THINK CYBER CYBERIUM ARENA SIMULATOR TRAINING

NX222 – PENETRRATION TESTING

PROJECT TITLE: VULNER
(OLALEKAN ILORI – S4)

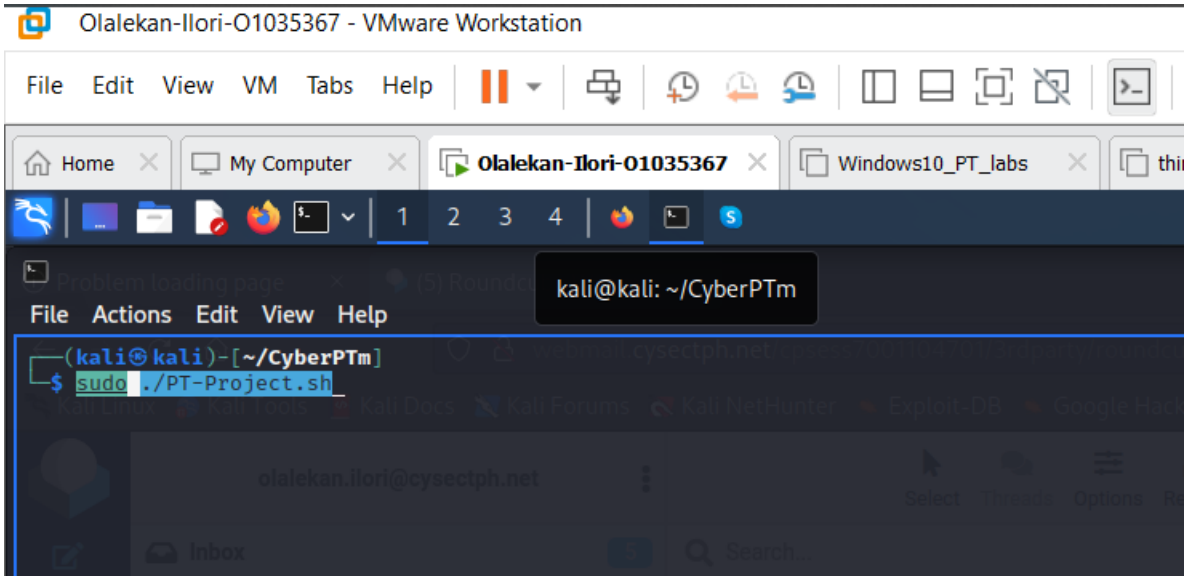
Automated tool for Mapping LAN, checking for Common Vulnerabilities and Testing for Weak Password

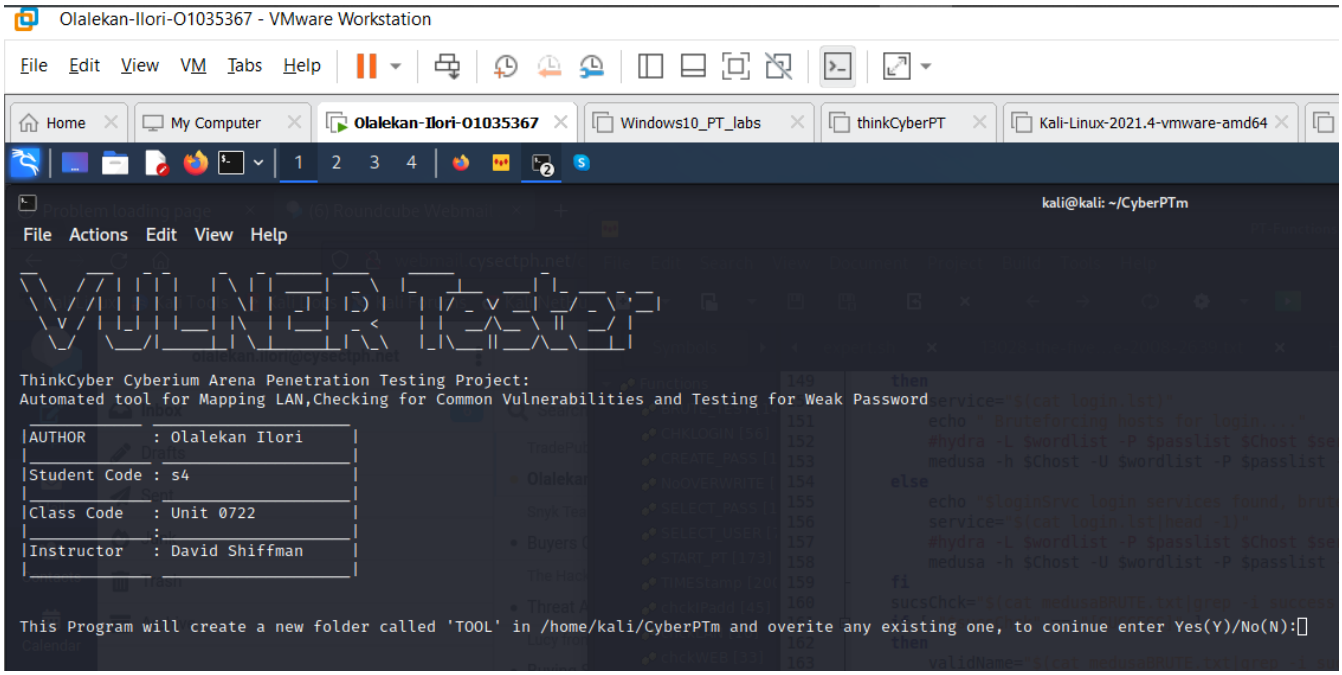
#STUDENT NAME; Olalekan Ilori
#STUDENT CODE; s4
#CLASS CODE; Unit 0722
#LECTURER'S NAME; David Shiffman

***Two scripts are used, the executable (./PT-Project.sh) and function script(PT-Functions.sh) imported into the executable script. The two files must be placed inside same folder to run ***

<u>SCREENSHOTS OF FUNCTIONS BEING EXECUTED ON KALI LINUX TERMINAL</u> *(bash script displayed after each image)*

