



# LAB BOOK PORTFOLIO #3

## Advanced Computer Security

Date: 2022-04-02

Lab Book Portfolio #3 | Advanced Computer Security | Course Code: INFO2231

Sunraj Sharma  
Sunraj751@gmail.com

**Table of Contents**

|  |           |
|--|-----------|
| <b>Table of Contents .....</b>                     | <b>1</b>  |
| <b>Lab 6 - Customize a Password List .....</b>     | <b>2</b>  |
| <b>Part 1 .....</b>                                | <b>2</b>  |
| Description.....                                   | 2         |
| Preparation .....                                  | 2         |
| Observations .....                                 | 2         |
| Screenshots.....                                   | 3         |
| Reflection.....                                    | 4         |
| <b>Lab 7 – Hack Passwords Offline.....</b>         | <b>4</b>  |
| <b>Part 1 .....</b>                                | <b>4</b>  |
| Description.....                                   | 4         |
| Preparation .....                                  | 5         |
| Observations .....                                 | 5         |
| Screenshots.....                                   | 5         |
| Reflection.....                                    | 6         |
| <b>Lab 8 – Legion and Metasploit Console .....</b> | <b>7</b>  |
| <b>Part 1 .....</b>                                | <b>7</b>  |
| Description.....                                   | 7         |
| Preparation .....                                  | 7         |
| Observations .....                                 | 7         |
| Screenshots.....                                   | 7         |
| Reflection.....                                    | 9         |
| <b>References .....</b>                            | <b>10</b> |

## Lab 6 - Customize a Password List

### Part 1

#### Description

This lab's purpose was to enable us to explore password requirements for websites, and create our own password list. In order to customize our own password list, we are utilizing the SecList password list resources. With these created password lists, we will use them in an attack, which changes password lists using tools such as mentalist and wordlister.

#### Preparation

In order to prepare for this lab, I went through the preparation portion of the lab document, so first I ran the command `git clone https://github.com/danielmiessler/SecLists` and then moved the contents into the `/usr/share/wordlists`. After, I ran the command `git clone https://github.com/sc0tfree/mentalist` from the `/home/kali` directory. I then followed the document further and ran the `cd /home/kali/mentalist; sudo python3 setup.py install` command. Lastly, I ran the `wget cd /home/kali/mentalist; sudo python3 setup.py install` command.

#### Observations

- (Task #1) After looking at various sites, I decided to look into twitter and their account policies. According to PasswordPit, Twitter requires that users who register for their platform have a minimum password length of 6 characters (PasswordPit, 2015). There are no other requirements such as numbers, symbols, upper-case, lower-case, among other things. Twitter only requires 6 character length password.
- (Task #2) I then changed directories as specified in the Lab 6 document and ran the command `python3 wordlister.py`. After, I was able to see the various command options.
- After, I ran the command in the terminal in order to generate my password list that consisted of a minimum of 6 characters and made sure that the name list was named the proper name.
- I then launched mentalist, and did the necessary changes. I used the plus sign in order to add other "manglers". I appended a "small" subset of numbers ranging from 0-100 according to the mentalist tool.
- Finally, I then sorted the file and removed the duplicated by running the command in the document. After I showcased the output of the file (check screenshots)

