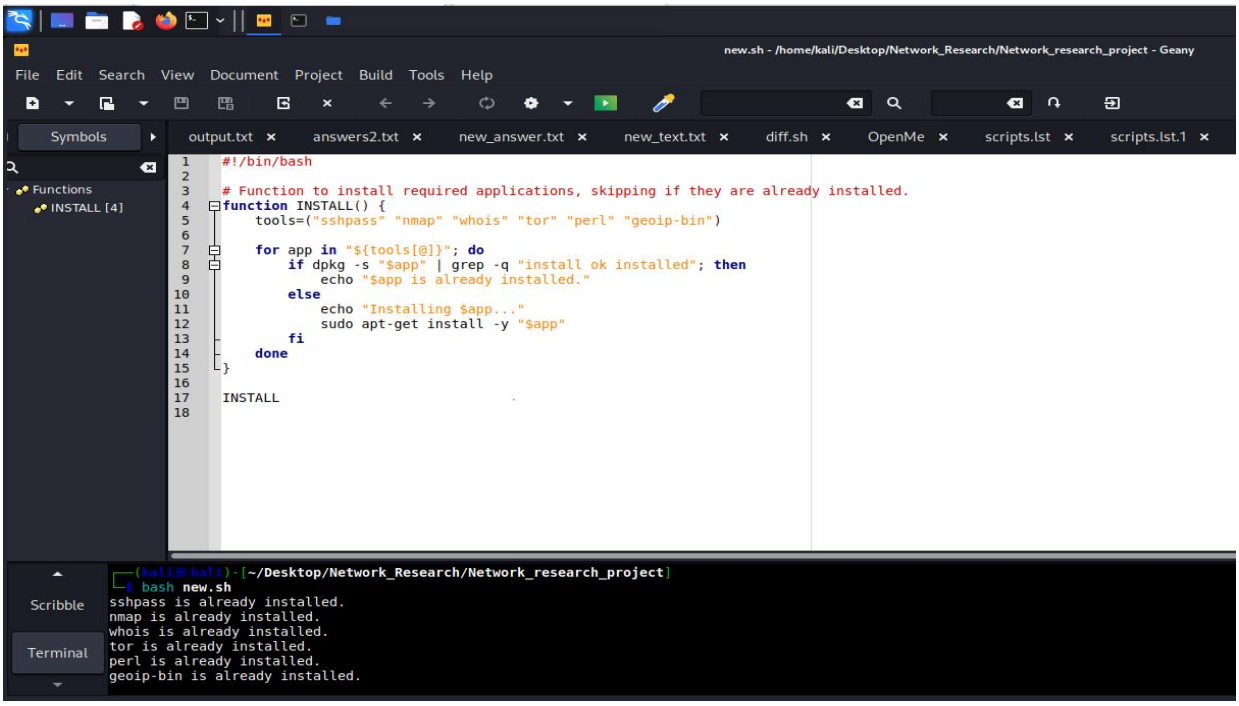
Unit Name: UNIT RW-University-II

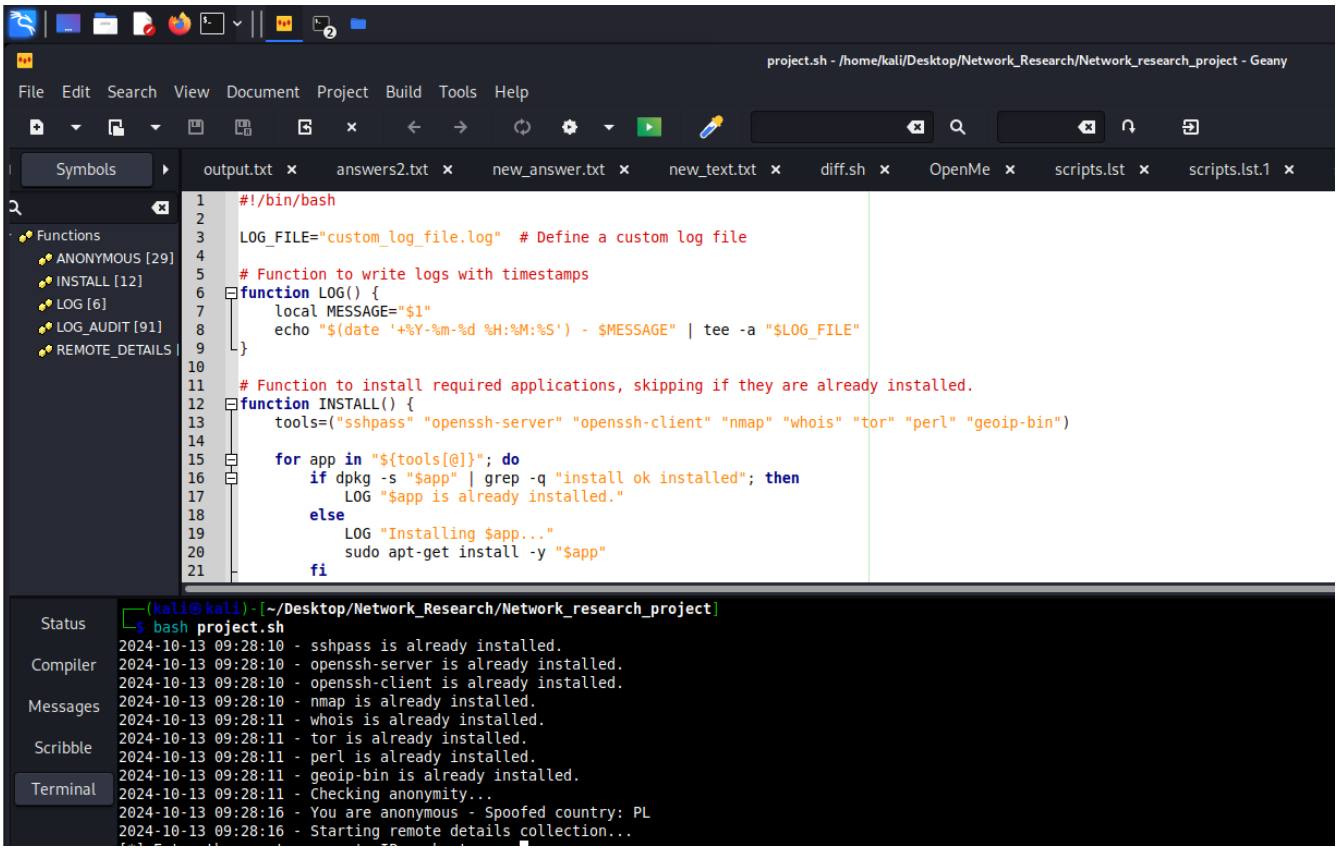
NETWORK RESERCH | PROJECT: REMOTE CONTROL

University of Rwanda (College of science and Technology)
Dept.: Information Technology

