02.1: TCP, HTTP

TCP #1 (netstat, lsof, nc)

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

```
hali5@instance-1:~$ sudo netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      502/sshd: /usr/sbin
tcp6       0      0 :::22                   :::*                    LISTEN      502/sshd: /usr/sbin
```

Note: This is done through Google Compute Engine VM because WSL2 does not provide any information when running the netstat -tlnp command.

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.

ssh 22/tcp # SSH Remote Login 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?

None of the ports on the Google Compute Engine VM only have a number.

Login to linux.cs.pdx.edu. Run the netstat command again, but do not use sudo as this is a machine managed by CAT. Include a screenshot of the output.

```
hali5@ada:~$ netstat -tlnp
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:49311           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6013          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6012          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6015          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6014          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6011          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6023          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6022          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6016          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6018          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6025          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6024          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6027          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:6026          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:39087         0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::51701                :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 :::113                  :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 ::1:6016                :::*                    LISTEN      -
tcp6       0      0 ::1:6018                :::*                    LISTEN      -
tcp6       0      0 ::1:6022                :::*                    LISTEN      -
tcp6       0      0 ::1:6023                :::*                    LISTEN      -
tcp6       0      0 ::1:6024                :::*                    LISTEN      -
tcp6       0      0 ::1:6025                :::*                    LISTEN      -
tcp6       0      0 ::1:6026                :::*                    LISTEN      -
tcp6       0      0 ::1:6027                :::*                    LISTEN      -
tcp6       0      0 ::1:6010                :::*                    LISTEN      -
tcp6       0      0 ::1:6011                :::*                    LISTEN      -
tcp6       0      0 ::1:6012                :::*                    LISTEN      -
tcp6       0      0 ::1:6013                :::*                    LISTEN      -
tcp6       0      0 ::1:6014                :::*                    LISTEN      -
tcp6       0      0 ::1:6015                :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 ::1:25                  :::*                    LISTEN      -
hali5@ada:~$
```

What services does this machine provide for external access?

25 - Simple Mail Transfer Protocol

22 - SSH Remote Login Protocol
111 - Sun Remote Procedure Call
113 - Authentication service used in SUN Remote Procedure Call
631 - Internet Printing Protocol

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

```
hali5@instance-1:~$ sudo lsof -i -s
COMMAND    PID    USER   FD   TYPE DEVICE SIZE NODE NAME
dhclient   327    root    9u  IPv4  10307       UDP *:bootpc
chronyd    399 _chrony    5u  IPv4  10377       UDP localhost:323
chronyd    399 _chrony    6u  IPv6  10378       UDP localhost:323
google_gu 402     root   11u  IPv4  11432       TCP instance-1.c.cloud-tran-hali5.internal:53192->metadata.google.internal:http (ESTABLISHED)
google_os 471     root    3u  IPv4  10448       TCP instance-1.c.cloud-tran-hali5.internal:53066->metadata.google.internal:http (ESTABLISHED)
sshd       502    root    3u  IPv4  10625       TCP *:ssh (LISTEN)
sshd       502    root    4u  IPv6  10627       TCP *:ssh (LISTEN)
sshd       572    root    4u  IPv4  10727       TCP instance-1.c.cloud-tran-hali5.internal:ssh->35.235.241.209:46261 (ESTABLISHED)
sshd       594   hali5    4u  IPv4  10727       TCP instance-1.c.cloud-tran-hali5.internal:ssh->35.235.241.209:46261 (ESTABLISHED)
hali5@instance-1:~$
```

Include for your lab notebook, the version of ssh that is being used.

