CS341 Network Lab 2: Network Traffic Analysis Lab

Logistics

- The due date is 11:59 pm on Oct 15.
- Submit your files in a zip format (see Gradescope Submission section) via Gradescope with its name as {SID}_lab2.zip.
- Discussion and Q&A: Lab2 at Classum

Table of Content

Overview

In this lab, you will analyze network traffic captured in a live network environment. Often, we collect network traffic data to understand the behavior of network applications, detect network anomalies, and troubleshoot network problems. This lab will help you understand how to analyze network traffic data and extract useful information from it.

We use one of the most widely used applications we use in our daily lives, Zoom, as a case study. Zoom is a popular video conferencing application that allows multiple participants to join a video conference. In this lab, you will analyze network traffic data captured from Zoom to understand the behavior of participants in a video conference. Through this exercise, you will experience how to analyze network traffic data, extract useful information from it, and infer particular user activities in a video conference.

Note 1.

This lab handout provides quite a number of undocumented details of the Zoom protocol. As these protocol details are not always publicly available, you are encouraged to read the provided information carefully and use it to solve the tasks in this lab. Network traffic analysis always involves some level of guesswork and inference (especially when the protocol details are not publicly available), and this lab is designed to help you develop these skills.

Note 2.

You can find the required documents for this lab in the Google form link provided alongside this lab handout in the Classum announcement. You will be asked to sign and agree to a confidentiality agreement before downloading the necessary files.

Note 3.

When you write a timestamp of a specific packet, you should use the following timestamp format. Mismatches of timestamp format will lead to deduction.

```
1970-01-01 01:02:03.123456
```

- Although the timezone is not described in this format, always assume KST timezone.
- This can be represented as `strftime('%Y-%m-%d %H:%M:%S.%f')` in Python
- If you use Wireshark, we recommend you to set the time display format as “Date and Time of Day”.
    - Wireshark > View > Time Display Format > Date and Time of Day (1970-01-01 01:02:03.123456)

Note 4.

We recommend the following tools for each task.

- Task 1: wireshark
- Task 2: wireshark
- Task 3: wireshark, scapy (with python>=3.7)

You can use any tool and programming language you want for the tasks as long as your output uses the correct format (e.g., the required timestamp format). However, the hints provided in this handout may not be directly applicable to tools other than the ones above.

Task 1: Simple Packet Analysis (20 points)

This task is designed to help you get familiar with packet analysis using Wireshark. The given task1.pcap is a packet trace captured from John's computer. Your task is to analyze the packet trace and answer the following questions.

(1) What browser software did John use? Choose one from the list: Chrome, Edge, Firefox, Safari. (5 points) (Hint: You can infer the browser software by analyzing the User-Agent header in the HTTP packets.)
(2) Find the IP address of the DNS server that John's computer primarily uses. (5 points)
(3) John has created a new Zoom conference room via Zoom Workspace application to host a meeting with his teammates. The application sent a request to *us05www3.zoom.us* when John created the meeting room. Find the DNS response packet, which contains an IP address of *us05www3.zoom.us* from task1.pcap, and answer both (1) DNS response packet's timestamp and (2) retrieved IP address. (Answer Format: 2024-01-23 11:22:33.123456, 1.2.3.4) (5 points)
(4) In task1.pcap, let's assume that the most frequent (ip, port) pair from which John receives packets is the source of zoom's streaming-related packets. Find the (ip, port) pair from which zoom's streaming-related packets come. (Answer Format: 1.2.3.4, 443) (5 points) (Hint: Wireshark's "Statistics" menu can be useful.)

Submission

- Please write your answers in a {SID}_lab2_task1.txt file, with each answer on a separate line: the first answer on the first line, the second on the second line, and so on, without any extra spaces or lines.