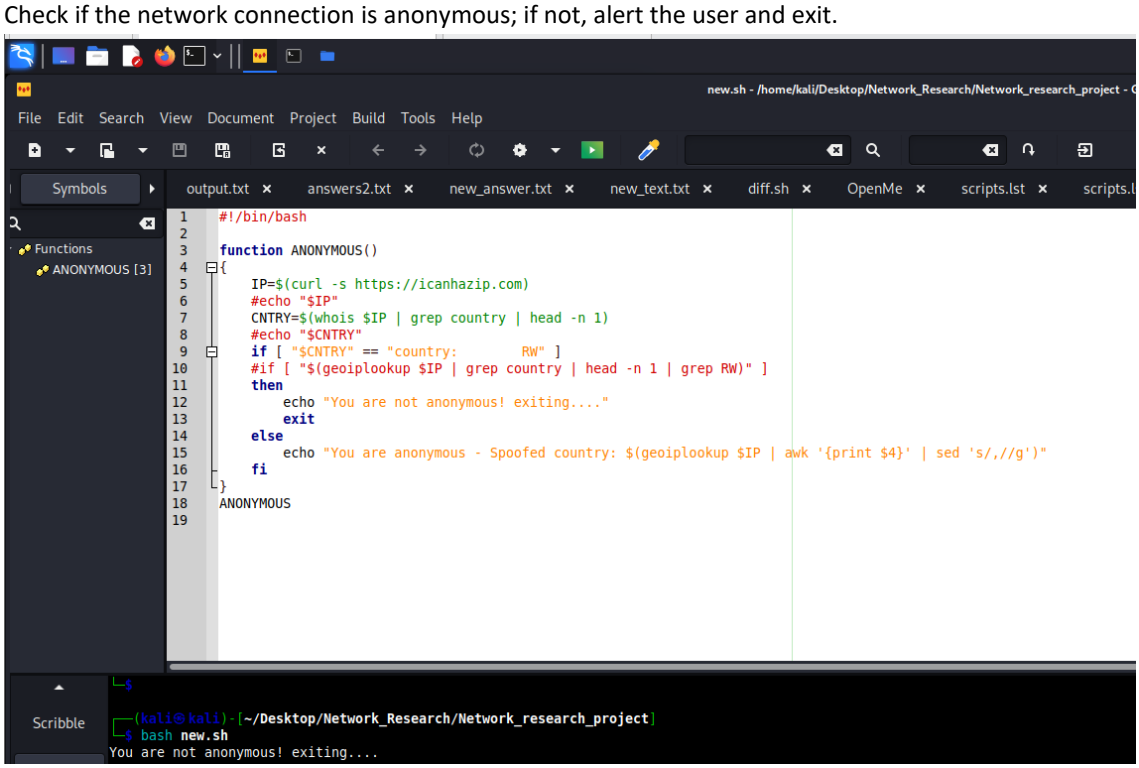
Project Structure

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.

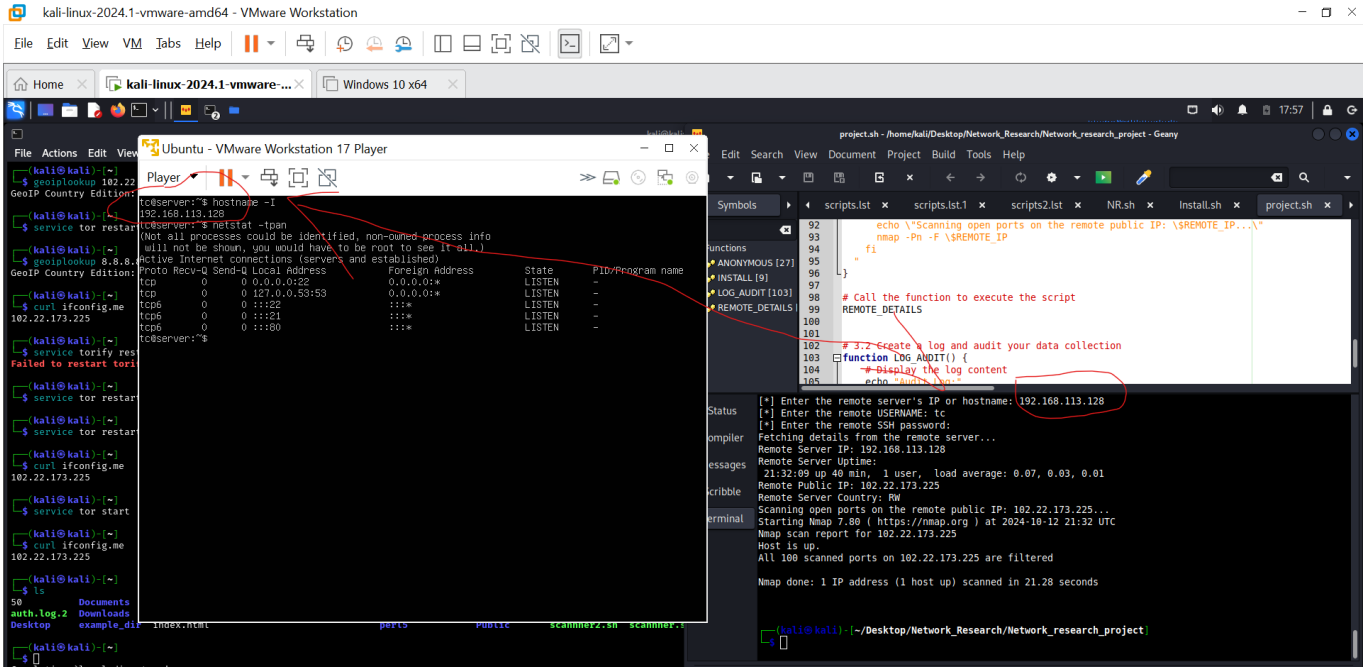
```
┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
└─$ sudo ./nipe.pl restart

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
└─$ sudo ./nipe.pl status

[!] ERROR: sorry, it was not possible to establish a connection to the server.


┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
└─$ sudo ./nipe.pl restart

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
└─$ sudo ./nipe.pl status

[+] Status: true
[+] Ip: 185.220.101.34
```

```
#!/bin/bash

function ANONYMOUS()
{
    IP=$(curl -s https://icanhazip.com)
    #echo "$IP"
    CNTRY=$(whois $IP | grep country | head -n 1)
    #echo "$CNTRY"
    if [ "$CNTRY" == "country:        RW" ]
    #if [ "$(geoiplookup $IP | grep country | head -n 1 | grep RW)" ]
    then
        echo "You are not anonymous! exiting...."
        exit
    else
        echo "You are anonymous - Spoofed country: $(geoiplookup $IP | awk '{print $4}' | sed 's/,//g')"
    fi
}
ANONYMOUS
```

