SCRIPT FLOWCHART

flowchart.JPG

MAIN PROGRAM

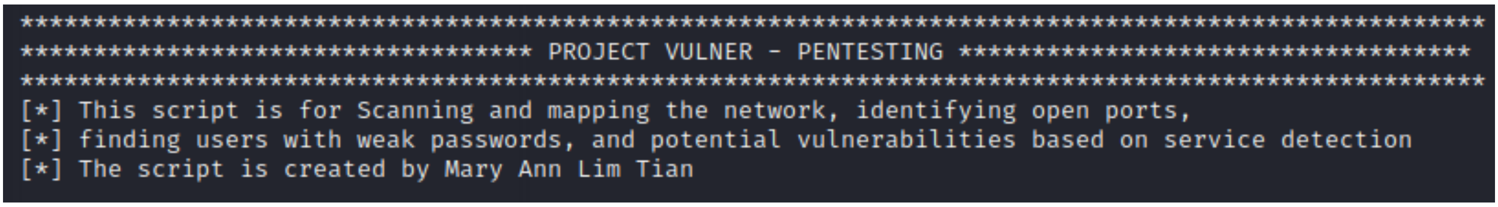
```
#START OF SCRIPT TO RUN
#call FUNCTION 14 - Introduction message of the script
intro_message
#call FUNCTION 13 - group of functions to execute this script
call_function
```

FUNCTION 1 - Introduction message of the script

```
function intro_message()
{
# Set the message to display
message=" PROJECT VULNER - PENTESTING "

# Set the width and height of the banner
width=100
height=5

# Define the character used for the banner border
borderChar="*"

# Calculate the length of the message and the left and right padding needed
messageLength=${#message}
paddingLength=$(( (width - messageLength) / 2 ))
leftPadding=$(printf "%0.s${borderChar}" $(seq 1 $paddingLength))
rightPadding=$(printf "%0.s${borderChar}" $(seq 1 $paddingLength))

# Print the top border of the banner
printf "%0.s${borderChar}" $(seq 1 $width)
echo ""

# Print the message with padding
echo "${leftPadding}${message}${rightPadding}"

# Print the bottom border of the banner
printf "%0.s${borderChar}" $(seq 1 $width)
echo ""

echo "[*] This script is for Scanning and mapping the network, identifying open ports,"
echo "[*] finding users with weak passwords, and potential vulnerabilities based on service
detection"
echo -e "[*] The script is created by Mary Ann Lim Tian \n"
```

RESULT

OUTPUT EXPECTED:



```
*****************************************************************************************************
*********************************** PROJECT VULNER - PENTESTING *************************************
*****************************************************************************************************
[*] This script is for Scanning and mapping the network, identifying open ports,
[*] finding users with weak passwords, and potential vulnerabilities based on service detection
[*] The script is created by Mary Ann Lim Tian
```

FUNCTION 2 - call_function group of functions to execute this script

```
function call_function()
{
        #start timestamp
        Nmap_Device_StartTimestamp=$(echo "$(date '+%D %r') - ")
    #Call FUNCTION 1 - log creation
        log_creation
        #call FUNCTION 2 - Host Discovery
        ip_range_info
        #call FUNCTION 3 - to check IP address entry if correct or not
        TargetIP_check
        #call FUNCTION 6 - check if user has own username list or to create
        user_list_option
        #call FUNCTION 7 - check if user has own password list or to create
        pass_list_option
        #call FUNCTION 8 - TCP Scan for open ports ,services and OS detection info
        nmap_tcp_scan
        #call FUNCTION 9 - UDP Scan for open ports
        udp_scan
        #call FUNCTION 10 - Bruteforce Target device
        bruteforcing
        #call FUNCTION 11 - scan for vulnerabilities of each target devices open port services
        vulnerability_check
}
```

FUNCTION 3 - Audit Log Creation

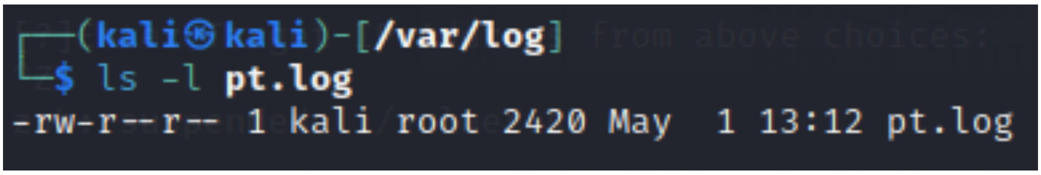
```
function log_creation()
{
 if [ -f /var/log/pt.log ]                     #if pt.log (scanned audit log) exist
 then
        currentuser=$(whoami)                       #to get current user
        sudo chown $currentuser /var/log/pt.log  #to provide current user who is not a root user to write in /var/log/pt.log
        echo "[#]/var/log/pt.log exist"
 else
        currentuser=$(whoami)                       #to get current user
```

```
                        #create PT Log to audit enumeration done on Target Address
        sudo touch /var/log/pt.log                       #to create custom log file in /var/log
        sudo chown $currentuser /var/log/pt.log  #to permit currentuser to write in the
custom audit log file <pt.log>
        echo "[#]/var/log/pt.log created"
 fi
}
```

RESULT

OUTPUT EXPECTED:

the script should create pt.log in directory /var/log
the current user will have write permission to the audit file



```
┌──(kali㉿kali)-[/var/log]
└─$ ls -l pt.log
-rw-r--r-- 1 kali root 2420 May  1 13:12 pt.log
```

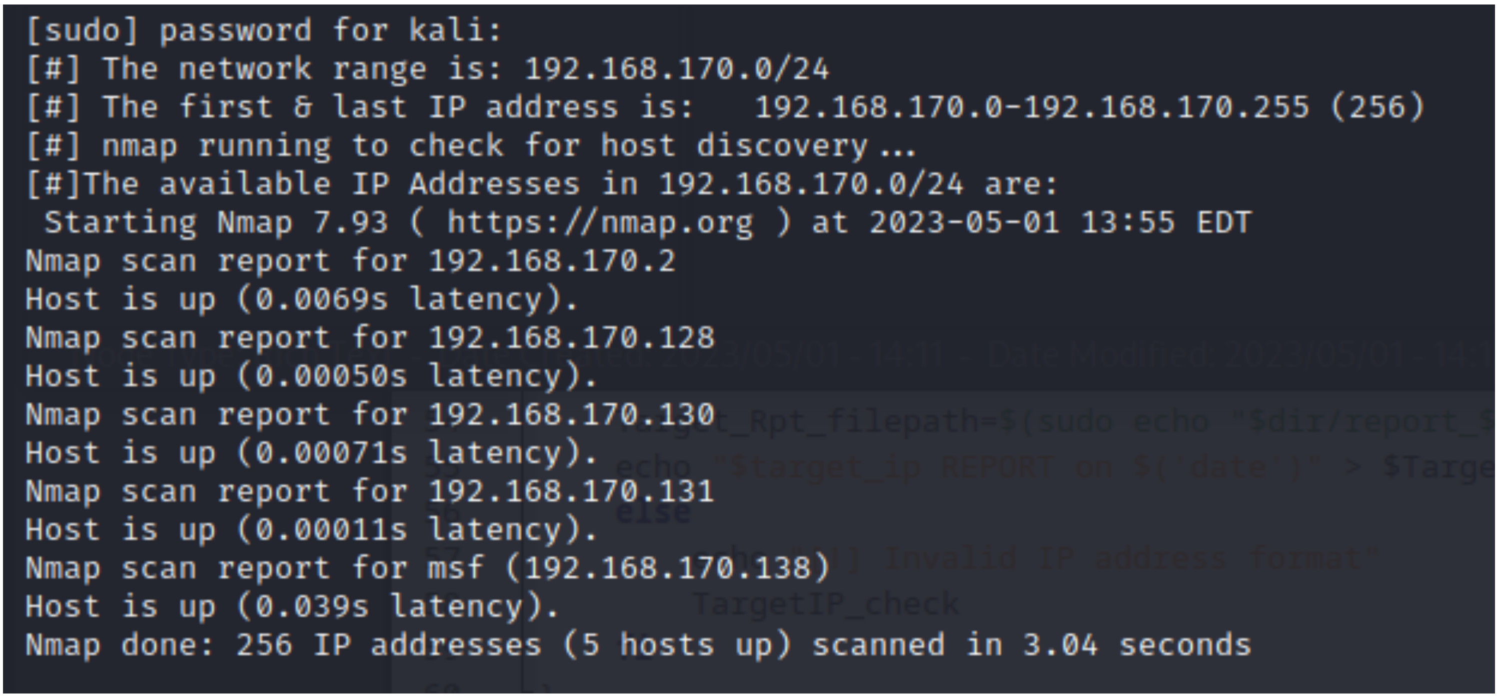
FUNCTION 4 - host discovery

```
function ip_range_info()
{
        #~ 1.1 Automatically identify the LAN network range
                ip_range_cidr=$(ip route | grep -oE '([0-9]{1,3}[\.]){3}[0-9]{1,3}/[0-9]
{1,2}') #to get the Network Address/CIDR notation
                ip_range=$(netmask -r $ip_range_cidr) #to get IP range
                echo "[#] The network range is: $ip_range_cidr" #display network address/cidr
                echo "[#] The first & last IP address is: $ip_range" #display network range
                echo "[#] nmap running to check for host discovery... "
        #~ 1.2 Automatically scan the current LAN and
        #~ 1.3 Enumerate each live host using nmap
                #do nmap to scan or do a host dicovery of available IP Addresses within the
network range specified
                Avail_IP_Addr=$(nmap -sn $ip_range_cidr)
                echo -e "[#]The available IP Addresses in $ip_range_cidr are: \n
$Avail_IP_Addr \n"
}
```

