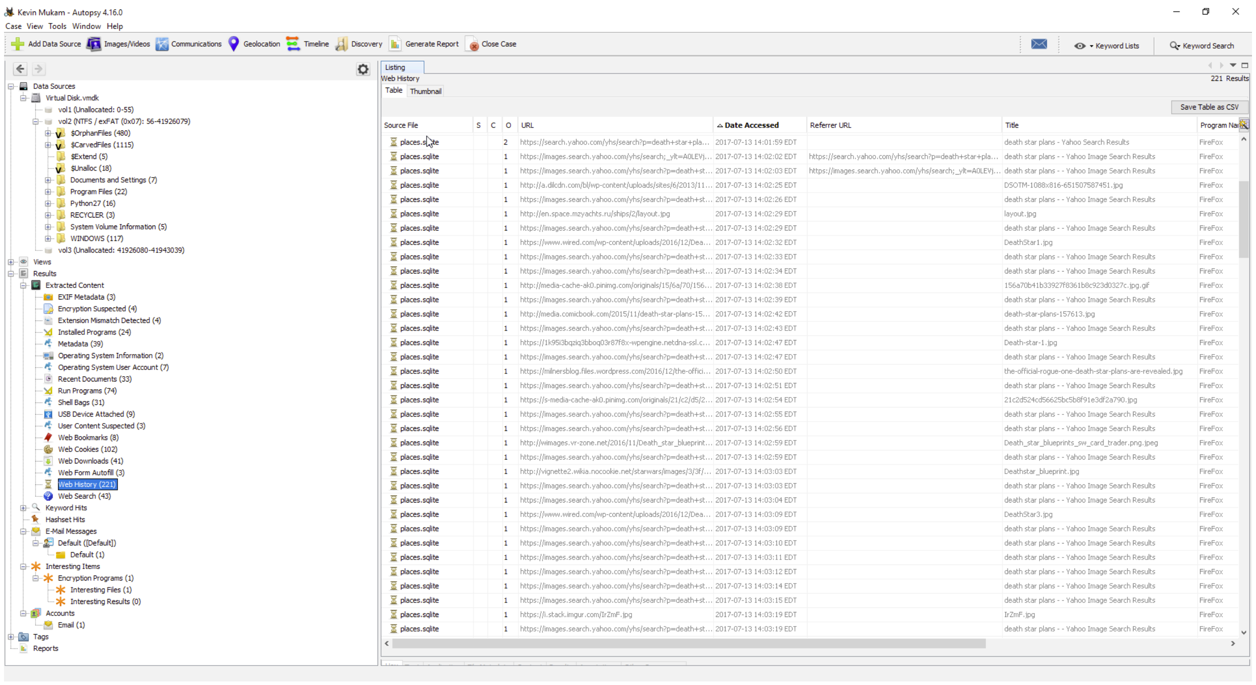
3.1 Virtual Disk Analysis

There are several actions that took place in the attacker’s computer on July 13, 2017. They are summarized below:

3.1.1 Web History

July 13, 2017 13:45
The attacker accessed the VeraCrypt download website. VeraCrypt is a software which is used to hide files within files. Encrypted volume solutions like VeraCrypt allow users to have a false bottom for the volume. I suspected that the attacker could be downloading VeraCrypt to hide his malware files within other files, so I took note of this event.

July 13, 2017
From 14:01:59 to 14:03:24
The attacker made 32 image searches on yahoo titled “Death Star Plans” on his Firefox browser. This is a very bizarre name to search on Google, and also so many searches in such a short time sounded like a red flag to me. I took note of this event and I extracted the images that the attacker downloaded.



Source File	S	C	O	URL	Date Accessed	Referrer URL	Title	Program Na
places.sqlite			2	https://search.yahoo.com/yhs/search?p=death+star+pla...	2017-07-13 14:01:59 EDT		death star plans - Yahoo Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search;_ylt=A0LEVj...	2017-07-13 14:02:02 EDT	https://search.yahoo.com/yhs/search?p=death+star+pla...	death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:03 EDT	https://images.search.yahoo.com/yhs/search;_ylt=A0LEVj...	death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://a.dilcdn.com/bl/wp-content/uploads/sites/6/2013/11...	2017-07-13 14:02:25 EDT		DSOTM-1088x816-651507587451.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:26 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://en.space.mzyachts.ru/ships/2/layout.jpg	2017-07-13 14:02:29 EDT		layout.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:29 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://www.wired.com/wp-content/uploads/2016/12/Dea...	2017-07-13 14:02:32 EDT		DeathStar1.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:33 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:34 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://media-cache-ak0.pinimg.com/originals/15/6a/70/156...	2017-07-13 14:02:38 EDT		156a70b41b33927f8361b8c923d0327c.jpg.gif	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:39 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://media.comicbook.com/2015/11/death-star-plans-15...	2017-07-13 14:02:42 EDT		death-star-plans-157613.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:43 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://1k95i3bqziq3bboq03r87f8x-wpengine.netdna-ssl.c...	2017-07-13 14:02:47 EDT		Death-star-1.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:47 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://milnersblog.files.wordpress.com/2016/12/the-offici...	2017-07-13 14:02:50 EDT		the-official-rogue-one-death-star-plans-are-revealed.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:51 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://s-media-cache-ak0.pinimg.com/originals/21/c2/d5/2...	2017-07-13 14:02:54 EDT		21c2d524cd56625bc5b8f91e3df2a790.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:55 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:56 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://wimages.vr-zone.net/2016/11/Death_star_blueprint...	2017-07-13 14:02:59 EDT		Death_star_blueprints_sw_card_trader.png.jpeg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:02:59 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	http://vignette2.wikia.nocookie.net/starwars/images/3/3f/...	2017-07-13 14:03:03 EDT		Deathstar_blueprint.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:03 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:04 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://www.wired.com/wp-content/uploads/2016/12/Dea...	2017-07-13 14:03:09 EDT		DeathStar3.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:09 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:10 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:11 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:12 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:14 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:15 EDT		death star plans - - Yahoo Image Search Results	FireFox
places.sqlite			1	https://i.stack.imgur.com/IrZmF.jpg	2017-07-13 14:03:19 EDT		IrZmF.jpg	FireFox
places.sqlite			1	https://images.search.yahoo.com/yhs/search?p=death+st...	2017-07-13 14:03:19 EDT		death star plans - - Yahoo Image Search Results	FireFox

Figure 2 – Attacker's Image Search

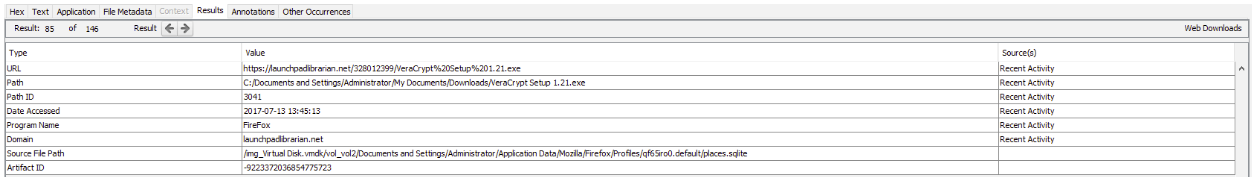
3.1.2 Web Downloads

July 13, 2017 13:45:13

The attacker downloaded VeraCrypt using Firefox browser.

https://launchpadlibrarian.net/328912399/VeraCrypt%20Setup%201.21exe

The source file is located in the folder: /img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/Application_Data/Mozilla/Firefox/Profiles/qf65iro0.default/places.sqlite.



Type	Value	Source(s)
URL	https://launchpadlibrarian.net/328012399/VeraCrypt%20Setup%201.21.exe	Recent Activity
Path	C:/Documents and Settings/Administrator/My Documents/Downloads/VeraCrypt Setup 1.21.exe	Recent Activity
Path ID	3041	Recent Activity
Date Accessed	2017-07-13 13:45:13	Recent Activity
Program Name	FireFox	Recent Activity
Domain	launchpadlibrarian.net	Recent Activity
Source File Path	/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/Application Data/Mozilla/Firefox/Profiles/qf65iro0.default/places.sqlite	
Artifact ID	-9223372036854775723	

Figure 3 – VeraCrypt Download Location

July 13, 2017

From 14:02:25 to 14:03:19

The attacker downloaded 12 images from his "Death star plans" image search.

All the downloaded images are in: /img_Virtual Disk.vmdk/vol_vol2/Documents and_Settings/Administrator/Application_Data/Mozilla/Firefox/Profiles/qf65iro0.default/places.s qlite

I extracted the downloaded images but I couldn't obtain relevant information for a potential malware from those images.

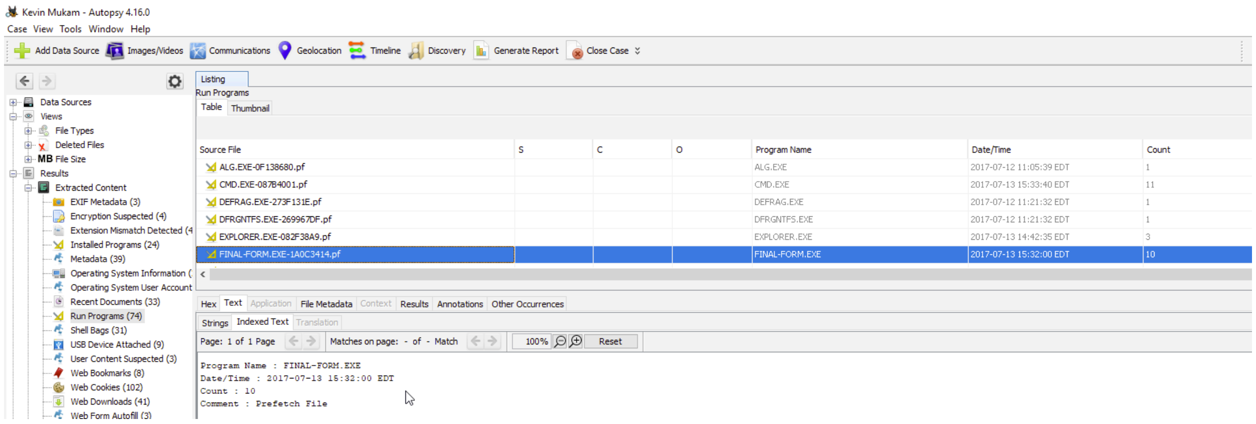
3.1.3 Run Programs

July 13, 2017 15:32:00

The attacker ran the program FINAL-FORM.EXE-1A0C3414.pf
MD5 hash: 3f15b917bbe1255c2f758bba3f1d0c4b
Path: /img_Virtual Disk.vmdk/vol_vol2/WINDOWS/Prefetch/FINAL-FORM.EXE-1A0C3414.pf

What caught my attention was the name of the program. The Imperial Force asked me to find the final form of the malware writer's malware. Going into more detail of this file, under the text tab, and indexed text, there is the following path

Text \DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\ADMINISTRATOR\MY DOCUMENTS\CODE\DIST\FINAL-FORM.EXE



Source File	S	C	O	Program Name	Date/Time	Count
ALG.EXE-0F138680.pf				ALG.EXE	2017-07-12 11:05:39 EDT	1
CMD.EXE-087B4001.pf				CMD.EXE	2017-07-13 15:33:40 EDT	11
DEFRAG.EXE-273F131E.pf				DEFRAG.EXE	2017-07-12 11:21:32 EDT	1
DFRGNTFS.EXE-269967DF.pf				DFRGNTFS.EXE	2017-07-12 11:21:32 EDT	1
EXPLORER.EXE-082F38A9.pf				EXPLORER.EXE	2017-07-13 14:42:35 EDT	3
FINAL-FORM.EXE-1A0C3414.pf				FINAL-FORM.EXE	2017-07-13 15:32:00 EDT	10

```
Program Name : FINAL-FORM.EXE
Date/Time : 2017-07-13 15:32:00 EDT
Count : 10
Comment : Prefetch File
```

Figure 4 – Some of the Programs Ran on July 13, 2017

July 13, 2017 15:11:27

The attacker ran the program OBIWAN2.EXE-1B80EA86.pf
MD5: 6780bd4cf91921405a4419a23301c081
Path: /img_Virtual Disk.vmdk/vol_vol2/WINDOWS/Prefetch/OBIWAN2.EXE-1B80EA86.pf

During my previous experience as a Forensic Analyst at University of Maryland, I had come across a similar file with the name obiwan.exe. Therefore, this was a major red flag when I came across the running program called obiwan2.exe. I extracted the executable file which I would later use to perform some more advanced analysis.

