PROJECT

REMOTE CONTROL

NETWORK RESEARCH

DARREN LEE

12th April 2023

CONTENTS

OBJECTIVE

EXPLAINATION OF THE METHODS

OBJECTIVE

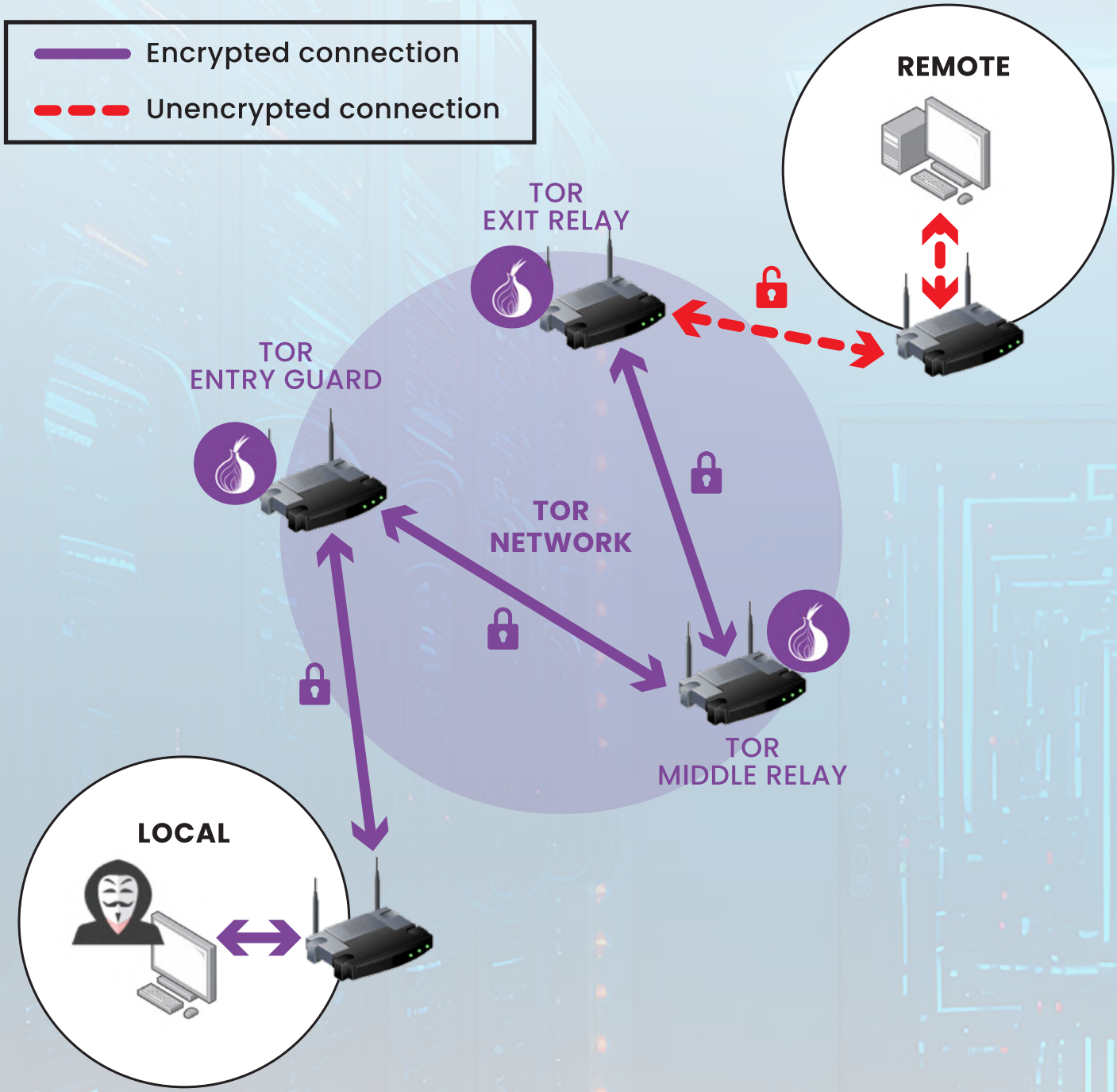
I. PURPOSE OF THE PROJECT

- To create a bash script that automate specific task that can help improve efficiency and reduce human error while establish a connection with a remote server anonymously.

