

Lab 0: Introduction to Network Protocols using Python Sockets

Course: FSCT 8561

Instructor: Dr. Maryam R. Aliabadi

Lab Duration: 3 hours

Overview

In Lab0, students are introduced to the fundamentals of network programming using Python's socket library. The lab focuses on understanding basic network protocols and establishing simple client-server communication. Students will explore how sockets enable data exchange over TCP/IP networks, learning to create a basic server that can accept connections and a client that can send messages. Additionally, students will analyze the security implications of network communication.

Learning Objectives

- Understand the role of network protocols in computer communication
- Explain the client–server communication model
- Use Python sockets to establish basic TCP connections
- Analyze how protocol behavior affects security and reliability
- Demonstrate learning through explanation, screenshots, and code annotation

Required Reading

- Mastering Python for Networking and Security – Chapters 1 and 2
 - <https://learning.oreilly.com/library/view/mastering-python-for/9781839217166/>
 - Source code on: <https://github.com/PacktPublishing/Mastering-Python-for-Networking-and-Security-Second-Edition>
- Python Official Documentation: socket module
 - **Official Python Documentation – socket Module**
<https://docs.python.org/3/library/socket.html>
 - **Real Python – Socket Programming in Python**
<https://realpython.com/python-sockets/>
 - **GeeksforGeeks – Socket Programming in Python**
<https://www.geeksforgeeks.org/socket-programming-python/>
 - **DigitalOcean – How To Use Sockets in Python 3**

<https://www.digitalocean.com/community/tutorials/python-socket-programming>

Required Tools

- Python 3.9 or later
- Operating System: Windows, macOS, or Linux
- Code editor (VS Code recommended)
- Command-line access (Terminal or PowerShell)

Pre-Lab Preparation (Mandatory)

Before doing the lab, students must:

1. Read Chapters 1 and 2 of the textbook
2. Review the Python socket tutorial provided in this lab
3. Ensure Python is installed and accessible via command line

Lab Tasks

Task 1: Environment Verification

- Verify Python installation using `python --version`
- Import the `socket` module without errors

Task 2: Understanding Sockets

- Create a simple Python script that imports `socket`
- Identify and explain the purpose of `socket.AF_INET` and `socket.SOCK_STREAM`

Task 3: Simple Client Connection

- Write a Python script that creates a TCP socket
- Connect to a known public server (example: example.com on port 80)
- Send a simple HTTP GET request and print the response
- Capture and explain the connection behavior

Task 4: Create a TCP Server

Write a Python script that:

1. Binds to localhost on a chosen port (e.g., 12345)
2. Listens for incoming connections
3. Accepts a connection and receives a message from the client
4. Sends a response back

Task 5: Create a TCP Client

Write a Python script that:

1. Connects to your server on localhost:12345
2. Sends a custom message
3. Receives and prints the server's response

Task 5: Security Reflection

- Identify at least **three security risks** related to raw socket programming.
- Explain why **input validation, access control, and protocol awareness** matter at the network level.
- Suggest ways to **secure your simple client-server application**

Deliverables

Students must submit a single PDF file containing:

1. Answers to reflection questions
2. Annotated Python code snippets
3. Screenshots showing successful execution
4. A short paragraph explaining what was learned and what was unclear
5. submit your report with the following name format through the Learning Hub.

Filename: Lab0-FirstName-Lastname-StdNo.PDF

Good luck!