July 13, 2017 13:58:18
VeraCrypt was running with the source file VERACRYPT FORMAT.EXE-2F2B3235.pf
Path: /img_Virtual Disk.vmdk/vol_vol2/WINDOWS/Prefetch/VERACRYPT FORMAT.EXE-2F2B3235.pf

VeraCrypt in its own is not a malicious software. However, at this moment in time I suspected that the attacker was using VeraCrypt to hide other personal data, so I had to take note of this event and I later on ran VeraCrypt to check some of these folders.

Note:
The .pf file extension was developed by Microsoft and it refers to files that Windows Prefetcher created. Prefetcher is responsible for speeding the boot process of a certain application. For instance, when a user opens a program, it does not instantly appear on screen and the user cannot use it at once because the application is still loading. During the loading process, the system relies on finding .pf files and accesses these files in order for the boot process to speed up. These files contain trace records and are always essential when accessing any application.

3.1.4 Recent Documents

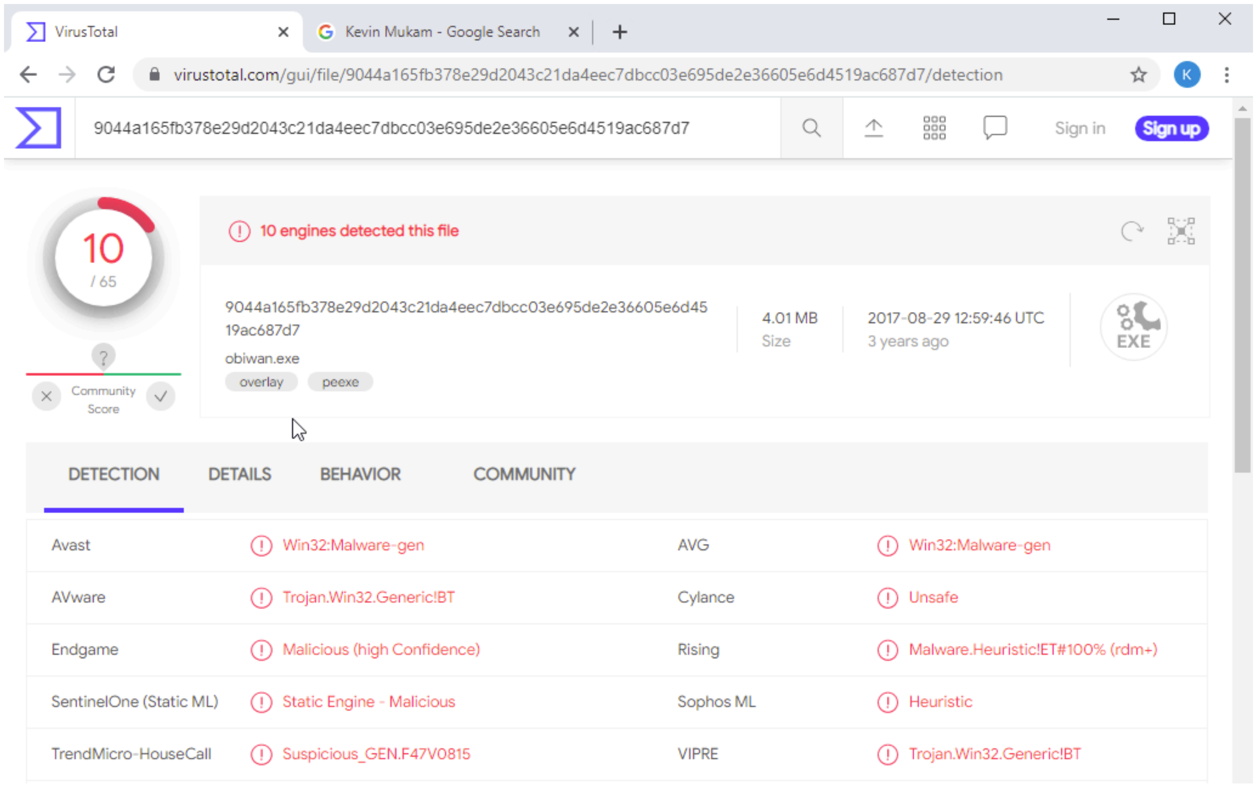
During one of my previous forensics investigations, I successfully detected malware by going through the recent documents folder, so I analyzed this folder to look at any relevant information I could find that could help me in the process.

July 13, 2017 14:02:59:	Death_star_blueprints_sw_card_trader.png.lnk
July 13, 2017 14:02:25:	Death Star Plans.lnk
July 13, 2017 14:02:47:	Death-star-1.lnk
July 13, 2017 14:02:42:	Death-star-plans-157613.lnk
July 13, 2017 14:40:48:	final-form.lnk
July 13, 2017 14:36:28:	obiwan.lnk
July 13, 2017 14:38:52:	obiwan2.lnk

An LNK file is a shortcut or "link" used by Windows as a reference to an original file, folder, or application similar to an alias on the Macintosh platform. It contains the shortcut target type, location, and filename as well as the program that opens the target file and an optional shortcut key.

3.2 Network Analysis

After analyzing the various files and going through all the folders in the virtual disk, I extracted the executable files obiwan.exe and obiwan2.exe to my desktop. I ran both files on virustotal.com and the files were red flagged. The first file (obiwan.exe) contains trojan, and is deemed suspicious and by many trusted antivirus scanners.


VirusTotal
Kevin Mukam - Google Search
virustotal.com/gui/file/9044a165fb378e29d2043c21da4eec7dbcc03e695de2e36605e6d4519ac687d7/detection
9044a165fb378e29d2043c21da4eec7dbcc03e695de2e36605e6d4519ac687d7
Sign in
Sign up
10
/ 65
Community Score
10 engines detected this file
9044a165fb378e29d2043c21da4eec7dbcc03e695de2e36605e6d4519ac687d7
obiwan.exe
overlay
peexe
4.01 MB
Size
2017-08-29 12:59:46 UTC
3 years ago
EXE
DETECTION
DETAILS
BEHAVIOR
COMMUNITY

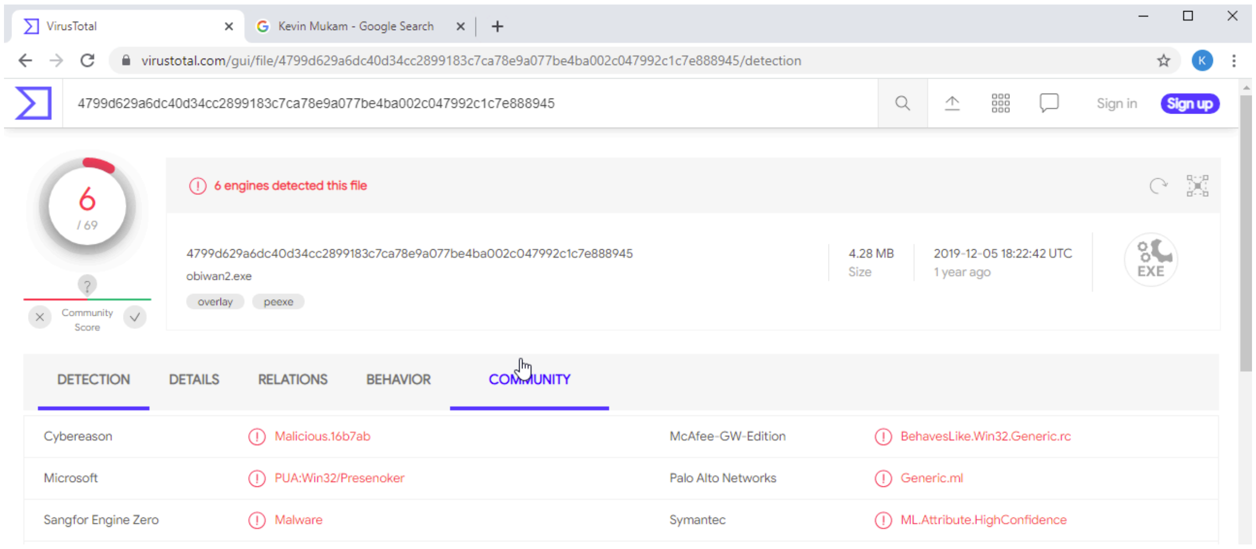
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
AVware	Trojan.Win32.Generic!BT	Cylance	Unsafe
Endgame	Malicious (high Confidence)	Rising	Malware.Heuristic!ET#100% (rdm+)
SentinelOne (Static ML)	Static Engine - Malicious	Sophos ML	Heuristic
TrendMicro-HouseCall	Suspicious_GEN.F47V0815	VIPRE	Trojan.Win32.Generic!BT

Figure 5 - Virus Total Result for obiwan.exe

Some properties are also provided by virustotal.com such as the MD5 hash value which in this case is 97c90c038e2b060acc2354b9eb5f69d3.

I repeated the same process for obiwan2.exe and found out that it is also a malware file (see figure 6) with MD5 hash value of cb183dc18d160ec8ca1651e717182a04.

I ran both executable files on the command line and I launched Wireshark to perform a network analysis and determine what the executable file was doing. By analyzing the packet, the idea is to intercept and log traffic that passes over the computer network. When traffic is captured, the contents of packets are recorded. Captured information is decoded into a human-readable format. Protocol analyzers vary in their abilities to display and analyze data.


VirusTotal
Kevin Mukam - Google Search
virustotal.com/gui/file/4799d629a6dc40d34cc2899183c7ca78e9a077be4ba002c047992c1c7e888945/detection
Sign in
Sign up
6
/ 69
Community Score
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY

6 engines detected this file

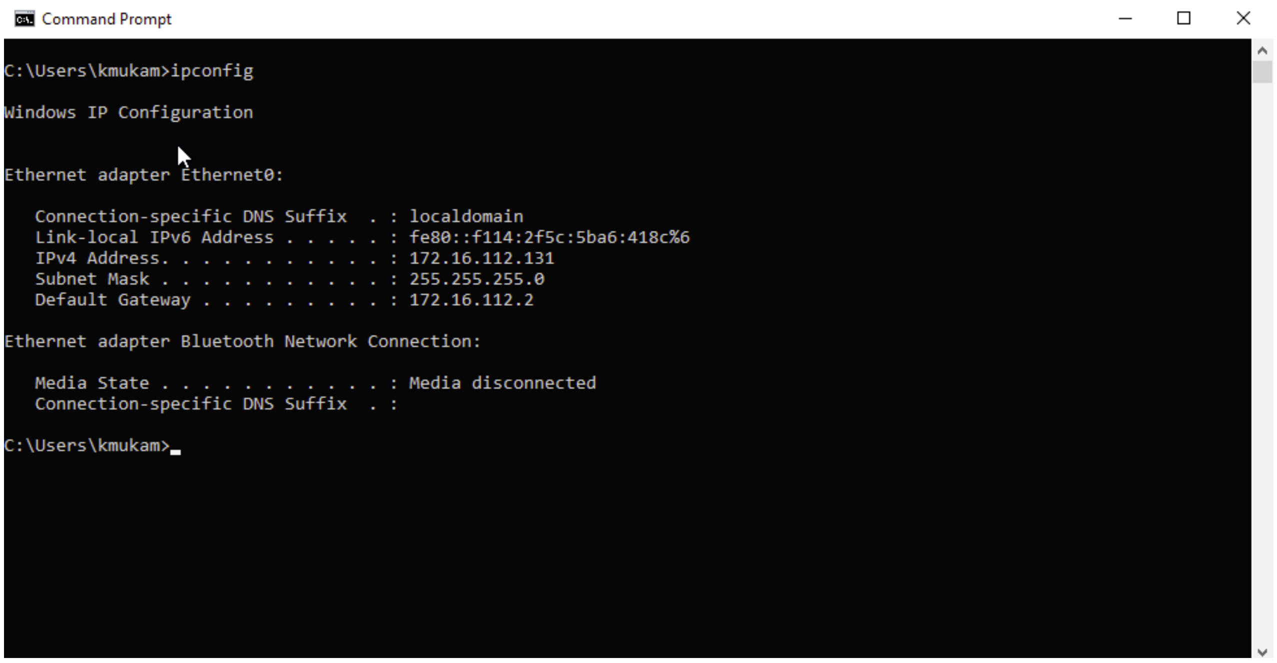
4799d629a6dc40d34cc2899183c7ca78e9a077be4ba002c047992c1c7e888945
obiwan2.exe
overlay peexe

4.28 MB Size
2019-12-05 18:22:42 UTC 1 year ago
EXE

Cybereason	Malicious.16b7ab	McAfee-GW-Edition	BehavesLike.Win32.Generic.rc
Microsoft	PUA:Win32/Presenoker	Palo Alto Networks	Generic.ml
Sangfor Engine Zero	Malware	Symantec	ML.Attribute.HighConfidence

