# Final Report

## Summary

From the provided image, we are able to find a file called *obiwan2.exe*  in the *code* folder*.* Autopsy also finds multiple files which are supposedly encrypted (due to high entropy) but we would still need a key to decrypt these files. When we run the *obiwan2.exe* file, and capture the packets using wireshark, we see a few HTTP requests being made. However, since these pageshave already been moved we get a 301 error. On analyzing these requests, we are able to find the *key* for the encrypted files. Upon decryption of the *not-the-droids-youre-looking-for.mp3* using the key that we found, we are provided with another folder containing a few images and another *exe* file which is the *final malware.* We run the file and capture the packets using wireshark. We get the final message which is ***We have the blue prints to the Death Star. We will defeat Darth Vader.***

## Tools Used

- Autopsy
- Veracrypt
- Wireshark

## Repository - Image of the Rebel's System

a. **Following are some of the evidence we find from the image**

    i. Copy of malwares

    ii. Encrypted Files

    iii. WebHistory

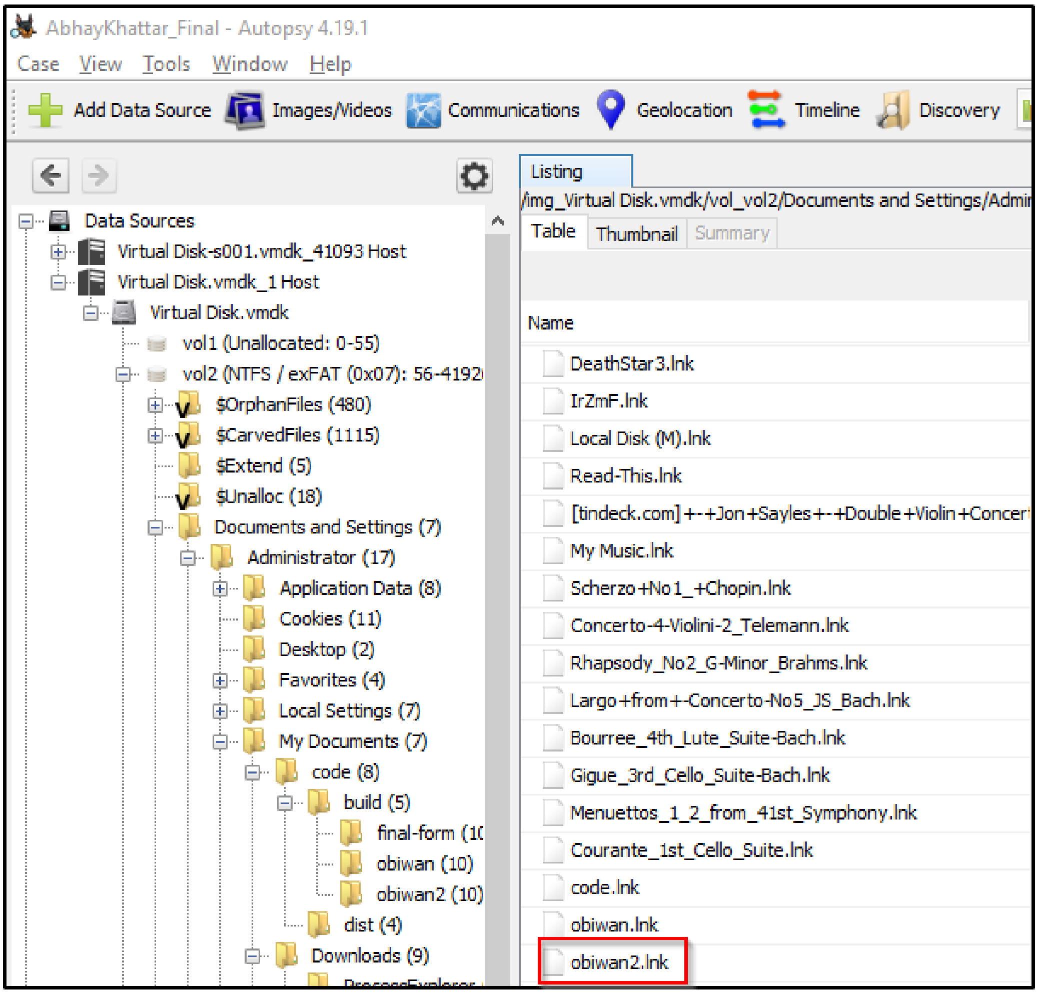
    iv. LogFile

    v. Email

b. **Analysis of the relevant evidence**

    ▼ Copy of Malwares

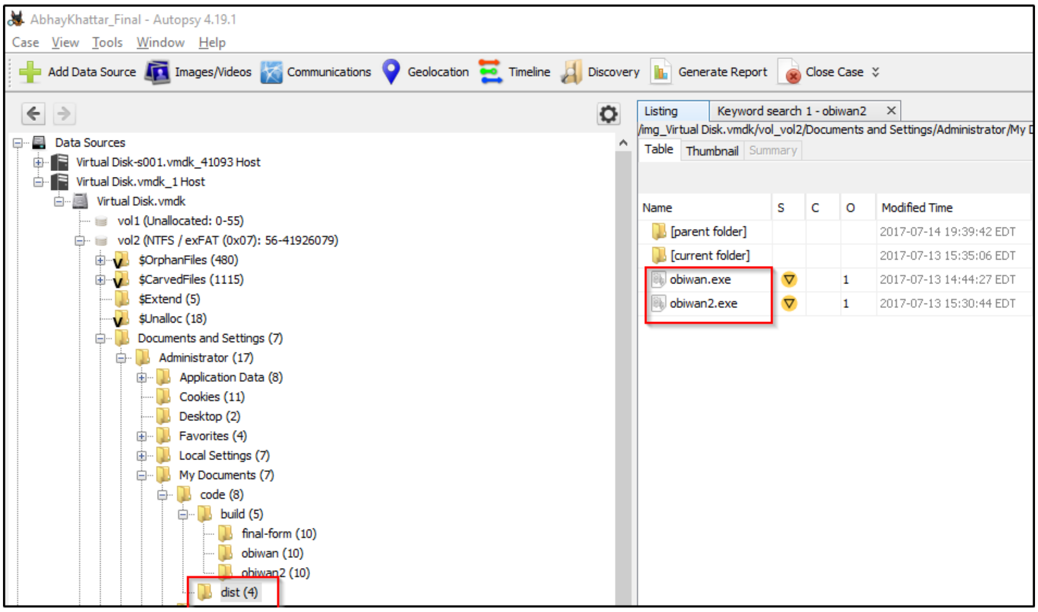
    - We find that there are two files called *obiwan.lnk* and *obiwan2.lnk* in the tab under *Recent Documents.* We find the *exe* files obiwan.exe *and* obiwan2.exe in the My Documents\codes\dist folder.



```
Data Sources
  Virtual Disk-s001.vmdk_41093 Host
  Virtual Disk.vmdk_1 Host
    Virtual Disk.vmdk
      vol1 (Unallocated: 0-55)
      vol2 (NTFS / exFAT (0x07): 56-4192
        $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)
                build (5)
                  final-form (1(
                  obiwan (10)
                  obiwan2 (10)
                dist (4)
              Downloads (9)
                ProcessExplorer
```

