EECE 655 – Internet Security – Assignment 1
Group 5
Project Manager: Fouad Trad
Attack Team: Chadi El Hakim, Mohammad Kachmar
Detection Team: Layan Barhoumi, Anthony Sleilaty

TCP SYN Flooding Attack and Detection

1. Introduction

In this assignment, we worked on the implementation of the TCP SYN Flood attack along with its detection mechanism in python. The tool we built constitutes a prototype that can be used for testing and experimentation purposes. We tested our tool on one device, and on two devices, and it was effective in both cases. In this document, we briefly explain how to use this tool and we show the results of our conducted tests.

2. Software Specifications

To run the tools (to perform the attack and/or detection), the following is required:

- Python must be installed on your system [2].
- After that, Scapy (v2.4.4) [3] and netifaces [4] libraries should be installed.
- If no sniffer is installed on your system (like Wireshark), you have to download NPcap [5].
- You should download the codes from the GitHub repository [6].
- Finally, to run the codes, you can use an IDE (like VS Code), or the command line interface.

3. Files description

3.1 Attack.py

The goal of this file is to execute the SYN Flooding attack. The user has to initially specify the destination IP (the victim) (which can be the local host or any other host), the destination port, the number of packets he wants to send (it can be infinite), and if he wants to use one or multiple spoofed IPs for the attack. He also can choose if he wants the spoofed IPs to be on his subnet or not. To do this, we divide our work into multiple functions like **getDestIP(), getDestPort(), getNumofPackets(), getSubnet (), getLocalIP(), getNetwork()** and **random_ip().** The role of **getSubnet(), getLocalIP()** and **getNetwork()** functions is to get the subnetmask of the currently connected network, and computes the network subnet in the format "XX.XX.XX.XX/X" this is then passed to the **random_ip()** function to randomize an IP under a given subnet.
After having all this information, we execute the **SYN_DOS()** function which fills the SYN packet's fields for both the IP and TCP layer and sends it to the destination. In the IP layer we have to specify the source and destination IPs. For the destination IP, we use the one that the user provided at the beginning. As for the source, we use the **random_ip()** function depending on whether the user chose to have the spoofed source IPs on the same subnet or not "0.0.0.0/0" is passed for completely randomized IP. For the TCP layer, we randomized the source port, the sequence number, the window size and we most importantly defined the flag as being S for constantly sending SYN packets. We then stack up the IP and TCP layers and send a packet. Finally, we send the packet to the destination IP. We keep doing this loop, until we reach the maximum number of packets specified by the user. If the maximum number is infinite, the loops keeps executing until we stop it manually.

3.2 Detection.py

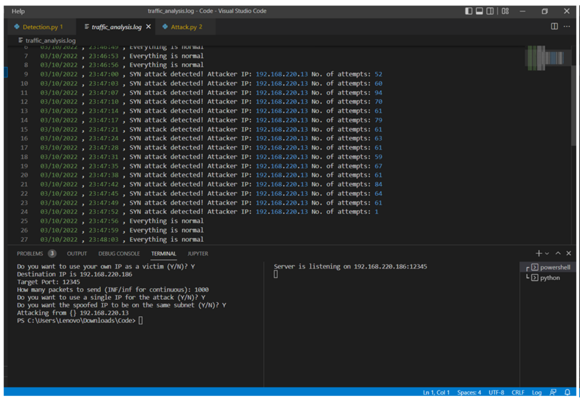
The goal of this code is to detect a SYN Flooding attack. The user simply has to run the code and observe the traffic analysis log to be notified whether everything is normal, there is an attack from one IP, or there is an attack

