Nicolas Barraclough
11/15/2020
CS373 - Defense Against the Dark Arts

Homework 3

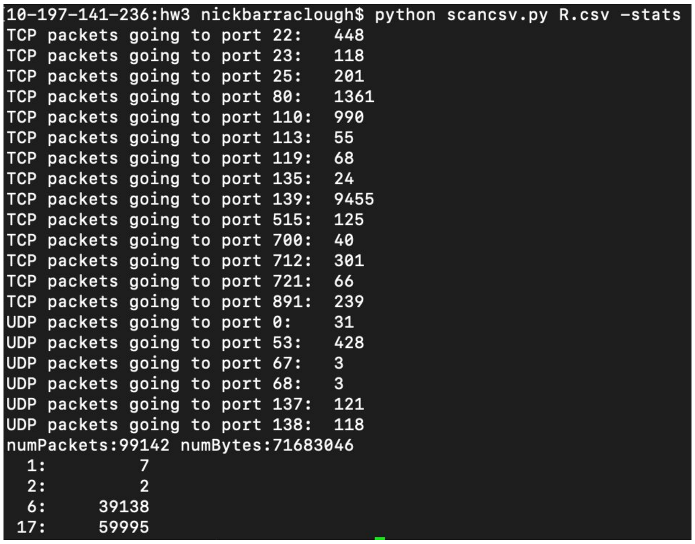
I ran the starter script on both .csv files to ensure that they initially functioned properly. I will be using the python script because I am most compatible with python. I am fairly familiar with IP protocols, but I don't know about IGMP, so I will first research that.

Here is what I could find about IGMP:

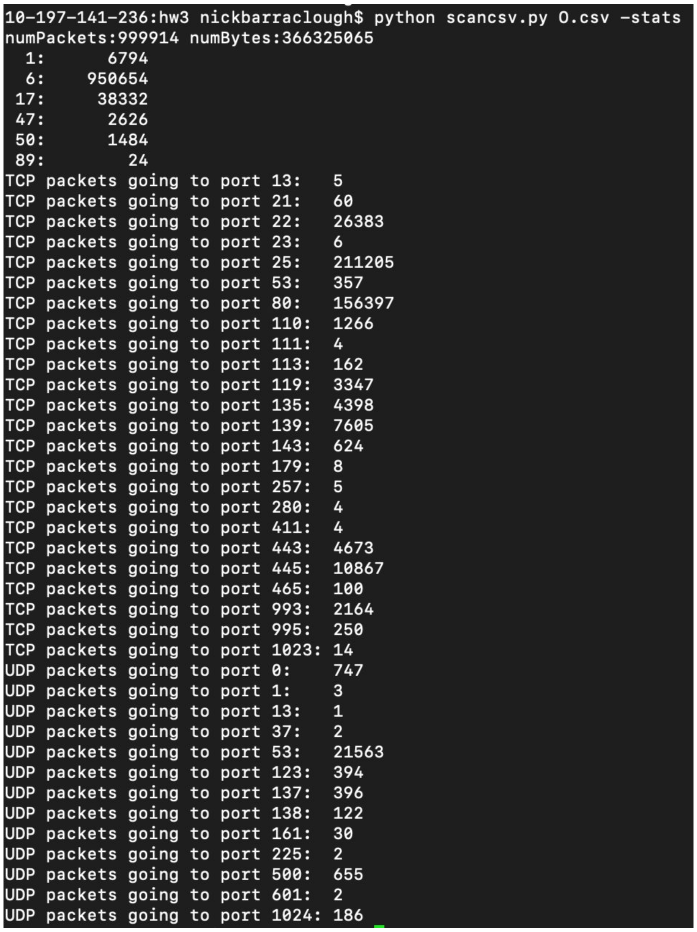
- Stands for Internet Group Management Protocol
- Is used to establish multicast group memberships for hosts and adjacent routers on IPv4 networks.
- After reading the very next part of the Lab, a general description describes the use as router-to-router communication.

Find Statistics on TCP and UDP Services

1. **Extend your script's statistics gathering to count the use of all well-known destination port numbers for TCP and UDP (ports 1-1024). For example, you should be able to look up in your output how many TCP packets have destination port 80 and how many UDP packets have destination port 53. Run your new script on R and O data. Enable this function using a '-stats' flag (i.e., the script should have no output unless there is a -stats flag in the command line).**