Figure 6 - Virus Total Result for obiwan2.exe

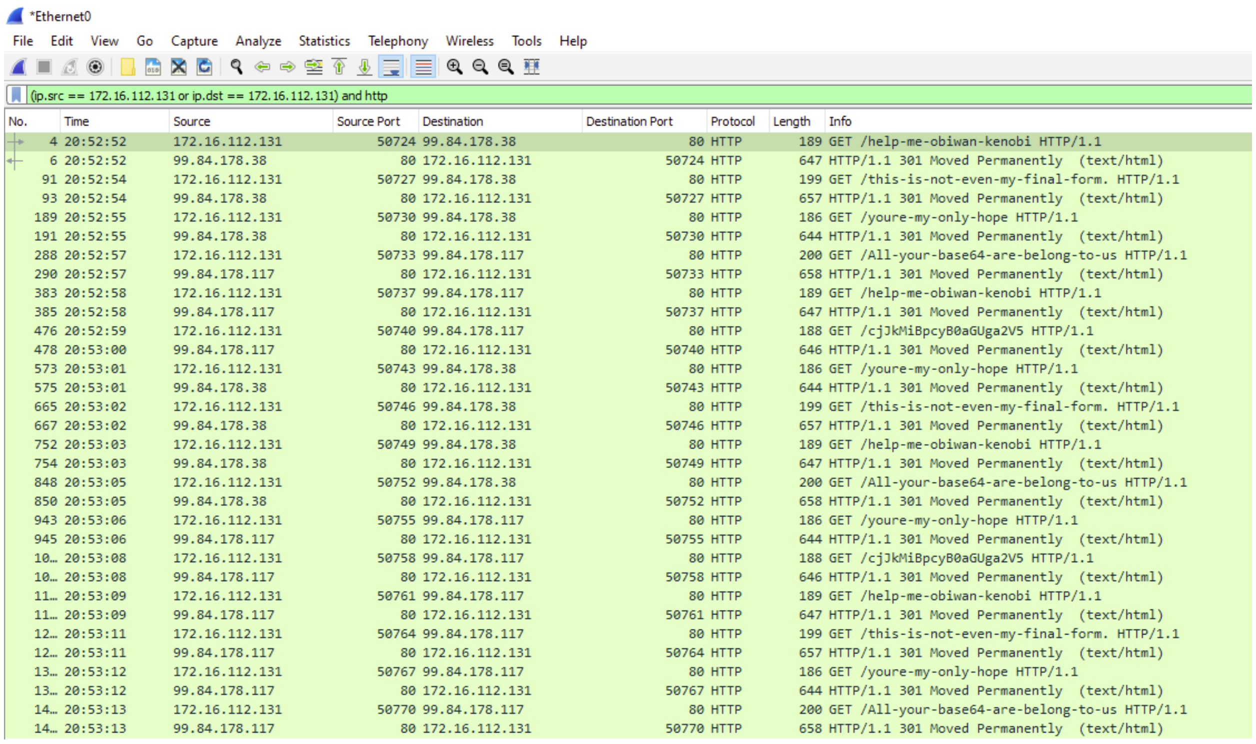
The filters in Wireshark are very important in the analysis process because with the tens of thousands of packets captured, filters remove unnecessary packets from the list and display the packets we want to see. Display Filters are used for filtering through already captured packets. I used the command line with the command “ipconfig” to obtain my virtual machine’s IP address (172.16.112.131) which I would later use to filter packets to and from the VM.



```
C:\Users\kmukam>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::f114:2f5c:5ba6:418c%6
   IPv4 Address. . . . . . . . . . . : 172.16.112.131
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.112.2

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\kmukam>_
```

Figure 7 – Obtaining My IP Address

With my IP, I used the filters ip.src == 172.16.112.131 or ip.dst == 172.16.112.131 to filter the network traffic to and from my VM's IP address. I also used the http keyword to filter HTTP traffic only.


*Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

(ip.src == 172.16.112.131 or ip.dst == 172.16.112.131) and http

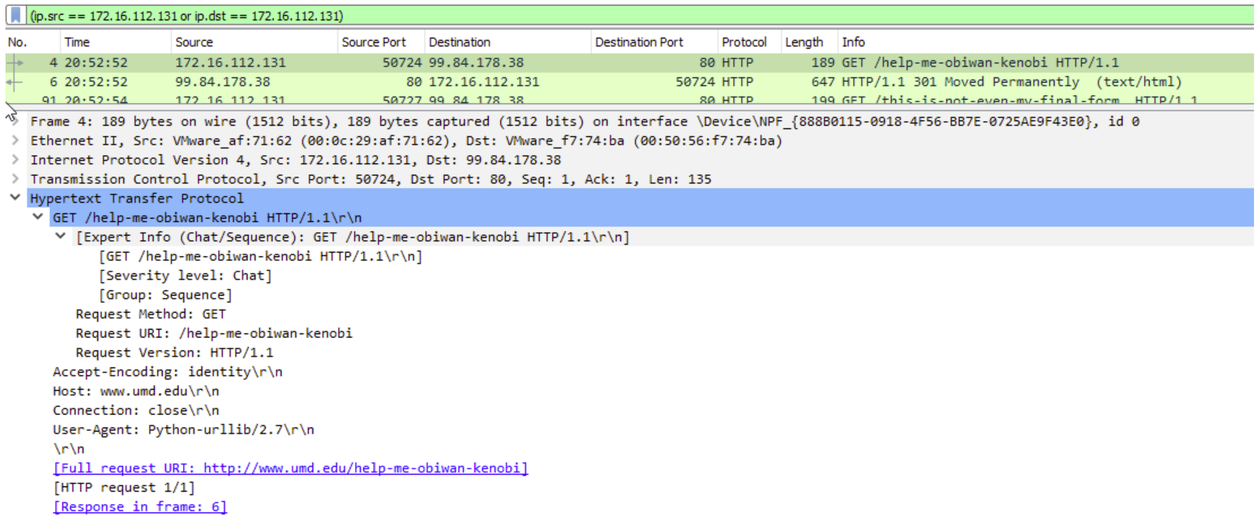
No.	Time	Source	Source Port	Destination	Destination Port	Protocol	Length	Info
4	20:52:52	172.16.112.131	50724	99.84.178.38	80	HTTP	189	GET /help-me-obiwan-kenobi HTTP/1.1
6	20:52:52	99.84.178.38	80	172.16.112.131	50724	HTTP	647	HTTP/1.1 301 Moved Permanently (text/html)
91	20:52:54	172.16.112.131	50727	99.84.178.38	80	HTTP	199	GET /this-is-not-even-my-final-form. HTTP/1.1
93	20:52:54	99.84.178.38	80	172.16.112.131	50727	HTTP	657	HTTP/1.1 301 Moved Permanently (text/html)
189	20:52:55	172.16.112.131	50730	99.84.178.38	80	HTTP	186	GET /youre-my-only-hope HTTP/1.1
191	20:52:55	99.84.178.38	80	172.16.112.131	50730	HTTP	644	HTTP/1.1 301 Moved Permanently (text/html)
288	20:52:57	172.16.112.131	50733	99.84.178.117	80	HTTP	200	GET /All-your-base64-are-belong-to-us HTTP/1.1
290	20:52:57	99.84.178.117	80	172.16.112.131	50733	HTTP	658	HTTP/1.1 301 Moved Permanently (text/html)
383	20:52:58	172.16.112.131	50737	99.84.178.117	80	HTTP	189	GET /help-me-obiwan-kenobi HTTP/1.1
385	20:52:58	99.84.178.117	80	172.16.112.131	50737	HTTP	647	HTTP/1.1 301 Moved Permanently (text/html)
476	20:52:59	172.16.112.131	50740	99.84.178.117	80	HTTP	188	GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1
478	20:53:00	99.84.178.117	80	172.16.112.131	50740	HTTP	646	HTTP/1.1 301 Moved Permanently (text/html)
573	20:53:01	172.16.112.131	50743	99.84.178.38	80	HTTP	186	GET /youre-my-only-hope HTTP/1.1
575	20:53:01	99.84.178.38	80	172.16.112.131	50743	HTTP	644	HTTP/1.1 301 Moved Permanently (text/html)
665	20:53:02	172.16.112.131	50746	99.84.178.38	80	HTTP	199	GET /this-is-not-even-my-final-form. HTTP/1.1
667	20:53:02	99.84.178.38	80	172.16.112.131	50746	HTTP	657	HTTP/1.1 301 Moved Permanently (text/html)
752	20:53:03	172.16.112.131	50749	99.84.178.38	80	HTTP	189	GET /help-me-obiwan-kenobi HTTP/1.1
754	20:53:03	99.84.178.38	80	172.16.112.131	50749	HTTP	647	HTTP/1.1 301 Moved Permanently (text/html)
848	20:53:05	172.16.112.131	50752	99.84.178.38	80	HTTP	200	GET /All-your-base64-are-belong-to-us HTTP/1.1
850	20:53:05	99.84.178.38	80	172.16.112.131	50752	HTTP	658	HTTP/1.1 301 Moved Permanently (text/html)
943	20:53:06	172.16.112.131	50755	99.84.178.117	80	HTTP	186	GET /youre-my-only-hope HTTP/1.1
945	20:53:06	99.84.178.117	80	172.16.112.131	50755	HTTP	644	HTTP/1.1 301 Moved Permanently (text/html)
10…	20:53:08	172.16.112.131	50758	99.84.178.117	80	HTTP	188	GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1
10…	20:53:08	99.84.178.117	80	172.16.112.131	50758	HTTP	646	HTTP/1.1 301 Moved Permanently (text/html)
11…	20:53:09	172.16.112.131	50761	99.84.178.117	80	HTTP	189	GET /help-me-obiwan-kenobi HTTP/1.1
11…	20:53:09	99.84.178.117	80	172.16.112.131	50761	HTTP	647	HTTP/1.1 301 Moved Permanently (text/html)
12…	20:53:11	172.16.112.131	50764	99.84.178.117	80	HTTP	199	GET /this-is-not-even-my-final-form. HTTP/1.1
12…	20:53:11	99.84.178.117	80	172.16.112.131	50764	HTTP	657	HTTP/1.1 301 Moved Permanently (text/html)
13…	20:53:12	172.16.112.131	50767	99.84.178.117	80	HTTP	186	GET /youre-my-only-hope HTTP/1.1
13…	20:53:12	99.84.178.117	80	172.16.112.131	50767	HTTP	644	HTTP/1.1 301 Moved Permanently (text/html)
14…	20:53:13	172.16.112.131	50770	99.84.178.117	80	HTTP	200	GET /All-your-base64-are-belong-to-us HTTP/1.1
14…	20:53:13	99.84.178.117	80	172.16.112.131	50770	HTTP	658	HTTP/1.1 301 Moved Permanently (text/html)

Figure 8 - Packets Captured in Wireshark

Looking at these packets captured, it is clear that there is communication going on between the extracted executable files and an external source. The executable files use the HTTP protocol to send a GET request to the destination within the subnet that contains the IP addresses 99.84.178.X. The GET method is used when the browser requests an object, with the requested object identified in the URL field. By clicking on a request packet, I can view the Packet details which will display the time frame, timeline, URL, the source port, destination port, and all other important metadata information (Figure 9). For the GET request, the URL is http://www.umd.edu/help-me-obiwan-kenobi. From here I can see that the malware writer was trying to reach out to an external source asking for help (while the executable file was running).