from multiple IPs. To do this, we create a socket that will host a service on the IP of the detector's device, and a reserved port. We employ an asynchronous packet sniffer, and run the functions for logging and accepting users in separate independent threads, optimizing the performance of the tool even during high activity, and preventing the tool from "blocking", and allowing us to stop it using a simple keyboard interrupt.
We initialize two hash maps: **synCount** that keeps track of which IPs the host received SYN packets from, and **ackCount** that keeps track of which IPs the host received ACK packets from. These maps will be updated in the analyze function. We declare the sniffer and specify the interfaces it sniffs on by passing list **["\\Device\\NPF_Loopback", conf.iface]** to the iface option. The first interface is the loopback interface, which is needed incase the attack-detection tool is being tested on the same device, and the second interface is the default WiFi interface of the host that is running the detection tool. Note that an exception is raised when passing a list to iface, but this is specific to scapy 2.4.5. To run this successfully, scapy 2.4.4 is required.
We begin by running the sniffer, which passes every packet it sees to the analyze function. The analyze function first checks if the IP layer is present in the packet. If so, it then checks if the packet is a TCP SYN packet, and accordingly it increments the SYN count for the source IP in the **synCount** map. Next, it checks if the packet is a TCP ACK packet, and similarly it increments the ACK count for the source IP in the **ackCount** map.
Meanwhile, the logging function is running on its own thread and is printing the host's state every 3.5 seconds in a log file. To do this, we first check if the **synCount** map is not empty, then we check the conditions as follows:
We compare the number of SYN packets of the IP address having sent the most of them to 2 different numbers. First, we check if it is at least 3 times bigger than the ACK count received by the same IP, and then we check if this number represents more than 80% of the total number of SYN packets received by calculating their ratio. If both those conditions are true at the same time, we can assume we are being attacked by a single IP. Of course the 80% number is not fixed and can be changed according to the use case.
Otherwise, we compare the number of half open connections to the number of full connections by comparing the lengths of the **synCount** and **ackCount** hash maps. If there are 5 times more entries in synCount than in ackCount, then there are 5 times more half open connections than full connections, so we notify the user that there is an attack from multiple IPs. Also the numbers here are just for the sake of the simulation and can be changed according to the use case.
Otherwise, there is no attack and the behavior is normal.

4. Experiments

We conducted experiments to test our tool on one device (section 4.1) and on 2 devices (section 4.2). In each case, we test it according to different combinations of whether one or multiple IPs should be used for the attack, and whether the spoofed IP (or IPs) should be on the same subnet of the host or not. Screenshots are provided in the next two sections.

4.1 Attack and Detection running on one device

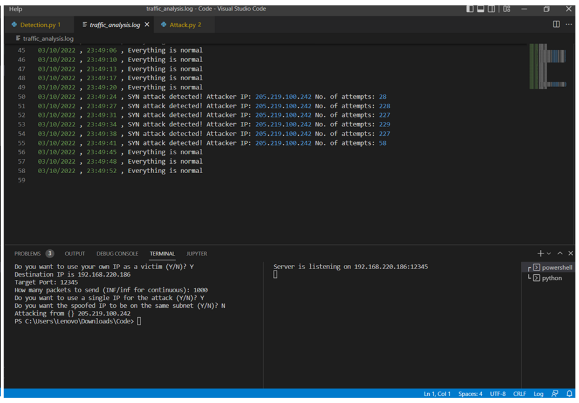


```
03/10/2022 , 23:46:49 , Everything is normal
03/10/2022 , 23:46:53 , Everything is normal
03/10/2022 , 23:46:56 , Everything is normal
03/10/2022 , 23:47:00 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 52
03/10/2022 , 23:47:03 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 60
03/10/2022 , 23:47:07 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 94
03/10/2022 , 23:47:10 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 70
03/10/2022 , 23:47:14 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 61
03/10/2022 , 23:47:17 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 79
03/10/2022 , 23:47:21 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 61
03/10/2022 , 23:47:24 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 63
03/10/2022 , 23:47:28 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 61
03/10/2022 , 23:47:31 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 59
03/10/2022 , 23:47:35 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 67
03/10/2022 , 23:47:38 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 61
03/10/2022 , 23:47:42 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 84
03/10/2022 , 23:47:45 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 64
03/10/2022 , 23:47:49 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 61
03/10/2022 , 23:47:52 , SYN attack detected! Attacker IP: 192.168.220.13 No. of attempts: 1
03/10/2022 , 23:47:56 , Everything is normal
03/10/2022 , 23:47:59 , Everything is normal
03/10/2022 , 23:48:03 , Everything is normal
```

```
Do you want to use your own IP as a victim (Y/N)? Y
Destination IP is 192.168.220.186
Target Port: 12345
How many packets to send (INF/inf for continuous): 1000
Do you want to use a single IP for the attack (Y/N)? Y
Do you want the spoofed IP to be on the same subnet (Y/N)? Y
Attacking from {} 192.168.220.13
PS C:\Users\Lenovo\Downloads\Code>
```

```
Server is listening on 192.168.220.186:12345
```