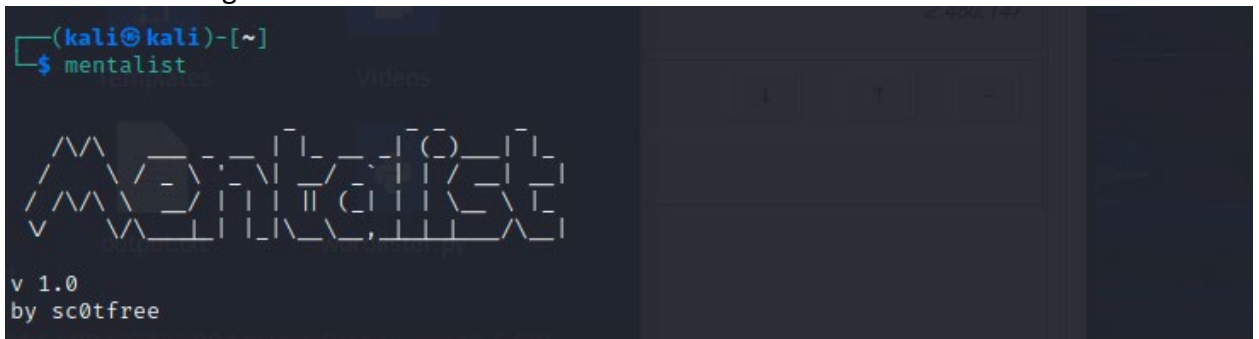
## Screenshots

- Running command to generate list



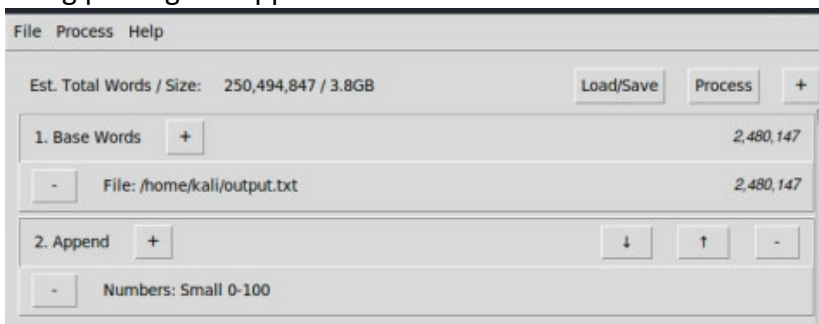
```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ python3 wordlister.py --input /usr/share/wordlists/SecLists/Passwords/probable-v2-top1  
575.txt --perm 2 --min 6 --max 0 > /home/kali/bb0334-wordlisted-pws.txt  
(kali@kali)-[~]  
$ python3 wordlister.py --input /usr/share/wordlists/SecLists/Passwords/probable-v2-top1  
575.txt --perm 2 --min 6 --max 32 > /home/kali/bb0334-wordlisted-pws.txt
```

- Mentalist running.



```
(kali@kali)-[~]  
$ mentalist  
  
v 1.0  
by sc0tfree
```

- Using plus sign to append.



- (Task #4) and (Task #5.) sorting and then using md5sum command.

bb0334-Kali [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali: ~

File Actions Edit View Help

```
(kali@kali)-[~]
$ sort bb0334-after-mentalist-pws.txt | uniq -u >bb0334-finalpws.txt

(kali@kali)-[~]
$

(kali@kali)-[~]
$ head bb0334-finalpws.txt
00000000
0000000000
00000000000
000000000000
0000000000000
00000000000000
000000000000000
0000000000000000
00000000000000000
000000000000000001
00000000000000010

(kali@kali)-[~]
$ last bb0334-finalpws.txt

wtm begins Fri Feb 11 18:25:18 2022

(kali@kali)-[~]
$ wc bb0334-finalpws.txt
249171417 249171417 4106289391 bb0334-finalpws.txt

(kali@kali)-[~]
$ md5sum bb0334-finalpws.txt
4241f9f59deddeeb7a4fb6eb787362fd bb0334-finalpws.txt
```

## Reflection

- 1.) While doing this lab 6, I ran into some interesting things in regards to the mentalist tool, and generating the list of passwords. I found it interesting how the mentalist had so many different ways to generate lists, and how large the files could become. I found that some could have went into the hundreds of Gigabytes if you appending certain items to the password list. Overall, I had problems downloading everything because of my poor internet.

## Lab 7 – Hack Passwords Offline

## Part 1

### Description

The purpose of Lab 7 Hack Passwords Offline is to build off of what was already completed in Lab 6 Customize a Password List. In Lab 7, the point is to be using a tool called the “john” tool. It is an offline password hash cracking tool, which will be explored and used in the lab in order to try to find passwords within a password file.

