Week#2 Labs

Neha Agrawal

Table of Contents

02.1: TCP, HTTP

1. TCP #1 (netstat, lsof, nc)

netstat

Examine the man page for netstat to determine the 4 flags that you can pass the tool to list all TCP sockets in a LISTEN state on an IPv4 address and the program that is using it.

- **Run the command and take a screenshot of the output to include in your lab notebook.**

  sudo netstat -t -l -4 --program

```
agrawal@agrawal-VirtualBox:~$ sudo netstat -t -l -4 --program
[sudo] password for agrawal:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      441/systemd-resolve
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      599/sshd: /usr/sbin
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN      559/cupsd
tcp        0      0 localhost:6010          0.0.0.0:*               LISTEN      2406/sshd: agrawal@
tcp        0      0 localhost:37693         0.0.0.0:*               LISTEN      590/containerd
```

- **For port numbers that are named, examine /etc/services and find the port number that corresponds to it. Include this mapping in your lab notebook.**
  - *cat /etc/services | grep domain*

```
agrawal@agrawal-VirtualBox:~$ cat /etc/services | grep domain
domain          53/tcp                          # Domain Name Server
domain          53/udp
domain-s        853/tcp                         # DNS over TLS [RFC7858]
domain-s        853/udp                         # DNS over DTLS [RFC8094]
```

  - *cat /etc/services | grep ssh*

```
agrawal@agrawal-VirtualBox:~$ sudo cat /etc/services | grep ssh
ssh             22/tcp                          # SSH Remote Login Protocol
```

  - *cat /etc/services | grep ipp*

```
agrawal@agrawal-VirtualBox:~$ cat /etc/services | grep ipp
ipp             631/tcp                        # Internet Printing Protocol
```

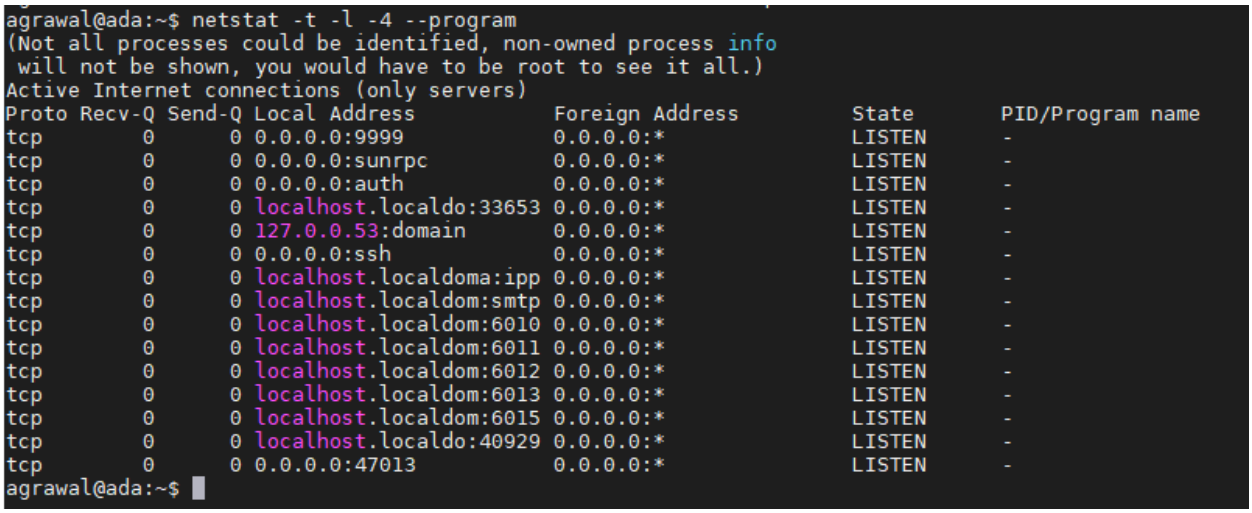
Named Port numbers	Port Number	Service Name
domain	53	Domain Name server
ssh	22	SSH Remote Login Protocol
ipp	631	Internet Printing protocol

- **For ports that only have a number, what service might it be providing based on the name of the program that is being run?**

Program name	Service provided
sshd	Secure Shell Daemon is a part of the OpenSSH implementation
containerd	Containerd daemon acts as API façade for various containers and OS

Login to linux.cs.pdx.edu

- **Run the netstat command again and include a screenshot of the output**



- **What services does this machine provide for external access?**

Port name/Port number	Service provided
9999	Datagram protocol – Allows transmission of datagram from one computer to an application running on another computer
sunrpc	TCP, UDP – allows remote procedural call
auth	TCP, UDP – authentication service
domain	DNS sevices
ssh	Secure Remote Login

ipp	Internet printing protocol
smtp	Simple mail transfer protocol

lsof

- **Use the -i and the -s flag of lsof to generate a listing that is equivalent to the one generated with netstat previously and include it in your lab notebook**

```
agrawal@agrawal-VirtualBox:~$ sudo lsof -i TCP | grep LISTEN | grep IPv4
systemd-r  441 systemd-resolve   13u  IPv4  20149      0t0  TCP localhost:domain (LISTEN)
cupsd      559            root    7u  IPv4  23082      0t0  TCP localhost:ipp (LISTEN)
container  590            root    8u  IPv4  24283      0t0  TCP localhost:37693 (LISTEN)
sshd       599            root    3u  IPv4  21998      0t0  TCP *:ssh (LISTEN)
sshd      2406         agrawal   11u  IPv4  42235      0t0  TCP localhost:6010 (LISTEN)
agrawal@agrawal-VirtualBox:~$
```

Or

```
agrawal@agrawal-VirtualBox:~$ sudo lsof -i4 -i TCP | grep LISTEN
systemd-r  441 systemd-resolve   13u  IPv4  20149      0t0  TCP localhost:domain (LISTEN)
cupsd      559            root    7u  IPv4  23082      0t0  TCP localhost:ipp (LISTEN)
container  590            root    8u  IPv4  24283      0t0  TCP localhost:37693 (LISTEN)
sshd       599            root    3u  IPv4  21998      0t0  TCP *:ssh (LISTEN)
sshd      2406         agrawal   11u  IPv4  42235      0t0  TCP localhost:6010 (LISTEN)
```

nc

- **Include for your lab notebook, the version of ssh that is being used. (Type Control-c to exit)**

```
agrawal@agrawal-VirtualBox:~$ nc linux.cs.pdx.edu 22
SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.1
```

2. TCP #2 (iperf)

No screenshots required

3. Throughput tests

- **Show a screenshot of the measured bandwidth available between your us-west1-b VM and each of the other Compute Engine VMs. Explain the relative differences (or lack thereof) in your results.**

```
agrawal@instance-1:~$ iperf -c 10.142.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.142.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.9 port 48992 connected with 10.142.0.2 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   240 MBytes   200 Mbits/sec
agrawal@instance-1:~$ iperf -c 10.152.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.152.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.9 port 51590 connected with 10.152.0.2 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   116 MBytes  97.1 Mbits/sec
agrawal@instance-1:~$ iperf -c 10.166.0.2 -p 80
------------------------------------------------------------
Client connecting to 10.166.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.9 port 45936 connected with 10.166.0.2 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   151 MBytes   126 Mbits/sec
```

Source VM location	Destination VM location	Bandwidth	Distance
Us-west1-b	Us-east	200 Mb/s	2500 MILES
Us-west1-b	australia	97.1 Mb/s	4500 miles
Us-west1-b	europe	126 Mb/s	8500 miles

As the relative distance between the us west1 VM and the other VM instance increases, the bandwidth decreases.

So basically, Bandwidth is inversely proportional to distance between the VMs.

The VM closest to the source(us-west1-b) has been allocated the maximum bandwidth.

4. HTTP #3 (Browser tools)

No screenshots required

5. Developer tools

Click on the very first request to bring up the connection details of the request and answer the following questions in your lab notebook.