Figure 1 Attack from single IP on the same subnet

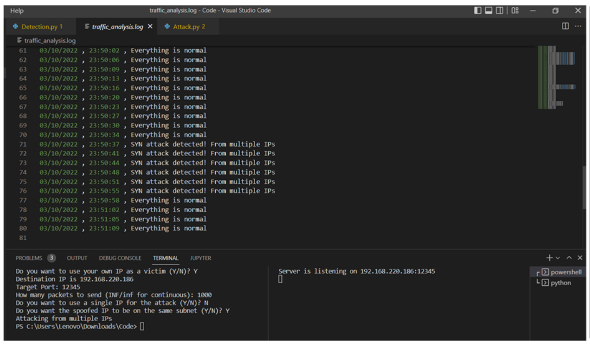


```
03/10/2022 , 23:49:06 , Everything is normal
03/10/2022 , 23:49:10 , Everything is normal
03/10/2022 , 23:49:13 , Everything is normal
03/10/2022 , 23:49:17 , Everything is normal
03/10/2022 , 23:49:20 , Everything is normal
03/10/2022 , 23:49:24 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 28
03/10/2022 , 23:49:27 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 228
03/10/2022 , 23:49:31 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 227
03/10/2022 , 23:49:34 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 229
03/10/2022 , 23:49:38 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 227
03/10/2022 , 23:49:41 , SYN attack detected! Attacker IP: 205.219.100.242 No. of attempts: 58
03/10/2022 , 23:49:45 , Everything is normal
03/10/2022 , 23:49:48 , Everything is normal
03/10/2022 , 23:49:52 , Everything is normal
```

```
Do you want to use your own IP as a victim (Y/N)? Y
Destination IP is 192.168.220.186
Target Port: 12345
How many packets to send (INF/inf for continuous): 1000
Do you want to use a single IP for the attack (Y/N)? Y
Do you want the spoofed IP to be on the same subnet (Y/N)? N
Attacking from {} 205.219.100.242
PS C:\Users\Lenovo\Downloads\Code>
```

```
Server is listening on 192.168.220.186:12345
```

Figure 2 Attack from a single IP on a different subnet

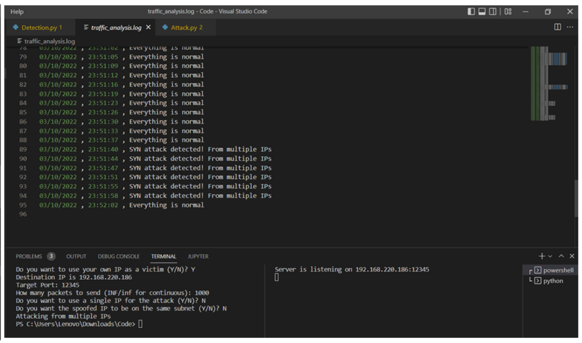


```
03/10/2022 , 23:50:02 , Everything is normal
03/10/2022 , 23:50:06 , Everything is normal
03/10/2022 , 23:50:09 , Everything is normal
03/10/2022 , 23:50:13 , Everything is normal
03/10/2022 , 23:50:16 , Everything is normal
03/10/2022 , 23:50:20 , Everything is normal
03/10/2022 , 23:50:23 , Everything is normal
03/10/2022 , 23:50:27 , Everything is normal
03/10/2022 , 23:50:30 , Everything is normal
03/10/2022 , 23:50:34 , Everything is normal
03/10/2022 , 23:50:37 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:41 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:44 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:48 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:51 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:55 , SYN attack detected! From multiple IPs
03/10/2022 , 23:50:58 , Everything is normal
03/10/2022 , 23:51:02 , Everything is normal
03/10/2022 , 23:51:05 , Everything is normal
03/10/2022 , 23:51:09 , Everything is normal
```

```
Do you want to use your own IP as a victim (Y/N)? Y
Destination IP is 192.168.220.186
Target Port: 12345
How many packets to send (INF/inf for continuous): 1000
Do you want to use a single IP for the attack (Y/N)? N
Do you want the spoofed IP to be on the same subnet (Y/N)? Y
Attacking from multiple IPs
PS C:\Users\Lenovo\Downloads\Code> 
```

```
Server is listening on 192.168.220.186:12345
```