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -stats
TCP packets going to port 22:   448
TCP packets going to port 23:   118
TCP packets going to port 25:   201
TCP packets going to port 80:   1361
TCP packets going to port 110:  990
TCP packets going to port 113:  55
TCP packets going to port 119:  68
TCP packets going to port 135:  24
TCP packets going to port 139:  9455
TCP packets going to port 515:  125
TCP packets going to port 700:  40
TCP packets going to port 712:  301
TCP packets going to port 721:  66
TCP packets going to port 891:  239
UDP packets going to port 0:    31
UDP packets going to port 53:   428
UDP packets going to port 67:   3
UDP packets going to port 68:   3
UDP packets going to port 137:  121
UDP packets going to port 138:  118
numPackets:99142 numBytes:71683046
  1:         7
  2:         2
  6:     39138
 17:     59995
```



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -stats
numPackets:999914 numBytes:366325065
  1:      6794
  6:    950654
 17:     38332
 47:      2626
 50:      1484
 89:        24
TCP packets going to port 13:   5
TCP packets going to port 21:   60
TCP packets going to port 22:   26383
TCP packets going to port 23:   6
TCP packets going to port 25:   211205
TCP packets going to port 53:   357
TCP packets going to port 80:   156397
TCP packets going to port 110:  1266
TCP packets going to port 111:  4
TCP packets going to port 113:  162
TCP packets going to port 119:  3347
TCP packets going to port 135:  4398
TCP packets going to port 139:  7605
TCP packets going to port 143:  624
TCP packets going to port 179:  8
TCP packets going to port 257:  5
TCP packets going to port 280:  4
TCP packets going to port 411:  4
TCP packets going to port 443:  4673
TCP packets going to port 445:  10867
TCP packets going to port 465:  100
TCP packets going to port 993:  2164
TCP packets going to port 995:  250
TCP packets going to port 1023: 14
UDP packets going to port 0:    747
UDP packets going to port 1:    3
UDP packets going to port 13:   1
UDP packets going to port 37:   2
UDP packets going to port 53:   21563
UDP packets going to port 123:  394
UDP packets going to port 137:  396
UDP packets going to port 138:  122
UDP packets going to port 161:  30
UDP packets going to port 225:  2
UDP packets going to port 500:  655
UDP packets going to port 601:  2
UDP packets going to port 1024: 186
```

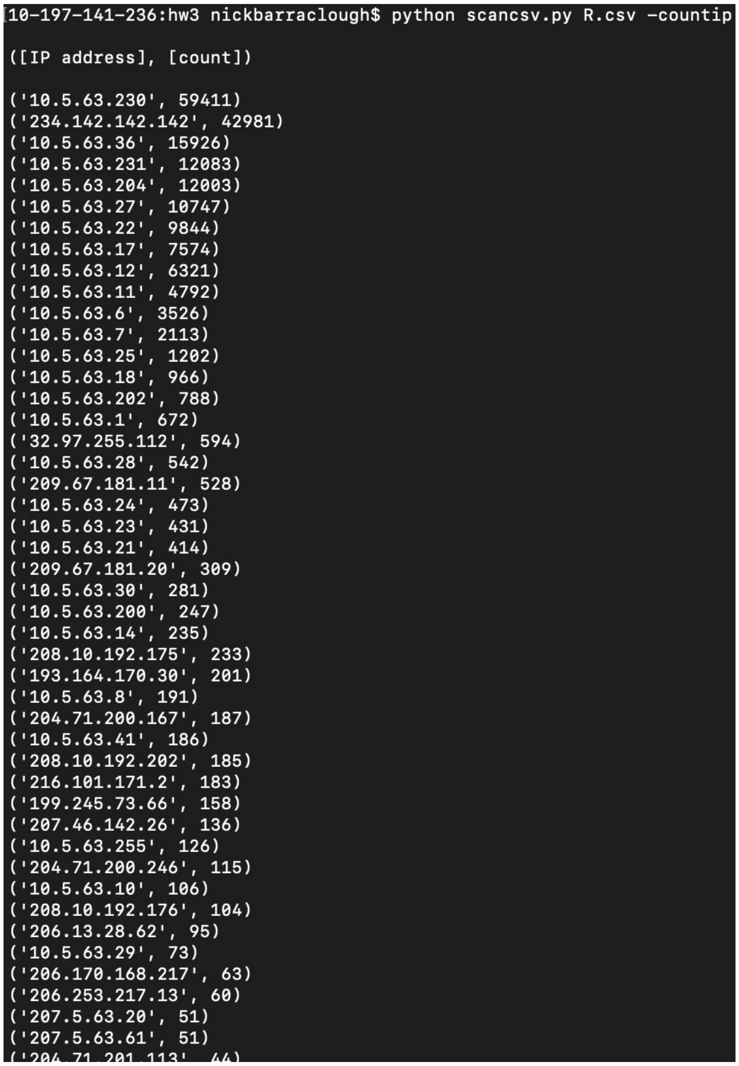
2. **Based on this information, characterize the main functions on each network. What kind of a network is it? (e.g., work, home, data center, ISP)**

The packets captured in R.csv show that it most likely a home or work network, as there are a significant number of packets on ports 80, 110, and especially 139. Port 80 is for HTTP (web), port 110 is for POP3 (email) and 139 is for file sharing and printing. If I had to guess, I would suspect this is a work network.