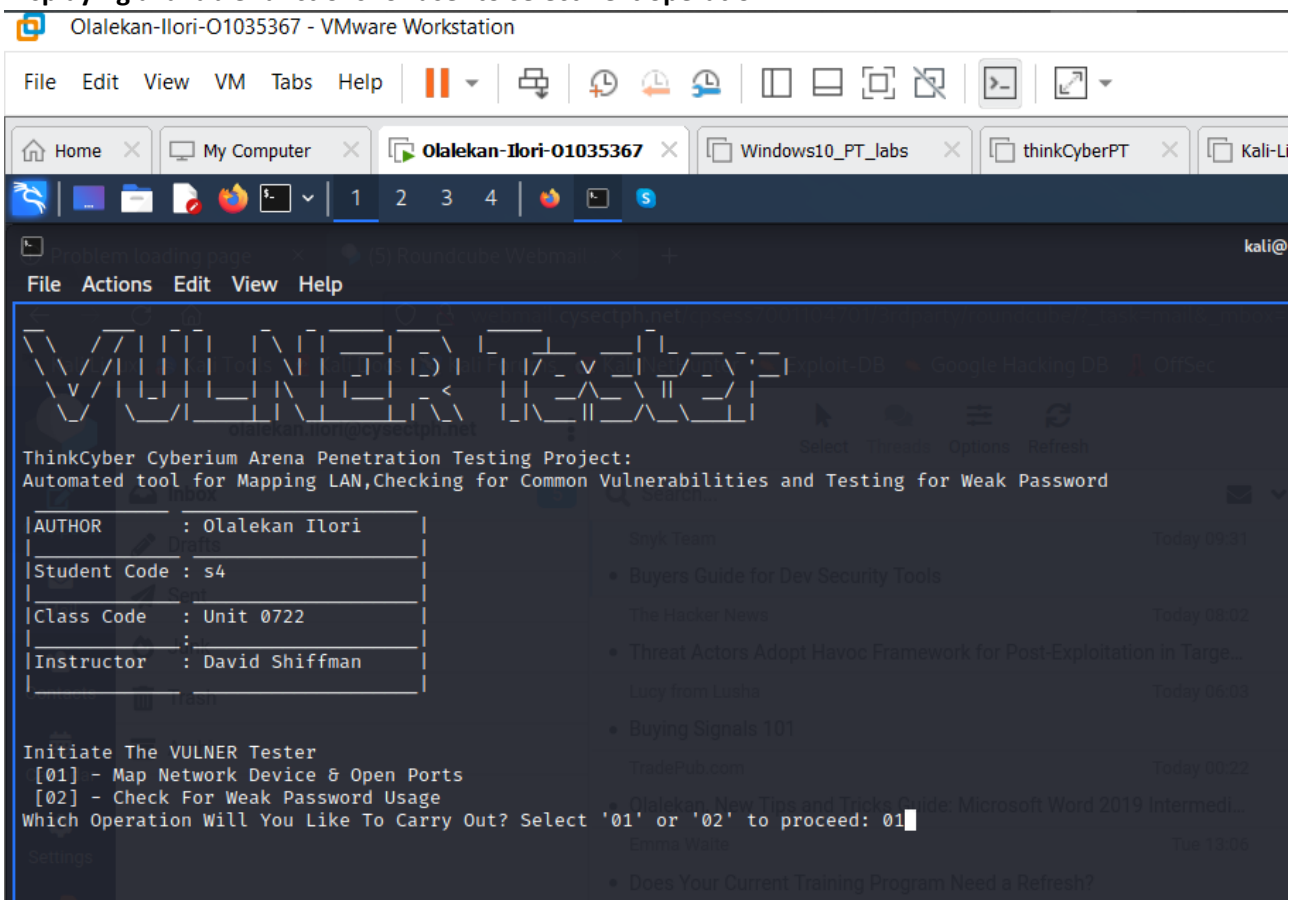
- **Initiating the executable bash script**






```
function BANNER()
{
        figlet "VULNER Tester"
        echo -e "ThinkCyber Cyberium Arena Penetration Testing Project:\nAutomated tool for Mapping
LAN,Checking for Common Vulnerabilities and Testing for Weak Password"
        echo " _____ _____"
        echo "|AUTHOR      : Olalekan Ilori     |"
        echo "|_____ _____|"
        echo "|Student Code : s4                |"
        echo "|_____ _____|"
        echo "|Class Code   : Unit 0722         |"
        echo "|_____:_____|"
        echo "|Instructor   : David Shiffman     |"
        echo "|_____ _____|"
        echo -e "  \n "
}
read -p "This Program will create a new folder called 'TOOL' in $(pwd) and overite any existing one, to continue enter
Yes(Y)/No(N):" PERMIT
if [ "$PERMIT" == "Y" ] || [ "$PERMIT" == "y" ]
then
        WorkFOLDER
        PROG_STARTS
elif [ "$PERMIT" == "N" ] || [ "$PERMIT" == "n" ]
then
        echo "_____"
        echo ""
        echo "!!!!!!W A R N I N G!!!!!"
        echo "_____"
        NoOVERWRITE
        START_PT
        WEAKpassChck
        #exit
fi
```