SSH-2.0-OpenSSH_8.9p1

Throughput test

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.

```
hali5@vm-us-west1-b:~$ iperf -c 35.244.111.182 -p 80
------------------------------------------------------------
Client connecting to 35.244.111.182, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.12 port 47320 connected with 35.244.111.182 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   171 MBytes   142 Mbits/sec
hali5@vm-us-west1-b:~$ iperf -c 35.195.135.81 -p 80
------------------------------------------------------------
Client connecting to 35.195.135.81, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.12 port 41350 connected with 35.195.135.81 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   189 MBytes   157 Mbits/sec
hali5@vm-us-west1-b:~$ iperf -c 35.231.39.142 -p 80
------------------------------------------------------------
Client connecting to 35.231.39.142, TCP port 80
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 10.138.0.12 port 46464 connected with 35.231.39.142 port 80
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   416 MBytes   348 Mbits/sec
hali5@vm-us-west1-b:~$
```

The relative difference between the throughput and bandwidth of the connection to Australia, Europe, and US East VM's respectively are due to their distance from the client's VM (US-West). Australia, being the furthest, had the lowest throughput and bandwidth, followed by Europe which is the second farthest, and lastly US East, being the closest, had the highest throughput and bandwidth.

Developer Tools

First Request:

What is the URL being requested?

The URL being requested is http://google.com.

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

The HTTP status code in the response is 301 Moved Permanently which indicates that the requested resource has been definitely/permanently moved to the URL given by the location header in the response.

Second request:

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

The URL being requested is http://www.google.com/. It is using HTTP.

What are the Host: (HTTP 1.1) or :authority: (HTTP 2.0) headers sent by the browser? What is the User-Agent: HTTP header that is sent?

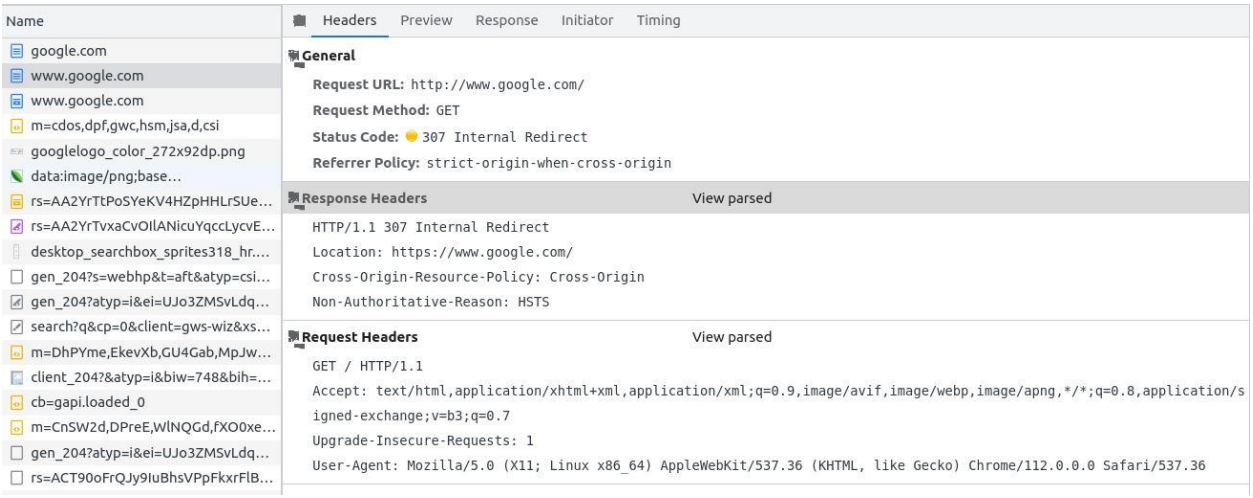
The authority header sent by the browser is google.com. The user-agent header that is sent is *Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36*

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?

The HTTP status code in the response is a 307 Internal Redirect. This status code means the resource requested has been temporarily moved to the URL given by the location header and

commands the browser to redirect to that new URL. Yes, it is different from the first status code (301 Moved Permanently).

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



Third Request:

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

The URL being requested is https://www.google.com/. It is using HTTPS.

What is the HTTP status code in the response?

The HTTP status code in the response was 200 Ok status which means the request was successful and the server was able to fetch and deliver what was needed (web page in this case).

