Remote Control

Objective: To perform reconnaissance activity through a compromised machine to prevent detection

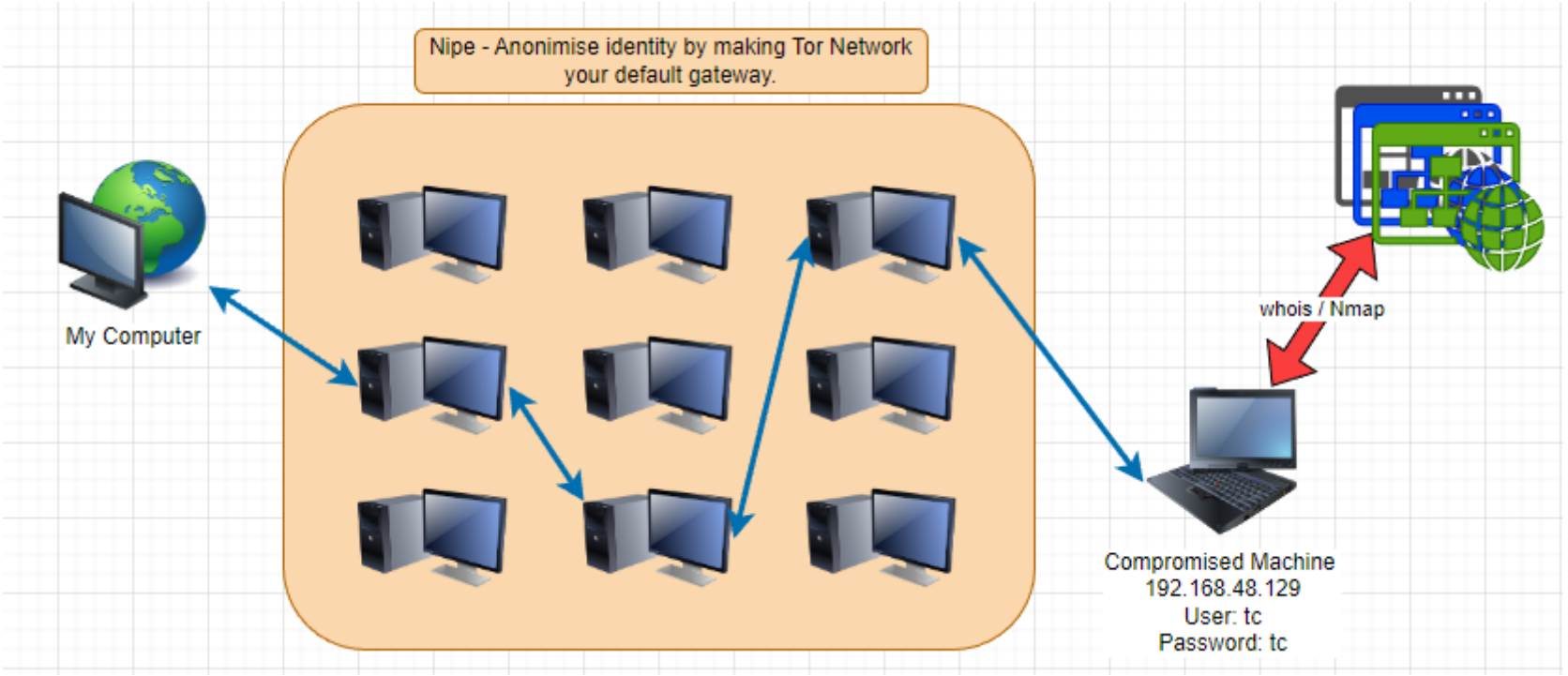

Figure A -Script operation overview

In Figure B, User can choose to use ‘source ~/Documents/script/remotectrlv1.sh’ to run the script from any current working directory point to the path where the shell script is being stored. Alternatively, the user can also choose to go to the working directory of the shell script to execute ‘bash remotectrlv1.sh’

