APRIL 2023

NETWORK RESEARCH

Remote Control Project

PRESENTED TO

Centre for Cybersecurity

PRESENTED BY

Ryan Tan

TABLE OF CONTENTS

Objective	3
Methodology	4
Areas for improvements	13
References	14

OBECJTIVE

During attacks on the enemy's server, we waste valuable time typing wrong commands due to stress and are even exposed because the source addresses are exposed.

Attacking targets behind enemy lines can become much more practical, fast, and accurate if the execution of the tasks becomes automatic.

The source addresses cannot be revealed, as these are sensitive military facilities.

01 Install applications

02 Check if the network connection is anonymous

03 Connect & execute commands on remote server

04 Saving reports and logs on local computer

METHODOLOGY

01 INSTALL APPLICATIONS

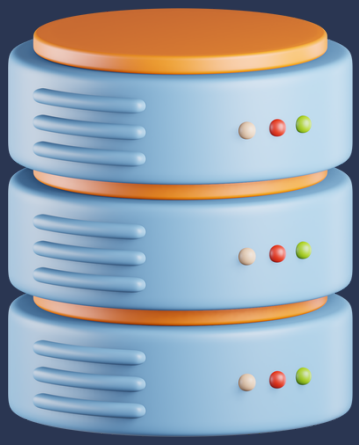
- Geoiplookup
- Nipe

02 NETWORK CONNECTION

- Activate Nipe
- Echo the spoofed IP address' country

Spoofed IP Address

SSH



REMOTE SERVER

03 REMOTE SERVER

- Connect via ssh
- Execute commands
  - curl ifconfig.co
  - geoiplookup
  - uptime
- whois & nmap scans on target IP

04 RESULTS ON LOCAL COMPUTER

- Save whois & nmap reports
- Log remote server activities in a file

(1) INSTALL APPLICATIONS

```
#!/bin/bash

echo -e '\e[1;33mWelcome to the remote controller\e[0m' && sleep 3
echo ' '
echo -e  '\e[1;32m(1) This remote controller makes your connection anonymous\e[0m' && sleep 3
echo -e '\e[1;32m(2) Thereafter it executes a whois and nmap scan on a target IP on a specificed remote server\e[0m' && sleep 5
echo ' '
echo -e '\e[1;33mBefore we start this process, we must ensure you have all the necessary applications installed\e[0m' && sleep 4

# '\e[1;33m <text> \e[0m' uses color codes to add some visual appeal to your output
    # \e is the escape sequence that tells the terminal emulator that a color code is about to follow.
    # \e[33m: yellow
    # \e[32m: green
    # \e[0m is another escape sequence that tells the terminal emulator to stop interpreting color codes.
```

```
Welcome to the remote controller

(1) This remote controller makes your connection anonymous
(2) Thereafter it executes a whois and nmap scan on a target IP on a specificed remote server

Before we start this process, we must ensure you have all the necessary applications installed
```

- These are just instructions to get the remote controller started

```
echo ' '
geoipapp=$(dpkg -l | grep geoip-bin)
geoipinstall=$(echo $geoipapp)

# geoiplookup helps to give the country of a specified IP adress
# dpkg is a package manager for Debian-based systems
    # dpkg -l lists the installed packages on the system
    # We 'grep geoip-bin' to check if geoiplookup has been installed
# geoipinstall just echoes the status of the geoiplookup installation
# Credits: https://www.baeldung.com/linux/list-installed-packages

if [ -z "$geoipinstall" ]
then
    echo -e "\e[1;33mInstalling geoiplookup package, please input password if prompted\e[0m" && sleep 2
    sudo apt-get install geoip-bin && sleep 3
    echo -e '\e[1;32mGeoiplookup has been installed\e[0m'
else
    sleep 2
    echo -e '\e[1;32mNice, you have Geoiplookup installed\e[0m'
fi
# -z means 'if empty'
# Therefore if geouplookup is uninstalled, the script will install it.
# If installed, it will not install it again.
```