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

Yes, the server believes the client can use HTTP3/QUIC. The value of the alt-svc key is *h3=":443"; ma=2592000,h3-29=":443"; ma=2592000*

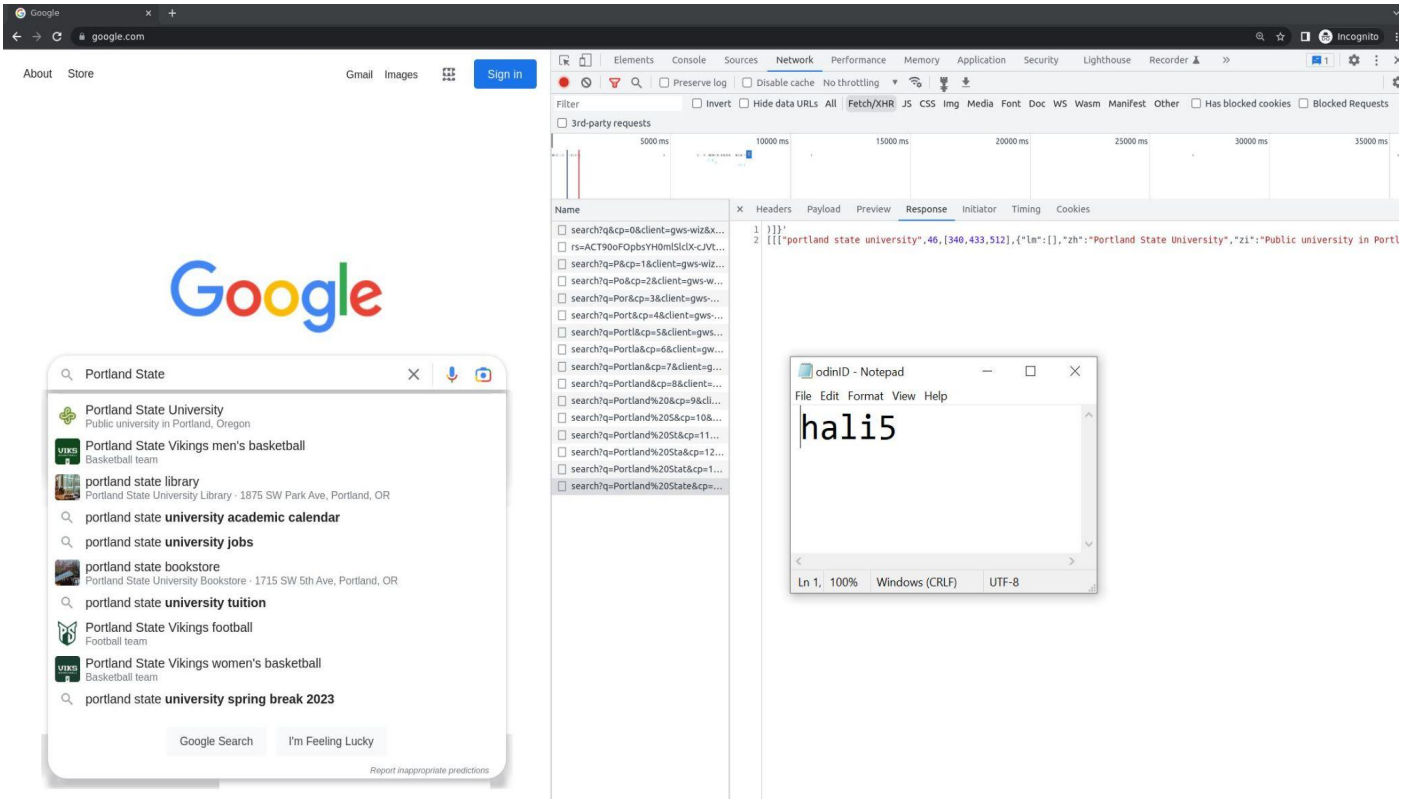
Examine the HTTP response headers for cookies. Show the cookies that are set and which ones specify that no SameSite restrictions are in place. What does the setting indicate about the cookies that are set?

The cookies that were sent by the HTTP response as well as the ones that specify that no SameSite restrictions are in place was: *1P_JAR=2023-04-13-01; expires=Sat, 13-May-2023 01:07:37 GMT; path=/; domain=.google.com; Secure; SameSite=none*

The SameSite=None setting indicates that the cookie that is set is designated for cross-site access, meaning external resources on the web page may access the cookie despite the domain name of that cookie not matching the site domain name.

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

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?

