

# Week #2 Labs

<b>2.1: TCP, HTTP</b>	<b>2</b>
1. TCP #1: Sockets	2
2. -	4
4. -	5
5. HTTP #3: Requests	6
6. -	8
<b>2.2: DNS, Recap</b>	<b>9</b>
1. DNS reconnaissance #1 (dig)	9
2. Iterative DNS lookups	13
3. Reverse DNS lookups	15
4. Host enumeration	17
5. Geographic DNS #5	17
6/7. Wireshark Lab #3, -	19
10. Analyze trace	20

## 2.1: TCP, HTTP

### 1. TCP #1: Sockets



Using ChatGPT, find a single command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it. Note that you will need to run this command using sudo since administrator privileges are required to list resources being used by other programs.



To list all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using them (requiring `sudo` for admin privileges), you can use the following command:

bash

```
sudo ss -ltp4
```

Copy code

metens

#### Explanation:

- `sudo` : Run the command with superuser privileges to see processes being used by other programs.
- `ss` : Utility to investigate sockets.
- `-l` : Only display listening sockets.
- `-t` : Filter for TCP sockets.
- `-p` : Show the process using each socket.
- `4` : Restrict the output to IPv4 addresses.

Running this command will provide you with the list of all listening IPv4 TCP sockets along with the associated processes.

```
ssh.cloud.google.com/v2/ssh/projects/cloud-metens/zones/us-west1-a/instances/course-vm2?authuser=0&hl=en_US&projectN...
ssh.cloud.google.com/v2/ssh/projects/cloud-metens/zones/us-west1-a/instances/course-vm2?authuser=0&hl=en_US&pr...
SSH-in-browser
Linux course-vm2 6.1.0-25-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct  7 03:48:40 2024 from 35.235.241.65
metens@course-vm2:~$ sudo ss -ltp4
State  Recv-Q Send-Q Local Address:Port          Peer Address:Port  Process
LISTEN 0        128      0.0.0.0:ssh             0.0.0.0:*          users:(("sshd",pid=800,fd=3))
LISTEN 0        4096     127.0.0.54:domain       0.0.0.0:*          users:(("systemd-resolve",pid=365,fd=21))
LISTEN 0        4096     0.0.0.0:5355            0.0.0.0:*          users:(("systemd-resolve",pid=365,fd=12))
LISTEN 0        4096     127.0.0.53:lo:domain      0.0.0.0:*          users:(("systemd-resolve",pid=365,fd=19))
LISTEN 0        20       127.0.0.1:smtp           0.0.0.0:*          users:(("exim4",pid=706,fd=4))

metens@course-vm2:~$
```

Based on the previous screenshot, a service that can be contacted from any interface on the machine is **0.0.0.0:ssh** and a service that can only be contacted by local processes is **127.0.0.1:smtp**.

```
[metens@ada:~$ ss -ltcp4
State    Recv-Q   Send-Q      Local Address:Port      Peer Address:Port  Process
LISTEN     0       128          127.0.0.1:6100        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6101        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6102        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6103        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6104        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6105        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6106        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6107        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6108        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6109        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6110        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6111        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6112        0.0.0.0:*
LISTEN     0       128          127.0.0.1:6113        0.0.0.0:*
LISTEN     0      4096         127.0.0.53%lo:domain  0.0.0.0:*
LISTEN     0       100          127.0.0.1:smtp        0.0.0.0:*
LISTEN     0      4096         127.0.0.1:ipp         0.0.0.0:*
LISTEN     0       511          127.0.0.1:38341       0.0.0.0:*
LISTEN     0      4096         127.0.0.54:domain     0.0.0.0:*
metens@ada:~$
```

Since the previous screenshot only lists internal access, I changed the command:

```
[metens@ada:~$ ss -ltunp4 | grep -v '127.0.0.1'
Netid State  Recv-Q Send-Q      Local Address:Port  Peer Address:PortProcess
udp  UNCONN  0     0          0.0.0.0:36880      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:38641      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:39407      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:39512      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:40050      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:49477      0.0.0.0:*
udp  UNCONN  0     0          0.0.0.0:52699      0.0.0.0:*
udp  UNCONN  0     0          127.0.0.54:53       0.0.0.0:*
udp  UNCONN  0     0          127.0.0.53%lo:53    0.0.0.0:*
udp  UNCONN  0     0          131.252.208.103%ens3:68 0.0.0.0:*
tcp  LISTEN  0     4096       127.0.0.53%lo:53    0.0.0.0:*
tcp  LISTEN  0     4096       127.0.0.54:53       0.0.0.0:*
metens@ada:~$
```

Here, we can see that there are some externally accessible services such as **0.0.0.0:36880**, **0.0.0.0:38641**, and **0.0.0.0:52699**.

2. -

 Using ChatGPT, find a single `lsof` command and its command-line flags that, when executed, lists all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using it. Note that you can leverage the conversation in the previous step and simply ask ChatGPT to repeat the task using `lsof`.

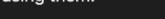
To list all TCP sockets in a LISTEN state on an IPv4 address, showing the program that is using them using `lsof`, you can use the following command:

```
bash  Copy code  
sudo lsof -iTCP -sTCP:LISTEN -nP | grep IPv4
```

**Explanation:**

- `sudo` : Run the command with superuser privileges to view sockets opened by other users or processes.
- `lsof` : Command to list open files.
- `-iTCP` : Show only TCP network connections.
- `-sTCP:LISTEN` : Filter for TCP sockets in the LISTEN state.
- `-n` : Prevent DNS resolution (speeds up output by showing raw IP addresses).
- `-P` : Show raw port numbers instead of service names.
- `grep IPv4` : Filters the output to show only IPv4 addresses.

This command will give you a list of all listening IPv4 TCP sockets, along with the program that is using them.