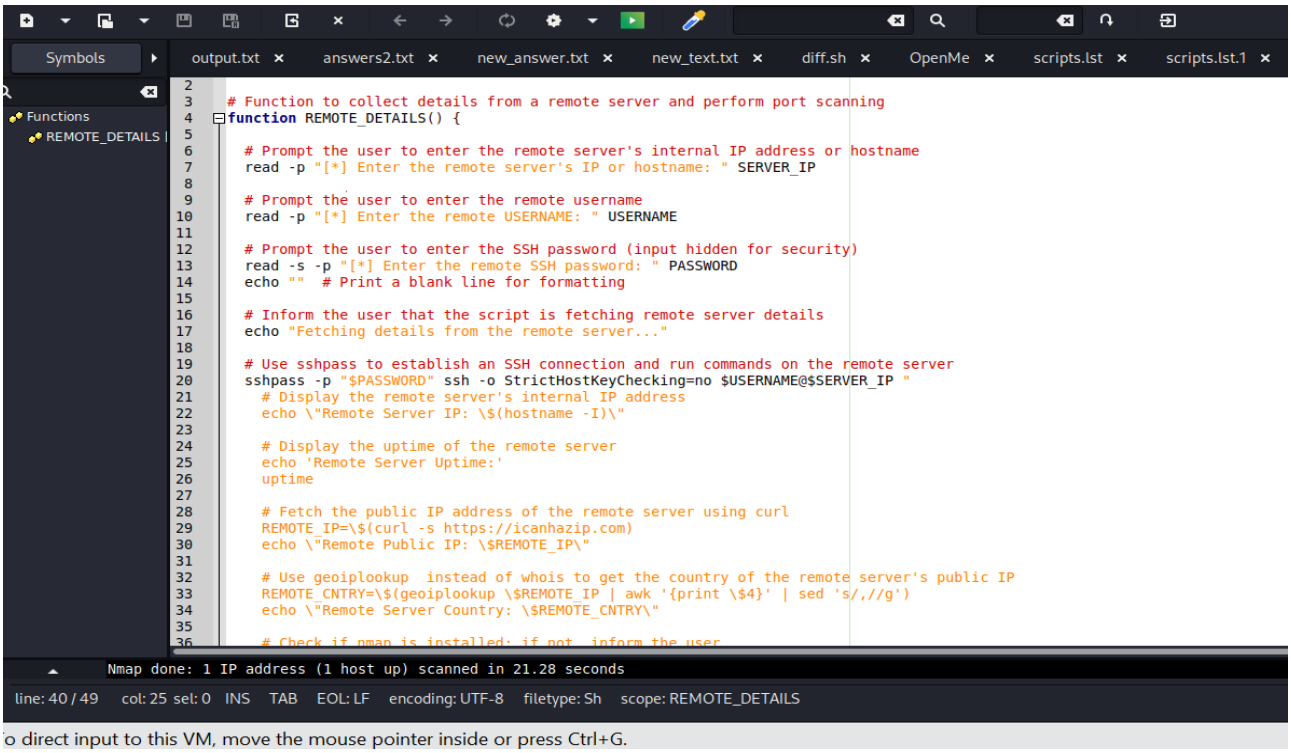
```
You are not anonymous! exiting....

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$ bash new.sh
You are anonymous - Spoofed country: PL

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$
```