- **What is the URL being requested?**

  `http://google.com/`

- **What are the `Host:` and `User-Agent:` HTTP request headers being sent by the browser?**

  Host: google.com

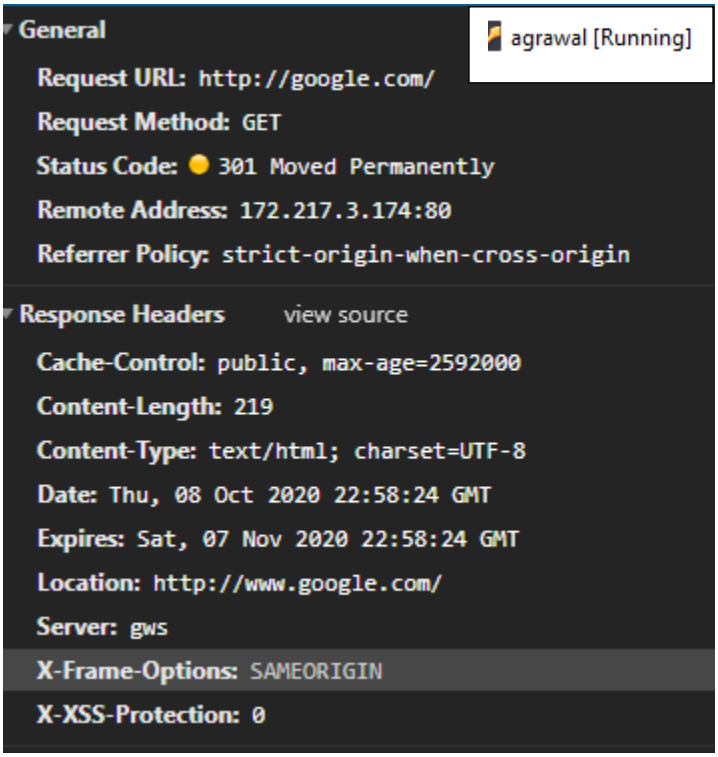
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36

- **What is the HTTP status code in the response and what does it mean?**

  Status Code: 301 Moved Permanently

  This means that the requested url i.e. http://google.com has been definitively moved to the url given by the location header which is “http://www.google.com/”

- **Look up the status code. Show the associated HTTP response header that is sent in conjunction with this status code for the request.**



Click on the second request to bring up its connection details. Answer the following questions in your lab notebook.

- **What is the URL being requested? Is it using HTTP or HTTPS?**

  Request URL: http://www.google.com/
  It is using http

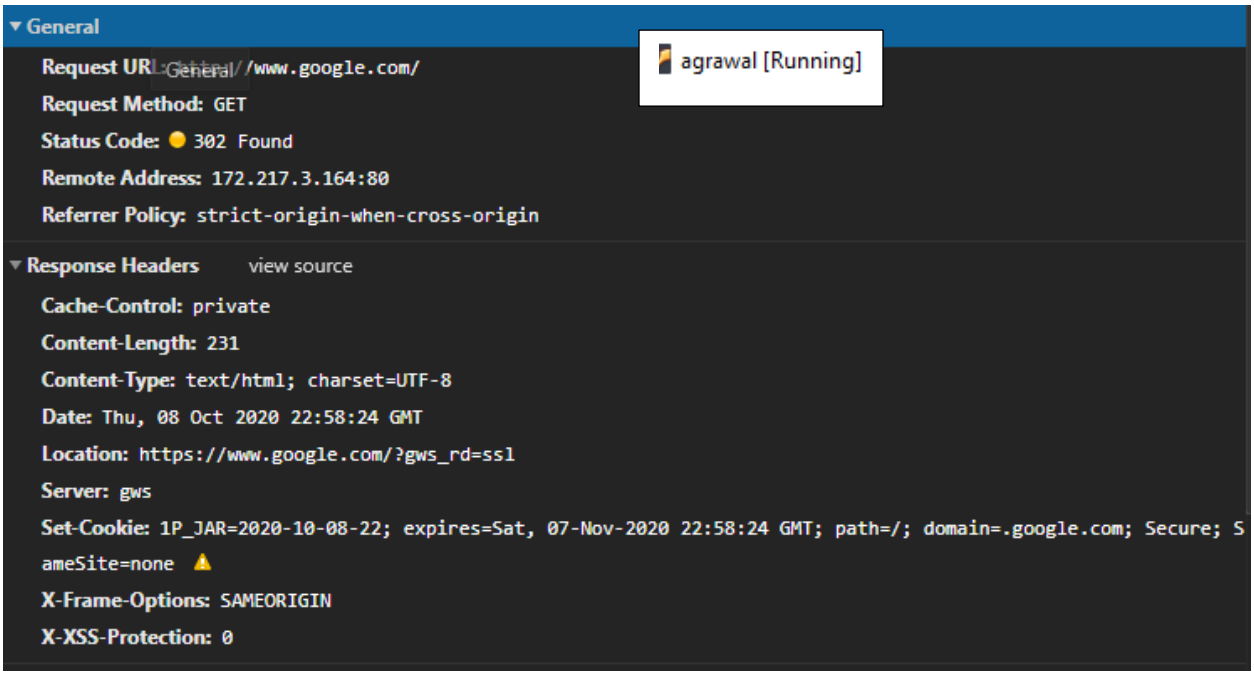
- **What is the HTTP status code in the response and what does it mean? Is it different from the first status code? If so, what is the semantic difference?**

  Status Code: 302 Found
  Http 302 status code means that the requested URL has been temporarily moved to the U RL given by the location header i.e. “https://www.google.com/?gws_rd=ssl”

  Yes, it is different from the first status code.

  A **301** redirect means that the page has permanently moved to a new location.
  A **302** redirect means that the move is only temporary

- **Show the associated HTTP response header that is sent in conjunction with this status code for the request.**



Click on the third request to bring up its connection details. Answer the following questions in your lab notebook.

- **What is the URL being requested? Is it using HTTP or HTTPS?**

  Request URL: https://www.google.com/?gws_rd=ssl
  It is using HTTPS

- **What is the HTTP status code in the response?**

  Status code: 200

- **Look for an alt-svc: HTTP response header. Does the server believe the client can use HTTP3/QUIC?**

  alt-svc: `h3-Q050=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T051=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"`

  Yes, the server believes that the client can use HTTP3/QUIC

- **Examine the HTTP response headers for cookies. Show the cookies that are set and their associated SameSite setting. What does the setting indicate about the cookies that are set?**

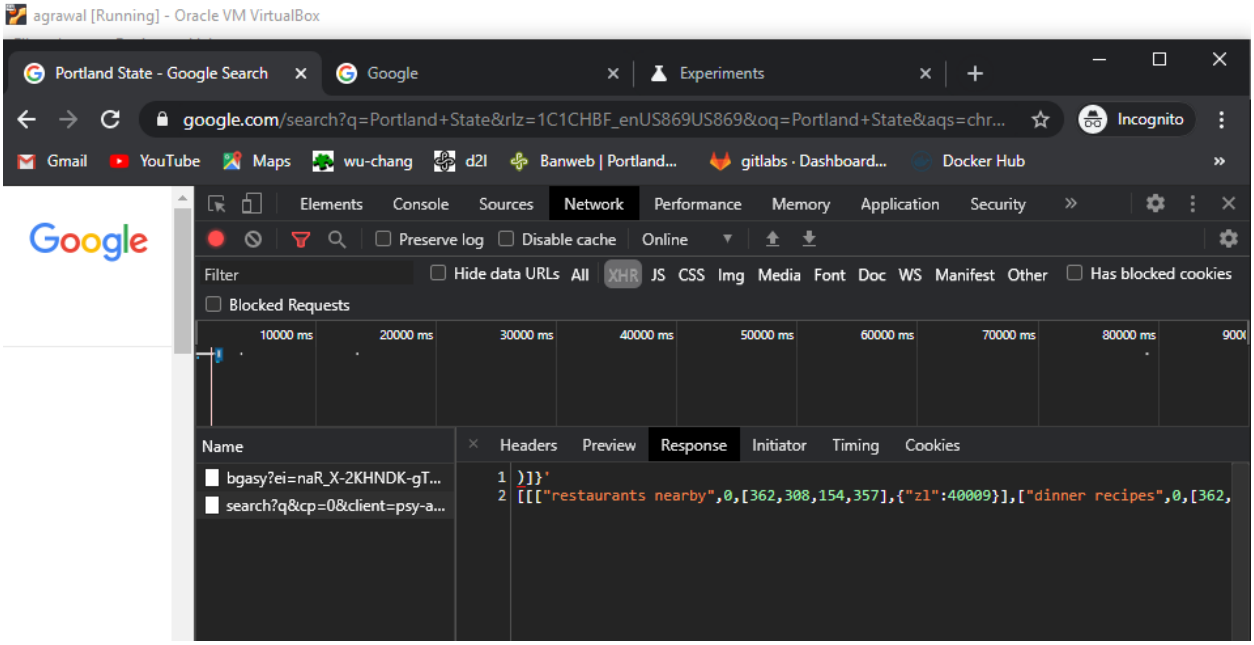
  set-cookie: `1P_JAR=2020-10-08-22; expires=Sat, 07-Nov-2020 22:58:24 GMT; path=/; domain=.google.com; Secure; SameSite=none`

