ELEC-H417 - Project Assignment

version 1.0

Denis Verstraeten, Wilson Daubry
{denis.verstraeten, wilson.daubry}@ulb.be

November 27, 2022

1 Introduction

The goal of this project is to design and implement a TOR network that enables anonymous usage of a network. It will be an opportunity to have a practical understanding of different concepts in cryptography and networking studied in the theoretical classes.

Using your TOR network, you should be able to anonymously send any request to some server. As a second challenge which elaborates on top of the TOR network, you should implement a centralized server which will accommodate a *challenge-response* authentication scheme for a client.

We ask you to be creative as well, and to implement any features that feels funny and/or interesting to you. Feel free to go any directions, as long as you match the requirements.

2 Requirements

2.1 TOR Network

The TOR protocol was introduced in the 90's by a US Army research project and enabled circuit-based anonymous connections. It relies on a pool of relays that connects a client to its destination. Each request coming from a client will go through several hops within this pool and will be subject to different layers of encryption. Your project should fulfill the following properties.

Peer-to-Peer Network The nodes in the pool should be connected following a Peer-to-Peer architecture. You should design a protocol that can support an arbitrary large number of peers and listen for new ones which would try to join. We suggest that you run these peers locally on your computer for development purpose.

Anonymous Connection from a Client A client should be able to send requests to and receive answers from a destination address through the TOR pool. The source address should be anonymous for any observer eavesdropping the network. At this point, the destination can be anything, from an actual webpage, to a simple local HTTP server replying with a *Hello world!* HTML file.

2.2 Anonymous Challenge-Response Authentication

You should implement a server running a *challenge-response* based authentication protocol. This should allow a client to successfully authenticate to the server using a password, without leaking it. Your client should be able to contact the authentication server through the TOR network to achieve anonymous authentication.

2.3 Remarks

You should develop your project using Python. Moreover, if something is not specifically mentioned in this assignment, it means that you are free to take any decision concerning that point. You are allowed to use libraries to do this project. For some parts like cryptography and networking, you even should use them, since it is a good practice. Do not reinvent the wheel. However, it is forbidden to find a off-the-shelf TOR library and just instantiate it, or to copy-paste the source code of a TOR protocol. You can copy-paste code coming from online forums, but if you do so, please specify it clearly in the comments of your code for the sake of intellectual honesty.

Before diving into the coding, take some time to think about the architecture of your project, as well as about a suitable communication protocol. This will save you time later as it will dramatically reduce the time you will need to debug. You will need to come up with a mechanism to make the IP address of your TOR gateway and of your authentication server available to your clients, you have the choice on how you want to design and implement this, as long as it is explained and documented.

3 Deliverable

You should deliver your code as well as a report. Both need to be submitted via git, as explained in Section 3.3. The deadline for the git submission is the **23$^{\textbf{rd}}$ of December at noon** as stated in the `ELEC-H417_Labs_Organization.pdf`.

Three files will have to be present in the repository:

- A `README`
- A `requirements.txt`
- Your report in `pdf` format.

For the rest, the structure is up to you, as long as it is explained.

3.1 Code

Your `README` file should be clear and understandable for someone who has not taken part into your project. It should explain how to run the code, whether libraries need to be installed and/or imported, if so how to do it. It should also state whether there are known compatibility issues. Finally, it should present the different features of the app and how to use them. All of these guidelines are very common and are good practice when you want to post code online. It should not be a report, but a file oriented towards the actual usage of the app.

Your `requirements.txt` file is standard when developing in Python and lists all the libraries needed to run the project. There many resources online explaining how to use this feature.

Your source code should be well structured, clear and easy to read. You should use comments whenever necessary. You should keep in mind the people reading it were not with you when you wrote it.

3.2 Report

In the report, you will have to discuss the following points:

Architecture Explain the architecture of your code, list the decisions that were taken and give a justification if this is relevant. If you want, you can include diagrams.

Innovation and creativity You are encouraged to show some creativity and to go further than the requirements of this assignment. You should discuss your added features in the report.

Challenges List the difficulties faced during the project and how you solved them.

You can choose the length of the report, in the sense that we want you to discuss anything that feels important to you regarding the project. We would suggest a length between 5 to 10 pages.

3.3 Submission

We want you to use the git versioning system throughout your development, as well as for the submission of your deliverables. You can use any git host that you want, such as GitHub or GitLab.

To submit your project, you have to send one e-mail per group to the teaching assistants with a link to your repository, with the following format in the subject: *ELEC-H417_GroupX_Project*, with X being your group number.

3.4 Grading

As said previously, we want you to show some creativity. Therefore, if you fully match the requirements, you will be graded 16/20. The mark will be computed as follows:

$$\text{Grade} = {}^{3}\!/_{4} \cdot \text{Code} + {}^{1}\!/_{4} \cdot \text{Report}$$

$$\text{Code} = {}^{4}\!/_{5} \cdot \text{Required} + {}^{1}\!/_{5} \cdot \text{Originality}$$

$$\text{Report} = {}^{4}\!/_{5} \cdot \text{Required} + {}^{1}\!/_{5} \cdot \text{Originality}.$$