The ANSWER sections explain that PSUs web services are run from a single server since there is a single IP address for the result of www.pdx.edu, while PSU's mail services are run multiple Google servers for

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.

```
hali5@ada:~$ dig @127.0.0.53 -t NS mashimaro.cs.pdx.edu +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @127.0.0.53 -t NS mashimaro.cs.pdx.edu +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60021
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      NS

;; AUTHORITY SECTION:
cs.pdx.edu.             247     IN      SOA     walt.ee.pdx.edu. support.cat.pdx.edu. 2023041000 600 300 1209600 300

;; Query time: 3 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (TCP)
;; WHEN: Fri Apr 14 16:14:15 PDT 2023
;; MSG SIZE  rcvd: 105

hali5@ada:~$ dig @walt.ee.pdx.edu -t A mashimaro.cs.pdx.edu +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @walt.ee.pdx.edu -t A mashimaro.cs.pdx.edu +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47568
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7221b1e5a002a3b1010000006439de678b000a770208a07e (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

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

;; Query time: 3 msec
;; SERVER: 131.252.208.38#53(walt.ee.pdx.edu) (TCP)
;; WHEN: Fri Apr 14 16:14:47 PDT 2023
;; MSG SIZE  rcvd: 93

hali5@ada:~$
```

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

```
hali5@ada:~$ dig @127.0.0.53 -t NS thefengs.com +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @127.0.0.53 -t NS thefengs.com +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65122
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;thefengs.com.                  IN      NS

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

;; ADDITIONAL SECTION:
ns-cloud1.googledomains.com. 58594 IN   A       216.239.32.106
ns-cloud2.googledomains.com. 132491 IN  A       216.239.34.106
ns-cloud3.googledomains.com. 132491 IN  A       216.239.36.106
ns-cloud4.googledomains.com. 132491 IN  A       216.239.38.106
ns-cloud1.googledomains.com. 72827 IN   AAAA    2001:4860:4802:32::6a
ns-cloud2.googledomains.com. 72827 IN   AAAA    2001:4860:4802:34::6a
ns-cloud3.googledomains.com. 72827 IN   AAAA    2001:4860:4802:36::6a
ns-cloud4.googledomains.com. 168039 IN  AAAA    2001:4860:4802:38::6a

;; Query time: 59 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (TCP)
;; WHEN: Fri Apr 14 18:47:51 PDT 2023
;; MSG SIZE  rcvd: 327

hali5@ada:~$ dig @ns-cloud2.googledomains.com -t A thefengs.com +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @ns-cloud2.googledomains.com -t A thefengs.com +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18770
;; 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(ns-cloud2.googledomains.com) (TCP)
;; WHEN: Fri Apr 14 18:48:45 PDT 2023
;; MSG SIZE  rcvd: 57
```

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)

The server knows which site to serve from by examining the Host header within the HTTP request.

Include the results of each query for your lab notebook (DNS iterative lookups)

```
hali5@ada:~$ dig @f.root-servers.net www.cs.pdx.edu -t A +norecurse +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @f.root-servers.net www.cs.pdx.edu -t A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55881
;; 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      e.edu-servers.net.
edu.                    172800  IN      NS      b.edu-servers.net.
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.
edu.                    172800  IN      NS      i.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
edu.                    172800  IN      NS      j.edu-servers.net.
edu.                    172800  IN      NS      k.edu-servers.net.
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
edu.                    172800  IN      NS      h.edu-servers.net.
edu.                    172800  IN      NS      l.edu-servers.net.
edu.                    172800  IN      NS      m.edu-servers.net.
```

```
hali5@ada:~$ dig @f.edu-servers.net www.cs.pdx.edu -t A +norecurse +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @f.edu-servers.net www.cs.pdx.edu -t A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53786
;; 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: 23 msec
;; SERVER: 192.35.51.30#53(f.edu-servers.net) (TCP)
;; WHEN: Sat Apr 15 11:05:49 PDT 2023
;; MSG SIZE  rcvd: 164
```

```
hali5@ada:~$ dig @dns1.pdx.edu www.cs.pdx.edu -t A +norecurse +tcp

; <<>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <<>> @dns1.pdx.edu www.cs.pdx.edu -t A +norecurse +tcp
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41958
;; flags: qr aa ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: f4fc19560612537d43bfb20d643ae82369642425f0a8c308 (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      dns0.pdx.edu.
cat.pdx.edu.            14400   IN      NS      dns1.pdx.edu.
cat.pdx.edu.            14400   IN      NS      walt.ee.pdx.edu.

;; ADDITIONAL SECTION:
walt.ee.pdx.edu.        14400   IN      A       131.252.208.38

;; Query time: 3 msec
;; SERVER: 131.252.120.129#53(dns1.pdx.edu) (TCP)
;; WHEN: Sat Apr 15 11:08:35 PDT 2023
;; MSG SIZE  rcvd: 195
```

