This is bash script main objective is to anonymously communicate to a remote server and execute tasks to get the details and scanned target ip address or domain. The actual public IP will be hidden using another tool as default gateway .

The following executions in the bash script are:

1. An introduction or banner message will display and ask if want to proceed with running the script. Once confirmed, it will execute the command to check if the needed tools are already installed. If not, it will be installed and messages wil be displayed once all installation are completed or if tools already installed.

Tools or applications to be installed are:

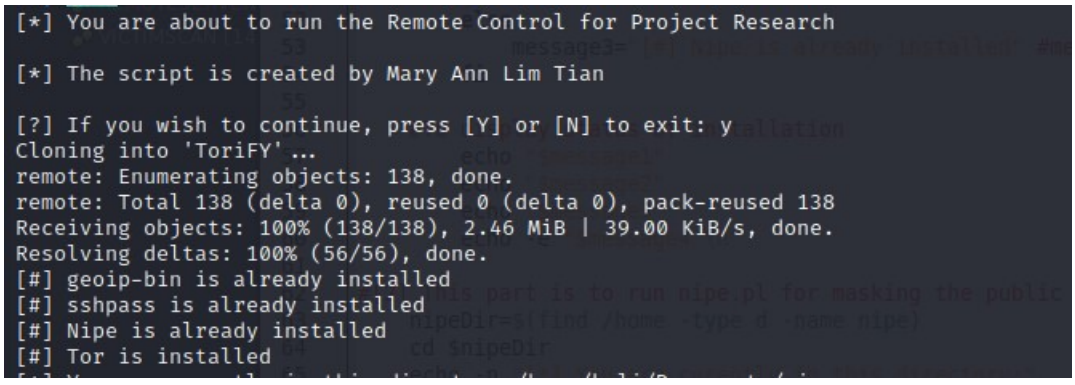
a. Geoip-bin - tool to check ip address or domain location or origin

b. SSHPASS - utility to run SSH using the keyboard-interactive password authentication mode, but in a non-interactive way.

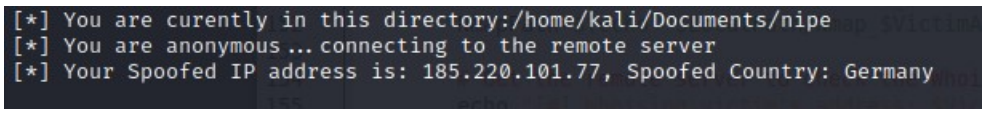
c. ToriFY - is a simple wrapper (shell script) that attempts to find the best underlying Tor wrapper available on a system.

d. Nipe - tool use to route the traffic from your machine to the Internet through Tor network, so you can surf the Internet masking or protecting your public ip from being exposed.

Results:



3. To start the nipe in order to get a spoof IP and identify the spoof country to be used as mask public ip when accessing remote server or surfing the net.



4. to get the target domain or ip address to scan for information

```
[?]Please specify a Domain/IP address to scan:
cnn.com
[?] Do you want to continue to scan address of cnn.com via Remote Server? [Y/N]
```

5. To get the remote server details to be used to execute the tasks of target's domain or ip address detail information. Password input is also mask for security purposes.

```
[***] Please provide the following remote server details to use to scan address [***]
Enter Remote Server IP Address:
192.168.170.130
Enter Remote Server Username:
tc
Enter Remote Server Password for :
```

6. To execute remote server information when it has been successfully connected.

```
Connecting to Remote Server....
[*] Uptime:  06:20:59 up 18:05,  1 user,  load average: 0.18, 0.17, 0.18
[*] IP Address: 192.168.170.130
[*] Country: Address not found
```

if unsuccessful connection, it will ask if want to try another remote server

```
Connecting to Remote Server....
[?] Unable to connect. Try another remote server? [Y/N]
y
[***] Please provide the following remote server details to use to scan address [***]
Enter Remote Server IP Address:
```