/img_Virtual Disk.vmdk/vol_vol2/Documents and Settings/Admi

| Name |
|---|
| DeathStar3.lnk |
| IrZmF.lnk |
| Local Disk (M).lnk |
| Read-This.lnk |
| [tindeck.com]+-+Jon+Sayles+-+Double+Violin+Concer |
| My Music.lnk |
| Scherzo+No1_+Chopin.lnk |
| Concerto-4-Violini-2_Telemann.lnk |
| Rhapsody_No2_G-Minor_Brahms.lnk |
| Largo+from+-Concerto-No5_JS_Bach.lnk |
| Bourree_4th_Lute_Suite-Bach.lnk |
| Gigue_3rd_Cello_Suite-Bach.lnk |
| Menuettos_1_2_from_41st_Symphony.lnk |
| Courante_1st_Cello_Suite.lnk |
| code.lnk |
| obiwan.lnk |
| obiwan2.lnk |



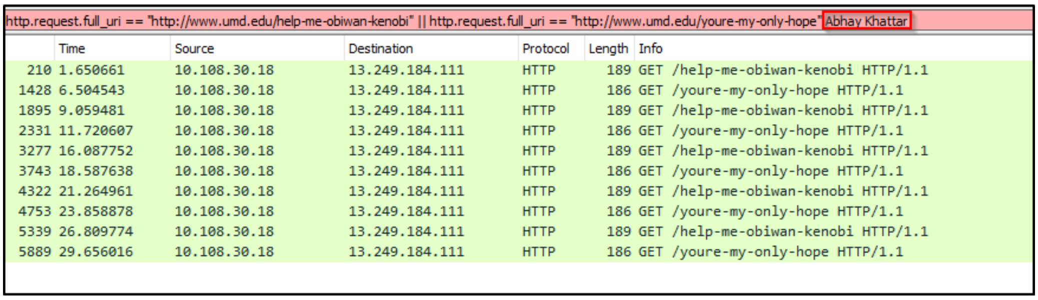
```
Data Sources
  Virtual Disk-s001.vmdk_41093 Host
  Virtual Disk.vmdk_1 Host
    Virtual Disk.vmdk
      vol1 (Unallocated: 0-55)
      vol2 (NTFS / exFAT (0x07): 56-41926079)
        $OrphanFiles (480)
        $CarvedFiles (1115)
        $Extend (5)
        $Unalloc (18)
        Documents and Settings (7)
          Administrator (17)
            Application Data (8)
            Cookies (11)
            Desktop (2)
            Favorites (4)
            Local Settings (7)
            My Documents (7)
              code (8)
                build (5)
                  final-form (10)
                  obiwan (10)
                  obiwan2 (10)
                dist (4)
```

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

| Name | S | C | O | Modified Time |
|---|---|---|---|---|
| [parent folder] | | | | 2017-07-14 19:39:42 EDT |
| [current folder] | | | | 2017-07-13 15:35:06 EDT |
| obiwan.exe | ▽ | | 1 | 2017-07-13 14:44:27 EDT |
| obiwan2.exe | ▽ | | 1 | 2017-07-13 15:30:44 EDT |

- We extract these two files to the host system and run them while capturing the traffic using wireshark. The reason why we use wireshark along with the malware is because we find *Log File* which has a part of the code of the *obiwan2.exe* file and it is creating python requests to open the webpages.