- **Displaying available functions for user to select next operation**



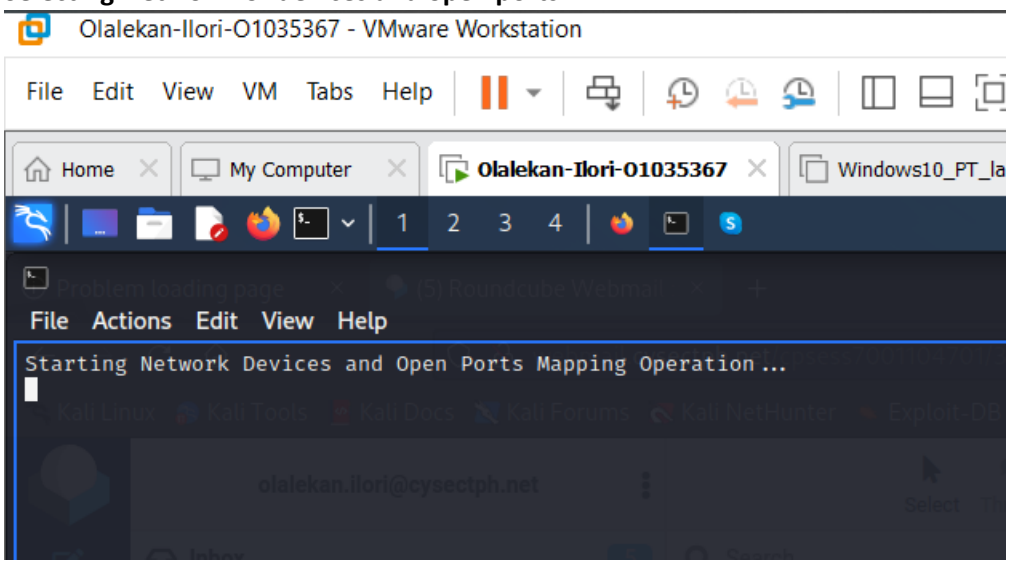

```
function  PROG_STARTS() #function to start running VULNER
{
        clear
        BANNER
        echo -e "Initiate The VULNER Tester"
        echo -e " [01] - Map Network Device & Open Ports\n [02] - Check For Weak Password Usage"
        read -p "Which Operation Will You Like To Carry Out? Select '01' or '02' to proceed: " selFUNC
if [ "$selFUNC" == "01" ]
then
        NetMAPER
        ListCVE
elif [ "$selFUNC" == "02" ]
then
        WEAKpassChck
else
        echo "Enter the right options to proceed....exiting "
        exit
fi
}
```

MAPPING NETWORK DEVICES & OPEN PORTS (screen shots after the script)

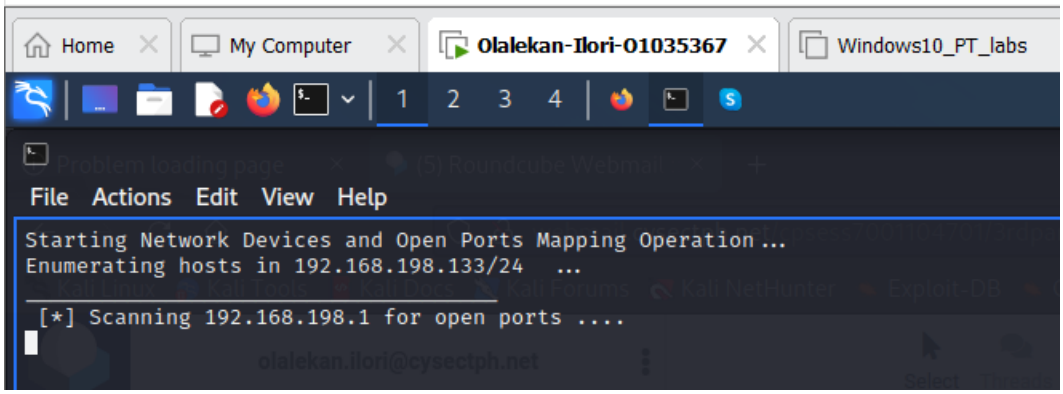
```
function NetMAPER()
{
        start=`date +%s`
        startime=`date +%T`
        clear
        echo "Starting Network Devices and Open Ports Mapping Operation..."
##FINDING THE LAN RANGE OF YOUR NETWORK
        s_nmask="$(ip -br addr show |grep -w eth0 |awk ' {print $3}')"
        echo "[*] Your LAN range is $s_nmask" >> TOOL/REPORT.txt
        DoubleDASH

##FINDING THE LIVE HOSTS IN YOUR NETWORK
        nmap -sn $s_nmask -oX TOOL/slan.xml >/dev/null
        host_lst="$(cat TOOL/slan.xml|grep ipv4|sed 's,\", ,g'|awk ' {print $3}')"
        echo "$host_lst" > TOOL/num.lst
        num1="$(cat TOOL/num.lst |wc -l)"
        echo "[*] Found $num1 live hosts on your LAN" >> TOOL/REPORT.txt
        spacer1

##SAVING OS VERSION FOUND ON HOSTS IN NETWORK INTO REPORT
        echo "Host        ---OS Type" >> TOOL/REPORT.txt
        shortDASH
        for i in $(cat TOOL/num.lst)
        do
         versn="$(nmap -O $i |grep "OS CPE"|awk -F: '{print $4,$5}')"
        if [ -z "$versn" ]
        then
         versn="OS Not Found"
        fi
         echo "$i : $versn" >> TOOL/versn.lst
        done
         cat TOOL/versn.lst >> TOOL/REPORT.txt
         DoubleDASH

##ENUMERATING HOSTS WITH OPEN PORTS IN YOUR LAN AND SAVING AS FILE IN CURRENT FOLDER
        echo "Enumerating hosts in $s_nmask  ... "
        echo "_____________________________________________ "
        #longDASH

        for i in $(cat TOOL/num.lst)
        do
         echo " [*] Scanning $i for open ports ...."
         masscan -p- $i --rate=15000 --banners 2>/dev/null >>TOOL/mass_Scan.lst
         echo " [*] Scanning for open ports on $i completed."
         echo " "
        done
         echo ".......Results of scans for open ports stored in TOOL folder as mass_Scan.lst"
         echo " "
         longDASH

##CALCULATING THE TOTAL NUMBER OF OPEN PORTS AND NUMBER OF HOSTS WITH OPEN PORTS
        echo "$(cat TOOL/mass_Scan.lst |awk ' {print $6}')" > TOOL/open_portList.lst
        num2="$(cat TOOL/open_portList.lst |wc -l)"
```

```
        num3="$(cat TOOL/open_portList.lst|uniq -c|wc -l)"

##SAVING DETAILS HOSTS WITH OPEN PORTS INTO REPORT
        cat TOOL/open_portList.lst|uniq -c|awk '{print $2}' > TOOL/open_host.lst
        echo "[++] Found $num2 Open ports on $num3 hosts in your LAN " >> TOOL/REPORT.txt
        echo " " >> TOOL/REPORT.txt
        echo "Open ports found on the following hosts:" >> TOOL/REPORT.txt
  #echo "[+] $(cat open_host.lst)"
        for i in $(cat TOOL/open_host.lst);do  echo "[+] $i" >> TOOL/REPORT.txt;done

##ENUMERATING SERVICES RUNNING ON HOSTS WITH OPEN PORTS
        ENUMSERVC

##SCANNING FOR COMMON VULNERABILITIES OF OPEN PORTS
        for lhost in $(cat TOOL/open_host.lst)
        do
         nmap  -sV  -F  $lhost -oX TOOL/lhost.xml >/dev/null
         echo "_____________________________________________________________________________________________________________"
>>TOOL/lhostCVE.lst
         echo "| " >>TOOL/lhostCVE.lst
         echo "|                    [*][*]FOUND VULNERABILITIES FOR $lhost[*][*]" >>TOOL/lhostCVE.lst
         #versn2="$(nmap -O $lhost |grep "OS CPE"|awk -F: '{print $4,$5}')"
         searchsploit --nmap TOOL/lhost.xml 2>/dev/null >> TOOL/lhostCVE.lst
        done
        cat TOOL/lhostCVE.lst >> TOOL/REPORT.txt
        echo " "
        echo "Scanning for Vulnerabilities completed successfuly.."
        pathf="$(pwd)"
        echo "Details of common vulnerabilities found is stored in $pathf/TOOL as lhostCVE.lst"
        echo "Report of Entire Scan Stored in $pathf/TOOL/REPORT.txt"
        echo " "
        scandate=`date +%F_%T`
        stoptime=`date +%T`
        end=`date +%s`
        echo "Scan was concluded on: $scandate" >> TOOL/REPORT.txt
        echo "Scan start time: $startime" >> TOOL/REPORT.txt
        echo "Scan stop time: $stoptime" >> TOOL/REPORT.txt
        echo Total duration of scan was `expr $end - $start` seconds >> TOOL/REPORT.txt

        ###DISPLAYING GENERAL STATISTICS ON TERMINAL
        spacerN
        echo " GENERAL SUMMARY OF NETWORK SCAN"
        longDASHn
        echo "[*] Your LAN range is $s_nmask"
        echo "[*] Found $num1 live hosts on your LAN"
        shwOPEN
        longDASHn
        echo "Report Of Services Found"
        cat TOOL/serVersn.lst|grep -B 1 open
        longDASHn
        TIMEStamp
}
```