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.

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.

```
hali5@ada:~$ dig espn.go.com | egrep ^espn.go.com | awk '{print $5}'
18.65.229.93
18.65.229.26
18.65.229.61
18.65.229.14
hali5@ada:~$ X=`dig espn.go.com | egrep ^espn.go.com | awk '{print $5}'`
hali5@ada:~$ for i in `echo $X`; do dig -x $i | egrep server | awk '{print $5}'; done
server-18-65-229-61.sea73.r.cloudfront.net.
server-18-65-229-26.sea73.r.cloudfront.net.
server-18-65-229-14.sea73.r.cloudfront.net.
server-18-65-229-93.sea73.r.cloudfront.net.
```

Host enumeration

```
hali5@ada:~$ for ip in {0..255}; do dig -x 131.252.220.$ip; done | egrep cs.pdx.edu | awk '{print $5}' > 220hosts.txt
hali5@ada:~$ cat 220hosts.txt | head -1000 | tail -1000
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.
rooster.cs.pdx.edu.
sheep.cs.pdx.edu.
snake.cs.pdx.edu.
tiger.cs.pdx.edu.
assault.cs.pdx.edu.
aztec.cs.pdx.edu.
backalley.cs.pdx.edu.
cbble.cs.pdx.edu.
dust.cs.pdx.edu.
estate.cs.pdx.edu.
havana.cs.pdx.edu.
inferno.cs.pdx.edu.
italy.cs.pdx.edu.
militia.cs.pdx.edu.
```

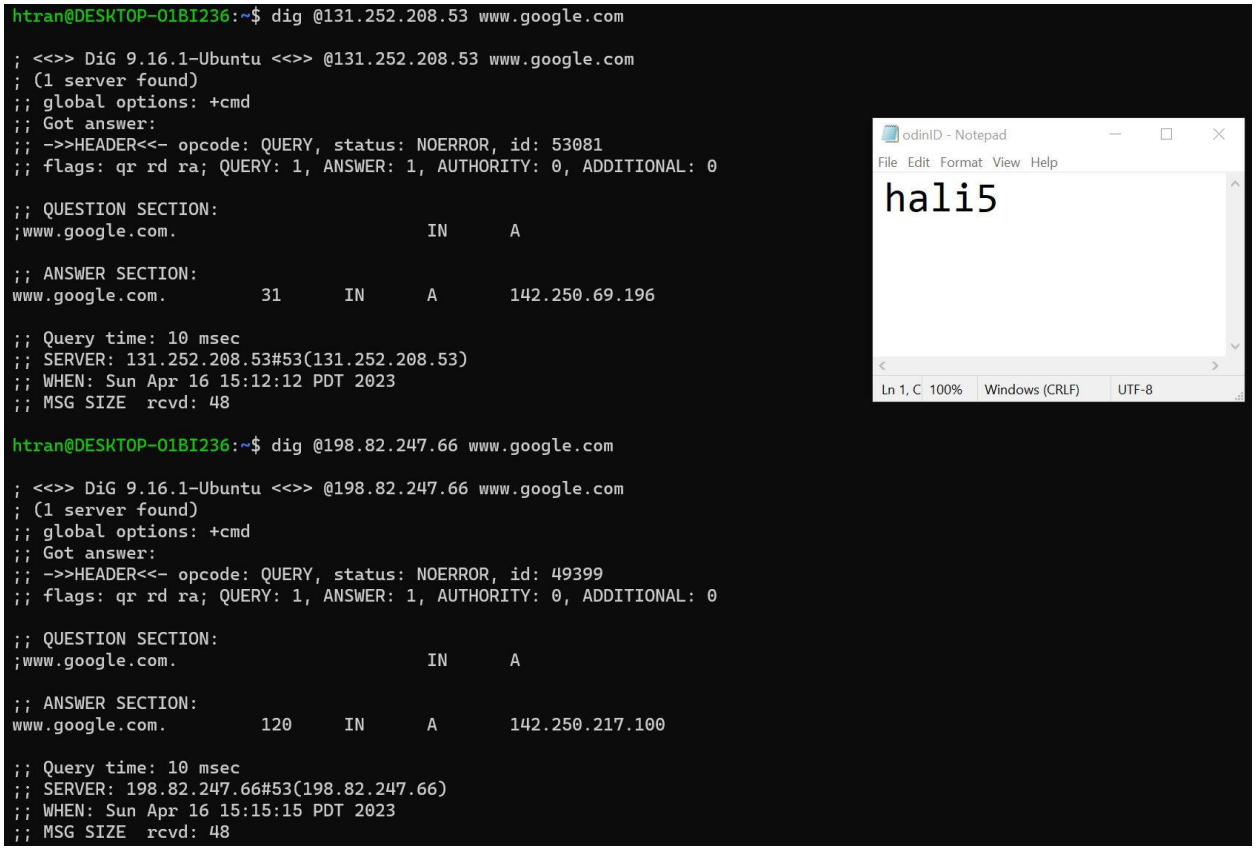
The screenshot does not show all of the results of the 220hosts text file, only a snippet since it is quite large.