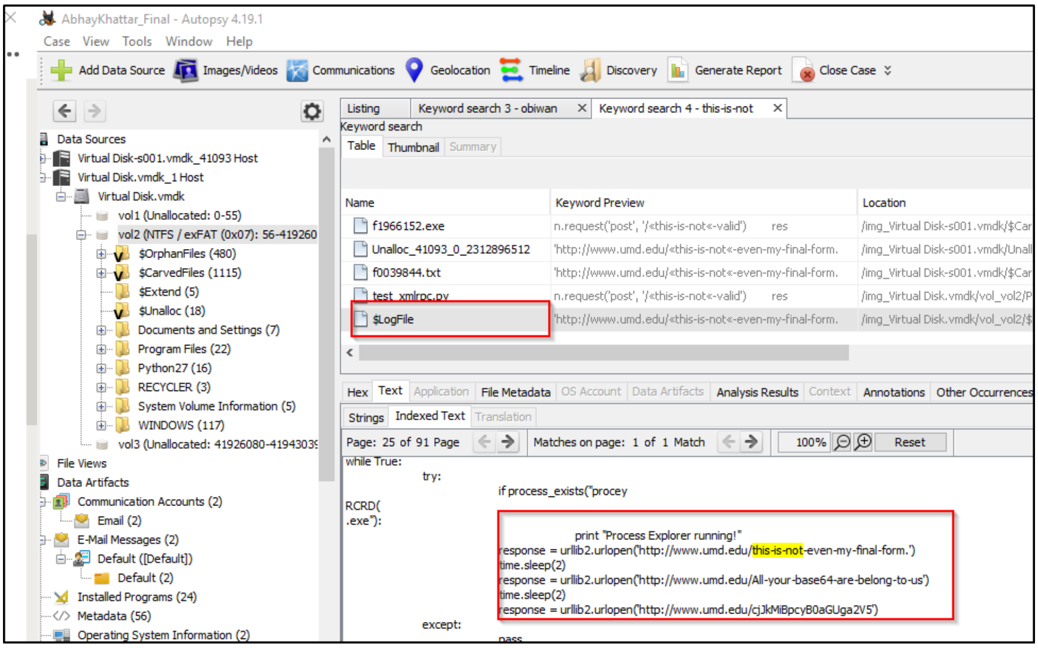
## Analysis of Obiwan.exe



http.request.full_uri == "http://www.umd.edu/help-me-obiwan-kenobi" || http.request.full_uri == "http://www.umd.edu/youre-my-only-hope" Abhay Khattar

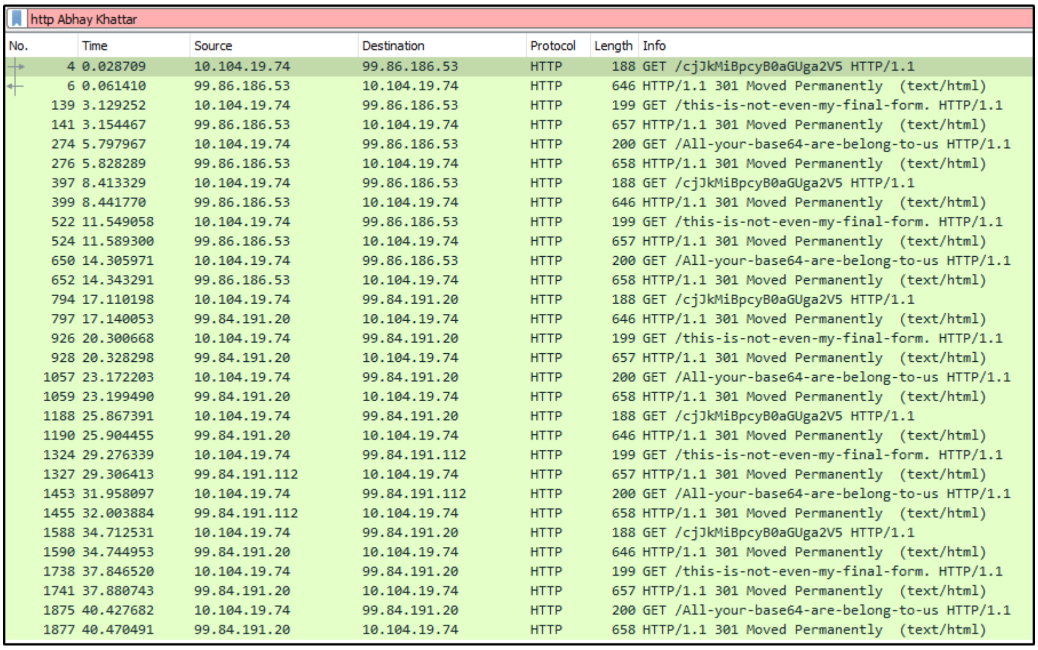
|  | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 210 | 1.650661 | 10.108.30.18 | 13.249.184.111 | HTTP | 189 | GET /help-me-obiwan-kenobi HTTP/1.1 |
| 1428 | 6.504543 | 10.108.30.18 | 13.249.184.111 | HTTP | 186 | GET /youre-my-only-hope HTTP/1.1 |
| 1895 | 9.059481 | 10.108.30.18 | 13.249.184.111 | HTTP | 189 | GET /help-me-obiwan-kenobi HTTP/1.1 |
| 2331 | 11.720607 | 10.108.30.18 | 13.249.184.111 | HTTP | 186 | GET /youre-my-only-hope HTTP/1.1 |
| 3277 | 16.087752 | 10.108.30.18 | 13.249.184.111 | HTTP | 189 | GET /help-me-obiwan-kenobi HTTP/1.1 |
| 3743 | 18.587638 | 10.108.30.18 | 13.249.184.111 | HTTP | 186 | GET /youre-my-only-hope HTTP/1.1 |
| 4322 | 21.264961 | 10.108.30.18 | 13.249.184.111 | HTTP | 189 | GET /help-me-obiwan-kenobi HTTP/1.1 |
| 4753 | 23.858878 | 10.108.30.18 | 13.249.184.111 | HTTP | 186 | GET /youre-my-only-hope HTTP/1.1 |
| 5339 | 26.809774 | 10.108.30.18 | 13.249.184.111 | HTTP | 189 | GET /help-me-obiwan-kenobi HTTP/1.1 |
| 5889 | 29.656016 | 10.108.30.18 | 13.249.184.111 | HTTP | 186 | GET /youre-my-only-hope HTTP/1.1 |

## Analysis of Obiwan2.exe



| Name | Keyword Preview | Location |
|---|---|---|
| f1966152.exe | n.request('post', '/«this-is-not«-valid') res | /img_Virtual Disk-s001.vmdk/$Car |
| Unalloc_41093_0_2312896512 | 'http://www.umd.edu/«this-is-not«-even-my-final-form. | /img_Virtual Disk-s001.vmdk/Unall |
| f0039844.txt | 'http://www.umd.edu/«this-is-not«-even-my-final-form. | /img_Virtual Disk-s001.vmdk/$Car |
| test_xmlrpc.py | n.request('post', '/«this-is-not«-valid') res | /img_Virtual Disk.vmdk/vol_vol2/P |
| $LogFile | 'http://www.umd.edu/«this-is-not«-even-my-final-form. | /img_Virtual Disk.vmdk/vol_vol2/$ |