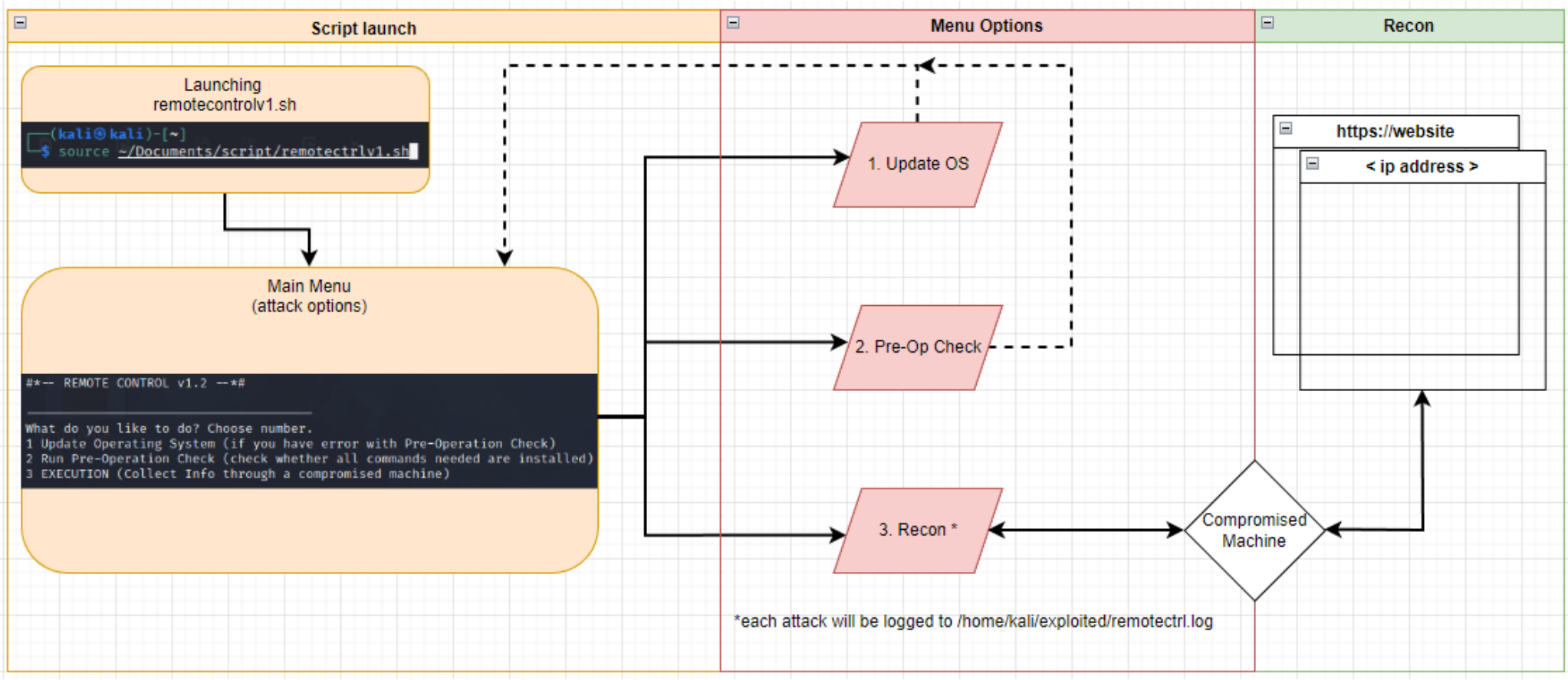

Figure B -Process flow of Script

Updating Operating System

In Figure 1, User can choose the option to update and upgrade the Operating System (OS). User can choose this if one finds error when running Pre-Operation check, which is usually caused by outdated OS. When using 'sudo' command, it is usually an interactive shell command. Since we are using a script, we will want it to be as non-interactive as possible. We can turn it into non-interactive by using '-S' flag. In line 23, we feed the password, in our case is 'kali', by piping echo <password> before 'sudo' command. By using '2>/dev/null', we can discard the password prompt message to achieve a clean terminal output.

Once the system has pulled the necessary packages, it will prompt user to input 'y' to upgrade (line 30) or 'n' to not upgrade and return to main menu (line 38). If it receives any other input, it will notify user and prompt user again (line 42).

```
19  #This Option allows user to check for update and upgrade of Operating System
20  #sudo -S receives password input by echo-ing password'kali' before sudo command
21      1)
22      echo "Checking and updating Linux System.."
23      echo kali | sudo -S apt-get update 2>/dev/null
24
25      sudo apt-get -u upgrade --assume-no >/dev/null 2>&1
26      function UPDATER() {
27          echo "Ready to upgrade? It will take a while. (y/n)"
28          read answer
29          case $answer in
30              y|Y)
31                  echo
32                  echo "---UPGRADING PACKAGES---"
33                  echo
34                  sudo apt-get upgrade -y
35                  echo "All done! Going back to Main Menu.."
36                  REMOTE
37                  ;;
38              n|N)
39                  echo "OK. Going back to Main Menu.."
40                  REMOTE
41                  ;;
42              *)
43                  echo
44                  echo "Wrong Input. Please answer 'y or 'n'."
45                  UPDATER
46                  ;;
47          esac
48
49      REMOTE
50      }
51      UPDATER
52      ;;
```

Figure 1 Shell Script to update Operating System (OS)

```
What do you like to do? Choose number.
1 Update Operating System (if you have error with Pre-Operation Check)
2 Run Pre-Operation Check (check whether all commands needed are installed)
3 Collect Information (using Victim's device to prevent detection)

1
Checking and updating Linux System..
Hit:1 http://deb.i2p2.no unstable InRelease
Hit:2 http://mirrors.jevincanders.net/kali kali-rolling InRelease
Reading package lists... Done
Ready to upgrade? It will take a while. (y/n)
```

Figure 2 Terminal Output of Option 1: Update Operating System

Pre-Operation Check

Pre-Operation Check is offered as Option 2 in the Main Menu. This makes sure that the host machine has the required tools to be able to perform the actual task which is offered as Option 3.