DNS #2 (Geographic DNS)

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

The geographical location that ipinfo.io and DB-IP return for the IP address of the DNS servers 131.252.208.53 and 198.82.247.66 were Oregon, USA and Virginia, USA respectively.

Record each result for your lab notebook.



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

The geographical distance between the Oregon DNS server and the Seattle Washington Google server is 170 miles and the distance between the Virginia DNS server and Seattle Washington Google server is over 2,500 miles.

Do the routes reveal any information on the accuracy of the geographic locations given?

Yes, the traceroute verifies the accuracy of the geographical locations given by the number of hops that the data packet travels to. For the IP address of servers located in Oregon, there were less hops than that of the server located in Virginia which is due to the distance of the server. This result was also true for the IP address resolved for www.google.com.

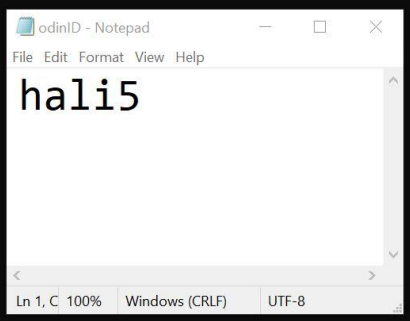
Perform traceroutes on all 4 IP addresses.

```
hali5@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)  1.153 ms  1.082 ms  1.033 ms
hali5@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)  2.073 ms  2.094 ms  2.169 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.854 ms  0.827 ms  0.769 ms
 3  131.252.5.213 (131.252.5.213)  0.940 ms  0.891 ms  0.894 ms
 4  port-psu-pe-01.net.linkoregon.org (199.165.177.48)  0.775 ms  0.723 ms  0.671 ms
 5  eugn-oh-vpn-01.net.linkoregon.org (207.98.126.3)  10.387 ms  10.323 ms  10.254 ms
 6  bois-gtwy-pe-01.net.linkoregon.org (207.98.126.135)  10.413 ms  10.376 ms  10.354 ms
 7  bois-gtwy-pe-01-loren.net.linkoregon.org (163.253.5.65)  10.000 ms  9.960 ms  9.987 ms
 8  hundredge-0-0-0-24.703.core1.bois.net.internet2.edu (163.253.5.64)  12.497 ms  12.433 ms  12.370 ms
 9  fourhundredge-0-0-0-0.4079.core2.salt.net.internet2.edu (163.253.1.249)  65.022 ms  65.665 ms  65.576 ms
10  fourhundredge-0-0-0-23.4079.core1.salt.net.internet2.edu (163.253.1.32)  63.641 ms fourhundredge-0-0-0-22.4079.core1.salt.net.internet2.edu (163.253.1.30)  66.293 ms
 fourhundredge-0-0-0-0.4079.core2.denv.net.internet2.edu (163.253.1.168)  65.020 ms
11  fourhundredge-0-0-0-0.4079.core1.denv.net.internet2.edu (163.253.1.170)  79.764 ms  65.369 ms  65.306 ms
