ECS 152A: Computer Networks

Fall 2025

Project 1

(100 points)

Due Date: (October 31st by 11:59 PM – before midnight)

Team: The project is to be done in a team of at most 2 students. You *cannot* discuss your code/data with other classmates (*except* your project partner)

All submissions (including your code) will be checked for ***plagiarism*** against other submissions as well as the public Internet. Plagiarized submissions will be entitled to ***zero*** points. Generative AI code is not allowed and also entitled to ***zero*** points.

Project 1 consists of three parts:

1. PCAP analysis
2. Implementing Iperf Client
3. Implementing a proxy server

Part 1: PCAP analysis (*50 points*)

This part has two subparts:
Part a: Monitoring live network traffic (20 points)
Part b: Analyzing network traffic in a pcap file (30 points)

Part a: Monitoring live network traffic

In the first part, you will perform certain network activities while running Wireshark in the background. Note that Wireshark captures all the traffic to and from your chosen network interface, including irrelevant packets generated from applications running in the background, network activity caused by the operating system, etc. These irrelevant packets will be mixed with the packets exchanged with the website you are visiting. You should try to minimize it (it is very difficult to completely stop all other network activity) by ensuring that no other application is running in the background and only one browser tab is open while visiting the websites. Also, make sure you are not using any ad blocker or any other such browser extension while visiting these websites.

After performing these actions, you will save the Wireshark capture for the site in a Pcap file (do not use PcapNg). You will analyze this Pcap file with the help of dpkt library. **You will generate a separate Pcap file for each network activity you will perform.**

Along with the report, you will also need to submit the Python code used for analysis and the pcap files you generated for each activity.

You will perform the following activities and capture individual pcaps for each:

1. Ping google.com for 20 packets via command line.
2. Visit https://example.com in your browser.
3. Visit http://httpforever.com in your browser.
4. Visit https://www.tmz.com in your browser.
5. Access an FTP server (Type “ftp ftp.gnu.org” in your terminal)
6. SSH into a CSIF machine(Accessing the CSIF Computers , SSH into CSIF)

Report: proj1_[name1]_[student_id1]_[name2]_[student_id2].pdf
Separate each part in the report clearly.

At the beginning of the page, specify the following:

1. Full name of student 1 (Student ID)
2. Full name of student 2 (Student ID)
3. Name of the Python source codes and Pcap files submitted.

Answer the following questions in your report**.**

1. List the different application layer protocols and their counts for each activity. In your report, specify how you figured out the protocol for each activity.
2. How many HTTP and HTTPS packets did you record while performing activities 2 and 3?
3. List the destination IP address used in each activity along with their timestamps. The destination IP address should be in the IPv4 format like x.x.x.x (e.g., “192.168.1.1”, “8.8.8.8”, “10.0.1.150”, etc.).
4. For activities 2, 3, and 4, can you tell which browser was used for these activities from the captured packets?

Part b: Analyzing Pcap files

In this part, you will be given 3 pcap files. The Pcap files were generated using Wireshark listening over wireless. You can find the files at this link.

The first pcap file (PCAP1_1.pcap) captures multiple requests some of which send secret sensitive information from a client to a server. Your job is to analyze the pcap file and list the secret/secrets that were sent to the server. Note that the presence of the secret will be obvious so you should know it when you see it. You will also answer the question about the first pcap file by writing a Python script that makes use of the dpkt library. Include your code in the submission. And put this output in your report.

The second and third pcap files (PCAP1_2.pcap and PCAP1_3.pcap) capture traffic from a very specific activity. Your job is to figure out the activity performed in the pcap files.

You will add to your report detailing what you think is happening in the two pcaps. Note these pcap files will have miscellaneous packets in them.

Part 2: Implementing iPerf (*25 points*)

iPerf is a network testing tool used to measure throughput (bandwidth) between two endpoints. For this part, you will build a UDP server and client (both hosted on localhost) using the socket API in Python. You will send 100 megabytes of data from the client to the server. The server should measure the throughput (amount of data received / time taken to receive them) and send it back to the client. The client should then print the throughput value received from the server. You will report the computed throughput in kilobytes per second in your report

Code file names:
udp_server[name1]_[student_id1]_[name2]_[student_id2].py
udp_client[name1]_[student_id1]_[name2]_[student_id2].py

Remember to include the names of all programs you submit in your report.