RESULT

OUTPUT EXPECTED:

the script will run using IP route to get the IP address/cidr, ip range and list discovered IP within the network



```
[sudo] password for kali:
[#] The network range is: 192.168.170.0/24
[#] The first & last IP address is:   192.168.170.0-192.168.170.255 (256)
[#] nmap running to check for host discovery ...
[#]The available IP Addresses in 192.168.170.0/24 are:
 Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-01 13:55 EDT
Nmap scan report for 192.168.170.2
Host is up (0.0069s latency).
Nmap scan report for 192.168.170.128
Host is up (0.00050s latency).
Nmap scan report for 192.168.170.130
Host is up (0.00071s latency).
Nmap scan report for 192.168.170.131
Host is up (0.00011s latency).
Nmap scan report for msf (192.168.170.138)
Host is up (0.039s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 3.04 seconds
```

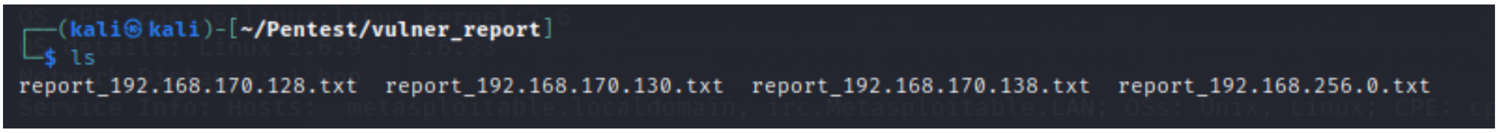
FUNCTION 5 - $target_ip variable will only match IP addresses with valid values

```
#FUNCTION 3 - $target_ip variable will only match IP addresses with valid values
function TargetIP_check()
{
#ask user to enter ip address from nmap results
echo "[?]Enter Target IP Address from above choices: "
read target_ip

        if [[ $target_ip =~ ^(([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.){3}([01]?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])$ ]]
        #format [01]?[0-9]?[0-9] matches numbers from 0 to 199
        #2[0-4][0-9] matches numbers from 200 to 249
        #25[0-5] matches numbers from 250 to 255
        then #to check if entered IP is in correct
                LocalPath=$(pwd) #to get the current working directory
                dir=$(echo "$LocalPath/vulner_report") #passing vulner_reort directory where target device report will be saved
                        if [ ! -d "$dir" ] #check if directory does not exist
                        then
                        echo "[#]Creating directory: $dir"
                        mkdir -p $LocalPath/vulner_report
                        else
                        echo "[#]vulner_report directory exist"
                        fi
        Target_Rpt_filepath=$(sudo echo "$dir/report_$target_ip.txt") #create target device report with this filename format in the current working directoy
        echo "$target_ip REPORT on $('date')" > $Target_Rpt_filepath #timestamp of report created
        else
                echo "[!] Invalid IP address format"
                TargetIP_check
        fi
}
```

RESULT

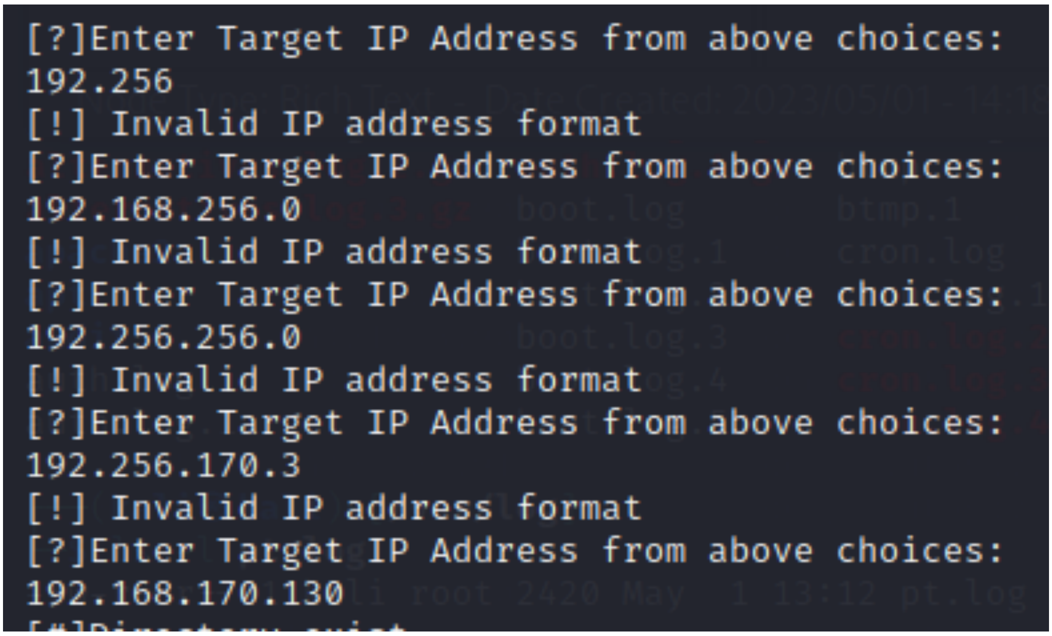
OUTPUT EXPECTED:

1. creation of vulner_report at users current folder to save target device results



```
┌──(kali㉿kali)-[~/Pentest/vulner_report]
└─$ ls
report_192.168.170.128.txt  report_192.168.170.130.txt  report_192.168.170.138.txt  report_192.168.256.0.txt
```

2. if IP format is not valid user will be prompted



```
[?]Enter Target IP Address from above choices:
192.256
[!] Invalid IP address format
[?]Enter Target IP Address from above choices:
192.168.256.0
[!] Invalid IP address format
[?]Enter Target IP Address from above choices:
192.256.256.0
[!] Invalid IP address format
[?]Enter Target IP Address from above choices:
192.256.170.3
[!] Invalid IP address format
[?]Enter Target IP Address from above choices:
192.168.170.130
[#]Directory exist
```

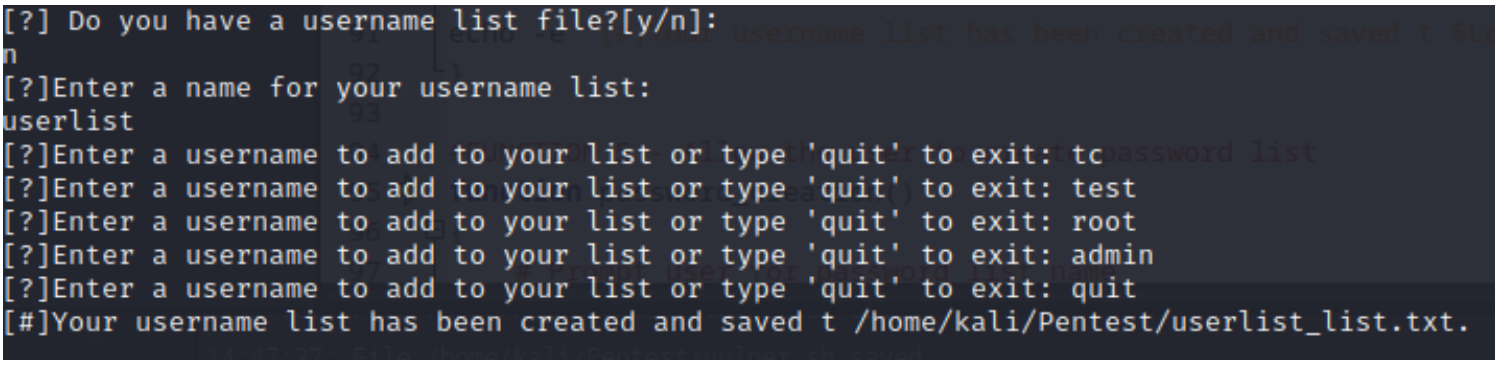
FUNCTION 6 - Allow the user to create username list

```
function username_creation()
{
# Prompt user for username list name
        echo "[?]Enter a name for your username list: "
        read user_list_name
# Create new username list file
        userlistfile="${user_list_name}_list.txt"
        touch "$userlistfile"

# Prompt user to enter username until they enter "quit"
while true; do
    read -p "[?]Enter a username to add to your list or type 'quit' to exit: " username

    # Check if user entered "quit", if so, break out of loop
    if [[ "$username" == "quit" ]]
    then
        break
    fi

    # Add username to username list file
    echo "$username" >> "$userlistfile"
done

echo -e "[#]Your username list has been created and saved t $LocalPath/$userlistfile. \n"
}
```

RESULT

OUTPUT EXPECTED:

if no, it prompt user to enter usernames list filename and create usernames
it will display where the create file is saved



```
[?] Do you have a username list file?[y/n]:
n
[?]Enter a name for your username list:
userlist
[?]Enter a username to add to your list or type 'quit' to exit: tc
[?]Enter a username to add to your list or type 'quit' to exit: test
[?]Enter a username to add to your list or type 'quit' to exit: root
[?]Enter a username to add to your list or type 'quit' to exit: admin
[?]Enter a username to add to your list or type 'quit' to exit: quit
[#]Your username list has been created and saved t /home/kali/Pentest/userlist_list.txt.
```