Figure 3 Attack from Multiple IPs, same subnet



```
03/10/2022 , 23:51:05 , Everything is normal
03/10/2022 , 23:51:09 , Everything is normal
03/10/2022 , 23:51:12 , Everything is normal
03/10/2022 , 23:51:16 , Everything is normal
03/10/2022 , 23:51:19 , Everything is normal
03/10/2022 , 23:51:23 , Everything is normal
03/10/2022 , 23:51:26 , Everything is normal
03/10/2022 , 23:51:30 , Everything is normal
03/10/2022 , 23:51:33 , Everything is normal
03/10/2022 , 23:51:37 , Everything is normal
03/10/2022 , 23:51:40 , SYN attack detected! From multiple IPs
03/10/2022 , 23:51:44 , SYN attack detected! From multiple IPs
03/10/2022 , 23:51:47 , SYN attack detected! From multiple IPs
03/10/2022 , 23:51:51 , SYN attack detected! From multiple IPs
03/10/2022 , 23:51:55 , SYN attack detected! From multiple IPs
03/10/2022 , 23:51:58 , SYN attack detected! From multiple IPs
03/10/2022 , 23:52:02 , Everything is normal
```

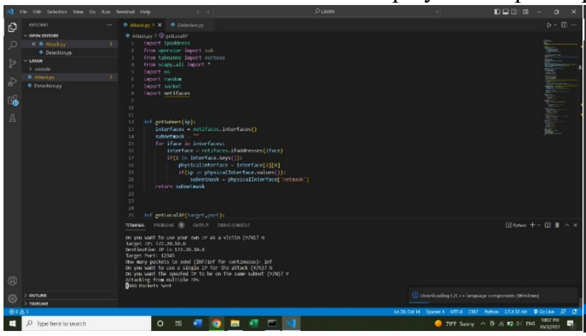
```
Do you want to use your own IP as a victim (Y/N)? Y
Destination IP is 192.168.220.186
Target Port: 12345
How many packets to send (INF/inf for continuous): 1000
Do you want to use a single IP for the attack (Y/N)? N
Do you want the spoofed IP to be on the same subnet (Y/N)? N
Attacking from multiple IPs
PS C:\Users\Lenovo\Downloads\Code> 
```

```
Server is listening on 192.168.220.186:12345
```

Figure 4 Attack from multiple IPs, different subnet

4.2 Attack and Detection running on one device

We also tested how our system behaves when the attack code is on one device and the detection code is on another one. Below we show screenshots from both devices when the attack is performed from a single IP on a different subnet. The other results were not displayed for space optimization, but they can be found on this link.



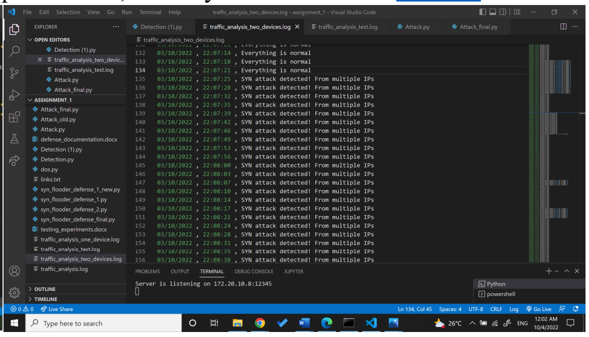
```python
import ipaddress
from operator import sub
from tabnanny import verbose
from scapy.all import *
import os
import random
import socket
import netifaces


def getSubnet(ip):
    interfaces = netifaces.interfaces()
    subnetmask = ""
    for iface in interfaces:
        interface = netifaces.ifaddresses(iface)
        if(2 in interface.keys()):
            physicalInterface = interface[2][0]
            if(ip in physicalInterface.values()):
                subnetmask = physicalInterface['netmask']
    return subnetmask


def getLocalIP(target,port):
```

```
Do you want to use your own IP as a victim (Y/N)? N
Target IP: 172.20.10.8
Destination IP is 172.20.10.8
Target Port: 12345
How many packets to send (INF/inf for continuous): inf
Do you want to use a single IP for the attack (Y/N)? N
Do you want the spoofed IP to be on the same subnet (Y/N)? Y
Attacking from multiple IPs
1000 Packets Sent
```