7. To execute the Whois and Nmap to get the information on the target domain or ip address

```
[#] Whoising victim's address: cnn.com
[#] Whois data was saved into: /home/kali/Documents/nipe/Whois_cnn.com
[#] Scanning victim's address: cnn.com
[#] Nmap scan was saved into: /home/kali/Documents/nipe/Nmap_cnn.com
[?] Do you want to scan another Domain/IP address? [Y/N]: y
[?] Please specify a Domain/IP address to scan: academy.cyberiumarena.com
[#] Whoising victim's address: academy.cyberiumarena.com
[#] Whois data was saved into: /home/kali/Documents/nipe/Whois_academy.cyberiumarena.com
[#] Scanning victim's address: academy.cyberiumarena.com
[#] Nmap scan was saved into: /home/kali/Documents/nipe/Nmap_academy.cyberiumarena.com
[?] Do you want to scan another Domain/IP address? [Y/N]: n
```

a. Whois - is a command searches for an object in a WHOIS database (a public database houses the information collected when someone registers a domain name or updates their DNS settings)

b Nmap - is a tool that is used to scan IP addresses and ports in a network and to detect installed applications.

8. To display where the scanned audit log called nr.log is stored and option to show it's content

```
[#] Scanned Whois and Nmap timestamp located in /var/log/nr.log
[?] Do you like to open the /var/log/nr.log file? [Y/N]: y
[#] Wed Jan 18 01:13:29 AM EST 2023- [*] Whois data collected for: 8.8.8.8
[#] Wed Jan 18 01:19:02 AM EST 2023- [*] Nmap data collected for: 8.8.8.8
[#] Wed Jan 18 01:21:01 AM EST 2023- [*] Whois data collected for: cnn.com
[#] Wed Jan 18 01:21:18 AM EST 2023- [*] Nmap data collected for: cnn.com
[#] Wed Jan 18 01:21:47 AM EST 2023- [*] Whois data collected for: academy.cyberiumarena.com
[#] Wed Jan 18 01:22:17 AM EST 2023- [*] Nmap data collected for: academy.cyberiumarena.com
```

Project Output:

```
┌──(kali㉿kali)-[~/Documents]
└─$ sudo bash remotecontrol1.sh
[*] You are about to run the Remote Control for Project Research

[*] The script is created by Mary Ann Lim Tian

[?] If you wish to continue, press [Y] or [N] to exit: y
[#] geoip-bin is already installed
[#] sshpass is already installed
[#] Nipe is already installed
[#] Tor is already installed

[*] You are curently in this directory:/home/kali/Documents/nipe
[*] You are anonymous ... connecting to the remote server
[*] Your Spoofed IP address is: 193.189.100.198, Spoofed Country: Sweden

[?]Please specify a Domain/IP address to scan:
academy.cyberiumarena.com
[?] Do you want to continue to scan address of academy.cyberiumarena.com via Remote Server? [Y/N]: y


[**] Please provide the following remote server details to use to scan address [**]
Enter Remote Server IP Address:
192.168.170.130
Enter Remote Server Username:
tc
Enter Remote Server Password for :


Connecting to Remote Server....
[?] Unable to connect. Try another remote server? [Y/N]
y
[**] Please provide the following remote server details to use to scan address [**]
Enter Remote Server IP Address:
192.168.170.128
Enter Remote Server Username:
kali
Enter Remote Server Password for :


Connecting to Remote Server....
[*] Uptime:  05:55:32 up 13:42,  3 users,  load average: 0.00, 0.01, 0.01
[*] IP Address: 192.168.170.128
[*] Country: Address not found

[#] Whoising victim's address: academy.cyberiumarena.com
[#] Whois data was saved into: /home/kali/Documents/nipe/Whois_academy.cyberiumarena.com
[#] Scanning victim's address: academy.cyberiumarena.com
[#] Nmap scan was saved into: /home/kali/Documents/nipe/Nmap_academy.cyberiumarena.com
[?] Do you want to scan another Domain/IP address? [Y/N]: Y
[?] Please specify a Domain/IP address to scan: 104.22.55.121
```

```
[#] Whois data was saved into: /home/kali/Documents/nipe/Whois_104.22.55.121
[#] Scanning victim's address: 104.22.55.121
[#] Nmap scan was saved into: /home/kali/Documents/nipe/Nmap_104.22.55.121
[?] Do you want to scan another Domain/IP address? [Y/N]: n
[#] Scanned Whois and Nmap timestamp located in /var/log/nr.log
[?] Do you like to open the /var/log/nr.log file? [Y/N]: y
[#] Wed Jan 18 05:51:32 AM EST 2023- [*] Whois data collected for: academy.cyberiumarena.com
[#] Wed Jan 18 05:54:12 AM EST 2023- [*] Nmap data collected for: academy.cyberiumarena.com
[#] Wed Jan 18 05:55:33 AM EST 2023- [*] Whois data collected for: academy.cyberiumarena.com
[#] Wed Jan 18 05:58:30 AM EST 2023- [*] Nmap data collected for: academy.cyberiumarena.com
[#] Wed Jan 18 06:00:17 AM EST 2023- [*] Whois data collected for: 104.22.55.121
[#] Wed Jan 18 06:14:17 AM EST 2023- [*] Nmap data collected for: 104.22.55.121
[*] You have exited the script.
[*] Have an AWESOME day!
```