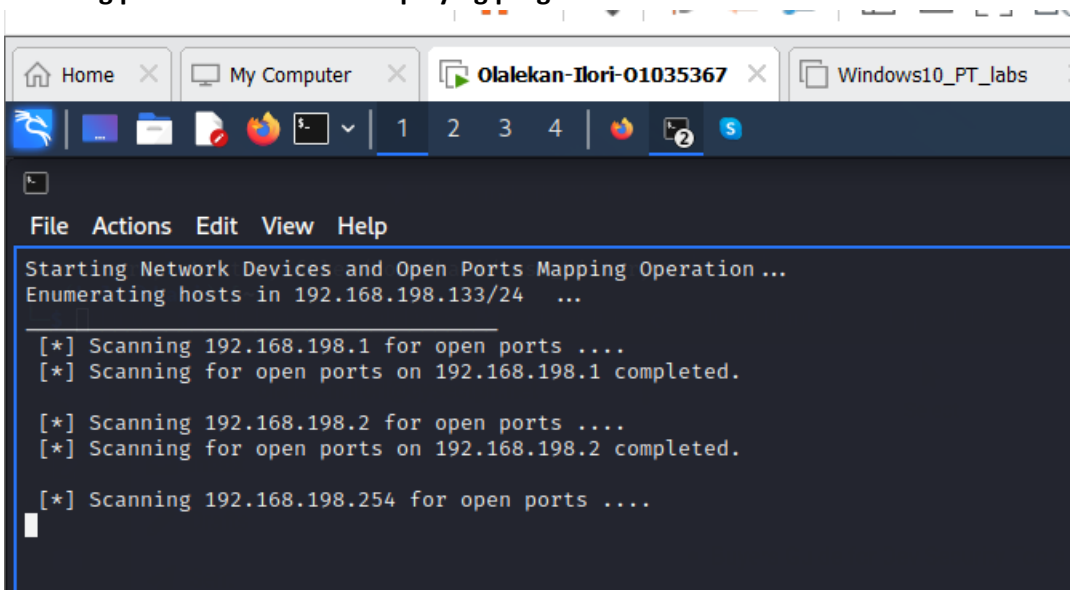
- **Selecting Network for devices and open ports**



