Project: Remote Control

Network Research module.

Ace Tools

Presented by

Eithan Saragosti

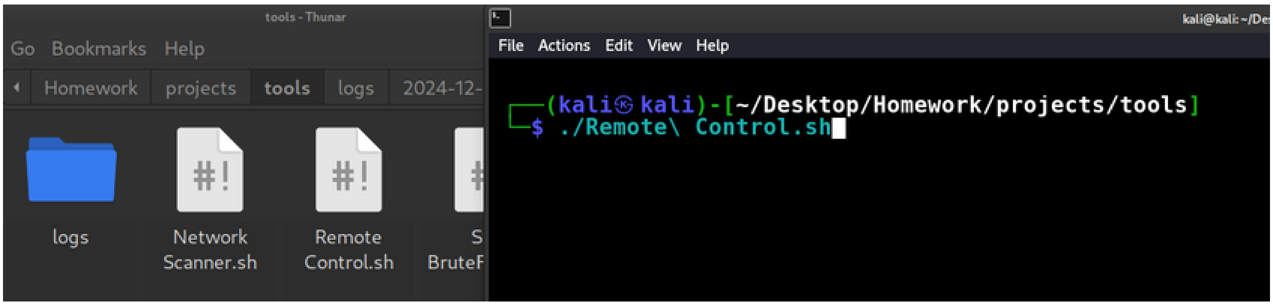
Table of Contents

Inrto:

The script is part of a network research project that allows you to check the anonymity of your network connection, install necessary tools, connect to a remote server via SSH, and perform various network reconnaissance tasks, including port scanning, Whois lookups, and collecting data. The script is intended to run in Kali Linux, using Tor to ensure anonymity during the scanning and reconnaissance operations.

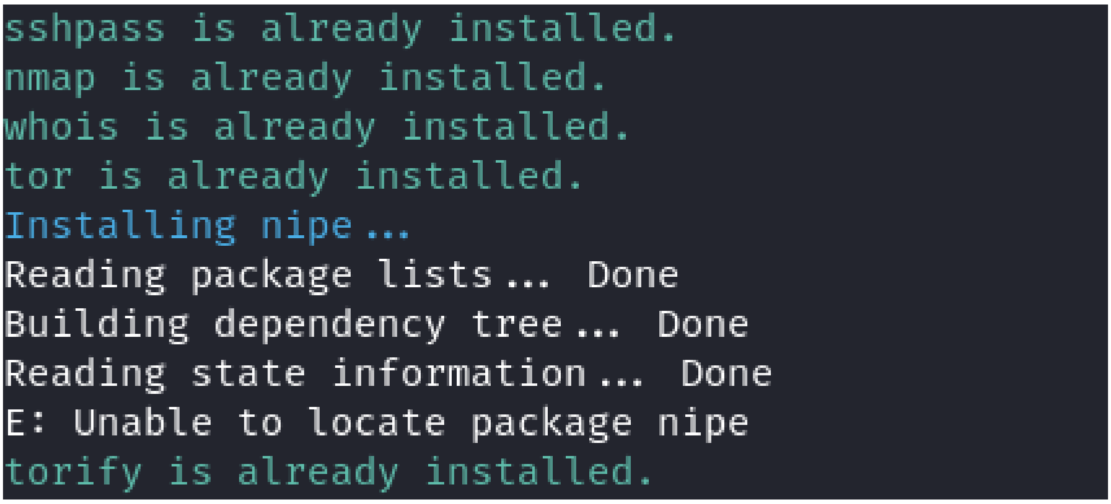
Demo of the Project:

1. I will execute my script by typing "./Remote Control.sh"



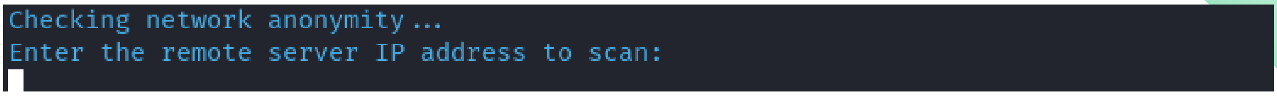
```
┌──(kali㉿kali)-[~/Desktop/Homework/projects/tools]
└─$ ./Remote\ Control.sh
```

2. You are greeted with a Welcome message and the script makes sure you have the needed tools installed, If a tool is not installed it will automatically will install it.