Task 2: Analyzing Zoom Packets (40 points)

In this task, let's analyze the network traffic data captured from a Zoom conference call and infer some user activities in the conference. As briefly mentioned above, the Zoom protocol is a proprietary protocol, and the details of the protocol are not publicly available. While the voice and video data in the payload are encrypted, some metadata in the packet headers is not encrypted. By analyzing these metadata, we may be able to infer some user activities in the conference.

The main scenario we focus on in this task is the situation where a user is in the waiting room of a Zoom conference call. When a user is joining a Zoom conference call operated by a host, the user typically first enters the waiting room. The host can then authorize the user to enter the conference room. The scenario we consider in this task is when the host does not authorize the user in the waiting room to enter the conference room. Thus the user is still in the waiting room while collecting network traffic at his/her end.

Users in the waiting room who are not yet authorized by the host in the Zoom conference room are not supposed to learn about the activities in the conference room. However, as we will see in this lab, some metadata in the packet headers are leaked to the users in the waiting room. By analyzing these leaked metadata, we can infer some user activities in the conference room.

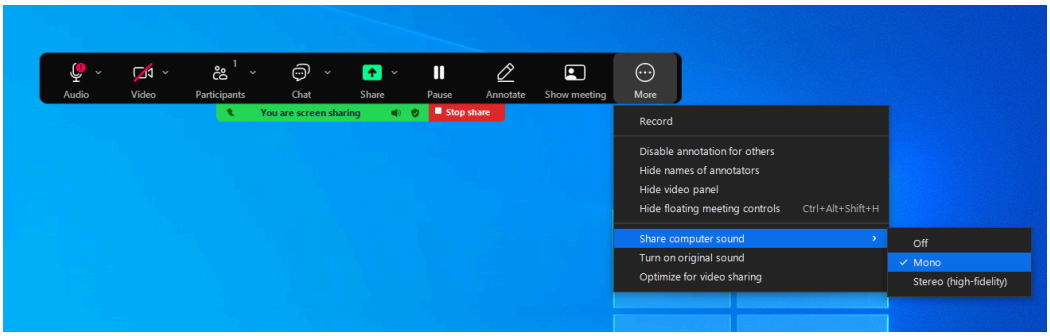
The given .pcap file includes information on packets captured using wireshark when user David is in the waiting room, and the .png file contains part of what is happening inside the conference room. Your task is to analyze these packets to infer what is happening in the room and complete the sub-tasks below. In this lab, assume that all participants turn on their microphone only when they speak, and turn off their microphone when they are not speaking.

Background: Zoom SR Packet

To make your analysis manageable, we provide you with some information about the Zoom SR packet:

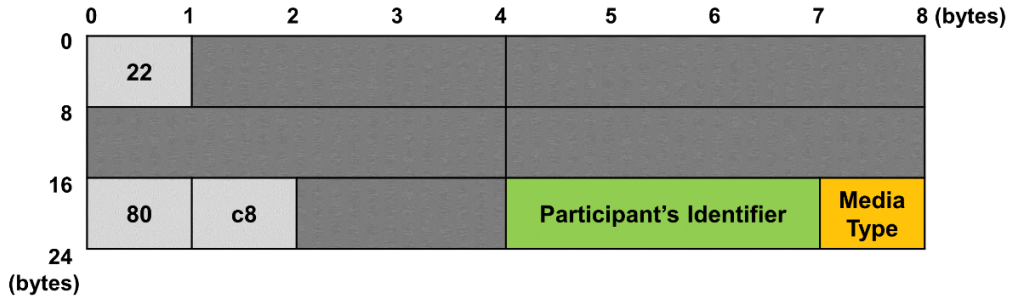
- The Zoom's Sender Report (SR) packet is the one that contains information about the conference room. The SR packet is delivered through the 8801 UDP port of some specific Zoom server.
- The first byte of the UDP payload of the SR packet is 05. (Packets whose first byte is not 05 in the UDP payload may be unnecessary for your analysis.)
- The SR packet contains an SR message that includes two key metadata: the identifiers and media stream type.
    - SR messages contain specific patterns for each participant's unique identifier which he/she newly receives when entering each room, where the first three bytes start at 01 00 04 and increase by four. (ex. 01 00 04, 01 00 08, 01 00 0c, …) In this lab, it is referred to as SSRC ID, and the SSRC ID value in this task does not exceed 01 00 2c.