The packets captured in O.csv show that it is most likely a workplace network. There are a significant number of packets coming in at ports 25, 80, and a lesser, yet still significant number of packets coming to ports 22, 445, and 53. The packets coming to port 53 are UDP packets, which seems to be linked to firewalls. TCP port 445 is used for direct TCP/IP MS Networking access, port 22 and 25 are used for SSH and STMP respectively. All these are indicative of a workplace network.

Investigate IP Addresses

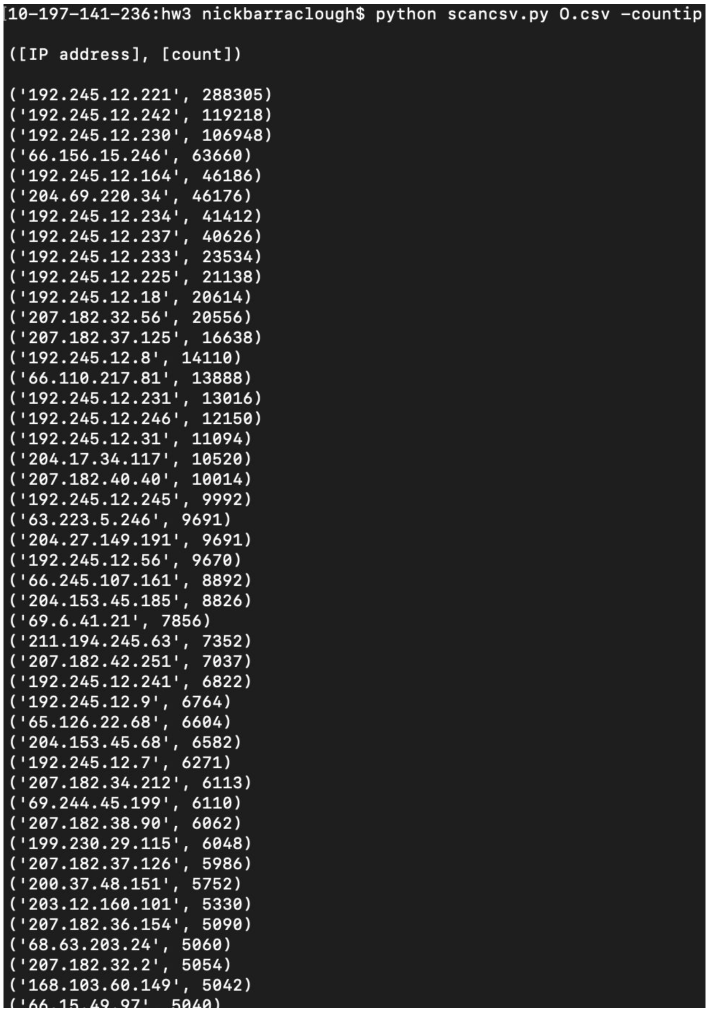
3. **Add to your script an option called "-countip" which creates a list of distinct IP addresses with their usage counts. Sort the list by the usage count, not by the IP address.**
4. **Run your countip script on R and O data. Does this inform your answer in [2]?**



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -countip

([IP address], [count])

('10.5.63.230', 59411)
('234.142.142.142', 42981)
('10.5.63.36', 15926)
('10.5.63.231', 12083)
('10.5.63.204', 12003)
('10.5.63.27', 10747)
('10.5.63.22', 9844)
('10.5.63.17', 7574)
('10.5.63.12', 6321)
('10.5.63.11', 4792)
('10.5.63.6', 3526)
('10.5.63.7', 2113)
('10.5.63.25', 1202)
('10.5.63.18', 966)
('10.5.63.202', 788)
('10.5.63.1', 672)
('32.97.255.112', 594)
('10.5.63.28', 542)
('209.67.181.11', 528)
('10.5.63.24', 473)
('10.5.63.23', 431)
('10.5.63.21', 414)
('209.67.181.20', 309)
('10.5.63.30', 281)
('10.5.63.200', 247)
('10.5.63.14', 235)
('208.10.192.175', 233)
('193.164.170.30', 201)
('10.5.63.8', 191)
('204.71.200.167', 187)
('10.5.63.41', 186)
('208.10.192.202', 185)
('216.101.171.2', 183)
('199.245.73.66', 158)
('207.46.142.26', 136)
('10.5.63.255', 126)
('204.71.200.246', 115)
('10.5.63.10', 106)
('208.10.192.176', 104)
('206.13.28.62', 95)
('10.5.63.29', 73)
('206.170.168.217', 63)
('206.253.217.13', 60)
('207.5.63.20', 51)
('207.5.63.61', 51)
('204.71.201.113', 44)
```

Yes, this discovery does support my answer in [2]. Judging by the number of 10.x.x.x IP addresses, this is most likely a workplace with 20+ local machines on a class A network.

O.csv: The number of IPs on this network is massive! The screenshot (right) shows about 5% (maybe) of the total



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -countip

([IP address], [count])

('192.245.12.221', 288305)
('192.245.12.242', 119218)
('192.245.12.230', 106948)
('66.156.15.246', 63660)
('192.245.12.164', 46186)
('204.69.220.34', 46176)
('192.245.12.234', 41412)
('192.245.12.237', 40626)
('192.245.12.233', 23534)
('192.245.12.225', 21138)
('192.245.12.18', 20614)
('207.182.32.56', 20556)
('207.182.37.125', 16638)
('192.245.12.8', 14110)
('66.110.217.81', 13888)
('192.245.12.231', 13016)
('192.245.12.246', 12150)
('192.245.12.31', 11094)
('204.17.34.117', 10520)
('207.182.40.40', 10014)
('192.245.12.245', 9992)
('63.223.5.246', 9691)
('204.27.149.191', 9691)
('192.245.12.56', 9670)
('66.245.107.161', 8892)
('204.153.45.185', 8826)
('69.6.41.21', 7856)
('211.194.245.63', 7352)
('207.182.42.251', 7037)
('192.245.12.241', 6822)
('192.245.12.9', 6764)
('65.126.22.68', 6604)
('204.153.45.68', 6582)
('192.245.12.7', 6271)
('207.182.34.212', 6113)
('69.244.45.199', 6110)
('207.182.38.90', 6062)
('199.230.29.115', 6048)
('207.182.37.126', 5986)
('200.37.48.151', 5752)
('203.12.160.101', 5330)
('207.182.36.154', 5090)
('68.63.203.24', 5060)
('207.182.32.2', 5054)
('168.103.60.149', 5042)
('66.15.49.97', 5040)
```

