EECE 655 Assignment 2 Report

Prof. Imad ElHajj

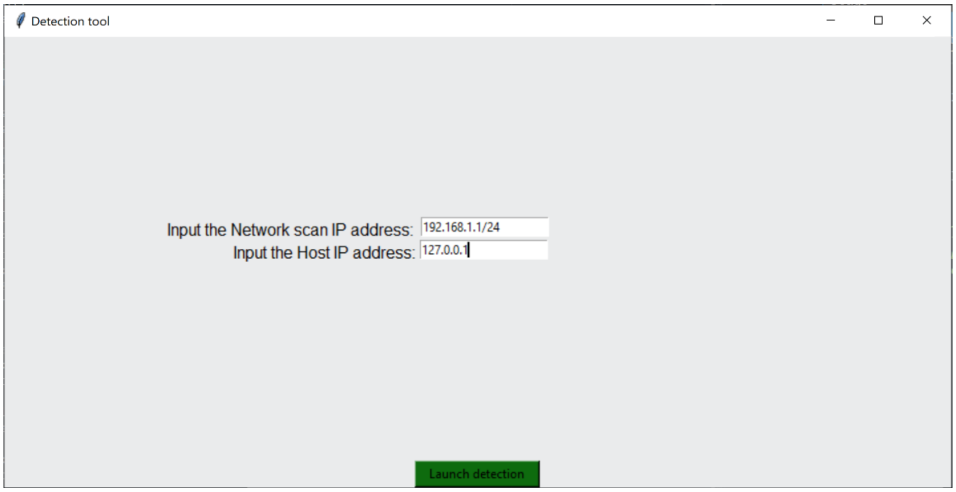
DoS Attack & Detection

Mohamad El Iskandarani, Naji Alayli, Jad Osman

Introduction

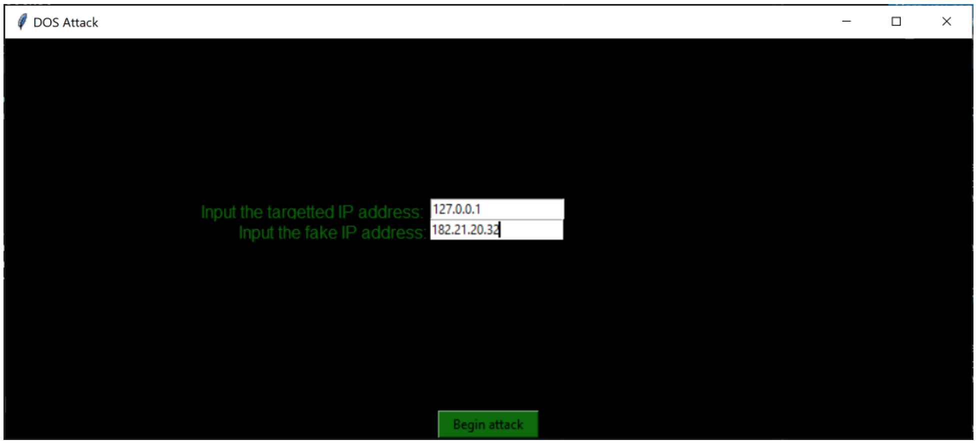
In our assignment, we implement a DoS attack and devise a tool to detect the attack. Each of the attack and tool have their own graphical user interface for a more user-friendly experience.

Graphical User Interface (Naji Alayli)



Input the Network scan IP address: 192.168.1.1/24
Input the Host IP address: 127.0.0.1

Detection Tool GUI



Input the targetted IP address: 127.0.0.1
Input the fake IP address: 182.21.20.32

Attack GUI

Both the attack GUI and the detecting GUI were made using tkinter in Python to make the attack and detecting tool easier to use.

