Assignment 1 – SYN Flood Attack

You must upload all the necessary files in a ZIP file named after both of the student's ID.

1) The assignment must be submitted by the date specified in the submission box.

2) All the assignment files (Code, PDF, any additional file) must be submitted in a ZIP file. The PDF must contain all the necessary screenshots with a description for each one of them.

3) The assignment must be done in single or pairs only, **no triplets!**

4) You can use any reference material available through the course Moodle website or any material passed during the exercises.

5) You can use any reference from the web, but copying codes from websites is forbidden; this includes codes from other GitHub repositories. A student who gets caught in the act will automatically fail this assignment (score 0). Every website you used to solve the assignment must be noted in the PDF.

6) No delays in submission without special permission. Lateness without approval will result in failure of the assignment (score 0). In exceptional cases, you should send an email to the course coordinator.

7) All submissions will be submitted via the course's Moodle website. Email submissions will not be accepted, resulting in a failure in the assignment.

8) The order and design of the code must be carefully considered. Make sure that the program output is as clear as possible, comments in the code, and meaningful variable names. You must also submit a makefile that compiles all the necessary C programs.

9) The assignment is personal for each single/pair, and you should not accept help from other people, whether outside the university or inside it. You can contact the course staff for help or raise a question in the course forum. Do not transfer code sections between students, upload solutions or parts of solutions to websites on the Internet or in various communication groups.

10) Students who copy a solution will receive a 0 in all assignments in the course and a report will be made to the institutional disciplinary committee.

11) It's recommended that you work with dockers.

Good Luck!

DDoS

We will try to do a DDOS today. We will need 3 computers: Attacker, Target, Monitor. The attacker will send 10,000 TCP-SYN packets to the target machine, in a 100 iterations loop. Total of 1 million packets. The target machine will run an Apache server. The monitor will send pings to the server during the attack.

<u>Note</u>: It's recommended you'll use dockers for this assignment.

The attacker will run two types of programs, one in C and one in Python. For each program, we will measure:

- The time the attacker needs to send a packet (avg).
- The ping's RTT from the monitor. You will have to send one ping every 5 seconds.

The above measurements would appear in measurement files, in the following scheme:

- **syns_results_c.txt** – Containing each syn request (an index of the request or a counter) sent from C program and the time it took to send it, separated by newline. At the end of the file, a measurement of how long it took for the program to send all the packets, and the average time to send a packet.

- **syns_results_p.txt** – Same as above, but for the Python program.

- **pings_results_c.txt** – Containing all the ping requests generated by the Monitor machine, while the Attacker attacked using the C program. Each ping request should be separated with newline. At the end of the file, you should add the average ping's RTT.

- **pings_results_p.txt** – Same as above, but while the Attacker used the Python program.

You'll need to create matplotlib graphs describing the 2 metrics for C and for Python.

- **Syn_pkts_c.png** & **Syn_pkts_p.png** – The graph should describe the time needed for the attacker to send a packet. The x axis should be the time needed to send a packet. The y axis will represent the number of packets sent.

- **Pings_c.png** & **Pings_p.png** – The graph should describe the RTT for the ping requests. The x axis should be the RTT in milliseconds. The y axis will represent the number of pings.

Notes:

- One graph for the Python attack and another for the C attack.
- The y axis should be logarithmic, and the x axis should be distinguishable.
- You'll also need to create a short report describing the results, with STD and average.
- You'll need to include your understandings of the diffs.

You'll upload a zip file including:

- **Attack.py & Attack.c** – Attack files.
- **DDOS.pdf** – Report of the lab.
- **Syn_pkts_c.png & Pings_c.png** – Graphs of C.
- **syns_results_c.txt & pings_results_c.txt** – Results files for C.
- **Syn_pkts_p.png & Pings_p.png** – Graphs of Python.
- **syns_results_p.txt & pings_results_p.txt** – Results files for Python.

Some notes:

- The DDOS attacks might run several hours depending on your hardware, do not start this lab 1 hour before the due date.

- After you saved the result files, you can duplicate them and manipulate the text so it will fit for the matplotlib for easier graph generation.

- The Monitor machine shouldn't consume too much resources since it only sends ping requests and saves them into a file, therefore you can give this machine less memory and put more into the Attacker machine so the attack will finish faster.

- Remember that printing into the console also consumes resources and it is not needed during the attack, but if you insist to see what is happening, you can print the progress into the console every X packets/pings (like every 500 packets for example).