CFC Project 3 (Remote control)

CFC020823

Network Research

Zulkarnaen

S17

Objective

1. Installations and Anonymity Check

1.1 Install the needed applications

1.2 If the applications are already installed, don’t install them again

1.3 Check if the network connection is anonymous; if not, alert the user and exit

1.4 If the network connection is anonymous, display the spoofed country name

1.5 Allow the user to specify the address to scan via remote server; save into a variable

2. Automatically Connect and Execute Commands on the Remote Server via SSH

2.1 Display the details of the remote server (country, IP, and Uptime)

2.2 Get the remote server to check the Whois of the given address

2.3 Get the remote server to scan for open ports on the given address

3. Results

3.1 Save the Whois and Nmap data into files on the local computer

3.2 Create a log and audit your data collecting

1. Installations and Anonymity Check

- Used Geany (Graphical (GUI) Text Editor) to create and test script more efficiently.
- Used a function to check if nipe is installed in the machine.

```
#1.1
echo "Checking if all neccesary applications are installed..."

nipe_file=$(find . -type f -name nipe.pl | wc -l)
tor_file=$(find . -type d -name tor-browser | wc -l)

function nipers
{
    if [ $nipe_file -gt 0 ]
    then
    echo "nipe is already installed"
    else
    echo "Installing nipe..."
    sudo git clone https://github.com/htrgouvea/nipe && cd nipe
    sudo cpanm --installdeps .
    sudo perl nipe.pl install
    fi
}
```

1. Installations and Anonymity Check

- Used another function to check if tor is installed in the machine.

```
function torbrowser
{
    if [ $tor_file -gt 0 ]
    then
    echo "tor is already installed"
    else
    echo "Installing tor..."
    sudo apt install torbrowser-launcher
    fi
}
```

1. Installations and Anonymity Check

- Used if else to check for the remaining applications sshpass, nmap, geoipbin.
- Called in the 2 previous functions into this if loop.

```
if command -v sshpass &> /dev/null && command -v nmap &> /dev/null && command -v geoiplookup &> /dev/null
then
    echo "sshpass is already installed"
    echo "nmap is already installed"
    echo "geoipbin is already installed"
    nipers
    torbrowser

else
    echo "Installing sshpass, nmap and geoipbin..."
    sudo apt-get install sshpass nmap tor nipe -y
    echo "Installed sshpass, nmap and geoipbin successfully"
fi
```

1. Installations and Anonymity Check

- Using nipe.pl to gain anonymity.
- Changed directory to nipe folder to allow sudo perl nipe.pl.
- As sometimes nipe may not start the first time, a condition loop is used to loop as long as nipe status is not true.
- Used geoiplookup to check if we have a spoofed IP address.

```
#1.3
echo "Checking if connection is anonymous..."
cd /home/kali/Desktop/tor_dir/nipe

nipestat=$(sudo perl nipe.pl status | grep "Status" | awk -F: '{print $2}')
spoofip=$(sudo perl nipe.pl status | grep "Ip" | awk -F: '{print $2}')

while [ $nipestat != true ]
do

    sudo perl nipe.pl restart
    sudo perl nipe.pl status

done

IPxCOUNT=$(geoiplookup $spoofip | awk -F: '{print $2}')

echo "Your connection is anonymous..."
```

1. Installations and Anonymity Check

- Displaying spoofed IP address on terminal using echo command.
- Displaying spoofed country on terminal using echo command.
- Requesting user to input domain or Ip address to scan on the remote server.
- Used read command to store input as a variable.

```
#1.4
echo "Your spoofed IP is : $spoofip "
echo "Your spoofed country is: $IPxCOUNT "

#1.5
echo "Specify a Domain/Ip address to scan: "
read DOMAIN
```

