INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

MEIC / METI 2018-2019 – 1st Semester

Digital Forensics Report

Daniel Nunes – 81016 **Diogo Freitas - 81586**

1 Can you determine how the malware has taken over Sally's computer?

In the beginning we thought that the malware might have taken over Sally's computer through the ImageJ software. To check this hypothesis, we used the **fls** tool - **Figure 1**, to pass between the disk's inodes and swept through the disk's files on the ImageJ installation directory, after finding no incriminating evidence or indication that the virus' origin was the ImageJ software, we discarded our first theory.

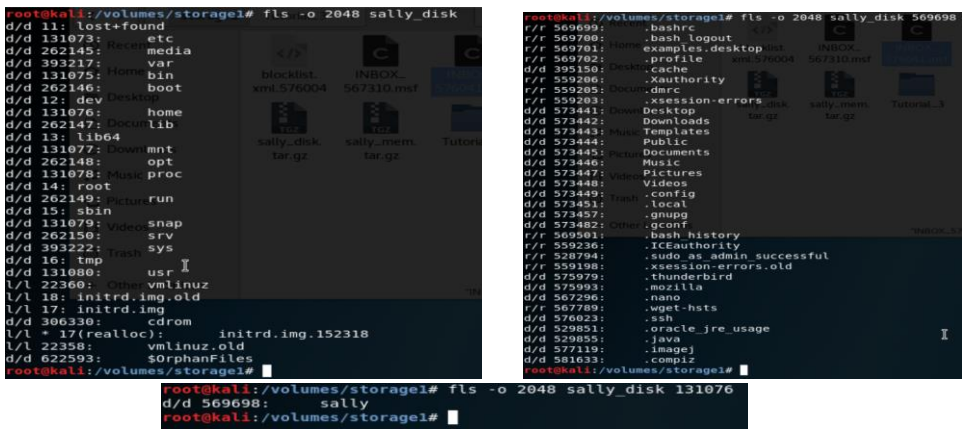

Figure 1

As such, we checked the remaining steps Sally took before the virus manifested itself in her computer. By doing this we discovered Sally checked her email before it all happened, so we searched the disk for evidence of the virus on the mail application Sally had installed in her computer, Mozilla Thunderbird. Mozilla Thunderbird is a free and open-source cross-platform email client developed by the Mozilla Foundation. When we entered into Thunderbird inode we found several folders and files. One of them, INBOX (inode 529856) - **Figure 2** corresponding to a .mbox file contained emails of jason_halloween@protonmail.com, the same email as the one where Sally should send the payment to retrieve her files. To see the content of the email we searched how the .mbox file was used in the Thunderbird application and found out that, paired with an .msf file (INBOX.msf – inode 567310)- **Figure 3** and inserted into our Thunderbird application we could gain access to the email's contents. By doing so we found out the email in question sent by jason_halloween@protonmail.com was sent with a file named "main" that we suspected to be the virus- **Figure 4**.

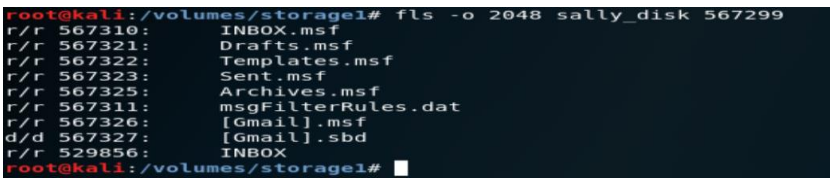

Figure 2

```
root@kali:/volumes/storage1# icat -o 2048 sally_disk 567310 > INBOX_567310.msf
root@kali:/volumes/storage1#
```

