

Network Research
Project
Leonard Yeo



```
#!/bin/bash
```

```
useColor="true"
```

Font Color

BLUE=' \033[0;34m'

CYAN=' \033[0;36m'

```
GREEN='\033[0;32m'
```

```
RED='\033[0;31m'
```

ORANGE=' \033[1;33m '

UNDERLINE=' \033[0;4m'

NOCOLOR=' \033[0m'

DARKRED='\033[0;31m'

DARKGREY=' \033[1;90m'

MAGENTA='\033[1;95m'

```
WHITE=' \033[97m'
```

DARKGREEN=' \033[32m'

clear

```
echo -e "\t\t"
```

```
echo -e "\t\t💀\t\t${ORANGE}Network Research Project ${CYAN}Version 1${RED}\t\t\t💀"
```

```
echo -e "\t\t💀\t\t ${MAGENTA}Done by Leonard Yeo (S8)\t\t\t💀"
```

```
echo -e "\t\t"
```

```
echo -e "\n"
```

```
echo -e "\n"
```

Specification of the Font Color code matrix to be used in this script

Short introduction of project name and student name with a fancy frame

```
echo -e "${DARKGREEN}Checking if required tools are installed...."

{
    if [ $(dpkg-query -l | grep '^ii' | awk '{print $2}' | grep -x tor) == "tor" ];
    then

        echo -e "${WHITE}[#] tor is already installed "

    else

        echo -e " ${RED}Installing ToriFY..."

        git clone https://github.com/Debajyoti0-0/ToriFY.git
        sudo pip3 install -r requirements.txt
        chmod +x *
        sudo ./install.sh
    fi
}
```



This checks if the package for "tor" is already installed. If it is not installed, it proceeds to clone a GitHub repository, install the required packages using pip, and runs an installation script.


```
{
    if [ $(dpkg-query -l | grep '^ii' | awk '{print $2}' | grep -x geoip-bin) == "geoip-bin" ];
    then

        echo -e "${WHITE}[#] geoip-bin is already installed"

    else

        echo -e "${RED}Installing geoip-bin..."
        sudo apt-get install geoip-bin

    fi
}
```



This checks if the package for "geoip-bin" is already installed. If it is not installed, it proceeds to clone a GitHub repository, install the required packages using sudo apt-get install, and runs an installation script.


```
{
if [ $(dpkg-query -l | grep '^ii' | awk '{print $2}' | grep -x sshpass) == "sshpass" ];
then

echo -e "${WHITE}[#] sshpass is already installed"

else

echo -e "${RED}Installing sshpass..."
sudo apt-get install sshpass

fi
},
```



This checks if the package for "sshpass" is already installed. If it is not installed, it proceeds to install the required packages using `sudo apt-get install`, and runs an installation script.


```
{
if [ $(dpkg-query -l | grep '^ii' | awk '{print $2}' | grep -x nmap) == "nmap" ];
then

echo -e "${WHITE}[#] nmap is already installed"

else

echo -e "${RED}Installing nmap..."
sudo apt-get install nmap

fi
},
```



This checks if the package for "nmap" is already installed. If it is not installed, it proceeds to install the required packages using `sudo apt-get install`, and runs an installation script.

```

}
if [ $(find ./ -name "nipe.pl") == "./nipe/nipe.pl" ];
then

echo -e "${WHITE}[#] Nipe is already installed"

else

git clone https://github.com/htrgouvea/nipe
cd "$(dirname "$(find ./ -name "nipe.pl")")" && sudo cpan install Try::Tiny Config::Simple JSON
cd "$(dirname "$(find ./ -name "nipe.pl")")" && sudo perl nipe.pl install

fi
}

```

This script checks if the file "nipe.pl" is already present in the current directory and its subdirectories using the find command. If it is found, the script prints a message saying "Nipe is already installed". If it is not found, the script proceeds to clone a GitHub repository, navigate to the directory where the file is located and install the required modules and run the installation of NIPE.

```

MYIP=$(curl -s ifconfig.me/ip)

cd "$(dirname "$(find ./ -name "nipe.pl")")" && sudo perl nipe.pl start && sudo perl nipe.pl restart
NIP=$(cd "$(dirname "$(find ./ -name "nipe.pl")")" && sudo perl nipe.pl status | grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}')

```

This retrieves the current public IP address of the system using the curl command in silent mode using the -s flag so that it won't display the progress and a public IP lookup service, and assigns it to the variable MYIP. Then the script navigates to the directory where the file "nipe.pl" is located using the find command, and runs two perl commands using the nipe.pl script, 'start' and 'restart'. The next line is checking the status of nipe.pl script by running the command 'status' and using the grep command to extract IP address from the output, which is assigned to variable NIP.


```
{
if [ "$MYIP" = "$NIP" ];

then

echo -e "${RED} NOT ANOYMOUS! EXITING NOW."

exit 0

else

echo -e "${CYAN}your Spoofed IP Address is '${WHITE}$NIP', ""${CYAN}Country of Spoofed IP is: " ${WHITE}$(geoiplookup $NIP | grep -oP '(?<=: ).*')

fi
}
```

This compares the variable MYIP, which contains the current public IP address of the system, with the variable NIP, which contains the IP address of the system as reported by the Nipe software after routing the traffic through the Tor network.

If the two variables are equal, it means that the IP address is not being spoofed, and the script echos a message "NOT ANONYMOUS! EXITING NOW." and exits the script with a exit code 0.