number of IP addresses. Most of the addresses are 192.245.12.x, which means that this is a class C network. The traffic in this capture are communications with certain telecommunications companies which are explained in [5].

5. **Attempt to determine the network number (network prefix) that seems to dominate the traffic.**

   R.csv

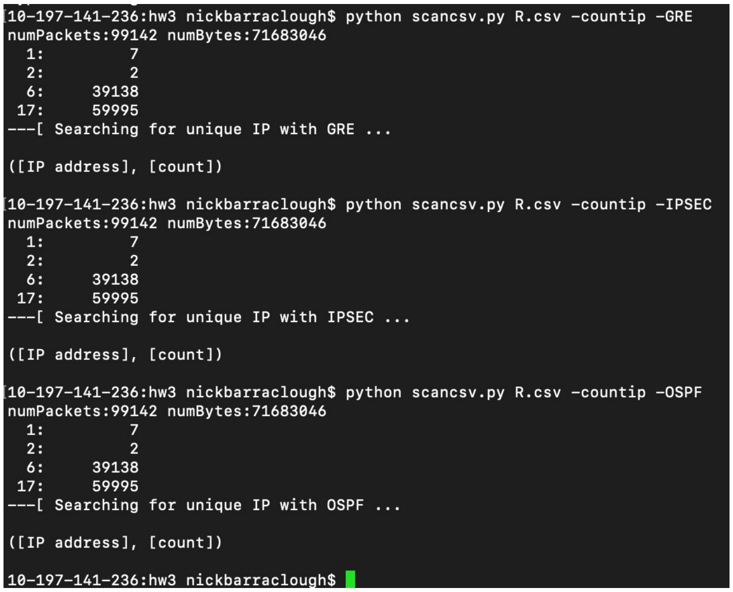
   234.142.142.142 - Multicast

   O.csv

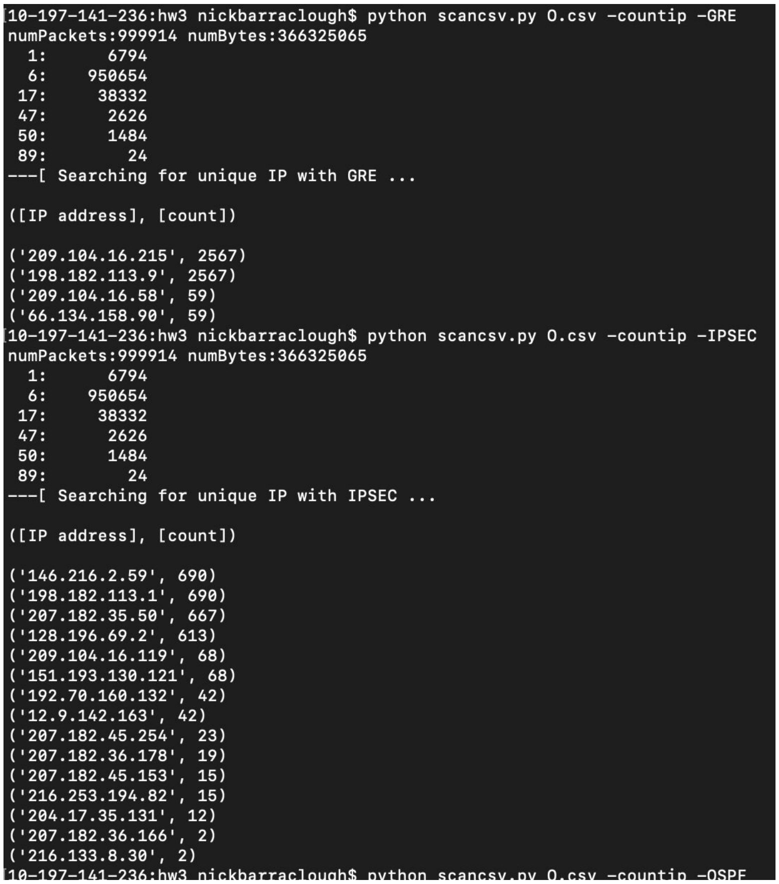
   66.156.15.246 - AT&T

There are some IP protocols that are typically used between routers or other special networking devices. Traffic from these protocols can identify the infrastructure of the network under observation.

6. **Generate sorted output from ‘-countip’ for the IP protocols to identify all the IP addresses that use:**
   a. **GRE (Generic Routing Encapsulation) – this is used to create tunnels between networks with overlapping address spaces. It is also the base protocol for PPTP, a remote access mechanism.**
   b. **IPSEC – this is the protocol that creates virtual private networks, creating an overlay network structure on top of the Internet. Most IPSEC is router-router these days.**
   c. **OSPF – Open Shortest Path First routing protocol. This is the ‘standard’ routing protocol for Internet routers, allowing them to discover the topology and choose the best routing paths as connections between routers appear and disappear.**