In Figure 3, line 64, 'command -v' is used to check if 'sshpass' command is installed in the machine, if not it will execute line 70 to install the package. Here, we used '>/dev/null/' to discard any message that will be shown on terminal. In Figure 4, the same script is used with slight alteration in line 80 and line 86 to check for 'whois' command.

```
61      # sshpass is a non-interactive ssh password authentication tool to remote access the compromised machine to do recon activities
62      function installsshpass()
63      {
64      if command -v sshpass >/dev/null
65      then
66          echo '[+] sshpass is installed'
67          return
68      else
69          echo '[-] sshpass NOT installed, installing...'
70          echo kali | sudo -S apt-get install sshpass -y 2>/dev/null
71
72      fi
73      installsshpass
74      }
75      installsshpass
```

Figure 3 Function to check and install sshpass to provide non-interactive ssh login

```
77      # Whois is used find out information about a website's record, like ip address, site's owner and site's origin etc
78      function installwhois()
79      {
80      if command -v whois >/dev/null
81      then
82          echo '[+] whois is installed'
83          return
84      else
85          echo '[-] whois NOT installed, installing...'
86          echo kali | sudo -S apt-get install whois -y 2>/dev/null
87      fi
88      installwhois
89      }
90      installwhois
```

Figure 4 Function to check and install whois

In Figure 5, the command used to check for nipe differs from the 'sshpass' and 'whois' as it is not an official package in linux and was created by Heitor Gouvêa and published in github. Nipe will be installed in the current working directory in the terminal where User execute the command. In Figure 5 line 96 and line 103, nipe was chosen to be installed in /home/kali for ease of access to the nipe working directory. Line 104 to 106 is the terminal commands used to install nipe.

```
92          # Nipe is an engine that makes the Tor network our default network gateway.
93          function installnipe()
94          {
95          # 'test -d' tests if this application exist in the directory as it is not installed in /usr/bin like other common applications
96          if test -d /home/kali/nipe
97          then
98              echo '[+] Nipe is installed'
99              cd /home/kali/nipe
100             return
101         else
102             echo '[-] Nipe not found. Installing nipe..'
103             cd ~
104             git clone https://github.com/htrgouvea/nipe && cd nipe
105             echo kali | sudo -S cpan install Try::Tiny Config::Simple JSON 2>/dev/null
106             echo kali | sudo -S perl nipe.pl install 2>/dev/null
107             cd /home/kali/nipe
108             return
109         fi
110         installnipe
111         }
112         installnipe
```

Figure 5 Function to check and install nipe engine

In Figure 6, line 117 checks if the directory to stores the log and information is present, if not it will proceed to create the required directory and sub-folders to prevent any storage errors when executing 'sshpass' later on to store the information that the User have executed.

```
114         # LOGDIR checks if the directory that will store our log and collected info exists, if not it will create it.
115         function LOGDIR()
116         {
117         if test -d ~/exploited && test -d ~/exploited/whois && test -d ~/exploited/nmap
118             then
119                 echo "[+] Directory '~/exploited' ready for storing log"
120                 return
121             else
122                 echo -e "\n>> no log directory. creating in process.. <<"
123                 mkdir ~/exploited && mkdir ~/exploited/whois && mkdir ~/exploited/nmap
124         fi
125         LOGDIR
126         }
127         LOGDIR
128
129     REMOTE
130     }
131     PRECHECK
132     ;;
```

Figure 6 Function to check and create directory for log and storing information

Figure 7 shows the terminal output of after running Option 2 (Run Pre-Operation Check). Once all the necessary checks have been done, it will display status as [+] and User can proceed to use Option 3 (EXECUTION) to perform the main task. If there is error when installing the package as shown in Figure 8, User should consider running Option 1 to update and upgrade the Operating System and then running Option 2 again.