- **Scanning process started**



- **Scanning process started and displaying progress**

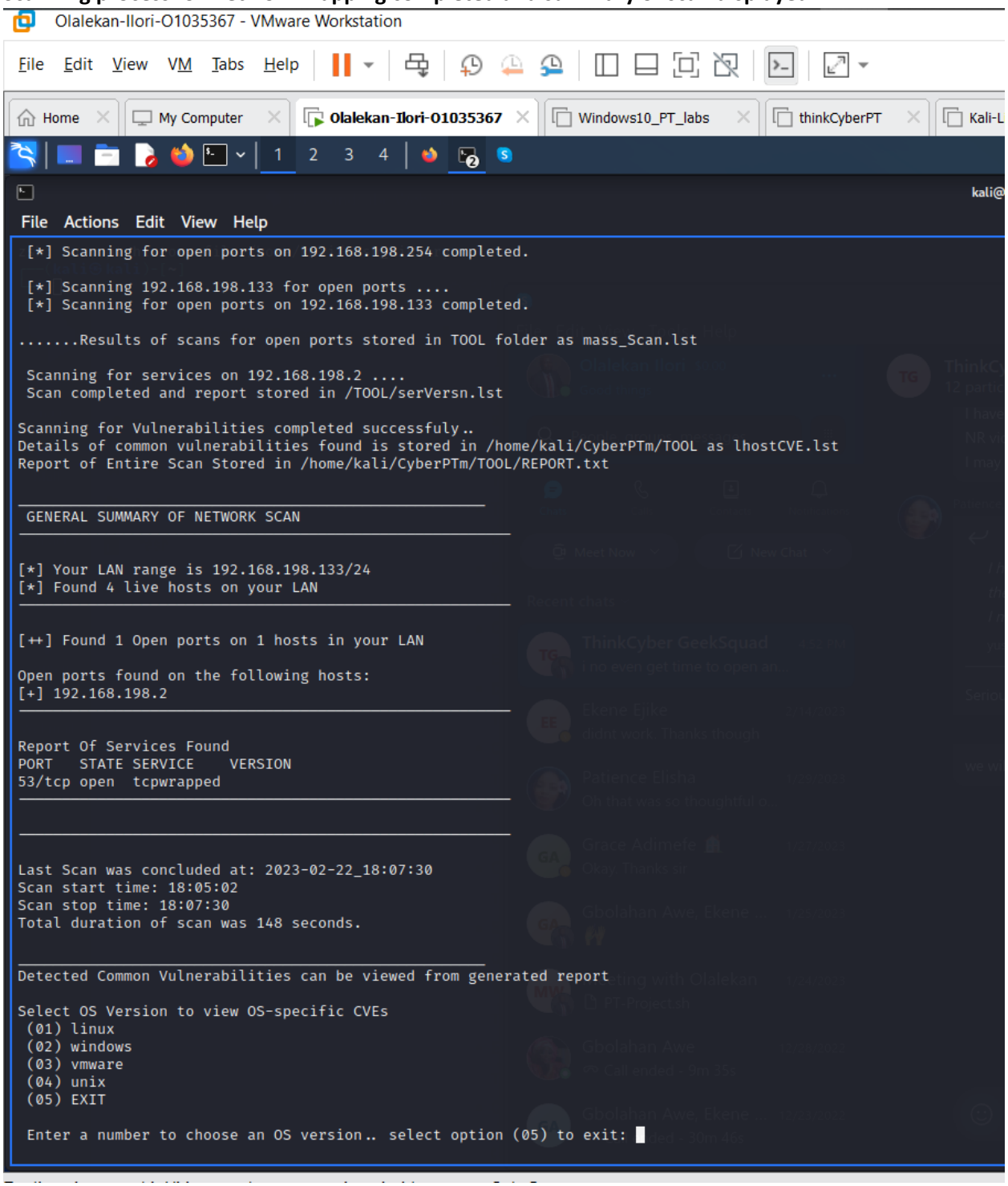


- **Scanning process completed and scanning for services available on open port starts**

-

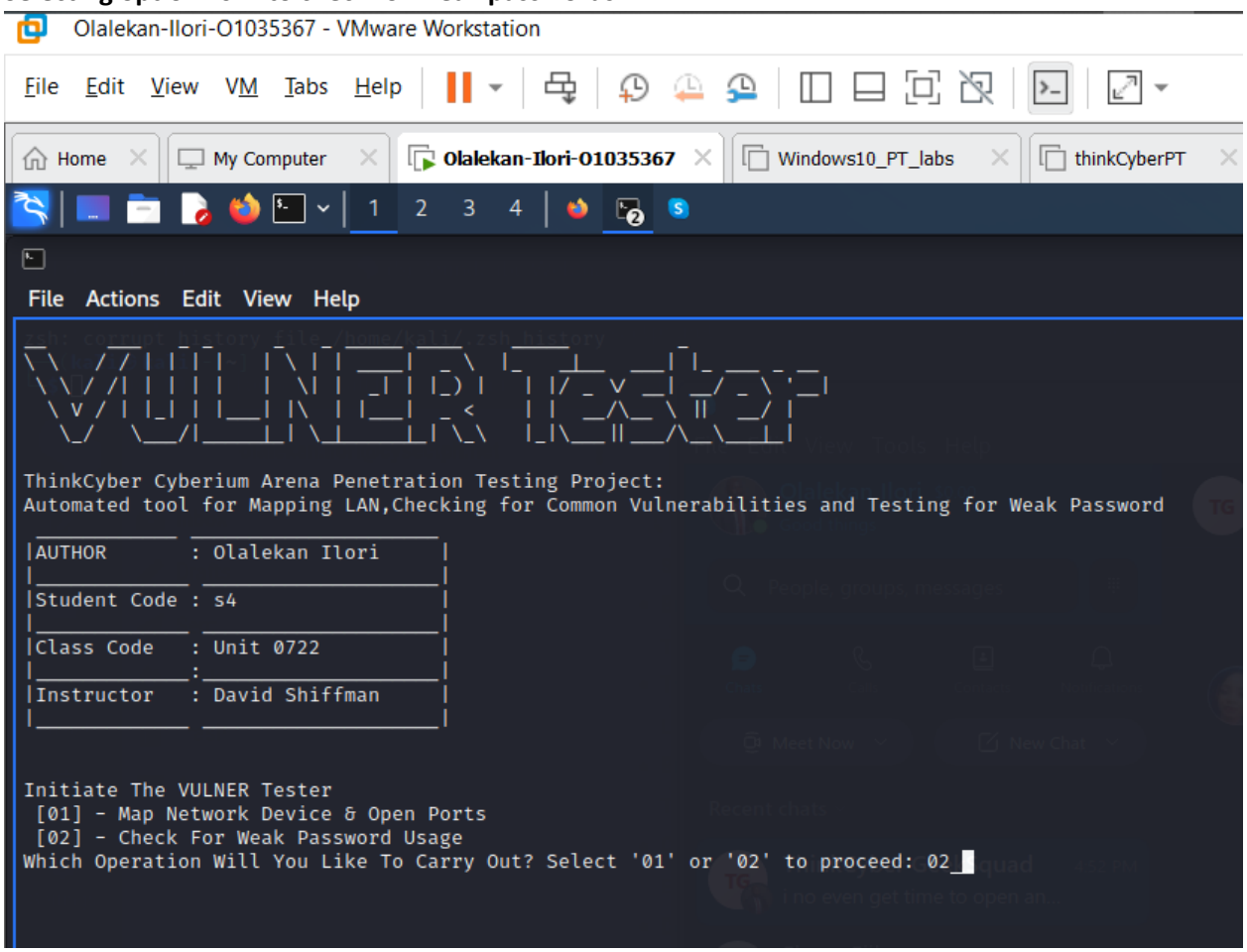
```
Starting Network Devices and Open Ports Mapping Operation ...
Enumerating hosts in 192.168.198.133/24   ...
_____________________________________
 [*] Scanning 192.168.198.1 for open ports ....
 [*] Scanning for open ports on 192.168.198.1 completed.

 [*] Scanning 192.168.198.2 for open ports ....
 [*] Scanning for open ports on 192.168.198.2 completed.

 [*] Scanning 192.168.198.254 for open ports ....
 [*] Scanning for open ports on 192.168.198.254 completed.

 [*] Scanning 192.168.198.133 for open ports ....
 [*] Scanning for open ports on 192.168.198.133 completed.

.......Results of scans for open ports stored in TOOL folder as mass_Scan.lst

 Scanning for services on 192.168.198.2 ....
```