   · Hint: create a new protocol argument to filter the data to ‘-countip’ to just include lines for these protocols. Alternately, use the GREP pipeline in the example above, for IGMP traffic, and change ‘2’ to the right protocol number.



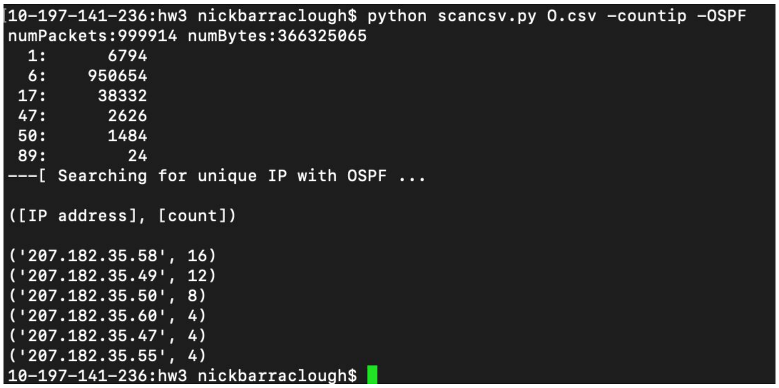
```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -countip -GRE
numPackets:99142 numBytes:71683046
  1:         7
  2:         2
  6:     39138
 17:     59995
---[ Searching for unique IP with GRE ...

([IP address], [count])

10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -countip -IPSEC
numPackets:99142 numBytes:71683046
  1:         7
  2:         2
  6:     39138
 17:     59995
---[ Searching for unique IP with IPSEC ...

([IP address], [count])

10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -countip -OSPF
numPackets:99142 numBytes:71683046
  1:         7
  2:         2
  6:     39138
 17:     59995
---[ Searching for unique IP with OSPF ...

([IP address], [count])

10-197-141-236:hw3 nickbarraclough$ 
```



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -countip -GRE
numPackets:999914 numBytes:366325065
  1:      6794
  6:    950654
 17:     38332
 47:      2626
 50:      1484
 89:        24
---[ Searching for unique IP with GRE ...

([IP address], [count])

('209.104.16.215', 2567)
('198.182.113.9', 2567)
('209.104.16.58', 59)
('66.134.158.90', 59)
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -countip -IPSEC
numPackets:999914 numBytes:366325065
  1:      6794
  6:    950654
 17:     38332
 47:      2626
 50:      1484
 89:        24
---[ Searching for unique IP with IPSEC ...

