

Unit Name: UNIT RW-University-II

NETWORK RESERCH | PROJECT: REMOTE CONTROL

University of Rwanda (College of science and Technology)

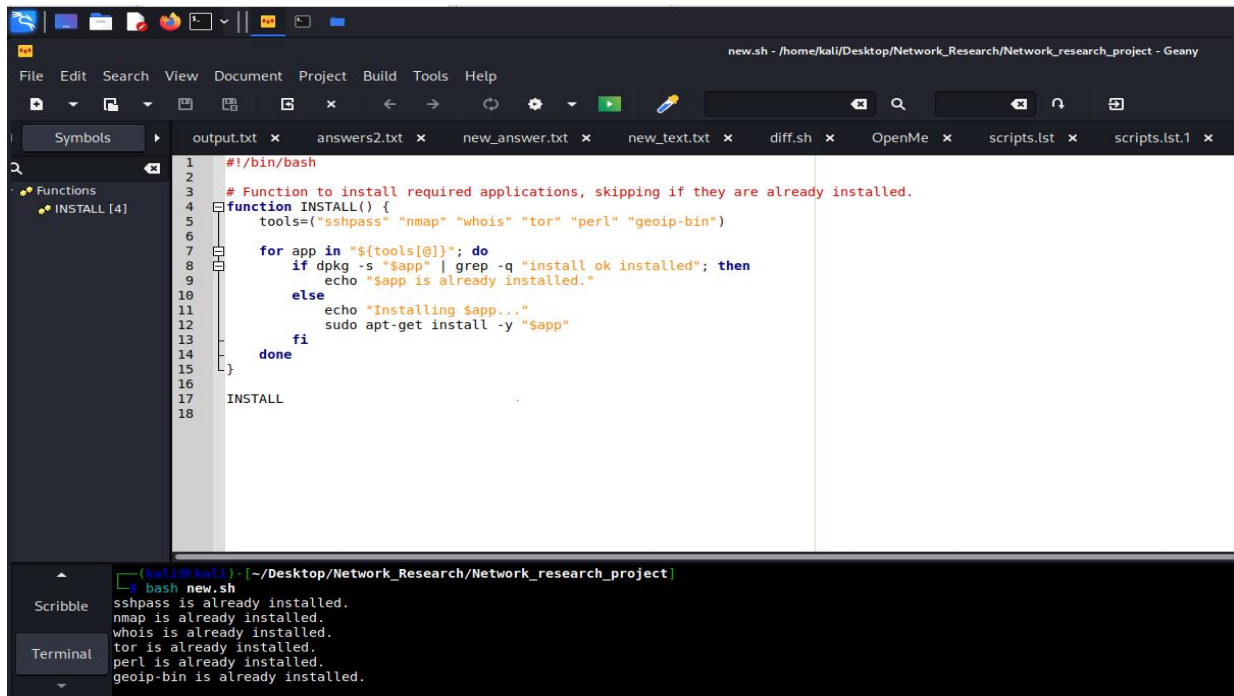
Dept.: Information Technology

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.



The screenshot displays the Geany IDE interface. The main editor window shows a Bash script named `new.sh` located at `/home/kali/Desktop/Network_Research/Network_research_project`. The script defines a function `INSTALL` that checks if several tools are installed and installs them if they are not. The tools listed are `sshpas`, `nmap`, `whois`, `tor`, `perl`, and `geop-bin`. The script then calls the `INSTALL` function. The terminal window at the bottom shows the output of running the script, confirming that all listed tools are already installed.

```
#!/bin/bash

# Function to install required applications, skipping if they are already installed.
function INSTALL() {
    tools=("sshpas" "nmap" "whois" "tor" "perl" "geop-bin")

    for app in "${tools[@]}; do
        if dpkg -s "$app" | grep -q "install ok installed"; then
            echo "$app is already installed."
        else
            echo "Installing $app..."
            sudo apt-get install -y "$app"
        fi
    done
}

INSTALL
```