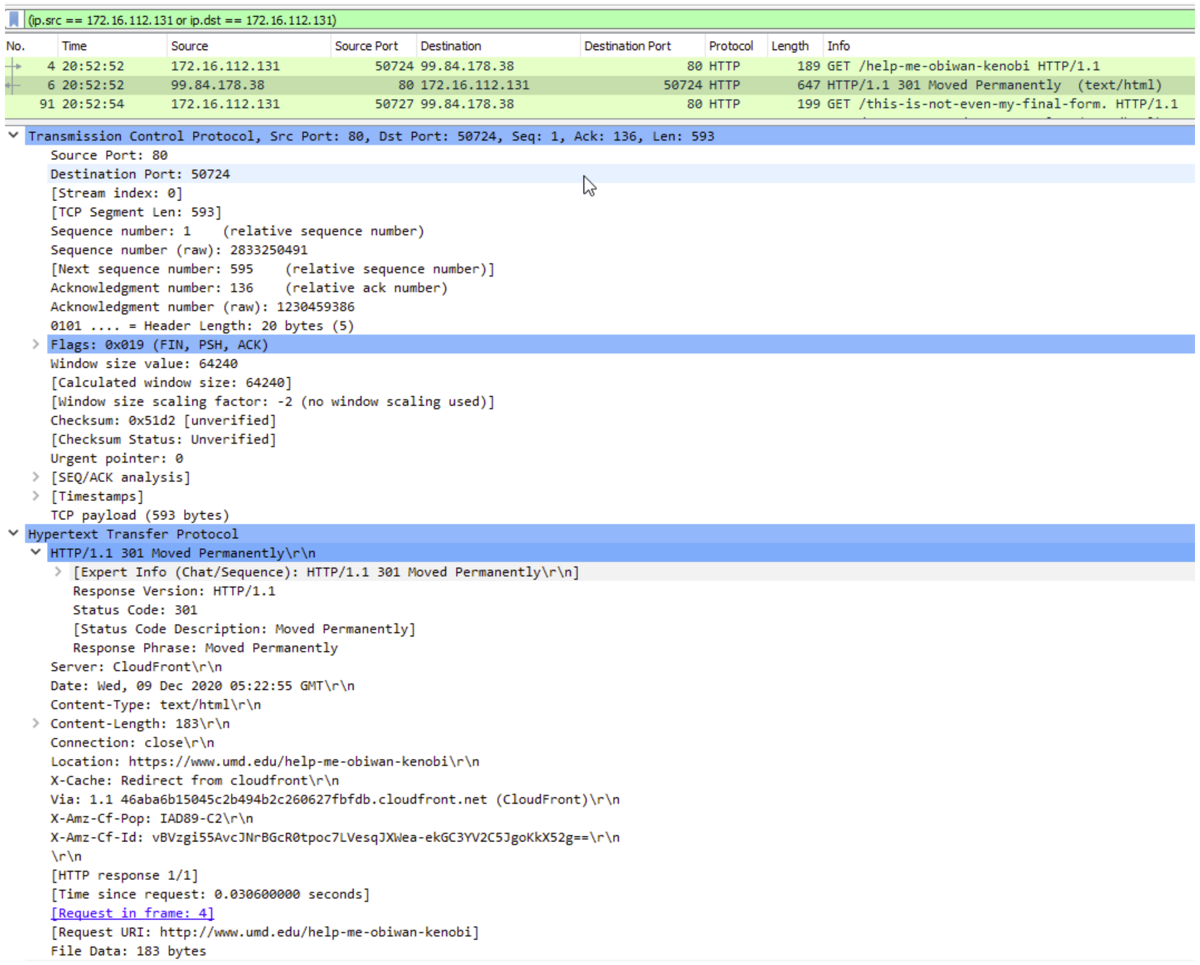
The TCP Layer is responsible for establishing connection between two computers.

After the GET request is sent, a response is sent from the server (figure 10). The status code is 301 which means that the requested object has been permanently moved. The new URL is specified in Location: header of the response message. The client software will automatically retrieve the new URL. The new URL is http://www.umd.edu/help-me-obiwan-kenobi/ which means that this is a loop. The executable file keeps reaching out to the destination at IP address 99.84.178.38 with the same message.



```
(ip.src == 172.16.112.131 or ip.dst == 172.16.112.131)

No.  Time      Source          Source Port Destination     Destination Port Protocol Length Info
   4 20:52:52  172.16.112.131        50724 99.84.178.38                  80 HTTP        189 GET /help-me-obiwan-kenobi HTTP/1.1
   6 20:52:52  99.84.178.38             80 172.16.112.131             50724 HTTP        647 HTTP/1.1 301 Moved Permanently  (text/html)
  91 20:52:54  172.16.112.131        50727 99.84.178.38                  80 HTTP        199 GET /this-is-not-even-my-final-form  HTTP/1.1

> Frame 4: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits) on interface \Device\NPF_{888B0115-0918-4F56-BB7E-0725AE9F43E0}, id 0
> Ethernet II, Src: VMware_af:71:62 (00:0c:29:af:71:62), Dst: VMware_f7:74:ba (00:50:56:f7:74:ba)
> Internet Protocol Version 4, Src: 172.16.112.131, Dst: 99.84.178.38
> Transmission Control Protocol, Src Port: 50724, Dst Port: 80, Seq: 1, Ack: 1, Len: 135
v Hypertext Transfer Protocol
  v GET /help-me-obiwan-kenobi HTTP/1.1\r\n
    v [Expert Info (Chat/Sequence): GET /help-me-obiwan-kenobi HTTP/1.1\r\n]
        [GET /help-me-obiwan-kenobi HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /help-me-obiwan-kenobi
      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/help-me-obiwan-kenobi]
    [HTTP request 1/1]
    [Response in frame: 6]
```

Figure 9 – Packet Details for the GET Request

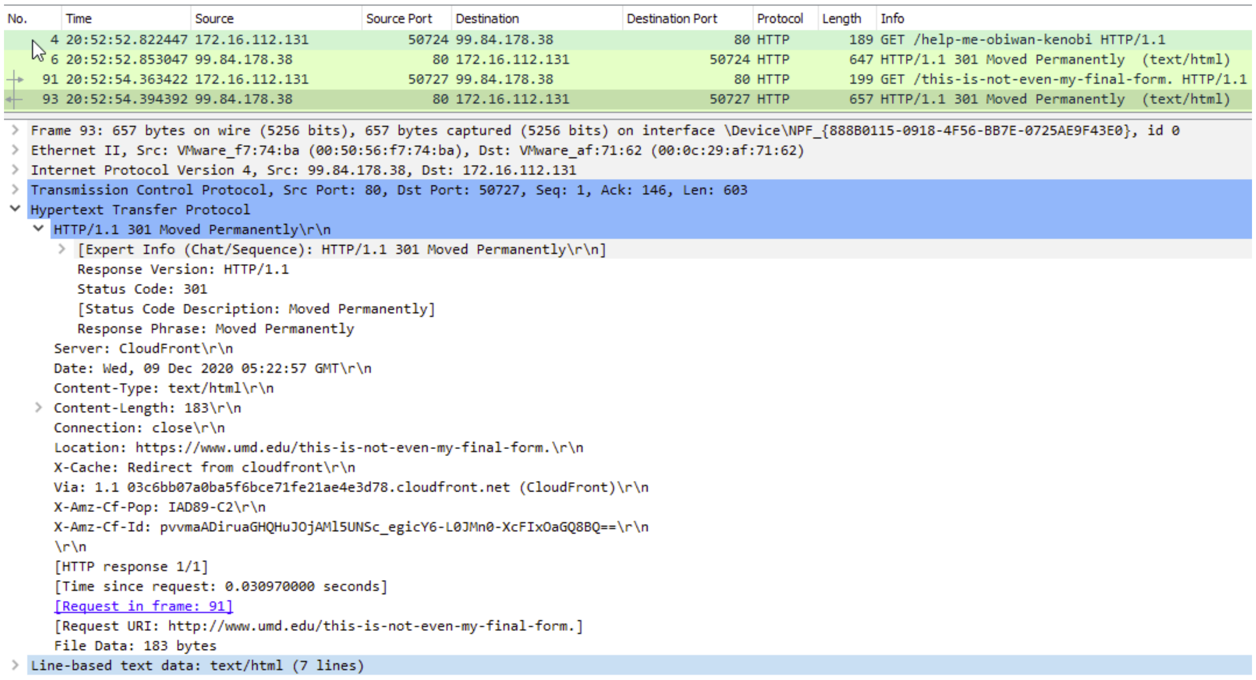


```
(ip.src == 172.16.112.131 or ip.dst == 172.16.112.131)

No.  Time      Source          Source Port Destination     Destination Port Protocol Length Info
   4 20:52:52  172.16.112.131        50724 99.84.178.38                  80 HTTP        189 GET /help-me-obiwan-kenobi HTTP/1.1
   6 20:52:52  99.84.178.38             80 172.16.112.131             50724 HTTP        647 HTTP/1.1 301 Moved Permanently  (text/html)
  91 20:52:54  172.16.112.131        50727 99.84.178.38                  80 HTTP        199 GET /this-is-not-even-my-final-form. HTTP/1.1

v Transmission Control Protocol, Src Port: 80, Dst Port: 50724, Seq: 1, Ack: 136, Len: 593
    Source Port: 80
    Destination Port: 50724
    [Stream index: 0]
    [TCP Segment Len: 593]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 2833250491
    [Next sequence number: 595    (relative sequence number)]
    Acknowledgment number: 136    (relative ack number)
    Acknowledgment number (raw): 1230459386
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x019 (FIN, PSH, ACK)
    Window size value: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x51d2 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
    TCP payload (593 bytes)
v Hypertext Transfer Protocol
  v HTTP/1.1 301 Moved Permanently\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
      Response Version: HTTP/1.1
      Status Code: 301
      [Status Code Description: Moved Permanently]
      Response Phrase: Moved Permanently
    Server: CloudFront\r\n
    Date: Wed, 09 Dec 2020 05:22:55 GMT\r\n
    Content-Type: text/html\r\n
  > Content-Length: 183\r\n
    Connection: close\r\n
    Location: https://www.umd.edu/help-me-obiwan-kenobi\r\n
    X-Cache: Redirect from cloudfront\r\n
    Via: 1.1 46aba6b15045c2b494b2c260627fbfdb.cloudfront.net (CloudFront)\r\n
    X-Amz-Cf-Pop: IAD89-C2\r\n
    X-Amz-Cf-Id: vBVzgi55AvcJNrBGcR0tpoc7LVesqJXWea-ekGC3YV2C5JgoKkX52g==\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.030600000 seconds]
    [Request in frame: 4]
    [Request URI: http://www.umd.edu/help-me-obiwan-kenobi]
    File Data: 183 bytes
```

Figure 10 – Packet Details for the HTTP Response

Figure 10 shows the HTTP response message after the GET request.

Two seconds later, the attacker makes another GET request to a URL called “This is not even my final form” (see packet #91 in Figure 8) and receives a response after 31 milliseconds. The response (packet #93) is still a Status 301 type of response. The requested object is permanently moved.



```
No.   Time            Source          Source Port Destination     Destination Port Protocol Length Info
    4 20:52:52.822447 172.16.112.131        50724 99.84.178.38                  80 HTTP        189 GET /help-me-obiwan-kenobi HTTP/1.1
    6 20:52:52.853047 99.84.178.38             80 172.16.112.131             50724 HTTP        647 HTTP/1.1 301 Moved Permanently  (text/html)
   91 20:52:54.363422 172.16.112.131        50727 99.84.178.38                  80 HTTP        199 GET /this-is-not-even-my-final-form. HTTP/1.1
   93 20:52:54.394392 99.84.178.38             80 172.16.112.131             50727 HTTP        657 HTTP/1.1 301 Moved Permanently  (text/html)

> Frame 93: 657 bytes on wire (5256 bits), 657 bytes captured (5256 bits) on interface \Device\NPF_{888B0115-0918-4F56-BB7E-0725AE9F43E0}, id 0
> Ethernet II, Src: VMware_f7:74:ba (00:50:56:f7:74:ba), Dst: VMware_af:71:62 (00:0c:29:af:71:62)
> Internet Protocol Version 4, Src: 99.84.178.38, Dst: 172.16.112.131
> Transmission Control Protocol, Src Port: 80, Dst Port: 50727, Seq: 1, Ack: 146, Len: 603
v Hypertext Transfer Protocol
  v HTTP/1.1 301 Moved Permanently\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
      Response Version: HTTP/1.1
      Status Code: 301
      [Status Code Description: Moved Permanently]
      Response Phrase: Moved Permanently
    Server: CloudFront\r\n
    Date: Wed, 09 Dec 2020 05:22:57 GMT\r\n
    Content-Type: text/html\r\n
  > Content-Length: 183\r\n
    Connection: close\r\n
    Location: https://www.umd.edu/this-is-not-even-my-final-form.\r\n
    X-Cache: Redirect from cloudfront\r\n
    Via: 1.1 03c6bb07a0ba5f6bce71fe21ae4e3d78.cloudfront.net (CloudFront)\r\n
    X-Amz-Cf-Pop: IAD89-C2\r\n
    X-Amz-Cf-Id: pvvmaADiruaGHQHuJOjAMl5UNSc_egicY6-L0JMn0-XcFIxOaGQ8BQ==\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.030970000 seconds]
    [Request in frame: 91]
    [Request URI: http://www.umd.edu/this-is-not-even-my-final-form.]
    File Data: 183 bytes
> Line-based text data: text/html (7 lines)
```