Part 3: Proxy server (25 *points*)

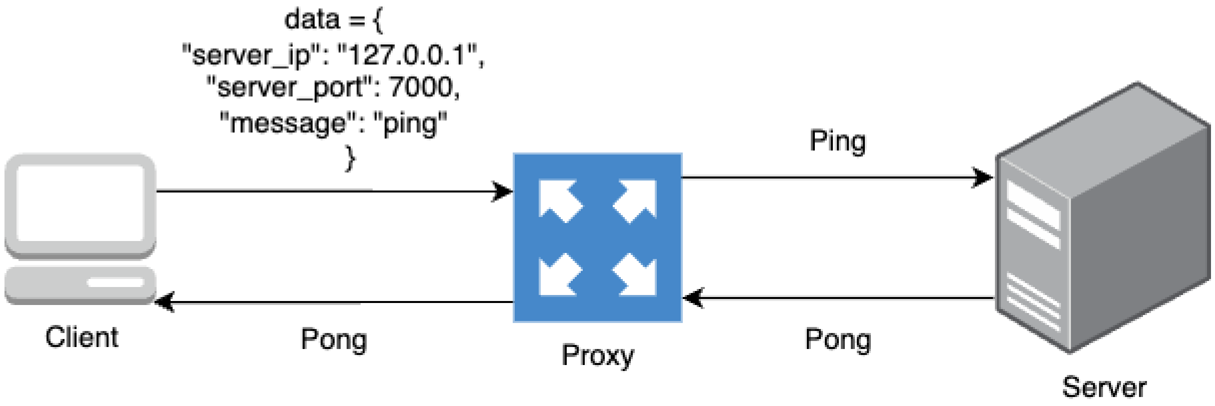
You have to implement ping-pong client servers using **TCP Sockets.** But instead of having the client talk to the server directly, you need to implement a proxy server that forwards requests from the client to the server.

The client should send data to the proxy server in the following JSON format:

```
data = {
"server_ip": "127.0.0.1",  # The server's IP (destination)
"server_port": 7000,       # The server's port (destination)
"message": "ping"          # The actual message
    }
```

The proxy server should extract the server's IP from the message and forward the message to the server. Once the server responds, the proxy forwards the message back to the client.

The proxy server also implements an IP blocklist consisting of several IP addresses. Whenever it finds that the server IP is in the blocklist, it does not forward the request and replies with an "Error" message instead.



You need to implement the client, proxy, and the server.

Code file name:
proxy_server[name1]_[student_id1]_[name2]_[student_id2].py
client[name1]_[student_id1]_[name2]_[student_id2].py
server[name1]_[student_id1]_[name2]_[student_id2].py

Testing Environment:

All submissions will be tested on Python 3+.

Submission Details:

Submit a zipped file with all your python files, your single report, and your generated pcap files to canvas. Only one person from the group submits. Make sure both students' names are in the report.

Late Submission Policy:

No late submissions are allowed. However, if you barely miss the deadline, you can get partial points up to 24 hours. The percentage of points you will lose is given by the equation below. This will give you partial points up to 24 hours after the due date and penalize you less if you narrowly miss the deadline.

$$Total\ Marks\ you\ get\ =\ (Actual\ Marks\ you\ would\ get\ if\ NOT\ late)\ \times\ \left[1\ -\ \frac{hours\ late}{24}\right]$$

Late Submissions (later than 24 hours from the due date) will result in zero points *unless you have our prior permission or documented accommodation*.

Best of luck

Submission Page

Include this signed page with your submission

I certify that all submitted work is my own work. I have completed all of the assignments on my own without assistance from others except as indicated by appropriate citation. I have read and understand the university policy on plagiarism and academic dishonesty. I further understand that official sanctions will be imposed if there is any evidence of academic dishonesty in this work. I certify that the above statements are true.

Team Member 1:

Full Name (Printed) Signature Date

Team Member 2:

Full Name (Printed) Signature Date