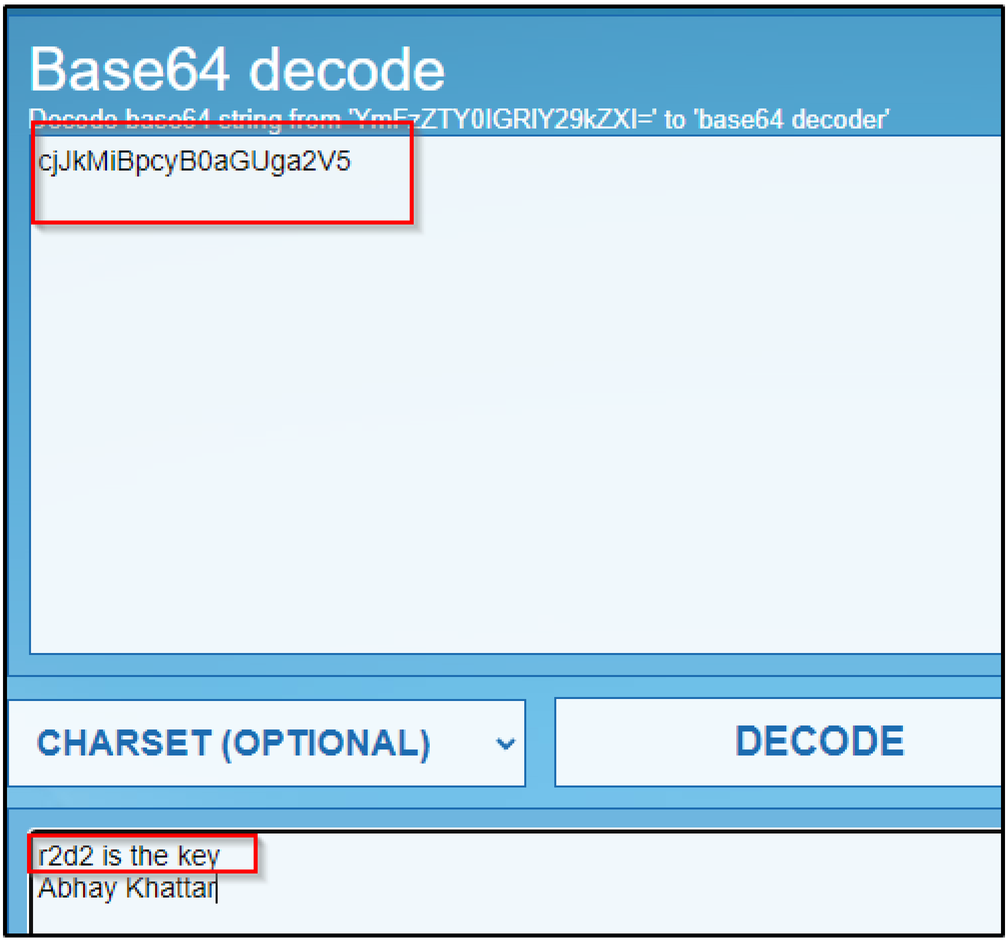
```
while True:
        try:
                if process_exists("procey
RCRD(
.exe"):
                        print "Process Explorer running!"
response = urllib2.urlopen('http://www.umd.edu/this-is-not-even-my-final-form.')
time.sleep(2)
response = urllib2.urlopen('http://www.umd.edu/All-your-base64-are-belong-to-us')
time.sleep(2)
response = urllib2.urlopen('http://www.umd.edu/cjJkMiBpcyB0aGUga2V5')
        except:
                pass
```



http Abhay Khattar

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 4 | 0.028709 | 10.104.19.74 | 99.86.186.53 | HTTP | 188 | GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1 |
| 6 | 0.061410 | 99.86.186.53 | 10.104.19.74 | HTTP | 646 | HTTP/1.1 301 Moved Permanently (text/html) |
| 139 | 3.129252 | 10.104.19.74 | 99.86.186.53 | HTTP | 199 | GET /this-is-not-even-my-final-form. HTTP/1.1 |
| 141 | 3.154467 | 99.86.186.53 | 10.104.19.74 | HTTP | 657 | HTTP/1.1 301 Moved Permanently (text/html) |
| 274 | 5.797967 | 10.104.19.74 | 99.86.186.53 | HTTP | 200 | GET /All-your-base64-are-belong-to-us HTTP/1.1 |
| 276 | 5.828289 | 99.86.186.53 | 10.104.19.74 | HTTP | 658 | HTTP/1.1 301 Moved Permanently (text/html) |
| 397 | 8.413329 | 10.104.19.74 | 99.86.186.53 | HTTP | 188 | GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1 |
| 399 | 8.441770 | 99.86.186.53 | 10.104.19.74 | HTTP | 646 | HTTP/1.1 301 Moved Permanently (text/html) |
| 522 | 11.549058 | 10.104.19.74 | 99.86.186.53 | HTTP | 199 | GET /this-is-not-even-my-final-form. HTTP/1.1 |
| 524 | 11.589300 | 99.86.186.53 | 10.104.19.74 | HTTP | 657 | HTTP/1.1 301 Moved Permanently (text/html) |
| 650 | 14.305971 | 10.104.19.74 | 99.86.186.53 | HTTP | 200 | GET /All-your-base64-are-belong-to-us HTTP/1.1 |
| 652 | 14.343291 | 99.86.186.53 | 10.104.19.74 | HTTP | 658 | HTTP/1.1 301 Moved Permanently (text/html) |
| 794 | 17.110198 | 10.104.19.74 | 99.84.191.20 | HTTP | 188 | GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1 |
| 797 | 17.140053 | 99.84.191.20 | 10.104.19.74 | HTTP | 646 | HTTP/1.1 301 Moved Permanently (text/html) |
| 926 | 20.300668 | 10.104.19.74 | 99.84.191.20 | HTTP | 199 | GET /this-is-not-even-my-final-form. HTTP/1.1 |
| 928 | 20.328298 | 99.84.191.20 | 10.104.19.74 | HTTP | 657 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1057 | 23.172203 | 10.104.19.74 | 99.84.191.20 | HTTP | 200 | GET /All-your-base64-are-belong-to-us HTTP/1.1 |
| 1059 | 23.199490 | 99.84.191.20 | 10.104.19.74 | HTTP | 658 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1188 | 25.867391 | 10.104.19.74 | 99.84.191.20 | HTTP | 188 | GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1 |
| 1190 | 25.904455 | 99.84.191.20 | 10.104.19.74 | HTTP | 646 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1324 | 29.276339 | 10.104.19.74 | 99.84.191.112 | HTTP | 199 | GET /this-is-not-even-my-final-form. HTTP/1.1 |
| 1327 | 29.306413 | 99.84.191.112 | 10.104.19.74 | HTTP | 657 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1453 | 31.958097 | 10.104.19.74 | 99.84.191.112 | HTTP | 200 | GET /All-your-base64-are-belong-to-us HTTP/1.1 |
| 1455 | 32.003884 | 99.84.191.112 | 10.104.19.74 | HTTP | 658 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1588 | 34.712531 | 10.104.19.74 | 99.84.191.20 | HTTP | 188 | GET /cjJkMiBpcyB0aGUga2V5 HTTP/1.1 |
| 1590 | 34.744953 | 99.84.191.20 | 10.104.19.74 | HTTP | 646 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1738 | 37.846520 | 10.104.19.74 | 99.84.191.20 | HTTP | 199 | GET /this-is-not-even-my-final-form. HTTP/1.1 |
| 1741 | 37.880743 | 99.84.191.20 | 10.104.19.74 | HTTP | 657 | HTTP/1.1 301 Moved Permanently (text/html) |
| 1875 | 40.427682 | 10.104.19.74 | 99.84.191.20 | HTTP | 200 | GET /All-your-base64-are-belong-to-us HTTP/1.1 |
| 1877 | 40.470491 | 99.84.191.20 | 10.104.19.74 | HTTP | 658 | HTTP/1.1 301 Moved Permanently (text/html) |