Figure 11 – Packet Details for the Second GET Request

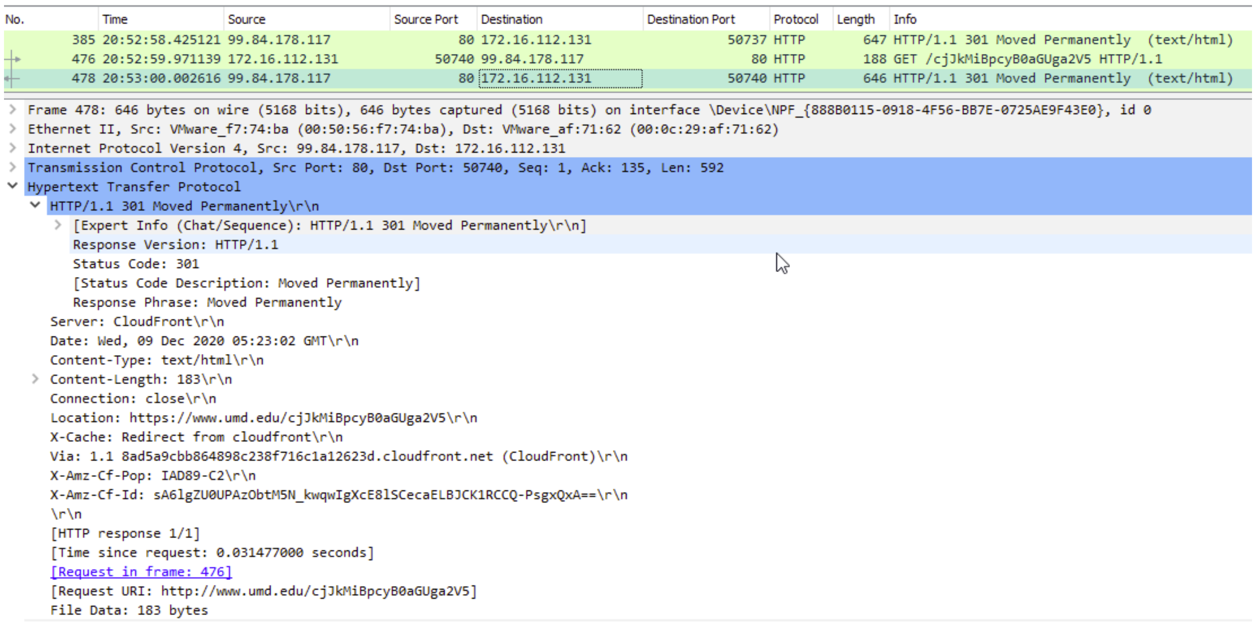
The attacker is trying to communicate implicitly and send messages to his external accomplice, because he sends another message which says “You’re my only hope” (packet #189). It is a GET message sent via the HTTP protocol with the URL http://www.umd.edu/youre-my-only-hope which also returns a message with status code 301. As long as the two executable files are running, these processes loop infinitely.

The next packet I analyze is a packet where the attacker says “All your base 64 are belong to us”. It is an HTTP request message (packet #288). Base64 is generally being used by hackers to try and hide their malware from users. Base64 is the most compact way to transfer binary information using printable ASCII characters.

Base64 provides a level of security making sensitive information difficult to decipher. The use of base64 provides significant advantage to attackers while providing minimal benefit to defenders. The use of base64 can result in the disclosure of passwords, bypass of data leakage protection systems or even cross scripting attacks. Because of these risks, detecting base64 usage on a network is an important part of security.

The response to this packet is found on packet #290, another response message with status code 301 (permanently moved).

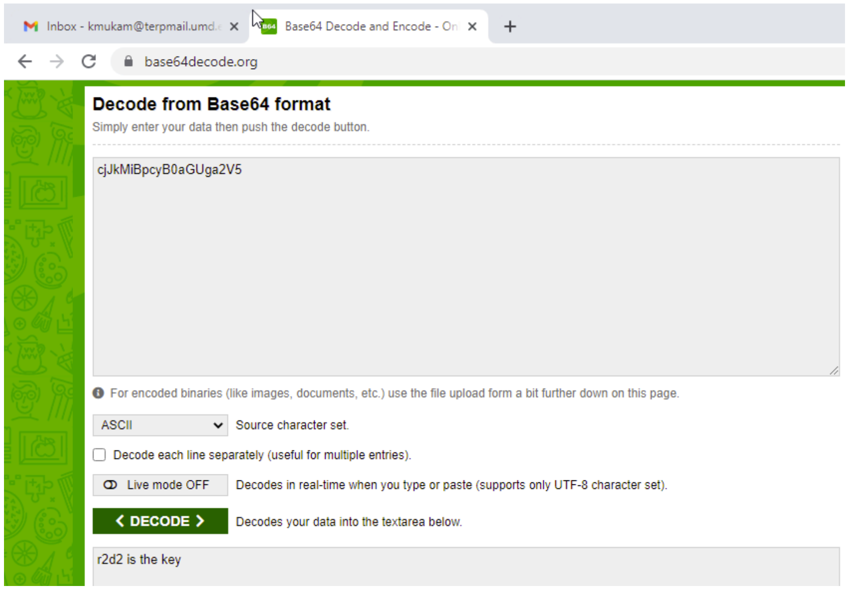
Then, on packet 476, the attacker sends an HTTP request message with request URI /cjJkMiBpcyB0aGUga2V5. I immediately suspected that this was a base64 message. After approximately 31 milliseconds, the response came (moved permanently).



```
No.    Time             Source          Source Port Destination     Destination Port Protocol Length Info
   385 20:52:58.425121 99.84.178.117            80 172.16.112.131             50737 HTTP      647 HTTP/1.1 301 Moved Permanently  (text/html)
   476 20:52:59.971139 172.16.112.131        50740 99.84.178.117                 80 HTTP      188 GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1
   478 20:53:00.002616 99.84.178.117            80 172.16.112.131             50740 HTTP      646 HTTP/1.1 301 Moved Permanently  (text/html)

> Frame 478: 646 bytes on wire (5168 bits), 646 bytes captured (5168 bits) on interface \Device\NPF_{888B0115-0918-4F56-BB7E-0725AE9F43E0}, id 0
> Ethernet II, Src: VMware_f7:74:ba (00:50:56:f7:74:ba), Dst: VMware_af:71:62 (00:0c:29:af:71:62)
> Internet Protocol Version 4, Src: 99.84.178.117, Dst: 172.16.112.131
> Transmission Control Protocol, Src Port: 80, Dst Port: 50740, Seq: 1, Ack: 135, Len: 592
v Hypertext Transfer Protocol
  v HTTP/1.1 301 Moved Permanently\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
      Response Version: HTTP/1.1
      Status Code: 301
      [Status Code Description: Moved Permanently]
      Response Phrase: Moved Permanently
    Server: CloudFront\r\n
    Date: Wed, 09 Dec 2020 05:23:02 GMT\r\n
    Content-Type: text/html\r\n
  > Content-Length: 183\r\n
    Connection: close\r\n
    Location: https://www.umd.edu/cjJkMiBpcyB0aGUga2V5\r\n
    X-Cache: Redirect from cloudfront\r\n
    Via: 1.1 8ad5a9cbb864898c238f716c1a12623d.cloudfront.net (CloudFront)\r\n
    X-Amz-Cf-Pop: IAD89-C2\r\n
    X-Amz-Cf-Id: sA6lgZU0UPAzObtM5N_kwqwIgXcE8lSCecaELBJCK1RCCQ-PsgxQxA==\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.031477000 seconds]
    [Request in frame: 476]
    [Request URI: http://www.umd.edu/cjJkMiBpcyB0aGUga2V5]
    File Data: 183 bytes
```

Figure 12 – Response to the base64 Request

I copied that base64 code and I used an online tool (base64decode.org) to decode the message. The decoded message read "**r2d2 is the key**".



Decode from Base64 format

Simply enter your data then push the decode button.

cjJkMiBpcyB0aGUga2V5

For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

ASCII Source character set.

Decode each line separately (useful for multiple entries).

Live mode OFF Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE > Decodes your data into the textarea below.

r2d2 is the key

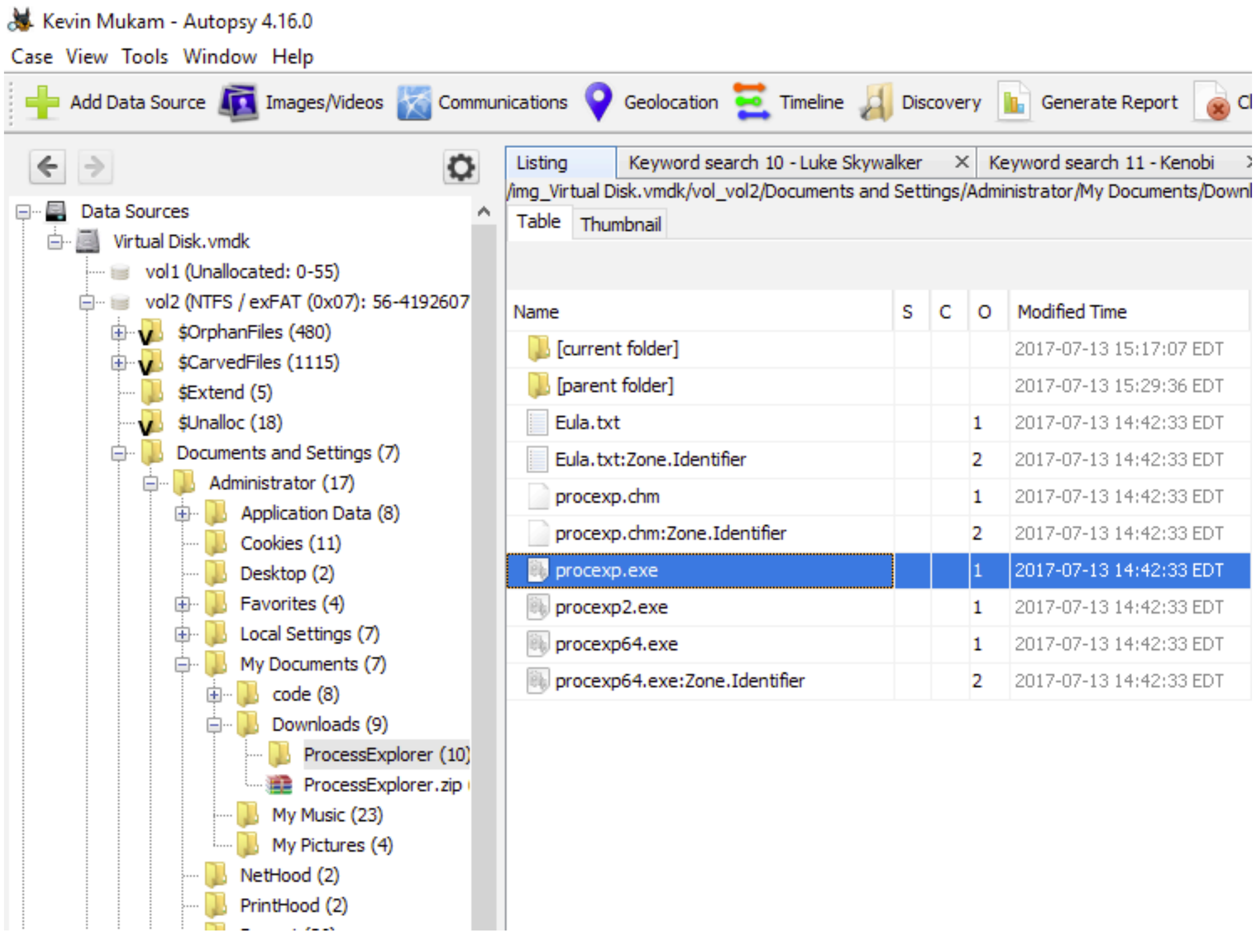
Figure 13 – Decoded Message from Packet Capture

To view the conversation between the attacker and his external collaborator more closely, I went to Statistics → Conversations → Follow Stream. Here, I could look at the conversations going back and forth in the packet capture. A stream refers to the flow of data back and forth over the course of a protocol conversation between the two endpoints.

3.3 Additional Research

3.3.1 Procexp.exe

In Autopsy, I kept scanning through the disk, and I made sure to check in the most common places where the malware could be, because attackers like to hide in plain sight. One of the folders I checked was Process Explorer. Documents & Settings → Administrator → My Documents → Downloads → Process Explorer.


