OREGON STATE UNIVERSITY

CS 373

WINTER 2019

Week 6 Lab 2

Author:
Thomas Noelcke

Instructor:
D. Kevin McGrath

I. Overview

In this document I go through the different questions in the lab. I will have a section where I answer lab questions directly. I will also have a section that includes the code that I wrote for the lab and finally I will have a conclusion section where I discuss any conclusions that may have been derived from the lab.

II. lab questions

A. Question 2

Based on this information, characterize the main functions on each network? What kind of network is it?

1) Network R: After running the stat command on network R I noticed that there were a lot of request on port TCP port 139 the SMB port. I also noticed that there were a lot of request for port 80, 110 and UDP port 53. Port 80 is used for web traffic. TCP port 110 is used for email port 53 is used for DNS. Based on the type of traffic I'm seeing in this network log it seems likely that this is some soft of office network where people are reading emails and sharing files.

2) Network O: For network O I noticed a large number of request for ports 25 and 80 as well as port 500 for UDP. I also notice that port 445 on TCP doesn't have a ton of traffic on it but it is non zero. Port 25 is designated for SMTP. Given that there are over 200k requests to this port on this network I think it is very likely that this network is designated for email.

B. Question 4

Run your countip script on R and O data. Does this inform your answer in [2]?

1) Network R: For network R I noticed that the network prefix that primarily dominated the network traffic was mostly 10.5.63.xxx. A quick google search shows that this address is designated for use with in a private network. This is the type of traffic I would expect for an office of some sort.

2) Network O: On network O I noticed that ip 199.249.33.xxx dominated the logs. Its also important to note that 207.182.xx.xxx was also a big feature of the logs. This is not a private IP. This means it could be possible that these addresses are used for passing along email but it isn't the smoking gun that the port information gave us in the above sections.

C. question 5

10..5.63.xxx dominated the traffic on network R and 199.249.33.xxx dominated network O.

D. Question 7

E. Network R

Network R didn't have any logs with any of these protocols in it. This seems indicative of a network that primarily handles local traffic.

1) Network O: There were very few requests with the OSPF protocol ID. IO don't think that network O is heavily using routers to handle requests. However, all of the request using protocol 89 were from more or less the same IP address. There are also very view request from the IPSEC protocol and most of these are from various addresses. Finally, there were even fewer request from the GRE protocol version. I'm beginning to think that network O has very little direct interaction with people much like an Email server.

F. question 8

I think this does help to inform some of the answer to question 2. However, I think it takes a combination of all of this information to answer question 2.

G. Question 10

1) Network R:

- 32.98.255.112
- 234.42.42.42
- 234.142.142.142
- 216.101.171.2 - Email Server
- 209.67.181.20
- 209.67.181.11
- 208.10.192.202
- 208.10.192.176
- 208.10.192.175
- 208.10.192.161
- 207.5.63.20
- 207.5.63.2 - DNS
- 207.46.143.254
- 207.44.165.251 - Email
- 206.253.217.13
- 206.170.168.217 - Printer
- 206.13.28.62 - Email
- 204.71.201.113
- 204.71.200.246

2) Network O: For network O my top twenty IP's didn't have any interesting ports listed for them. So i decide to save my self some time and not list them.

H. Question 11

The ip's I found in question 10 don't seem to be related to the IP's found in question 5.

III. 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): (v,k), reverse='
            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.totalPorts)
            for port in obj.portList():
                sys.stdout.write(port +”␣,”)
            print(”\n”)
```

IV. Bibliography