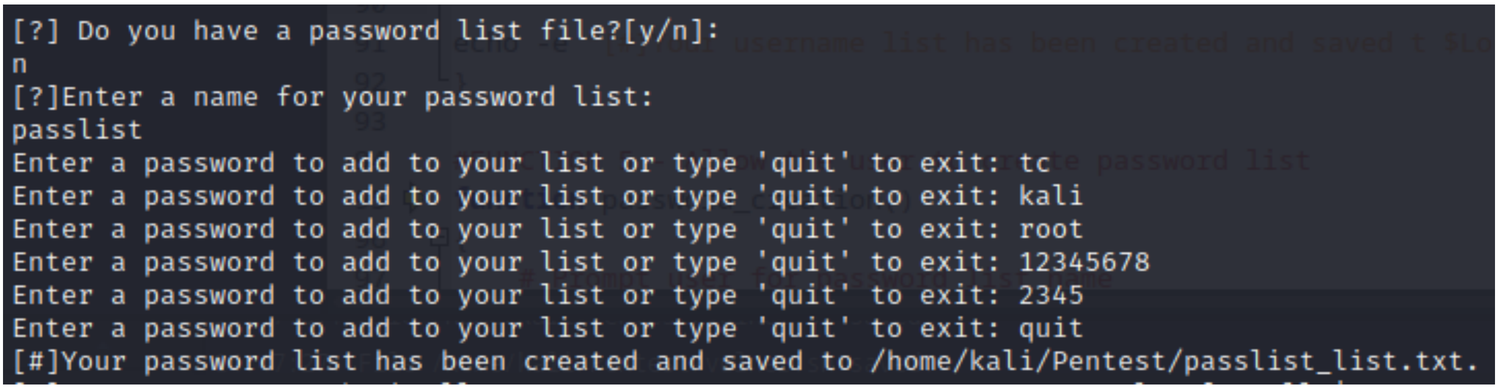
FUNCTION 7 - Allow the user to create password list

```
function password_creation()
{
        # Prompt user for password list name
        echo "[?]Enter a name for your password list: "
        read pass_list_name
# Create new password list file
        passlistfile="${pass_list_name}_list.txt"
        touch "$passlistfile"

# Prompt user to enter passwords until they enter "quit"
while true; do
    read -p "Enter a password to add to your list or type 'quit' to exit: " password

    # Check if user entered "quit", if so, break out of loop
    if [[ "$password" == "quit" ]]
    then
        break
    fi

    # Add password to password list file
    echo "$password" >> "$passlistfile"
done

echo "[#]Your password list has been created and saved to $LocalPath/$passlistfile."
}
```

RESULT

OUTPUT EXPECTED:

if no, prompt user to enter password list filename and create passwords
it will display where the create file is saved



```
[?] Do you have a password list file?[y/n]:
n
[?]Enter a name for your password list:
passlist
Enter a password to add to your list or type 'quit' to exit: tc
Enter a password to add to your list or type 'quit' to exit: kali
Enter a password to add to your list or type 'quit' to exit: root
Enter a password to add to your list or type 'quit' to exit: 12345678
Enter a password to add to your list or type 'quit' to exit: 2345
Enter a password to add to your list or type 'quit' to exit: quit
[#]Your password list has been created and saved to /home/kali/Pentest/passlist_list.txt.
```

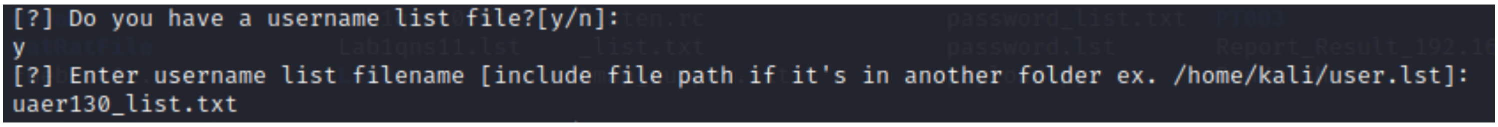
FUNCTION 8 - check if user has own username list

```
function user_list_option()
{
        echo "[?] Do you have a username list file?[y/n]: "
        read answer
        if [[ $answer == Y || $answer == y ]]
        then
                #Allow the user to specify a user list
                echo "[?] Enter username list filename [include file path if it's in another folder ex. /home/kali/user.lst]: "
                read userlistfile
        elif [[ $answer == N || $answer == n ]]
        then
                #call FUNCTION 4 username_creation
                username_creation
        else
                user_list_option
        fi
}
```

RESULT

OUTPUT EXPECTED:

if yes, user must enter the username list fiilename and it's path if not in the current directory



```
[?] Do you have a username list file?[y/n]:
y
[?] Enter username list filename [include file path if it's in another folder ex. /home/kali/user.lst]:
uaer130_list.txt
```

if no, username_creation function will be executed

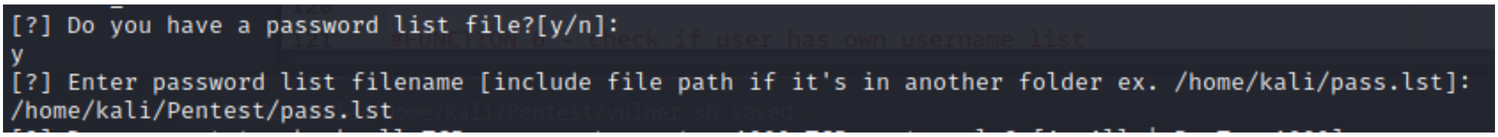
FUNCTION 9 - check if user has it's own password list

```
function pass_list_option()
{
        echo "[?] Do you have a password list file?[y/n]: "
        read answer
        #if condition statement to identify next action
        if [[ $answer == Y || $answer == y ]]
        then
                #Allow the user to specify a password list
                echo "[?] Enter password list filename [include file path if it's in another folder ex. /home/kali/pass.lst]: "
                read passlistfile
        elif [[ $answer == N || $answer == n ]]
        then
                #call FUNCTION 5 password_creation
                password_creation
        else
                pass_list_option
        fi
}
```

RESULT

OUTPUT EXPECTED:

if yes, user must enter the username list fiilename and it's path if not in the current directory



```
[?] Do you have a password list file?[y/n]:
y
[?] Enter password list filename [include file path if it's in another folder ex. /home/kali/pass.lst]:
/home/kali/Pentest/pass.lst
```

if no, password_creation function will be executed

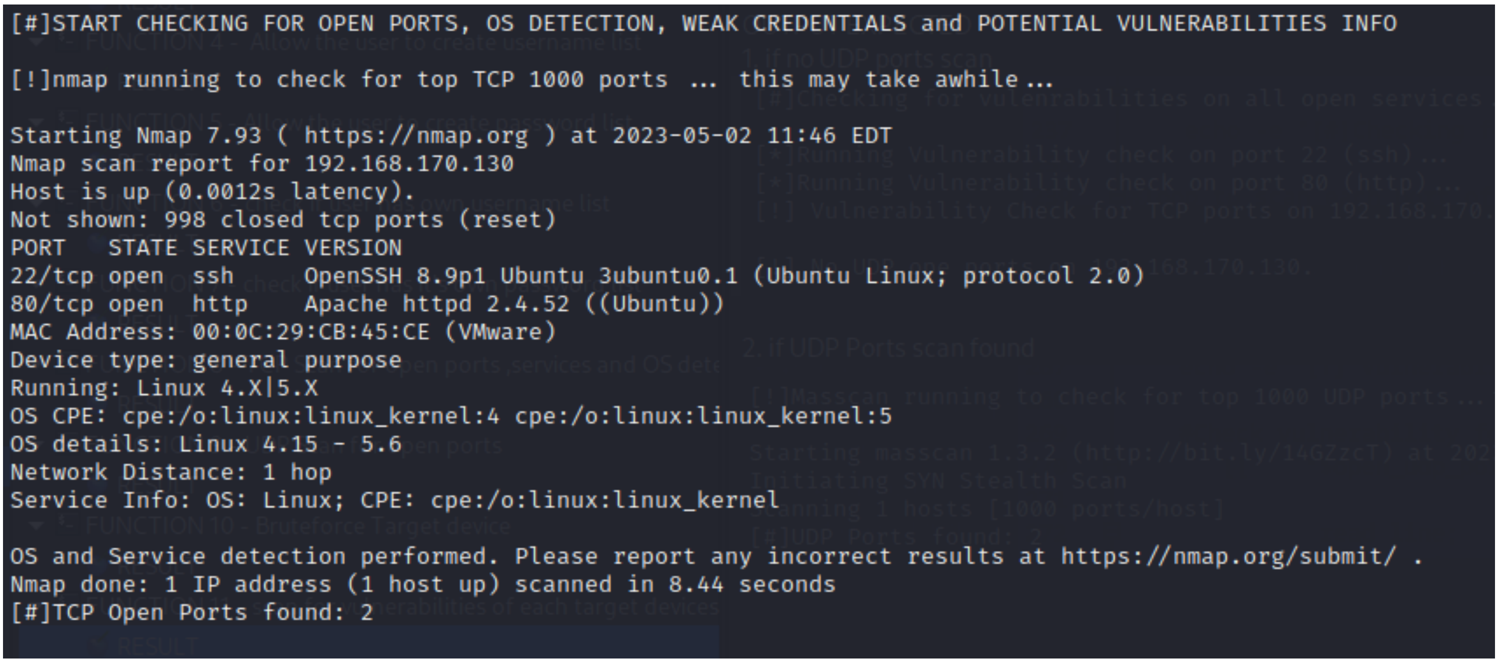
FUNCTION 10 - TCP Scan for open ports ,services and OS detection info

```
function nmap_tcp_scan()
{
        echo -e "Host discovery network range  $ip_range_cidr up and running devices are \n$Avail_IP_Addr" >> $Target_Rpt_filepath
        #~ 1.4 Find potential vulnerabilities for each device
                #provide option to check top 1000 ports or All ports
                echo -e "\n[?]Do you want to check all TCP open ports or top 1000 TCP ports only? [A: All | B: Top 1000]: "
                read scan_choice
                        case $scan_choice in
                        A|a)
                        echo -e "[#]START CHECKING FOR OPEN PORTS, OS DETECTION, WEAK CREDENTIALS and POTENTIAL VULNERABILITIES INFO \n"
                        echo "[!]nmap running to check for all open ports ... this may take awhile..."
                        nmap_tcp_output=$(sudo nmap -p- -sV -O $target_ip )  #scan all ports, display the open ports ,services and OS detection info
                        open_tcp_ports=$(echo "$nmap_tcp_output" | grep open | awk -F / '{ print $1 }')
                        open_tcp_protocol=$(echo "$nmap_tcp_output" | grep open | awk '{ print $3 }')
                        echo -e "\n$nmap_tcp_output"
                        echo -e "\n[#]These are TCP open port services for $target_ip \n$nmap_tcp_output \n" >> $Target_Rpt_filepath
                        ;;
                        B|b)
                        echo -e "[#]START CHECKING FOR OPEN PORTS, OS DETECTION, WEAK CREDENTIALS and POTENTIAL VULNERABILITIES INFO \n"
                        echo "[!]nmap running to check for top TCP 1000 ports ... this may take awhile..."
                    nmap_tcp_output=$(sudo nmap -sV -O $target_ip)  #scan all ports, display the open ports and OS detection info
                    open_tcp_ports=$(echo "$nmap_tcp_output" | grep open | awk -F / '{ print $1 }')
                        open_tcp_protocol=$(echo "$nmap_tcp_output" | grep open |awk '{ print $3 }')
                        echo -e "\n$nmap_tcp_output"
                        echo -e "\n[#]These are the TCP open port services for $target_ip \n$nmap_tcp_output \n" >> $Target_Rpt_filepath
                        ;;
                    *) #if input not within the list provided
                        echo "[!]Invalid Choice...try again"
                        nmap_results              #function to display and choose different type of attacks
                        ;;
                        esac
}
```