Design Steps:

1. We created the GUI window, and we specified its name, size (which we chose to be 900x720), and background color.
2. We added two input boxes each that would take the target IP and fake IP for the attack and the host IP and the IP for the network scan.
3. We added a button to the GUI that, when clicked, would either launch the attack for the attacker GUI or launch the mock server code and begin searching for danger for the detecting tool GUI

The Attack (Jad Osman)

The attack is straightforward and not complicated. Basically, we are trying to overwhelm the victim by sending him HTTP requests (port 80) indefinitely, using a spoofed IP address.

Inside an infinite loop, we open a socket to connect to the targeted victim, and then we send him HTTP requests.

Using the GUI, we can input the chosen targeted IP and the spoofed IP.

Detection Tool (Mohamad El Iskandarani)

Brief:

The tool detects possible DoS attacks by counting the number of connections made by a single IP in a short period of time. Upon reaching a threshold of connections in a given time, the tool raises concern for a DoS attack.

Features:

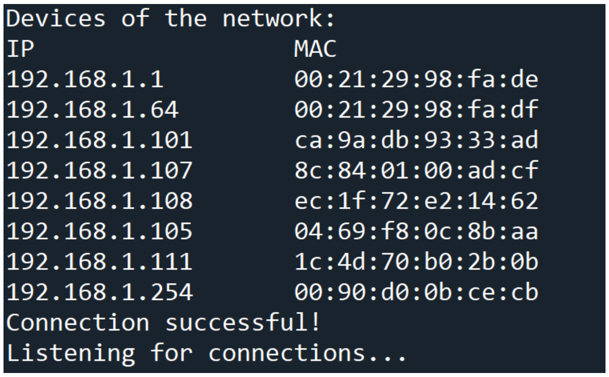
- Network scan: the first step is to scan the local network for all devices and store their IPs and MACs. This information will be used for possible attacker identification later on
- Listening and accepting connections: the server is constantly listening for connections
- Periodically flushed "recent connections" list: the server flushes the recent connections record every 5 seconds, since a DoS is considered a DoS if many connections are done in a short span in a manner that suffocates the resources. For example, a thousand connections over the span of a day is not flagged as a DoS, but a thousand connections in 1ms could be.
- Two thresholds: the first one (named "Alert") raises the administrator's attention to a possible DoS attack, while the other mark (named "Threshold") confirms a DoS attack. The values chosen in our code are arbitrary. Two thresholds allow the administrator to be more alert and prepared.

- IP comparison: upon confirming a DoS attack, the detection tool checks if the attacker IP is within the network or from outside the network for further identification. The attacker IP could be spoofed, but this step is a building block for possible future work.
- Connection number sensitive: the detection tool is sensitive towards the number of connections being made with the HTTP server (in our example). It does not detect spoofed IPs, so the fake IP being sent by the attacker to the HTTP server goes undetected by this tool, for it is responsible for detecting a DoS attack only. Future work could be done to enhance the tool's abilities.

Instructions:

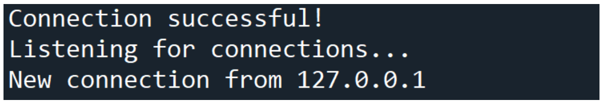
- Input the network scan IP, and host IP of the server
- Listen for connections

Results:



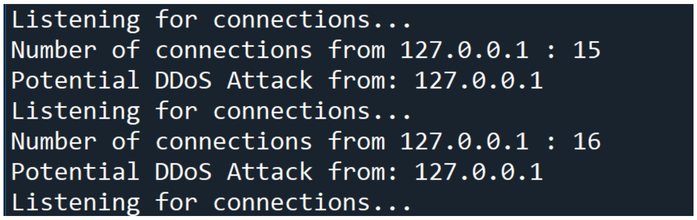
```
Devices of the network:
IP                  MAC
192.168.1.1         00:21:29:98:fa:de
192.168.1.64        00:21:29:98:fa:df
192.168.1.101       ca:9a:db:93:33:ad
192.168.1.107       8c:84:01:00:ad:cf
192.168.1.108       ec:1f:72:e2:14:62
192.168.1.105       04:69:f8:0c:8b:aa
192.168.1.111       1c:4d:70:b0:2b:0b
192.168.1.254       00:90:d0:0b:ce:cb
Connection successful!
Listening for connections...
```