Figure 3

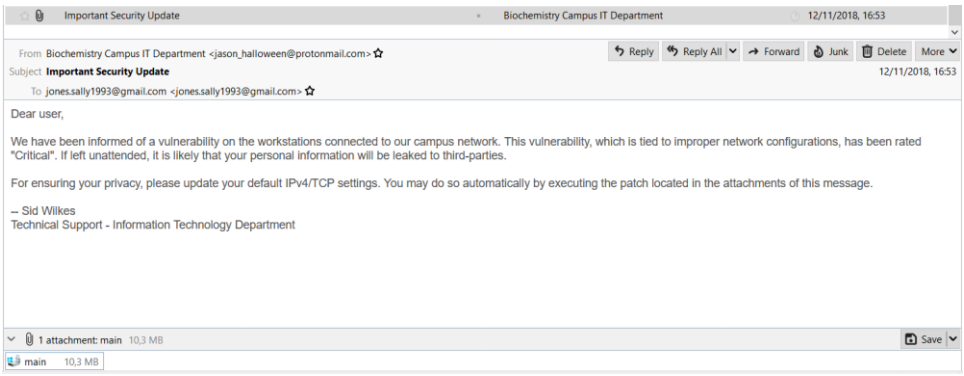

Figure 4

To check if this file was indeed downloaded by Sally to her computer, we saw the content of downloads.json (inode 529888)- **Figure 4** and we observed a download of this file "main" at 17:14, 2018-11-12 to the directory /home/sally/Downloads/ we noticed the file "main" had been deleted - **Figure 6**. As we had already gained access to the email's contents, the fact that the file had been deleted didn't prove a problem as we downloaded it from the Thunderbird application. Once we opened this file, we understood that it was indeed the virus as the same pop-up message that manifested in Sally's computer appeared on our screens- **Figure 7**.


```
{"list":[{"source":"imap://jones%2Esally1993%40gmail%2Ecom@imap.gmail.com:993/fetch%3EUID%3E/INBOX%3E22?
part=1.2&filename=main","target":"/home/sally/Downloads/
main","startTime":"2018-11-12T17:14:02.023Z","succeeded":true,"contentType":"application/octet-stream"}]}
```


Figure 5

```
root@kali:/volumes/storage1# fls -o 2048 sally_disk 573442
d/d 576046:     LiME
r/r * 529882(realloc):  main
d/d 529881:     Fiji.app
```

Figure 6

Figure 7

Finally, we used the volatility tool to search through the memory of Sally's computer for the process of the virus (main) and its process id. We used the command:

python vol.py -f sally_mem --profile=LinuxUbuntu160405x64 linux_pslist

As we went through the output of the command, we ran we found the process main we were looking for (PIDs: 14919 and 14921) proving that file ran on Sally's PC at 17:15, one minute after being installed - **Figure 8**.

```
0xffff91d5b56b5b00 main                14919          1211           1000           1000    0x00000000317ce00
0 2018-11-12 17:15:45 UTC+0000
0xffff91d5b56b0000 main                14921          14919          1000           1000    0x000000002fb0a00
0 2018-11-12 17:15:45 UTC+0000
```

Figure 8

FILE	MD5 VALUE
main	324DDC336159DD62E182E3ABF12C9B0A
INBOX_567310.msf	02713A86AD1D35DDC732E7B7E34A1462

2 Can you recover Sally's original files? If you do not succeed at retrieving the original files, can you at least extract some of its fragments?

On Sally's disk, on the directory /home/Documents, we found the results of the action of the virus, all of Sally's work was encrypted and the original files replaced with files containing the message "Jason's back!" - **Figure 9**. As such, and as we knew the algorithm that was used to encrypt the files (AES in AES_CTR_Mode with a key of 128bits and the counter being placed on the first 128bits of each encrypted file) we proceeded to elaborating a script that would decrypt these files so Sally could recover them.

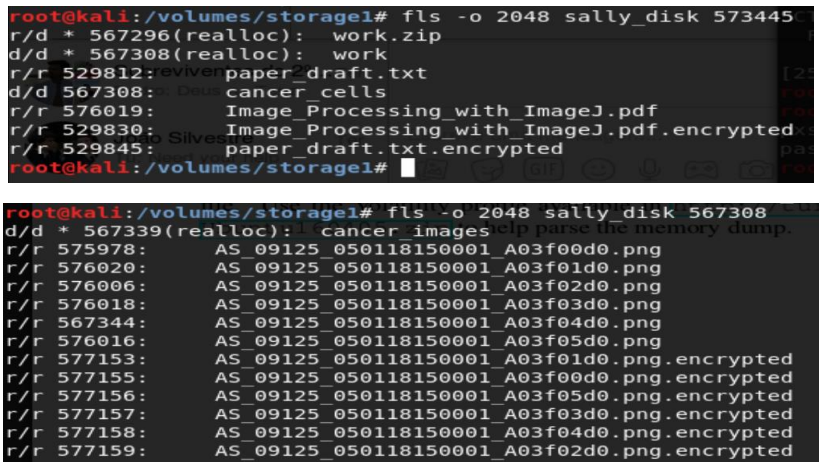

Figure 9

To do this, we first needed to find the key that was used for the algorithm so we could use it to reverse what had been done to Sally's files. To find this key we used the command:

python vol.py -f sally_mem --profile=LinuxUbuntu160405x64 linux_yarascan –D dump_files –profile=LinuxUbuntu16045x64 –yara-rules="jason" --pid=14921 > jason_search.out

We then searched the output of the command for any appearance of what might resemble a key to the AES algorithm, to which the results proved fruitful as we found a reference to a file key.txt preceded by an echo command of the key – **Figure 10**. We then proceeded we developing the script that would use the key we found to decrypt Sally's encrypted files – **Figure 11**.