Kevin Mukam - Autopsy 4.16.0
Case View Tools Window Help
Add Data Source
Images/Videos
Communications
Geolocation
Timeline
Discovery
Generate Report
Listing
Keyword search 10 - Luke Skywalker
Keyword search 11 - Kenobi
Table
Thumbnail

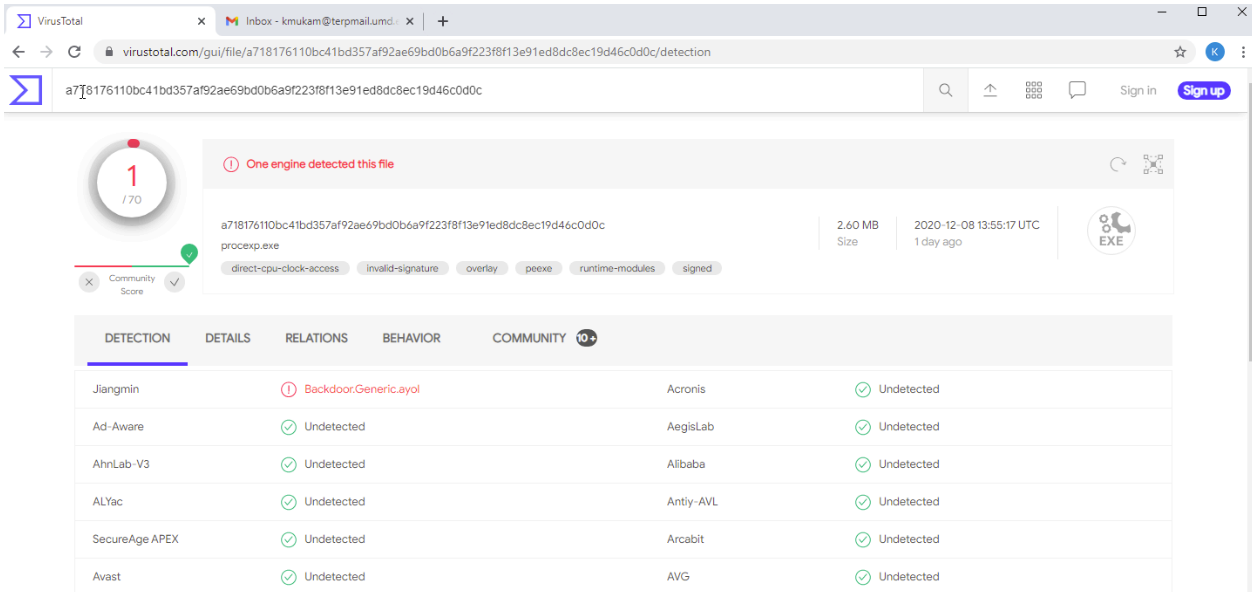
/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/Downl

- Data Sources
  - Virtual Disk.vmdk
    - vol1 (Unallocated: 0-55)
    - vol2 (NTFS / exFAT (0x07): 56-4192607
      - $OrphanFiles (480)
      - $CarvedFiles (1115)
      - $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)
            - Downloads (9)
              - ProcessExplorer (10)
              - ProcessExplorer.zip
            - My Music (23)
            - My Pictures (4)
          - NetHood (2)
          - PrintHood (2)

Name	S	C	O	Modified Time
[current folder]				2017-07-13 15:17:07 EDT
[parent folder]				2017-07-13 15:29:36 EDT
Eula.txt			1	2017-07-13 14:42:33 EDT
Eula.txt:Zone.Identifier			2	2017-07-13 14:42:33 EDT
procexp.chm			1	2017-07-13 14:42:33 EDT
procexp.chm:Zone.Identifier			2	2017-07-13 14:42:33 EDT
procexp.exe			1	2017-07-13 14:42:33 EDT
procexp2.exe			1	2017-07-13 14:42:33 EDT
procexp64.exe			1	2017-07-13 14:42:33 EDT
procexp64.exe:Zone.Identifier			2	2017-07-13 14:42:33 EDT

Figure 14 – Process Explorer Folder

In this folder, I found a few executable files, for which I copied the MD5 hashes and I checked on Virus Total. It turned out that procexp.exe and procexp2.exe are backdoors. Procexp.exe and Procexp2.exe have MD5 hash 62028945d0ab974e183756ebbb1ca07f which reveal backdoor properties on virustotal.com.


virustotal.com/gui/file/a718176110bc41bd357af92ae69bd0b6a9f223f8f13e91ed8dc8ec19d46c0d0c/detection

One engine detected this file

a718176110bc41bd357af92ae69bd0b6a9f223f8f13e91ed8dc8ec19d46c0d0c
procexp.exe
direct-cpu-clock-access invalid-signature overlay peexe runtime-modules signed

2.60 MB Size
2020-12-08 13:55:17 UTC 1 day ago

Jiangmin	Backdoor.Generic.ayol	Acronis	Undetected
Ad-Aware	Undetected	AegisLab	Undetected
AhnLab-V3	Undetected	Alibaba	Undetected
ALYac	Undetected	Antiy-AVL	Undetected
SecureAge APEX	Undetected	Arcabit	Undetected
Avast	Undetected	AVG	Undetected

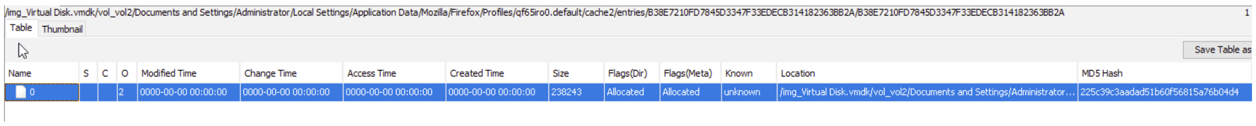
Figure 15 – Procexp.exe Analysis on Virus Total

A backdoor is a method by which authorized and unauthorized users are able to get around security measures and gain high level user access on a computer system, network or software application. Once they are in, cybercriminals use a backdoor to steal personal data, financial data, install malware and hijack devices.

With backdoors, hackers can install spyware (collects information about users, sites they visit, file they download, passwords, usernames), ransomware (designed to encrypt files and lockdown the computer while requesting payment), or they can use the captured computer in a DDoS attack (take command of a computer remotely and overwhelm a network with traffic from their hijacked computers).

3.3.2 Death Star Plans

In the path shown on figure 16, I found a file titled "0" which contained the text "The death star


/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/Local Settings/Application Data/Mozilla/Firefox/Profiles/qf65iro0.default/cache2/entries/B38E7210FD7845D3347F33EDECB314182363BB2A/B38E7210FD7845D3347F33EDECB314182363BB2A

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location	MD5 Hash
0			2	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	238243	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator...	225c39c3aadad51b60f56815a76b04d4

Figure 16 – Suspected File

plans are now in your hands".

I copied the text inside the details of the file (String), which is actually an HTML code. When I opened the web page, I found the following image and text:

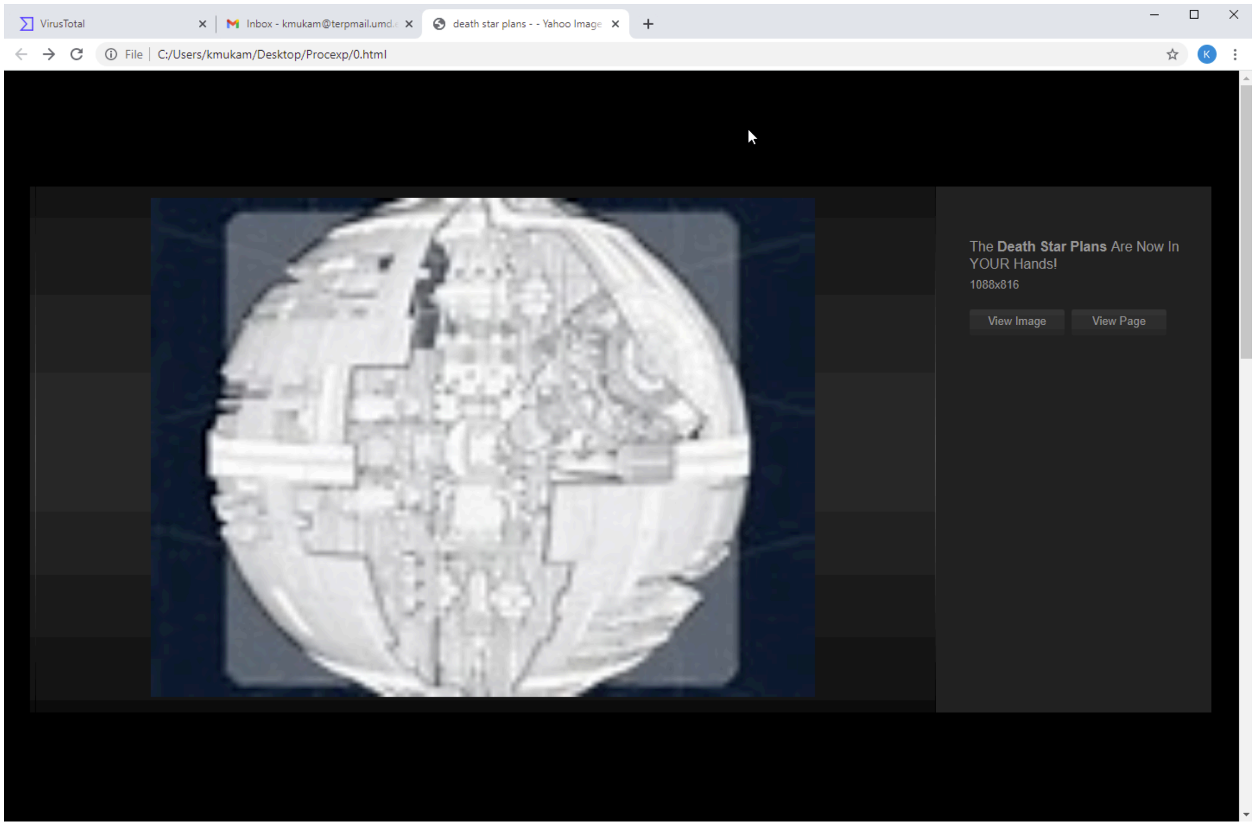

VirusTotal
Inbox - kmukam@terpmail.umd
death star plans - - Yahoo Image
File | C:/Users/kmukam/Desktop/Procexp/0.html
The Death Star Plans Are Now In YOUR Hands!
1088x816
View Image
View Page

Figure 17 – Death Star Plans

"View page" redirects here: https://starwars.com/news/contributor/rwindham and to me this was inconclusive.

3.4 Final Form

After further research and analysis, the final form of the malware writer's malware was found and the steps are explained below:

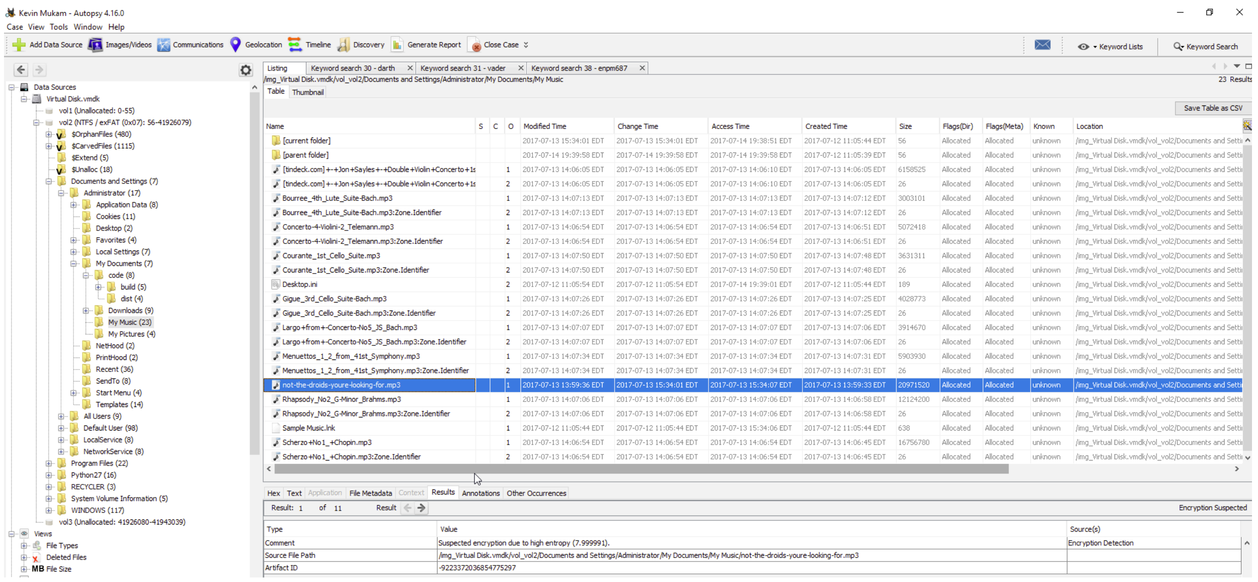
3.4.1 Extraction

It is common knowledge that attackers like to hide in plain sight, so after going through folders like config folders and most of the system folders, I returned to the Documents and Settings folder to perform further analysis.

Documents & Settings → Administrator → My Documents → Downloads → My Music.

In this folder, there are a few things to note. There is a file called "not-the-droids-youre-looking-for.mp3" which was created on July 13, 2017 at 13:59:33. During that time, VeraCrypt was running on the attacker's system (see section 3.1.3). Also, it was accessed at 15:34:07 and at that time, final-form.exe was running (see section 3.1.3). This raised a lot of suspicion, so I looked at more detail. I checked the indexed text, the file metadata, results, annotations and other occurrences.