RESULT

OUTPUT EXPECTED:

Nmap scanning for open TCP ports



```
[#]START CHECKING FOR OPEN PORTS, OS DETECTION, WEAK CREDENTIALS and POTENTIAL VULNERABILITIES INFO

[!]nmap running to check for top TCP 1000 ports ... this may take awhile...

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:46 EDT
Nmap scan report for 192.168.170.130
Host is up (0.0012s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
MAC Address: 00:0C:29:CB:45:CE (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.44 seconds
[#]TCP Open Ports found: 2
```

FUNCTION 11 - UDP Scan for open ports

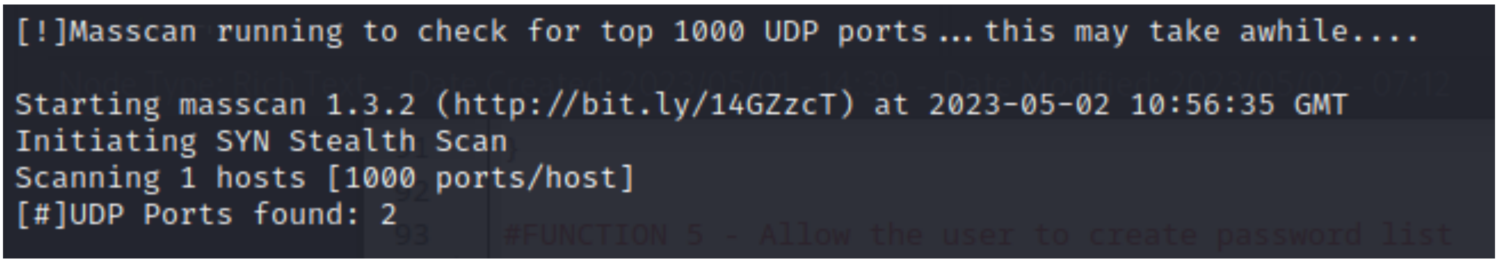
```
function udp_scan()
{
	#to scan UDP ports from 1 to 1000
	echo -e "\n[!]Masscan running to check for top 1000 UDP ports... this may take awhile....\n"
	mascan_udp_output=$(sudo masscan $target_ip -pU:1-1000 --banners --open)
	open_udp_ports=$(echo "$mascan_udp_output" |grep open |awk -F / '{print $1}'|awk '{print $4}')
	if [[ -n "$open_udp_ports" ]]
	then
	echo "UDP Ports found: $(echo "$open_udp_ports" | wc -w)"
	echo -e "\n[#]These are the top 1000 UDP open port services for $target_ip \n$mascan_udp_output \n" >> $Target_Rpt_filepath
	else
	echo "There are no open UDP ports found"
	echo -e "There are no open UDP ports found \n$mascan_udp_output \n" >> $Target_Rpt_filepath
	fi
}
```

RESULT

OUTPUT EXPECTED

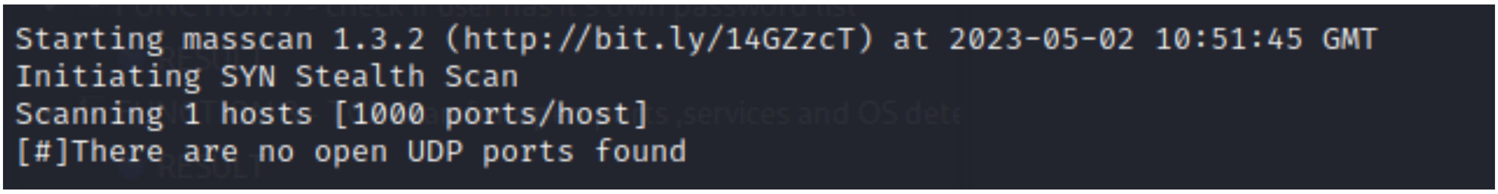
only top 1000 UDP ports will be masscan due to it is connectionless and will take much longer

if there's UDP open ports it will show the result with total count of open ports



```
[!]Masscan running to check for top 1000 UDP ports ... this may take awhile....

Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-05-02 10:56:35 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1000 ports/host]
[#]UDP Ports found: 2
```

if no UDP open ports , below is the result



```
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-05-02 10:51:45 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1000 ports/host]
[#]There are no open UDP ports found
```

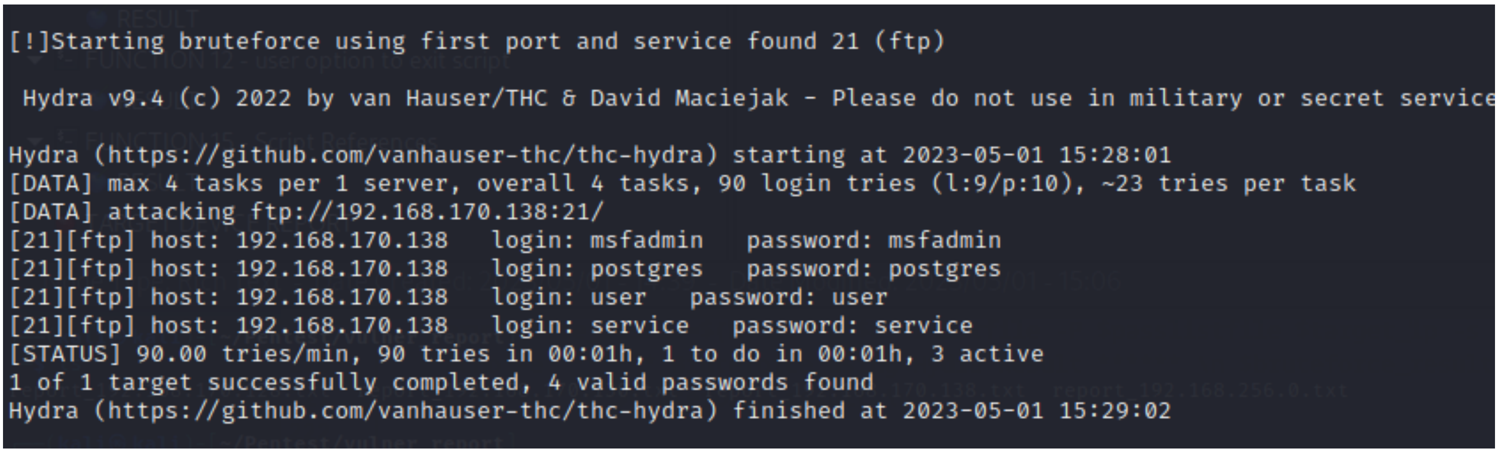
FUNCTION 12 - Bruteforce Target device

```
function bruteforcing()
{
if [[ $(echo "$open_tcp_protocol" |wc -l) -gt 1 ]] then
    # If more than one login service is available, choose the first service -
        first_port=$(echo "$open_tcp_ports" |head -n 1)
        first_protocol=$(echo "$open_tcp_protocol" |head -n 1)
        echo -e "\n[!]Starting bruteforce using first port and service found $first_port ($first_protocol)"
    bruteforce_result=$(hydra -L $userlistfile -P $passlistfile $target_ip -s $first_port $first_protocol -t4 -I)
    echo -e "****** Bruteforce using Hydra Result ****** \n $bruteforce_result" >> $Target_Rpt_filepath
    echo -e "\n $bruteforce_result \n"

else
    echo -e "\n[!]Starting bruteforce using only port and service found $open_tcp_ports ($open_tcp_protocol)"
    bruteforce_result=$(hydra -L $userlistfile -P $passlistfile $target_ip -s $open_tcp_ports $open_tcp_protocol -t4 -I)
    echo -e "****** Bruteforce using Hydra Result ****** \n $bruteforce_result" >> $Target_Rpt_filepath
    echo -e "\n $bruteforce_result \n"

fi
num_device=$(echo "$Avail_IP_Addr" |grep -oE '([0-9]{1,3}[\.]){3}[0-9]{1,3}'|wc -l)
num_openport=$(echo "$open_ports" |wc -l)
Nmap_Device_Timestamp=$(echo "$Nmap_Device_StartTimestamp $(date '+%r') - [*] Nmap Scanning on $target_ip from Network Address/CIDR: $ip_range_cidr with Network Range: $ip_range, devices found: $num_device, Open Ports found: $num_openport ")
echo "[#]$Nmap_Device_Timestamp" >> /var/log/pt.log
}
```