### Preparation

To prepare for this lab, I followed the instructions laid out within the Lab 7 Hack Passwords Offline document. There were not many requirements stated for preparation in the document, but I did firstly ensure that I was running the appropriate and up to date Kali Linux Virtual Machine. I also made sure that my metasploitable2 virtual machine was also in working order before attempting to start the lab.

### Observations

- I became the root user for the metasploitable virtual machine. After, I then ran the cat command in order to display all of the passwords I created in lab 6. As the lab explained, I broke out of the list populating and picked on of the passwords on display, which was 00000000cannabis3.
- As the root user in the metasploitable virtual machine, I then ran the command cat /etc/shadow to show a list of all the of star wars characters. After, I changed the password for darth\_vader and changed it to 00000000cannabis3. After I changed it with the passwd darth\_vader command, I once again ran the cat /etc/shadow to list all of the characters again. I compared the hash string, and it was different for darth\_vader after I changed the password.
- I then copy and pasted the file so I could have a local copy.
- After that, I ran the "john" tool. I ran the tool against the password file created in order to decrypt the password file that I was originally using. I did this and paid specific attention to the user (darth\_vader) that I previously changed the password for in the other tasks.

### Screenshots

- Changing password for existing star wars characters.

```

File Machine View Input Devices Help
syslog:*:16176:0:99999:7:::
messagebus:*:18564:0:99999:7:::
sshd:*:18564:0:99999:7:::
statd:*:18564:0:99999:7:::
vagrant:$6$NABMMgx0$Tz1vEhAr.j0Im.jvR0ySg8vka/r8MWhhzNgT3Z5FS1LcPS5D325ESK5LjFJymb
2.jo/m4NmDg8aE10TMMI3la.Y3/:18564:0:99999:7:::
lrangr:*:18564:0:99999:7:::
leia_organa:$1$N6D1bGGZ$LPERCrfi8IXiNebhQuYlK/:18564:0:99999:7:::
luke_skywalker:$1$/7D550zb$Y/aKb.UNrDS2w7n2Uq.L1/:18564:0:99999:7:::
han_solo:$1$6jIF3qTC$7jEXfQsNENuWYe06cK7m1.:18564:0:99999:7:::
artoo_detoo:$1$tfvzyRnu$maunXAR4GyABt8rtn7Df.v.:18564:0:99999:7:::
c_three_pio:$1$1x7tKuo$xuM4AxkByTUD78BaJdYdG.:18564:0:99999:7:::
ben_kenobi:$1$5nfRD/bA$y7ZZD0NimJThX9FtuhHJX1:18564:0:99999:7:::
darth_vader:$1$rLuMkR1R$YHumHBxhsunf07eTUufHJ.:18564:0:99999:7:::
anakin_skywalker:$1$j1peszLc$PW4IPiuLTwiSH5YaTIRaB0:18564:0:99999:7:::
jar_jar_binks:$1$SNokFi0c$F.Su.j2QjYRSuoBuobRUMh1:18564:0:99999:7:::
lando_calrissian:$1$Af1ek3xT$nkC8.jk30gMQWw/6.ono0:18564:0:99999:7:::
boba_fett:$1$TjxlmU4.j$K/rG1vb4.p.j.z0yFWJ.ZD0:18564:0:99999:7:::
jabba_hutt:$1$9rpNcs3v$//v21t.j5MYhfUOHYVaz.jD/:18564:0:99999:7:::
greedo:$1$vuU.f3T.j$tsGBZJbBS4JwchsbUW0a1:18564:0:99999:7:::
cheebacca:$1$.qt4t8zH$RdKbdafugc7rYiDXSoQCI.:18564:0:99999:7:::
xylo_ren:$1$rpuxsssI$H0BC/qL92d0GgmD/uSELx.:18564:0:99999:7:::
mysql:!:18564:0:99999:7:::
avahi:*:18564:0:99999:7:::
colord:*:18564:0:99999:7:::
root@metasploitable3-ub1404:/home/vagrant# passwd darth_vader
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@metasploitable3-ub1404:/home/vagrant#