Bash Script Screenshot:

remotecontrol1.sh

```bash
#!/bin/bash

#Define an exit message to be displayed
ExitMessage1=$(echo "[*] You have exited the script.")
ExitMessage2=$(echo "[*] Have an AWESOME day!")

#Introduction message of the script
echo -e "[*] You are about to run the Remote Control for Project Research \n"
echo -e "[*] The script is created by Mary Ann Lim Tian \n"
echo -n "[?] If you wish to continue, press [Y] or [N] to exit: "
read answer
if [[ $answer == y || $answer == Y ]]
then
#[1] This part is to install required tools to remotely access server anonymously
    # to check if the tools required have been installed or not
    GeoIPBinCheck=$(dpkg-query -l |grep geoip-bin|awk '{print ($2)}'|wc -c) #to get the keyword 'geoip-bin'
    SSHPassheck=$(dpkg-query -l |grep sshpass|awk '{print ($2)}'|wc -c) #to get the keyword 'sshpass'
    NipeFolderCheck=$(find . -type d -name nipe  | awk -F / '{print $(NF-0)}') #to get the keyword 'nipe'
    TorFolderCheck=$(find . -type d -name ToriFY  | awk -F / '{print $(NF-0)}') #to get the keyword 'ToriFY'

    #[A] GEOIP-BIN - if condition to check if geoip-bin tool has been installed or not in the local server
    #A tool to look for country of any ip address or hostname orginates from
        if [[ $GeoIPBinCheck != 10 ]] #[[ ]] evaluates if either true or false
        then #to install geoip-bin
            sudo apt-get -y install geoip-bin #install flag -y is used to answer prompt question to yes or force to yes
            message1="[#] geoip-bin is installed" #message when installation is completed
        else
            message1="[#] geoip-bin is already installed" #message when tool already installed
        fi
    #[B] SSHPASS - if condition to check if sshpass tool has been installed or not in the local server
    #A tool for password-based or password-less authentication to log into the remote server using SSH
        if [[ $SSHPassheck != 8 ]]
        then #to install sshpass
            sudo apt-get -y install sshpass #install flag -y is used to answer prompt question to yes or force to yes
            message2="[#] sshpass is installed" #message when installation is completed
        else
            message2="[#] sshpass is already installed" ##message when tool already installed
        fi
```

```
    #[C] TORIFY -  if condition to check if ToriFY tool has been installed or not in the local server
    #A tool that allows you to make TOR your default gateway and send all internet connections under TOR (as transparent proxy)
    # to increase privacy/anonymity without extra unnecessary code
        if [[ $TorFolderCheck != 'ToriFY' ]]
        then #to install ToriFY
            git clone https://github.com/Debajyoti0-0/ToriFY.git
            message4="[#] Tor is installed" #message when installation is completed
        else
            message4="[#] Tor is already installed" #message when tool already installed
        fi

    #[D] NIPE - if condition to check if nipe tool has been installed or not in the local server
    #A tool that makes Tor network our default gateway to surf the network with anonymity
        if [[ $NipeFolderCheck != 'nipe' ]]
        then #to install nipe
            git clone https://github.com/htrgouvea/nipe && cd nipe
            sudo cpan -y install Try::Tiny Config::Simple JSON
            sudo perl nipe.pl -y install
            message3="[#] Nipe is installed" #message when installation is completed
        else
            message3="[#] Nipe is already installed" #message when tool already installed
        fi

    #To display status of installation
        echo "$message1"
        echo "$message2"