RESULT

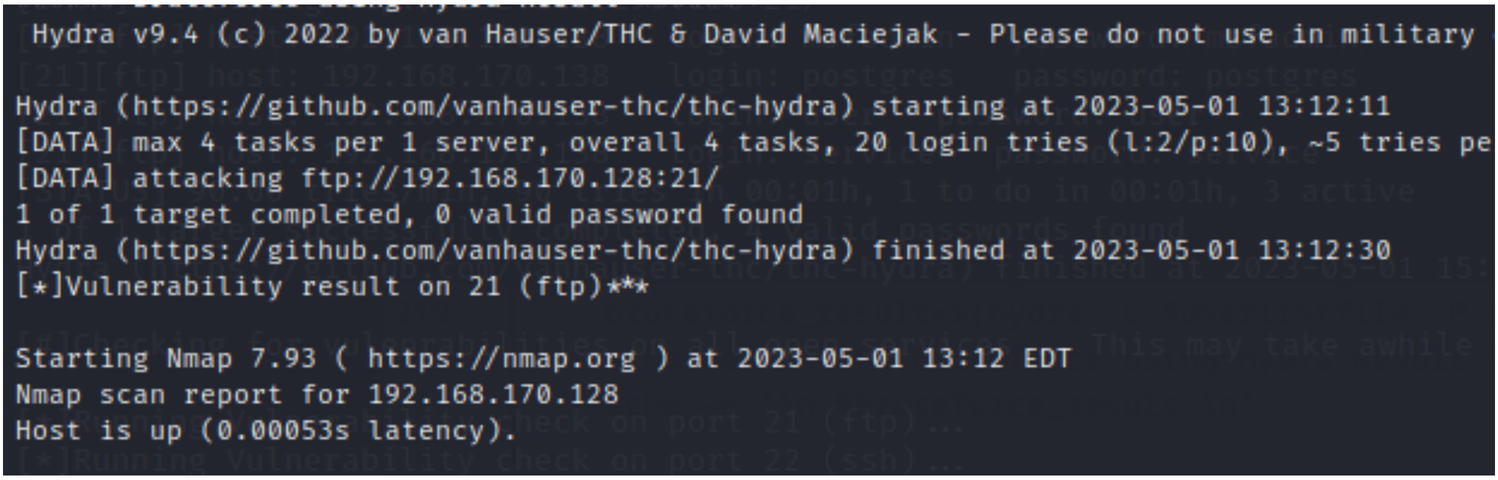
OUTPUT EXPECTED

1. if valid credential found

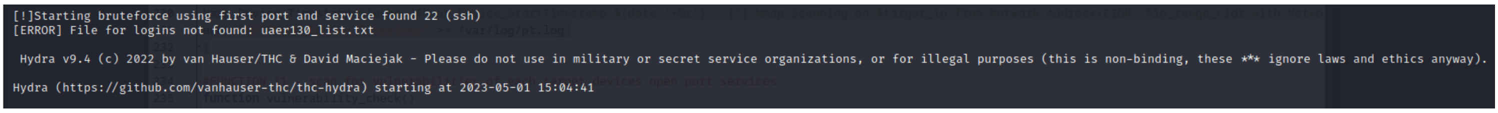


```
[!]Starting bruteforce using first port and service found 21 (ftp)

 Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-01 15:28:01
[DATA] max 4 tasks per 1 server, overall 4 tasks, 90 login tries (l:9/p:10), ~23 tries per task
[DATA] attacking ftp://192.168.170.138:21/
[21][ftp] host: 192.168.170.138   login: msfadmin   password: msfadmin
[21][ftp] host: 192.168.170.138   login: postgres   password: postgres
[21][ftp] host: 192.168.170.138   login: user   password: user
[21][ftp] host: 192.168.170.138   login: service   password: service
[STATUS] 90.00 tries/min, 90 tries in 00:01h, 1 to do in 00:01h, 3 active
1 of 1 target successfuly completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-01 15:29:02
```

2. if no valid credential found



```
 Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-01 13:12:11
[DATA] max 4 tasks per 1 server, overall 4 tasks, 20 login tries (l:2/p:10), ~5 tries pe
[DATA] attacking ftp://192.168.170.128:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-01 13:12:30
[*]Vulnerability result on 21 (ftp)**

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-01 13:12 EDT
Nmap scan report for 192.168.170.128
Host is up (0.00053s latency).
```



```
[!]Starting bruteforce using first port and service found 22 (ssh)
[ERROR] File for logins not found: uaer130_list.txt

 Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-01 15:04:41
```

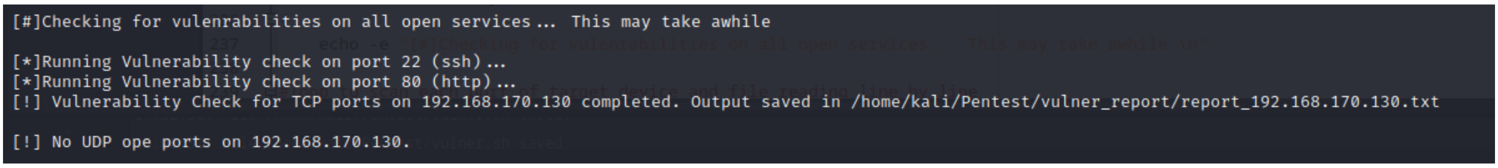
FUNCTION 13 - scan for vulnerabilities of each target devices open port services

```
function vulnerability_check()
{
        echo -e "[#]Checking for vulenrabilities on all open services... This may take awhile
\n"

#loop to scan each port of target device and file reading line by line
        #Check vulnerability for TCP open ports
        while read line; do
          if echo "$line" | grep -q open; then
                port=$(echo $line | cut -d '/' -f 1)
                service=$(echo $line | cut -d ' ' -f 3)

                echo "[*]Running Vulnerability check on port $port ($service)..."
                port_vuln_result=$(nmap --script vuln -p $port $target_ip -sV)
                echo -e "[*]Vulnerability result on $port ($service)** \n" >>
$Target_Rpt_filepath
                echo "$port_vuln_result" >> $Target_Rpt_filepath
          fi
        done <<< "$(echo "$nmap_tcp_output" | grep open)" # to redirect the input of a loop
to come from a string
        echo -e "[!] Vulnerability Check for TCP ports on $target_ip completed. Output saved
in $Target_Rpt_filepath\n"

        #Check vulnerability for UDP open ports
        if [[ -n "$open_udp_ports" ]]
        then
                while read line; do #loop to read each line from $open_udp_ports variable
                        if [[ -n "$open_udp_ports" ]] #check length of string not zero
                        then
                            echo "[*]Running Vulnerability check on port $open_udp_ports..."
                                port_vuln_result=$(nmap --script vuln -sU -p $open_udp_ports
$target_ip -sV)
                                echo -e "[*]Vulnerability result on $open_udp_ports*** \n" >>
$Target_Rpt_filepath
                                echo "$port_vuln_result" >> $Target_Rpt_filepath
                        fi
                done <<< $open_udp_ports # to redirect the input of a loop to come from a
string
        echo -e "[!] Vulnerability Check for UDP Ports on $target_ip completed. Output saved
in $Target_Rpt_filepath\n"
        else
        echo -e "[!] No UDP open ports on $target_ip. \n"
        fi

        #call FUNCTION 12 - Exit script
        exit_script
}
```

RESULT

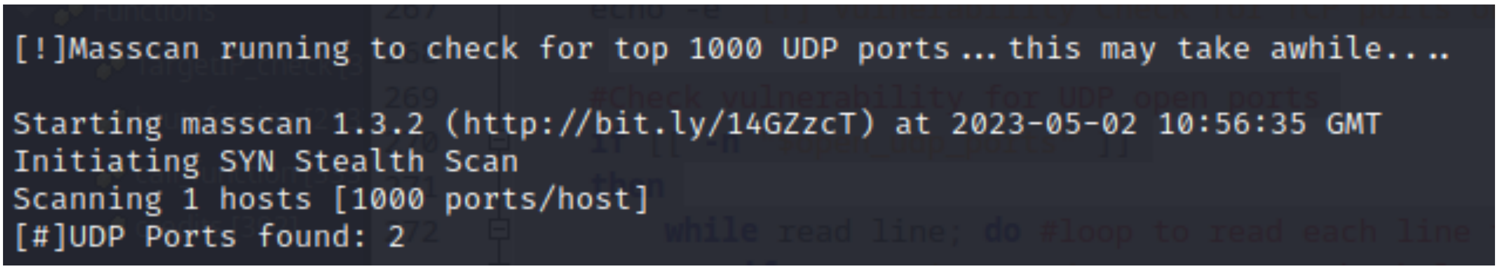
OUTPUT EXPECTED

1. if no UDP ports scan



```
[#]Checking for vulenrabilities on all open services... This may take awhile

[*]Running Vulnerability check on port 22 (ssh)...
[*]Running Vulnerability check on port 80 (http)...
[!] Vulnerability Check for TCP ports on 192.168.170.130 completed. Output saved in /home/kali/Pentest/vulner_report/report_192.168.170.130.txt

[!] No UDP ope ports on 192.168.170.130.
```

2. if UDP Ports scan found



```
[!]Masscan running to check for top 1000 UDP ports...this may take awhile...

Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-05-02 10:56:35 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [1000 ports/host]
[#]UDP Ports found: 2
```