If the two variables are not equal, it means that the IP address has been successfully spoofed.

```

echo "Specify a Domain/IP address to scan:"
read input1
cd - > /dev/null 2>&1

sshpas -p 'tc' ssh tc@192.168.220.131 'echo "${CYAN}Connecting to Remote Server:"'
echo -e "${CYAN}Uptime: ${WHITE}$(uptime)"
echo -e "${CYAN}IP Address:${WHITE}$(curl -s ifconfig.me/ip)"
RIPADD=$(curl -s ifconfig.me/ip)
echo -e "${CYAN}Country: ${WHITE}$(geoiplookup $RIPADD| grep -oP '(?<=: ).*')"
```



```

echo -e "${CYAN}[@] Whoising Victim's Address:"
echo -e "${MAGENTA}Whois data was saved into $(pwd)/whois_${input1}.txt"
whois $input1 >> whois_${input1}.txt
echo "${TZ="Singapore" date) - [*] whois data is collected for $input1 " >> NR.log
```



```

echo -e "${CYAN}[@] Scanning Victim's Address:"
echo -e "${MAGENTA}Nmap scan was saved into $(pwd)/nmap_${input1}.txt"
nmap $input1 >> nmap_${input1}.txt
echo "${TZ="Singapore" date) - [*] nmap data is collected for $input1 " >> NR.log
```



```

echo -e "${GREEN}whois & namp log time added to NR.log"
echo -e "${RED} Script End "
```

This prompts the user to input a domain or IP address and assigns the input to the variable input1.

Then it uses sshpass to connect to a remote server with IP address 192.168.220.131 and a password of 'tc' and runs several commands on the remote server:

- Prints a message "Connecting to Remote Server"

- Prints the uptime of the remote server

- Prints the public IP address of the remote server

- Prints the country of the remote server based on its IP address

- Runs the 'whois' command on the input1 domain/IP address and saves the output to a file named "whois_\${input1}.txt" in the current working directory.

- Runs the 'nmap' command on the input1 domain/IP address and saves the output to a file named "nmap_\${input1}.txt" in the current working directory.

- It also logs the date and time of when the whois and nmap commands were run and saved to a file named "NR.log" in the current working directory.

Script Execution

```
(kali)kali: ~
❯❯❯
Network Research Project Version 1
Done by Leonard Yeo (S8)

Checking if required tools are installed....
[#] tor is already installed
[#] geoip-bin is already installed
[#] sshpass is already installed
[#] whois is already installed
[#] nmap is already installed
[#] Nipe is already installed
your Spoofed IP Address is ' 185.220.102.253', Country of Spoofed IP is: DE, Germany
Specify a Domain/IP address to scan:
scanme.nmap.com
Connecting to Remote Server:
Uptime:      12:21:54 up 6:45, 1 user, load average: 0.28, 0.23, 0.22
IP Address:   192.42.116.191
Country:      NL, Netherlands
[@] Whoising Victim's Address:
Whois data was saved into /home/kali/NRproject/whois_scanme.nmap.com.txt
[@] Scanning Victim's Address:
Nmap scan was saved into /home/kali/NRproject/nmap_scanme.nmap.com.txt
whois & nmap log time added to NR.log
Script End
```


Script Execution (Saved Files)

```
(kali@kali)-[~/NRproject]
$ ls
F.sh  nipe  nmap.log  nmap_scanme.nmap.com.txt  NR.log  NR.sh  scripts  test2.sh  test.sh  whois_scanme.nmap.com.txt
```

Saved Namp result

Saved whois result

Saved Log file

Script Execution (log file)

```
(kali㉿kali)-[~/NRproject]
$ cat NR.log
Thu Jan 19 11:15:26 PM +08 2023 - [*] whois data is collected for 192.168.220.131
Thu Jan 19 11:15:26 PM +08 2023 - [*] nmap data is collected for 192.168.220.131
Thu Jan 19 11:43:05 PM +08 2023 - [*] whois data is collected for 192.168.220.130
Thu Jan 19 11:43:05 PM +08 2023 - [*] nmap data is collected for 192.168.220.130
Thu Jan 19 11:48:56 PM +08 2023 - [*] whois data is collected for 192.168.220.130
Thu Jan 19 11:48:56 PM +08 2023 - [*] nmap data is collected for 192.168.220.130
Fri Jan 20 12:42:18 AM +08 2023 - [*] whois data is collected for 192.168.220.131
Fri Jan 20 12:42:19 AM +08 2023 - [*] nmap data is collected for 192.168.220.131
Fri Jan 20 12:44:31 AM +08 2023 - [*] whois data is collected for 192.168.220.130
Fri Jan 20 12:44:31 AM +08 2023 - [*] nmap data is collected for 192.168.220.130
Fri Jan 20 01:10:08 AM +08 2023 - [*] whois data is collected for 192.168.220.131
Fri Jan 20 01:10:08 AM +08 2023 - [*] nmap data is collected for 192.168.220.131
Fri Jan 20 01:20:44 AM +08 2023 - [*] whois data is collected for scanme.nmap.com
Fri Jan 20 01:20:46 AM +08 2023 - [*] nmap data is collected for scanme.nmap.com
Fri Jan 20 01:21:57 AM +08 2023 - [*] whois data is collected for scanme.nmap.com
Fri Jan 20 01:22:11 AM +08 2023 - [*] nmap data is collected for scanme.nmap.com
```

End