#### 4. -

```
metens@instance-1-us-west1-b:~$ 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)
-----
[ 1] local 10.138.0.9 port 58146 connected with 10.142.0.2 port 80
[ ID] Interval      Transfer      Bandwidth
[ 1] 0.0000-10.0933 sec   374 MBytes   311 Mbits/sec
metens@instance-1-us-west1-b:~$ 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)
-----
[ 1] local 10.138.0.9 port 57634 connected with 10.152.0.2 port 80
[ ID] Interval      Transfer      Bandwidth
[ 1] 0.0000-10.1993 sec   183 MBytes   151 Mbits/sec
metens@instance-1-us-west1-b:~$ iperf -c 10.132.0.2 -p 80
-----
Client connecting to 10.132.0.2, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[ 1] local 10.138.0.9 port 50532 connected with 10.132.0.2 port 80
[ ID] Interval      Transfer      Bandwidth
[ 1] 0.0000-10.2361 sec   194 MBytes   159 Mbits/sec
metens@instance-1-us-west1-b:~$ █
```

- 10.142.0.2 refers to us-east1-c and its bandwidth is 311 Mbits/sec.
- 10.152.0.2 refers to australia and its bandwidth is 151 Mbits/sec.
- 10.132.0.2 refers to europe with a bandwidth of 159 Mbits/sec.

It appears that more data is available to be transmitted the closer a VM is located to the origin.  
Both Australia and Europe are much further than us-east1-c is to the us-west1-b VM.

## 5. HTTP #3: Requests

Name	Status	Type	Initiator	Size
google.com	301	document / Redir...	google.com/	(disk)
www.google.com	200	document	google.com/	
google.com	200	document / Redir...	Other	
m=cdos,hsm,jsa,mb4ZUb,d,csi,cEt90b,SNUn3,qddgKe,stSdM...	200	stylesheet	(index):15	
m=cdos,hsm,jsa,mb4ZUb,d,csi,cEt90b,SNUn3,qddgKe,stSdM...	200	script	(index):15	
googlelogo_color_272x92dp.png	200	png	(index):63	
hpba?v=3&cs=0&ei=KX0iZ7LHCebH0PEPn5jwiA4&async=...	200	xhr	(index):68	
rs=A2YrTuWzR02q-lADuEe1tzdVix-MATZjg	200	png	(index):117	
rs=A2YrTuTufgcOsX6KH6kYXj5labmF8SGA	200	script	(index):114	
desktop_searchbox_sprites318_hr.webp	200	stylesheet	(index):121	
gen_2047s=webhp&t=aft&atyp=csi&ei=KX0iZ7LHCebH0PEP...	204	ping	(index):12	
gen_2047atyp=csi&ei=KX0iZ7LHCebH0PEPnjiwA4&s=webh...	204	ping	m=cdos,hsm,js...	
gen_2047s=async&atyp=hpba&atyp=csi&ei=KX0iZ7CGCr....	204	text/html	(index):3	
search?q=cp=0&client=gws-wiz&xssi=t&gs_prcrt=2&hl=e...Iz...	200	xhr	m=cdos,hsm,js...	
m=sb_wiz,aa,abd,sysk,sys,syse,syfx,sy,sy4,sy10...,sybs...	200	script	m=cdos,hsm,js...	
rs=ACT90oHA8s-nYSOYY3G...	200	fetch	m=cdos,hsm,js...	
client_2047atyp=&hlbw=4198&hlbi=831&pr=2&ei=KX0iZ7LHC...	204	text/html	(index):3	

34 requests | 70.7 kB transferred | 2.3 MB resources | Finish: 2.46 s | DOMContentLoaded: 199 ms | Load: 358 ms

The URL being requested for the first request is <http://google.com/>. The returned status code for this request is **301 Temporary Redirection**.

Name	Headers
google.com	General
www.google.com	Request URL: https://google.com/
google.com	Request Method: GET
m=cdos,hsm,jsa,mb4ZUb,d,csi,cEt90b,SNUn3,qddgKe,stSdM...	Status Code: 301 Moved Permanently (from disk cache)
m=cdos,hsm,jsa,mb4ZUb,d,csi,cEt90b,SNUn3,qddgKe,stSdM...	Remote Address: [2607:f8b0:400a:803:200e]:443
googlelogo_color_272x92dp...	Referrer Policy: strict-origin-when-cross-origin
hpba?v=3&cs=0&ei=KX0iZ7L...	Response Headers
rs=A2YrTuWzR02q-lADuEe1t...	Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
rs=A2YrTuTufgcOsX6KH6kYX...	Cache-Control: public, max-age=2592000
desktop_searchbox_sprites31...	Content-Length: 220
gen_2047s=webhp&t=aft&atyp...	Content-Security-Policy-Report-Only: object-src 'none';base-uri 'self';script-src 'nonce-NrRbd3dhPc_6C report-sample' 'unsafe-eval' 'unsafe-inline' https://report-uri
gen_2047atyp=csi&ei=KX0iZ7...	Content-Type: text/html; charset=UTF-8
search?q=cp=0&client=gws-w...	Cross-Origin-Opener-Policy: same-origin-allow-popups; report-to="gws"
m=sb_wiz,aa,abd,sysk,sys,syse,syfx,sy,sy4,sy10...,sybs...	Date: Fri, 11 Oct 2024 00:55:03 GMT
rs=ACT90oHA8s-nYSOYY3G...	Expires: Sun, 10 Nov 2024 00:55:03 GMT
client_2047atyp=&hlbw=419...	Location: https://www.google.com/
	Permissions-Policy: unload()

34 requests | 70.7 kB transferred

As we can see on the Alt-Svc line in the Response Headers section, it says "h3", which indicates that the version is HTTP/3 and this is based on QUIC.