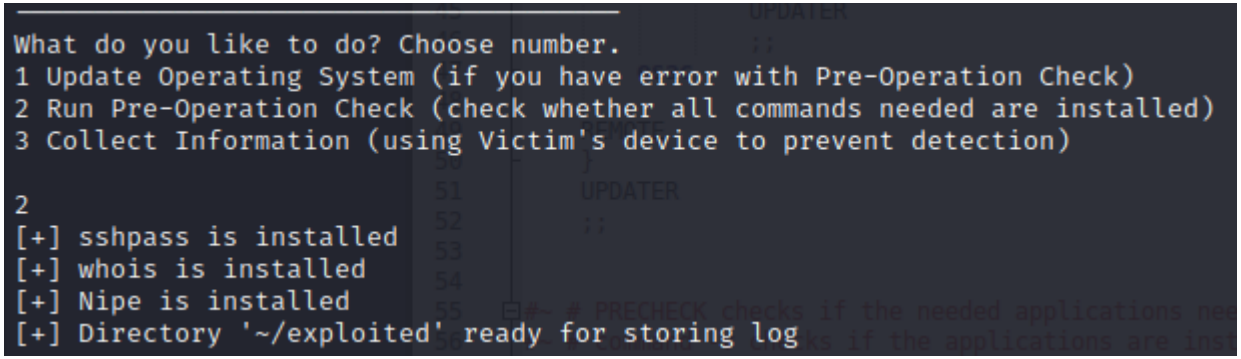

Figure 7 Pre-Operation Check Terminal Output

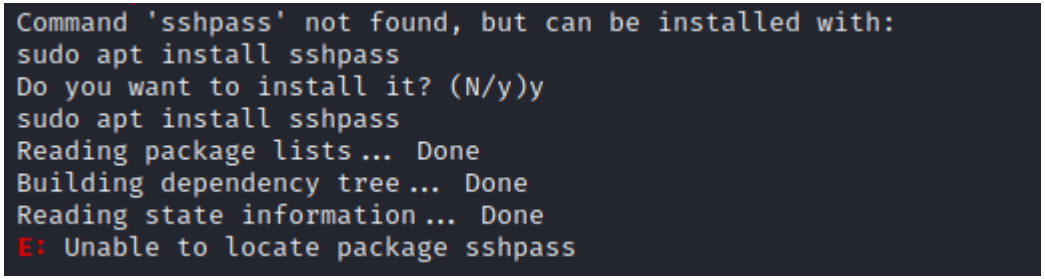

Figure 8 Error Output when installing sshpass

Execution

Figure 9 shows the first part of Execution. Anonymity check is placed into this part instead of Option 2(Pre-Execution Check) as a safety measure. After repeated usage of the script from the same user, one may forget to run Option 2 due to convenience, thus placing the User as risk of identity exposure. At times, the nipe engine may fail to start and return no ip address at all, especially when the user just started up his machine. Line 155 to 158 aims to resolves this until the engine can output an IP address. However, it doesn't guarantee that a foreign ip address have been successful used as our default gateway, so Line 160 and 167 will compare with our original public IP. When *spoofip* is not the same as *myip,* the script will determine that User can safely carry out his next activity, if not it will not allow user to carry out the next activity.

```
139     # ANONCHECK checks if User have successfully spoofed a foreign IP address using Nipe application
140     # This is to ensure identity anonimity to prevent traceback before executing activities
141     function ANONCHECK()
142     {
143     clear -x
144     echo "Checking Anonymity..."
145     # Declaring variables.
146     # myip is User's machine IP, which can be changed when this script is used by another machine
147     # niperr is used to start Nipe
148     # spoofip and spoofcc stores the spoofed ip and country code
149     myip='101.127.159.84'
150     niperr=$(echo 'kali' | sudo -S perl nipe.pl restart 2>/dev/null)
151     spoofip=$(curl -s ifconfig.io)
152     spoofcc=$(curl -s ifconfig.io/country_code)
153     echo -e "\n[*] Your Public IP is $myip"
154     # if spoofip returns an empty string, it will call function ANONCHECK to start nipe again
155     if [[ -z "$spoofip" ]]
156     then
157         echo "spoof is warming up.."
158         ANONCHECK
159     # once spoofip is not empty, it will compare with User's IP to ensure anonimity
160     elif [[ "$myip" != "$spoofip" ]]
161     then
162         echo "[+] You are ANONYMOUS now!"
163         echo "[*] Your spoofed IP address is: $spoofip"
164         echo "[*] Your spoofed Country: $spoofcc"
165         sleep 2
166         return
167     elif [[ "$myip" == "$spoofip" ]]
168     then
169         echo "[X] ALERT ALERT! You are traceable! Exiting.."
170         cd /home/kali/nipe
171         sudo perl nipe.pl stop
172         sleep 2
173         exit
174     fi
175     ANONCHECK
176     }
177     ANONCHECK
```

Figure 9 Option 3- Execution (Part 1): Anonimity Check

Figure 10 shows the Second part of Execution. From Line 191 to 193, the script checks if the User has input a valid IP Address, if it is not, then it will prompt user to input a valid IP again. If the User gives a domain address, it will carry out the script in Figure 11, which is largely similar with minor alteration to the head and tail of the script to suit the usage for domain address input.