  set-cookie: `NID=204=wYEidKX9hEOUXhn9OMq003IB8lYIrSvHTiZc_28B6wVZOnxIqjsjN-JBYvVhgAxnNcrPI3HttkVICAAlx-DehPuIt1TmRIngIAv-MTF7NxQ_MxgtkPyoQSTIA3XT4SZEBaqowIuAmbKu-f6lDoSQ8FMzz3G3LpdFaJhrz582wwQ; expires=Fri, 09-Apr-2021 22:58:24 GMT; path=/; domain=.google.com; Secure; HttpOnly; SameSite=none`

  SameSite = none will allow the cookies for cross-site access. An additional secure attribute must be used so that cross-site cookies will be available for external access, provided they are being accessed from secure connections.

6. Asynchronous HTTP requests

- **Show the requests and responses in the listing. Click on the last request sent, then click on the response to see that its payload has returned the data that is then rendered on the search page similar to what is shown below for "rabbid"**



02.2: DNS, Recap

1. DNS #1 (dig)

- **Use `dig` to query the local DNS server for the `A` record of `www.pdx.edu` using TCP. Then, use dig to do the same for the `MX` record of `pdx.edu`. What do the ANSWER sections explain about where PSU's web/mail services are run from?**

1. dig www.pdx.edu +tcp -t A

```
agrawal@ada:~$ dig www.pdx.edu +tcp -t A

; <<>> DiG 9.16.1-Ubuntu <<>> www.pdx.edu +tcp -t A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12828
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.pdx.edu.                   IN      A

;; ANSWER SECTION:
www.pdx.edu.            12659   IN      A       131.252.115.150

;; AUTHORITY SECTION:
pdx.edu.                161300  IN      NS      ns-cloud-e1.googledomains.com.
pdx.edu.                161300  IN      NS      ns-cloud-e4.googledomains.com.
pdx.edu.                161300  IN      NS      ns-cloud-e2.googledomains.com.
pdx.edu.                161300  IN      NS      ns-cloud-e3.googledomains.com.

;; ADDITIONAL SECTION:
ns-cloud-e1.googledomains.com. 334100 IN A      216.239.32.110
ns-cloud-e1.googledomains.com. 334100 IN AAAA   2001:4860:4802:32::6e
ns-cloud-e2.googledomains.com. 334100 IN A      216.239.34.110
ns-cloud-e2.googledomains.com. 334100 IN AAAA   2001:4860:4802:34::6e
ns-cloud-e3.googledomains.com. 334100 IN A      216.239.36.110
ns-cloud-e3.googledomains.com. 334100 IN AAAA   2001:4860:4802:36::6e
ns-cloud-e4.googledomains.com. 334100 IN A      216.239.38.110
ns-cloud-e4.googledomains.com. 334100 IN AAAA   2001:4860:4802:38::6e

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sat Oct 10 12:48:03 PDT 2020
;; MSG SIZE  rcvd: 353
```

TO display only the answer section, I will use +noall +answer options

```
agrawal@ada:~$ dig www.pdx.edu +tcp -t A +noall +answer
www.pdx.edu.            12565   IN      A       131.252.115.150
```

Here we can see that the domain www.pdx.edu points to the 131.252.115.150 IP address.

2. dig pdx.edu +tcp -t MX +noall +answer

```
agrawal@ada:~$ dig pdx.edu +tcp -t MX +noall +answer
pdx.edu.                75441   IN      MX      10 alt3.aspmx.l.google.com.
pdx.edu.                75441   IN      MX      5 alt1.aspmx.l.google.com.
pdx.edu.                75441   IN      MX      1 aspmx.l.google.com.
pdx.edu.                75441   IN      MX      10 alt4.aspmx.l.google.com.
pdx.edu.                75441   IN      MX      5 alt2.aspmx.l.google.com.
```

MX options allows us to specify all the mail servers for the pdx.edu domain. In the above case, we see the MX records for pdx.edu is google mail servers. There are multiple MX records. The first one “aspmx.l.google.com” is the most important MX record and have the highest priority with smallest priority value 1.

- **Find the authoritative server (NS record type, AUTHORITY section response) for mashimaro.cs.pdx.edu and then query that server for the A record of mashimaro.cs.pdx.edu. Show both.**

```
agrawal@ada:~$ dig mashimaro.cs.pdx.edu ns

; <<>> DiG 9.16.1-Ubuntu <<>> mashimaro.cs.pdx.edu ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28485
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      NS

;; AUTHORITY SECTION:
cs.pdx.edu.             300     IN      SOA     walt.ee.pdx.edu. support.cat.pdx.edu. 2020100701 600 300 1209600 300

;; Query time: 3 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sat Oct 10 14:48:09 PDT 2020
;; MSG SIZE  rcvd: 105
```

Authoritative server for mashimaro.cs.pdx.edu is walt.ee.pdx.edu.

We can query the authoritative server

```
agrawal@ada:~$ dig @walt.ee.pdx.edu mashimaro.cs.pdx.edu +tcp A

; <<>> DiG 9.16.1-Ubuntu <<>> @walt.ee.pdx.edu mashimaro.cs.pdx.edu +tcp A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30642
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

;; ANSWER SECTION:
mashimaro.cs.pdx.edu.   14400   IN      A       131.252.220.66

;; AUTHORITY SECTION:
cs.pdx.edu.             14400   IN      NS      dns0.pdx.edu.
cs.pdx.edu.             14400   IN      NS      walt.ee.pdx.edu.
cs.pdx.edu.             14400   IN      NS      dns1.pdx.edu.
cs.pdx.edu.             14400   IN      NS      phloem.uoregon.edu.

;; ADDITIONAL SECTION:
dns0.pdx.edu.           14400   IN      A       131.252.120.128
dns1.pdx.edu.           14400   IN      A       131.252.120.129
walt.ee.pdx.edu.        14400   IN      A       131.252.208.38

;; Query time: 0 msec
;; SERVER: 131.252.208.38#53(131.252.208.38)
;; WHEN: Sat Oct 10 14:50:16 PDT 2020
;; MSG SIZE  rcvd: 202
```