After the initial redirection, the second request goes to <https://www.google.com/>. The returned status code for this request is **200 OK**. The Alt-Svc line is again using the HTTP/3 version:

Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Cache-Control:	private, max-age=0
Content-Encoding:	br
Content-Length:	64008
Content-Security-Policy-Report-Only:	object-src 'none';base-u... report-sample' 'unsafe-eval' 'unsafe-inline' https://report-uri
	https://csp.withgoogle.com/csp/gws/other-hp

The final request goes to the actual URL for Google: <http://google.com/>. This time, the returned status code is **200 Temporary Redirect**. The version of the HTTP protocol is HTTP/3 again:

Accept: Ch...	Sec-Ch-Ua: "Not..."
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Cache-Control:	private, max-age=0
Content-Encoding:	br

metens

As we can see from the **location** response header in the initial request, the redirection is to <https://www.google.com/>:

▼ General	
Request URL:	<a href="http://google.com/">http://google.com/</a>
Request Method:	GET
Status Code:	301 Temporary Redirect
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers	
Alt-Svc:	h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
Cache-Control:	public, max-age=2592000
Content-Length:	220
Content-Security-Policy-Report-Only:	object-src 'none';base-uri 'self';script-src 'nonce-b... dynamic' 'report-sample' 'unsafe-eval' 'unsafe-inlin... https://csp.withgoogle.com/csp/report-to/gws/other-hp
Content-Type:	text/html; charset=UTF-8
Cross-Origin-Opener-Policy:	same-origin-allow-popups; report-to="gws"
Date:	Fri, 11 Oct 2024 01:49:19 GMT
Expires:	Sun, 10 Nov 2024 01:49:19 GMT
Location:	<a href="https://www.google.com/">https://www.google.com/</a>
Permissions-Policy:	unload=()



After the original request to Google we see the **Set-Cookie**:

Name	X	Headers	Preview	Response	Initiator	Timing	Cookies
google.com							report-sample unsafe-eval unsafe-inline https://csp.withgoogle.com/csp/report-to/gws/other-hp
www.google.com							text/html; charset=UTF-8
google.com							same-origin-allow-popups; report-to="gws"
m=cdos,hsm,jsa,mb4ZUb,d,cs...							Fri, 11 Oct 2024 01:49:20 GMT
m=cdos,hsm,jsa,mb4ZUb,d,cs...							-1
googlelogo_color_272x92dp.p...							P3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
data:image/png;base...							Permissions-Policy: unload=()
rs=AA2YrTuWzR02q-lADuEe1t...							Report-To: {"group": "gws", "max_age": 2592000, "endpoints": [{"url": "https://csp.withgoogle.com/csp/report-to/gws/other"}]}
rs=AA2YrTutjgcOSyX6KH6kYX...							Server: gws
desktop_searchbox_sprites31...							Set-Cookie: AEC=AVYB7cp6YVoJ7LtiZpCrIjSuRmg8Q-iTAL-kxLiHud-tH4r
gen_204?s=webhp&t=aft&at...							Apr-2025 01:49:20 GMT; path=/; domain=.google.com; Secure
gen_204?atyp=csi&ei=IIQIZ97...							NID=518=reTLJNz_zjTDDRA8Dl28ogMvHxvEdA2pzA5YHfg_HV
search?q=&cp=0&client=gws...							VgMBJAM4jtVxdXFU4M9YGuZOkY-Orb_fH2YYRpIrmPyCyc7LJ
m=sb_wiz,aa,abd,sysk,sysj,sy...							sctfj_XjcdOqn5UzFLeg58zyKmoTpjh5Dl2H3NS4elBQC1qpTb
rs=ACT90oHA4_TOuM7JuEJF...							2025 01:49:20 GMT; path=/; domain=.google.com; Secure; Http
client_204?atyp=i&biw=207&...							max-age=31536000
m=syjw,syo?xjs=s4							SAMEORIGIN
m=sy1ef,P10Owf,sy1d8,sy1d6,...							0
gen_204?atyp=i&ei=IIQIZ97F...							
client_204?cs=1&opi=899784...							
Request Headers							
:authority:							

metens

After refreshing the page, we get the Cookie:

The screenshot shows the Network tab in the Chrome DevTools. A request to `www.google.com` is selected. The Headers section displays the following key cookie entries:

- `:authority: www.google.com`
- `:method: GET`
- `:path: /`
- `:scheme: https`
- `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8`
- `Accept-Encoding: gzip, deflate, br, zstd`
- `Accept-Language: en-US,en;q=0.9`
- `Cache-Control: max-age=0`
- `Cookie: AEC=AVYB7cp6YVoJ7LtiZpCrljSuRmg8Q-iTAL-kxLiHud-tH4r...; NID=518=UsZBbpHipU-sqK6ZGJxQXyKD9Nu-...`
- `Priority: u=0, i`
- `Sec-Ch-Prefers-Color-Scheme: dark`
- `Sec-Ch-Ua: "Google Chrome", v="129", "N...`
- `Sec-Ch-Ua-Arch: "arm"`
- `Sec-Ch-Ua-Bitness: "64"`
- `Sec-Ch-Ua-Form-Factors: "Desktop"`
- `Sec-Ch-Ua-Full-Version: "129.0.6668.91"`

## 6. -

The screenshot shows the Network tab in the Chrome DevTools. A search request to `google.com` is selected. The Headers section shows the following search parameters:

- `search?q=Portland State`
- `search?oq=Portland State`
- `search?hl=en`
- `search?sa=X&tbo=exp`

