week 6 lab 2

Kevin Turkington

I. Question 2

Based on the program I created I noticed alot of traffic going to specfically TCP 139/5631 and UDP 5001 for R.csv. both TCP ports from speed guide.net say these are windows specfic and the port being abused by UDP 5001 is common used by trojans. Which indicates the the machine is used for at home use. And for O.csv there was alot of traffic for Ports 21563 UDP and TCP 33251 both of which are unassigned with a collection of random requests to other places indicating it is a data center.

II. Question 4

After running my countip function on O.csv has many connections reaching out to a multitude of different ip's, but the most prominent one being from the domain 192.245.12.XXX which indicates a data center redirecting traddic around. As for R.csv a majority of domains recieving traffic have been from 10.5.63.XXX, from a quick google search shows this ip is designated for a private address within a network.

III. Question 5

R.csv: 10.5.63.XXX dominates the most traffic
O.csv: 192.245.12.XXX dominates the most traffic

IV. Question 7

O.csv: 207.182.40.XXX specfically 207.182.40.40 has been using protocol 1 (ICMP) which is the primary communication for data centers according to week 6 lectures.
R.csv: 234.142.142.142 which serves as a multicasting address most likely for a home use computer connecting to a router.

V. Question 8

The OSPF information of the top most used ips for both csv files help inforce my answer from question 2 because one uses mainly default and smaller host number ip addresses often provided from a local home network. While the other utilizes an ipv4 range that can host much more connecitons.

VI. Question 10

R.csv:

- 10.5.63.1
- 10.5.63.10
- 10.5.63.11
- 10.5.63.12 -¿ DNS
- 10.5.63.14
- 10.5.63.15
- 10.5.63.16
- 10.5.63.17
- 10.5.63.18
- 10.5.63.27

O.csv:

- 12.4.100.38
- 130.76.32.144
- 130.76.64.14
- 134.160.240.1
- 140.247.177.182
- 141.158.4.203
- 147.240.170.29
- 192.165.221.230
- 192.165.221.231
- 192.245.12.237 -¿ DNS

VII. Question 11

both answers from question five stay the same. For O.csv 192.245.12.XXX dominates and for R.csv 10.5.63.XX dominates.

VIII. Code

```
from CSVPacket import Packet, CSVPackets
import sys, re

IPProtos = [0 for x in range(256)]
numBytes = 0
numPackets = 0

csvfile = open(sys.argv[1],'r')

class ipMap(object):
    def __init__(self, ipaddr):
        self.addr = ipaddr
        self.ports = []
```

```
    def addPort(self,pType,pNum):
        port = "%s/%d"%(pType,pNum)
        if port not in self.ports:
            self.ports.append(port)
        def totalPorts(self):
            return len(self.ports)
        def portList(self):
            return self.ports

portNumMax = 65535
if __name__ == "__main__":
    if "-stats" in sys.argv: # clean up with dictionary
        portListTCP = [0]*portNumMax
        postListUDP = [0]*portNumMax
        for pkt in CSVPackets(csvfile):
            if (pkt.proto & 0xff) == 6:
                portListTCP[pkt.tcpdport]+=1
            if (pkt.proto & 0xff) == 17:
                portListUDP[pkt.udpdport]+=1
        for i in range(1,portNumMax)
            if portListTCP[i] != 0:
                print "TCP Port: %d -> %d" %(o,[portListTCP[i]])
        for i in range(1,portNumMax)
            if postListUDP[i] != 0:
                print "UDP Port: %d -> %d" %(o,[postListUDP[i]])
    elif "-countip" in sys.argv:
        ipAddrList = {}
        for pkt in CSVPackets(csvfile):
            tcp = str(pkt.ipsrc)
            udp = str(pkt.ipdst)
            if not tcp in ipAddrList:
                ipAddrList[tcp] = (1,(pkt.proto & 0xff))
            elif tcp in ipAddrList:
                count = ipAddrList[tcp][0]
                ipAddrList[tcp] = (count + 1,(pkt.proto & 0xff))

            if not udp in ipAddrList:
                ipAddrList[udp] = (1,(pkt.proto & 0xff))
            elif udp in ipAddrList:
                count = ipAddrList[udp][0]
                ipAddrList[udp] = (count + 1,(pkt.proto & 0xff))
        for key,val in sorted(ipAddrList.iteritems(), key=lambda (k,v): (
            print "%s : usage: %d : proto: %d" %(key,val[0],val[1])
    elif "connto" in sys.argv:
```

```
ipObjs = {}
bcastAddr = re.compile('.*/.255$')#ignore broadcast addresses
for pkt in CSVPackets(csvfile):
    ipdst = str(pkt.ipdst)
    if not ipdst in ipObjs and not bcastAddr.match(ipdst):
        ipObjs[ipdst]= ipMap(ipdst)
    if ipdst in ipObjs and (pkt.proto & 0xff) == 6:
        ipObjs[ipdst].addPort("TCP",pkt.tcpdport)
    elif ipdst in ipObjs and (pkt.proto & 0xff) == 17:
        ipObjs[ipdst].addPort("UDP",pkt.udpdport)
for ip,obj in sorted(ipObjs.iteritems(), reverse=True):
    print "ipdst %s has %d distinct ipsrc on ports: " %(ip,obj.to
    for port in obj.portList():
        sys.stdout.write(port +" ,")
    print("\n")
```