- From these requests, we identify that the strange looking text could be encoded in Base64. We get the following key when we decode the message.



Base64 decode

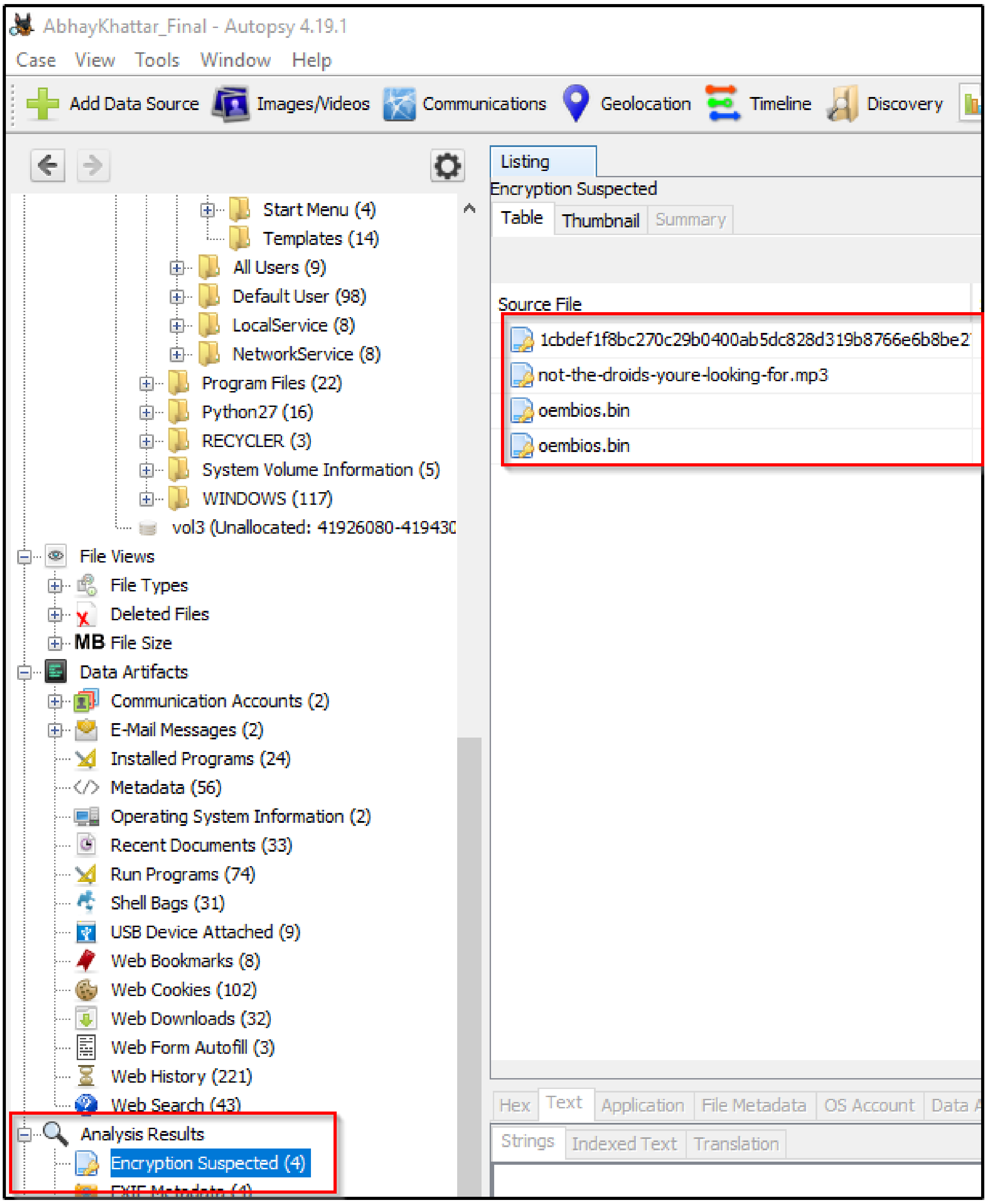
Decode base64 string from 'YmFzZTY0IGRlY29kZXI=' to 'base64 decoder'

```
cjJkMiBpcyB0aGUga2V5
```

```
r2d2 is the key
Abhay Khattar
```