1) Network scan and server initiation



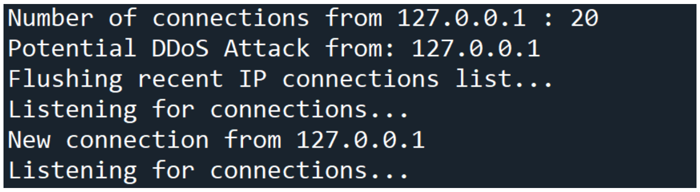
```
Connection successful!
Listening for connections...
New connection from 127.0.0.1
```

2) Server handling a normal number of requests



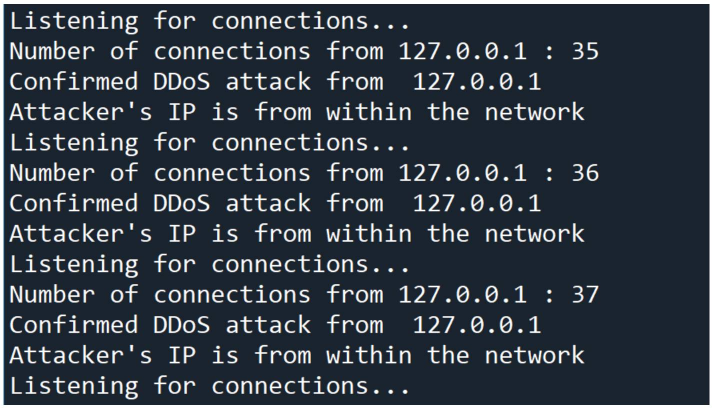
```
Listening for connections...
Number of connections from 127.0.0.1 : 15
Potential DDoS Attack from: 127.0.0.1
Listening for connections...
Number of connections from 127.0.0.1 : 16
Potential DDoS Attack from: 127.0.0.1
Listening for connections...
```

3) Server receiving a number of connections above the alert level



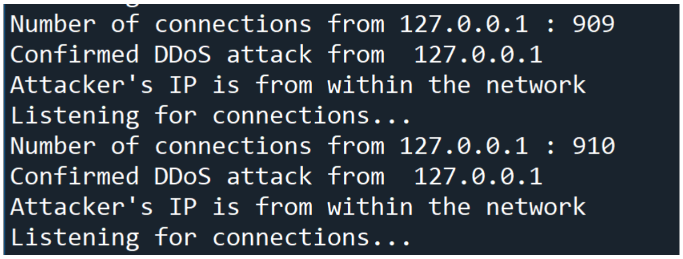
```
Number of connections from 127.0.0.1 : 20
Potential DDoS Attack from: 127.0.0.1
Flushing recent IP connections list...
Listening for connections...
New connection from 127.0.0.1
Listening for connections...
```

4) Server neglecting the previous alert level due to flushing of recent IPs



```
Listening for connections...
Number of connections from 127.0.0.1 : 35
Confirmed DDoS attack from  127.0.0.1
Attacker's IP is from within the network
Listening for connections...
Number of connections from 127.0.0.1 : 36
Confirmed DDoS attack from  127.0.0.1
Attacker's IP is from within the network
Listening for connections...
Number of connections from 127.0.0.1 : 37
Confirmed DDoS attack from  127.0.0.1
Attacker's IP is from within the network
Listening for connections...
```

5) Server detecting a DoS attack



```
Number of connections from 127.0.0.1 : 909
Confirmed DDoS attack from  127.0.0.1
Attacker's IP is from within the network
Listening for connections...
Number of connections from 127.0.0.1 : 910
Confirmed DDoS attack from  127.0.0.1
Attacker's IP is from within the network
Listening for connections...
```

6) The attack at full power