The Response section shows the JSON results for the search query:

```
[{"id": 1, "label": "SearchResult", "text": "Portland State University\nUniversity in Portland, Oregon"}, {"id": 2, "label": "SearchResult", "text": "Portland State Vikings football\nFootball team"}, {"id": 3, "label": "SearchResult", "text": "Portland state university jobs\nSearch results for \"Portland state university jobs\""}, {"id": 4, "label": "SearchResult", "text": "Portland state farmers market\nPortland Farmers Market at PSU - 1803 SW Park Ave, Portland, OR"}, {"id": 5, "label": "SearchResult", "text": "Portland state library\nPortland State University Library - 1875 SW Park Ave, Portland, OR"}, {"id": 6, "label": "SearchResult", "text": "Portland state rec center\nPSU Campus Recreation Center - 1800 SW 6th Ave, Portland, OR"}, {"id": 7, "label": "SearchResult", "text": "Portland state academic calendar\nAcademic calendar"}, {"id": 8, "label": "SearchResult", "text": "Portland state football schedule\nFootball schedule"}, {"id": 9, "label": "SearchResult", "text": "Portland state bookstore\nPortland State University Bookstore - 1715 SW 5th Ave, Portland, OR"}, {"id": 10, "label": "SearchResult", "text": "Portland state university tuition\nTuition and fees"}]
```

## 2.2: DNS, Recap

### 1. DNS reconnaissance #1 (dig)

 produce a dig command that queries PSU's local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP

 To query PSU's local DNS server at 131.252.208.53 for the A record of www.pdx.edu using TCP, you can use the following `dig` command:

bash Copy code

```
dig +tcp @131.252.208.53 www.pdx.edu A
```

**Explanation:**

- `dig` : Command used to query DNS servers.
- `+tcp` : Forces the query to use TCP (instead of the default UDP).
- `@131.252.208.53` : Specifies the DNS server to query (PSU's local DNS server).
- `www.pdx.edu` : The domain name you're querying for.
- `A` : The type of DNS record to query for (IPv4 address).

This command will return the A record for www.pdx.edu from the DNS server 131.252.208.53 using TCP.

October 10, 2022

 Message ChatGPT metens|

Using the command provided by ChatGPT for the A record of [www.pdx.edu](http://www.pdx.edu) using TCP:

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 85x27
[metens@ada:~$ dig +tcp @131.252.208.53 www.pdx.edu A
; <>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <>> +tcp @131.252.208.53 www.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46914
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 1232
; COOKIE: 7ddca4ce0bf78e1301000006708914e161c26e8df0cd7d3 (good)
;; QUESTION SECTION:
;www.pdx.edu.          IN      A
;;
;; ANSWER SECTION:
www.pdx.edu.      51      IN      A      108.138.94.58
www.pdx.edu.      51      IN      A      108.138.94.27
www.pdx.edu.      51      IN      A      108.138.94.13
www.pdx.edu.      51      IN      A      108.138.94.85
;;
;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Thu Oct 10 19:45:34 PDT 2024
;; MSG SIZE rcvd: 132
metens@ada:~$ ]
```

Using the command provided by ChatGPT for the MX record of [pdx.edu](http://pdx.edu) using TCP:

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 87x33
[metens@ada:~$ dig +tcp @131.252.208.53 pdx.edu MX
; <>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <>> +tcp @131.252.208.53 pdx.edu MX
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46763
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 3
;;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 1232
; COOKIE: 7cc1f98b83ac7ac0100000670891e4fde2382586c3058d (good)
;; QUESTION SECTION:
;pdx.edu.          IN      MX
;;
;; ANSWER SECTION:
pdx.edu.        72720   IN      MX      10 alt3.aspmx.l.google.com.
pdx.edu.        72720   IN      MX      5 alt2.aspmx.l.google.com.
pdx.edu.        72720   IN      MX      1 aspmx.l.google.com.
pdx.edu.        72720   IN      MX      10 alt4.aspmx.l.google.com.
pdx.edu.        72720   IN      MX      5 alt1.aspmx.l.google.com.
;;
;; ADDITIONAL SECTION:
aspmx.l.google.com. 158     IN      A      142.250.107.26
aspmx.l.google.com. 112     IN      AAAA    2607:f8b0:400e:c06::1b
;;
;; Query time: 1 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Thu Oct 10 19:48:04 PDT 2024
;; MSG SIZE rcvd: 226
metens@ada:~$ ]
```

Using the following IP address: 108.138.94.27 from the `dig +tcp @131.252.208.53 www.pdx.edu A` command in the previous screenshot we get this from the iplocations.net website:

Geolocation data from	DB-IP	Product: API, real-time
 IP ADDRESS: 108.138.94.27	 ISP: Amazon.com, Inc.	
 COUNTRY: United States 	 ORGANIZATION: Amazon.com, Inc.	
 REGION: Oregon	 LATITUDE: 45.5152	
 CITY: Portland	 LONGITUDE: -122.678	
Incorrect location? Contact DB-IP		 <a href="#">view map</a>

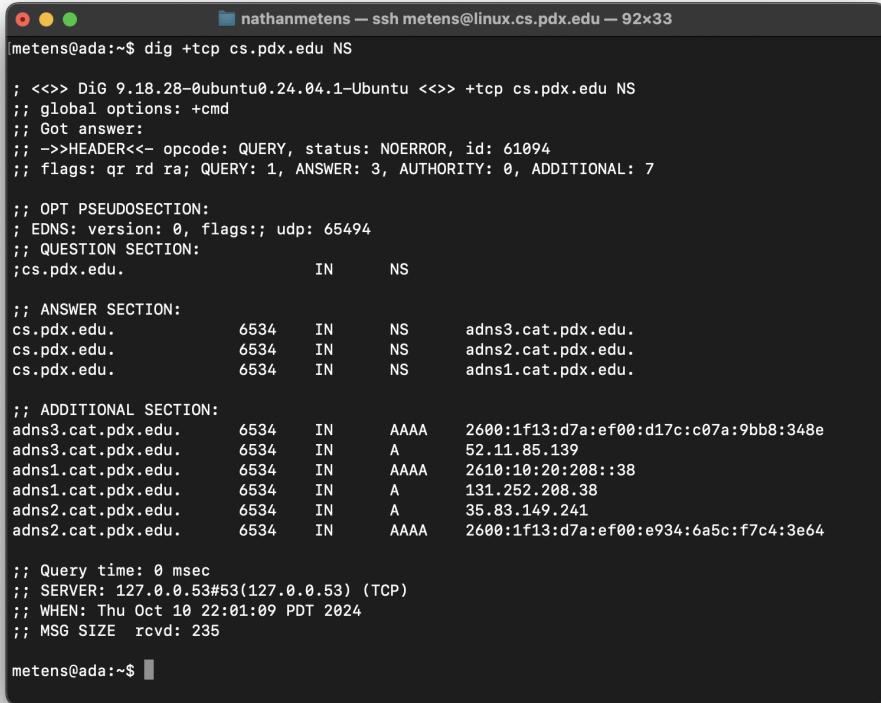
It appears that Amazon is the cloud provider for [www.pdx.edu](http://www.pdx.edu).