        echo "$message3"
        echo -e "$message4 \n"


#[2] This part is to run nipe.pl for masking the public IP address. To activate and check the status nipe.pl
    nipeDir=$(find /home -type d -name nipe)
    cd $nipeDir                                             #change directory to nipe and run nipe.pl in this directory
    echo -n "[*] You are curently in this directory:"       #to show the directory path is in nipe
    pwd
    #$(sudo perl nipe.pl restart)                           #to start nipe service
    #$(sudo perl nipe.pl status |grep -o activated)         #to check and get nipe service status

    until [[ $(sudo perl nipe.pl status |grep -o activated) = 'activated' ]]
    do
        $(sudo perl nipe.pl restart) #to start nipe service
    done
    SpoofIp=$(sudo perl nipe.pl status|grep -Eo '([0-9]{1,3}[\.]){3}[0-9]{1,3}') #to get the nipe service spoofed ip address
    echo -e "[*] You are anonymous...connecting to the remote server"
#[3] This part is to lookup the for the spoof country and IP use to mask public IP address

    SpoofCountry=$(geoiplookup $SpoofIp |awk '{print ($5,$6)}')                         #to get look up the provide spoof country of the IP address provided
    echo -e "[*] Your Spoofed IP address is: $SpoofIp, Spoofed Country: $SpoofCountry \n" #to show the con IP address and con country IP location

    #to get the victim IP address / domain to scan
    echo "[?]Please specify a Domain/IP address to scan: "
    read VictimAddress
    echo -n "[?] Do you want to continue to scan address of $VictimAddress via Remote Server? [Y/N]: "
    read answer
    echo -e "\n"
    if [[ $answer == y || $answer == Y ]]
    then
        #to check if scanned audit log exist or not
          if [ -f /var/log/nr.log ]                     #if nr.log (scanned audit log) exist
          then
            currentuser=$(whoami)                       #to get current user
            sudo chown $currentuser /var/log/nr.log  #to provide current user who is not a root user to write in /var/log/nr.log

          else
            currentuser=$(whoami)                       #to get current user
            #create NR Log to audit the Whois and Namp of Victim's Address
            sudo touch /var/log/nr.log                  #to create custom log file in /var/log
            sudo chown $currentuser /var/log/nr.log  #to permit currentuser to write in the custom audit log file <nr.log>
        fi
    else
        echo "$ExitMessage1"
        echo "$ExitMessage2"
        exit
```

```
    fi

#Function 1: This is a function of Remote Server to access by Local device
function REMOTESERVERLOGIN()
{
            echo "[***] Please provide the following remote server details to use to scan address [***]"
            echo "[?] Enter Remote Server IP Address: "
            read RemoteIP
            echo "[?] Enter Remote Server Username: "
            read RemoteUsername
            echo "[?] Enter Remote Server Password for $RemoteUserName: "
            stty -echo #to hide the inputed data of the password
            read RemotePassword
            stty echo
            echo -e "\n"
            echo "Connecting to Remote Server...."
            #to execute below command login to remote server using sshpass for passowrd and ssh to connect to host and IP address
            RemoteStatus=$(sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP echo ok 2>&1)
            #-o stricthostkeychecking=no to bypass verification step when performing ssh
            if [[ $RemoteStatus == ok ]]
            then
            #to get uptime of remote server.
            RemoteServerUptime=$(sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP uptime)