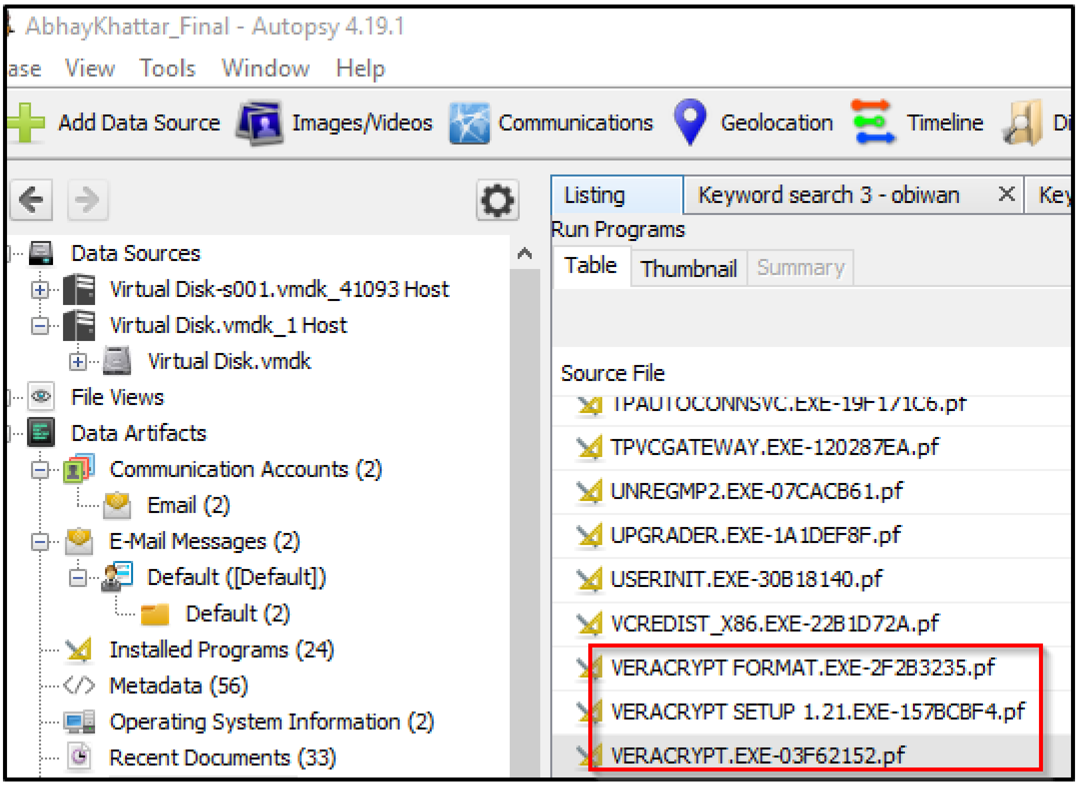
▼ Encrypted Files

Autopsy is able  to detect some files which have a very high entropy and suspected of being encrypted. We extract these files and using the key we found previously we would be able to decrypt the file(s)



| Source File |
|---|
| 1cbdef1f8bc270c29b0400ab5dc828d319b8766e6b8be2 |
| not-the-droids-youre-looking-for.mp3 |
| oembios.bin |
| oembios.bin |

We find VeraCrypt as a part of the WebHistory and also as a an exe which had been installed. We also find multiple images of VeraCrypt. This lead to the conclusion that the files would have been encrypted using VeraCrypt.

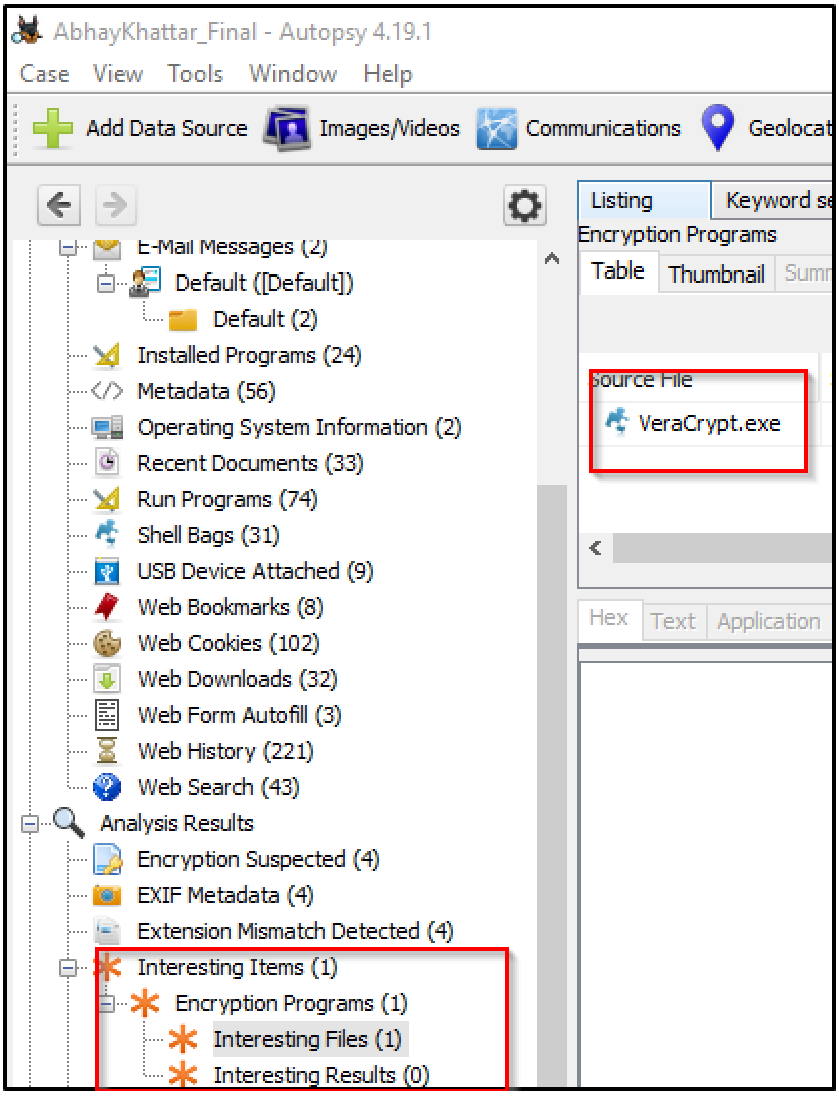


AbhayKhattar_Final - Autopsy 4.19.1

- Data Sources
  - Virtual Disk-s001.vmdk_41093 Host
  - Virtual Disk.vmdk_1 Host
    - Virtual Disk.vmdk
- File Views
- Data Artifacts
  - Communication Accounts (2)
    - Email (2)
  - E-Mail Messages (2)
    - Default ([Default])
      - Default (2)
  - Installed Programs (24)
  - Metadata (56)
  - Operating System Information (2)
  - Recent Documents (33)

