**NETWORK RESEARCH**

**PROJECT - REMOTE CONTROL**

**CFC090423**

**GAN LAI SOON S19**

1. Introduction:

The purpose of this project is to create an automation system that allows cyber units to execute commands from their local devices while having access to a remote server. The project aims to achieve anonymous communication with the remote server and perform automated tasks. This report provides an overview and explanation of the updated script developed for this project.

1. Script Breakdown: Methodologies

2a checking whether all the necessary tool for the project is already installed on the local machine, if not initiate to install with the if, else statements and apt-get install commands

**A screenshot of a computer

Description automatically generated**

**2b Nipe**

**Nipe** is a program that uses the Tor network as the user’s default gateway, routing all traffic on the Tor network, which is often used to provide privacy and anonymity.

**Below is a Screenshot of script running - showing local computer without nipe installed, script is installing nipe and running the rest of the script as intended.**

**A screen shot of a computer

Description automatically generated with medium confidence**

For documentation purposes, I will be using nipe as an example as nipe takes the longest to install and more steps as well.

Here is the screenshot that showcases the script works when nipe is not installed.

The script enabled nipe installation to be carried out, run nipe command and enabled the rest of the script to run as intended.

The nipe command works as intended as shown by the spoofed Ip address and country name.

**Screenshot of script running on local computer having nipe already installed, running scans on ‘8.8.8.8’**

A screen shot of a computer

Description automatically generated with medium confidence

A white background with black text

Description automatically generated

Above ,the script commands to access SSH on the remote machine.

For demonstration purpose, I will focus on nipe.

The screenshot shows the script works when the local machine already has nipe installed; where our ip address has been spoofed with location in Germany.

From above, the local machine has gained access to the remote machine via SSH service and run the whois and nmapscans with the sshpass command

The script runs as intended; where the nmap and whois scans results are saved in the home folder and a log of their scans has been saved in the /var/log.

A screen shot of a computer

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with low confidence

**The above screenshots show the 2 nmap and whois scan results being saved in the home folder.**

**Files are named “whois\_scanresults.txt” and “nmap\_scanresults.txt”**

**Screenshot of the log show the activity of the nmap and whois scans.**

**A screenshot of a computer

Description automatically generated with low confidence**

The log file of the result from the whois and nmap command was saved as “scan\_log.txt

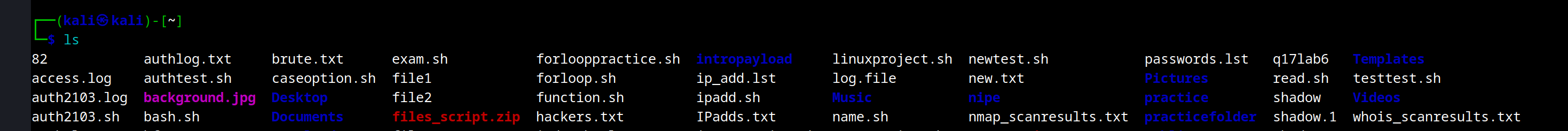
Here the results show the scans from the ip address: 8.8.8.8

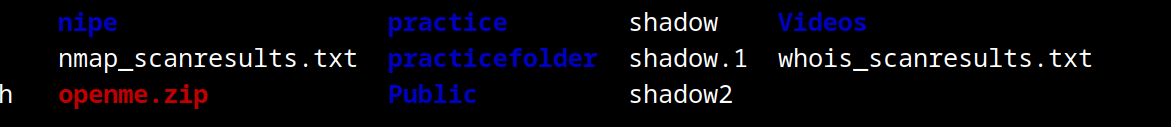
**SCREENSHOTS RUNNING SCANS ON ‘SCANME.NMAP.COM’**

A picture containing text, screenshot, font

Description automatically generated

The screenshot above shows the results using the domain: scanme.nmap.com.





The results of the whois and nmap scans have been saved in the /home/kali folder.

A screenshot of a computer

Description automatically generated with medium confidence

Nmap scan of scanme.nmap.com results saved in a file.

A screenshot of a computer error

Description automatically generated with medium confidence

Whois scans of scanme.nmap.com results saved in a file.

A screen shot of a computer program

Description automatically generated with low confidence

Log file ‘scan\_log.txt’ of the nmap and whois scan of the scanme.nmap.com saved in the /var/log folder

A picture containing text, screenshot, font

Description automatically generated

Here are the results from the log file.

**SCREENSHOT RUNNING SCANS ON ‘8.8.8.8’**

A screenshot of a computer screen

Description automatically generated with medium confidence

For submission purposes: I will be using Ip address 8.8.8.8 to be scanned.

Just like the above , the scans from both whois and Nmap have been saved in the home folder

and the log file has been saved in the /var/log/ folder.

A picture containing text, screenshot, software, font

Description automatically generated

Results of the whois scan of 8.8.8.8 shown in the saved file

A screenshot of a computer

Description automatically generated

Results of nmap scan of 8.8.8.8 shown in the saved file

A screenshot of a computer program

Description automatically generated with medium confidence

Results of the nmap and whois scan of 8.8.8.8 shown in a log file

**Final Screenshot with the script running and better presentation of the displayed information.**

A screen shot of a computer

Description automatically generated with medium confidence

**3)Using function to create a log for the data collection process of the nmap and whois commands executed.**

A screenshot of a computer program

Description automatically generated with low confidence

A picture containing text, font, screenshot, white

Description automatically generated

It took quite some time to figure and experiment before getting it working. The line above sets the ‘log\_file” to the state path /var/log/scan\_log.txt.

The ‘log\_data()’ defines the function named as ‘log\_data’ . It allows us to group a set of commands within it and use it again and again if required.

The script here:

sudo echo "$(date): $1" >> "$log\_file"

the date command was used within the function combined with $1 with is an argument to provide input to the function to perform in this case the scanning activities of the nmap and whois command and saved the data to the log file specified by the ‘$log\_file” variable.

References used:

1) <https://ryanstutorials.net/bash-scripting-tutorial/bash-functions.php>

2) <https://www.tutorialkart.com/bash-shell-scripting/write-output-of-bash-command-to-log-file-example/#gsc.tab=0>

3) <https://www.cyberciti.biz/faq/linux-display-date-and-time/#:~:text=To%20display%20current%20date%20and,time%20as%20root%20user%20too>.

4) <https://www.baeldung.com/linux/use-command-line-arguments-in-bash-script>

5) <https://linuxhint.com/read-command-line-arguments-bash/>

6) <https://www.geeksforgeeks.org/bash-script-how-to-use-command-line-arguments/>

Conclusion

This network project providing a deep study into the network communication protocols through scripting and SSH exploration has been an insightful journey

Scripting exposed me to the intricacies of network automation, highlighting the importance of scalability and efficiency. It also deepened my understanding of syntax and the impact of flags. I gained appreciation for the technical details often hidden behind user-friendly applications, realizing their significance in network operations.

Exploring SSH revealed its robust security features and its ability to strengthen the CIA triad – Confidentiality, Integrity, and Availability.

The project opened my eyes to the intricate processes behind everyday network interactions, showcasing the complex orchestration between client and server guided by crucial protocols.

This hands-on experience has enhanced my understanding of network protocols and security, solidifying the essential role they play in our digital world.