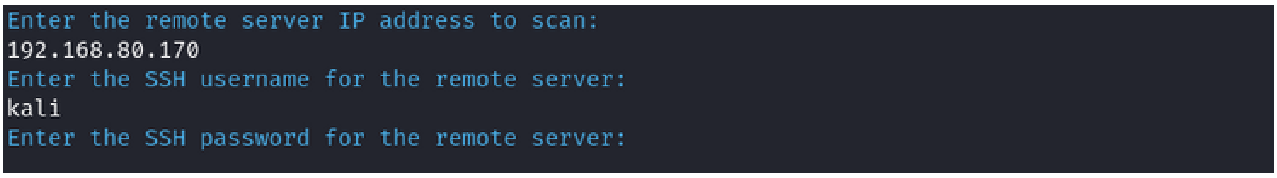
```
sshpass is already installed.
nmap is already installed.
whois is already installed.
tor is already installed.
Installing nipe...
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package nipe
torify is already installed.
```

3. After makeing sure we have all the needed apps install the next step is to check for anonymity, as you can see we are succesfully anonymus with a spoofed IP “103.251.167.20”



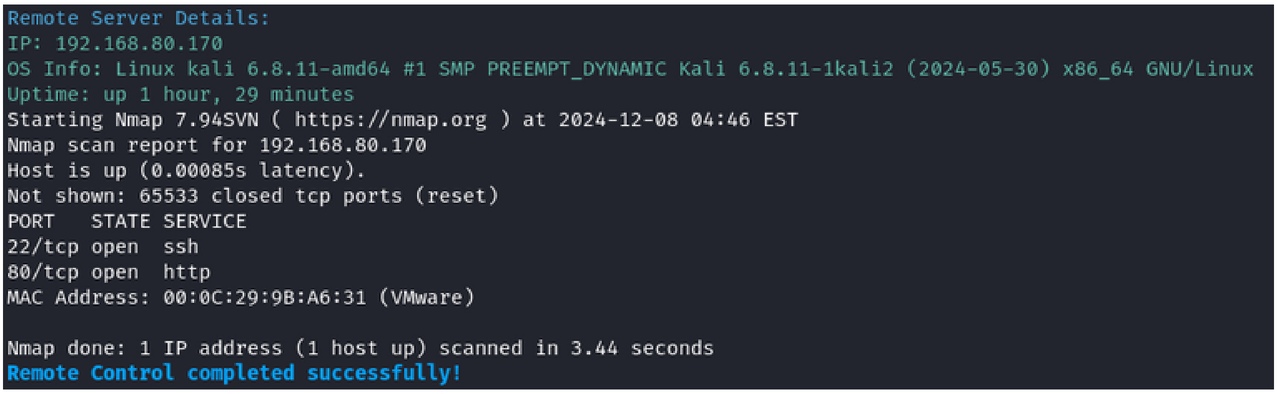
```
Checking network anonymity...
Enter the remote server IP address to scan:
```

4.Now we will add the victim’s IP, SSH username and SSH password
“192.168.80.168” --> “kali” --> “kali”



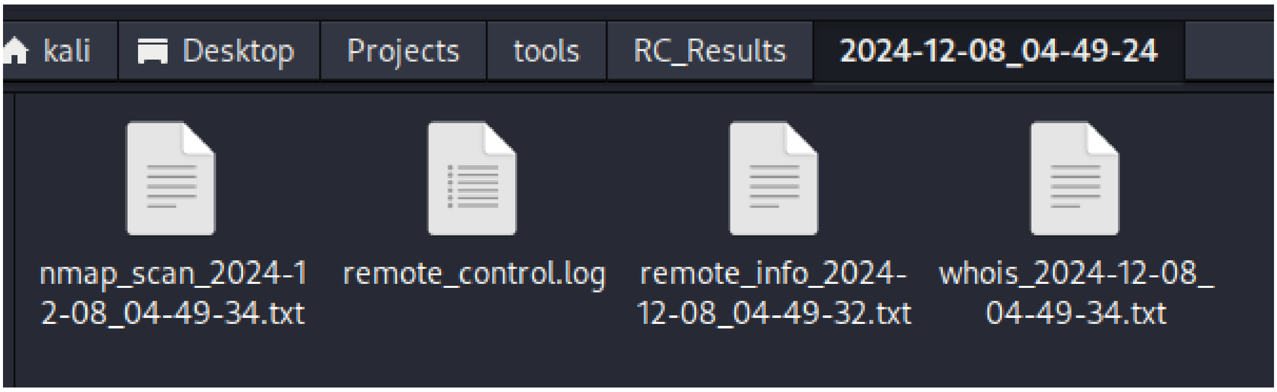
```
Enter the remote server IP address to scan:
192.168.80.170
Enter the SSH username for the remote server:
kali
Enter the SSH password for the remote server:
```