IP address for mashimaro.cs.pdx.edu is 131.252.220.66

- **Find the authoritative server for thefengs.com and then query that server for the A record of thefengs.com**

```
agrawal@ada:~$ dig thefengs.com NS

; <<>> DiG 9.16.1-Ubuntu <<>> thefengs.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12710
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;thefengs.com.                  IN      NS

;; ANSWER SECTION:
thefengs.com.           21600   IN      NS      ns-cloud4.googledomains.com.
thefengs.com.           21600   IN      NS      ns-cloud2.googledomains.com.
thefengs.com.           21600   IN      NS      ns-cloud3.googledomains.com.
thefengs.com.           21600   IN      NS      ns-cloud1.googledomains.com.

;; ADDITIONAL SECTION:
ns-cloud1.googledomains.com. 267756 IN  A       216.239.32.106
ns-cloud1.googledomains.com. 15031 IN   AAAA    2001:4860:4802:32::6a
ns-cloud2.googledomains.com. 288040 IN  A       216.239.34.106
ns-cloud2.googledomains.com. 15031 IN   AAAA    2001:4860:4802:34::6a
ns-cloud3.googledomains.com. 280297 IN  A       216.239.36.106
ns-cloud3.googledomains.com. 15031 IN   AAAA    2001:4860:4802:36::6a
ns-cloud4.googledomains.com. 191423 IN  A       216.239.38.106
ns-cloud4.googledomains.com. 191423 IN  AAAA    2001:4860:4802:38::6a

;; Query time: 15 msec
;; SERVER: 131.252.208.53#53(131.252.208.53)
;; WHEN: Sat Oct 10 14:51:50 PDT 2020
;; MSG SIZE  rcvd: 327
```

```
agrawal@ada:~$ dig @ns-cloud2.googledomains.com thefengs.com +tcp A

; <<>> DiG 9.16.1-Ubuntu <<>> @ns-cloud2.googledomains.com thefengs.com +tcp A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19958
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;thefengs.com.                  IN      A

;; ANSWER SECTION:
thefengs.com.           3600    IN      A       131.252.220.66

;; Query time: 71 msec
;; SERVER: 216.239.34.106#53(216.239.34.106)
;; WHEN: Sat Oct 10 13:50:59 PDT 2020
;; MSG SIZE  rcvd: 57
```

Ip address of thefengs.com is 131.252.220.66

- **When a web request hits port 80 of 131.252.220.66, how does the server know which site to serve from? (i.e. what protocol header)**

  HTTP header is used by the server to know which site to serve from.

DNS iterative lookups

Examine the man page for dig to find the query option that allows one to specify whether a query can recurse or whether it should be iterative. On linux.cs.pdx.edu, simulate the operation of a local DNS server. Choose a DNS name containing at least 4 parts (e.g. www.cs.pdx.edu , console.cloud.google.com , www.unsw.edu.au , www.amazon.co.uk). Start by running dig with no arguments to list all root DNS servers that have been hard-coded into the tool. Locate the IPv4 address of the F root server.

Starting with the F root server, perform the iterative queries a local DNS server would perform on a lookup. In performing this sequence of queries, ensure the queries are iterative and use TCP. (MCECS networks block UDP DNS traffic). Ensure that you are traveling down the hierarchy with the servers being specified via the @. Ensure you use the appropriate DNS record type for specifying that the authoritative server should be returned.

- **Include the results of each query for your lab notebook.**

1. . Locate the IPv4 address of the F root server

```
agrawal@ada:~$ dig f.root-servers.net

; <<>> DiG 9.16.1-Ubuntu <<>> f.root-servers.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7325
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 13, ADDITIONAL: 26

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;f.root-servers.net.            IN      A

;; ANSWER SECTION:
f.root-servers.net.     429367  IN      A       192.5.5.241
```

2. Do an iterative query to IP addr of F root

```
agrawal@ada:~$ dig @192.5.5.241 +norecurse +tcp www.cs.pdx.edu

; <<>> DiG 9.16.1-Ubuntu <<>> @192.5.5.241 +norecurse +tcp www.cs.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58544
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65535
;; QUESTION SECTION:
;www.cs.pdx.edu.                        IN      A

;; AUTHORITY SECTION:
edu.                    172800  IN      NS      l.edu-servers.net.
edu.                    172800  IN      NS      b.edu-servers.net.
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.
edu.                    172800  IN      NS      e.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      h.edu-servers.net.
edu.                    172800  IN      NS      i.edu-servers.net.
edu.                    172800  IN      NS      j.edu-servers.net.
edu.                    172800  IN      NS      k.edu-servers.net.
edu.                    172800  IN      NS      m.edu-servers.net.

;; ADDITIONAL SECTION:
l.edu-servers.net.      172800  IN      A       192.41.162.30
l.edu-servers.net.      172800  IN      AAAA    2001:500:d937::30
b.edu-servers.net.      172800  IN      A       192.33.14.30
b.edu-servers.net.      172800  IN      AAAA    2001:503:231d::2:30
c.edu-servers.net.      172800  IN      A       192.26.92.30
c.edu-servers.net.      172800  IN      AAAA    2001:503:83eb::30
```

3. Do an iterative query to IP address of L TLD

```
agrawal@ada:~$ dig @192.41.162.30 +norecurse +tcp www.cs.pdx.edu

; <<>> DiG 9.16.1-Ubuntu <<>> @192.41.162.30 +norecurse +tcp www.cs.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35547
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.cs.pdx.edu.                        IN      A

;; AUTHORITY SECTION:
pdx.edu.                172800  IN      NS      ns-cloud-e1.googledomains.com.
pdx.edu.                172800  IN      NS      ns-cloud-e2.googledomains.com.
pdx.edu.                172800  IN      NS      ns-cloud-e3.googledomains.com.
pdx.edu.                172800  IN      NS      ns-cloud-e4.googledomains.com.

;; Query time: 19 msec
;; SERVER: 192.41.162.30#53(192.41.162.30)
;; WHEN: Sat Oct 10 15:41:50 PDT 2020
;; MSG SIZE  rcvd: 164
```

4. Do an iterative query to IP address of NS at googledomains:

   Ip addr of NS at googledomains:

```
agrawal@ada:~$ dig ns-cloud-e1.googledomains.com A

; <<>> DiG 9.16.1-Ubuntu <<>> ns-cloud-e1.googledomains.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54015
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ns-cloud-e1.googledomains.com. IN      A

;; ANSWER SECTION:
ns-cloud-e1.googledomains.com. 323521 IN A      216.239.32.110

;; AUTHORITY SECTION:
googledomains.com.      15471   IN      NS      ns8.googledomains.com.
googledomains.com.      15471   IN      NS      ns5.googledomains.com.
googledomains.com.      15471   IN      NS      ns7.googledomains.com.
googledomains.com.      15471   IN      NS      ns6.googledomains.com.

;; ADDITIONAL SECTION:
ns5.googledomains.com.  15471   IN      A       216.239.32.10
ns5.googledomains.com.  15471   IN      AAAA    2001:4860:4802:32::a
ns6.googledomains.com.  15471   IN      A       216.239.34.10
```

   dig @216.239.32.110 +norecurse +tcp www.cs.pdx.edu

```
agrawal@ada:~$ dig @216.239.32.110 +norecurse +tcp www.cs.pdx.edu

; <<>> DiG 9.16.1-Ubuntu <<>> @216.239.32.110 +norecurse +tcp www.cs.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8756
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.cs.pdx.edu.                        IN      A

;; AUTHORITY SECTION:
cs.pdx.edu.             14400   IN      NS      dns0.pdx.edu.
cs.pdx.edu.             14400   IN      NS      dns1.pdx.edu.
cs.pdx.edu.             14400   IN      NS      walt.ee.pdx.edu.
cs.pdx.edu.             14400   IN      NS      phloem.uoregon.edu.

;; ADDITIONAL SECTION:
dns0.pdx.edu.           14400   IN      A       131.252.120.128
dns1.pdx.edu.           14400   IN      A       131.252.120.129
walt.ee.pdx.edu.        14400   IN      A       131.252.208.38

;; Query time: 11 msec
;; SERVER: 216.239.32.110#53(216.239.32.110)
;; WHEN: Sat Oct 10 15:46:10 PDT 2020
;; MSG SIZE  rcvd: 180
```