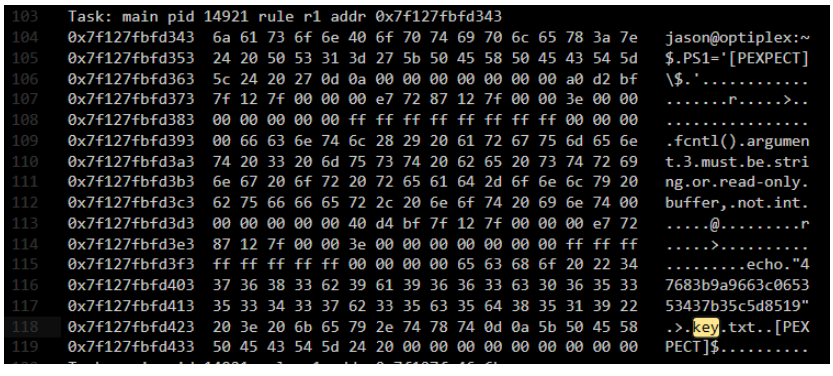

Figure 10

```python
#Decrypt AES - CTR_MODE 128bit Key Size

import sys, struct
from Crypto.Cipher import AES
from Crypto.Util import Counter

key = '47683b9a9663c065353437b35c5d8519'

def decrypt_aes(filename):
    file = open(filename, 'rb')
    data = file.read()
    file.close()

    ctr_hex = data[0:16].encode('hex')
    counter = int(ctr_hex, 16)

    data_hex = data[16:].encode('hex')

    cipher = AES.new(key.decode('hex'), AES.MODE_CTR, counter=Counter.new(128, initial_value=counter))
    decoded_data = ''
    for i in range(0, len(data_hex), 32):
            data_block = data_hex[i:i+32].decode('hex')
            dec_block = cipher.decrypt(data_block)
            decoded_data += dec_block

    tmp = filename.split('.')
    new_filename = tmp[0] + '.' + tmp[1]
    new_file = open(new_filename, 'wb')
    new_file.write(decoded_data)
    new_file.close()

if __name__ == '__main__':
    decrypt_aes(sys.argv[1])

```

Figure 11

Once the script was completed, all that was left to do was to use the **icat** command to get the encrypted files from Sally's disk and run them through our script to get the original files – **Figure 12**.

```
root@kali:/volumes/storage1# fls -o 2048 sally_disk 573445
r/d * 567296(realloc):  work.zip
d/d * 567308(realloc):  work
r/r 529812:     paper_draft.txt
d/d 567308:     cancer_cells
r/r 576019:     Image_Processing_with_ImageJ.pdf
r/r 529830:     Image_Processing_with_ImageJ.pdf.encrypted
r/r 529845:     paper_draft.txt.encrypted
root@kali:/volumes/storage1# icat -o 2048 sally_disk 529830 > Image_Processing_with_ImageJ.pdf.encrypted
root@kali:/volumes/storage1# python decrypt_script.py Image_Processing_with_ImageJ.pdf.encrypted
root@kali:/volumes/storage1#
```

Figure 12

FILE	MD5 VALUE
AS_09125_050118150001_A03f00d0.png	B58303DD6F4026663FB1AACACCF5BF94
AS_09125_050118150001_A03f01d0.png	1E33B87269C463474F68DF10D95EB67B
AS_09125_050118150001_A03f02d0.png	DEFA8C84D13338CF83668CF44CCBE016
AS_09125_050118150001_A03f03d0.png	32DE7CAAAC1E191FEBE5C7E4D48C839A
AS_09125_050118150001_A03f04d0.png	1A6093F96040770A97DD257A3D487231
AS_09125_050118150001_A03f05d0.png	F75BAF3C3F4E06D14355133A6EDAE13B
Image_Processing_with_ImageJ.pdf	23F432689A13006CFE0E982F8AE71459
paper_draft.txt	AA4D4B8006C1941FFA3684F26747B696

3 What can you tell about the identity of the attacker?

After analysing all files, we found a possible IP of the attacker: 185.70.40.136 on the file INBOX (inode 529856). Then, we used the site cua-es-mi-ip to locate the IP in map and we realized it came from switzerland. Altough we saw later that this IP was from the mail server used to send the virus: ProtonMail.

ProtonMail is located in Switzerland and no personal information is necessary to create the mail account, not saving the Client IP. So, the location of the attacker remains unknown.

Figure 13

4 Elaborate a timeline of the most significant events of the case.

16:53, 12th November 2018 – The email was received from the address jason.halloween@protonmail.com containing the file main;

17:14, 12th November 2018 – The email was opened moments prior to this timestamp and the download of the file main contained in the email was initiated;

17:15, 12th November 2018 – The download of the file main was completed and Sally ran this file which caused the malware to take action on her files, encrypting them and displaying the pop-up mentioned earlier on this report. Finally, the file main was deleted automatically.