```

- Copy the files so I have a local copy

```

darth_vader:$6$/heXmZ8z$KA0cIRC7EHQ/xn91r9XCPaxy4BznQ6G20T4p7gU8uo5PvcN20yVafN3g
darth_vader:$6$/heXmZ8z$KA0cIRC7EHQ/xn91r9XCPaxy4BznQ6G20T4p7gU8uo5PvcN20yVafN3g

```

## Reflection

- 1.) The lab had various interesting moments. After creating the password list in the 6<sup>th</sup> lab, using the metasploitable virtual machine in order to gain access to the password list and then change the password for a specific star wars character was interesting. I think this lab showcased how simple it can be to change the password and consequentially, change the hash along with it. I did not personally have any difficult following the instructions of this lab.
- 2.) An alternative for generating hashes and then checking them against the what is in the file, just like the john tool, could be the THC Hydra tool. After researching online, apparently the THC Hydra tool is a great alternative to the John tool, according to LinuxSecurity.expert.

## Lab 8 – Legion and Metasploit Console

### Part 1

#### Description

The purpose of Lab 8 – Legion and Metasploit Console is to use the legion tool in conjunction with the Metasploitable3 Virtual Machine in order to search for possible places to attack on the Metasploitable3 virtual machine. The lab will be utilizing the Metasploit Console by essentially configure and attack one of the vulnerable areas.

#### Preparation

In order to prepare for Lab 8 – Legion and Metasploit Console, I followed the preparation instructions laid out in the Lab 8 document. I first ensured that I had my Kali Linux Virtual machine running correctly and that there was no issues in it running, then, I checked out and made sure that my Metasploitable 3 virtual machine was in working order. Finally, I followed the instructions set out in the lecture and the link provided so I could setup the Legion tool. I was then done preparing after everything was installed and was working.