- Commands to check the installation status of whois

```
nipeapp=$(find ~ -type d -name nipe)
nipeinstall=$(echo $nipeapp)
# nipe helps to anonymise your ip address by using the onion router (Tor) network as a user's default gateway
# find in /home/$user if a directory named nipe exists
# nipeinstall just echoes the directory location

if [ -z "$nipeinstall" ]
then
    echo -e '\e[1;33mInstalling nipe, please input password if prompted\e[0m' && sleep 1
    git clone https://github.com/htrgouvea/nipe && cd nipe
    # We must clone this repository from GitHub

    sudo cpan install Try::Tiny Config::Simple JSON
    # To install the libraries and dependencies

    sudo perl nipe.pl install && sleep 3
    # To install Nipe dependencies or a Perl script
        # Perl is a family of script programming languages that is similar in syntax to the C language

    echo -e '\e[1;32mNipe has been installed\e[0m'
    # Credits: https://www.geeksforgeeks.org/how-to-install-nipe-tool-in-kali-linux/
else
    sleep 2
    echo -e '\e[1;32mNice, you have Nipe installed\e[0m'
fi
# -z means 'if empty'
# Therefore, if nipe directory does not exists, the script will install nipe.
# If directory exists, it will not install it again.
```

- Commands to check the installation status of nipe

```
Nice, you have Geoiplookup installed

Nice, you have Nipe installed
```

- Output if applications are installed

```
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following NEW packages will be installed:
  geoip-bin
0 upgraded, 1 newly installed, 0 to remove and 1401 not upgraded.
Need to get 0 B/73.6 kB of archives.
After this operation, 478 kB of additional disk space will be used.
Selecting previously unselected package geoip-bin.
(Reading database ... 388862 files and directories currently installed.)
Preparing to unpack .../geoip-bin_1.6.12-10_arm64.deb ...
Unpacking geoip-bin (1.6.12-10) ...
Setting up geoip-bin (1.6.12-10) ...
Processing triggers for kali-menu (2022.4.1) ...
Processing triggers for man-db (2.11.0-1+b1) ...
Geoiplookup has been installed

Installing nipe, please input password if prompted
Cloning into 'nipe' ...
remote: Enumerating objects: 1714, done.
remote: Counting objects: 100% (185/185), done.
remote: Compressing objects: 100% (108/108), done.
remote: Total 1714 (delta 73), reused 145 (delta 57), pack-reused 1529
Receiving objects: 100% (1714/1714), 262.90 KiB | 77.00 KiB/s, done.
Resolving deltas: 100% (886/886), done.
Loading internal logger. Log::Log4perl recommended for better logging
Reading '/root/.cpan/Metadata'
  Database was generated on Thu, 13 Apr 2023 11:17:02 GMT
Try::Tiny is up to date (0.31).
Config::Simple is up to date (4.58).
JSON is up to date (4.10).
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
tor is already the newest version (0.4.7.13-1).
iptables is already the newest version (1.8.9-2).
0 upgraded, 0 newly installed, 0 to remove and 1401 not upgraded.
Nipe has been installed
```

- Output if applications are not installed

(2) SPOOF IP ADDRESS

```
nipedir=$(find ~ -type d -name nipe)
cd $nipedir
# Find in /home/$user to print the nipe directory
# We then change directories into the nipe folder to start the service

echo ' ' && sleep 1
echo -e '\e[1;33mStarting the nipe service, please input password if prompted\e[0m'
sudo perl nipe.pl stop && sleep 3
sudo perl nipe.pl start && sleep 3
sleep 2 && echo -e '\e[32mPlease wait as we run the service\e[0m'
echo ' '

nipestatus=$(echo "$(sudo perl nipe.pl status)" | grep -i false)
IPX=$(echo "$(sudo perl nipe.pl status)" | grep -i ip | awk -F: '{print $2}' | tr -d "[:blank:]")

# nipestatus checks if the nipe service has not started
    # grep -i ignores case sensitivity
# IPX greps the IP address from the nipestatus output
    # awk -F: "{print $2}' prints the 2nd column with ':' as the separator
    # tr -d "[:blank]" deletes blank spaces
```

```
if [ -z $nipestatus ]
then
    echo 'Spoofed country name:'
    geoiplookup "$IPX" | awk -F: '{print $2}'
else
    echo -e '\e[1;33mConnection is not anonymous, please exit\e[0m'
    kill -9 $$
    # The -9 option tells kill to send a SIGKILL signal, which immediately terminates the process.
    # $$ is a variable used to store the PID of a running script.
    # Credits: https://www.javatpoint.com/kill-command-in-linux#:~:text=It%20is%20used%20for%20manually,%24%20type%20%2Da%20kill
fi
# -z means 'if empty'
# Therefore, if the nipe has started, the script will echo the spoofed country
# If nipe service has not started, the script will stop running
```