Now, using the following URL: **aspmx.l.google.com** from the `dig +tcp @131.252.208.53 pdx.edu MX` command in the previous screenshot we get this from the iplocations.net website:

Geolocation data from	DB-IP	Product: API, real-time
 DOMAIN NAME: aspmx.l.google.com	 ISP: Google LLC	
 COUNTRY: United States 	 ORGANIZATION: Google LLC	
 REGION: California	 LATITUDE: 37.4225	
 CITY: Mountain View	 LONGITUDE: -122.085	
Incorrect location? Contact DB-IP		 <a href="#">view map</a>

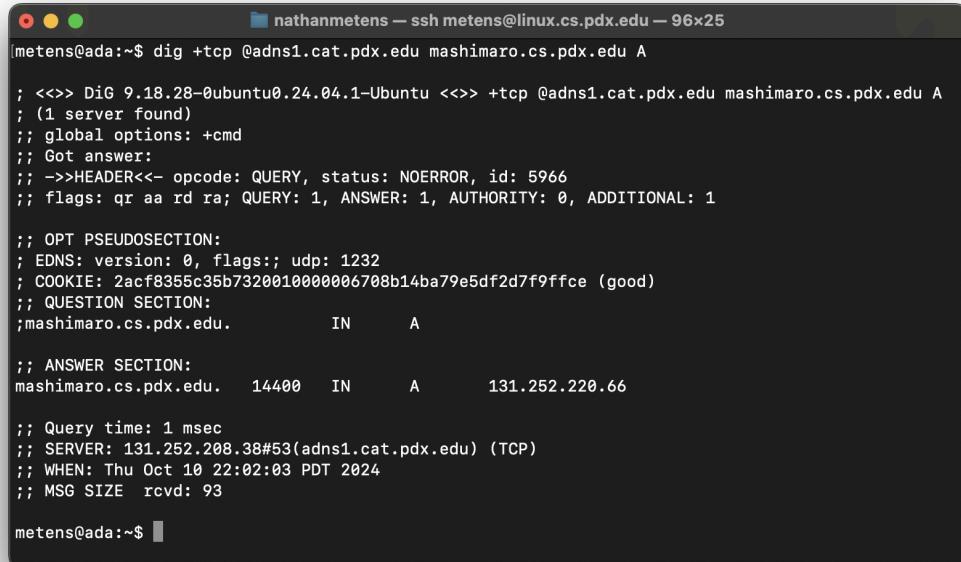
We see that the Google cloud provider handles email for pdx.edu.

Using the dig command to find the authoritative server for mashimaro:



```
nathanmetens — ssh metens@linux.cs.pdx.edu — 92x33
[metens@ada:~$ dig +tcp cs.pdx.edu NS
; <>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <>> +tcp cs.pdx.edu NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61094
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 7
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;cs.pdx.edu.           IN      NS
;;
;; ANSWER SECTION:
cs.pdx.edu.        6534    IN      NS      adns3.cat.pdx.edu.
cs.pdx.edu.        6534    IN      NS      adns2.cat.pdx.edu.
cs.pdx.edu.        6534    IN      NS      adns1.cat.pdx.edu.
;;
;; ADDITIONAL SECTION:
adns3.cat.pdx.edu. 6534    IN      AAAA   2600:1f13:d7a:ef00:d17c:c07a:9bb8:348e
adns3.cat.pdx.edu. 6534    IN      A       52.11.85.139
adns1.cat.pdx.edu. 6534    IN      AAAA   2610:10:20:208::38
adns1.cat.pdx.edu. 6534    IN      A       131.252.208.38
adns2.cat.pdx.edu. 6534    IN      A       35.83.149.241
adns2.cat.pdx.edu. 6534    IN      AAAA   2600:1f13:d7a:ef00:e934:6a5c:f7c4:3e64
;;
;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (TCP)
;; WHEN: Thu Oct 10 22:01:09 PDT 2024
;; MSG SIZE rcvd: 235
metens@ada:~$ ]
```

And then querying that server for the A record of mashimaro.cs.pdx.edu:



```
nathanmetens — ssh metens@linux.cs.pdx.edu — 96x25
[metens@ada:~$ dig +tcp @adns1.cat.pdx.edu mashimaro.cs.pdx.edu A
; <>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <>> +tcp @adns1.cat.pdx.edu mashimaro.cs.pdx.edu A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5966
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;;
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 2acf8355c35b7320010000006708b14ba79e5df2d7f9ffce (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.           IN      A
;;
;; ANSWER SECTION:
mashimaro.cs.pdx.edu. 14400    IN      A       131.252.220.66
;;
;; Query time: 1 msec
;; SERVER: 131.252.208.38#53(adns1.cat.pdx.edu) (TCP)
;; WHEN: Thu Oct 10 22:02:03 PDT 2024
;; MSG SIZE rcvd: 93
metens@ada:~$ ]
```

## 2. Iterative DNS lookups

Using `dig` without any arguments, I found the IPv4 F root server:

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 84x23

;; ADDITIONAL SECTION:
c.root-servers.net.    7087   IN      A      192.33.4.12
m.root-servers.net.    7087   IN      A      202.12.27.33
l.root-servers.net.    7087   IN      A      199.7.83.42
d.root-servers.net.    7087   IN      A      199.7.91.13
d.root-servers.net.    7087   IN      AAAA   2001:500:2d::d
a.root-servers.net.    7087   IN      A      198.41.0.4
f.root-servers.net.    7087   IN      A      192.5.5.241
i.root-servers.net.    7087   IN      AAAA   2001:7fe::53
a.root-servers.net.    7087   IN      AAAA   2001:503:ba3e::2:30
b.root-servers.net.    7087   IN      A      170.247.170.2
f.root-servers.net.    7087   IN      AAAA   2001:500:2f::f
j.root-servers.net.    7087   IN      AAAA   2001:503:c27::2:30
k.root-servers.net.    7087   IN      A      193.0.14.129
h.root-servers.net.    7087   IN      A      198.97.190.53
c.root-servers.net.    7087   IN      AAAA   2001:500:2::c
g.root-servers.net.    7087   IN      AAAA   2001:500:12::d0d
e.root-servers.net.    7087   IN      A      192.203.230.10
m.root-servers.net.    7087   IN      AAAA   2001:dc3::35
k.root-servers.net.    7087   IN      AAAA   2001:7fd::1
h.root-servers.net.    7087   IN      AAAA   2001:500:1::53
j.root-servers.net.    7087   IN      A      192.58.128.30
```

Now using the `+tcp` and `+norecurse` and `-4` flags with the `f.root-servers.net` IP address to query for the com NS(`dig +tcp +norecurse -4 @192.5.5.241 com NS`), we find this:

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 84x23

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65535
;; QUESTION SECTION:
;com.                      IN      NS

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

;; ADDITIONAL SECTION:
e.gtld-servers.net.       172800  IN      A      192.12.94.30
e.gtld-servers.net.       172800  IN      AAAA   2001:502:1ca1::30
```

Next, we use this to find the next query(dig +tcp +norecurse -4 @f.gtld-servers.net google.com NS) for the google.com:

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 84x23
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.           IN      NS

;; AUTHORITY SECTION:
google.com.        172800  IN      NS      ns2.google.com.
google.com.        172800  IN      NS      ns1.google.com.
google.com.        172800  IN      NS      ns3.google.com.
google.com.        172800  IN      NS      ns4.google.com.

;; ADDITIONAL SECTION:
ns2.google.com.    172800  IN      AAAA   2001:4860:4802:34::a
ns2.google.com.    172800  IN      A       216.239.34.10
ns1.google.com.    172800  IN      AAAA   2001:4860:4802:32::a
ns1.google.com.    172800  IN      A       216.239.32.10
ns3.google.com.    172800  IN      AAAA   2001:4860:4802:36::a
ns3.google.com.    172800  IN      A       216.239.36.10
ns4.google.com.    172800  IN      AAAA   2001:4860:4802:38::a
ns4.google.com.    172800  IN      A       216.239.38.10

;; Query time: 23 msec
;; SERVER: 192.35.51.30#53(f.gtld-servers.net) (TCP)
```

Now we perform the last query: dig +tcp +norecurse -4 @ns1.google.com console.cloud.google.com A

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 112x26
;; MSG SIZE  rcvd: 90
metens@ada:~$ dig +tcp +norecurse -4 @ns1.google.com console.cloud.google.com A
; <>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <>> +tcp +norecurse -4 @ns1.google.com console.cloud.google.com A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39568
[;; flags: qr aa; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1]

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

;; ANSWER SECTION:
console.cloud.google.com. 300    IN      CNAME   www3.l.google.com.
www3.l.google.com.         300    IN      A       142.251.33.110

;; Query time: 29 msec
;; SERVER: 216.239.32.10#53(ns1.google.com) (TCP)
;; WHEN: Thu Oct 10 22:51:13 PDT 2024
;; MSG SIZE  rcvd: 90
metens@ada:~$
```

### 3. Reverse DNS lookups

Using ChatGPT, produce a single command line with commands dig, egrep, and awk, to list all IPv4 addresses that espn.go.com points to

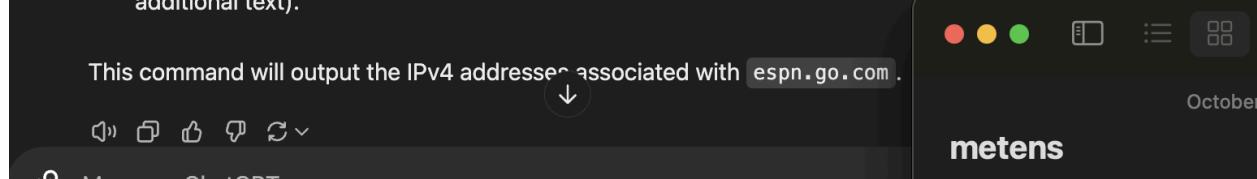
You can use a combination of `dig`, `egrep`, and `awk` in a single command to list all IPv4 addresses that `espn.go.com` points to. Here's the command:

```
bash Copy code
dig +short espn.go.com A | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print $1}'
```

**Breakdown:**

- `dig +short espn.go.com A` : Queries the A record (IPv4 addresses) for `espn.go.com` and returns the output in short format (only the IPs or CNAME).
- `egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$'` : Filters the output to show only valid IPv4 addresses, matching the pattern of an IP address.
- `awk '{print $1}'` : Prints the first field, which is the IP address (this is useful to remove any additional text).