- **Scanning process for Network mapping completed and summary of scan displayed**



CHECK WEAK PASSWORD FUNCTION

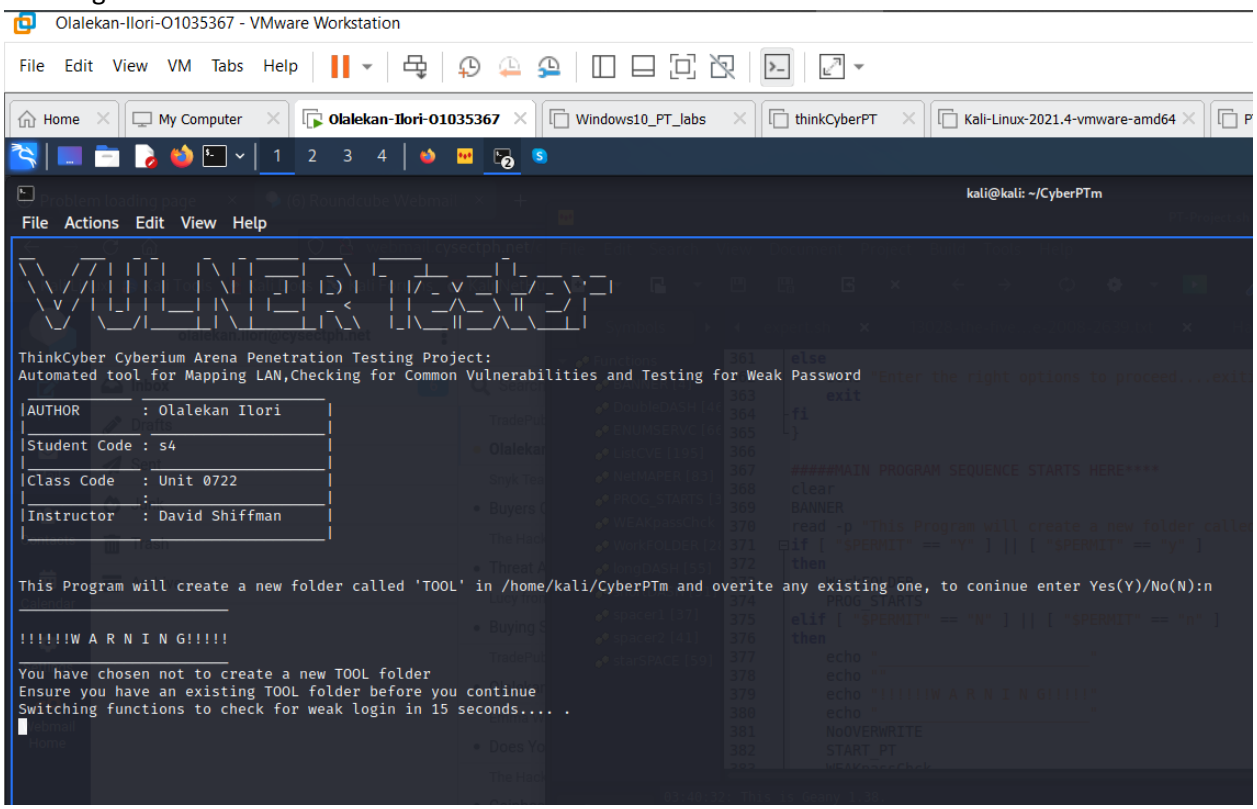
- **Selecting option “02” to check for weak passwords**




```
function BANNER()
{
    figlet "VULNER Tester"
    echo -e "ThinkCyber Cyberium Arena Penetration Testing Project:\nAutomated tool for Mapping LAN,Checking for Common Vulnerabilities and Testing for Weak Password"
    echo " _____________ _______________________"
    echo "|AUTHOR      : Olalekan Ilori     |"
    echo "|_____________ _______________________|"
    echo "|Student Code : s4              |"
    echo "|_____________ _______________________|"
    echo "|Class Code   : Unit 0722        |"
    echo "|_____________:_______________________|"
    echo "|Instructor   : David Shiffman     |"
    echo "|_____________ _______________________|"
    echo -e "  \n "
}
function  PROG_STARTS() #function to start running VULNER
{
    clear
    BANNER
    echo -e "Initiate The VULNER Tester"
    echo -e " [01] - Map Network Device & Open Ports\n [02] - Check For Weak Password Usage"
    read -p "Which Operation Will You Like To Carry Out? Select '01' or '02' to proceed: " selFUNC
if [ "$selFUNC" == "01" ]
then
    NetMAPER
```

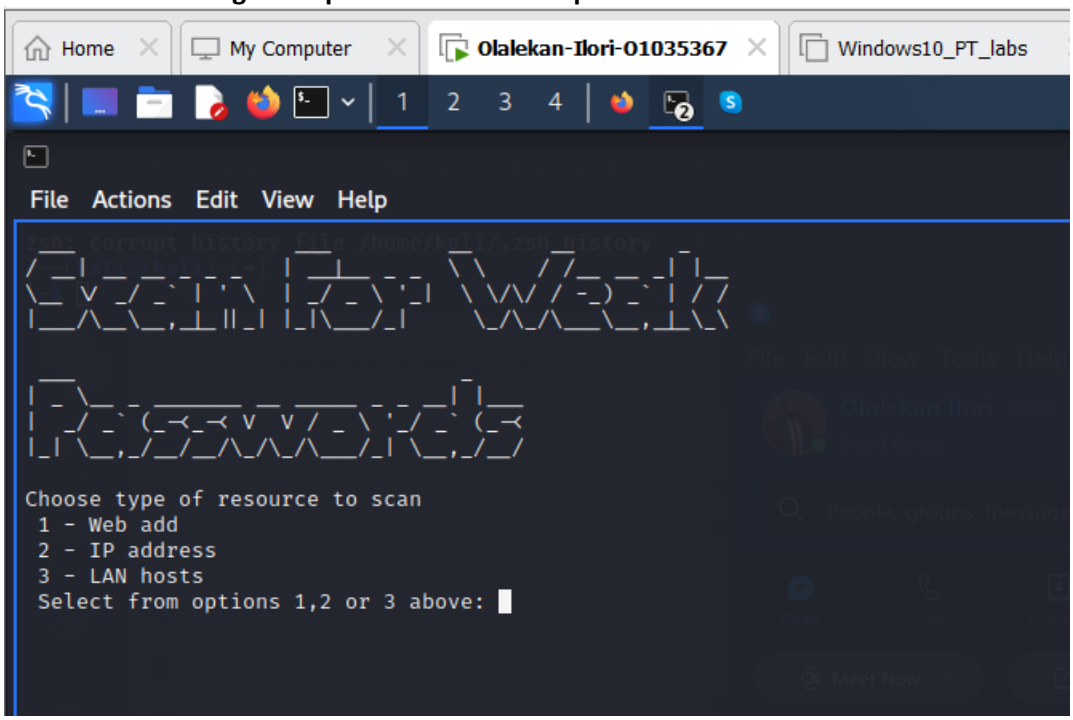
```
        ListCVE
    elif [ "$selFUNC" == "02" ]
    then
        WEAKpassChck
    else
        echo "Enter the right options to proceed....exiting "
        exit
    fi
    }
```

