Network Research Project

Prepared By:
Michelle Lai

CFC020823

S23

PROJECT OBJECTIVES

1 Installations and Anonymity Check

- Install the needed applications
- If the applications are already installed, don't install them again
- Check if the network connection is anonymous; if not, alert the user and exit
- If the network connection is anonymous, display the spoofed country name
- 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

- Display the details of the remote server (Country, IP, and Uptime)
- Get the remote server to check the Whois of the given address
- Get the remote server to scan for open ports on the given address

3 Results

- Save the Whois and Nmap data into files on the local computer
- Create a log and audit your data collection

INSTALLATIONS AND ANONYMITY CHECK

 # Install the needed applications
 # If the applications are already installed, don't install them again

```
function WHOIS()
{
    if command -v whois >/dev/null;
    then
        echo "whois is already installed."
    else
        echo "whois is not installed."
        sudo apt-get update
        sudo apt install whois
    fi
    }
WHOIS
```

WHOIS is a function to check if the program 'whois' is being installed. If the program is being installed, it will return "whois is already installed." If not, it will return "whois is not installed." and will proceed to install the program.

```
command -v whois >/dev/null;
```

Search for the PATH environment variable for the program and return its location if it is found

```
└─$ command -v whois
/usr/bin/whois
```

/dev/null: A null device
As long as there is a path, the IF condition will come back as true

Credits to Saturn Cloud:
https://saturncloud.io/blog/how-to-check-if-a-program-exists-from-a-bash-script/

INSTALLATIONS AND ANONYMITY CHECK

 # Check if the network connection is anonymous; if not, alert the user and exit
 # If the network connection is anonymous, display the spoofed country name

```
#1.3 Check if the network connection is anonymous; if not, alert the user and exit
status=$(sudo perl nipe.pl status| grep -i status | awk '{print$3}')
if $status == true;
then
    echo "You are annoymous."
        #1.4 If the network connection is anonymous, display the spoofed country name
        Ipx=$(sudo perl nipe.pl status| grep -i ip | awk '{print$3}')
        Country=$(geoiplookup $Ipx | awk '{print$5$6}')
        echo "Your Spoofed IP Address is $Ipx"
        echo "Spoofed Country is $Country."
            echo "Connecting to Remote Server..."
            sshpass -p 'kali' ssh kali@192.168.254.130 "$(declare -f detail);detail"

else
    echo "You are not annoymous."
```

After nipe is start (to make the user anonymous), 'IF' condition is being used to confirm if nipe is running. If the status of nipe is true, user will be informed of his spoofed IP address and country. It will also auto connect to the remote server (192.168.254.130). If the status of nipe is not true, user will be informed that he is not anonymous and it will not connect to the remote server.

```
sshpass -p 'kali' ssh kali@192.168.254.130 "$(declare -f detail);detail"
```

Sshpass is used to ssh into the remote server while executing a function.
To note: SSH has to be running at the remote server for the sshpass to work.

This prints the definition of the function; detail is a function (see next page)

Output:

```
You are anonymous.
Your Spoofed IP Address is 192.42.116.187
Spoofed Country is Netherlands.
```

AUTOMATICALLY CONNECT AND EXECUTE COMMANDS ON THE REMOTE SERVER VIA SSH

Function: detail

```
function detail()
{
    #1.5 Allow the user to specify the address to scan via remote server; save into a variable
    echo "Specify a Domain/IP address to scan: "
    read DOMAIN

    #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)
    UPT=$(uptime)
    echo "Remote Server Uptime: $UPT"
    Public_IP=$(curl -s ifconfig.io)
    echo "Remote Server IP address: $Public_IP"
    Ctry=$(geoiplookup $Public_IP | awk '{print$5}')
    echo "Remote Server Country: $Ctry"

    #2.2 Get the remote server to check the Whois of the given address
    cd /home/kali/Desktop
    whois $DOMAIN >> /home/kali/Desktop/whois_$DOMAIN.txt

    #2.3 Get the remote server to scan for open ports on the given address
    nmap $DOMAIN -oN /home/kali/Desktop/nmap_$DOMAIN.txt

    #3. Results
    #3.1 Save the Whois and Nmap data into files on the local computer
    sshpass -p "kali" scp /home/kali/Desktop/whois_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data
    echo "Whois data was saved into /home/kali/Desktop/Data/whois_$DOMAIN.txt"
    sshpass -p "kali" scp /home/kali/Desktop/nmap_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data
    echo "Nmap scan was saved into /home/kali/Desktop/Data/nmap_$DOMAIN.txt"

    #3.2 Create a log and audit your data collection
    DATE=$(date)
    echo "$DATE Nmap data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log
    echo "$DATE whois data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log
    sshpass -p "kali" scp /home/kali/Desktop/nr.log 192.168.254.129:/home/kali/Desktop/Data
    echo "Data log for Nmap and whois is created in /home/kali/Desktop/nr.log"
        }
```

Commands in function 'detail' will be explained in the next few pages.

AUTOMATICALLY CONNECT AND EXECUTE COMMANDS ON THE REMOTE SERVER VIA SSH

 # Allow the user to specify the address to scan via remote server; save into a variable
 # Display the details of the remote server (Country, IP, and Uptime)

```
#1.5 Allow the user to specify the address to scan via remote server; save into a variable
echo "Specify a Domain/IP address to scan: "
read DOMAIN

#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)
UPT=$(uptime)
echo "Remote Server Uptime: $UPT"
Public_IP=$(curl -s ifconfig.io)
echo "Remote Server IP address: $Public_IP"
Ctry=$(geoiplookup $Public_IP | awk '{print$5}')
echo "Remote Server Country: $Ctry"
```

User specifies the domain/ip address he wants to scan. The input will be the 'DOMAIN'. After inputting the domain, it will inform the user of the remote server's details (Country, IP, and Uptime).

