ECS 152: Computer Networks

Fall 2023

Project 1

(100 points)

Due Date:

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 Internet including ChatGPT and other such tools. Plagiarized submissions will be entitled to **zero** points.

Project 1 consists of three parts:

1. Pcap analysis
2. Iperf implementation
3. Mitmproxy

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 activity while running wireshark in the background. Note that Wireshark captures all the traffic to and from your chosen network interface, which can include irrelevant packets generated from applications running in the background, network activity caused by the operating system, etc. These irrelevant packets will be mixed with 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 generated for each activity.

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

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

Report: proj1a_[name1]_[student_id1]_[name2]_[student_id2].pdf

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 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 and 3, 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 (ass1_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 files and list all 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. Your code should output each secret in a separate line.

The second and third pcap files (ass1_2.pcap and ass1_3.pcap) capture traffic from a very specific activity with a subtle difference. Your job is to figure out the activity performed in the pcap files and identify what was different when performing the activity across the two pcap files.

You will write a report detailing what you think is happening in the two pcaps. You will also answer the following questions about the pcap files by writing a Python script that makes use of the dpkt library. Submit both the report and the code.

For both parts, you will first implement the packet analysis yourself using dpkt and then get assistance from ChatGPT. In your submission, include your original implementation, a link to your chat session with ChatGPT and your implementation after interacting with ChatGPT. Note that you may have to tweak the implementation suggested by ChatGPT. You will include your final implementation after making the required tweaks.

Report: proj1b_[name1]_[student_id1]_[name2]_[student_id2].pdf

Remember to include the name of your submitted code in the report.

Statistics to report:

1. List the unique source and destination IP addresses do you see in each pcap file?
2. For both pcaps, iterate over the packets and print the packet number, source ip address and destination ip address for each packet. The list you print should be sorted in ascending order of packet number.

```
packet_number, source ip, destination ip
1, 192.168.1.1, 8.8.8.8
…
```

Part 2: iPerf implementation (25 points)

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 kilobytes 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.

You will first implement this yourself and then get assistance from ChatGPT. In your submission, include your original implementation, a link to your chat session with ChatGPT and your implementation after interacting with ChatGPT. Note that you may have to tweak the implementation suggested by ChatGPT. You will include your final implementation after making the required tweaks.

Report: proj2_[name1]_[student_id1]_[name2]_[student_id2].pdf

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

Part 2: Mitmproxy (25 points)

In this part, you will capture packets with and without mitmproxy configured. For this part, you should use the requests library in Python to make network requests.

First, With Wireshark running, you will send a request to the following URL:
https://kartik-labeling-cvpr-0ed3099180c2.herokuapp.com/ecs152a_ass1

In this request, you will add a request header that specifies your student ID as below:

- Student-Id: 123456789 (here replace this with your own student ID)

The server will respond with a secret key that you need to identify in Wireshark. In your report, answer the following questions:

1. Can you tell what the secret key is?

2. If yes, what is it? If not, why so?

Now configure mitmproxy on your machine and make sure it is running (using mitmproxy or mitmweb).

Repeat the process of sending the request with the appropriate header. Now, answer the following questions:

1. Can you tell what the secret key is?
2. If yes, what is it? If not, why so?
3. If your answer to 2 changed from above, what do you think mitmproxy changed?

You will also include screenshots from Wireshark and mitmproxy/mitmweb that show the server response. You can refer to these screenshots in your report to explain how you were / were not able to extract the secret key.

You will first implement code to send the request yourself and then get assistance from ChatGPT. In your submission, include your original implementation, a link to your chat session with ChatGPT and your implementation after interacting with ChatGPT. Note that you may have to tweak the implementation suggested by ChatGPT. You will include your final implementation after making the required tweaks.
You will first implement this yourself and then get assistance from ChatGPT. In your submission, include your original implementation, a link to your chat session with ChatGPT and your implementation after interacting with ChatGPT. Note that you may have to tweak the implementation suggested by ChatGPT. You will include your final implementation after making these tweaks.

Report: proj3_[name1]_[student_id1]_[name2]_[student_id2].pdf

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

Testing Environment:

All submissions will be tested on Python 3+.

Late Submission Policy:

No late submissions are allowed. However, if you barely miss the deadline, you can get partial points upto 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 penalizes you less if you narrowly miss the deadline.

Total marks = (Actual Marks you would get if NOT late) x [1 - hours late/24]

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

Include this signed page along with your submission

Submission Page

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