5. The Script will show the server details (IP\Uptime) and will preform the whois and nmap searches.

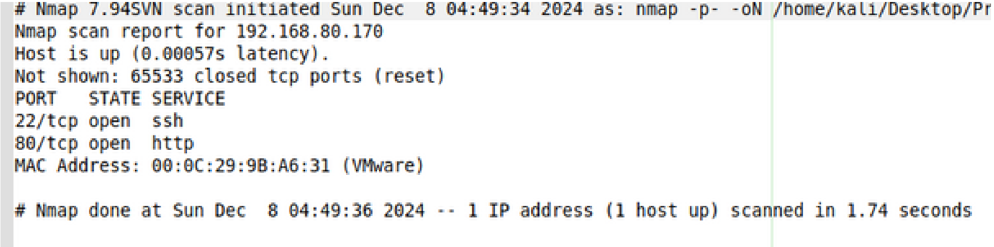


```
Remote Server Details:
IP: 192.168.80.170
OS Info: Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64 GNU/Linux
Uptime: up 1 hour, 29 minutes
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-08 04:46 EST
Nmap scan report for 192.168.80.170
Host is up (0.00085s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:9B:A6:31 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 3.44 seconds
Remote Control completed successfully!
```

6. The full results will be logged and saved in the directory.

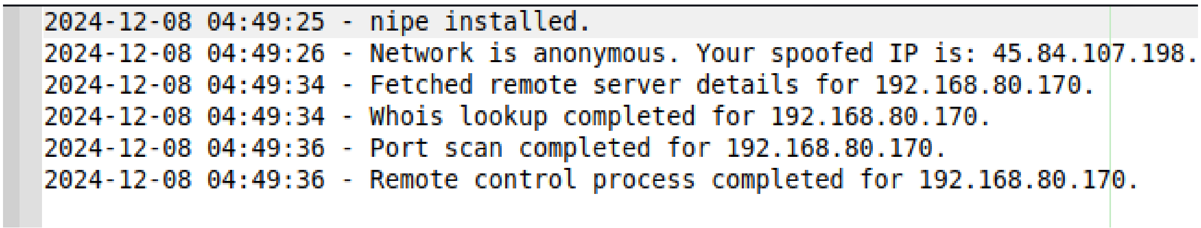

kali
Desktop
Projects
tools
RC_Results
2024-12-08_04-49-24
nmap_scan_2024-12-08_04-49-34.txt
remote_control.log
remote_info_2024-12-08_04-49-32.txt
whois_2024-12-08_04-49-34.txt

7.An exemple of the Nmap_scan:



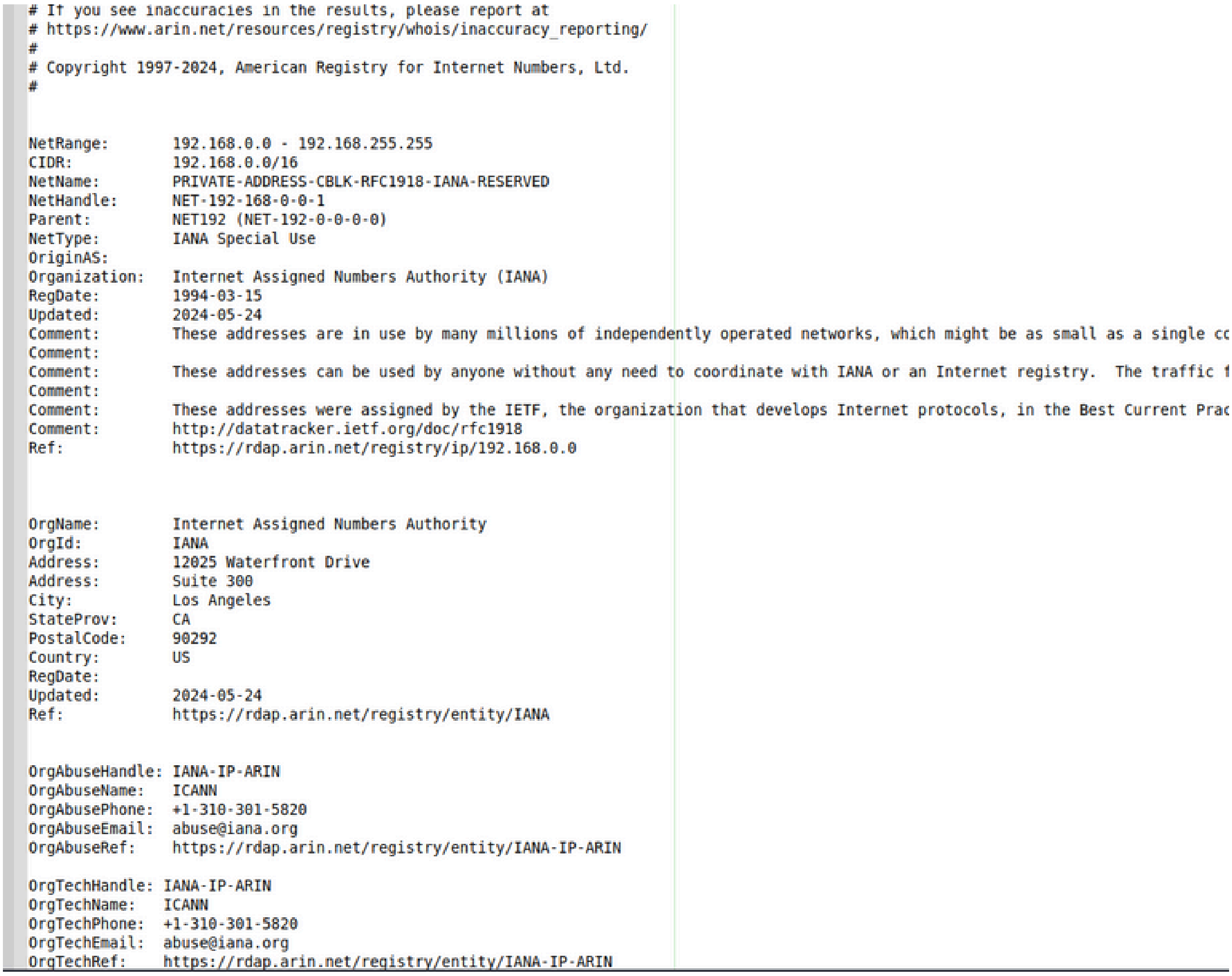
```
# Nmap 7.94SVN scan initiated Sun Dec  8 04:49:34 2024 as: nmap -p- -oN /home/kali/Desktop/Pr
Nmap scan report for 192.168.80.170
Host is up (0.00057s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:9B:A6:31 (VMware)

# Nmap done at Sun Dec  8 04:49:36 2024 -- 1 IP address (1 host up) scanned in 1.74 seconds
```

8.An exemple of the Log file:



```
2024-12-08 04:49:25 - nipe installed.
2024-12-08 04:49:26 - Network is anonymous. Your spoofed IP is: 45.84.107.198.
2024-12-08 04:49:34 - Fetched remote server details for 192.168.80.170.
2024-12-08 04:49:34 - Whois lookup completed for 192.168.80.170.
2024-12-08 04:49:36 - Port scan completed for 192.168.80.170.
2024-12-08 04:49:36 - Remote control process completed for 192.168.80.170.
```