From line 197, the argument is passed to check it the User machine is able to connect to the compromised machine. If not, in line 218, it will notify User that the machine is not available and carry out Execution with another compromised machine. Once User have successfully accessed the compromised machine in, the script checks the computer details like Uptime, IP address and Location. It will then proceed to collect information using 'nmap' and 'whois' into User's local machine.

```
184     # EXPLOIT checks if you have entered a valid ip address or domain name and proceed to execute NMAP & WHOIS and log activity in my local computer
185     # My ubuntu(192.168.48.129) is used as the compromised Remote Server
186     function EXPLOIT()
187     {
188     echo -e "\n[?] Specify the Domain/IP Address to scan:"
189     read ipd
190
191     if [[ $ipd =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]];
192     then
193         if [[ $(echo $ipd | awk -F. '{print $1}') -le 255 && $(echo $ipd | awk -F. '{print $2}') -le 255 && $(echo $ipd | awk -F. '{print $3}') -le 255 && $(echo $ipd | awk -F. '{print $4}') -le 255 ]]
194         then
195             export SSHPASS='tc'
196             echo -e "\n[*] Connecting to Remote Server:"
197             if sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 w >/dev/null 2>&1
198             then
199                 echo "System Uptime is:"
200                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "w|grep "up""
201                 echo "System IP Address:"
202                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "curl -s ifconfig.io"
203                 echo "System Country:"
204                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "curl -s ifconfig.io/country_code"
205                 echo -e "\n[*] whoising target address:"
206                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129  "whois $ipd" > /home/kali/exploited/whois/whois_$ipd
207                 # $? is the exit status of the most recently-executed command; by convention, 0 means success and anything else indicates failure.
208                 if [ $? -eq 0 ]
209                 then
210                 echo "[@] whois data is saved into /home/kali/exploited/whois/whois_$ipd"; echo "$(date) whois data collected for: $ipd" >> /home/kali/exploited/remotectrl.log
211                 echo -e "\n[*] Scanning victim address:"
212                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "nmap -sV -Pn -p 1-100 $ipd" > /home/kali/exploited/nmap/nmap_$ipd
213                 echo "[@] nmap data is saved into /home/kali/exploited/nmap/nmap_$ipd"; echo "$(date) nmap data collected for: $ipd" >> /home/kali/exploited/remotectrl.log
214                 else
215                 echo "whois/nmap not successful"
216                 fi
217             else
218                 echo "Remote Server is offline.. Please find another Online Remote Server"
219             fi
220             return
221         else
222             echo "an invalid ip"
223             EXPLOIT
224         fi
```

Figure 10 Option 3- Execution (Part 2): Exploit (IP address input)

```
226     else
227         export SSHPASS='tc'
228             echo -e "\n[*] Connecting to Remote Server:"
229             if sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 w >/dev/null 2>&1
230             then
231                 echo "System Uptime is:"
232                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "w|grep "up""
233                 echo "System IP Address:"
234                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "curl -s ifconfig.io"
235                 echo "System Country:"
236                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "curl -s ifconfig.io/country_code"
237                 echo -e "\n[*] whoising target address:"
238                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129  "whois $ipd" > /home/kali/exploited/whois/whois_$ipd
239                 if [ $? -eq 0 ]
240                 then
241                 echo "[@] whois data is saved into /home/kali/exploited/whois/whois_$ipd"; echo "$(date) whois data collected for: $ipd" >> /home/kali/exploited/remotectrl.log
242                 echo -e "\n[*] Scanning victim address:"
243                 sshpass -e ssh -o StrictHostKeyChecking=no tc@192.168.48.129 "nmap -sV -Pn -p 1-100 $ipd" > /home/kali/exploited/nmap/nmap_$ipd
244                 echo "[@] nmap data is saved into /home/kali/exploited/nmap/nmap_$ipd"; echo "$(date) nmap data collected for: $ipd" >> /home/kali/exploited/remotectrl.log
245                 else
246                 echo "whois/nmap not successful"
247                 fi
248             else
249                 echo "Remote Server is offline.. Please find another Online Remote Server"
250             fi
251             return
252     fi
253     REMOTE
254     }
255     EXPLOIT
256     ;;
257     esac
```

Figure 11 Option 3- Execution (Part 2): Exploit (Domain address input)

Results from Execution

Error Checking

Figure 12 shows the error message when the compromised machine is offline and Figure 13 shows the error message when invalid IP or invalid domain address is given. This help to prevent false positive results and logs being stored.