5. Query IP addr of authoritative server (dns0.pdx.edu) to get A record of www.cs.pdx.edu

```
agrawal@ada:~$ dig @131.252.120.128 +norecurse +tcp www.cs.pdx.edu

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.120.128 +norecurse +tcp www.cs.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37566
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3552671af57a79f0cf7826f75f823bcac4fe16546f99627c (good)
;; QUESTION SECTION:
;www.cs.pdx.edu.                        IN      A

;; ANSWER SECTION:
www.cs.pdx.edu.         14400   IN      CNAME   vhost-therest.cat.pdx.edu.
vhost-therest.cat.pdx.edu. 14400 IN     A       131.252.208.114

;; AUTHORITY SECTION:
cat.pdx.edu.            14400   IN      NS      walt.ee.pdx.edu.
cat.pdx.edu.            14400   IN      NS      dns0.pdx.edu.
cat.pdx.edu.            14400   IN      NS      phloem.uoregon.edu.
cat.pdx.edu.            14400   IN      NS      dns1.pdx.edu.

;; ADDITIONAL SECTION:
dns0.pdx.edu.           14400   IN      A       131.252.120.128
dns1.pdx.edu.           14400   IN      A       131.252.120.129
walt.ee.pdx.edu.        14400   IN      A       131.252.208.38
phloem.uoregon.edu.     82965   IN      A       128.223.32.35
phloem.uoregon.edu.     82965   IN      AAAA    2001:468:d01:20::80df:2023

;; Query time: 0 msec
;; SERVER: 131.252.120.128#53(131.252.120.128)
```

2. Reverse DNS lookups

- **Use a single command line with commands `dig`, `egrep`, and `awk`, to list all IPv4 addresses that `espn.go.com` points to.**

```
agrawal@ada:~$ dig espn.go.com +tcp +noall +answer |egrep espn.go.com | awk '{print $5}'
99.84.66.98
99.84.66.108
99.84.66.55
99.84.66.17
```

```
agrawal@ada:~$ X=`dig espn.go.com +tcp +noall +answer |egrep espn.go.com | awk '{print $5}'`
agrawal@ada:~$ echo $X
99.84.66.17 99.84.66.108 99.84.66.98 99.84.66.55
```

- **Take that list and create a single for loop in the shell that iterates over the list and performs a reverse lookup of each IP address to find each address's associated DNS name. As with the previous step, pipe the output of the `for` loop to `egrep` and `awk` so that the output consists only of the DNS names.**

```
X=`dig espn.go.com +tcp +noall +answer | egrep espn.go.com | awk ‘{print $5}’`

for i in `echo $X`
do
    dig -x $i +noall +answer | awk ‘{print $5}’
done
```

Output:

```
agrawal@ada:~$ for i in `echo $X`; do   dig -x $i +noall +answer| awk '{print $5}'; done
server-99-84-66-17.hio50.r.cloudfront.net.
server-99-84-66-98.hio50.r.cloudfront.net.
server-99-84-66-55.hio50.r.cloudfront.net.
server-99-84-66-108.hio50.r.cloudfront.net.
agrawal@ada:~$
```

3. Host enumeration

```
agrawal@ada:~$ for i in {0..255}; do  dig -x 131.252.220.$i +noall +answer | awk '{print $5}';done
colt45.cs.pdx.edu.
kingcobra.cs.pdx.edu.
mickeys.cs.pdx.edu.
magnum.cs.pdx.edu.
phatboy.cs.pdx.edu.
schlitz.cs.pdx.edu.
boar.cs.pdx.edu.
dog.cs.pdx.edu.
dragon.cs.pdx.edu.
horse.cs.pdx.edu.
monkey.cs.pdx.edu.
ox.cs.pdx.edu.
rabbit.cs.pdx.edu.
rat.cs.pdx.edu.
```

The range of hosts that have car brand names is in between .156 and .186

```
agrawal@agrawal-VirtualBox:~$ head -185 220hosts.txt | tail -30
acura.cs.pdx.edu.
astonmartin.cs.pdx.edu.
audi.cs.pdx.edu.
bentley.cs.pdx.edu.
bmw.cs.pdx.edu.
cadillac.cs.pdx.edu.
ferrari.cs.pdx.edu.
fiat.cs.pdx.edu.
ford.cs.pdx.edu.
honda.cs.pdx.edu.
hummer.cs.pdx.edu.
jaguar.cs.pdx.edu.
jeep.cs.pdx.edu.
lamborghini.cs.pdx.edu.
landrover.cs.pdx.edu.
lexus.cs.pdx.edu.
lotus.cs.pdx.edu.
maserati.cs.pdx.edu.
mazda.cs.pdx.edu.
mclaren.cs.pdx.edu.
mercedes.cs.pdx.edu.
nissan.cs.pdx.edu.
panoz.cs.pdx.edu.
porsche.cs.pdx.edu.
subaru.cs.pdx.edu.
toyota.cs.pdx.edu.
tvr.cs.pdx.edu.
ultima.cs.pdx.edu.
volvo.cs.pdx.edu.
vw.cs.pdx.edu.
```

4. DNS #2 (Geographic DNS)

Visit https://www.iplocation.net/ and lookup the geographical location of the following DNS servers: 131.252.208.53 and 198.82.247.66.

- **What geographic locations do ipinfo.io and DB-IP return?**

Ip address		Geographic Location
131.252.208.53	Ipinfo.io	Portland State University, Portland
	DB-IP	Portland State University, Portland
198.82.247.66	Ipinfo.io	Raleigh, North Carolina
	DB-IP	Raleigh, North Carolina

Then, using dig, resolve www.google.com from each of the DNS servers (dig @<DNS_server_IP> www.google.com).

- **Record each result for your lab notebook**

```
agrawal@ada:~$ dig @131.252.208.53 www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @131.252.208.53 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51677
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         298     IN      A       172.217.3.164

;; AUTHORITY SECTION:
google.com.             143343  IN      NS      ns3.google.com.
google.com.             143343  IN      NS      ns4.google.com.
google.com.             143343  IN      NS      ns2.google.com.
google.com.             143343  IN      NS      ns1.google.com.
```

```
agrawal@ada:~$ dig @198.82.247.66 www.google.com

; <<>> DiG 9.16.1-Ubuntu <<>> @198.82.247.66 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48866
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8025a8254937c7a7c8586eb25f8256965a3262cc3745686c (good)
;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         3       IN      A       142.250.31.104
www.google.com.         3       IN      A       142.250.31.147
www.google.com.         3       IN      A       142.250.31.99
www.google.com.         3       IN      A       142.250.31.105
www.google.com.         3       IN      A       142.250.31.106
www.google.com.         3       IN      A       142.250.31.103
```

Go back to https://www.iplocation.net/ and lookup the geographical location of each IP address returned. What geographic locations do ipinfo.io and DB-IP return?

Ip address		Geographic Location
172.217.3.164	Ipinfo.io	Tacoma, Washington
	DB-IP	Seattle, Washington
142.250.31.99 142.250.31.103 142.250.31.104 142.250.31.105 142.250.31.106 142.250.31.107	Ipinfo.io	Dallas, Texas
	DB-IP	Montreal, Quebec

- **What is the geographic distance between each pair of DNS server and web server?**

DNS Server	Web server - www.google.com IP address		Geographic Location	Distance
131.252.208.53 PSU, Portland	172.217.3.164	Ipinfo.io	Tacoma, Washington	144 miles
		DB-IP	Seattle, Washington	174 miles
198.82.247.66 Raleigh, North Carolina	142.250.31.99 142.250.31.103 …. 142.250.31.107	Ipinfo.io	Dallas, Texas	1188 miles
		DB-IP	Montreal, Quebec	845 miles

Perform a traceroute to all 4 IP addresses from a PSU network.