Terminal Output:

```
(kali@kali) ~/Desktop/Network_Research/Network_research_project
bash new.sh
sshpas is already installed.
nmap is already installed.
whois is already installed.
tor is already installed.
perl is already installed.
geop-bin is already installed.
```

```
1 #!/bin/bash
2
3 LOG_FILE="custom_log_file.log" # Define a custom log file
4
5 # Function to write logs with timestamps
6 function LOG() {
7     local MESSAGE="$1"
8     echo "$(date '+%Y-%m-%d %H:%M:%S') - $MESSAGE" | tee -a "$LOG_FILE"
9 }
10
11 # Function to install required applications, skipping if they are already installed.
12 function INSTALL() {
13     tools=("sshpas" "openssh-server" "openssh-client" "nmap" "whois" "tor" "perl" "geoip-bin")
14
15     for app in "${tools[@]}; do
16         if dpkg -s "$app" | grep -q "install ok installed"; then
17             LOG "$app is already installed."
18         else
19             LOG "Installing $app..."
20             sudo apt-get install -y "$app"
21         fi
22     done
23 }
```

Status: (kali@kali) - [~/Desktop/Network_Research/Network_research_project]

Compiler: \$ bash project.sh

Messages: 2024-10-13 09:28:10 - sshpass is already installed.
2024-10-13 09:28:10 - openssh-server is already installed.
2024-10-13 09:28:10 - openssh-client is already installed.
2024-10-13 09:28:10 - nmap is already installed.
2024-10-13 09:28:11 - whois is already installed.
2024-10-13 09:28:11 - tor is already installed.
2024-10-13 09:28:11 - perl is already installed.
2024-10-13 09:28:11 - geoip-bin is already installed.

Scribble: 2024-10-13 09:28:11 - Checking anonymity...
2024-10-13 09:28:16 - You are anonymous - Spoofed country: PL
2024-10-13 09:28:16 - Starting remote details collection...

1.3 Check if the network connection is anonymous; if not, alert the user and exit.

```
1 #!/bin/bash
2
3 function ANONYMOUS()
4 {
5     IP=$(curl -s https://icanhazip.com)
6     #echo "IP"
7     CNTRY=$(whois $IP | grep country | head -n 1)
8     #echo "CNTRY"
9     if [ "$CNTRY" == "country: RW" ]
10     then
11         if [ "$(geoiplookup $IP | grep country | head -n 1 | grep RW)" ]
12         then
13             echo "You are not anonymous! exiting...."
14             exit
15         else
16             echo "You are anonymous - Spoofed country: $(geoiplookup $IP | awk '{print $4}' | sed 's/,//g')"
17         fi
18     fi
19 }
```

Scribble: (kali@kali) - [~/Desktop/Network_Research/Network_research_project]

\$ bash new.sh

You are not anonymous! exiting....