9.An exemple of the whois analysis:



```
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2024, 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:        2024-05-24
Comment:        These addresses are in use by many millions of independently operated networks, which might be as small as a single cc
Comment:
Comment:        These addresses can be used by anyone without any need to coordinate with IANA or an Internet registry.  The traffic f
Comment:
Comment:        These addresses were assigned by the IETF, the organization that develops Internet protocols, in the Best Current Prac
Comment:        http://datatracker.ietf.org/doc/rfc1918
Ref:            https://rdap.arin.net/registry/ip/192.168.0.0



OrgName:        Internet Assigned Numbers Authority
OrgId:          IANA
Address:        12025 Waterfront Drive
Address:        Suite 300
City:           Los Angeles
StateProv:      CA
PostalCode:     90292
Country:        US
RegDate:
Updated:        2024-05-24
Ref:            https://rdap.arin.net/registry/entity/IANA


OrgAbuseHandle: IANA-IP-ARIN
OrgAbuseName:   ICANN
OrgAbusePhone:  +1-310-301-5820
OrgAbuseEmail:  abuse@iana.org
OrgAbuseRef:    https://rdap.arin.net/registry/entity/IANA-IP-ARIN

OrgTechHandle: IANA-IP-ARIN
OrgTechName:   ICANN
OrgTechPhone:  +1-310-301-5820
OrgTechEmail:  abuse@iana.org
OrgTechRef:    https://rdap.arin.net/registry/entity/IANA-IP-ARIN
```

SCRIPT OVERVIEW

VARIABLES:

- VERSION: Defines the version of the script.
- SCRIPT_DIR: The directory where the script is located.
- TIMESTAMP: A time-stamped variable used to create unique log folders.
- LOG_DIR: The directory where logs will be saved, created based on the current timestamp.
- LOG_FILE: The path to the main log file for storing the script's output.
- CONFIG_FILE: The path to the configuration file for persistent settings.

COLOURS AND FORMATTING:

- Colours are defined for terminal output, including red (RED), green (GREEN), cyan (CYAN), yellow (YELLOW), blue (BLUE), and bold text (BOLD). The RESET variable is used to reset the text formatting to the default state.

Tools & Technologies Used:

TOR (THE ONION ROUTER):

- **Purpose:** Tor is a free, open-source software for enabling anonymous communication over the internet. It aims to protect users' privacy by routing internet traffic through multiple layers of encryption (hence "onion routing").
- **Usage in the Script:** The script utilizes Tor to anonymize the user's network connection. This ensures that when performing tasks such as Whois lookups or port scanning, the user's real IP address is not exposed, protecting their identity.
- **How it Works:** Tor routes internet traffic through a network of volunteer-run servers (relays). Each relay only knows the previous and next relays, making it difficult to trace the user's original location.

NIPE:

- **Purpose:** Nipe is a tool that simplifies the process of routing all network traffic through Tor. It allows users to configure their system to route all traffic through the Tor network.
- **Usage in the Script:** Nipe is used to ensure that all network requests made by the script are routed through the Tor network, maintaining anonymity throughout the process.
- **How it Works:** Nipe works by modifying iptables (Linux's firewall) to route traffic through the Tor network. Once activated, it ensures that all applications on the system use Tor by default.

SSHPASS:

- **Purpose:** ***sshpass*** is a command-line tool used to provide a password to SSH (Secure Shell) in non-interactive environments. It allows for the automation of SSH connections without manually entering a password.
- **Usage in the Script:** The script uses ***sshpass*** to connect to a remote server over SSH, providing the username and password for authentication.
- **How it Works:** ***sshpass*** takes the password as an argument or from a file and feeds it to the SSH command, allowing for automation of SSH login without manual password entry.

NMAP:

- **Purpose:** ***nmap*** (Network Mapper) is an open-source tool used for network discovery and security auditing. It is commonly used to identify active devices on a network, scan for open ports, and detect vulnerabilities.

- **Usage in the Script:** The script uses ***nmap*** to scan a remote server for open ports. This helps identify which services are exposed to the internet, potentially revealing vulnerabilities.

- **How it Works:** ***nmap*** sends packets to a target machine and analyzes the responses to determine which ports are open and which services are running on those ports. It can be used to perform various types of scans, such as SYN scans (stealth scanning), full port scans, and service version detection.

WHOIS:

- **Purpose:** The ***whois*** command-line tool is used to query databases that store registered domain name information. This provides details about the ownership, registration date, and contact information of domains and IP addresses.

- **Usage in the Script:** The script uses ***whois*** to gather information about a specified domain or IP address, such as its registrar, the country where it is registered, and other relevant data.

- **How it Works:** When queried, ***whois*** sends a request to a ***Whois*** server, which returns the registration data for the domain or IP address in question. The results can be used to identify the owner and location of a domain, helping with reconnaissance and research.

WHOIS:

- **Purpose:** The ***whois*** command-line tool is used to query databases that store registered domain name information. This provides details about the ownership, registration date, and contact information of domains and IP addresses.