II. PROJECT'S OUTCOME

- By the end of the project, there should have a fully functional and secure bash script that can communicate with a remote server, automate tasks, maintain anonymity, transfering files, data or resources to a local computer.

III. DIAGRAMS



EXPLAINATION OF THE METHODS

SCREENSHOT SCRIPT WITH EXPLAINATION

Note: This script assumes that the geoip-bin (geoiplookup), whois, nmap, sshpass and nipe packages are installed on both the local computer and the remote server for successful execution.

The script begins by printing out message stating the current working directory and search for a directory “nipe” and prints its path. Using the command "sudo perl nipe.pl start" to begin the Nipe program and we grep for “Status” to print out the network status to show if it’s “true” (anonymous) or false (Not anonymous).

Then we store the the network status into a variable called "nipestatus" and scripted in a If and Else condition to see if the value of "nipestatus" is "true"then it’ll prints out "You are anonymous". Otherwise, it’ll prints out "You are exposed, Goodbye" and exit from the script.

**Nipe -- A perl script enables us to directly route all our traffic from our computer to the Tor network through which we can use the internet anonymously without having to worry about tracked or traced back.*

NRproj.sh

```
#!/bin/bash

#~ #geoip-bin is already installed.
#~ #tor is already installed.
#~ #sshpass is already installed.
#~ #Nipe is already installed.

#Check user's current working directory and move into Nipe folder
echo 'Your current working directory:'
pwd
echo ''
sleep 1.5

#Move into nipe directory
echo 'Searching for Nipe directory:'
cd nipe
pwd
echo ''

#Start Nipe to make connection become anonymous
echo 'Attempting to start Nipe...'
sudo perl nipe.pl start
sleep 5
echo ''

echo 'Checking connection, please wait...'
echo ''
sleep 1.5

#Check nipe status and grep for "Status: true"
sudo perl nipe.pl status | grep Status | awk '{print $2 $3}'
echo ''
sleep 6

#Create an IF statement that checks if the nipe connection status is true. IF yes (condition matches) echo "You are anonymous"
nipestatus=$(sudo perl nipe.pl status | grep Status | awk '{print $3}')
sleep 2

if [ $nipestatus == 'true' ]
then
    echo "You are annonymous"
else                                #The opposite condition
    echo 'You are exposed, Goodbye'
exit
fi
echo ''
```

On Line 50 to 56, we retrieves the spoofed IP address from the nipe status's check output and store it into a variable called "spoofedIP". Next, we search for the spoofed country based on the stored variable "spoofedIP" by using the "geoiplookup" command and grep for the IP address.

Line 61 prompts the user to enter the remote server's credentials and connects to it using the "sshpass" command with the provided credentials. The "-p" option specifies the password to be used for authentication.The "-s" option is to hide the input of the password for security purposes. The 'uptime' command inside the single quotes (') must be input in 1 line beside the ssh -p "$remote_password" ssh "$remote_user@$remote_host" syntax in order to executes and retrieves the uptime information of the remote server.

Note: Everytime when we want to input a command, it must be input beside the syntax ssh -p "$remote_password" ssh "$remote_user@$remote_host" <'command'>

On line 77 - 85 with the "curl -s ifconfig.me" command with "-s" option, silent mode, is to fetch only the external IP address without additional informations. We then store the obtained IP address of the remote server to "remote_serverIP" as a variable and perform "geoiplookup" command together with "awk -F" command using ":" (colon) as the field separator to extract only the country information.

NRproj.sh

```
47
48  #Print out user spoofed IP address
49  echo 'Your spoofed IP address is:'
50  sudo perl nipe.pl status | grep Ip | awk '{print $3}'
51  spoofedIP=$(sudo perl nipe.pl status | grep Ip | awk '{print $3}')
52  echo ''
53
54  #Print out user spoofed country
55  echo 'Your spoofed country:'
56  geoiplookup $spoofedIP | grep -i GeoIP | awk -F: '{print $2}' | sort | uniq
57  echo ''
58
59  #Get user input for remote server details
60  echo 'Remote server credentials'
61  read -p "Enter remote IP address: " remote_host
62  read -p "Enter remote username: " remote_user
63  read -s -p "Enter remote password: " remote_password
64  echo ''
65  echo ''
66  echo 'Connecting to remote server...'
67  echo ''
68  sleep 2
69
70  #Connect to remote server via sshpass and display it's uptime
71  echo "Remote server Uptime:"
72  sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'uptime'
73  echo ''
74
75  #Display the remote server's IP address
76  echo "Remote server IP address:"
77  sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'curl -s ifconfig.me'
78  sleep 2
79  echo ''
80  echo ''
81
82  #Display the remote server's country
83  echo "Remote server country:"
84  remote_serverIP=$(sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'curl -s ifconfig.me')
85  geoiplookup "$remote_serverIP" | awk -F: '{print $2}'
86  sleep 2
87  echo ''
88
89  #Do nmap scan & whoislook victim's IP address/domain on the remote server and save the result
90  echo "Perform Nmap scanning on victim IP address/domain: "
```