            #to check and get IP Address of remote server.
            RemoteServerIP=$(sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP ifconfig |grep inet |head -1|awk '{print ($2)}')
            #to check and get country of remote server.
            RemoteServerCountry=$(sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP geoiplookup 192.168.170.130 | awk '{print ($5,$6,$7)}')
            echo "[*] Uptime: $RemoteServerUptime"
            echo "[*] IP Address: $RemoteServerIP"
            echo -e "[*] Country: $RemoteServerCountry \n"
            else
            echo -n "[?] Unable to connect. Try another remote server? [Y/N]"
            read answer
                if [[ $answer == Y || $answer == y ]]
                then
                    REMOTESERVERLOGIN #to re-run function 1 if unable to connect to remote server
                else
```

```
                    exit
                fi


            fi
}
REMOTESERVERLOGIN

#Function 2: This is a function to execute scanning of Victim's Domain / IP address and to audit into nr.log
function VICTIMSCAN()
        {
        # Get the remote server to check the Whois of the given address
        LocalPath=$(pwd) #to get the current working directory
        WhoisPath=$(echo "$LocalPath/Whois_$VictimAddress") #this is the Whois directory path into a variable
        NmapPath=$(echo "$LocalPath/Nmap_$VictimAddress") #this is the Nmap directory path into a variable

        # Get the remote server to check the Whois of the given address
        echo "[#] Whoising victim's address: $VictimAddress"
        sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP whois $VictimAddress > Whois_$VictimAddress
        nrLogWhois=$(echo "`date`- [*] Whois data collected for: $VictimAddress")
        echo "[#] Whois data was saved into: $WhoisPath"

        # Get the remote server to check the Nmap of the given address
        echo "[#] Scanning victim's address: $VictimAddress"
        sshpass -p $RemotePassword ssh -o stricthostkeychecking=no $RemoteUsername@$RemoteIP nmap $VictimAddress -sV -F > Nmap_$VictimAddress
        nrLogNmap=$(echo "`date`- [*] Nmap data collected for: $VictimAddress")
        echo "[#] Nmap scan was saved into: $NmapPath"
        echo "[#] $nrLogWhois" >> /var/log/nr.log
        echo "[#] $nrLogNmap" >> /var/log/nr.log

        #to check if user want to scan more domain or ip address
        echo -n "[?] Do you want to scan another Domain/IP address? [Y/N]: "
        read answer
            if [[ $answer == Y || $answer == y ]]
            then
            #user to enter another
                echo -n "[?] Please specify a Domain/IP address to scan: "
                read VictimAddress
```

```
188                 VICTIMSCAN
189             else
190             #option to exit the bash script and check scan logs in nr.log file
191                 echo "[#] Scanned Whois and Nmap timestamp located in /var/log/nr.log"
192                 echo -n "[?] Do you like to open the /var/log/nr.log file? [Y/N]: "
193                 read answer
194                     if [[ $answer == Y || $answer == y ]]
195                     then
196                         cat /var/log/nr.log
197                         echo "$ExitMessage1"
198                         echo "$ExitMessage2"
199                     else
200                         echo "$ExitMessage1"
201                         echo "$ExitMessage2"
202                         exit
203                     fi
204
205
206             fi
207 }
208 VICTIMSCAN
209 else
210     echo "$ExitMessage1"
211     echo "$ExitMessage2"
212     exit
213
214 fi
```