2. Automatically Connect and Execute Commands on the Remote Server via SSH

- Using sshpass to connect to remote server and running commands automatically.
- Displaying remote server's Uptime, IP address and country data on terminal.

```
#2.1
echo "Connecting to remote server..."

#2.2
UPT=$(sshpass -p kali ssh kali@192.168.126.131 'uptime')
echo "Uptime: $UPT"
IPAD=$(sshpass -p kali ssh kali@192.168.126.131 'hostname -I')
echo "IP address: $IPAD"
COUNT=$(sshpass -p kali ssh kali@192.168.126.131 'whois 192.168.126.131 | grep -i country | sort | uniq')
echo "$COUNT"
```

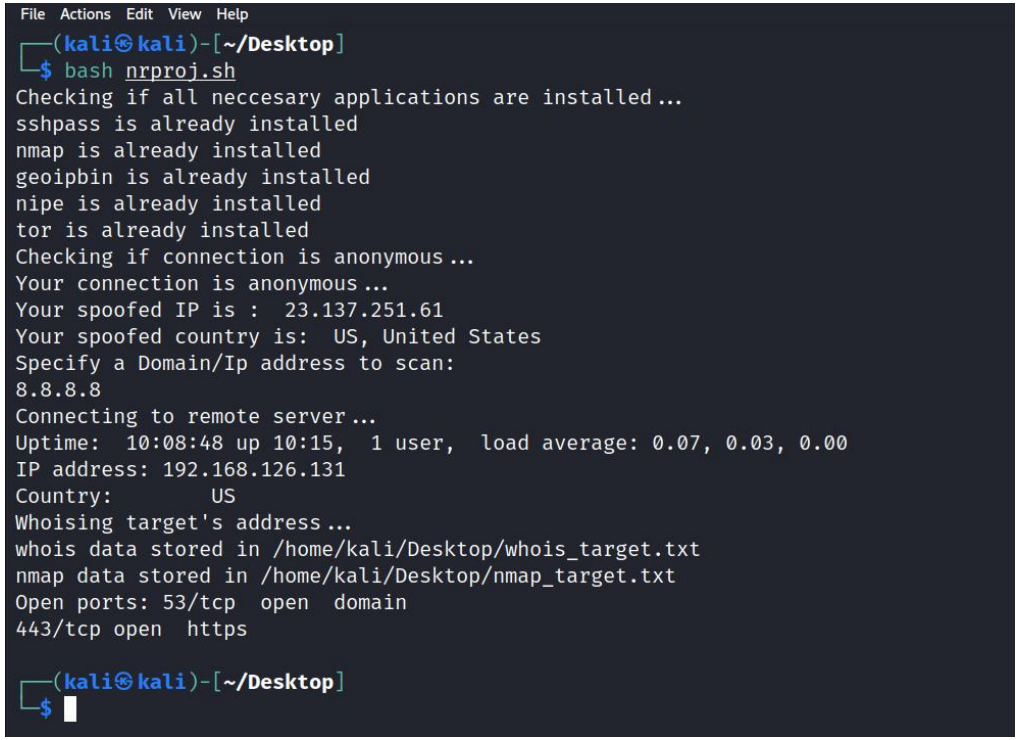
3. Results

- Whois and Nmap domain or ip address input by user through remote server.
- Storing results in separate files and a log file.
- Checking which ports are open on domain or ip address input by user and displaying on terminal.

```
#2.3 & 3.1 & 3.2

DATE=$(date)

echo "Whoising target's address..."
sshpass -p kali ssh kali@192.168.126.131 whois $DOMAIN > /home/kali/Desktop/whois_target.txt
echo "whois data stored in /home/kali/Desktop/whois_target.txt"
echo "$DATE : whois data from $DOMAIN collected" >> /home/kali/Desktop/script.log

sshpass -p kali ssh kali@192.168.126.131 nmap $DOMAIN > /home/kali/Desktop/nmap_target.txt
echo "nmap data stored in /home/kali/Desktop/nmap_target.txt"
echo "$DATE : nmap data from $DOMAIN collected" >> /home/kali/Desktop/script.log

OPPRT=$(sshpass -p kali ssh kali@192.168.126.131 nmap $DOMAIN | grep open)
echo "Open ports: $OPPRT"
```

Output in terminal

Script output.



Log file output.