

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

Creating a IF Statement to check if applications need to be installed

```
No symbols found 1  #!/bin/bash
2
3  #! 1. Installations and Anonymity Check
4
5  #! 1.2 If the applications are already installed, don't install them again
6
7  app1="geoipllookup"
8  app2="tor"
9  app3="sshpas"
10
11 if [[ $(which $app1) != "" ]];
12 then
13     echo "$app1 is installed"
14
15 else
16     echo "$app1 is not installed"
17     echo "Installing $app1..."
18     sudo apt-get install geoipl-bin
19     echo "Installation for $app1 completed"
20
21
```

In the first scenario, we will be using a IF statement to check whether a condition is true or false.

In our case, it will be whether a application is install (true) and not installed (false)

Save application names in variables so that we will be able to use it for our IF statement

Which command to output whether command exist. It not installed, it will output nothing. Hence != " " will be true if application is installed and false is not installed. Which will trigger the installation process

command to install the required application

Status	Message
(kali@kali) - [~/Desktop]	\$ bash NR S13.sh
Compiler	geoipllookup is installed tor is installed sshpas is installed
Messages	[sudo] password for kali:
Scribble	Nipe is installed

Message will show whether application is installed or if installation is in progress

Network Research Project – S13 Low Hong Jun

Same for NIPE, IF statement to check if NIPE file is in the system using the find command

```
50 No symbols found
51 if [[ $(sudo find / -type f -name niipe.pl 2>/dev/null) != '' ]];
52
53 then
54     echo "Niipe is installed"
55
56 else
57     echo "Niipe is not installed"
58     echo "Installing Niipe..."
59
60     git clone https://github.com/htrigouvea/niipe && cd niipe
61     sudo apt-get install cpanminus
62     sudo cpanm --installdeps .
63     sudo perl niipe.pl install
64
65     echo "Installation for Niipe completed"
66
67 fi
68
69 #! 1.3 Check if the network connection is anonymous; if not, alert the user and exit
70
```

For niipe we are doing it differently. We find the niipe.pl file to determine if niipe is installed instead.

2>/dev/null in this case send error message to the /dev/null folder instead of showing up in our output

Installation command for niipe

Status: (kali@kali) - [~/Desktop]
\$ bash NR_S13.sh

Compiler: geiopllookup is installed
tor is installed
sshpas is installed

Messages: [sudo] password for kali:
Niipe is installed

Scribble: sudo authentication required to view file
Niipe is installed message will show if the file we use find command exist

IF statement to check if TOR service is enabled

```
69 #! 1.3 Check if the network connection is anonymous; if not, alert the user and exit
70 echo -e
71
72 if [[ $(sudo netstat -tpan | grep tor) == "" ]];
73
74 then
75     echo "Your connection is not anonymous, aborting mission! Please enable your tor service!"
76     exit
77
78 #! 1.4 If the network connection is anonymous, display the spoofed country name
79
80 else
81     echo "Your connection is anonymous, establishing connection with remote server..."
82
83     spoofedip=$(sudo netstat -tpan | grep -E '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | awk '{print $(NF-2)}' | grep -E '(9001|9030|9040|9050|9051|9150)' | awk -F: '{NR==1 {print $1}}')
84     spoofedcountry=$(geoipllookup $spoofedip | awk -F: '{print $2}')
85
86     echo "Your Spoofed IP Address is : $spoofedip"
87     echo "Your Spoofed Country is : $spoofedcountry"
88
89 fi
90
91 #! 1.5 Allow the user to specify the address to scan via remote server; save into a variable\
92
93 echo -e
```

Checking if connection is anonymous by checking if we are using tor service

If the output grep tor == "" return empty, it means that no tor service was detected in the netstat command which check our network connections

Using If statement to see whether a condition is true/false.

In this case, if grep tor in netstat returns empty, it means that there are no tor network detected. So our connection are not anonymous. The script will then exit.

If output returns with results, it will check for the spoofed ip and country

Variables

geoipllookup using results from spoofedip

Filter for ip address, tor ports in the netstat command check. Grep -E indicates we are searching for a regular pattern.
TOR ports, 9001,9030,9040,9050,9051,9150

Status: (kali@kali) - [~/Desktop]
\$ bash NR_S13.sh

Compiler: geiopllookup is installed
tor is installed
sshpas is installed

Messages: [sudo] password for kali:
Niipe is installed

Scribble: Your connection is anonymous, establishing connection with remote server...

Terminal: Your Spoofed IP Address is : 116.203.205.96
Your Spoofed Country is : DE, Germany

\$spoofedip = 116.203.205.96
\$spoofedcountry = DE, Germany

Network Research Project – S13 Low Hong Jun

Collecting User Input

```
90
91 #! 1.5 Allow the user to specify the address to scan via remote server; save into a variable\
92
93 echo -e
94 echo "Please enter the Domain/IP to be scanned :"
95 read IP
96
97
98 #! 2. Automatically Connect and Execute Commands on the Remote Server via SSH
99
100 #! 2.1 Display the details of the remote server (country, IP, and Uptime)
101
102 echo -e
103
104 UPT=$(uptime)
```

