CS4458/CS9636 – Network Security
Assignment 4

Department of Computer Science
University of Western Ontario
Fall 2023

5% of Final Grade Due Date: **November 23, 2023**

Purpose

- Programmatically analyze the network traffic given in a network trace file.

Description

Use Python 3.7+ and the `dpkt` library (for more information, visit the documentation and Jeff Silverman's documentation) to analyze the packets contained in the given PCAP files to complete the following tasks:

1. (40 pt) Implement a function named `packet_summary` that takes as input a file name and hierarchically prints by type the number of packets seen in the given file. The function must calculate the number of Ethernet, IP (both IPv4 and IPv6), TCP, HTTP, HTTPS, FTP, SSH, SMTP, UDP, DHCP, NTP, Non-IP, and ARP packets seen. You only need to consider IPv4 packets for calculating the number of TCP and UDP packets. You can find ports used by the protocols we are interested in on Wikipedia.
2. (30 pt) Implement a function named `subnet_summary` that takes as input a file name and prints the different subnets (sorted in descending order by count) appearing in the packets contained in the given file. You only need to consider IPv4 packets and a subnet is represented by anything prior to the second dot in the address. For example, in the address 111.222.150.200, the subnet is 111.222.
3. (30 pt) Implement a function named `detect_syn_scanning` that takes as input a file name and prints IP addresses that potentially performed SYN scans. SYN scans are a type of port scanning where the scanner sends a large number of TCP SYN packets to hosts on the network and watches for SYN+ACK response packets (for more information, visit the NMAP documentation). A scanner can usually be identified as sending significantly more SYN packets than the SYN+ACK packets they receive because most hosts are not open to receiving connections. For this assignment, you should look for IPs that sent three times as many (and at least 50) SYN packets than the number of SYN+ACK packets they received. You should also print beside each IP address the number of SYN packets they sent and the number of SYN+ACK packets they received.

Provided Files

You are given the following files along with the assignment outline:

- `main.py`: The Python file you are required to complete and submit.
- `part1.pcap`: Sample PCAP file for testing the `packet_summary` and `subnet_summary` functions.
- `part2.pcap`: Sample PCAP file for testing the `detect_syn_scanning` function.

Expected Output

The expected output of the testing code is given below (split into three columns for concision). The packet counts and IPs have been modified to preserve the correct answers.

```
Packet Summary:
Ethernet: 1000
  IP: 900
    TCP: 500
      HTTP: 100
      HTTPS: 200
      FTP: 100
      SSH: 50
      SMTP: 50
    UDP: 400
      DHCP: 300
      NTP: 100
  Non-IP: 100
    ARP: 100
```

```
Subnet Summary:
192.111      10000
192.122       8000
4.90          5000
10.10         3000
192.234       2000
20.20         1000
15.10          500
200.80         200
100.12          10
```

```
SYN Scanners (sent, received):
10.10.100.100 (1000, 30)
10.128.0.1 (3000, 0)
170.16.10.3 (500, 20)
170.16.10.2 (500, 5)
```

Submission Instructions

- Upload your submission to OWL and indicate if you wish to use any late coupons for this assignment in the OWL submission text box. For example, you may simply write: "I would like to use 1 late coupon for this assignment." For detailed information about late coupons, refer to the course outline.
- Please upload the files of your submission directly and not as part of a ZIP archive. Written responses should be submitted as a single PDF file (i.e., not a Microsoft Word document).
- For programming questions, follow standard Python conventions for good code style. This includes, but is not limited to:
  - Meaningful variable names that follow Python naming conventions.
  - Use of constants instead of "magic numbers".
  - Appropriate use of indentation, white space, and consistent styling to improve readability.
- For skeleton code files, do not modify the given code in ways such as changing the structure or method signatures. Your programming answers must conform to the constraints and restrictions imposed by the skeleton code and assignment outline.