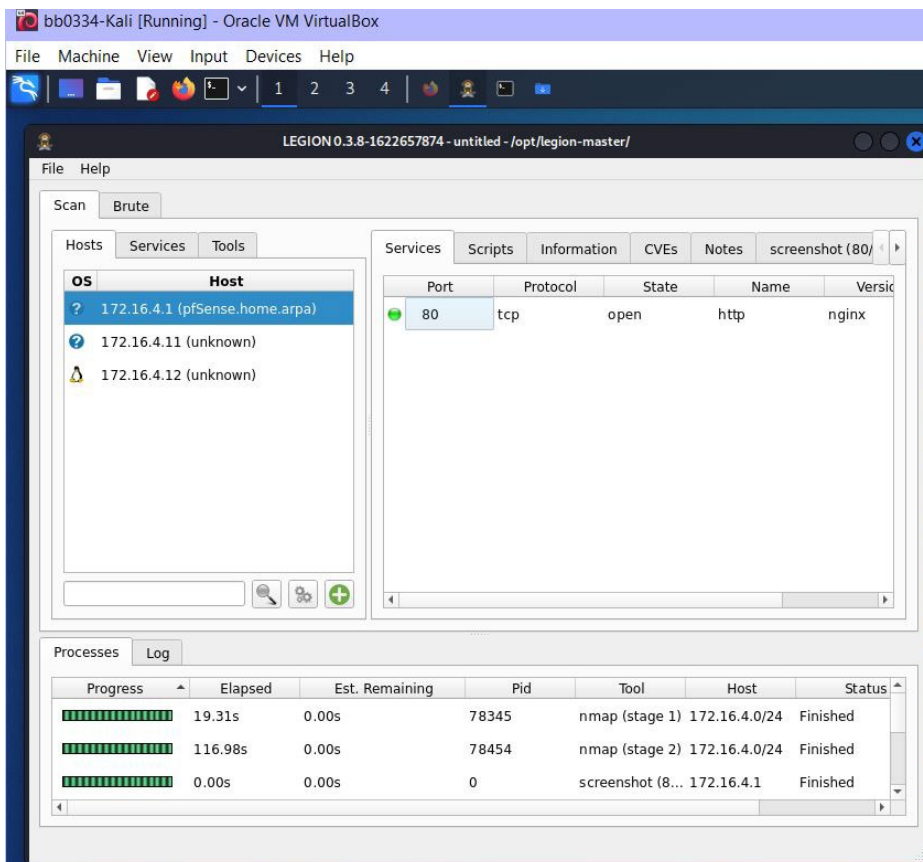
#### Observations

- For the first task, I first installed everything that had to do with legion, afterwards, I ran legion and did a scan of my virtual machines put inserting the ip 172.16.4.0/24. After, the scan showed results under the “hosts” tab.
- I looked for various ways to do an attack, so I chose to target the ssh and do an attack on that. In order to do this, I ran ssh [vagrant@172.16.4.12](#) from kali, and I was able to gain access to the metasploitable3 virtual machine by doing this.
- I then ran command sudo su, followed by the command cat /etc/shadow, and I was able to get the list of hashes that was found with the metasploitable3 virtual machine from my kali linux virtual machine.
- I then went back again, and used the msfconsole to search for possible known attacks and I then used of the known attacks. After, I configured the options, along with the payload, in order to launch the attack.