Under the results tab, there is a comment which says "Suspected encryption due to high entropy", "Encryption Detection". As a result of this, I had to extract the file to perform further analysis.


Kevin Mukam - Autopsy 4.16.0
Case View Tools Window Help
Add Data Source Images/Videos Communications Geolocation Timeline Discovery Generate Report Close Case
Keyword Lists Keyword Search
Listing | Keyword search 30 - darth | Keyword search 31 - vader | Keyword search 38 - enpm687
Table Thumbnail
Save Table as CSV
Hex Text Application File Metadata Context Results Annotations Other Occurrences

/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/My Music

23 Results

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known	Location
[current folder]				2017-07-13 15:34:01 EDT	2017-07-13 15:34:01 EDT	2017-07-14 19:38:51 EDT	2017-07-12 11:05:44 EDT	56	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
[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 11:05:39 EDT	56	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
[tindeck.com]+-+Jon+Sayles+-+Double+Violin+Concerto+1s			1	2017-07-13 14:06:05 EDT	2017-07-13 14:06:05 EDT	2017-07-13 14:06:10 EDT	2017-07-13 14:06:05 EDT	6158525	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
[tindeck.com]+-+Jon+Sayles+-+Double+Violin+Concerto+1s			2	2017-07-13 14:06:05 EDT	2017-07-13 14:06:05 EDT	2017-07-13 14:06:10 EDT	2017-07-13 14:06:05 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Bourree_4th_Lute_Suite-Bach.mp3			1	2017-07-13 14:07:13 EDT	2017-07-13 14:07:13 EDT	2017-07-13 14:07:13 EDT	2017-07-13 14:07:12 EDT	3003101	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Bourree_4th_Lute_Suite-Bach.mp3:Zone.Identifier			2	2017-07-13 14:07:13 EDT	2017-07-13 14:07:13 EDT	2017-07-13 14:07:13 EDT	2017-07-13 14:07:12 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Concerto-4-Violini-2_Telemann.mp3			1	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:51 EDT	5072418	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Concerto-4-Violini-2_Telemann.mp3:Zone.Identifier			2	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:51 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Courante_1st_Cello_Suite.mp3			1	2017-07-13 14:07:50 EDT	2017-07-13 14:07:50 EDT	2017-07-13 14:07:50 EDT	2017-07-13 14:07:48 EDT	3631311	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Courante_1st_Cello_Suite.mp3:Zone.Identifier			2	2017-07-13 14:07:50 EDT	2017-07-13 14:07:50 EDT	2017-07-13 14:07:50 EDT	2017-07-13 14:07:48 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Desktop.ini			2	2017-07-12 11:05:54 EDT	2017-07-12 11:05:54 EDT	2017-07-14 19:39:01 EDT	2017-07-12 11:05:44 EDT	189	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Gigue_3rd_Cello_Suite-Bach.mp3			1	2017-07-13 14:07:26 EDT	2017-07-13 14:07:26 EDT	2017-07-13 14:07:26 EDT	2017-07-13 14:07:25 EDT	4028773	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Gigue_3rd_Cello_Suite-Bach.mp3:Zone.Identifier			2	2017-07-13 14:07:26 EDT	2017-07-13 14:07:26 EDT	2017-07-13 14:07:26 EDT	2017-07-13 14:07:25 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Largo+from+-Concerto-No5_JS_Bach.mp3			1	2017-07-13 14:07:07 EDT	2017-07-13 14:07:07 EDT	2017-07-13 14:07:07 EDT	2017-07-13 14:07:06 EDT	3914670	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Largo+from+-Concerto-No5_JS_Bach.mp3:Zone.Identifier			2	2017-07-13 14:07:07 EDT	2017-07-13 14:07:07 EDT	2017-07-13 14:07:07 EDT	2017-07-13 14:07:06 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Menuettos_1_2_from_41st_Symphony.mp3			1	2017-07-13 14:07:34 EDT	2017-07-13 14:07:34 EDT	2017-07-13 14:07:34 EDT	2017-07-13 14:07:31 EDT	5903930	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Menuettos_1_2_from_41st_Symphony.mp3:Zone.Identifier			2	2017-07-13 14:07:34 EDT	2017-07-13 14:07:34 EDT	2017-07-13 14:07:34 EDT	2017-07-13 14:07:31 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
not-the-droids-youre-looking-for.mp3			1	2017-07-13 13:59:36 EDT	2017-07-13 15:34:01 EDT	2017-07-13 15:34:07 EDT	2017-07-13 13:59:33 EDT	20971520	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Rhapsody_No2_G-Minor_Brahms.mp3			1	2017-07-13 14:07:06 EDT	2017-07-13 14:07:06 EDT	2017-07-13 14:07:06 EDT	2017-07-13 14:06:58 EDT	12124200	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Rhapsody_No2_G-Minor_Brahms.mp3:Zone.Identifier			2	2017-07-13 14:07:06 EDT	2017-07-13 14:07:06 EDT	2017-07-13 14:07:06 EDT	2017-07-13 14:06:58 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Sample Music.lnk			1	2017-07-12 11:05:44 EDT	2017-07-12 11:05:44 EDT	2017-07-13 15:34:06 EDT	2017-07-12 11:05:44 EDT	638	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Scherzo+No1_+Chopin.mp3			1	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:45 EDT	16756780	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti
Scherzo+No1_+Chopin.mp3:Zone.Identifier			2	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:54 EDT	2017-07-13 14:06:45 EDT	26	Allocated	Allocated	unknown	/img_Virtual Disk.vmdk/vol_vol2/Documents and Setti

Result: 1 of 11 Result

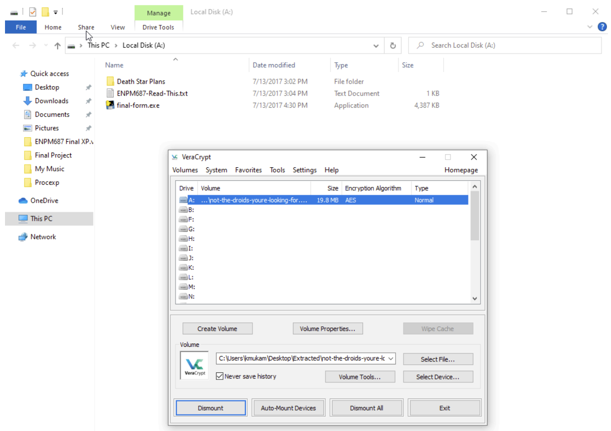
Encryption Suspected

Type	Value	Source(s)
Comment	Suspected encryption due to high entropy (7.999991).	Encryption Detection
Source File Path	/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Administrator/My Documents/My Music/not-the-droids-youre-looking-for.mp3	
Artifact ID	-9223372036854775297	

Figure 18 - Encrypted File Hidden in Plain Sight

3.4.2 Decryption

After extracting the mp3 file, I launched VeraCrypt. I took the following steps:
Select a Volume → Select file → Select the mp3 file → Mount → Password: **r2d2** (obtained in section 3.2). This successfully decrypted the file in a separate folder in as shown in figure 19.



Name	Date modified	Type	Size
Death Star Plans	7/13/2017 3:02 PM	File folder	
ENPM687-Read-This.txt	7/13/2017 3:04 PM	Text Document	1 KB
final-form.exe	7/13/2017 4:30 PM	Application	4,387 KB

Drive	Volume	Size	Encryption Algorithm	Type
A:	...\not-the-droids-youre-looking-for....	19.8 MB	AES	Normal
B:				
F:				
G:				
H:				
I:				
J:				
K:				
L:				
M:				
N:				

Volume: C:\Users\kmukam\Desktop\Extracted\not-the-droids-youre-lc

[✓] Never save history

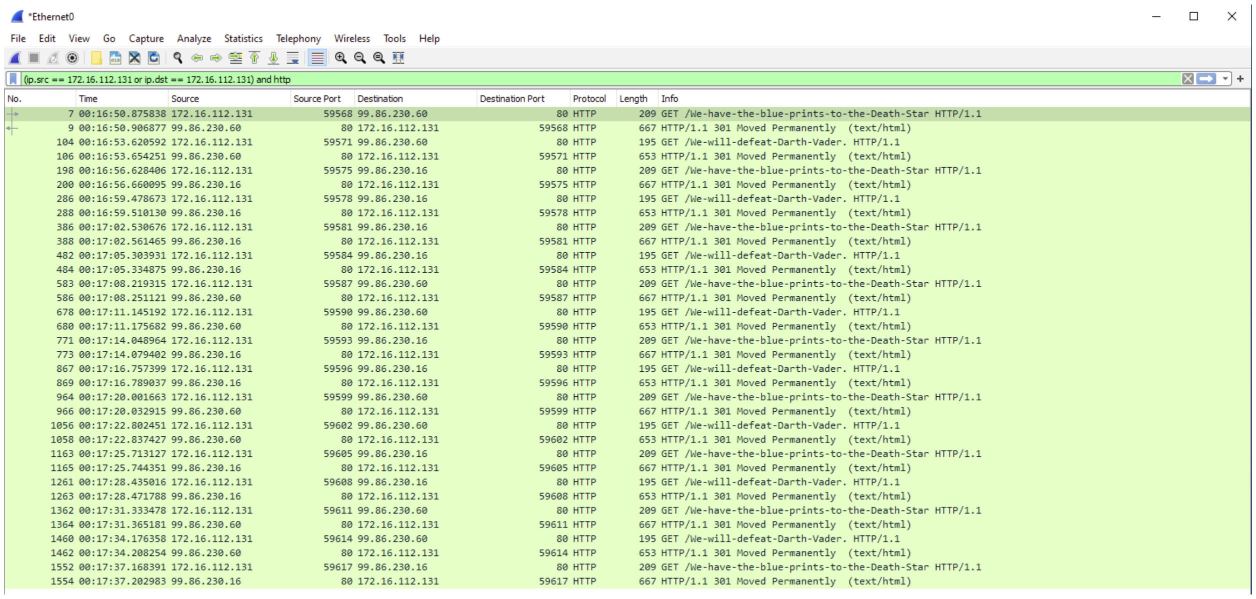
Figure 19 – Decrypted Files with VeraCrypt

The decrypted files contain the final form executable file, a text file and pictures in a Death Star Plans folder.

The text file states: “To complete the last part of this project, you will need to determine what the message sent by final-form.exe is”.

3.4.3 Network Analysis

To determine what the final form message sent is, I ran the final-form.exe file. Then, I launched Wireshark to capture traffic while this file was running. I created a filter to capture traffic going to and from my virtual machine.


*Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(ip.src == 172.16.112.131 or ip.dst == 172.16.112.131) and http