- **Usage in the Script:** The script uses ***whois*** to gather information about a specified domain or IP address, such as its registrar, the country where it is registered, and other relevant data.

- **How it Works:** When queried, ***whois*** sends a request to a ***Whois*** server, which returns the registration data for the domain or IP address in question. The results can be used to identify the owner and location of a domain, helping with reconnaissance and research.

CURL:

- **Purpose:** ***curl*** is a command-line tool used to transfer data from or to a server using various network protocols (such as HTTP, FTP, etc.). It is widely used to interact with APIs and web services.

- **Usage in the Script:** ***curl*** is used in the script to check the status of the Tor connection by querying check.torproject.org/api/ip, as well as to gather server details like geolocation and uptime.

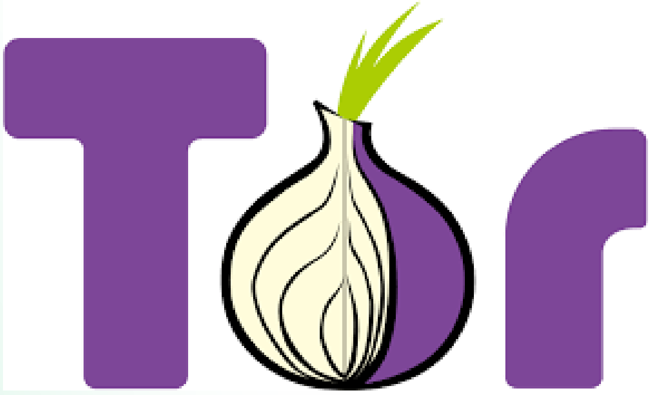
- **How it Works:** ***curl*** sends HTTP requests to specified URLs and retrieves the response. In the script, it is used to interact with APIs to gather information such as the public IP address or server uptime.

TORIFY:

- **Purpose:** ***torify*** is a command-line tool that wraps network commands to route traffic through the Tor network.

- **Usage in the Script:** The script uses torify to ensure that requests made through curl are anonymized by routing them through the Tor network. This ensures that the server queries are made from a spoofed IP address rather than the user’s real IP.

- **How it Works:** By using torify, any command (such as curl, wget, or others) can be routed through the Tor network. It sets up the necessary environment variables and modifies the system’s routing to ensure anonymity.



Summary of Tools and Their Roles in the Script

Tool	Purpose	Role in the Script
Tor	Provides anonymity by routing traffic through a distributed network.	Ensures the user's network connection is anonymized.
Nipe	Routes all system traffic through Tor.	Configures the system to use Tor for all network traffic.
sshpass	Allows automated SSH login by providing a password non-interactively.	Facilitates SSH login to remote servers for executing commands.
nmap	Scans networks for open ports and vulnerabilities.	Performs a network scan to identify open ports on remote servers.
whois	Queries domain registration information.	Retrieves Whois data for remote servers or domains.
curl	Transfers data to/from a server using network protocols.	Used to check the status of Tor and gather server information.
torify	Routes network commands through the Tor network.	Ensures all network commands are anonymized via Tor.

Helper Functions:

log_message():

- Logs messages to the LOG_FILE with a timestamp. This is useful for debugging and auditing the script's operations.

check_root():

- Checks if the script is being run as the root user. This is important because many network-related commands require root privileges.

install_apps():

- Installs necessary applications (such as sshpass, nmap, whois, tor, nipe, and torify) if they are not already installed on the system. This ensures that the environment is ready for the script's operations.

start_tor():

- Starts the Tor service if it's not running. The script requires Tor to ensure anonymity, and this function ensures Tor is active before proceeding with any operations.

check_anonymity():

- Uses the check.torproject.org/api/ip endpoint to check if the network connection is anonymized via Tor. If Tor is running, it fetches the public IP address and checks if the "IsTor": true field is present in the response, confirming anonymity.

get_remote_info():

- Connects to a remote server via SSH and retrieves its details (IP address, country, uptime) using curl and uptime. The server's country information can be fetched using the geolocation-db.com API.

scan_remote_ports():

- Scans the remote server for open ports using nmap. This helps to identify potential vulnerabilities in the server.

perform_whois():

- Performs a Whois lookup for a specified address, retrieving domain registration information, which can provide insights into the owner and location of the domain.