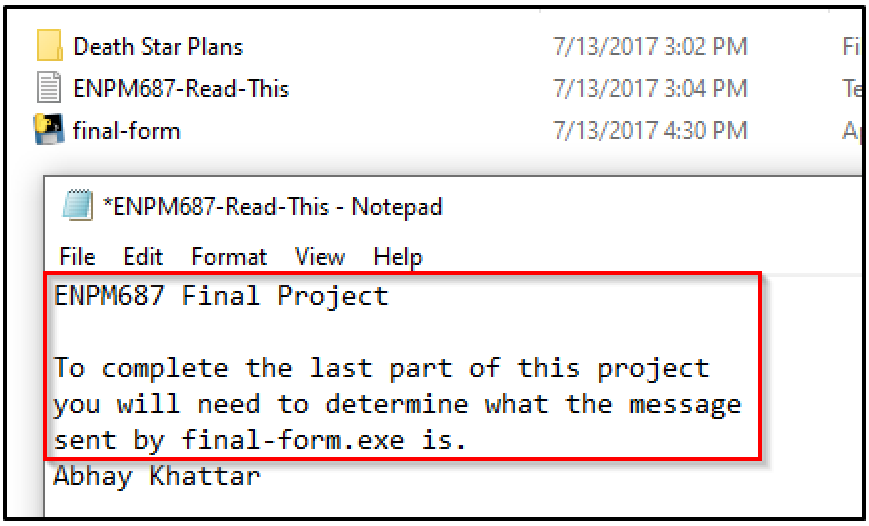
Run Programs

| Source File |
|---|
| TPAUTOCONNSVC.EXE-19F171C6.pf |
| TPVCGATEWAY.EXE-120287EA.pf |
| UNREGMP2.EXE-07CACB61.pf |
| UPGRADER.EXE-1A1DEF8F.pf |
| USERINIT.EXE-30B18140.pf |
| VCREDIST_X86.EXE-22B1D72A.pf |
| VERACRYPT FORMAT.EXE-2F2B3235.pf |
| VERACRYPT SETUP 1.21.EXE-157BCBF4.pf |
| VERACRYPT.EXE-03F62152.pf |



- ▼ Decrypting the Files

  The key *r2d2* is able to decrypt only one particular file *not-the-droids-youre-looking-for.mp3* and we are find the following contents after decryption.

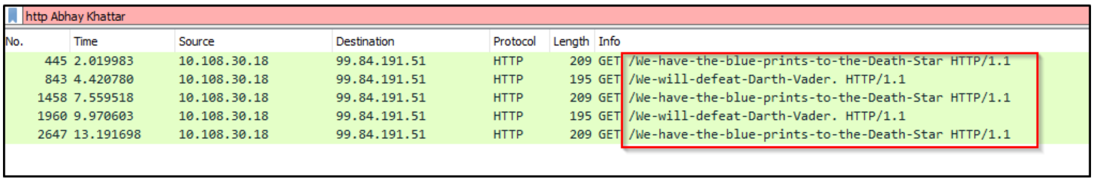


```
Death Star Plans      7/13/2017 3:02 PM   Fi
ENPM687-Read-This     7/13/2017 3:04 PM   Te
final-form            7/13/2017 4:30 PM   A
```

```
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.
Abhay Khattar
```

To find the final message, we run the file and capture the packets using wireshark. There are multiple requests being made which together form the final message.

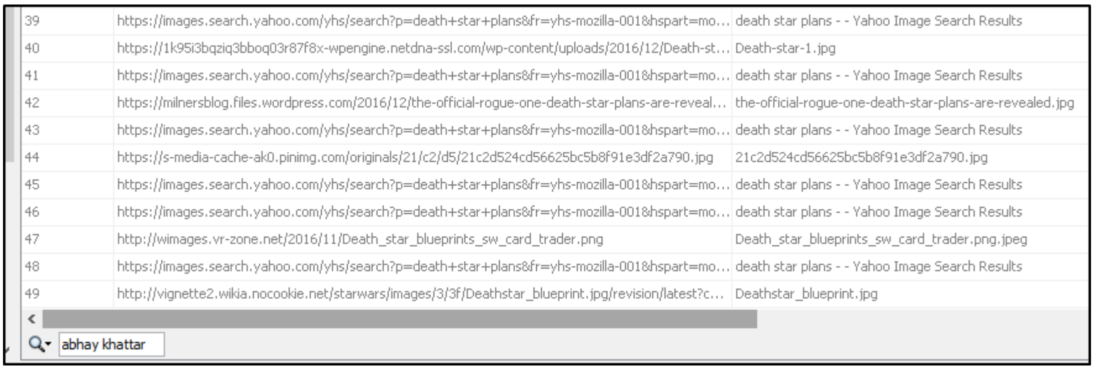


http Abhay Khattar

```
No.    Time       Source        Destination   Protocol Length Info
   445 2.019983   10.108.30.18  99.84.191.51  HTTP     209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
   843 4.420780   10.108.30.18  99.84.191.51  HTTP     195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  1458 7.559518   10.108.30.18  99.84.191.51  HTTP     209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
  1960 9.970603   10.108.30.18  99.84.191.51  HTTP     195 GET /We-will-defeat-Darth-Vader. HTTP/1.1
  2647 13.191698  10.108.30.18  99.84.191.51  HTTP     209 GET /We-have-the-blue-prints-to-the-Death-Star HTTP/1.1
```

### WebHistory

According to the web history, the user had been trying to find the plans of the Death Star.