([IP address], [count])

('146.216.2.59', 690)
('198.182.113.1', 690)
('207.182.35.50', 667)
('128.196.69.2', 613)
('209.104.16.119', 68)
('151.193.130.121', 68)
('192.70.160.132', 42)
('12.9.142.163', 42)
('207.182.45.254', 23)
('207.182.36.178', 19)
('207.182.45.153', 15)
('216.253.194.82', 15)
('204.17.35.131', 12)
('207.182.36.166', 2)
('216.133.8.30', 2)
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -countip -OSPF
```

- None from R.csv (left)

- O.csv (right)



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -countip -OSPF
numPackets:999914 numBytes:366325065
  1:      6794
  6:    950654
 17:     38332
 47:      2626
 50:      1484
 89:        24
---[ Searching for unique IP with OSPF ...

([IP address], [count])

('207.182.35.58', 16)
('207.182.35.49', 12)
('207.182.35.50', 8)
('207.182.35.60', 4)
('207.182.35.47', 4)
('207.182.35.55', 4)
10-197-141-236:hw3 nickbarraclough$
```

7. **Find another network prefix that also seems to be associated with this traffic.**

   R.csv
   32.97.255.112 - AT&T
   O.csv
   207.182.32.56 - Opus-One Catering Company

8. **Does the OSPF information inform your answer to question 2?**

   Open Shortest Path First (OSPF) was designed as an interior gateway protocol (IGP), for use in an autonomous system such as a local area network (LAN). Given the search result for OSPF, that network O from O.csv appears to be the LAN network for Opus-One Catering Company.

9. **Add an option to your script '-connto', which counts the number of packets sent to each service (ports 1-1024) on the network. For example, a dictionary maps each ipdst to the tuple <proto, dport>, where proto is tcp or udp, based on the IP protocol (6 or 17) and dport is the value of tcpdport or udpdport.**
   - **Sort the output by the number of distinct source IP addresses – source port combinations, so that servers which serve a lot of different connections all cluster at one end of the output.**
   - **For output, generate a summary line that shows, for each destination IP address, how many distinct source IP addresses accessed it, and what ports were referenced:**

     ipdst 1.2.3.4 has 334 distinct ipsrc on ports: udp/53, tcp/80, tcp/443
     ipdst 5.6.7.8 has 335 distinct ipsrc on ports: tcp/22, tcp/25
     …

   Since lab time is short, here are some programming hints you may wish to use:

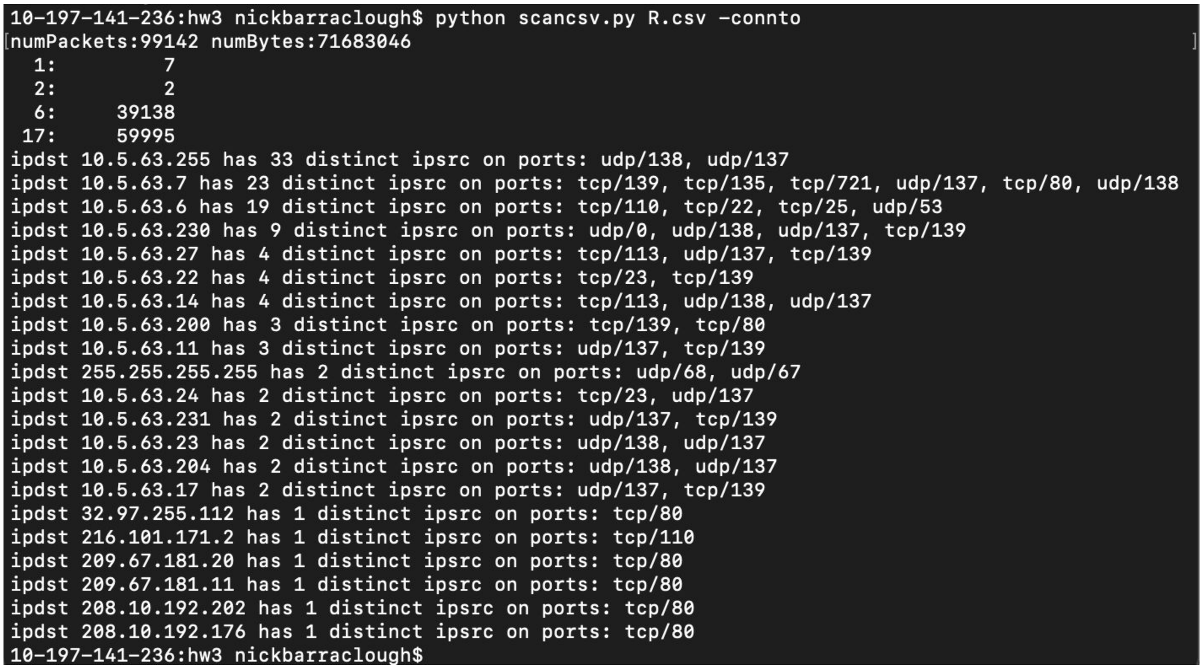
   a. **To create the ports output, create a set for each "ipdst" that contains the string "udp/" or "tcp/" appended to the port number (e.g., udp/53, tcp/360). (In languages without an explicit set class, use a dictionary or hash where each entry maps to TRUE or 1)**
   b. **You can use the same trick to compute the distinct ipsrc for the summary line. In this case, put the ipsrc in the string from [a], as ipsrc-proto/port. For example, dict['1.2.3.4'] is a set containing 205.9.3.55-udp/53, means that 205.9.3.55 connects to 1.2.3.4 on UDP port 53.**

c. **You can use leading zeros to make these formats sort correctly without fuss, such as: "tcp/00033" or "udp/00721". However, you must still arrange for the program output not to have leading zeros.**

10. Run your -connto option on R and O data (ignore anything that ends in .255 – this is a broadcast address). Does this suggest a set of servers to you?