Main Script Logic:

Step-by-Step Process:

- **Install Required Applications:** The script checks if the necessary applications (sshpass, nmap, whois, tor, etc.) are installed. If any of these are missing, the script installs them automatically.

- **Network Anonymity Check:** The script checks if the network connection is anonymized using Tor. If Tor is not running, it starts the Tor service. It then verifies the anonymity by checking the public IP address using torify and the check.torproject.org/api/ip endpoint. If the connection is anonymous, the script confirms the spoofed IP address.

- **User Input for Remote Server Information:** After verifying the network anonymity, the script asks the user to input the remote server's IP address, SSH username, and SSH password. This is necessary to connect to the server via SSH and perform operations on it.

- **Remote Server Details:** The script fetches and displays information about the remote server, including its IP address, country, and uptime. This is done by sending a curl request to a geolocation service and running the uptime command on the remote server using SSH.

- **Perform Whois Lookup:** A Whois lookup is performed on the remote server's address using the whois command. This gives the user insights into the domain registration information (e.g., the registrant, country, etc.).

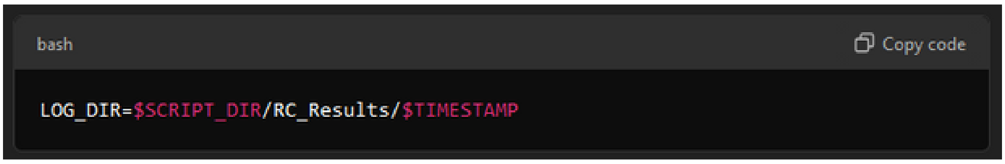
- **Port Scan:** The script performs a full port scan on the remote server using nmap. It saves the results in a file within the LOG_DIR folder for later analysis.

- **Saving Results:** The Whois and Nmap scan results are saved in log files (whois_ and nmap_scan_) within the LOG_DIR directory. This allows the user to review the collected data after the script finishes running.

- **Completion:** Once all the tasks are completed, the script logs the success message, thanks the user for using the tool, and ends the process.

Log Directory and File Structure:

- Path Definition: The LOG_DIR is dynamically created using the script's directory and a timestamp. The timestamp ensures that each execution of the script creates a unique directory, preventing any overwriting of previous logs.



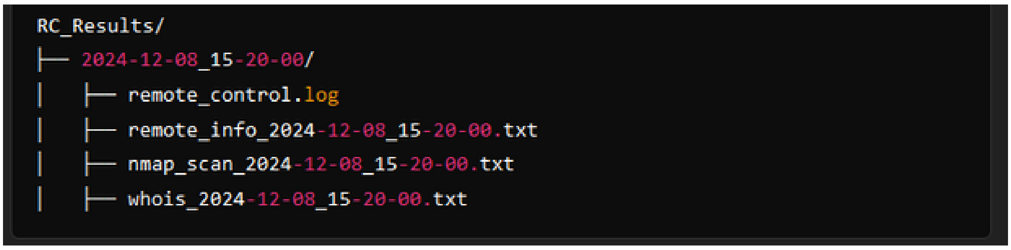
```bash
LOG_DIR=$SCRIPT_DIR/RC_Results/$TIMESTAMP
```

- $SCRIPT_DIR: This is the directory where the script is located. It serves as the root for organizing logs.

- RC_Results: This is a subdirectory within the script directory where all logs are stored.

- $TIMESTAMP: The current date and time (formatted as YYYY-MM-DD_HH-MM-SS) is appended to ensure that each log session gets a unique directory.

- Log Directory Structure: Within the RC_Results folder, each execution of the script creates a new subdirectory, based on the timestamp. This directory will contain all the logs related to that particular session, ensuring that the logs from different runs do not get mixed up.



```
RC_Results/
├── 2024-12-08_15-20-00/
│   ├── remote_control.log
│   ├── remote_info_2024-12-08_15-20-00.txt
│   ├── nmap_scan_2024-12-08_15-20-00.txt
│   ├── whois_2024-12-08_15-20-00.txt
```

Credits:

ChatGPT – https://chatgpt.com

Reddit - r/Kalilinux. - https://www.reddit.com/r/Kalilinux/

Github - https://github.com/

Nmap - https://nmap.org/

Ace Tools

Questions?
Contact Me.

in https://www.linkedin.com/in/eithan-saragosti/

https://github.com/Eithan200/AceTools

✉ Eithan200@gmail.com