On line 91, performs an Nmap scan using the "nmap" command on a victim IP address/domain from the remote server with various options such as "-F" to scan 100 ports only, "-Pn" option is no ping, don't detect host. "-sV" options is to Identify the service version. "-vv" option is verbose — to print out the scanning progress. "-oN" option, normal scan, is to inject the output into a file normal scan and also show the commands being used. The results are saved to a file "nmapdata.txt". on the remote server.

Line 98, performs a "whois" command to scan on a victim IP address/domain on the remote server. The results are saved to a file "whoisdata.txt" on the remote server.

Line 107 prompts the user to enter the credentials of the local machine's IP address, username, password and destination file path to upload the files nmapdata.txt and whoisdata.txt from the remote server to the local machine.

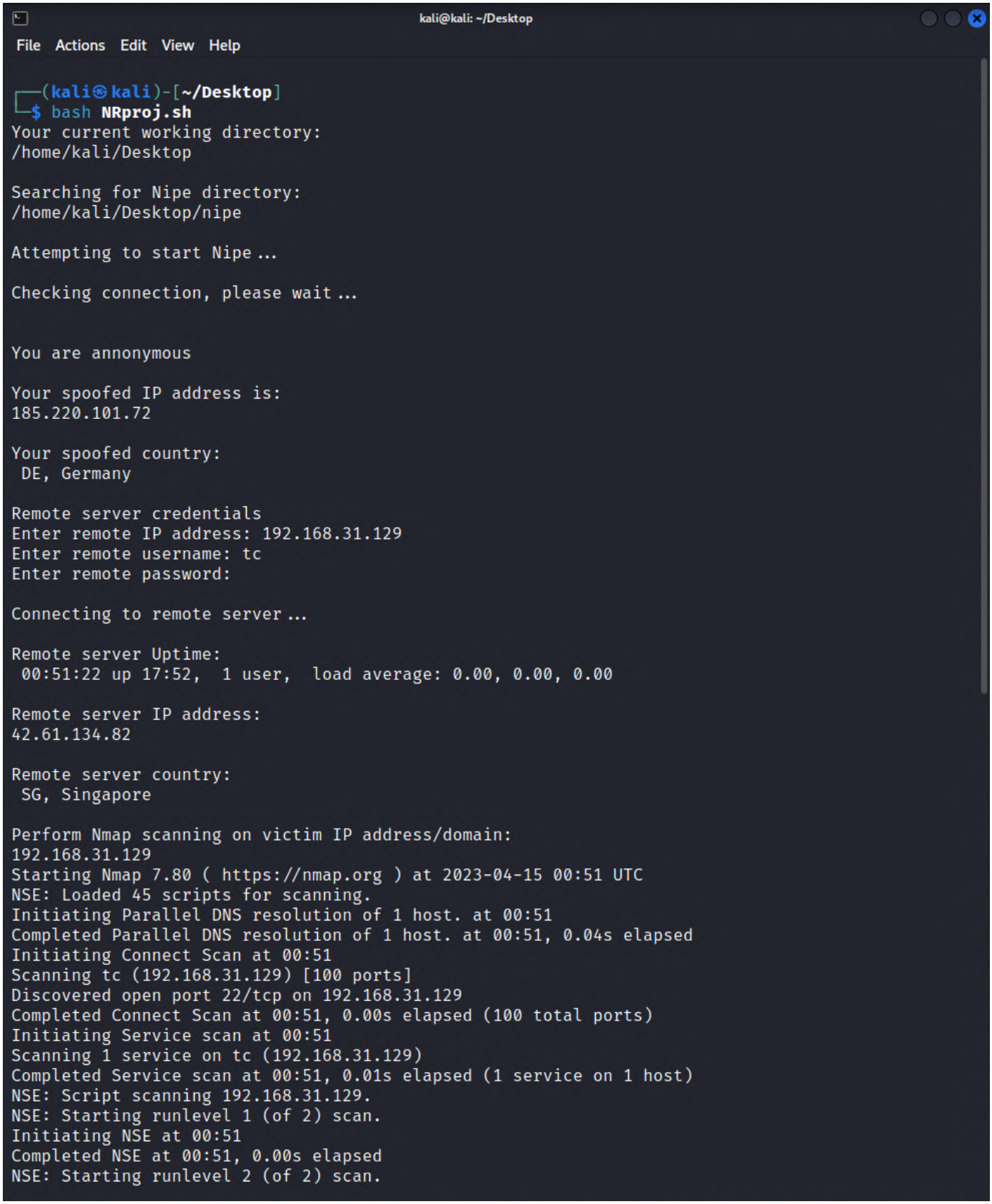
The "ls" command on line 120, list out the contents of current working directory such as files and folder. With the "rm" command to delete the files nmapdata.txt and whoisdata.txt on the remote server. Afterwards we do the "ls" command again to check that the files have been deleted then we exit from the remote server.

