ECE 404 Homework #8

Due: Tuesday 3/21/2023 at 5:59PM

Introduction

The goal of this assignment is to give you a deeper grasp of TCP vulnerabilities and denial-of-service (DoS) attacks.

Problem Statement

Write a Python script that implements the SYN flood attack and SYN scanning to detect open ports. Your script should also spoof the host IP address. You will need to use `tcpdump`, or some equivalent tool, to monitor the network.

Program Requirements

Construct a class called `TcpAttack` that implements SYN scanning and a SYN flood attack.
A breakdown of how you might accomplish this is as follows:

1. Define the constructor of the class:
   - The constructor is an inbuilt function of the class that gets executed when creating new instances of that class, denoted by the function signature: `def __init__(self)`
   - Every instance of the `TcpAttack` class has two instance variables, `spoofIP` and `targetIP`. Thus, the constructor of this class accepts two strings as arguments, both in dotted decimal notation.
     - `spoofIP`: any IP that is not your own machine's
     - `targetIP`: the target of the SYN scan and SYN Flood Attack.
2. Define the `scanTarget` class method
   - This method accepts two arguments as integers:
     - `rangeStart`: The first port in the range of ports to be scanned
     - `rangeEnd`: The last port in the range of ports to be scanned.
   - This method scans the target computer for open ports in the range [`rangeStart`, `rangeEnd`] and writes all open ports detected into an output file called `openports.txt`.
   - The format of `openports.txt` should be one open port per line of the file, in ascending order.
3. Define the `attackTarget` class method
   - This method accepts two arguments as integers:
     - `port`: The port number on which the attack will be mounted on

  - `numSyn`: The number of `SYN` packets to be sent to the target on the specified port.
- This method first verifies if the specified port is open and then performs a DoS attack on the target using the port.
- If the port is open, it should perform the DoS attack and return 1. Otherwise, return 0 if the port is not open.
- To mount the attack, you will need to use a combination of functions from the `socket` and `scapy` libraries. Feel free to consult with the official documentation of these libraries as well as Prof. Kak's implementation in Lecture 16.15 for inspiration.
  - `scapy`: is a module that allows you to create and send network packets using Python.
  - `socket`: is a module allows you to set up a network connection with Python.

Shown below in the following code listing is a skeleton of what your `TcpAttack` class is expected to look like.

```
class TcpAttack:
    def __init__(self,spoofIP,targetIP):
        """
        spoofIP (str): IP address to spoof
        targetIP (str): IP address of the target computer to be attacked
        """

     def scanTarget(self,rangeStart,rangeEnd):
        """
        rangeStart (int): The first port in the range of ports being scanned.
        rangeEnd (int):   The last port in the range of ports being scanned
        No return value, but writes open ports to openports.txt
        """

    def attackTarget(self,port,numSyn):
        """
        port (int):   The port that the attack will use
        numSyn (int): Number of SYN packets to send to target IP address and port.
        If the port is open, perform DoS attack and return 1. Otherwise return 0.
        """
```

Mounting a SYN Flood Attack

Note that SYN flood attacks have become more difficult to mount over the years. As shown in Section 16.14 of the lecture notes, most ISPs now use BCP 38 ingress filtering to prevent spoofing over a router. Therefore you would have to do the spoofing attack between two computers (ask a friend if they could spare their's) on the same LAN where the packets would not go through a router.
It is acceptable if you do not actually manage to cause a DoS outside your LAN or do not have the means to do it with another computer on the same LAN. We are simply looking to see that a theoretical attack is implemented correctly (**you should still be able to test your program's port scanning, though**).

How to Tell That Your Program is Working

To test that the target machine is actually receiving packets, you should run `tcpdump` (or some equivalent program) while your script is running to see that you are actually sending packets to the target IP address (i.e. start tcpdump and then run your program). If you are using Windows, you can use **Wireshark** instead of `tcpdump` to look at the packets.
In the event that you are on a busy network, you can use `tcpdump` to selectively sniff packets as outlined in Lecture 16. To further avoid clutter, you can optionally turn off all other applications connecting to the internet. As mentioned below, you will include output from these programs in your homework submission.
If you do not have access to another computer to test on, you can try using your ECN account's public IP address to send packets to (this should work at least for port scanning). While you cannot run `tcpdump` on your ECN account (due to the need for superuser privileges to run it), you can run it on the machine running your script to see that there are outgoing packets with the target IP address as their destination.

How Your Code Will Be Tested

Your code will be tested with a script similar to the one below:

```
from TcpAttack import * #Your TcpAttack class should be named as TcpAttack

# Will contain actual IP addresses in real script
spoofIP='10.1.1.1' ; targetIP='10.1.1.2'
rangStart=<int> ; rangeEnd=<int> ; port=<int>
Tcp = TcpAttack(spoofIP,targetIP)
Tcp.scanTarget(rangeStart, rangeEnd)

if Tcp.attackTarget(port,10):
   print('port was open to attack')
```

Remember, in the event that the user wants to scan the computer for open ports, your script should subsequently report the open ports in an output file called `openports.txt`. In the event that the user wants to attack the computer, your script should first check if the port (passed as an argument to attackTarget) is open.

Submission Instructions

- For this homework you will be submitting a zip file titled `hw08_<last name>_<first name>.zip`, which consists of:
  - A pdf titled `hw08_<last name>_<first name>.pdf` containing:
    - Output (e.g. screenshots) from **tcpdump** (or equivalent program) at least for the port scanning part of your program. **Your PDF should indicate in the tcpdump output (e.g. highlight, circle, etc.) which packets were sent as a result of the program you wrote.**
  - The file `TcpAttack.py` containing your code for the programming problem.
- Please include comments in your code.