TOR CLI Routing Project Lab Guide

This lab will walk you through setting up a basic client-server application using the Tor network, and installing tools to monitor and verify the system configuration. You will work with tools such as Python, the Tor service, Nyx for monitoring, and network verification tools to ensure your system is routing traffic through Tor.

Lab Requirements

- A system running **Kali Linux** or another **Debian-based Linux distribution**.
- **Root** access (sudo) to install and configure software.

Useful paths:

- `/etc/tor/torrc`

Step 1: Install Necessary Dependencies

1. Open a terminal and update your system:

   ```
   sudo apt update
   ```

2. Install **Python 3** and **pip**:

   ```
   sudo apt install python3 python3-pip
   ```

Step 2: Set Up a Python Virtual Environment

1. Navigate to your project directory:

   ```
   cd ./TOR_CLI-Routing-Project/TOR_CLI-Routing-Presentation/
   ```

2. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate
```

Note: If you face issues with sourcing the virtual environment, specify the full path: `./venv/bin/python ./venv/bin/pip`

3. Install the required Python packages from `requirements.txt`

This installs the necessary Python libraries like `PySocks` , `pycryptodome` , and `stem` .

Step 3: Install and Configure the Tor Service

1. Install the **Tor** service:

```
sudo apt install tor
```

2. Start and Ebale the **Tor service**

3. Verify that **Tor** is running:

{ provide a screenshot }

Step 4: Configure Tor for Control Access

1. Open the **Tor configuration file**

Create a backup of the file and Add or uncomment the line to enable the **ControlPort**

2. **Restart the Tor service** to apply the changes:

```
sudo systemctl restart tor.service
```

{ submit diff of the file }

Step 5: Verify the Tor Setup

1. To verify if your system is routing traffic through Tor, use **curl**:

Use https://checkip.amazonaws.com to receive your public IP, with curl

Perform curl with and without redirecting traffic through tor.

{ submit a diff of the outputs }

You might need to use --socks5 127.0.0.1:9050

Step 6: Run the Client-Server Application

1. **Run the server** script on one machine (this will listen for incoming connections):

   ```
   python server.py
   ```

2. **Run the client** script on another machine (this will send message to the server via Tor):

   ```
   python client.py
   ```

Note: you need another machine to be able to run a client and a server, and you need to be able to forward port 25000 on the router to be able to run the server.

3. Send a message from client to the server

{ screenshot route that is shown on the client.py }

Step 7: Install Nyx for Tor Monitoring

1. Install **Nyx** (a tool for monitoring Tor circuits):

   ```
   sudo apt install nyx
   ```

2. Launch **Nyx** to visualize the Tor circuit status:

   ```
   nyx
   ```

Use GETINFO circuit-status and /info “sendos”(as an example router).

To show more information about the circuit that was used when you send a message from client.py

3. In Nyx, run the following command to check the status of your Tor circuit:

```
GETINFO circuit-status
```

4. To view information about your Tor instance:

```
/info <intrance name>
```

{ provide a screenshot of intances/routers information that were used in the circuit }