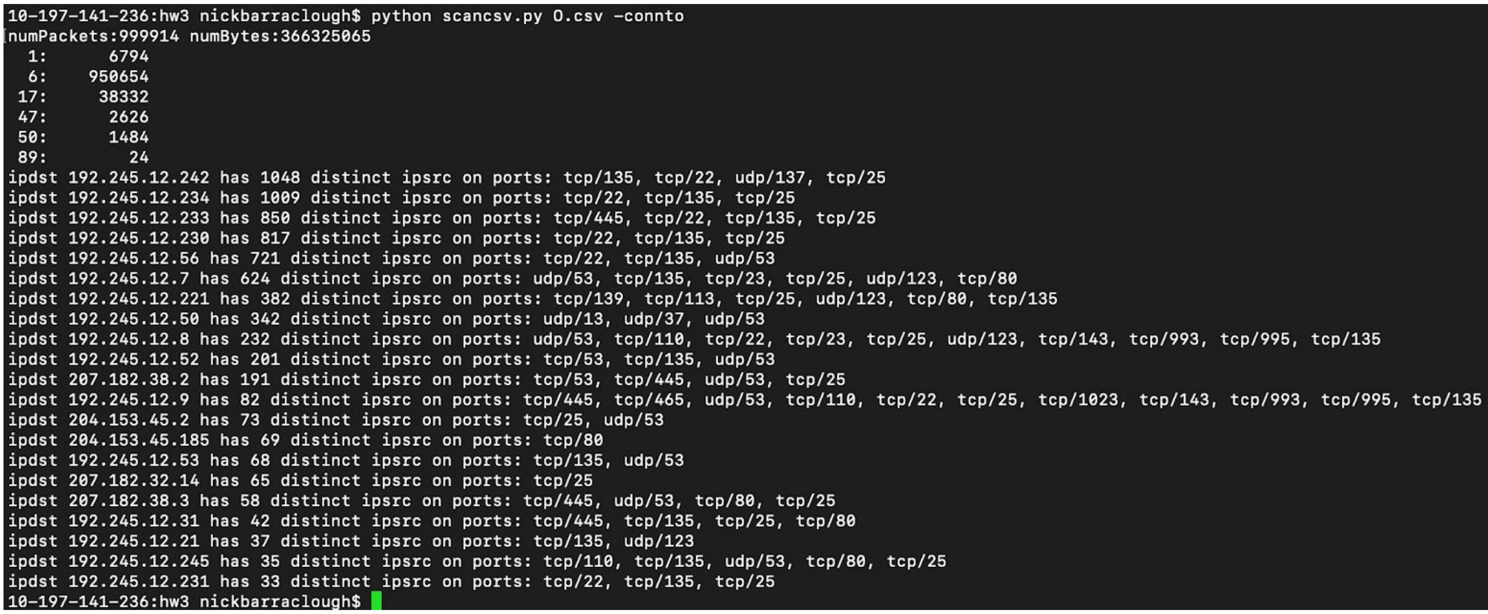
a. **Return the top 20 servers from your 'connto' output.**

R.csv:



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py R.csv -connto
numPackets:99142 numBytes:71683046
  1:         7
  2:         2
  6:     39138
 17:     59995
ipdst 10.5.63.255 has 33 distinct ipsrc on ports: udp/138, udp/137
ipdst 10.5.63.7 has 23 distinct ipsrc on ports: tcp/139, tcp/135, tcp/721, udp/137, tcp/80, udp/138
ipdst 10.5.63.6 has 19 distinct ipsrc on ports: tcp/110, tcp/22, tcp/25, udp/53
ipdst 10.5.63.230 has 9 distinct ipsrc on ports: udp/0, udp/138, udp/137, tcp/139
ipdst 10.5.63.27 has 4 distinct ipsrc on ports: tcp/113, udp/137, tcp/139
ipdst 10.5.63.22 has 4 distinct ipsrc on ports: tcp/23, tcp/139
ipdst 10.5.63.14 has 4 distinct ipsrc on ports: tcp/113, udp/138, udp/137
ipdst 10.5.63.200 has 3 distinct ipsrc on ports: tcp/139, tcp/80
ipdst 10.5.63.11 has 3 distinct ipsrc on ports: udp/137, tcp/139
ipdst 255.255.255.255 has 2 distinct ipsrc on ports: udp/68, udp/67
ipdst 10.5.63.24 has 2 distinct ipsrc on ports: tcp/23, udp/137
ipdst 10.5.63.231 has 2 distinct ipsrc on ports: udp/137, tcp/139
ipdst 10.5.63.23 has 2 distinct ipsrc on ports: udp/138, udp/137
ipdst 10.5.63.204 has 2 distinct ipsrc on ports: udp/138, udp/137
ipdst 10.5.63.17 has 2 distinct ipsrc on ports: udp/137, tcp/139
ipdst 32.97.255.112 has 1 distinct ipsrc on ports: tcp/80
ipdst 216.101.171.2 has 1 distinct ipsrc on ports: tcp/110
ipdst 209.67.181.20 has 1 distinct ipsrc on ports: tcp/80
ipdst 209.67.181.11 has 1 distinct ipsrc on ports: tcp/80
ipdst 208.10.192.202 has 1 distinct ipsrc on ports: tcp/80
ipdst 208.10.192.176 has 1 distinct ipsrc on ports: tcp/80
10-197-141-236:hw3 nickbarraclough$
```

O.csv



