Lab Assignment 3

Application Layer (Extract Data from PCAP)

Due Date: Saturday, October 28^{th} by 11:59 PM

Instructor: Dr. Abdelhak Bentaleb

Submission Deadline

28^{th} October 2023 (Saturday) 11:59 pm sharp. A 5-mark penalty will be imposed on late submission (Late submission refers to submission or re-submission after the deadline). The submission folder will be closed on **30^{th} October 2023 (Monday) 11:59 pm sharp**, and no late submissions will be accepted afterward.

Objectives

In this lab, our primary goal is to gain proficiency in the analysis of network packets and the extraction of diverse information from pcap files.

This lab should be completed **individually** and is worth **50 marks** in total.

Setup

You should use your local machine for all tasks in this assignment. The tool you need (on your local machine or lab machine) is `Python`. **In this lab, you can not use Wireshark**.

Submission

Create a **single zip file** that contains **only** your **python** code (`part1.py`, `part2.py`) and their outputs, and submit it to the corresponding assignment folder in Moodle. Name your file as ≪Student number≫.zip, where ≪Student number≫ refers to your student ID number.

Plagiarism Warning

You are free to discuss this assignment with your friends. **However, you should refrain from sharing your answers**. We highly recommend that you attempt this assignment on your own and figure things out along the way as many resources are available online.

We employ a zero-tolerance policy against plagiarism. If a suspicious case is found, the student will be asked to explain his/her answers to the evaluator in person. The confirmed breach may result in a zero mark for the assignment and further disciplinary action from the department.

Question & Answer

If you have any doubts about this assignment, please post your questions on Moodle or consult the TA. However, the TA will NOT debug issues for students. The intention of Q&A is to help clarify misconceptions or give you necessary directions.

Part 1: Analyzing network traffic using pcap files

You have to develop a Python script to analyze network packets from the pcap files `https://comp445.github.io/wireshark-labs/Lab3-pcap-1.zip`. For each file, your script should extract various statistics as follows:

1. **[5 marks]** Determine the of packets in each pcap file and the total number in all the pcap files.
2. **[5 marks]** Identify distinct source IP addresses and the number of packets for each IP address, sorting them in descending order.
3. **[5 marks]** List distinct destination TCP ports and the number of packets sent to each port, in descending order.
4. **[5 marks]** Calculate the number of distinct source IP and destination TCP port pairs, sorting them in descending order.

Part 2: Finding scanning/probing in network traffic

You have to develop a Python script that is able to identify probing and scanning in a stream of packets. Intuitively, probing is when an agent makes repeated attempts to access or discover a service on a port. A scanning is when an agent tries to map large parts of the IP address/port space to see if there are any running services on those ports. Your code will read in a packet trace as a pcap file, a target IP address, and output a list of probing and scanning found against that IP address, as well as the originating IP addresses of the probing and scanning.

Figure 1, displayed below, provides a visual representation of the formal definitions for probing and scanning. The graph employs time on the X-axis and port number on the Y-axis. When a packet arrives at a specific time t with port p (pertaining to a particular IP address), a point is plotted at the coordinates (t, p). Probing and scanning are conceptually defined as clusters of points within the time-versus-port space, along a specific axis.

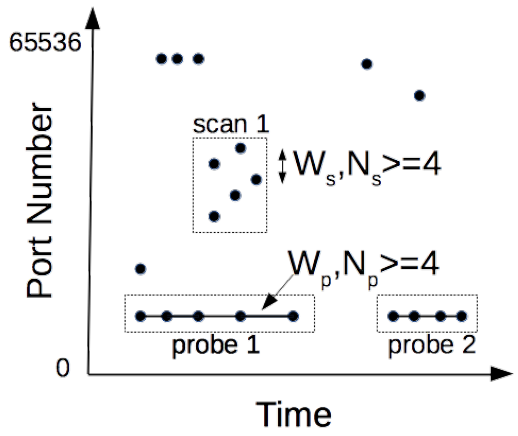

Figure 1: Ports vs. Time in a packet stream define probing and scanning.

A probing is characterized as a collection of points with identical port numbers grouped closely together in the time dimension. Conversely, a scanning comprises a group of points that share the same port space, represented on the Y-axis. To illustrate, in Figure 1, two separate time periods are denoted as containing two distinct probes, and a single scan is observed encompassing a portion of the port space.

There are many algorithms that group points into logical clusters, however, this lab assignment will use a simple one. Clusters are defined by 2 parameters: (1) the width, (which are W_p and W_s for probing

and scanning, respectively), and (2) the number (which are N_p and N_s, for probing and scanning, respectively). The number is the minimum number of points needed for a group to be considered; that is, there must be at least N_p or N_s points (i.e. packets) in a group to report a probing or scanning.

A cluster on a given axis is defined by points that are closer together; if a point is too far away it is considered to be in another cluster. The width is the distance between consecutive points for those points to be considered in the same group. That is, the invariant for a point to be in a group (probing or scanning) is that a point must be at least W_p/W_s units (seconds or port ID) to at least one point in the same group.

Your Task

[30 marks] Your code needs to read a pcap file (in total there are 4; 2 for probing and 2 for scanning) and take six parameters as inputs using the following options (in parenthesis):

1. The filename of the pcap file. (-f)
2. A target IP address. (-t)
3. The width for probing, in seconds, W_p. (-l)
4. The minimum number of packets in a probing, N_p. (-m)
5. The width for scanning, in port ID, W_s. (-n)
6. The minimum number of packets in a scanning, N_s. (-p)

The output of your code is lists of identified probing and scanning, and the source IPs of each probing and scanning. One set of reports will be for TCP and the other for UDP. You can download pcap files from: `https://comp445.github.io/wireshark-labs/Lab3-pcap-2.zip`.