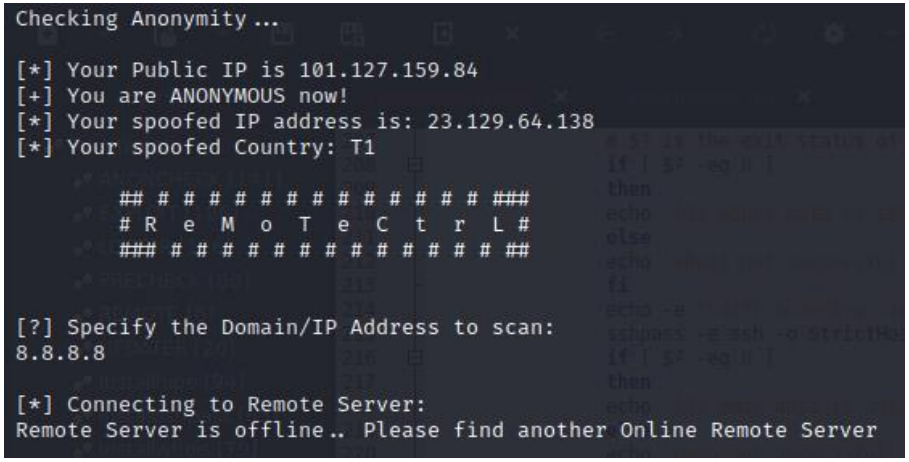

Figure 12 Error Checking: Compromised Machine is offline

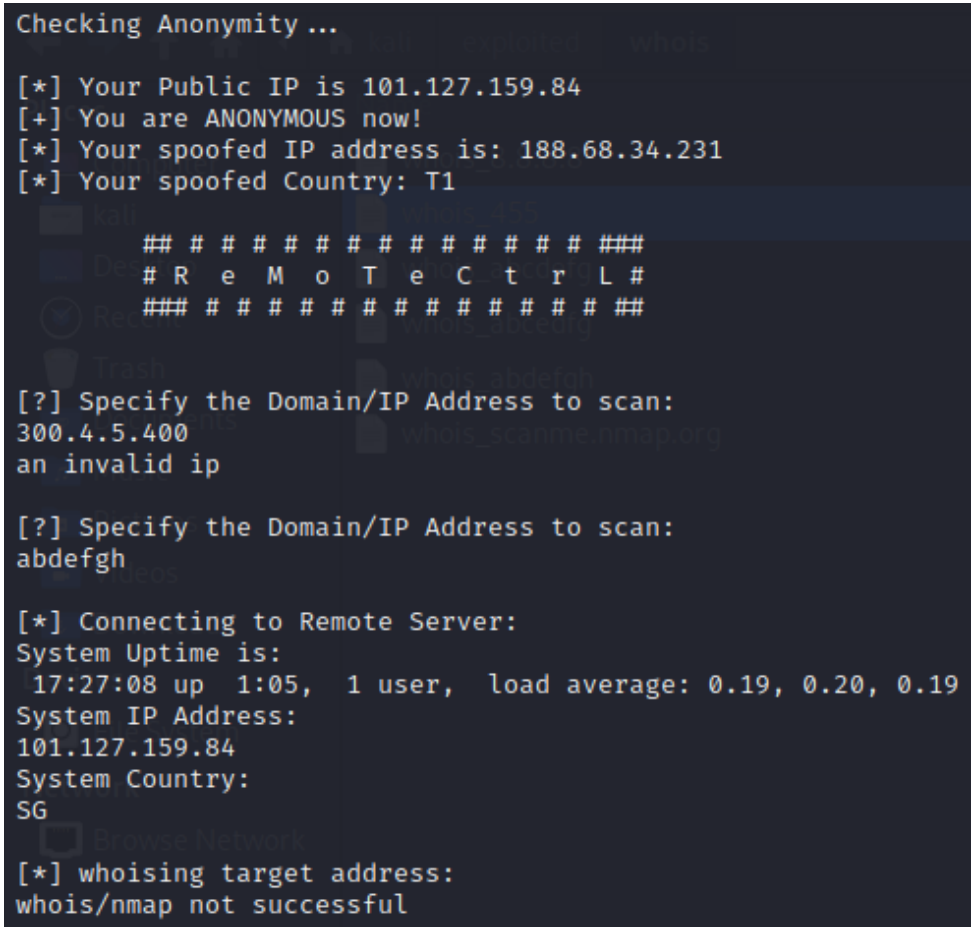

Figure 13 Error Checking: Invalid IP and Domain address

Valid Input with Results

If valid IP address and Domain Address is given the following output will be displayed in the terminal as shown in Figure 14 and Figure 15. Figure 16 and Figure 17 shows the content being stored.