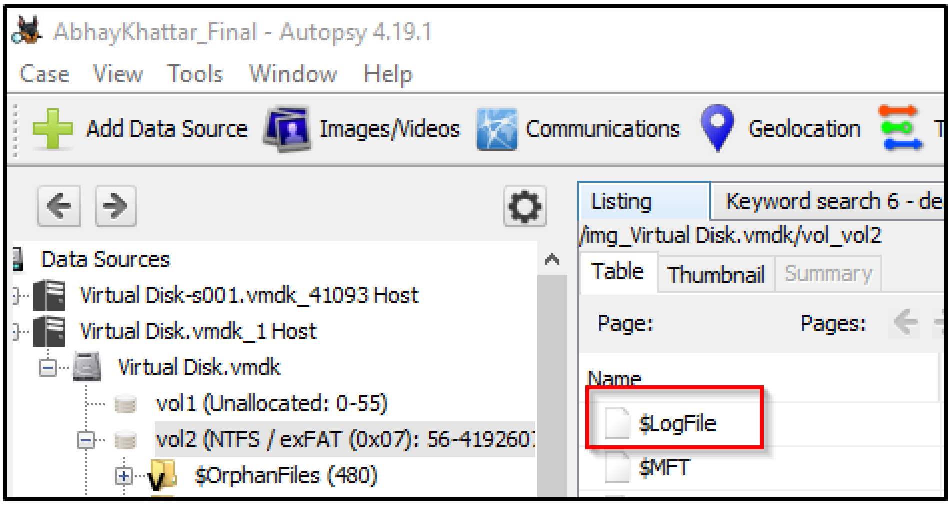


| | | |
|---|---|---|
| 39 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 40 | https://1k95i3bqziq3bboq03r87f8x-wpengine.netdna-ssl.com/wp-content/uploads/2016/12/Death-st... | Death-star-1.jpg |
| 41 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 42 | https://milnersblog.files.wordpress.com/2016/12/the-official-rogue-one-death-star-plans-are-reveal... | the-official-rogue-one-death-star-plans-are-revealed.jpg |
| 43 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 44 | https://s-media-cache-ak0.pinimg.com/originals/21/c2/d5/21c2d524cd56625bc5b8f91e3df2a790.jpg | 21c2d524cd56625bc5b8f91e3df2a790.jpg |
| 45 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 46 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 47 | http://wimages.vr-zone.net/2016/11/Death_star_blueprints_sw_card_trader.png | Death_star_blueprints_sw_card_trader.png.jpeg |
| 48 | https://images.search.yahoo.com/yhs/search?p=death+star+plans&fr=yhs-mozilla-001&hspart=mo... | death star plans - - Yahoo Image Search Results |
| 49 | http://vignette2.wikia.nocookie.net/starwars/images/3/3f/Deathstar_blueprint.jpg/revision/latest?c... | Deathstar_blueprint.jpg |

abhay khattar

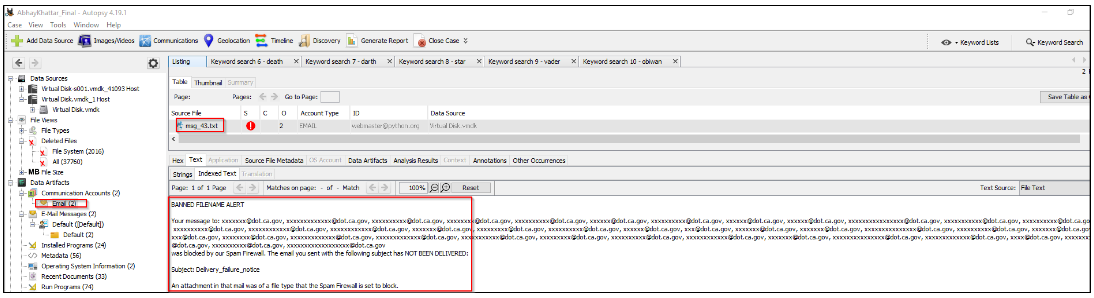
### LogFile

The Log File has a lot of information which would be helpful.



### Emails

We find an email which tells that multiple requests were made to send an email along with an attachment to users with *dot.ca.gov* domains. However, these emails had been blocked by the firewall because the file type of the file was set to block by the Spam Filter.



| Source File | S | C | O | Account Type | ID | Data Source |
|---|---|---|---|---|---|---|
| msg_43.txt | ❗ |  | 2 | EMAIL | webmaster@python.org | Virtual Disk.vmdk |

BANNED FILENAME ALERT

Your message to: xxxxxxx@dot.ca.gov, xxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxx@dot.ca.gov, xxxxxx@dot.ca.gov, xxxxxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.go
xxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxx@dot.ca.gov, xxxxxxx@dot.ca.gov, xxxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxx@dot.ca.gov, xxx@dot.ca.gov, xxxxxxx@dot.ca.gov, xxxxxxx@dot.ca.gov, xxxxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxxx@dot.ca.go
xxx@dot.ca.gov, xxxxxxxx@dot.ca.gov, xxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxxx@dot.ca.gov, xxxxxxxxx@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxx@dot.ca.gov, xxxxxxx@dot.ca.gov, xxxxxxxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxxx@dot.ca.gov, xxxx@dot.ca.gov, xxxxxxxx
@dot.ca.gov, xxxxxxxxxx@dot.ca.gov, xxxxxxxxxxxxxxxxxx@dot.ca.gov
was blocked by our Spam Firewall. The email you sent with the following subject has NOT BEEN DELIVERED:

Subject: Delivery_failure_notice

An attachment in that mail was of a file type that the Spam Firewall is set to block.

## Recommendations

We would recommend removing the malware from the system immediately. Also, since the requests are being sent every 2 seconds, it might be a DOS attack being performed on the websites. The requests made using a python library and should be monitored as they can be used for a malicious purpose. Also, only HTTPS requests should be allowed so that a person is not able to extract infomration even if they are sniffing the network.

## Problems Faced

1. It took some time to understand how to decrypt the file using veracrypt
2. It took some time to understand that I had to capture the packets using wireshark to get the requests.