Week Two

Midterm Exam

Matthew Semple

National University

Course code: CYB333 Security Automation

Bruce Tukuafu

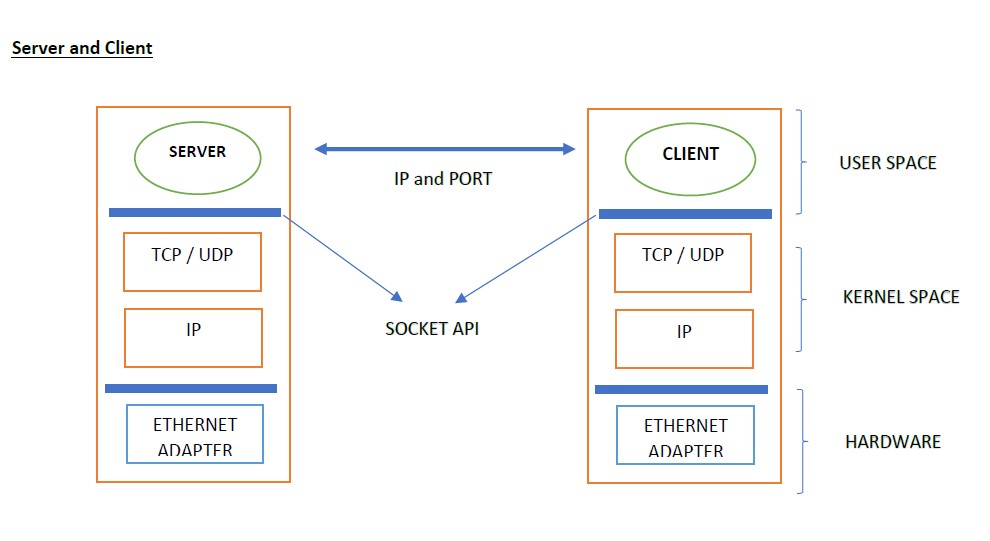
April 20, 2025

.

I conducted research into port-scanning tools that have been used to enhance security. Sockets are the building blocks for mechanisms that allow inter-device message transmission over network lines. This midterm lab will demonstrate the socket module through the setup and functioning of a straightforward chat server and client application in Python. The server handles communication between several clients, while the chat application provides an interactive environment through which messages can be sent and received.

First, I installed Python 3.x on my machine, which can be acquired from the official website. A beginner may create a new folder for the project and switch into it. I named a file server.py and opened it with a text editor. The imported libraries are socket and threading. I created the connection handling and communication function for each client and received data from the client in an infinite loop in the handle\_client function. If no data came through, the client disconnected, terminating the loop. Otherwise, I decoded the data and reported it on the server console. I crafted a message and sent it back to the client. . It begins by elaborating on the importance of timing functions in measuring the duration of operations. This measurement is critical in security processes like port scanning and penetration testing. Knowing how long something takes to happen will help an attacker hone some of their security-related strategies. The author reviews the advent of Python 3 from Python 2 and recommends using Python 3 for its new features and enhanced security. The video discusses socket connection establishment with Python in detail. It introduces the library for socket programming, clarifies the socket families and types- TCP and UDP. I established the primary purpose for which my server existed. The main function created a socket object, server\_socket, an endpoint for communicating with the server. The socket function inside the module is used to create this object, which accepts two arguments. The socket.AF\_INET signifies the address family for the socket.SOCK\_STREAM, which indicates the type of socket involved, and together make a new socket object with which a TCP communication over IPv4 is formed. The server socket always listens to incoming connections from clients. After being created and configured to exist within TCP communications, it is bound to a specific IP address (127.0.0.1) and port number (197.0.10.5). The server should also be able to accept connections from clients and start a thread to serve that client. The handle\_client function is then called from within that thread to allow the server to serve more than one client at a time. I opened another file named client.py in your text editor. For the server, import all required libraries: socket and threading. I defined the main method of client application. Its primary function creates a socket for the client and connects it via the server's IP (127.0.0.1) and port number (13515). Users can send messages via the sendall method and have them displayed to them. The command python server.py is run on the Terminal to execute the server application. A port scanner is a great tool that can be developed in Python to strengthen security testing. I installed Python 3, which comes with many Linux distributions and macOS by default and installed it in Windows. It is a good idea to have a Python integrated development environment, and this lab uses Visual Studio Code and Vim on Linux. Port scanners help discover open ports that could be vulnerable on remote systems. This is a simple Python program to write, but another option is to pull the source code from another developer's project. Many programming projects are available on repository sites such as GitHub. I used code in the following section (*Python TCP Networking Sockets Client/Server*, 2022). A port scanner in Python is a great way to save one’s time. A port scanner in Python is better than the existing tools, such as Nmap language. When looking at the language, it gives further insight into how a simple port scanner works. One adds unique functionality or features to build a tool that works well for that objective. Python is used across various operating systems (Garn, 2024).

Attached are the screenshots for the Midterm Lab



A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A computer code with black text

AI-generated content may be incorrect.A table with text on it

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.A computer screen shot of a computer code

AI-generated content may be incorrect.A computer screen with colorful code

AI-generated content may be incorrect.

**References**

Garn, D. (2024). *How to build a Python port scanner*. Search Security; TechTarget. https://www.techtarget.com/searchsecurity/tutorial/How-to-build-a-Python-port-scanner

*Python TCP Networking Sockets client/server*. (2022, December 10). Discussions on Python.org. https://discuss.python.org/t/python-tcp-networking-sockets-client-server/21811

‌