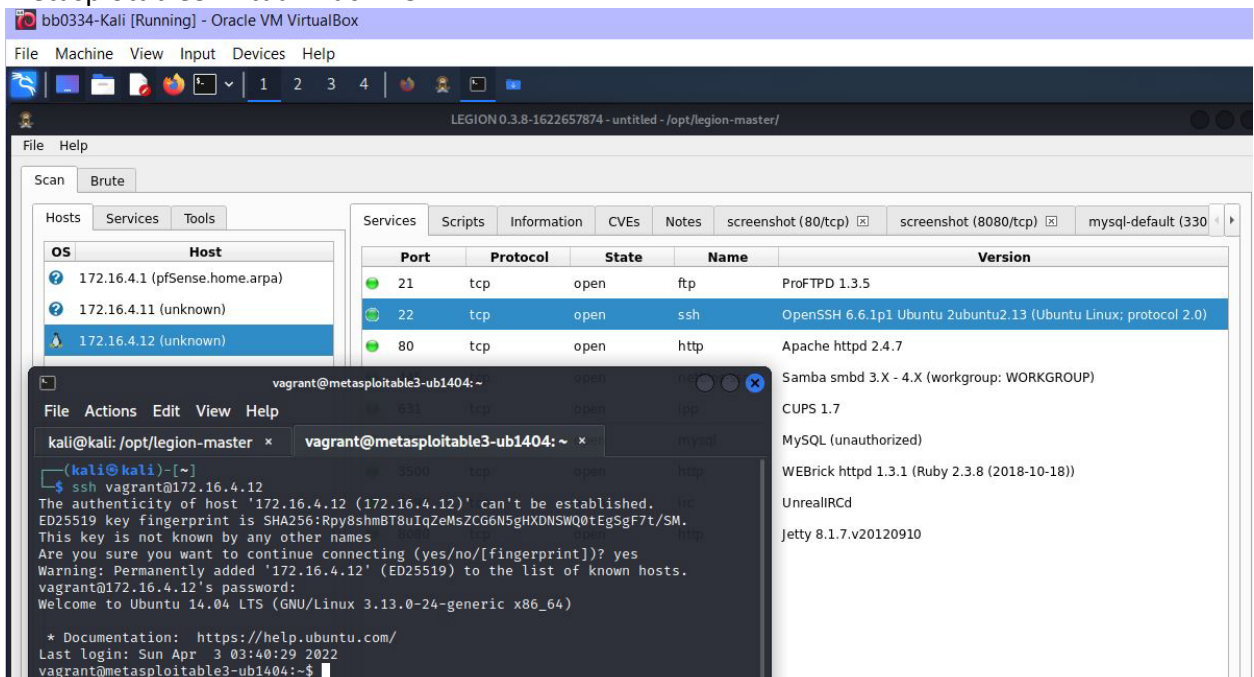
#### Screenshots

- Create a legion scan of both virtual machines

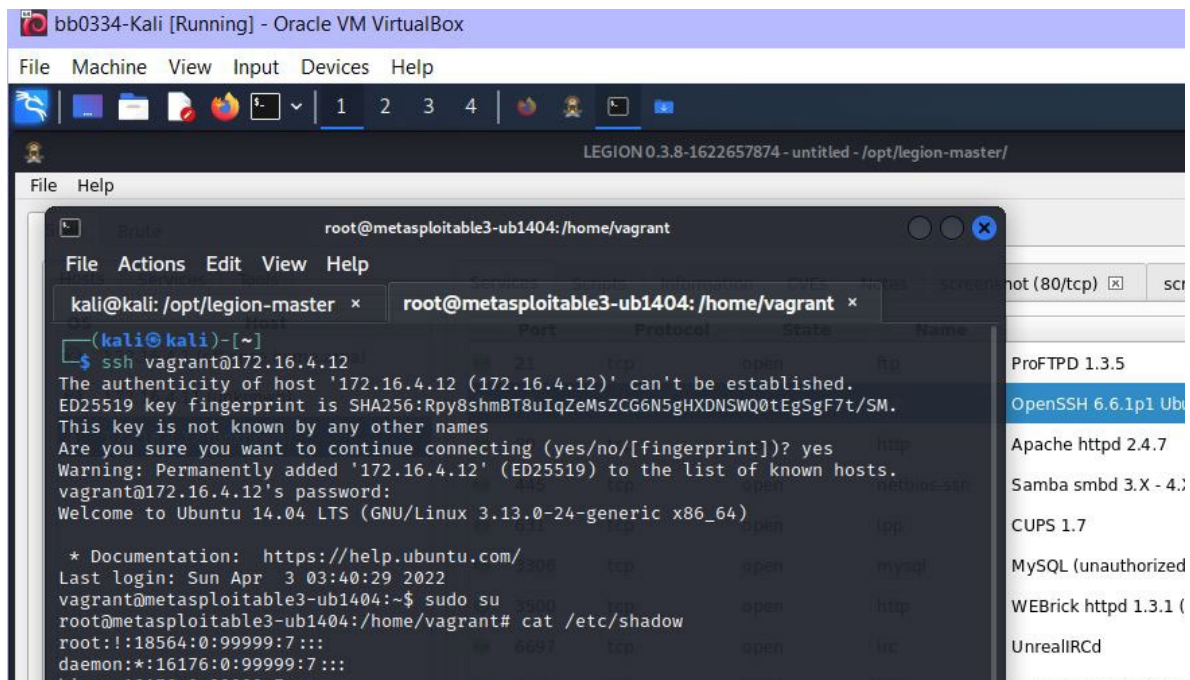




- Locate potential area for attack. Running the attack and gaining access to the metasploit3 virtual machine



- Getting access to the hashes from metasploit3 from my kali linux VM



## Reflection

- 1.) I noticed that there were a variety of different things that appeared after I ran the legion tool and scanned the IP. Under the services tab in the legion tool, it showed a list of ports of which all of them were open. Port 21, 22, 80, 445, 631, 3306, 3500, 6698, and 8080 were open. During the lab, I decided to attack the OpenSSH which was under port 22. The legion tool said it was using TCP protocol and that the state was 'open'. The most difficult part of the lab was installing everything as my internet is horrible.
- 2.) I did some further research into the exploit I decided to use. Since I chose the exploit OpenSSH, I was able to find some information on what exactly that is.

## **References**

PasswordPit. (2015, June 8). What are Twitter's password requirements? PasswordPit. Retrieved April 2, 2022, from <https://www.passwordpit.com/twitter-password-requirements/>