```
10-197-141-236:hw3 nickbarraclough$ python scancsv.py O.csv -connto
numPackets:999914 numBytes:366325065
  1:      6794
  6:    950654
 17:     38332
 47:      2626
 50:      1484
 89:        24
ipdst 192.245.12.242 has 1048 distinct ipsrc on ports: tcp/135, tcp/22, udp/137, tcp/25
ipdst 192.245.12.234 has 1009 distinct ipsrc on ports: tcp/22, tcp/135, tcp/25
ipdst 192.245.12.233 has 850 distinct ipsrc on ports: tcp/445, tcp/22, tcp/135, tcp/25
ipdst 192.245.12.230 has 817 distinct ipsrc on ports: tcp/22, tcp/135, tcp/25
ipdst 192.245.12.56 has 721 distinct ipsrc on ports: tcp/22, tcp/135, udp/53
ipdst 192.245.12.7 has 624 distinct ipsrc on ports: udp/53, tcp/135, tcp/23, tcp/25, udp/123, tcp/80
ipdst 192.245.12.221 has 382 distinct ipsrc on ports: tcp/139, tcp/113, tcp/25, udp/123, tcp/80, tcp/135
ipdst 192.245.12.50 has 342 distinct ipsrc on ports: udp/13, udp/37, udp/53
ipdst 192.245.12.8 has 232 distinct ipsrc on ports: udp/53, tcp/110, tcp/22, tcp/23, tcp/25, udp/123, tcp/143, tcp/993, tcp/995, tcp/135
ipdst 192.245.12.52 has 201 distinct ipsrc on ports: tcp/53, tcp/135, udp/53
ipdst 207.182.38.2 has 191 distinct ipsrc on ports: tcp/53, tcp/445, udp/53, tcp/25
ipdst 192.245.12.9 has 82 distinct ipsrc on ports: tcp/445, tcp/465, udp/53, tcp/110, tcp/22, tcp/25, tcp/1023, tcp/143, tcp/993, tcp/995, tcp/135
ipdst 204.153.45.2 has 73 distinct ipsrc on ports: tcp/25, udp/53
ipdst 204.153.45.185 has 69 distinct ipsrc on ports: tcp/80
ipdst 192.245.12.53 has 68 distinct ipsrc on ports: tcp/135, udp/53
ipdst 207.182.32.14 has 65 distinct ipsrc on ports: tcp/25
ipdst 207.182.38.3 has 58 distinct ipsrc on ports: tcp/445, udp/53, tcp/80, tcp/25
ipdst 192.245.12.31 has 42 distinct ipsrc on ports: tcp/445, tcp/135, tcp/25, tcp/80
ipdst 192.245.12.21 has 37 distinct ipsrc on ports: tcp/135, udp/123
ipdst 192.245.12.245 has 35 distinct ipsrc on ports: tcp/110, tcp/135, udp/53, tcp/80, tcp/25
ipdst 192.245.12.231 has 33 distinct ipsrc on ports: tcp/22, tcp/135, tcp/25
10-197-141-236:hw3 nickbarraclough$
```

b. **For the R data, identify the web servers, the printers, the mail servers, the DNS servers**

web servers: 208.10.192.176 (tcp/80)
printers: 10.5.63.255 (tcp/138)
mail servers: 10.5.63.6 (tcp/110, tcp/25)
DNS servers: 10.5.63.6 (tcp/53)

c. **For the O data, identify the mail servers, the pop/imap servers, the DNS servers**

mail servers: 192.245.12.242, 192.245.12.234, 192.245.12.233, 192.245.12.230
POP/IMAP servers: 192.245.12.8
DNS servers: 192.245.12.56, 192.245.12.7, 192.245.12.50

11. **Update your answer from [5] based on this information.**

R.csv is most likely a workplace multicast network. Port 138 was used the most which

I really don’t know what O could be… O.csv had a lot of traffic at port 135 which is for in Internet — Remote procedure call (RPC), a communication process that allows for executing a subroutine or procedure in another address space.