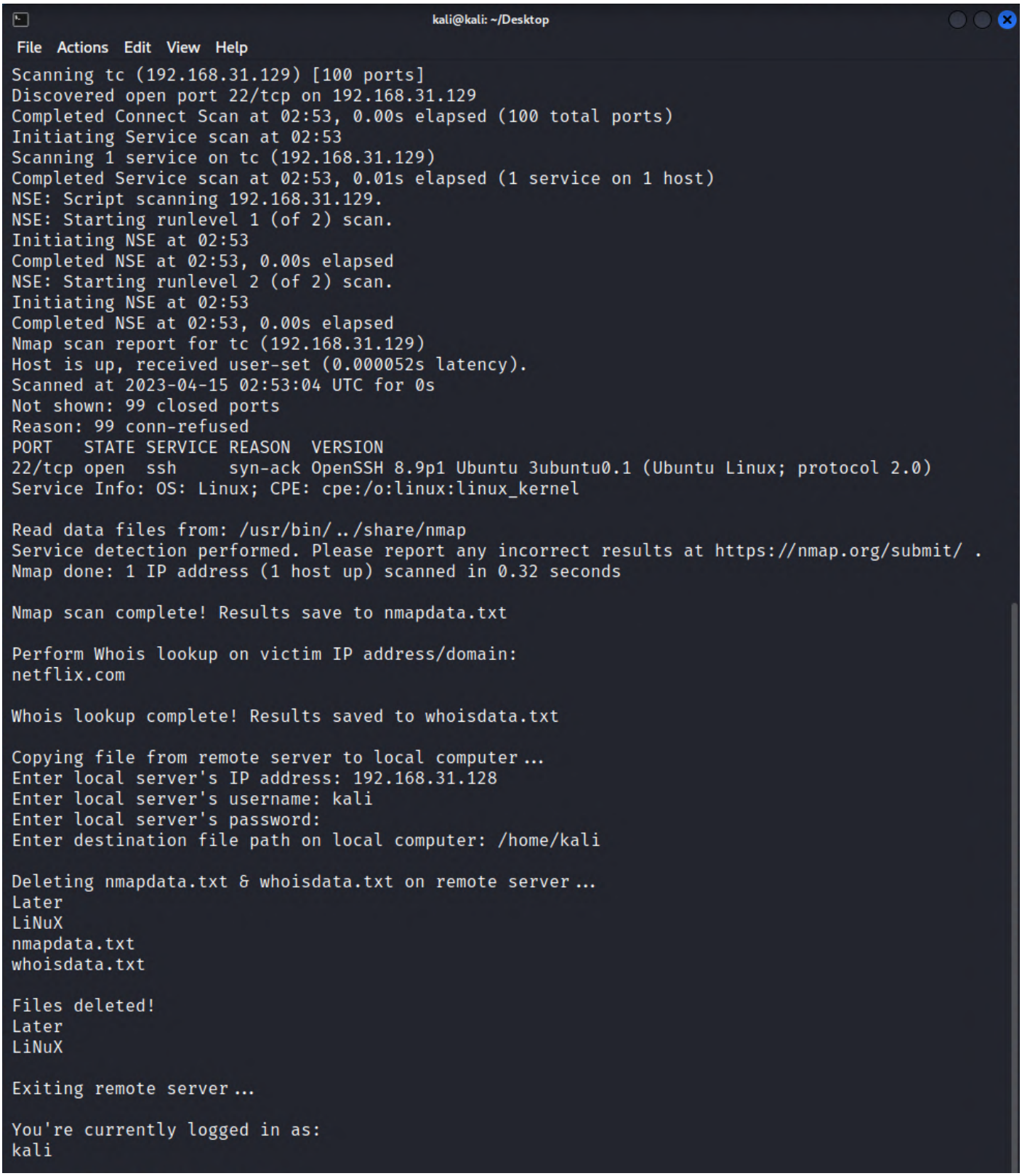
Finally we then do "whoami" command to display the current logged-in user on the local computer to confirm we have exited from the remote server.