- Commands to ensure nipe is running to spoof IP address

```
Starting the nipe service, please input password if prompted
[sudo] password for kali:
Please wait as we run the service

Spoofed country name:
 DE, Germany
```

- Output when nipe is successfully running

(3) REMOTE SERVER

```
echo -e '\e[1;33mTarget IP Address for Whois & Nmap scan:\e[0m'
read targetip
# read allows for users of the script to input a string
```

```
Target IP Address for Whois & Nmap scan:
```

- Commands to get user to input target IP address for scans

```
echo ' '
echo -e '\e[1;32mConnecting to the remote server, please wait\e[0m' && sleep 1
echo ' '
echo -e '\e[1;33mPlease input the remote server IP Address:\e[0m'
read IPadd
echo -e '\e[1;33mPlease input the username of the remote server:\e[0m'
read Username
echo -e '\e[1;33mPlease input the password of the remote server:\e[0m'
read -s Password
# -s does not echo input, credits: https://www.baeldung.com/linux/bash-hide-user-input
sleep 2
```

```
Connecting to the remote server, please wait

Please input the remote server IP Address:
192.168.34.132
Please input the username of the remote server:
kali
Please input the password of the remote server:
```

- Getting users to input details to ssh into the remote server

```
echo ' '
date | tee -a ~/remoteseverlog.txt && sleep 1
remoteIP=$(sshpass -p $Password ssh -o StrictHostKeyChecking=no $Username@$IPadd "curl -s ifconfig.me")

# tee prints the output of a command
    # -a appends to a given file, in this case /home/$user/remoteserver.log.txt
    # Credits: https://www.geeksforgeeks.org/tee-command-linux-example/
# Syntax of sshpass -> sshpass -p <Password> ssh <Username>@Ipaddress
    # " " is to specify the commands to execute after entering the remote server
    # -o StrictHostKeyChecking=no skips the host key checking everytime you do a new SSH

echo ' '
echo "Remote Server IP: $remoteIP" | tee -a ~/remoteseverlog.txt && sleep 1
echo ' '
echo "Remote Sever IP Country: $(geoiplookup $remoteIP | awk -F: '{print $2}')" | tee -a ~/remoteseverlog.txt && sleep 1
echo ' '
echo "Remote Sever uptime: $(uptime)" | tee -a ~/remoteseverlog.txt && sleep 1
echo ' '
echo "****************************************End of session****************************************" | tee -a ~/remoteseverlog.txt
# geoiplookup finds the country of a particular IP Address
# uptime prints the current time, the length of time the system has been up, the number of users online, and the load average.
    # The load average is the number of runnable processes over the preceding 1-, 5-, 15-minute intervals.
    # Credits: https://www.ibm.com/docs/en/aix/7.2?topic=u-uptime-command

echo ' ' && sleep 2
echo -e '\e[1;33mRemote server activities are logged\e[0m' && sleep 2
echo -e '\e[32mLogs are stored in remoteseverlog.txt that is saved in your User Directory\e[0m' && sleep 5
```

```
Fri 14 Apr 2023 04:37:24 PM +08

Remote Server IP: 39.109.210.31

Remote Sever IP Country:  SG, Singapore

Remote Sever uptime:  16:37:28 up 1 day, 17:10,  1 user,  load average: 0.00, 0.02, 0.00

****************************************End of session****************************************

Remote server activities are logged
Logs are stored in remoteseverlog.txt that is saved in your User Directory
```

- Perform multiple commands in the remote server to find key info
  - information is then logged into a file

```
echo ' ' & sleep 2
echo -e '\e[1;32mRunning whois and nmap scan, please wait\e[0m'
sshpass -p $Password ssh -o StrictHostKeyChecking=no $Username@$IPadd "whois $targetip > whoisreport.txt"
sshpass -p $Password ssh -o StrictHostKeyChecking=no $Username@$IPadd "nmap $targetip -Pn -T5 -oN nmapscan.txt"
# > whoisreport.txt saves output of whois to a text file
# nmap scans for open ports of a server
    # -Pn used to skip the host discovery stage of the scanning process
        # Assume that the target host(s) are online and available for scanning.
    # -T5 sets the speed of the scan
    # -oN injects output of the scan into a file
```

```
Running whois and nmap scan, please wait
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-14 16:37 +08
Nmap scan report for 192.168.34.132
Host is up (0.00013s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

- Running whois and nmap scans and saving the reports

(4) SAVING REPORTS