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

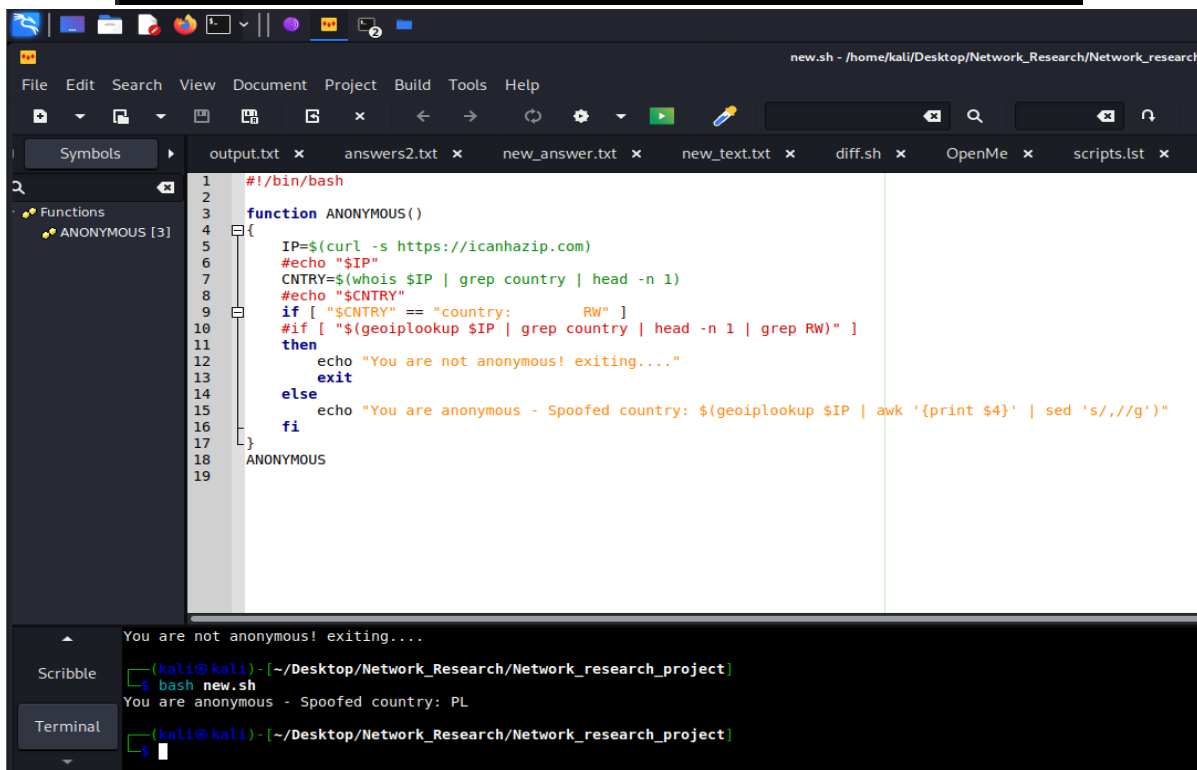
(kali@kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
$ sudo ./nipe.pl status

[!] 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

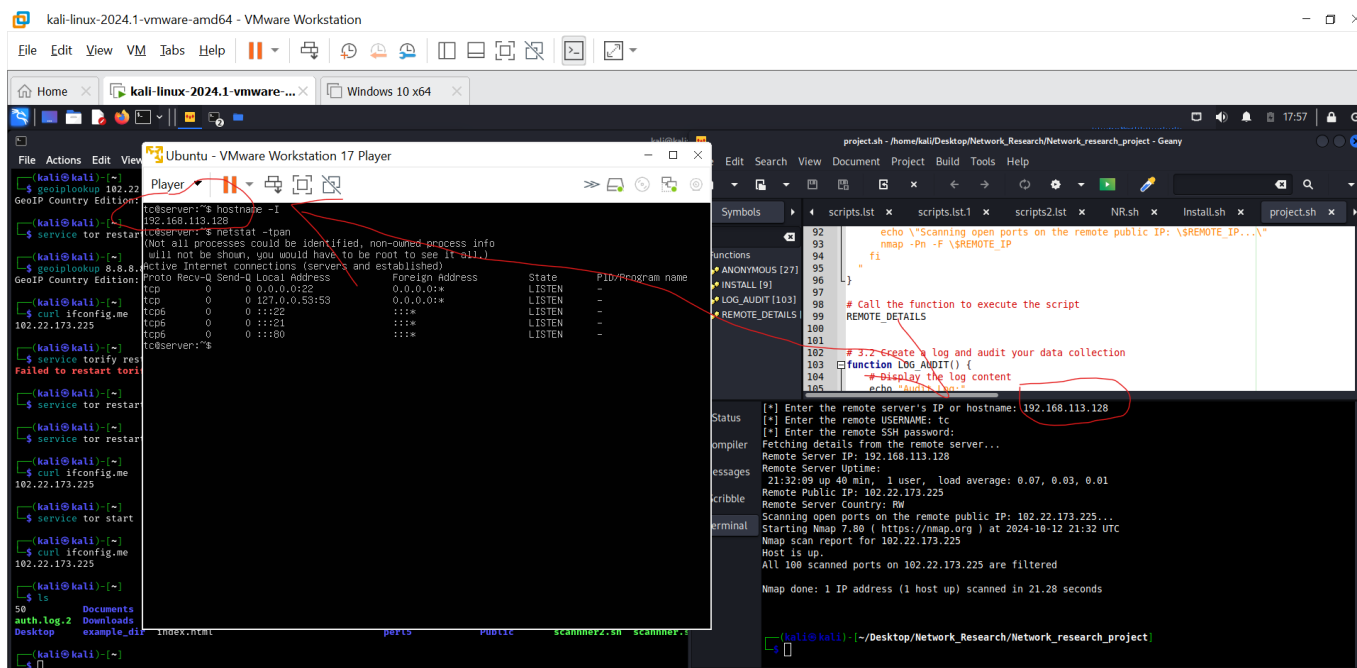
(kali@kali)-[~/Desktop/Network_Research/Network_research_project/nipe]
$ sudo ./nipe.pl status