- The single Media Type byte that follows the identifier specifies different media stream types. In this lab, you should identify each type byte indicating audio (voice from microphone, shared computer sound, …) and screen. (Hint: Range is 01~04).
- Exception: The SSRC ID value in the SR packet may not follow the pattern mentioned above. If computer sound sharing is enabled during screen sharing, the SSRC ID value in the SR packet for the shared sound will appear with the SSRC ID part increased by 2 from the last 1 byte of the original value. (ex. 01 00 18 → 01 00 1a)

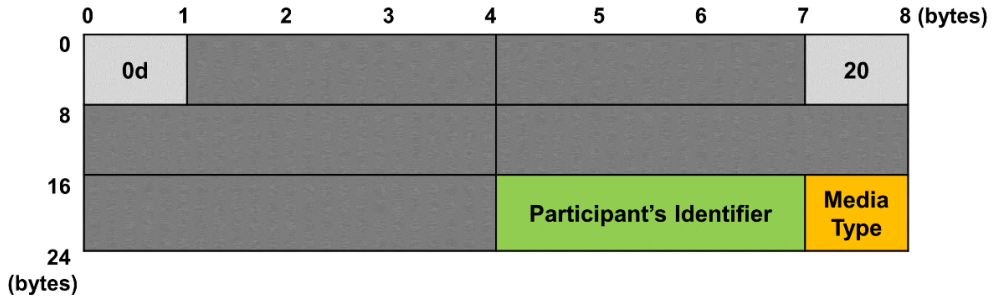


- The UDP payload of the SR packet contains the following sequence depending on the media stream type. (Audio Streaming / Screen Streaming)

< Audio Streaming >



< Screen Streaming >



If you want to reproduce zoom leakage and confirm these SR packet's characteristics by yourself, please refer to Note in Task 4.

Sub-tasks

(1) What is the SSRC ID of Bob? (5 points)
(2) Which Media Type byte (1 byte) indicates audio stream in SR packet? (5 points)
(3) Which Media Type byte (1 byte) indicates screen stream in SR packet? (5 points)
(4) Estimate the time when Alice started sharing the screen. Write down the timestamp of the packet that contains this information for the first time. (timestamp format: 2024-09-26 12:34:56.123456) (5 points)
(5) Did Alice turn on sound sharing during the presentation? If so, guess when Alice first turned on sound sharing. Write down the timestamp of the packet that contains this information for the first time. The answer should be “No” or a specific timestamp. (timestamp format: 2024-09-26 12:34:56.123456) (10 points)
(6) Answer who asked a question in the middle of the presentation. (Choose Bob/Charlie/None.) (10 points)

Submission

- Please write your answers in a {SID}_lab2_task2.txt file, with each answer on a separate line: the first answer on the first line, the second on the second line, and so on, without any extra spaces or lines.

Task 3: More Analysis on Zoom Packets (40 points)

Now that you have analyzed the Zoom SR packets, let's move on to the next task to infer more user activities in the Zoom conference room. In Task 2, you were able to analyze various packets leaked from a Zoom conference room to a Zoom waiting room, simply using their structure and Wireshark GUI. In Task 3, you will analyze the same type of packets but for tracking user activities with slightly more effort. To be more specific, you will infer the 'Active Speaking' state of each participant in the Zoom conference room. This will enable you to understand more interesting dynamics happening in the conference room.

Background: How to infer ‘Active Speaking’ state?

- When a participant’s microphone is on, the participant’s audio stream is classified into two different states in the Zoom system:
    - State 1. Not speaking: only background noise is transmitted to the stream when the participant is not speaking.
    - State 2. Active speaking: the participant is actively speaking, so both background noise and the participant's clear voice are transmitted to the stream.