- **Do the routes reveal any information on the accuracy of the geographic locations given? (Answer might be no)**

```
agrawal@ada:~$ traceroute 131.252.208.53
traceroute to 131.252.208.53 (131.252.208.53), 30 hops max, 60 byte packets
 1  rdns.cat.pdx.edu (131.252.208.53)  0.964 ms  0.913 ms  0.875 ms
```

Yes

```
agrawal@ada:~$ traceroute 198.82.247.66
traceroute to 198.82.247.66 (198.82.247.66), 30 hops max, 60 byte packets
 1  radiant.seas.pdx.edu (131.252.208.212)  1.229 ms  1.245 ms  1.347 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.604 ms  0.576 ms  0.540 ms
 3  10.252.5.10 (10.252.5.10)  1.649 ms  1.684 ms  1.599 ms
 4  ptck-pe1-gw.nero.net (199.165.177.18)  1.433 ms  1.381 ms  1.346 ms
 5  ae-0.701.rtsw.port.net.internet2.edu (198.71.45.218)  1.433 ms  1.383 ms  1.331 ms
 6  et-7-0-0.4070.rtsw.seat.net.internet2.edu (162.252.70.83)  5.093 ms  5.022 ms  5.049 ms
 7  ae-1.4079.rtsw.minn.net.internet2.edu (162.252.70.173)  37.452 ms  37.515 ms  37.439 ms
 8  ae-1.4079.rtsw.eqch.net.internet2.edu (162.252.70.106)  45.520 ms  45.384 ms  45.455 ms
 9  ae-0.4079.rtsw3.eqch.net.internet2.edu (162.252.70.163)  45.246 ms  45.264 ms  45.324 ms
10  ae-1.4079.rtsw.clev.net.internet2.edu (162.252.70.130)  51.549 ms  51.593 ms  51.482 ms
11  ae-0.4079.rtsw.ashb.net.internet2.edu (162.252.70.128)  58.730 ms  58.777 ms  58.697 ms
12  192.122.175.14 (192.122.175.14)  59.329 ms  59.629 ms  59.156 ms
13  vtacs-1.msap.cns.vt.edu (192.70.187.18)  65.849 ms  66.019 ms  65.946 ms
14  isb-core.et-5-1-0.0.cns.vt.edu (128.173.0.206)  66.114 ms  66.586 ms  66.554 ms
15  cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  66.576 ms  66.310 ms  66.279 ms
16  jeru.cns.vt.edu (198.82.247.66)  65.660 ms  65.675 ms  65.563 ms
```

Yes, vt.edu -> Virginia Tech which is in North Carolina

```
agrawal@ada:~$ traceroute 172.217.3.164
traceroute to 172.217.3.164 (172.217.3.164), 30 hops max, 60 byte packets
 1  radiant.seas.pdx.edu (131.252.208.212)  1.167 ms  3.324 ms  3.256 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.890 ms  0.832 ms  0.782 ms
 3  10.252.5.10 (10.252.5.10)  2.974 ms  2.915 ms  2.856 ms
 4  google.nwax.net (198.32.195.34)  4.716 ms  4.675 ms  5.009 ms
 5  108.170.245.113 (108.170.245.113)  4.286 ms 108.170.245.97 (108.170.245.97)  5.949 ms 108.170.245.113 (108.170.245.113)  4.300 ms
 6  108.170.233.157 (108.170.233.157)  4.737 ms 108.170.233.159 (108.170.233.159)  4.894 ms  5.037 ms
 7  sea15s11-in-f164.1e100.net (172.217.3.164)  4.334 ms  4.190 ms  4.601 ms
```

No

```
agrawal@ada:~$ traceroute 142.250.31.99
traceroute to 142.250.31.99 (142.250.31.99), 30 hops max, 60 byte packets
 1  radiant.seas.pdx.edu (131.252.208.212)  8.142 ms  8.270 ms  8.320 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.551 ms  0.509 ms  0.477 ms
 3  10.252.5.10 (10.252.5.10)  1.591 ms  1.644 ms  1.534 ms
 4  google.nwax.net (198.32.195.34)  4.979 ms  4.655 ms  4.628 ms
 5  108.170.245.123 (108.170.245.123)  4.877 ms  5.173 ms 108.170.245.107 (108.170.245.107)  5.842 ms
 6  142.250.228.150 (142.250.228.150)  12.188 ms 142.250.237.168 (142.250.237.168)  12.381 ms 216.239.
 7  209.85.250.4 (209.85.250.4)  36.427 ms 108.170.235.196 (108.170.235.196)  36.552 ms 209.85.250.4 (
 8  209.85.251.154 (209.85.251.154)  44.908 ms 172.253.74.22 (172.253.74.22)  45.555 ms  45.405 ms
 9  142.250.235.126 (142.250.235.126)  46.906 ms  46.978 ms *
10  * 142.250.232.127 (142.250.232.127)  55.626 ms 72.14.234.8 (72.14.234.8)  54.878 ms
11  209.85.253.249 (209.85.253.249)  70.152 ms 209.85.252.39 (209.85.252.39)  70.544 ms 209.85.253.249
12  142.250.236.149 (142.250.236.149)  71.388 ms  71.358 ms 172.253.74.193 (172.253.74.193)  71.019 ms
13  209.85.248.53 (209.85.248.53)  69.755 ms  69.522 ms 216.239.46.31 (216.239.46.31)  69.346 ms
14  172.253.72.41 (172.253.72.41)  69.294 ms 172.253.72.67 (172.253.72.67)  70.320 ms 172.253.72.51 (1
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  142.250.31.99 (142.250.31.99)  69.714 ms  68.724 ms *
```

No

When we lookup iplocations.net, then the different sources provide different location for an IP address of www.google.com. This makes it very difficult to pinpoint geolocation of an IP address,
Also, the traceroutes did not provide any information on the accuracy of the geographic locations given.

5. Network Recap Lab #3

Ip Address of the VM: 192.168.1.20
Name of the Local Virtual ethernet: enp0s3
IP address of the default router: 192.168.1.1

- **Include it in your lab notebook**

```
agrawal@agrawal-VirtualBox:~$ dig -x 1.1.1.1

; <<>> DiG 9.16.1-Ubuntu <<>> -x 1.1.1.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20076
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;1.1.1.1.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
1.1.1.1.in-addr.arpa.   66      IN      PTR     one.one.one.one.

;; Query time: 4 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sat Oct 10 19:01:59 PDT 2020
;; MSG SIZE  rcvd: 78
```

Dump ARP table

```
agrawal@agrawal-VirtualBox:~$ arp -an | awk -F '[()]' '{print $2}' > arp_entries
agrawal@agrawal-VirtualBox:~$ cat arp_entries
192.168.1.15
192.168.1.1
192.168.1.12
```

6. Collect and analyze the network trace of a connection

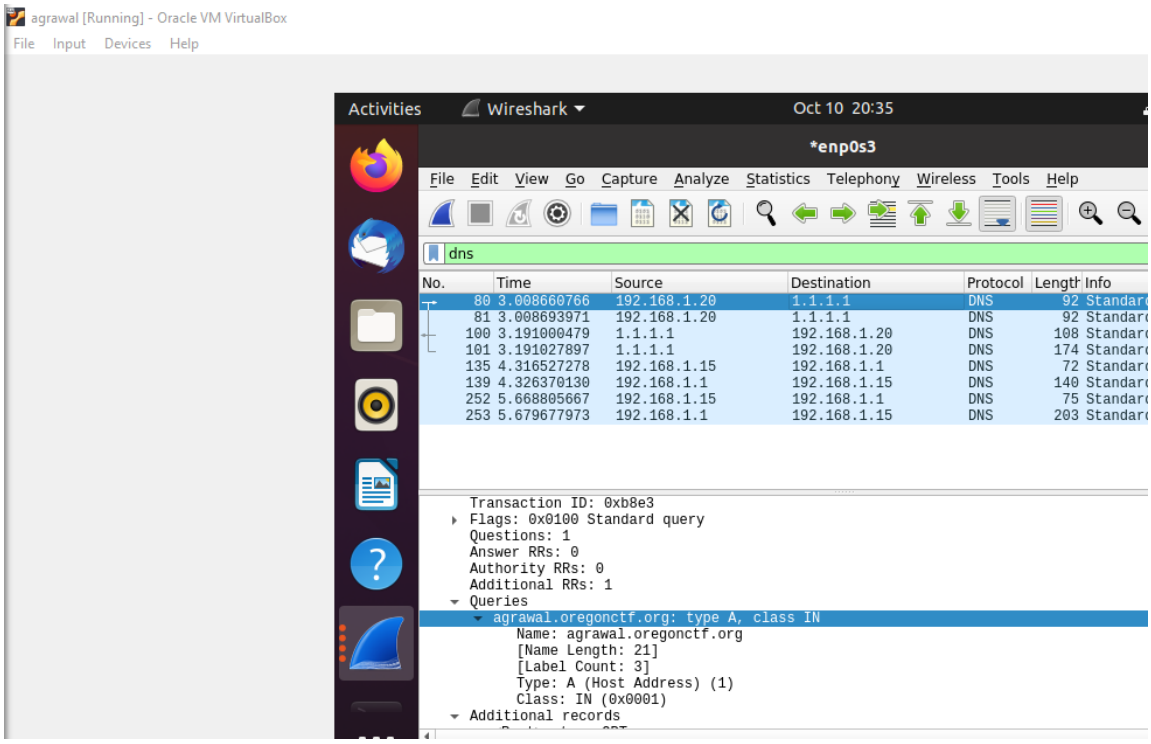
- **Take a screenshot of the trace within Wireshark and include an annotation of the packets in the trace to explain the purpose of each of the packets being exchanged.**

