

Final Report (Module: Onion)

Documentation:

- **Dependencies:**

Dependencies are listed in requirements.txt in the root directory of the repository and can be installed through `pip install -r requirements.txt`

- **Install instructions:**

1. Make sure that the dependencies are installed using the command above
2. Use python 2.7
3. Make sure the RPS module is running and listening on the port mentioned in config.ini. For the mock verification, mock RPS Server can be started by `python rpsServer/main.py`
4. Make sure the Onion Authentication module is running and listening on the port mentioned in config.ini. For the mock verification, mock Onion Authentication Server can be started by `python onionAuthServer/main.py`
5. `cd src`
`python main.py -c ../config.ini`
6. The Service will start listening for the UI Module connection. Mock UI Server can be run by `python uiServer/main.py`
7. You can send sample commands to the Onion Module like build tunnel, send data using this UI mock server

- **Known Issues:**

1. Tunnel Destroy does not work at the moment.
2. Once RPS or UI disconnect, the service does not try to rebuild the connections.

Deviations from the Initial Design:

1. UDP design has been scrapped in favor of complete TCP architecture since it was adding a lot of overhead per packet.
2. We haven't used a purely multithreaded architecture
3. Tunnel Error has not been implemented

Future Enhancements:

1. Better Error handling
2. Better throughput with multi threaded architecture

Division of Work:

1. Mock Servers were implemented by Zain
2. P2P protocol was implemented by Furqan.
3. API connection to the other modules and integration was performed together.

Effort Spent on the Project:

About 100 hours were spent by each team member. We would usually work together and help each other out.