- You can infer whether each participant is in 'Active Speaking' state through the '*packet count*' field of Audio Streaming SR packets.
    - Assume that an Audio Streaming SR packet contains *packet count* field which indicates the number of packets sent by a specific participant between adjacent Audio Streaming SR packets for the participant.
        - Where is the *packet count* field? That's what you should figure out :-)
    - If the packet count of a participant's current Audio Streaming SR packet *P'* exceeds that of the same participant's previous Audio Streaming SR packet by a certain threshold, we will consider that the participant is actively speaking at the moment of *P'*.

Sub-task 1: Where is a *packet count* field? (20 points)

Based on the above background, figure out a location of *packet count* field and its endianness ('big' or 'little') in an Audio Streaming SR packet.

Hint:

- It is located after the SSRC ID and Media Type fields.
- Its size is 4 bytes.
- You may need to do some coding.
- You may use task3-sample.pcap, task3-sample-output.txt, and used threshold=30 to specify location of *packet count* field. For the format of task3-sample-output.txt, please check the "Output format" of Sub-task 2.

Please write your answers in a {SID}_lab2_task3/task3-1.txt file in the following format.

Format
<Offset from the next byte of Media Type field> <Endianness (big / little)>

Example (Where *packet count* is located right next to the Media Type and follows little-endian.)
0 little

Sub-task 2: Write a script to check Active Speaking state! (20 points)

In this sub-task, you are asked to write a script that analyzes a .pcap file and outputs timestamps of packets where participants' Active Speaking state is revealed. Your script "speaker-analysis.py" should be able to analyze .pcap files (ex. task3-sample.pcap) which are captured under the following scenario:

- Alice and Bob are in the Zoom conference room to have a conversation before David enters the waiting room
- After David enters the waiting room and starts to capture packets, Alice and Bob turn on their microphones
- Alice turns on the microphone before Bob does
- Alice and Bob use only the microphone and not any other functions (e.g., screen sharing, cameras)

Here are some requirements for speaker-analysis.py:

- Your script should receive input in the following format of command-line argument, and should print analysis results to stdout in specific format.
- For grading, TAs will run your script as follows and will check the output file (thus, any deviation from the format will lead to deduction):

```
python speaker-analysis.py <pcap file> <threshold> > <output file>

# Example: How TAs created task3-sample-output.txt
python speaker-analysis.py "task3-sample.pcap" 30 >
task3-sample-output.txt
```

- Input format
  - <pcap file> is the .pcap file captured by David under the scenario which is described in previous detail.
  - <threshold> is the value used to determine Active Speaking state, and threshold=30 will be used for grading.
- Output format
  - <output file> is the file where stdout generated by the script is saved.
  - It should contain the SSRC ID of Alice and Bob, and timestamps of packets where the Active Speaking state of each participant is revealed. The format and example are as follows:

Format
Alice <Alice's SSRC ID>

<Timestamp1> <Timestamp2> … Bob <Bob's SSRC ID> <Timestamp1> <Timestamp2> …

Example
Alice 010014 2024-10-03 11:11:11.123456 2024-10-03 11:22:22.123456 Bob 01001c 2024-10-03 11:33:33.123456 2024-10-03 11:44:44.123456

- For your convenience, the last newline character will be automatically stripped in the grading process.
- Check the sample .pcap file and the corresponding correct output file to test your script. (task3-sample.pcap, task3-sample-output.txt)
- You will earn full points if your script produces the correct output for both the sample test input and hidden test inputs.
    - Reminder: Same participant may have different SSRC ID in different conference rooms.

Submission

- {SID}_lab2_task3 (ex. 20241234_lab2_task3)
    - README.md (optional)
    - task3-1.txt
    - speaker-analysis.py
    - requirements.txt

If you want to provide any further information for your script, please leave it on the README.md file, especially when you do not use recommended tools which are scapy with python>=3.7.