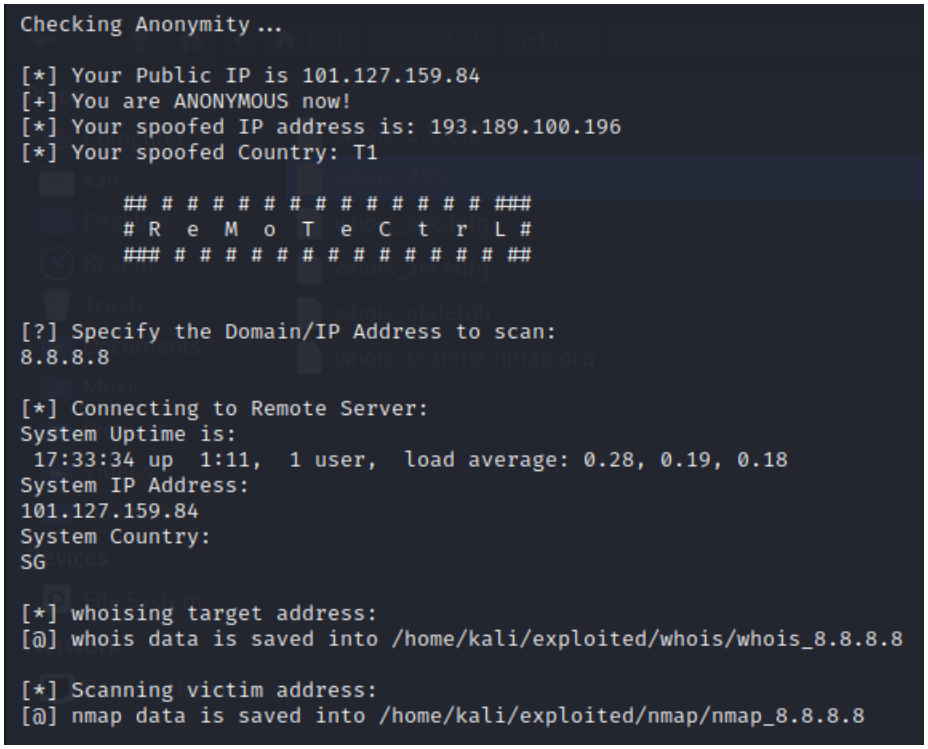

Figure 14 Valid Input: IP Address

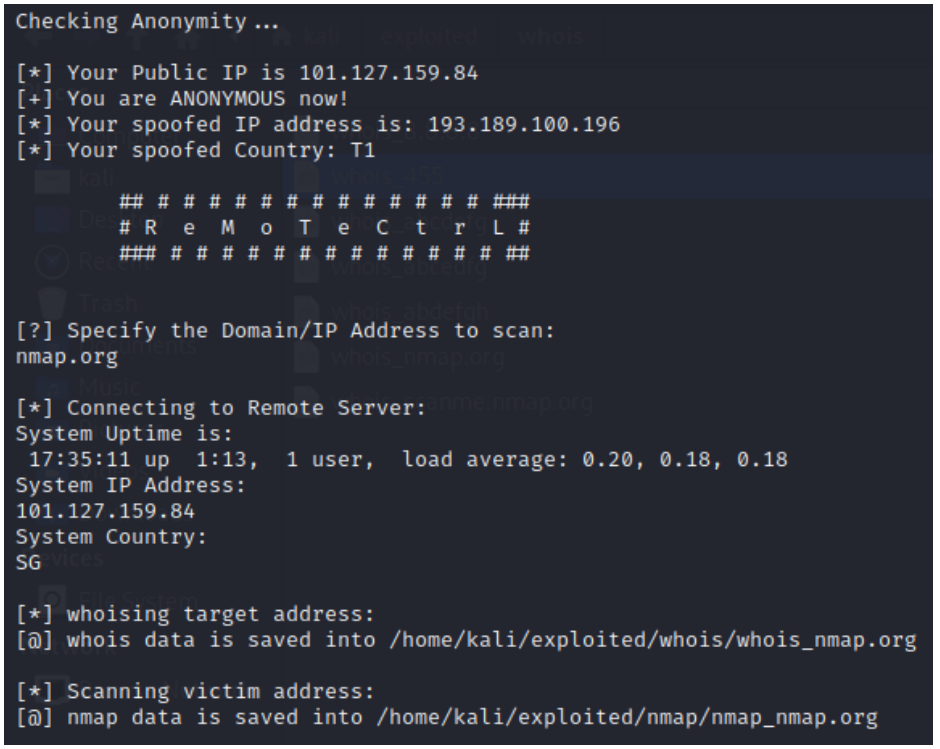

Figure 15: Valid Input: Domain Address

```
┌──(kali㉿kali)-[~/nipe]
└─$ cat /home/kali/exploited/nmap/nmap_nmap.org
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-20 17:35 UTC
Nmap scan report for nmap.org (45.33.49.119)
Host is up (0.18s latency).
Other addresses for nmap.org (not scanned): 2600:3c01:e000:3e6::6d4e:7061
rDNS record for 45.33.49.119: ack.nmap.org
Not shown: 97 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
25/tcp open  smtp    Postfix smtpd
80/tcp open  http    Apache httpd 2.4.6
Service Info: Host:  ack.nmap.org

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.29 seconds

┌──(kali㉿kali)-[~/nipe]
└─$ cat /home/kali/exploited/whois/whois_nmap.org
Domain Name: nmap.org
Registry Domain ID: 5ed7a21fc9f74f97b55511f9857111f0-LROR
Registrar WHOIS Server: http://whois.fabulous.com
Registrar URL: http://www.fabulous.com
Updated Date: 2020-01-14T05:38:40Z
Creation Date: 1999-01-18T05:00:00Z
Registry Expiry Date: 2028-01-18T05:00:00Z
Registrar: Sea Wasp, LLC
Registrar IANA ID: 411
Registrar Abuse Contact Email: support@fabulous.com
Registrar Abuse Contact Phone: +61.282133006
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: Insecure.Com LLC
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: WA
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: US
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
```

Figure 16 Information stored in nmap_<domain address> and whois_<domain address> when given valid input

```
┌──(kali㉿kali)-[~/nipe]
└─$ cat /home/kali/exploited/nmap/nmap_8.8.8.8
Starting Nmap 7.80 ( https://nmap.org ) at 2023-03-20 17:33 UTC
Nmap scan report for dns.google (8.8.8.8)
Host is up (0.035s latency).
Not shown: 99 filtered ports
PORT   STATE SERVICE    VERSION
53/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.57 seconds

┌──(kali㉿kali)-[~/nipe]
└─$ cat /home/kali/exploited/whois/whois_8.8.8.8

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
#


# start

NetRange:       8.0.0.0 - 8.127.255.255
CIDR:           8.0.0.0/9
NetName:        LVLT-ORG-8-8
NetHandle:      NET-8-0-0-0-1
Parent:         NET8 (NET-8-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Level 3 Parent, LLC (LPL-141)