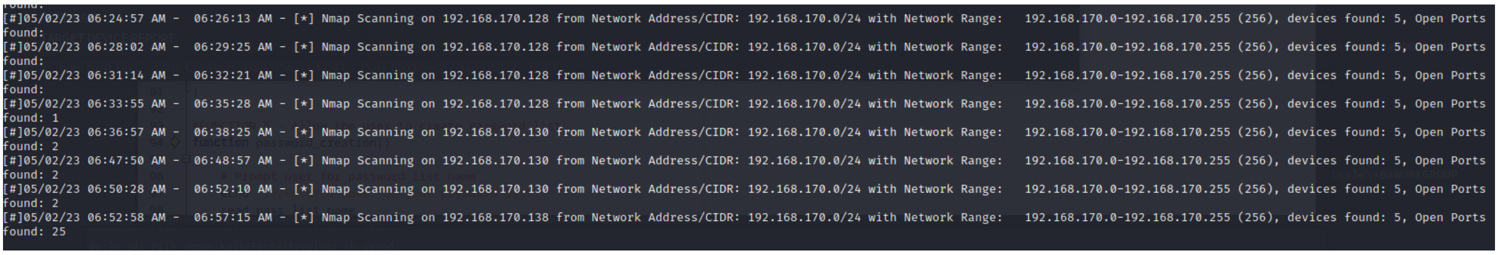
FUNCTION 14 - user option to exit script

```
function exit_script()
{
        #Define an exit message to be displayed
        ExitMessage1=$(echo "[*] You have exited the script.")
        ExitMessage2=$(echo "[*] Have an AWESOME day!")
        #to check if user want to scan another target device
        echo -n "[?] Do you want to check another target device? [Y/N]: "
        read answer
                        if [[ $answer == Y || $answer == y ]]
                        then
                        #user to enter another
                                #call Function 13 - main program
                            call_function
                        elif [[ $answer == N || $answer == n ]]
                        then
                                echo -n "[?] Do you like to open the log file? [Y/N]: "
                                read answer
                                if [[ $answer == Y || $answer == y ]]
                                then
                                        cat /var/log/pt.log
                                        echo -e "\n"
                                        echo "[#]Log file is located at /var/log/pt.log"
                                        echo "Target device report is located $dir"
                                        echo "$ExitMessage1"
                                        echo "$ExitMessage2"
                                        #call credits function
                                        credits
                                else
                                        echo "Log file is located at /var/log/pt.log"
                                        echo "Target device report is located $dir"
                                        echo "$ExitMessage1"
                                        echo "$ExitMessage2"
                                        #call credits function
                                        credits
                                        exit
                                fi
                        else
                                exit_script
                        fi
}
```

RESULT

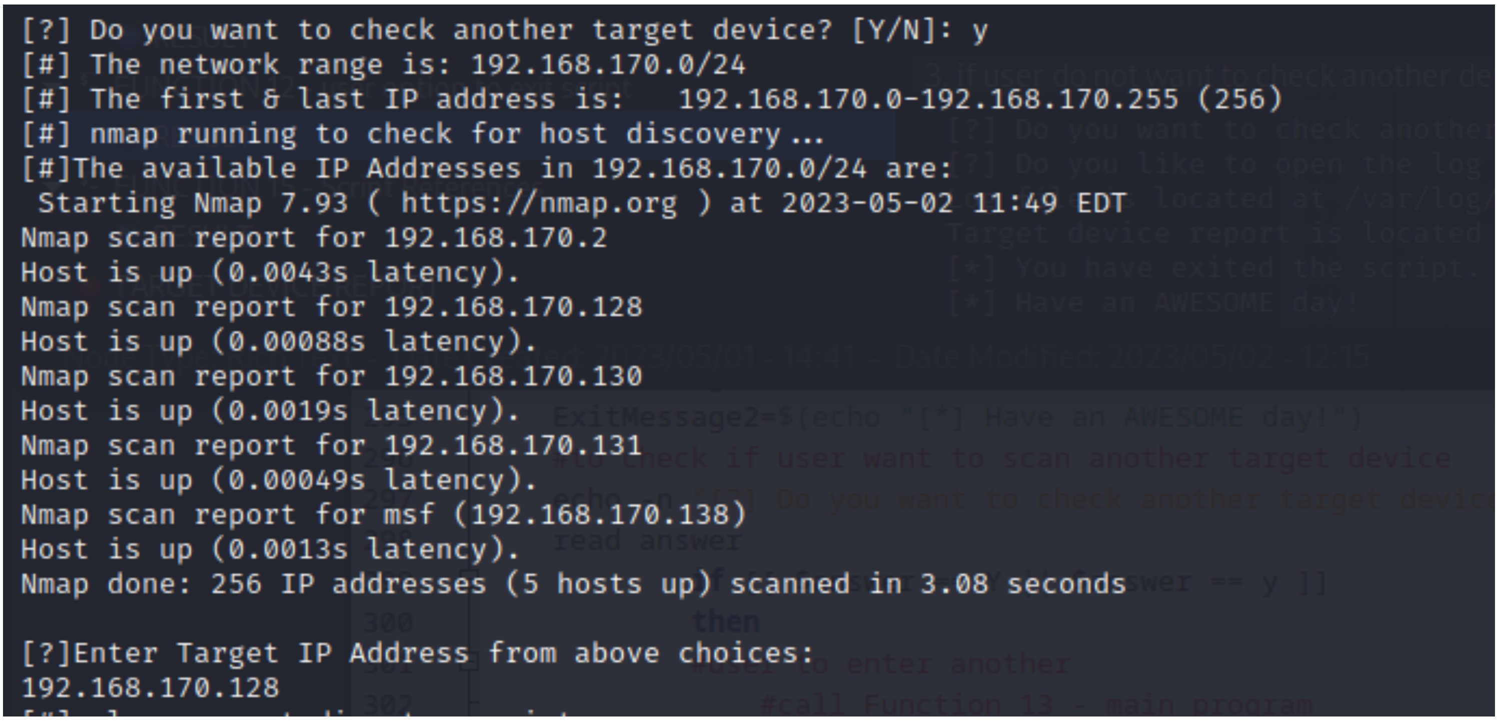
OUTPUT EXPECTED

1. if user do not want to check another device and wants to view the audit log



```
found:
[#]05/02/23 06:24:57 AM -  06:26:13 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found:
[#]05/02/23 06:28:02 AM -  06:29:25 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found:
[#]05/02/23 06:31:14 AM -  06:32:21 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found:
[#]05/02/23 06:33:55 AM -  06:35:28 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/02/23 06:36:57 AM -  06:38:25 AM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 2
[#]05/02/23 06:47:50 AM -  06:48:57 AM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 2
[#]05/02/23 06:50:28 AM -  06:52:10 AM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 2
[#]05/02/23 06:52:58 AM -  06:57:15 AM - [*] Nmap Scanning on 192.168.170.138 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 25
```

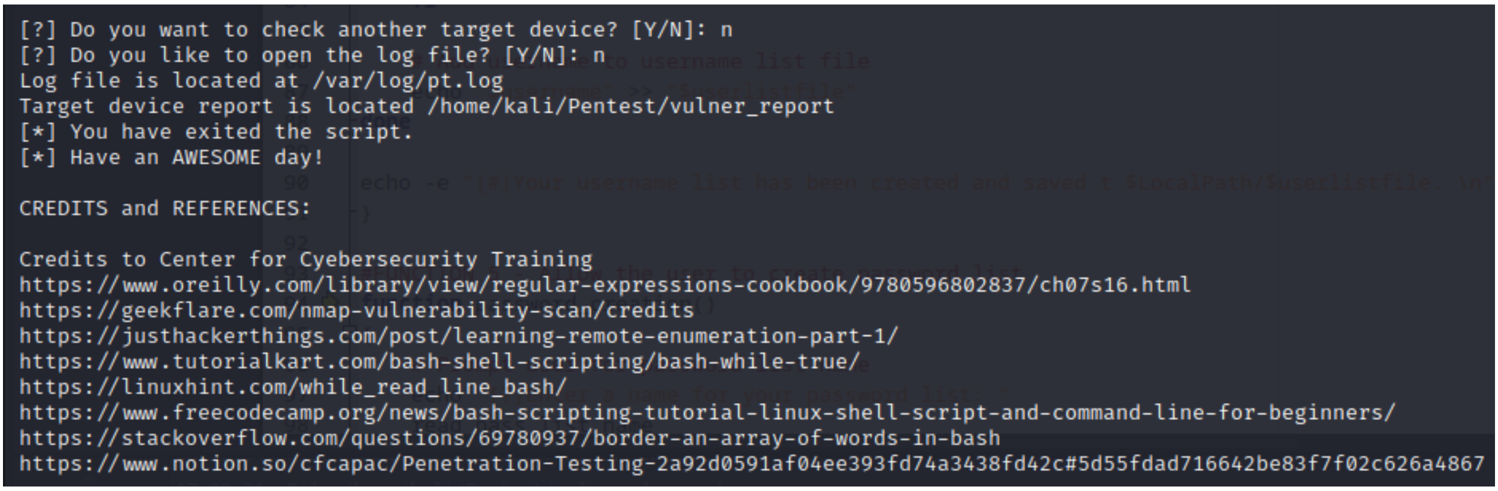
2. if user want to check another target device the call_function will be executed



```
[?] Do you want to check another target device? [Y/N]: y
[#] The network range is: 192.168.170.0/24
[#] The first & last IP address is:   192.168.170.0-192.168.170.255 (256)
[#] nmap running to check for host discovery...
[#]The available IP Addresses in 192.168.170.0/24 are:
 Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:49 EDT
Nmap scan report for 192.168.170.2
Host is up (0.0043s latency).
Nmap scan report for 192.168.170.128
Host is up (0.00088s latency).
Nmap scan report for 192.168.170.130
Host is up (0.0019s latency).
Nmap scan report for 192.168.170.131
Host is up (0.00049s latency).
Nmap scan report for msf (192.168.170.138)
Host is up (0.0013s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 3.08 seconds

[?]Enter Target IP Address from above choices:
192.168.170.128
```

.