The requirements.txt file should contain python libraries with versions that are required in speaker-analysis.py. You may try `pip freeze` and use its stdout as a requirements.txt. You may remove lines about other libraries which are not required in your speaker-analysis.py script.

Here's an example for requirements.txt:

```
scapy==2.6.0
more-itertools==10.5.0
```

Task 4: Bonus (optional; up to 50 points)

Completing Task 1, 2, and 3, you already deserve full marks for this lab. However, if you are interested in network traffic analysis and want to challenge yourself, you can try the bonus task. This **optional** bonus task is an open-ended task where you can analyze your **own** network traffic data captured from your **own** Zoom conference call.

You may set up a Zoom conference call with a few participants under **your control** and capture the network traffic data. You can initiate various activities in the conference call (outside the scope of the tasks in this lab) and analyze the network traffic data to see whether you can infer these activities from the packet headers.

Note: The metadata information regarding the user activities in the conference room appears to be leaked to the network traffic data of the participants in the waiting room when the Zoom client's "Audio > Automatically join audio by computer when joining" setting is enabled. We encourage you to enable this setting when you set up your own Zoom conference call to see whether you can infer the user activities in the conference room from the packet headers.

Grading

If you happen to discover that any previously unknown information of user activities can be inferred from the packet headers, you can write a report on your findings and submit it as a bonus task. Bonus points will be awarded based on the novelty and creativity of your findings.

Your report should include the following:

- README.md: A brief description of the activities you initiated in the conference call and the information you inferred from the packet headers.
- Screen recording: A screen recording of the activities in the conference call and the packet analysis process.
- Script (optional): If you wrote a script to analyze the packet headers, you can submit it as well.

Responsible Disclosure

If you discover any new findings, the teaching crew will help you responsibly disclose your findings to Zoom. Please do not disclose your findings to the public or any third party without consulting the teaching crew. Should you have any questions or concerns, please reach out to the teaching crew.

Submission

Please submit your bonus task in a zip format. Your zip file name should be {SID}_lab2_task4.zip.

Gradescope Submission

- It should be submitted in a zip format.
- Your zip file name should be like: {SID}_lab2.zip.
- {SID}_lab2.zip should contain:
  - Task1: {SID}_lab2_task1.txt
  - Task2: {SID}_lab2_task2.txt
  - Task3
    - {SID}_lab2_task3/task3-1.txt
    - {SID}_lab2_task3/README.md (optional)
    - {SID}_lab2_task3/speaker-analysis.py
    - {SID}_lab2_task3/requirements.txt
  - Task4 (optional)
- Files with different file name formats will be either penalized or not graded at all.

Document History

- V1: Initial version
- V2: Minor update
  - Add document history.
  - Specify maximum points for Task 3 > Sub-task 2.
- V3: Major update
  - (Clarify) Task 1-(4) description is updated.
    - Before
      - “It turns out that the most frequent packet type in task1.pcap is zoom’s streaming-related packets.”
    - After
      - “In task1.pcap, let’s assume that the most frequent (ip, port) pair from which John receives packets is the source of zoom’s streaming-related packets.”
  - (Clarify) Task 2 description is updated.
    - In this lab, you should identify each type byte indicating audio (voice from microphone, shared computer sound, …) and screen.
    - If computer sound sharing is enabled during screen sharing, …
  - (Convenience) Task 3-2 new description about output format is added.
    - “For your convenience, the last newline character will be automatically stripped in the grading process.”

- V4: Minor update
    - Typo correction: task3-sample-output.pcap → task3-sample-output.txt in Task 3 > Sub-task 1.
- V5: **Critical** update
    - (Clarify) Task 2 & 3 description is updated to clarify definition of SSRC ID.
        - Task 2
            - SR messages contain specific patterns for each participant's unique identifier which he/she newly receives when entering each room, …
        - Task 3
            - You will earn full points if your script produces the correct output for both the sample test input and hidden test inputs.
                - Reminder: Same participant may have different SSRC ID in different conference rooms.