12  fourhundredge-0-0-0-0.4079.core1.kans.net.internet2.edu (163.253.1.243)  64.633 ms  64.520 ms fourhundredge-0-0-0-0.4079.core1.chic.net.internet2.edu (163.253.2.28)
 64.451 ms
13  fourhundredge-0-0-0-3.4079.core2.chic.net.internet2.edu (163.253.1.244)  64.769 ms  64.720 ms  64.693 ms
14  fourhundredge-0-0-0-3.4079.core2.eqch.net.internet2.edu (163.253.2.19)  68.315 ms  64.799 ms  64.700 ms
15  fourhundredge-0-0-0-0.4079.core2.clev.net.internet2.edu (163.253.2.16)  64.631 ms  64.566 ms  65.057 ms
16  fourhundredge-0-0-0-3.4079.core2.ashb.net.internet2.edu (163.253.1.138)  65.752 ms  65.684 ms  65.616 ms
17  192.122.175.14 (192.122.175.14)  64.755 ms  64.700 ms  64.627 ms
18  vtacs-1.msap.cns.vt.edu (192.70.187.18)  67.652 ms  67.584 ms  67.519 ms
19  hill-border.xe-5-0-2.0.cns.vt.edu (128.173.0.194)  67.549 ms  67.566 ms  67.889 ms
20  128.173.0.210 (128.173.0.210)  68.730 ms  68.660 ms  68.606 ms
21  cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  67.494 ms  67.462 ms  67.334 ms
22  jeru.cns.vt.edu (198.82.247.66)  67.264 ms  67.200 ms  67.197 ms
hali5@ada:~$ traceroute 142.250.69.196
traceroute to 142.250.69.196 (142.250.69.196), 30 hops max, 60 byte packets
 1  radiant.seas.pdx.edu (131.252.208.212)  2.019 ms  1.483 ms  1.560 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.853 ms  0.807 ms  0.767 ms
 3  131.252.5.213 (131.252.5.213)  1.621 ms  1.572 ms  0.868 ms
 4  google.nwax.net (198.32.195.34)  21.174 ms  21.110 ms  21.044 ms
 5  74.125.243.177 (74.125.243.177)  5.294 ms  5.225 ms 74.125.243.193 (74.125.243.193)  4.057 ms
 6  142.251.48.211 (142.251.48.211)  4.524 ms  4.200 ms  4.270 ms
 7  sea30s08-in-f4.1e100.net (142.250.69.196)  4.227 ms  4.024 ms  4.137 ms
hali5@ada:~$ traceroute 142.250.217.100
traceroute to 142.250.217.100 (142.250.217.100), 30 hops max, 60 byte packets
 1  radiant.seas.pdx.edu (131.252.208.212)  1.390 ms  4.446 ms  4.405 ms
 2  CORE1.net.pdx.edu (131.252.5.142)  0.886 ms  0.845 ms  0.794 ms
 3  131.252.5.213 (131.252.5.213)  0.976 ms  0.930 ms  4.032 ms
 4  google.nwax.net (198.32.195.34)  4.187 ms  4.116 ms  4.124 ms
 5  108.170.245.113 (108.170.245.113)  4.165 ms 108.170.245.97 (108.170.245.97)  5.090 ms 108.170.245.113 (108.170.245.113)  3.891 ms
 6  142.251.55.203 (142.251.55.203)  4.358 ms  4.193 ms 142.251.55.201 (142.251.55.201)  4.365 ms
 7  sea09s30-in-f4.1e100.net (142.250.217.100)  3.847 ms  3.969 ms  4.230 ms
hali5@ada:~$
```