3. if user do not want to check another device and do not want to open pt.log



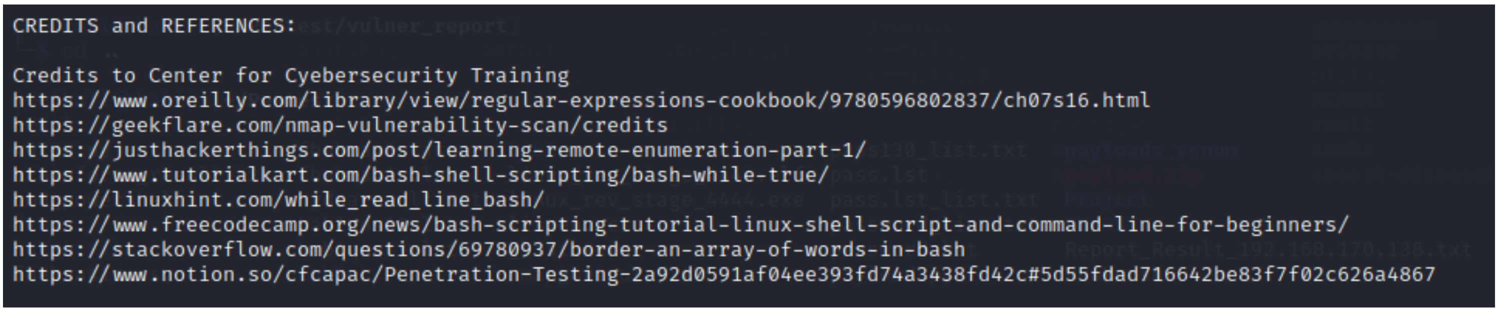
```
[?] Do you want to check another target device? [Y/N]: n
[?] Do you like to open the log file? [Y/N]: n
Log file is located at /var/log/pt.log
Target device report is located /home/kali/Pentest/vulner_report
[*] You have exited the script.
[*] Have an AWESOME day!

CREDITS and REFERENCES:

Credits to Center for Cyebersecurity Training
https://www.oreilly.com/library/view/regular-expressions-cookbook/9780596802837/ch07s16.html
https://geekflare.com/nmap-vulnerability-scan/credits
https://justhackerthings.com/post/learning-remote-enumeration-part-1/
https://www.tutorialkart.com/bash-shell-scripting/bash-while-true/
https://linuxhint.com/while_read_line_bash/
https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/
https://stackoverflow.com/questions/69780937/border-an-array-of-words-in-bash
https://www.notion.so/cfcapac/Penetration-Testing-2a92d0591af04ee393fd74a3438fd42c#5d55fdad716642be83f7f02c626a4867
```

FUNCTION 15 - Script References

```
function credits()
{
echo -e "\nCREDITS and REFERENCES: \n"
echo "Credits to Center for Cyebersecurity Training "
echo "https://www.oreilly.com/library/view/regular-expressions-cookbook/9780596802837/
ch07s16.html"
echo "https://geekflare.com/nmap-vulnerability-scan/credits"
echo "https://justhackerthings.com/post/learning-remote-enumeration-part-1/"
echo "https://www.tutorialkart.com/bash-shell-scripting/bash-while-true/"
echo "https://linuxhint.com/while_read_line_bash/"
echo "https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/"
echo "https://stackoverflow.com/questions/69780937/border-an-array-of-words-in-bash"
echo "https://www.notion.so/cfcapac/Penetration-Testing-2a92d0591af04ee393fd74a3438fd42c#5d55fdad716642be83f7f02c626a4867"
}
```

RESULT

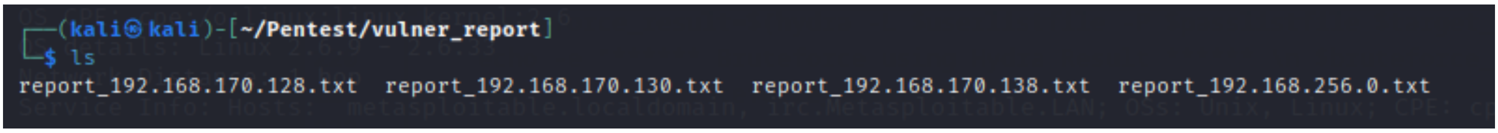
OUTPUT EXPECTED



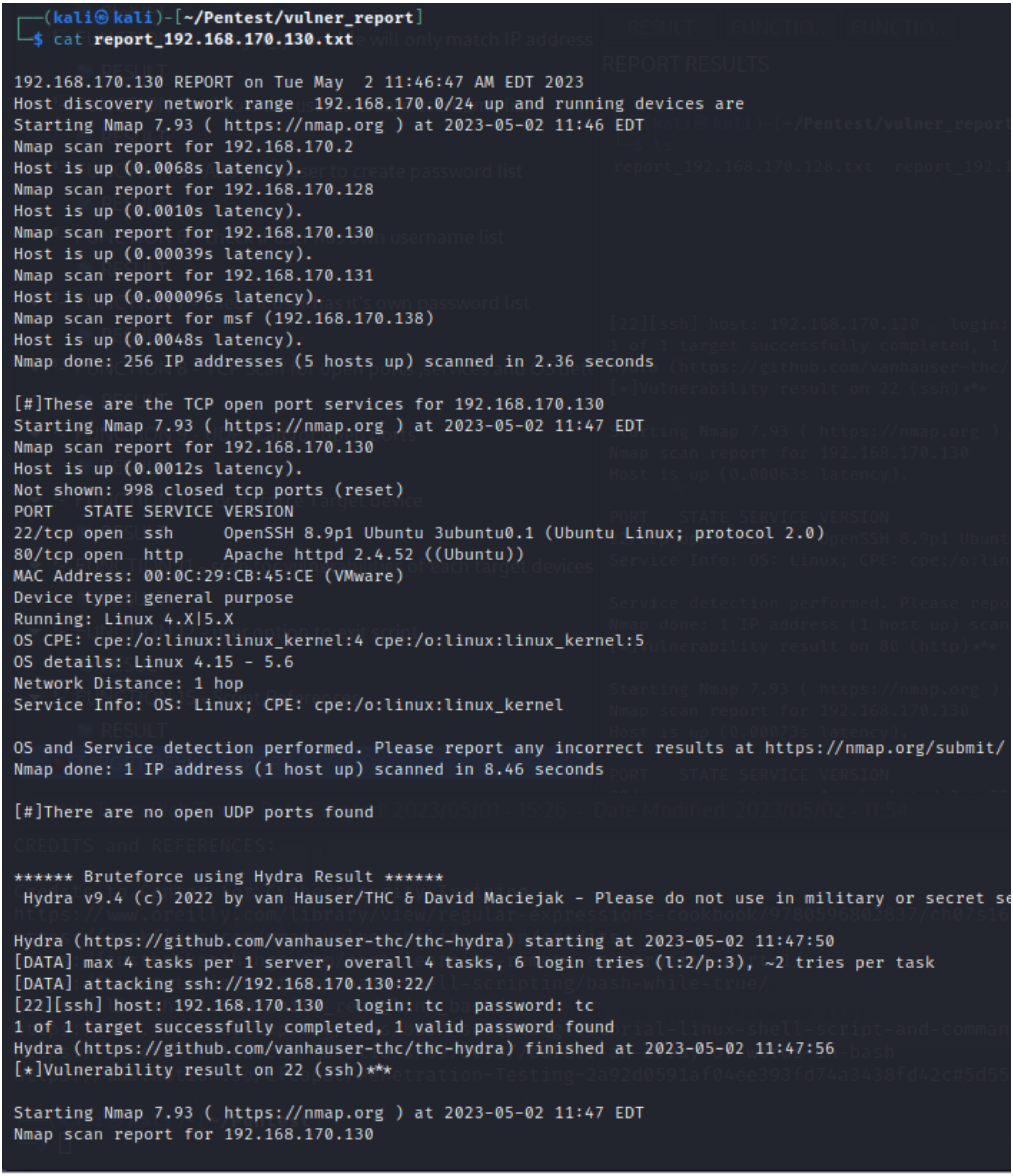
```
CREDITS and REFERENCES:

Credits to Center for Cyebersecurity Training
https://www.oreilly.com/library/view/regular-expressions-cookbook/9780596802837/ch07s16.html
https://geekflare.com/nmap-vulnerability-scan/credits
https://justhackerthings.com/post/learning-remote-enumeration-part-1/
https://www.tutorialkart.com/bash-shell-scripting/bash-while-true/
https://linuxhint.com/while_read_line_bash/
https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/
https://stackoverflow.com/questions/69780937/border-an-array-of-words-in-bash
https://www.notion.so/cfcapac/Penetration-Testing-2a92d0591af04ee393fd74a3438fd42c#5d55fdad716642be83f7f02c626a4867
```