Warning notice



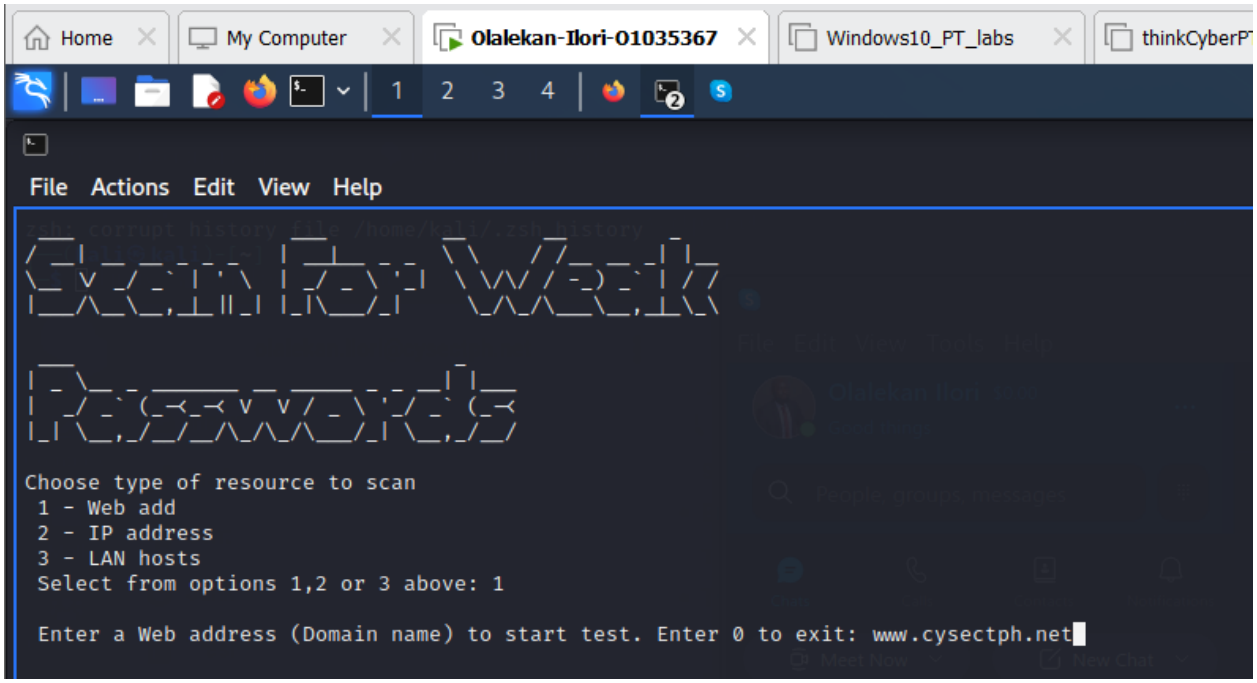

```
BANNER
read -p "This Program will create a new folder called 'TOOL' in $(pwd) and overite any existing one, to continue enter Yes(Y)/No(N):" PERMIT
if [ "$PERMIT" == "Y" ] || [ "$PERMIT" == "y" ]
then
        WorkFOLDER
        PROG_STARTS
elif [ "$PERMIT" == "N" ] || [ "$PERMIT" == "n" ]
then
        echo "_____________________________"
        echo ""
        echo "!!!!!!W A R N I N G!!!!!"
        echo "_____________________________"
        NoOVERWRITE
        START_PT
        WEAKpassChck
        #exit
fi
```

Banner for checking weak password function options

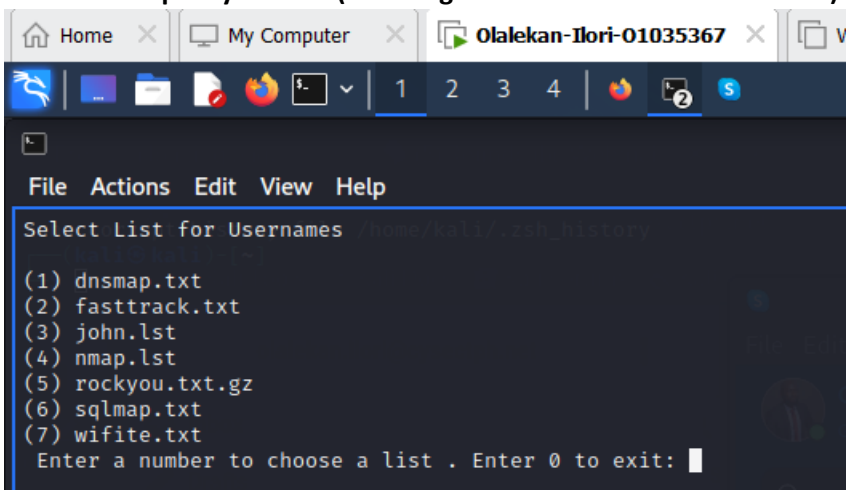



```
function START_PT()
{
clear
figlet -f small "Scan For Weak Passwords"
echo "Choose type of resource to scan"
echo " 1 - Web add"
echo " 2 - IP address"
echo " 3 - LAN hosts"
read -p " Select from options 1,2 or 3 above: " opChos
}
```