This command will output the IPv4 addresses associated with `espn.go.com`.



```
nathanmetens — ssh metens@linux.cs.pdx.edu — 102x8
[metens@ada:~$ ls
metens@ada:~$ dig +short espn.go.com A | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print $1}'
18.161.6.38
18.161.6.94
18.161.6.89
18.161.6.80
metens@ada:~$ ]
```

Using `man dig` I found the flag `-x addr` for reverse lookups where `addr` is the IP address:

```
as described in RFC 3597.  
-u This option indicates that print query times should be provided in microseconds instead of milliseconds.  
-v This option prints the version number and exits.  
-x addr  
This option sets simplified reverse lookups, for mapping addresses to names. The addr is an IPv4 address in dotted-decimal notation, or a colon-delimited IPv6 address. When the _x option is used, there is no need to provide the name, class, and type arguments. dig automatically performs a lookup for a name like 94.2.0.192.in-addr.arpa and sets the query type and class to PTR and IN respectively. IPv6 addresses are looked up using nibble format under the IP6.ARPA domain.  
-y [hmac:]keyname:secret  
This option signs queries using TSIG with the given authentication key. keyname is the name of the key, and secret is the base64-encoded shared secret. hmac is the name of the key algorithm; valid choices are hmac-md5, hmac-sha1, hmac-sha224, hmac-sha256, hmac-sha384, or hmac-sha512. If hmac is not specified, the default is hmac-md5; if MD5 was disabled, the default is hmac-sha256.  
Manual page dig(1) line 119 (press h for help or q to quit)
```

Here is the reverse lookup for each of the IPv4 addresses:

```
[metens@ada:~$ cat rever-lookup.sh  
#!/bin/bash  
  
espn=$(dig +short espn.go.com A | egrep '^+[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print $1}')  
  
for i in $espn  
do  
    echo "Reverse Lookup for IPv4 $i:"  
    dig -x $i | egrep '([a-zA-Z0-9-]+\.)+[a-zA-Z]{2,}' | awk '{print $5}'  
done  
[metens@ada:~$ chmod +x rever-lookup.sh  
[metens@ada:~$ ./rever-lookup.sh  
Reverse Lookup for IPv4 18.161.6.94:  
  
server-18-161-6-94.hio52.r.cloudfront.net.  
Reverse Lookup for IPv4 18.161.6.89:  
  
server-18-161-6-89.hio52.r.cloudfront.net.  
Reverse Lookup for IPv4 18.161.6.80:  
  
server-18-161-6-80.hio52.r.cloudfront.net.  
Reverse Lookup for IPv4 18.161.6.38:  
  
server-18-161-6-38.hio52.r.cloudfront.net.  
metens@ada:~$ ]
```

## 4. Host enumeration

```
nathanmetens — ssh metens@linux.cs.pdx.edu — 101x15
[metens@ada:~$ nmap 131.252.220.0/24 | egrep 'Nmap scan report for' | awk '{print $5}' > 220hosts.txt ]
[metens@ada:~$ cat 220hosts.txt
mashimaro.cs.pdx.edu
pochacco.cs.pdx.edu
saises.cs.pdx.edu
ventoux.cs.pdx.edu
bmw.cs.pdx.edu
ferrari.cs.pdx.edu
131.252.220.251
sizzle.cs.pdx.edu
[metens@ada:~$ cat 220hosts.txt | head -6 | tail -2
bmw.cs.pdx.edu
ferrari.cs.pdx.edu
metens@ada:~$ ]
```

## 5. Geographic DNS #5

For the 131.252.208.53 IPv4 address, the location that ipinfo.io and DB-IP give is Portland, OR (PSU). For 198.82.247.66 IPv4, ipinfo.io and DB-IP give the location of Blacksburg, Virginia.

Using `dig @131.252.208.53 www.google.com | egrep 'IN' | awk '{print $5}'` to display all the www.google.com IP addresses, I chose 142.250.217.100 for the google representation for PSU, and 142.251.167.147 for the google representation for Virginia Polytechnic Institute and State Univ:

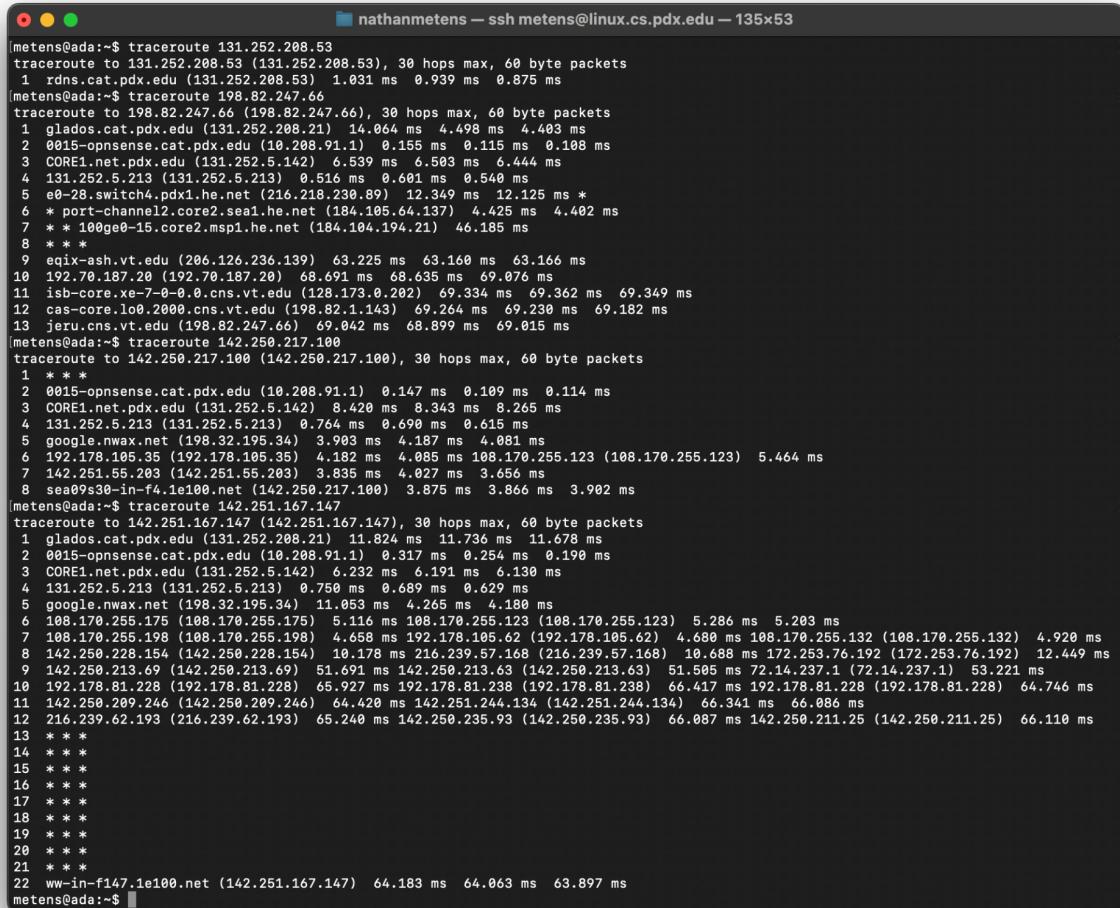
```
nathanmetens — ssh metens@linux.cs.pdx.edu — 101x22
www.google.com.    240      IN      A      142.251.167.99
www.google.com.    240      IN      A      142.251.167.103
www.google.com.    240      IN      A      142.251.167.104
www.google.com.    240      IN      A      142.251.167.105

;; Query time: 69 msec
;; SERVER: 198.82.247.66#53(198.82.247.66) (UDP)
;; WHEN: Sat Oct 12 23:08:47 PDT 2024
;; MSG SIZE  rcvd: 167

[metens@ada:~$ dig @131.252.208.53 www.google.com | egrep 'IN' | awk '{print $5}' ]
142.250.217.100
[metens@ada:~$ dig @198.82.247.66 www.google.com | egrep 'IN' | awk '{print $5}' ]
142.251.167.99
142.251.167.104
142.251.167.106
142.251.167.103
142.251.167.147
142.251.167.105
metens@ada:~$ ]
```