TARGET DEVICE REPORT

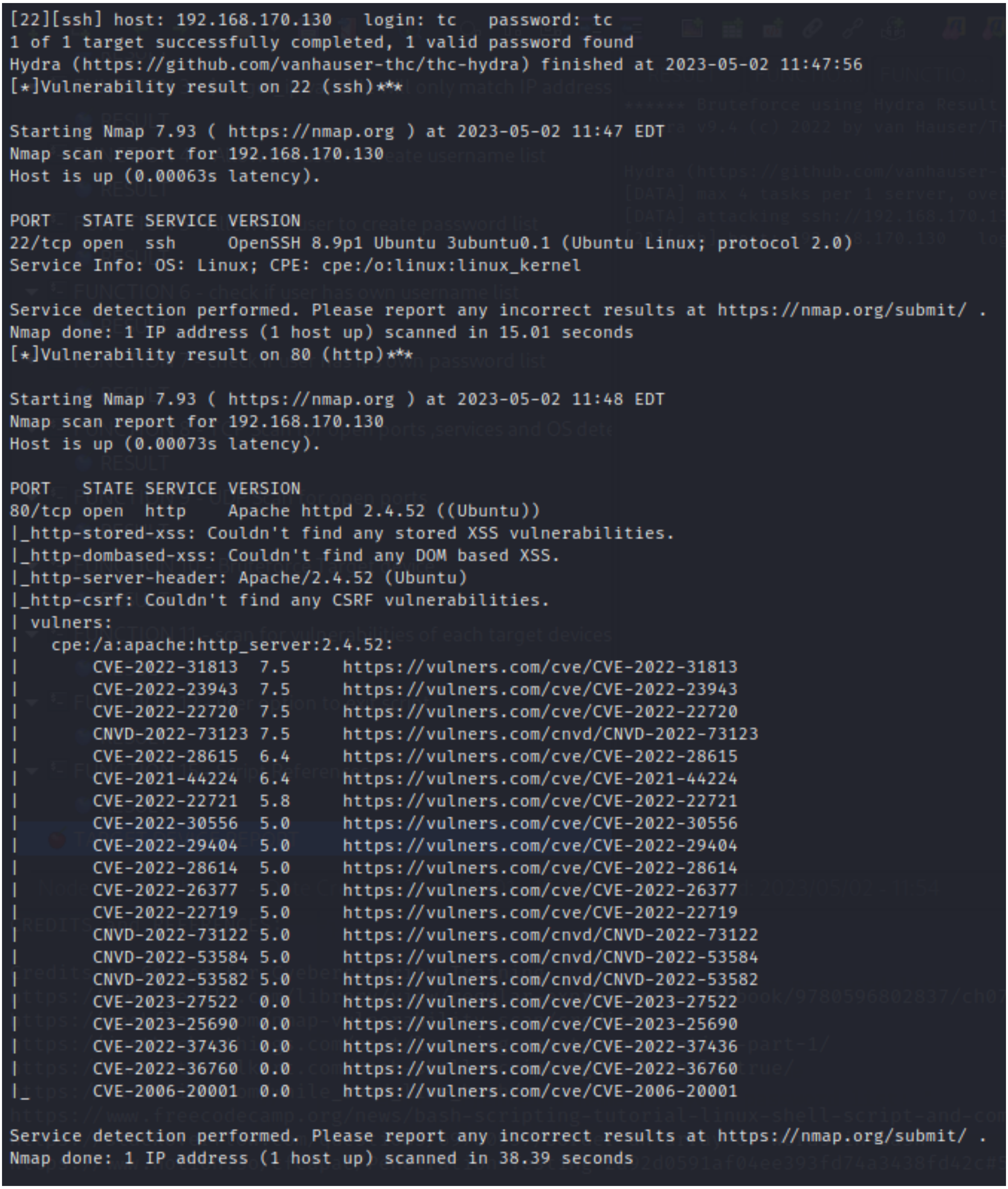
REPORT RESULTS



```
┌──(kali㉿kali)-[~/Pentest/vulner_report]
└─$ ls
report_192.168.170.128.txt  report_192.168.170.130.txt  report_192.168.170.138.txt  report_192.168.256.0.txt
```

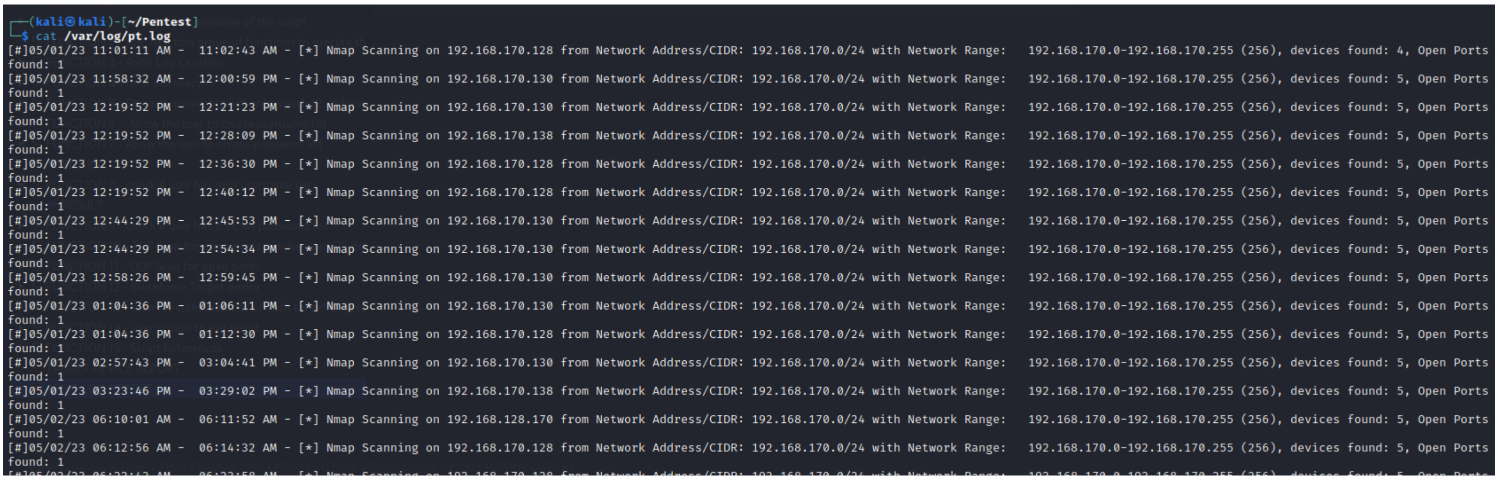


```
┌──(kali㉿kali)-[~/Pentest/vulner_report]
└─$ cat report_192.168.170.130.txt

192.168.170.130 REPORT on Tue May  2 11:46:47 AM EDT 2023
Host discovery network range  192.168.170.0/24 up and running devices are
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:46 EDT
Nmap scan report for 192.168.170.2
Host is up (0.0068s latency).
Nmap scan report for 192.168.170.128
Host is up (0.0010s latency).
Nmap scan report for 192.168.170.130
Host is up (0.00039s latency).
Nmap scan report for 192.168.170.131
Host is up (0.000096s latency).
Nmap scan report for msf (192.168.170.138)
Host is up (0.0048s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.36 seconds

[#]These are the TCP open port services for 192.168.170.130
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:47 EDT
Nmap scan report for 192.168.170.130
Host is up (0.0012s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
MAC Address: 00:0C:29:CB:45:CE (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 8.46 seconds

[#]There are no open UDP ports found

****** Bruteforce using Hydra Result ******
 Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret se

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-02 11:47:50
[DATA] max 4 tasks per 1 server, overall 4 tasks, 6 login tries (l:2/p:3), ~2 tries per task
[DATA] attacking ssh://192.168.170.130:22/
[22][ssh] host: 192.168.170.130   login: tc   password: tc
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-02 11:47:56
[*]Vulnerability result on 22 (ssh)***

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:47 EDT
Nmap scan report for 192.168.170.130
```



```
[22][ssh] host: 192.168.170.130   login: tc   password: tc
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-02 11:47:56
[*]Vulnerability result on 22 (ssh)***

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:47 EDT
Nmap scan report for 192.168.170.130
Host is up (0.00063s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.01 seconds
[*]Vulnerability result on 80 (http)***

Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-02 11:48 EDT
Nmap scan report for 192.168.170.130
Host is up (0.00073s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| vulners:
|   cpe:/a:apache:http_server:2.4.52:
|       CVE-2022-31813  7.5     https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5     https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5     https://vulners.com/cve/CVE-2022-22720
|       CNVD-2022-73123 7.5     https://vulners.com/cnvd/CNVD-2022-73123
|       CVE-2022-28615  6.4     https://vulners.com/cve/CVE-2022-28615
|       CVE-2021-44224  6.4     https://vulners.com/cve/CVE-2021-44224
|       CVE-2022-22721  5.8     https://vulners.com/cve/CVE-2022-22721
|       CVE-2022-30556  5.0     https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  5.0     https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-28614  5.0     https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-26377  5.0     https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  5.0     https://vulners.com/cve/CVE-2022-22719
|       CNVD-2022-73122 5.0     https://vulners.com/cnvd/CNVD-2022-73122
|       CNVD-2022-53584 5.0     https://vulners.com/cnvd/CNVD-2022-53584
|       CNVD-2022-53582 5.0     https://vulners.com/cnvd/CNVD-2022-53582
|       CVE-2023-27522  0.0     https://vulners.com/cve/CVE-2023-27522
|       CVE-2023-25690  0.0     https://vulners.com/cve/CVE-2023-25690
|       CVE-2022-37436  0.0     https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-36760  0.0     https://vulners.com/cve/CVE-2022-36760
|_      CVE-2006-20001  0.0     https://vulners.com/cve/CVE-2006-20001

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.39 seconds
```

AUDIT LOG REPORT



```
┌──(kali㉿kali)-[~/Pentest]
└─$ cat /var/log/pt.log
[#]05/01/23 11:01:11 AM -  11:02:43 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 4, Open Ports
found: 1
[#]05/01/23 11:58:32 AM -  12:00:59 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:19:52 PM -  12:21:23 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:19:52 PM -  12:28:09 PM - [*] Nmap Scanning on 192.168.170.138 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:19:52 PM -  12:36:30 PM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:19:52 PM -  12:40:12 PM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:44:29 PM -  12:45:53 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:44:29 PM -  12:54:34 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 12:58:26 PM -  12:59:45 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 01:04:36 PM -  01:06:11 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 01:04:36 PM -  01:12:30 PM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 02:57:43 PM -  03:04:41 PM - [*] Nmap Scanning on 192.168.170.130 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/01/23 03:23:46 PM -  03:29:02 PM - [*] Nmap Scanning on 192.168.170.138 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/02/23 06:10:01 AM -  06:11:52 AM - [*] Nmap Scanning on 192.168.128.170 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/02/23 06:12:56 AM -  06:14:32 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
found: 1
[#]05/02/23 06:22:43 AM -  06:23:58 AM - [*] Nmap Scanning on 192.168.170.128 from Network Address/CIDR: 192.168.170.0/24 with Network Range:   192.168.170.0-192.168.170.255 (256), devices found: 5, Open Ports
```