Answer the following questions:

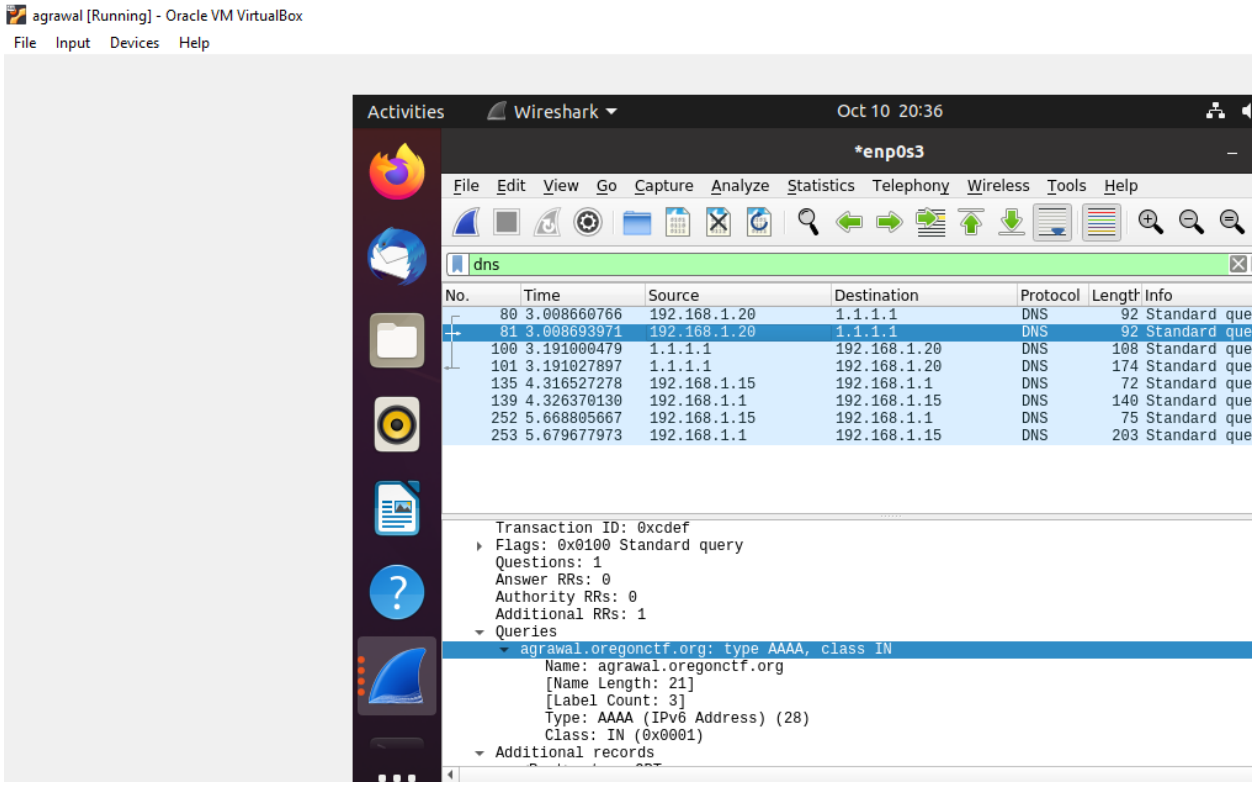
- **How many DNS requests are made?**

  2 DNS query requests were made – one for ipv4 and another for ipv6:

  i) From my machine's IP address (192.168.1.20) to the 1.1.1.1 dns server to resolve agrawal.oregonctf.org for type A record

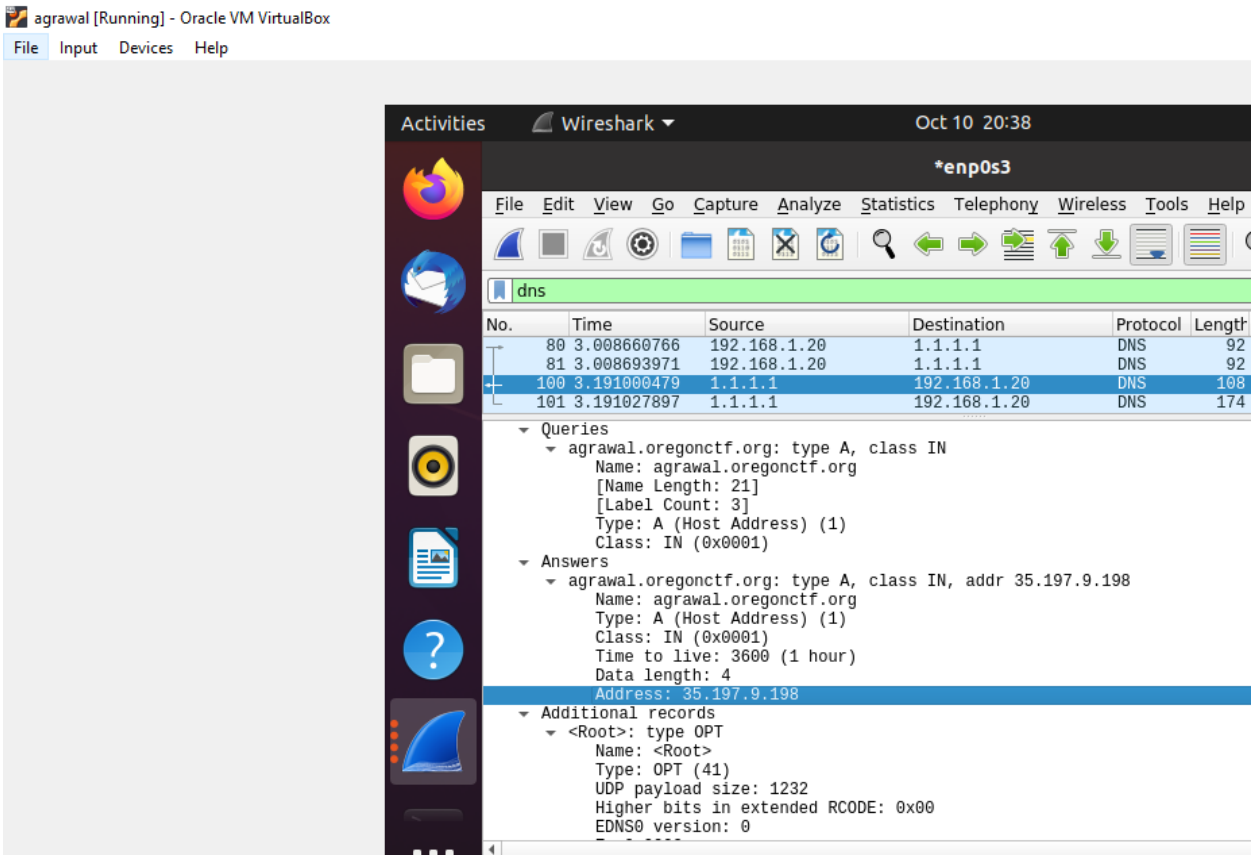


ii) From my machine's IP address (192.168.1.20) to the 1.1.1.1 dns server to resolve agrawal.oregonctf.org for type AAAA record