Command to save user input.
Read input and save to the variable \$IP

Status (kali@kali) - [~/Desktop]
\$ bash NR_S13.sh

Compiler geotracklookup is installed
tor is installed

Messages sshpass is installed
[sudo] password for kali:

Scribble Nipe is installed

Terminal Your connection is anonymous, establishing connection with remote server...
Your Spoofed IP Address is : 116.203.205.96
Your Spoofed Country is : DE, Germany

Please enter the Domain/IP to be scanned :
192.168.233.130

192.168.233.130 will be save as the variable \$IP

Accessing remote server and automating tasks

```
97
98 #! 2. Automatically Connect and Execute Commands on the Remote Server via SSH
99
100 #! 2.1 Display the details of the remote server (country, IP, and Uptime)
101
102 echo -e
103
104 UPT=$(sshpass -p tc ssh tc@$IP uptime)
105 IPR=$(sshpass -p tc ssh tc@$IP hostname -I)
106 Country=$(whois $IP | grep -i country | awk '{print $2}' | tr -s " ")
107 RC=$(sshpass -p tc ssh tc@$IP echo $Country)
108
109 echo "Logging in to remote server..."
110 echo -e
111 echo "Uptime : $UPT"
112 echo "This is the remote server IP Address : $IPR"
113 echo "This is the remote server country : $RC"
114
115 #! 3. Results
116 #! 3.1 Save the Whois and Nmap data into files on the local computer
```

Using whois command and filter for the remote server country.
As the command is quite long and we still require it in another automated command, we save it in a variable.

We using sshpass so that the ssh process can be automated.
-p flag indicates the password
After we input the whole command :
sshpass -p 'password' ssh user@IP
We can input the command we can it to do at the remote server
Example uptime, hostname

Variables

Output of script written

Status Your Spoofed Country is : DE, Germany

Compiler Please enter the Domain/IP to be scanned :
192.168.233.130

Messages Logging in to remote server...

Scribble Uptime : 04:34:06 up 3:41, 1 user, load average: 0.00, 0.00, 0.00
This is the remote server IP Address : 192.168.233.130
This is the remote server country : US

Network Research Project – S13 Low Hong Jun

Collecting data from the remote server and saving it into files and logs automatically

```
No symbols found 115 #! 2.2 Get the remote server to check the Whois of the given address
116 #! 3. Results
117 #! 3.1 Save the Whois and Nmap data into files on the local computer
118
119 echo -e
120
121 date=$(date "+%c %Y")
122
123 echo "Collecting information on victim's server using Whois..."
124
125 #! 3.1 Save the Whois and Nmap data into files on the local computer
126 mkdir -p /home/kali/Desktop/NR/whois
127 whois $IP > /home/kali/Desktop/NR/whois/whois_$IP
128
129 echo "Creating directories..."
130 echo "Information collected and stored at /home/kali/Desktop/NR/whois as whois_$IP"
131
132 #! 3.2 Create a log and audit your data collecting
133 echo "$date whois data collected for : $IP" >> nr_nmap_whois.log
134
135 #! 2.3 Get the remote server to scan for open ports on the given address
136
137 echo -e
138
139 echo "Scanning all open ports for victim's server using nmap..."
140
141 #! 3.1 Save the Whois and Nmap data into files on the local computer
142 mkdir -p /home/kali/Desktop/NR/nmap
143 nmap $IP > /home/kali/Desktop/NR/nmap/nmap_$IP
144
145 echo "Creating directories..."
146 echo "Open ports scanned and saved into /home/kali/Desktop/NR/nmap as nmap_$IP"
147
```

Annotations:

- date command. %c to display day, time, zone. %Y to display year. Goes into a variable 'date' for use in the log file later
- Create a directory for the whois output to create files into it
- whois command to output into a file into the full file path as shown
- whois_\$IP
- Creating log with timestamp of when the nmap and whois is completed
- \$IP allows the file name to follow the IP input by user
- Create a directory for the nmap output to create files into it
- nmap_\$IP

Output from the above script

```
Collecting information on victim's server using Whois...
Creating directories...
Information collected and stored at /home/kali/Desktop/NR/whois as whois_192.168.233.130

Scanning all open ports for victim's server using nmap...
Creating directories...
Open ports scanned and saved into /home/kali/Desktop/NR/nmap as nmap_192.168.233.130

(kali@kali) - [~/Desktop]
$

(kali@kali) - [~/Desktop]
$ cat nr_nmap_whois.log
Thu 12 Oct 2023 10:01:19 AM EDT 2023 whois data collected for : 192.168.233.130
Thu 12 Oct 2023 10:01:19 AM EDT 2023 Nmap data collected for : 192.168.233.130
Thu 12 Oct 2023 10:01:35 AM EDT 2023 whois data collected for : 192.168.233.129
Thu 12 Oct 2023 10:01:35 AM EDT 2023 Nmap data collected for : 192.168.233.129
Thu 12 Oct 2023 10:07:27 AM EDT 2023 whois data collected for : 192.168.233.130
Thu 12 Oct 2023 10:07:27 AM EDT 2023 Nmap data collected for : 192.168.233.130

(kali@kali) - [~/Desktop]
$
```

Annotations:

- file path to save the files
- whois_\$IP
- Output from script
- log that records whois and nmap data collection
- Timestamp of data collection recorded