```
echo ' '
echo -e '\e[1;33mSaving Whois & Nmap report, please choose file location:\e[0m'
echo -e '\e[32mA) User Directory B) Downloads C) New Directory\e[0m'
read OPTIONS

case $OPTIONS in
    A|a)
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the whois report:\e[0m'
        scp $Username@$IPadd:/home/$Username/whoisreport.txt ~
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the nmap report:\e[0m'
        scp $Username@$IPadd:/home/$Username/nmapscan.txt ~
        echo ' ' && sleep 1
        echo -e '\e[32mReports are saved in your User Directory\e[0m'
    ;;
    B|b)
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the whois report:\e[0m'
        scp $Username@$IPadd:/home/$Username/whoisreport.txt ~/Downloads
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the nmap report:\e[0m'
        scp $Username@$IPadd:/home/$Username/nmapscan.txt ~/Downloads
        echo ' ' && sleep 1
        echo -e '\e[32mReports are saved in ~/Downloads\e[0m'
```

```
    C|c)
        echo -e '\e[1;33mPlease input name of new directory\e[0m'
        read newdir
        mkdir ~/$newdir
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the whois report:\e[0m'
        scp $Username@$IPadd:/home/$Username/whoisreport.txt ~/$newdir
        echo -e '\e[1;33mIf prompted, please input password of remote server to download the nmap report:\e[0m'
        scp $Username@$IPadd:/home/$Username/nmapscan.txt ~/$newdir
        echo ' ' && sleep 1
        echo -e "\e[32mReports are saved in ~/$newdir\e[0m"
        ;;
        esac
# Case allows for the user to choose from a variety of options
# scp copies files or directories between a local and a remote system
    # The syntax is <username>@<remoteserverIP>:<file location> <destination>
```

- Gives the user the options on where to save the reports on their local computer

```
Saving Whois & Nmap report, please choose file location:
A) User Directory B) Downloads C) New Directory
A
If prompted, please input password of remote server to download the whois report:
kali@192.168.34.132's password:
whoisreport.txt                                              100% 2781     1.3MB/s   00:00
If prompted, please input password of remote server to download the nmap report:
kali@192.168.34.132's password:
nmapscan.txt                                                 100%  346   273.0KB/s   00:00

Reports are saved in User Directory
```

```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads  nipe          nrproject.sh  Public           Templates  whoisreport.txt
Documents  Music      nmapscan.txt  Pictures      remoteseverlog.txt  Videos
```

- Output for option A

```
Saving Whois & Nmap report, please choose file location:
A) User Directory B) Downloads C) New Directory
B
If prompted, please input password of remote server to download the whois report:
kali@192.168.34.132's password:
whoisreport.txt                                              100% 2781     1.5MB/s   00:00
If prompted, please input password of remote server to download the nmap report:
kali@192.168.34.132's password:
nmapscan.txt                                                 100%  347   481.2KB/s   00:00

Reports are saved in ~/Downloads
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls
nmapscan.txt  Ubuntu_22.10_Kinetic_Kudu.png  whoisreport.txt
```

- Output for option B

```
Saving Whois & Nmap report, please choose file location:
A) User Directory B) Downloads C) New Directory
C
Please input name of new directory
testing
If prompted, please input password of remote server to download the whois report:
tc@164.92.189.42's password:
whoisreport.txt                                              100% 2781     5.2KB/s   00:00
If prompted, please input password of remote server to download the nmap report:
tc@164.92.189.42's password:
nmapscan.txt                                                 100%  356     0.7KB/s   00:00

Reports are saved in ~/testing
```

```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads  nipe          Pictures  remoteseverlog.txt  testing
Documents  Music      nrproject.sh  Public    Templates           Videos
```

```
┌──(kali㉿kali)-[~/testing]
└─$ ls
nmapscan.txt  whoisreport.txt
```

- Output for option C