Output:

```
Specify a Domain/IP address to scan:
cnn.com
Remote Server Uptime:  00:51:25 up  2:44,  2 users,  load average: 0.31, 0.24, 0.20
Remote Server IP address: 220.255.
Remote Server Country: Singapore
```

AUTOMATICALLY CONNECT AND EXECUTE COMMANDS ON THE REMOTE SERVER VIA SSH

 # Get the remote server to check the Whois of the given address
 # Get the remote server to scan for open ports on the given address
 # Save the Whois and Nmap data into files on the local computer

```
#2.2 Get the remote server to check the Whois of the given address
cd /home/kali/Desktop
whois $DOMAIN >> /home/kali/Desktop/whois_$DOMAIN.txt

#2.3 Get the remote server to scan for open ports on the given address
nmap $DOMAIN -oN /home/kali/Desktop/nmap_$DOMAIN.txt

#3. Results
#3.1 Save the Whois and Nmap data into files on the local computer
sshpass -p "kali" scp /home/kali/Desktop/whois_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data
echo "Whois data was saved into /home/kali/Desktop/Data/whois_$DOMAIN.txt"
sshpass -p "kali" scp /home/kali/Desktop/nmap_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data
echo "Nmap scan was saved into /home/kali/Desktop/Data/nmap_$DOMAIN.txt"
```

Command: whois $DOMAIN >> /home/kali/Desktop/whois_$DOMAIN.txt

- Saves the result of the whois command to a text file

Command: nmap $DOMAIN -oN /home/kali/Desktop/nmap_$DOMAIN.txt

- Saves the result of nmap to a text file

Command: sshpass -p "kali" scp /home/kali/Desktop/whois_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data

- copy the whois text file from the remote server (192.168.254.130) to the local computer (192.168.254.129)

Command: sshpass -p "kali" scp /home/kali/Desktop/nmap_$DOMAIN.txt 192.168.254.129:/home/kali/Desktop/Data

- copy the nmap text file from the remote server (192.168.254.130) to the local computer (192.168.254.129)

To note:

- For the scp command to run, ssh has to be running in the local computer.
- A directory 'Data' was created in the local computer to keep the whois and nmap text file.

RESULTS

Create a log and audit your data collection

```
DATE=$(date)
echo "$DATE Nmap data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log
echo "$DATE whois data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log
sshpass -p "kali" scp /home/kali/Desktop/nr.log 192.168.254.129:/home/kali/Desktop/Data
echo "Data log for Nmap and whois is created in /home/kali/Desktop/nr.log"
```

Command: DATE=$(date)

- DATE is a terminal variable that stores the output from the command 'date'.

Command: echo "$DATE Nmap data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log

- creates a log that indicates that Nmap data for the 'DOMAIN' has been collected.

Command: echo "$DATE whois data collected for: $DOMAIN" >> /home/kali/Desktop/nr.log

- creates a log that indicates that whois data for the 'DOMAIN' has been collected.

Command: sshpass -p "kali" scp /home/kali/Desktop/nr.log 192.168.254.129:/home/kali/Desktop/Data

- copy the nr.log from the remote server to the local computer (192.168.254.129)

Output of nr.log:

```
└─$ cat nr.log
Thu Oct 12 12:49:28 AM +08 2023 Nmap data collected for: 8.8.8.8
Thu Oct 12 12:49:28 AM +08 2023 whois data collected for: 8.8.8.8
Thu Oct 12 12:51:12 AM +08 2023 Nmap data collected for: scanme.nmap.com
Thu Oct 12 12:51:12 AM +08 2023 whois data collected for: scanme.nmap.com
Thu Oct 12 12:52:22 AM +08 2023 Nmap data collected for: cnn.com
Thu Oct 12 12:52:22 AM +08 2023 whois data collected for: cnn.com
Thu Oct 12 01:19:02 AM +08 2023 Nmap data collected for: google.com
Thu Oct 12 01:19:02 AM +08 2023 whois data collected for: google.com
```

OUTPUT OF THE SCRIPT

```
┌──(kali㉿kali)-[~/Desktop]
└─$ bash nrproject.sh
Student name: Michelle Lai
Student code: S23
Class Code: CFC020823
Lecturer name: Tushar

nipe is already installed.
sshpass is already installed.
tor is already installed.
nmap is already installed.
whois is already installed.
geoip is already installed.
You are anonymous.
Your Spoofed IP Address is 185.220.101.2
Spoofed Country is Germany.
Connecting to Remote Server ...
Specify a Domain/IP address to scan:
8.8.8.8
Remote Server Uptime:  01:50:51 up  3:43,  1 user,  load average: 0.17, 0.16, 0.17
Remote Server IP address: 220.255.
Remote Server Country: Singapore
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-12 01:50 +08
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.0060s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.90 seconds
Whois data was saved into /home/kali/Desktop/Data/whois_8.8.8.8.txt
Nmap scan was saved into /home/kali/Desktop/Data/nmap_8.8.8.8.txt
Data log for Nmap and whois is created in /home/kali/Desktop/nr.log
```