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 SSI-I



2.1 Display the details of the remote server (country, IP, and Uptime).



```
┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$ bash new.sh
[*] Enter the remote server's IP: 192.168.113.128
[*] Enter the remote USERNAME: tc
[*] Enter the remote SSH password:
Fetching details from the remote server...
Remote Server IP: 192.168.113.128
Remote Server Uptime:
 21:05:07 up 13 min,  1 user,  load average: 0.00, 0.01, 0.00
Remote Public IP: 102.22.173.225
Remote Server Country: RW

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$
```

2.2 Get the remote server to check the Whois of the given address.

```
ANONYMOUS

# Function to collect details from a remote server and perform port scanning
function REMOTE_DETAILS() {
    LOG "Starting remote details collection..."

    read -p "[*] Enter the remote server's IP or hostname: " SERVER_IP
    read -p "[*] Enter the remote USERNAME: " USERNAME
    read -s -p "[*] Enter the remote SSH password: " PASSWORD
    echo ""  # Print a blank line for formatting

    LOG_AUDIT
    LOG "Fetching details from the remote server..."

    # Store the output of the SSH session locally and log relevant information
    sshpass -p "$PASSWORD" ssh -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP << 'EOF' >> remote_output.log

        echo "Remote Server IP: $(hostname -I)"
        echo "Remote Server Uptime:"
        uptime
        REMOTE_IP=$(curl -s https://icanhazip.com)
```

```
2024-10-13 09:28:16 - Starting remote details collection...
[*] Enter the remote server's IP or hostname: 192.168.113.128
[*] Enter the remote USERNAME: tc
[*] Enter the remote SSH password:
2024-10-13 09:32:00 - Appending log entry in the audit log file.
2024-10-13 09:32:00 - Fetching details from the remote server...
Pseudo-terminal will not be allocated because stdin is not a terminal.
2024-10-13 09:32:04 - SSH session completed. Fetching remote files...
2024-10-13 09:32:05 - Downloaded whois.lst successfully.
2024-10-13 09:32:05 - Downloaded nmap_file.lst successfully.
2024-10-13 09:32:05 - Remote details collection completed. See remote_output.log for details.
```

2.3 Get the remote server to scan for open ports on the given address.

```
┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$ cat remote_output.log
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Sun Oct 13 10:14:53 AM UTC 2024

  System load:  0.09               Processes:               231
  Usage of /:   40.8% of 9.75GB    Users logged in:         1
  Memory usage: 5%                 IPv4 address for ens33:  192.168.113.128
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 09:35 EDT
Nmap scan report for 192.168.113.128
Host is up (0.00066s latency).
Not shown: 96 closed tcp ports (reset)
PORT   STATE    SERVICE
21/tcp open     ftp
22/tcp open     ssh
53/tcp filtered domain
80/tcp open     http
MAC Address: 00:0C:29:7A:81:85 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.76 seconds
```