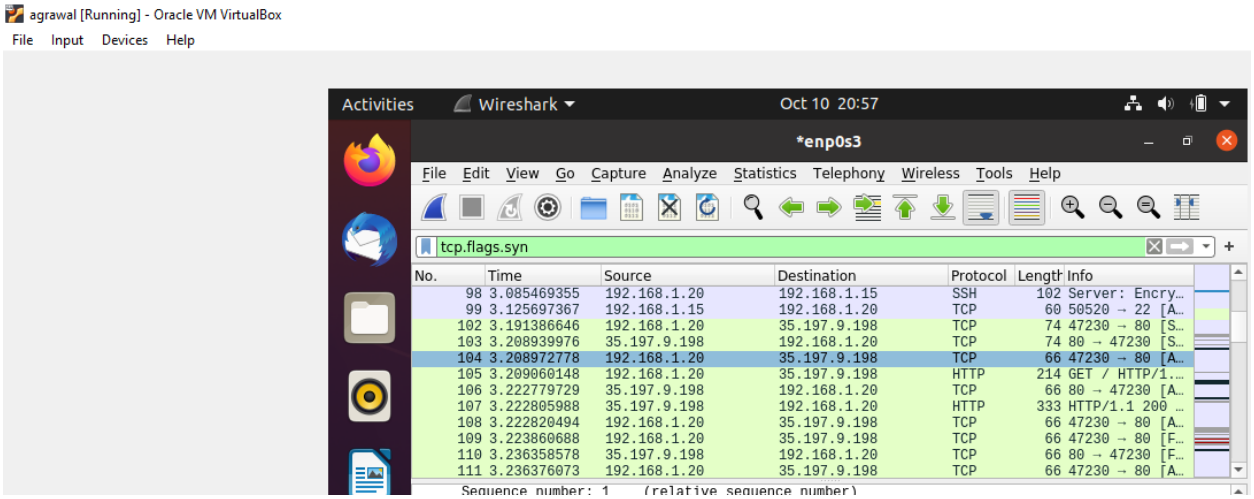
There were 2 DNS responses for the above two dns requests:

The IPv4 address returned: 35.197.9.198



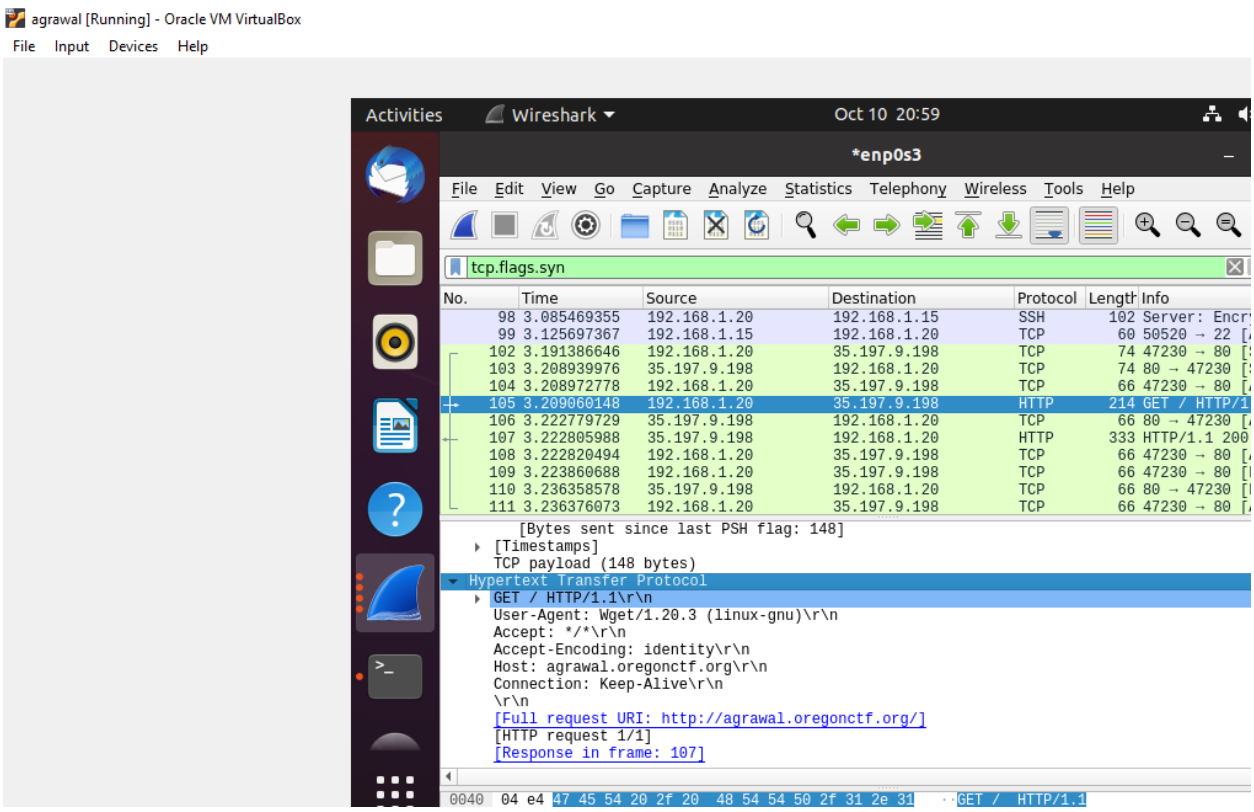
- **How many TCP connections does the browser initiate simultaneously to the site?**

  6 TCP connections were initiated from my browser to the 35.197.9.198



- **How many HTTP GET requests are there for embedded objects?**

  1 HTTP request


agrawal [Running] - Oracle VM VirtualBox
File Input Devices Help
Activities
Wireshark
Oct 10 20:59
*enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.flags.syn
No. Time Source Destination Protocol Length Info
98 3.085469355 192.168.1.20 192.168.1.15 SSH 102 Server: Encr
99 3.125697367 192.168.1.15 192.168.1.20 TCP 60 50520 → 22 [
102 3.191386646 192.168.1.20 35.197.9.198 TCP 74 47230 → 80 [
103 3.208939976 35.197.9.198 192.168.1.20 TCP 74 80 → 47230 [
104 3.208972778 192.168.1.20 35.197.9.198 TCP 66 47230 → 80 [
105 3.209060148 192.168.1.20 35.197.9.198 HTTP 214 GET / HTTP/1
106 3.222779729 35.197.9.198 192.168.1.20 TCP 66 80 → 47230 [
107 3.222805988 35.197.9.198 192.168.1.20 HTTP 333 HTTP/1.1 200
108 3.222820494 192.168.1.20 35.197.9.198 TCP 66 47230 → 80 [
109 3.223860688 192.168.1.20 35.197.9.198 TCP 66 47230 → 80 [
110 3.236358578 35.197.9.198 192.168.1.20 TCP 66 80 → 47230 [
111 3.236376073 192.168.1.20 35.197.9.198 TCP 66 47230 → 80 [
[Bytes sent since last PSH flag: 148]
[Timestamps]
TCP payload (148 bytes)
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
User-Agent: Wget/1.20.3 (linux-gnu)\r\n
Accept: */*\r\n
Accept-Encoding: identity\r\n
Host: agrawal.oregonctf.org\r\n
Connection: Keep-Alive\r\n
\r\n
[Full request URI: http://agrawal.oregonctf.org/]
[HTTP request 1/1]
[Response in frame: 107]
0040 04 e4 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31 ··GET / HTTP/1.1