No.   Time            Source          Source Port Destination    Destination Port Protocol Length Info
    7 00:16:50.875838 172.16.112.131        59568 99.86.230.60                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
    9 00:16:50.906877 99.86.230.60             80 172.16.112.131            59568 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
  104 00:16:53.620592 172.16.112.131        59571 99.86.230.60                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  106 00:16:53.654251 99.86.230.60             80 172.16.112.131            59571 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
  198 00:16:56.628406 172.16.112.131        59575 99.86.230.16                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  200 00:16:56.660095 99.86.230.16             80 172.16.112.131            59575 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
  286 00:16:59.478673 172.16.112.131        59578 99.86.230.16                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  288 00:16:59.510130 99.86.230.16             80 172.16.112.131            59578 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
  386 00:17:02.530676 172.16.112.131        59581 99.86.230.16                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  388 00:17:02.561465 99.86.230.16             80 172.16.112.131            59581 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
  482 00:17:05.303931 172.16.112.131        59584 99.86.230.16                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  484 00:17:05.334875 99.86.230.16             80 172.16.112.131            59584 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
  583 00:17:08.219315 172.16.112.131        59587 99.86.230.60                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  586 00:17:08.251121 99.86.230.60             80 172.16.112.131            59587 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
  678 00:17:11.145192 172.16.112.131        59590 99.86.230.60                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  680 00:17:11.175682 99.86.230.60             80 172.16.112.131            59590 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
  771 00:17:14.048964 172.16.112.131        59593 99.86.230.16                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  773 00:17:14.079402 99.86.230.16             80 172.16.112.131            59593 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
  867 00:17:16.757399 172.16.112.131        59596 99.86.230.16                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  869 00:17:16.789037 99.86.230.16             80 172.16.112.131            59596 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
  964 00:17:20.001663 172.16.112.131        59599 99.86.230.60                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  966 00:17:20.032915 99.86.230.60             80 172.16.112.131            59599 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
 1056 00:17:22.802451 172.16.112.131        59602 99.86.230.60                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
 1058 00:17:22.837427 99.86.230.60             80 172.16.112.131            59602 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
 1163 00:17:25.713127 172.16.112.131        59605 99.86.230.16                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
 1165 00:17:25.744351 99.86.230.16             80 172.16.112.131            59605 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
 1261 00:17:28.435016 172.16.112.131        59608 99.86.230.16                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
 1263 00:17:28.471788 99.86.230.16             80 172.16.112.131            59608 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
 1362 00:17:31.333478 172.16.112.131        59611 99.86.230.60                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
 1364 00:17:31.365181 99.86.230.60             80 172.16.112.131            59611 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
 1460 00:17:34.176358 172.16.112.131        59614 99.86.230.60                 80 HTTP      195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
 1462 00:17:34.208254 99.86.230.60             80 172.16.112.131            59614 HTTP      653 HTTP/1.1 301 Moved Permanently  (text/html)
 1552 00:17:37.168391 172.16.112.131        59617 99.86.230.16                 80 HTTP      209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
 1554 00:17:37.202983 99.86.230.16             80 172.16.112.131            59617 HTTP      667 HTTP/1.1 301 Moved Permanently  (text/html)
```

Figure 20 - Network Analysis of final-form.exe

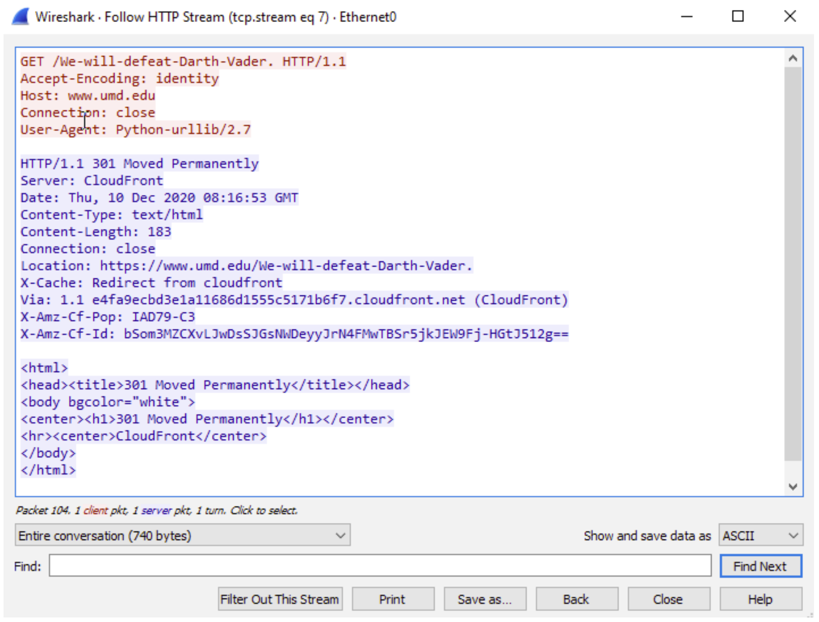
The attacker was trying to communicate the following message to his external collaborator “We have the blue prints to the death star, we will defeat Darth Vader”. I caught the blue prints (they are attached to this document in a ZIP file).

Like in the previous sections, this communication would loop infinitely while final-form.exe was running. To view the conversation more closely,

Wireshark → Statistics → Conversations → Follow Stream

shows the conversations going back and forth in the packet capture.

In a clear ASCII code, we can see that that the stream content is displayed in the same sequence as it appeared on the network. Traffic from final-form.exe to the host is marked in red, while the reverse traffic is marked in blue (figure 21).


Wireshark · Follow HTTP Stream (tcp.stream eq 7) · Ethernet0
Packet 104. 1 client pkt, 1 server pkt, 1 turn. Click to select.
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help

```
GET /We-will-defeat-Darth-Vader. HTTP/1.1
Accept-Encoding: identity
Host: www.umd.edu
Connection: close
User-Agent: Python-urllib/2.7

HTTP/1.1 301 Moved Permanently
Server: CloudFront
Date: Thu, 10 Dec 2020 08:16:53 GMT
Content-Type: text/html
Content-Length: 183
Connection: close
Location: https://www.umd.edu/We-will-defeat-Darth-Vader.
X-Cache: Redirect from cloudfront
Via: 1.1 e4fa9ecbd3e1a11686d1555c5171b6f7.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: IAD79-C3
X-Amz-Cf-Id: bSom3MZCXvLJwDsSJGsNWDeyyJrN4FMwTBSr5jkJEW9Fj-HGtJ512g==

<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>CloudFront</center>
</body>
</html>
```

Entire conversation (740 bytes)

Show and save data as: ASCII

Figure 21 – HTTP Stream

The process is repeated over and over as long as final-form.exe is running.

4. Repository

The following files and information were detected and extracted during the investigation:

obiwan.exe

Executable file containing malware (see figure 5) that was extracted from the attacker's computer. MD5 hash: 97c90c038e2b060acc2354b9eb5f69d3.

obiwan2.exe

Executable file containing malware (see figure 6) that was extracted from the attacker's computer. MD5 hash: cb183dc18d160ec8ca1651e717182a04.

obiwan packet capture

Network analysis showing the conversation between the attacker and an external source, where the attacker asks for help, and also claims to have all the base64 (see figure 8).

procexp.exe

Executable file extracted from the attacker's computer. Contains a backdoor and is not safe. MD5 hash: 62028945d0ab974e183756ebbb1ca07f.

not-the-droids-youre-looking-for.mp3

MP3 file that was hidden in plain sight. Found in "My Documents", in "My Music" folder of the attacker, this music file is actually encrypted and hides other files such as the death star plans and the final-form.exe file.

final-form.exe

The holy grail. The file I had been looking for since the beginning of the investigation. The key to all the investigation.

final form packet capture

Network analysis showing the final conversation between the attacker and his external source, where he confirms having the blue prints to death star and claims that he will defeat Darth Vader.

All these flags are provided to the Imperial Force, alongside this report.

5. Recommendations

The following recommendations are suggested for the Imperial Forces:

Finding: An attacker claims to have all the base64 files of the Imperial Forces.

Recommendation: To avoid base64 attacks from attackers, block the IP address of the attackers instead of domains. A next generation blacklisting solution is likely to function if an attacker keeps changing domains, but stays on the same IP address.

Base 64 represents a very real risk to organizations that rely on computers, networking and the internet. The detection of Base 64 should be part of a monitoring program inside the Imperial Forces.

The Imperial Forces should implement a program of detection that involves continual reviewing of alerts and tuning of the system. If done properly, base 64 detection can become an effective component of overall information security program.

Finding: Process Explorer installs backdoors in the system.

Recommendation: The Imperial Forces should keep up on your updates with content management software. Regularly patch the systems.

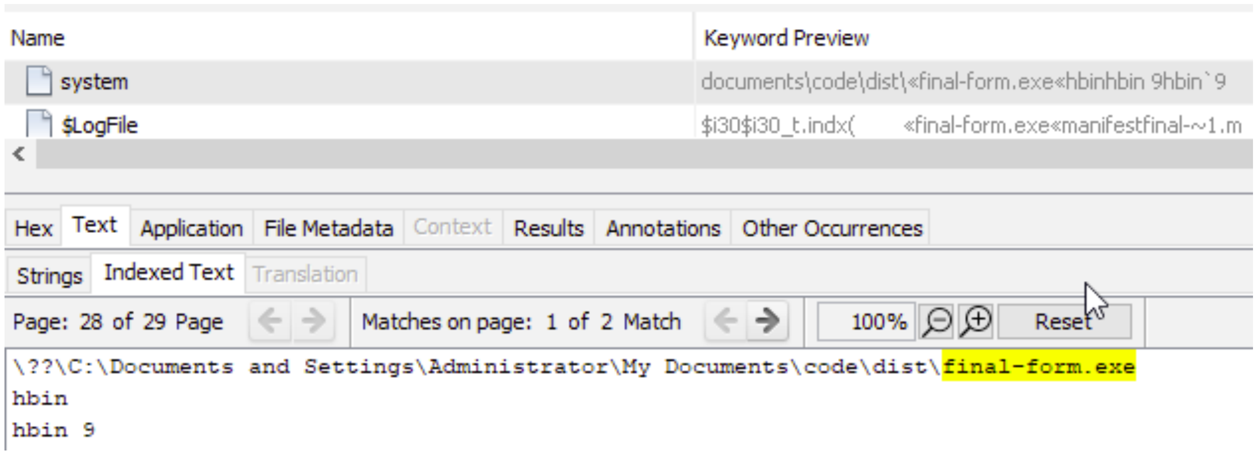
To protect against backdoors, change default passwords, enable multi-factor authentication for every device. Also, continuously monitor network activity. Choose applications and plugins carefully.

Appendix A: Challenges

I used OpenStego several times because I thought that the final form was hidden in image files, so for every image I found, I tried to decode a potential hidden file or relevant information.

I used VeraCrypt several times to try to decode images.

When I searched in the "Keyword Search" bar for "final-form", the results showed that there was a final-form.exe in the "My Documents" folder, however when I opened the document there was nothing there. It was just a remnant of the file.



Name	Keyword Preview
system	documents\code\dist\«final-form.exe«hbinhbin 9hbin`9
$LogFile	$i30$i30_t.indx(«final-form.exe«manifestfinal-~1.m

```
\??\C:\Documents and Settings\Administrator\My Documents\code\dist\final-form.exe
hbin
hbin 9
```

Appendix B: Acronyms

Table 1 - Acronyms

Acronym	**Literal Translation**
ASCII	American Standard Code for Information Interchange
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MD5	Message Digest Algorithm
NTFS	New Technology File System
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
VMDK	Virtual Machine Disk

Appendix C: Glossary

Table 2 - Glossary

Term	Definition
Antivirus	Computer program used to prevent, detect, and remove malware.
Artifacts	A piece of data that may or may not be relevant to the investigation.
Attacker	Individual or organization who breaches the information system of another individual or organization to perform some malicious activity and disrupt the network.
Backdoor	A method by which authorized and unauthorized users are able to get around normal security measures and gain high level user access on a computer system.
Decoding	Process of converting an encoded format back into the original sequence of characters.
Encoding	Transforms data into another format using a scheme that is publicly available so that it can easily be reversed.
Encryption	A security method where information is encoded and can only be accessed or decrypted by a user with the correct encryption key.
Firewall	A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.
Hypervisor	A collection of software modules that provides virtualization of hardware resources (such as CPU, Memory, Network and Storage) and enables multiple computing stacks (made of an operating system and Applications) called to be run on a single physical host.
IP Address	The unique identifying number assigned to every device connected to the internet.
Malware	Any software intentionally designed to cause damage to a computer, server, client, or computer network.
Network Protocol	Established set of rules that determine how data is transmitted between different devices in the same network.
Open-source	Software for which the original source code is made freely available and may be redistributed and modified.
Packet	Small amount of data sent over a network. Each packet includes a source and destination as well as the content being transferred.
Ransomware	A type of malicious software designed to deny access to a computer system or data until a ransom is paid.
Steganography	The practice of concealing a file, message, image, or video within another file, message, image, or video.
Threat	A potential negative action or event facilitated by a vulnerability that results in an unwanted impact to a computer system or application.
Trojan	A type of malware that is often disguised as legitimate software.
Virus	A type of malicious code or program written to alter the way a computer operates
Vulnerabilities	A weakness which can be exploited by a threat actor, such as an attacker, to cross privilege boundaries within a computer system.
Worm	A type of malware that spreads copies of itself from computer to computer.

Appendix D: Referenced Documents

Table 3 - Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
.PF File Extension	https://bit.ly/3g2upuo	2020
.LNKFile Extension	https://fileinfo.com/extension/lnk	12/23/2019
VMware	https://www.vmware.com/products/fusion/faq.html	2020
Autopsy (software)	https://en.wikipedia.org/wiki/Autopsy_(software)	06/13/2020
Wireshark	https://www.wireshark.org/	1998
NTFS (NT File System)	https://searchwindowsserver.techtarget.com/definition/NTFS	November 2008
Base 64 Can Get You Pwned	https://www.sans.org/reading-room/whitepapers/auditing/base64-pwned-33759	04/13/2011
Backdoor Computing Attacks	https://www.malwarebytes.com/backdoor/	2020
VeraCrypt	https://www.veracrypt.fr/en/Home.html	

Appendix E: Approvals

The undersigned acknowledge that they have reviewed the Forensics Investigation Report and agree with the information presented within this document. Changes to this report will be coordinated with, and approved by, the undersigned, or their designated representatives.

Table 4 - Approvals

Document Approved By	Date Approved
Name: **Kevin Mukam**, Author – DFIR Student	**12/11/2020** Date
Name: **Jonas Amoonarquah**, Teacher – DFIR	Date

s