- **Selecting option to provide url to test for weak password**



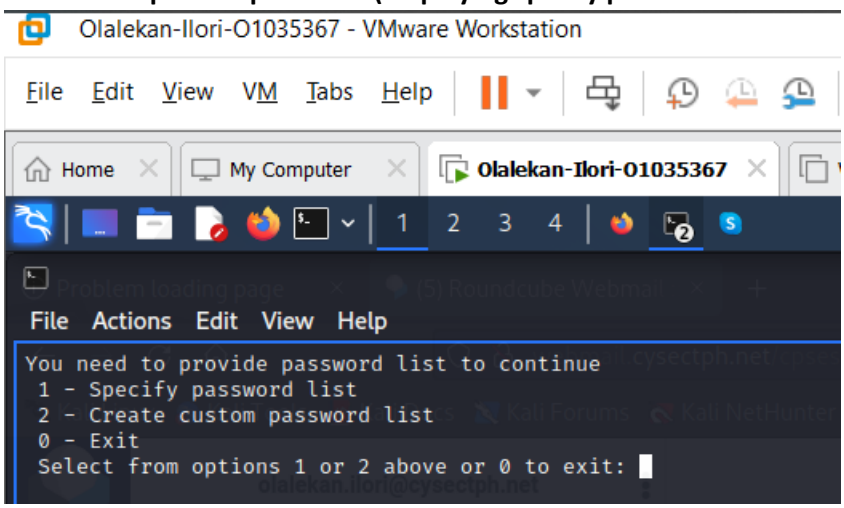
- **Function to specify user list(selecting from wordlist available on kali)**




```
function SELECT_USER()
 {
  echo "Select List for Usernames"
  echo " "
  echo $(ls /usr/share/wordlists) > tryuser.lst
  nos=0

          if [ -a tryuser2.lst ]
                  then
                  rm tryuser2.lst
          fi
          for user in $(cat tryuser.lst|tr " " "\n"|grep "txt\|lst")
                  do
                          let nos+=1
                          echo "($nos) $user" >>tryuser2.lst
                  done
  cat tryuser2.lst
  read -p " Enter a number to choose a list . Enter 0 to exit: " Clist
```

```
        if [ $Clist == 0 ]
        then
                exit
        fi
listitem="$(cat tryuser2.lst|grep $Clist|awk '{print $2}')"  #determines list to use based on the number chosen as Clist
wordlist="/usr/share/wordlists/$listitem"
clear
}
```

- **Function to provide password (Displaying specify password list or create)**




```
function SELECT_PASS()
 {
        echo "Select List for Password"
        echo " "
#echo $(ls /usr/share/wordlists) > tryuser.lst
        if [ -a trypass.lst ]
                then
                rm trypass.lst   #removing file if already existing
        fi
ncount=0
        for user in $(cat tryuser.lst|tr " " "\n"|grep "txt\|lst")
                do
                        let ncount+=1
                        echo "[$ncount] $user" >>trypass.lst
                done
        cat trypass.lst
        read -p " Enter a number to choose a list . Enter 0 to exit: " Cplist
        if [ $Cplist == 0 ]
        then
                exit
        fi
plsitem="$(cat trypass.lst|grep $Cplist|awk '{print $2}')"
passlist="/usr/share/wordlists/$plsitem"
clear
 }
```

- **Function to create password list selected**

```
You need to provide password list to continue
 1 - Specify password list
 2 - Create custom password list
 0 - Exit
 Select from options 1 or 2 above or 0 to exit: 2
 Enter a list of names seperated by comma to create a password list admin,msf,password,123456,olalekan,david
```

```
function CREATE_PASS()
 {
read -p " Enter a list of names seperated by comma to create a password list " PassList
echo "$PassList" > PassList.lst
cat PassList.lst|tr "," "\n" > pass.lst
 }
```

- **Brute forcing with Medusa to check for weak password on selected resource**

```
[==========]
Checking 162.0.209.87 for weak passwords using following parameters:
 [*] Username list as  /usr/share/wordlists/john.lst
 [*] Password list as /usr/share/wordlists/dnsmap.txt
[==========]
Checking for login services .....

2 Login services found

2 login services found, bruteforcing first login service
ERROR: Thread EFA1D6C0: Host: 162.0.209.87 Cannot connect [unreachable], retrying (1 of 3 retries)
ERROR: Thread EFA1D6C0: Host: 162.0.209.87 Cannot connect [unreachable], retrying (2 of 3 retries)
ERROR: Thread EFA1D6C0: Host: 162.0.209.87 Cannot connect [unreachable], retrying (3 of 3 retries)
NOTICE: ftp.mod: failed to connect, port 21 was not open on 162.0.209.87
No Weak login found on host/Network
Scan was concluded on:
Scan start time: 03:49:12
Scan stop time: 03:55:54
Total duration of scan was 402 seconds.
```

```
function BRUTE_TEST( )
 {
    cat loginSrvc.lst|awk '{print $3}' >login.lst
    echo " "
    if [ "$loginSrvc" == 1 ]
    then
            service="$(cat login.lst)"
            echo " Bruteforcing hosts for login...."
            #hydra -L $wordlist -P $passlist $Chost $service -t 5
            medusa -h $Chost -U $wordlist -P $passlist -M $service  -O medusaBRUTE.txt -b  > /dev/null
```

```
	else
		echo "$loginSrvc login services found, bruteforcing first login service"
		service="$(cat login.lst|head -1)"
		medusa -h $Chost -U $wordlist -P $passlist -M $service -O medusaBRUTE.txt -b  > /dev/null
	fi
	sucsChck="$(cat medusaBRUTE.txt|grep -i success|uniq |awk '{print $NF}')"
	if [ "$sucsChck" == "[SUCCESS]" ]
	then
		validName="$(cat medusaBRUTE.txt|grep -i success|uniq |awk '{print $7}')"
		validPass="$(cat medusaBRUTE.txt|grep -i success|uniq |awk '{print $9}')"
		echo "Successful login attempt found on $Chost using Username- $validName and Password- $validPass
"
	else
		echo " "
		echo "No Weak login found on host/Network"
	fi

}
```