File Edit Selection View Go Run Terminal Help
LAVAN
Attack.py 3
Detection.py
Attack.py > getLocalIP
TERMINAL
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
Python
Downloading C/C++ language components (Windows)
Ln 30, Col 14 Spaces: 4 UTF-8 CRLF Python 3.7.4 32-bit Go Live
Type here to search
79°F Sunny
ENG



```
03/10/2022 , 22:07:14 , Everything is normal
03/10/2022 , 22:07:18 , Everything is normal
03/10/2022 , 22:07:21 , Everything is normal
03/10/2022 , 22:07:25 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:28 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:32 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:35 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:39 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:42 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:46 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:49 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:53 , SYN attack detected! From multiple IPs
03/10/2022 , 22:07:56 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:00 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:03 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:07 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:10 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:14 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:17 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:21 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:24 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:28 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:31 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:35 , SYN attack detected! From multiple IPs
03/10/2022 , 22:08:38 , SYN attack detected! From multiple IPs
```

```
Server is listening on 172.20.10.8:12345
```

File Edit Selection View Go Run Terminal Help
traffic_analysis_two_devices.log - assignment_1 - Visual Studio Code
EXPLORER
OPEN EDITORS
Detection (1).py
traffic_analysis_two_devices.log
traffic_analysis_test.log
Attack.py
Attack_final.py
ASSIGNMENT_1
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
JUPYTER
Python
powershell
OUTLINE
TIMELINE
Ln 134, Col 45 Spaces: 4 UTF-8 CRLF
Type here to search
26°C
ENG

Figure 5 Attacker's device

Figure 6 Detector's device

5. Contributions

The day we got the assignment, we set up an internal plan to handle who does what and an approximate timeline for each task to be able to deliver on time as shown in Figure 7, and we followed that plan throughout the assignment. The detailed division of the code is commented on the actual code files.

Handled by	Sept 20	Sept 21	Sept 22	Sept 23	Sept 24	Sept 25	Sept 26	Sept 27	Sept 28	Sept 29	Sept 30	Oct 1	Oct 2	Oct 3	Oct 4
The Attack team		1. Identify an attack to work on 2. Divide teams			Work on the attack implementation					Documentation			Implement changes	Approve report	Submit
The Detection team					Work on the Detection implementation										
The Project Manager	Set a project plan				Follow up with both teams					Testing, Implementing changes, and Integration of both tools		Feedback on additional changes		Writing report	

Figure 7 Task division

6. References

[1] R. Fatemi, 'synflood-attack-detection'. Oct. 13, 2020. Accessed: Oct. 03, 2022. [Online]. Available: https://github.com/rmfatemi/synflood-attack-detection

[2] 'Download Python', *Python.org*. https://www.python.org/downloads/ (accessed Oct. 02, 2022).

[3] 'Download and Installation — Scapy 2.5.0 documentation'. https://scapy.readthedocs.io/en/latest/installation.html (accessed Oct. 02, 2022).

[4] A. Houghton, 'netifaces: Portable network interface information.' Accessed: Oct. 03, 2022. [Online]. Available: https://github.com/al45tair/netifaces

[5] 'Npcap: Windows Packet Capture Library & Driver'. https://npcap.com/#download (accessed Oct. 02, 2022).

[6] 'fouadtrad/EECE655-TCPSynFlooding', *GitHub*. https://github.com/fouadtrad/EECE655-TCPSynFlooding (accessed Oct. 03, 2022).