```
sshpass -p $Password ssh -o StrictHostKeyChecking=no $Username@$IPadd "rm whoisreport.txt && rm nmapscan.txt"
# Delete files from the remote server

echo ' '  && sleep 2
echo -e '\e[1;33mThank you for using this remote controller\e[0m' && sleep 2
echo -e '\e[1;32mPlease contact Ryan, @rustygrapes via telegram if there are any issues\e[0m' && sleep 4
echo ' '
echo -e '\e[1;35mThank you once again & Goodbye~\e[0m' && sleep 2
```

```
Thank you for using this remote controller
Please contact Ryan, @rustygrapes via telegram if there are any issues

Thank you once again & Goodbye~
```

- Delete the reports from the remote server
- Echos final few sentences to end the script

```
┌──(kali㉿kali)-[~/testing]
└─$ cat nmapscan.txt
# Nmap 7.92 scan initiated Fri Apr 14 09:06:25 2023 as: nmap -Pn -T5 -oN nmapscan.txt 192.168.34.132
Nmap scan report for 192.168.34.132
Host is up.
All 1000 scanned ports on 192.168.34.132 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

# Nmap done at Fri Apr 14 09:07:16 2023 -- 1 IP address (1 host up) scanned in 51.19 seconds
```

- nmap report results

```
┌──(kali㉿kali)-[~/testing]
└─$ cat whoisreport.txt

#
# 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.
#


NetRange:       192.168.0.0 - 192.168.255.255
CIDR:           192.168.0.0/16
NetName:        PRIVATE-ADDRESS-CBLK-RFC1918-IANA-RESERVED
NetHandle:      NET-192-168-0-0-1
Parent:         NET192 (NET-192-0-0-0-0)
NetType:        IANA Special Use
OriginAS:
Organization:   Internet Assigned Numbers Authority (IANA)
RegDate:        1994-03-15
Updated:        2013-08-30
Comment:        These addresses are in use by many millions of independently operated networks, which might be as sm
all as a single computer connected to a home gateway, and are automatically configured in hundreds of millions of de
vices.  They are only intended for use within a private context  and traffic that needs to cross the Internet will n
eed to use a different, unique address.
```

- whois report results

```
┌──(kali㉿kali)-[~]
└─$ cat remoteseverlog.txt
Fri 14 Apr 2023 04:37:24 PM +08
Remote Server IP: 39.109.210.31
Remote Sever IP Country:  SG, Singapore
Remote Sever uptime:  16:37:28 up 1 day, 17:10,  1 user,  load average: 0.00, 0.02, 0.00
****************************************End of session****************************************
Fri 14 Apr 2023 04:54:07 PM +08
Remote Server IP: 39.109.210.31
Remote Sever IP Country:  SG, Singapore
Remote Sever uptime:  16:54:10 up 1 day, 17:27,  1 user,  load average: 0.15, 0.06, 0.01
****************************************End of session****************************************
Fri 14 Apr 2023 05:01:02 PM +08
Fri 14 Apr 2023 05:02:31 PM +08
Remote Server IP: 164.92.189.42
Remote Sever IP Country:  DE, Germany
Remote Sever uptime:  17:02:40 up 1 day, 17:35,  1 user,  load average: 0.09, 0.05, 0.01
****************************************End of session****************************************
Fri 14 Apr 2023 05:06:00 PM +08
Remote Server IP: 164.92.189.42
Remote Sever IP Country:  DE, Germany
Remote Sever uptime:  17:06:07 up 1 day, 17:39,  1 user,  load average: 0.06, 0.05, 0.01
****************************************End of session****************************************
```

- Remote server logs so that user can track of ssh usage & outputs

AREAS FOR IMPROVEMENT

1) Could explore the use of functions

- might reduce the number of lines in my script

2) For ssh, allow users to use username & password files to allow for brute forcing into the remote server

3) Remote server log files can also include the actual commands used

REFERENCES

https://www.baeldung.com/linux/list-installed-packages

https://www.geeksforgeeks.org/how-to-install-nipe-tool-in-kali-linux/

https://www.javatpoint.com/kill-command-in-linux#:~:text=It%20is%20used%20for%20manually,%24%20type%20%2Da%20kill

https://www.geeksforgeeks.org/tee-command-linux-example/

https://www.ibm.com/docs/en/aix/7.2?topic=u-uptime-command