NRproj.sh

```
89  #Do nmap scan & whoislook victim's IP address/domain on the remote server and save the result
90  echo "Perform Nmap scanning on victim IP address/domain: "
91  sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'read target; nmap -F -Pn -sV "$target" -vv -oN nmapdata.txt'
92  echo ''
93  sleep 1
94  echo "Nmap scan complete! Results save to nmapdata.txt"
95  echo ''
96  sleep 1
97  echo "Perform Whois lookup on victim IP address/domain: "
98  sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'read target; whois "$target" > whoisdata.txt'
99  echo ''
100 sleep 1
101 echo "Whois lookup complete! Results saved to whoisdata.txt"
102 echo ''
103 sleep 1
104
105 #Prompt user credential to upload file to local computer
106 echo "Copying file from remote server to local computer..."
107 read -p "Enter local server's IP address: " local_host
108 read -p "Enter local server's username: " local_user
109 read -s -p "Enter local server's password: " local_password
110 echo ''
111 read -p "Enter destination file path on local computer: " destination_file
112
113 #Copy file from remote to local machine
114 sshpass -p "$remote_password" scp "$remote_user@$remote_host:nmapdata.txt" "$destination_file"
115 sshpass -p "$remote_password" scp "$remote_user@$remote_host:whoisdata.txt" "$destination_file"
116 echo ''
117
118 #List out files on remote server and delete Nmap & Whois data save files
119 echo "Deleting nmapdata.txt & whoisdata.txt on remote server..."
120 sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'ls'
121 sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'rm nmapdata.txt whoisdata.txt'
122 echo ''
123 sleep 2
124
125 #List out files on remote server to confirm it's deleted
126 echo "Files deleted!"
127 sshpass -p "$remote_password" ssh "$remote_user@$remote_host" 'ls'
128 echo ''
129 sleep 2
130
131 echo "Exiting remote server..."
132 echo ''
133
134 #Check current logged in user
135 echo "You're currently logged in as:"
136 whoami
137
```

SCREENSHOT OF SCRIPT'S RESULT




kali@kali: ~/Desktop
File Actions Edit View Help
Scanning tc (192.168.31.129) [100 ports]
Discovered open port 22/tcp on 192.168.31.129
Completed Connect Scan at 02:53, 0.00s elapsed (100 total ports)
Initiating Service scan at 02:53
Scanning 1 service on tc (192.168.31.129)
Completed Service scan at 02:53, 0.01s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.31.129.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 02:53
Completed NSE at 02:53, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 02:53
Completed NSE at 02:53, 0.00s elapsed
Nmap scan report for tc (192.168.31.129)
Host is up, received user-set (0.000052s latency).
Scanned at 2023-04-15 02:53:04 UTC for 0s
Not shown: 99 closed ports
Reason: 99 conn-refused
PORT STATE SERVICE REASON VERSION
22/tcp open ssh syn-ack OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
Nmap scan complete! Results save to nmapdata.txt
Perform Whois lookup on victim IP address/domain:
netflix.com
Whois lookup complete! Results saved to whoisdata.txt
Copying file from remote server to local computer...
Enter local server's IP address: 192.168.31.128
Enter local server's username: kali
Enter local server's password:
Enter destination file path on local computer: /home/kali
Deleting nmapdata.txt & whoisdata.txt on remote server...
Later
LiNuX
nmapdata.txt
whoisdata.txt
Files deleted!
Later
LiNuX
Exiting remote server...
You're currently logged in as:
kali