Network Recap Lab #3

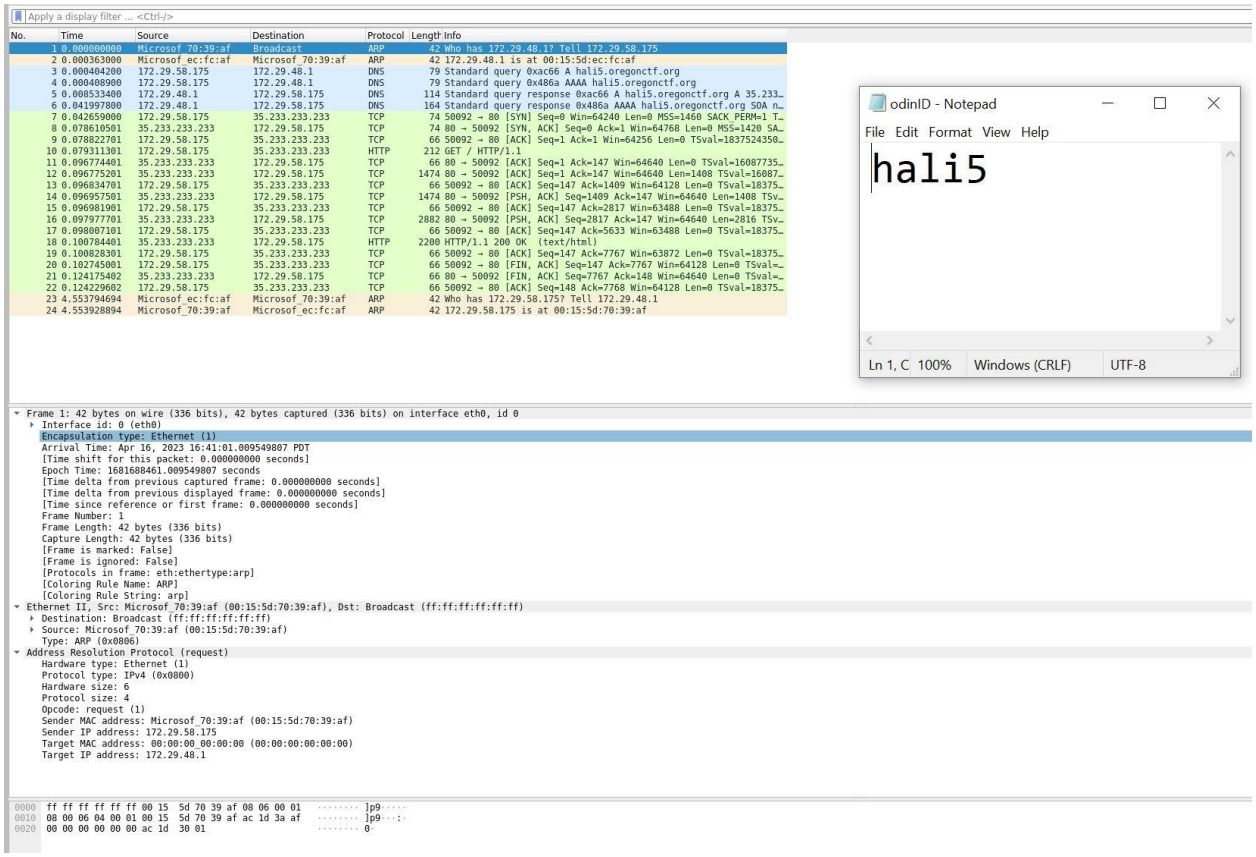
Perform a reverse DNS lookup on the DNS server to find its name. Include it in your lab notebook.

```
htran@DESKTOP-O1BI236:~$ 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: 34419
;; 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.   983     IN      PTR     one.one.one.one.

;; Query time: 10 msec
;; SERVER: 1.1.1.1#53(1.1.1.1)
;; WHEN: Sat Apr 15 14:36:28 PDT 2023
;; MSG SIZE  rcvd: 78
```



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.



How many DNS requests are made?

4 DNS requests were made.

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

The browser initiates 14 TCP connections simultaneously to the site.

How many HTTP GET requests are there for embedded objects?

There are 1 HTTP GET request for embedded objects.