RegDate:        1992-12-01
Updated:        2018-04-23
Ref:            https://rdap.arin.net/registry/ip/8.0.0.0



OrgName:        Level 3 Parent, LLC
OrgId:          LPL-141
Address:        100 CenturyLink Drive
City:           Monroe
```

Figure 17 Information stored in nmap_<ipaddress> and whois_<ipaddress> when given valid input

Figure 18 logs the activities being executed by the User for ease of tracking and back-tracing for future reference.

```
┌──(kali㉿kali)-[~/nipe]
└─$ cat /home/kali/exploited/remotectrl.log
Sun Mar 12 03:39:22 AM +08 2023 whois data collected for: scanme.nmap.org
Sun Mar 12 03:39:53 AM +08 2023 nmap data collected for: scanme.nmap.org
Sun Mar 12 03:42:56 AM +08 2023 whois data collected for: http://scanme.nmap.org/
Sun Mar 12 03:42:56 AM +08 2023 nmap data collected for: http://scanme.nmap.org/
Tue Mar 21 01:33:37 AM +08 2023 whois data collected for: 8.8.8.8
Tue Mar 21 01:33:42 AM +08 2023 nmap data collected for: 8.8.8.8
Tue Mar 21 01:35:14 AM +08 2023 whois data collected for: nmap.org
Tue Mar 21 01:35:28 AM +08 2023 nmap data collected for: nmap.org
```

Figure 18 logs stored on remotectrl.log

Credits

How to execute your shell script from any directory [User: ThangTD]

https://stackoverflow.com/questions/874452/change-the-current-directory-from-a-bash-script

How to check if a program is already installed in the machine [User: lhunath]

https://stackoverflow.com/questions/592620/how-can-i-check-if-a-program-exists-from-a-bash-script

How to use sudo in non-interactive mode [Author: Kai Yuan]

https://www.baeldung.com/linux/sudo-non-interactive-mode

How to check if given IP address is valid [User: shannonman]

https://stackoverflow.com/questions/13777387/check-for-ip-validity

How to bypass add hostkey prompt when using ssh/sshpass [User: MarkHu]

https://askubuntu.com/questions/45679/ssh-connection-problem-with-host-key-verification-failed-error

How to check if command is executed successfully [User: Wyzard]

https://stackoverflow.com/questions/7101995/what-does-if-eq-0-mean-for-shell-scripts#:~:text=%24%3F%20is%20the%20exit%20status,whether%20the%20grep%20command%20succeeded.

How to check for OS updates with prompts [User: SightSpirit]

https://gist.github.com/SightSpirit/d7ba05e94aaad9c7d8d127ce62a0373e