3. Results

3.1 Save the Whois and Nmap data into files on the local computer.

```
        echo "Remote Server IP: $(hostname -I)"
        echo "Remote Server Uptime:"
        uptime
        REMOTE_IP=$(curl -s https://icanhazip.com)
        echo "Remote Public IP: $REMOTE_IP"
        whois $REMOTE_IP >> whois.lst
        REMOTE_CNTRY=$(geoiplookup $REMOTE_IP | awk '{print $4}' | sed 's/,//g')
        echo "Remote Server Country: $REMOTE_CNTRY"

        if ! command -v nmap &> /dev/null; then
            echo "[!] nmap is not installed on the remote server."
        else
            echo "Scanning open ports on the remote server IP: $(hostname -I)..."
            nmap -Pn -F $(hostname -I) >> nmap_file.lst
        fi
EOF

    LOG "SSH session completed. Fetching remote files..."

    # Download files from the remote server
    sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:whois.lst . \
```

```
0-13 09:28:16 - Starting remote details collection...
ter the remote server's IP or hostname: 192.168.113.128
ter the remote USERNAME: tc
ter the remote SSH password:
0-13 09:32:00 - Appending log entry in the audit log file.
0-13 09:32:00 - Fetching details from the remote server...
-terminal will not be allocated because stdin is not a terminal.
0-13 09:32:04 - SSH session completed. Fetching remote files...
0-13 09:32:05 - Downloaded whois.lst successfully.
0-13 09:32:05 - Downloaded nmap_file.lst successfully.
0-13 09:32:05 - Remote details collection completed. See remote_output.log for details.
```

```
┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project/www]
└─$ ls
custom_log_file.log  nmap_file.lst  remote_output.log  whois.lst
```

3.2 Create a log and audit your data collecting.

Symbols | output.txt | answers2.txt | new_answer.txt | new_text.txt | diff.sh | OpenMe | scripts.ls

Functions
- ANONYMOUS [29]
- INSTALL [12]
- LOG [6]
- LOG_AUDIT [91]
- REMOTE_DETAILS

```
78
79      LOG "SSH session completed. Fetching remote files..."
80
81      # Download files from the remote server
82      sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:whois.lst . \
83          && LOG "Downloaded whois.lst successfully."
84      sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:nmap_file.lst . \
85          && LOG "Downloaded nmap_file.lst successfully."
86
87      LOG "Remote details collection completed. See remote_output.log for details."
88  }
89
90  # Function to log audit data
91  function LOG_AUDIT() {
92      LOG "Appending log entry in the audit log file."
93      echo "$(date) - Audit log entry" >> "$LOG_FILE"
94  }
95
96  # Call the function to execute the script
97  REMOTE_DETAILS
98
```

Status | Compiler | Messages | Scribble | Terminal

```
2024-10-13 09:28:16 - Starting remote details collection...
[*] Enter the remote server's IP or hostname: 192.168.113.128
[*] Enter the remote USERNAME: tc
[*] Enter the remote SSH password:
2024-10-13 09:32:00 - Appending log entry in the audit log file.
2024-10-13 09:32:00 - Fetching details from the remote server...
Pseudo-terminal will not be allocated because stdin is not a terminal.
2024-10-13 09:32:04 - SSH session completed. Fetching remote files...
2024-10-13 09:32:05 - Downloaded whois.lst successfully.
2024-10-13 09:32:05 - Downloaded nmap_file.lst successfully.
2024-10-13 09:32:05 - Remote details collection completed. See remote_output.log for details.

┌──(kali㉿kali)-[~/Desktop/Network_Research/Network_research_project]
└─$
```