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

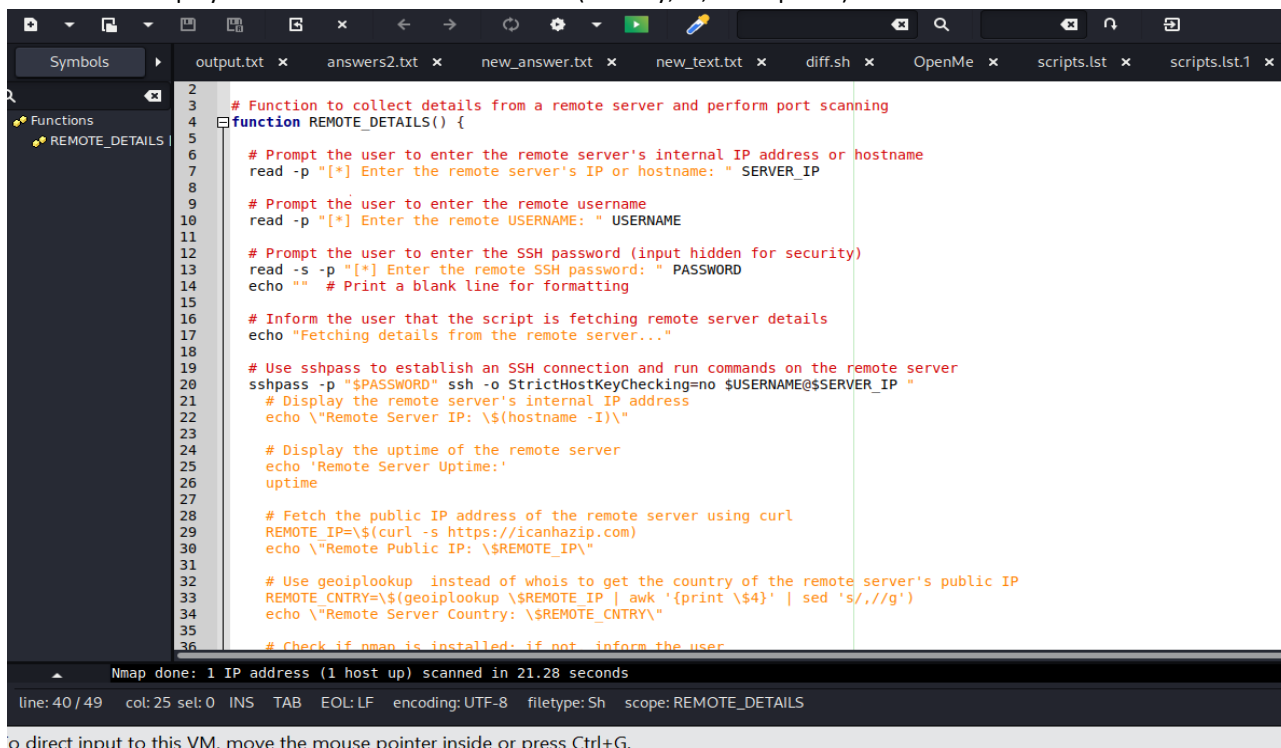


1.5 Allow the user to specify the address to scan via remote server; save into a variable.

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).



o direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
(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.

```
project.sh - /home/kali/Desktop/Network_Research/Network_research_project -
File Edit Search View Document Project Build Tools Help
output.txt x answers2.txt x new_answer.txt x new_text.txt x diff.sh x OpenMe x scripts.lst x scripts.l...
45 ANONYMOUS
46
47 # Function to collect details from a remote server and perform port scanning
48 function REMOTE_DETAILS() {
49     LOG "Starting remote details collection..."
50
51     read -p "[*] Enter the remote server's IP or hostname: " SERVER_IP
52     read -p "[*] Enter the remote USERNAME: " USERNAME
53     read -s -p "[*] Enter the remote SSH password: " PASSWORD
54     echo "" # Print a blank line for formatting
55
56     LOG AUDIT
57     LOG "Fetching details from the remote server..."
58
59     # Store the output of the SSH session locally and log relevant information
60     sshpass -p "$PASSWORD" ssh -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP << 'EOF' >> remote_output.log
61
62     echo "Remote Server IP: ${hostname -I}"
63     echo "Remote Server Uptime:"
64     uptime
65     REMOTE_IP=$(curl -s https://icanhazip.com)
66
67 Status 2024-10-13 09:28:16 - Starting remote details collection...
68 [*] Enter the remote server's IP or hostname: 192.168.113.128
69 [*] Enter the remote USERNAME: tc
70 [*] Enter the remote SSH password:
71 2024-10-13 09:32:00 - Appending log entry in the audit log file.
72 2024-10-13 09:32:00 - Fetching details from the remote server...
73 Pseudo-terminal will not be allocated because stdin is not a terminal.
74 2024-10-13 09:32:04 - SSH session completed. Fetching remote files...
75 2024-10-13 09:32:05 - Downloaded whois.lst successfully.
76 2024-10-13 09:32:05 - Downloaded nmap_file.lst successfully.
77 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.

```

62     echo "Remote Server IP: $(hostname -I)"
63     echo "Remote Server Uptime:"
64     uptime
65     REMOTE_IP=$(curl -s https://icanhazip.com)
66     echo "Remote Public IP: $REMOTE_IP"
67     whois $REMOTE_IP >> whois.lst
68     REMOTE_CNTRY=$(geotracklookup $REMOTE_IP | awk '{print $4}' | sed 's/,//g')
69     echo "Remote Server Country: $REMOTE_CNTRY"
70
71     if ! command -v nmap &> /dev/null; then
72         echo "[!] nmap is not installed on the remote server."
73     else
74         echo "Scanning open ports on the remote server IP: $(hostname -I)..."
75         nmap -Pn -F $(hostname -I) >> nmap_file.lst
76     fi
77 -EOF
78
79     LOG "SSH session completed. Fetching remote files..."
80
81     # Download files from the remote server
82     sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:whois.lst . \

```

```

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  answers.txt  new_answer.txt  new_text.txt  diff.sh  OpenMe  scripts.l
[+] Functions
  [+] ANONYMOUS [29]
  [+] INSTALL [12]
  [+] LOG [6]
  [+] LOG_AUDIT [91]
  [+] REMOTE_DETAILS [1]
178 LOG "SSH session completed. Fetching remote files..."
179
180 # Download files from the remote server
181 sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:whois.lst . \
182     && LOG "Downloaded whois.lst successfully."
183 sshpass -p "$PASSWORD" scp -o StrictHostKeyChecking=no $USERNAME@$SERVER_IP:nmap_file.lst . \
184     && LOG "Downloaded nmap_file.lst successfully."
185
186 LOG "Remote details collection completed. See remote_output.log for details."
187 }
188
189 # Function to log audit data
190 function LOG_AUDIT() {
191     LOG "Appending log entry in the audit log file."
192     echo "$(date) - Audit log entry" >> "$LOG_FILE"
193 }
194
195 # Call the function to execute the script
196 REMOTE_DETAILS
197
198
Status 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
Compiler [*] Enter the remote SSH password:
Messages 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.
Scribble 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.
Terminal 2024-10-13 09:32:05 - Remote details collection completed. See remote_output.log for details.
(kali@kali) - [~/Desktop/Network_Research/Network_research_project]
$
```