ECE 404 Homework #8

Due: Thursday 3/26/2020 at 4:29PM

Introduction

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

Problem Statement

Write a Perl or 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

Create a class called **TcpAttack** that takes in the parameters in the constructor as described below:

```
#rangeStart:
class TcpAttack:
       #spoofIP: String containing the IP address to spoof
       #targetIP: String containing the IP address of the target computer to attack
       def __init__(self,spoofIP,targetIP):
```

This class should have the following methods:

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

This method will scan the target computer for open ports, using the range of ports passed, and write ALL the open ports found into an output file called openports.txt. The format of openports.txt should be one open port number per line of the file, in ascending order.

```
#port: Integer designating the port that the attack will use
#numSyn: Integer of SYN packets to send to target IP address at the given port
#If the port is open, perform DoS attack and return 1. Otherwise return 0.
    def attackTarget(self,port,numSyn):
```

This method first verifies 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 passed is not open). For the purposes of this assignment, it is only necessary

to send a number of SYN packets equal to `numSyn`, rather than looping infinitely. You can look at the scripts listed in section 16.14 of the lecture notes for inspiration.

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 wouldn't 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 start tcpdump (or some equivalent program) before running your script to see that you are actually sending packets to the target IP address. 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 don't 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 can't 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
spoofIP='string' ; targetIP='string' #Will contain actual IP addresses in real script
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.

Important Notes

- There are some prerequisite software packages you need to install in order to use scapy. Also, you will need to monitor network traffic on a machine other than your ECN account (since you don't have superuser privileges on that account). **Therefore it is best to set it the necessary software sooner rather than later in case you encounter any problems**.
- If using Python, use the socket and scapy modules to handle raw socket packets. The socket module allows you to set up a network connection with Python. Scapy is a module that allows you to create and send network packets using Python. Useful links for installing scapy can be found in the Resources section of the ECE 404 website.
- If using Perl, use the `Net::RawIP` and `IO::Socket` modules from `www.cpan.org`.

Submission Requirements

- Your submission should include your code and a 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.**
- Please include comments in your code.
- Please make sure your shebang line works on ECE Grid (or ECN in general) by running the command `dos2unix TcpAttack.py; chmod +x TcpAttack.py; ./TcpAttack.py` in the terminal.

Electronic Turn-in

`turnin -c ece404 -p hw08 TcpAttack.pl hw08.pdf` (if using Perl)
`turnin -c ece404 -p hw08 TcpAttack.py hw08.pdf` (if using Python)