Using <https://www.iplocation.net/ip-lookup> to find the location of the two Google IP addresses, 142.250.217.100 is located in Seattle, Washington, and 142.251.167.147 is located in Reston, Virginia.

Performing a traceroute to all 4 IP addresses:



```
nathanmetens - ssh metens@linux.cs.pdx.edu - 135x53
[metens@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)  1.031 ms  0.939 ms  0.875 ms
[metens@ada:~$ traceroute 198.82.247.66
traceroute to 198.82.247.66 (198.82.247.66), 30 hops max, 60 byte packets
1  glados.cat.pdx.edu (131.252.208.21)  14.064 ms  4.498 ms  4.403 ms
2  0015-opsense.cat.pdx.edu (18.208.91.1)  0.155 ms  0.115 ms  0.108 ms
3  CORE1.net.pdx.edu (131.252.5.142)  6.539 ms  6.583 ms  6.444 ms
4  131.252.5.213 (131.252.5.213)  0.516 ms  0.601 ms  0.540 ms
5  e0-28.switch4.pdx1.he.net (216.218.230.89)  12.349 ms  12.125 ms *
6  * port-channel2.core2.sea1.he.net (184.185.64.137)  4.425 ms  4.402 ms
7  * * * 100ge0-15.core2.msp1.he.net (184.104.194.21)  46.185 ms
8  * * *
9  eqix-ash.vt.edu (206.126.236.139)  63.225 ms  63.160 ms  63.166 ms
10 192.70.187.20 (192.70.187.20)  68.691 ms  68.635 ms  69.076 ms
11  isb-core.xe-7-0-0.cns.vt.edu (128.173.0.202)  69.334 ms  69.362 ms  69.349 ms
12  cas-core.lo0.2000.cns.vt.edu (198.82.1.143)  69.264 ms  69.230 ms  69.182 ms
13  jeru.cns.vt.edu (198.82.247.66)  69.042 ms  68.899 ms  69.015 ms
[metens@ada:~$ traceroute 142.250.217.100
traceroute to 142.250.217.100 (142.250.217.100), 30 hops max, 60 byte packets
1  * * *
2  0015-opsense.cat.pdx.edu (18.208.91.1)  0.147 ms  0.109 ms  0.114 ms
3  CORE1.net.pdx.edu (131.252.5.142)  8.420 ms  8.343 ms  8.265 ms
4  131.252.5.213 (131.252.5.213)  0.764 ms  0.690 ms  0.615 ms
5  google.nwax.net (198.32.195.34)  3.983 ms  4.187 ms  4.081 ms
6  192.178.185.35 (192.178.185.35)  4.182 ms  4.085 ms  108.170.255.123 (108.170.255.123)  5.464 ms
7  142.251.55.203 (142.251.55.203)  3.835 ms  4.027 ms  3.656 ms
8  sea09s30-in-f4.1e100.net (142.250.217.100)  3.875 ms  3.866 ms  3.902 ms
[metens@ada:~$ traceroute 142.251.167.147
traceroute to 142.251.167.147 (142.251.167.147), 30 hops max, 60 byte packets
1  glados.cat.pdx.edu (131.252.208.21)  11.824 ms  11.736 ms  11.678 ms
2  0015-opsense.cat.pdx.edu (18.208.91.1)  0.317 ms  0.254 ms  0.190 ms
3  CORE1.net.pdx.edu (131.252.5.142)  6.232 ms  6.191 ms  6.130 ms
4  131.252.5.213 (131.252.5.213)  0.750 ms  0.689 ms  0.629 ms
5  google.nwax.net (198.32.195.34)  11.053 ms  4.265 ms  4.180 ms
6  108.170.255.175 (108.170.255.175)  5.116 ms  108.170.255.123 (108.170.255.123)  5.286 ms  5.203 ms
7  108.170.255.198 (108.170.255.198)  4.658 ms  192.178.105.62 (192.178.185.62)  4.688 ms  108.170.255.132 (108.170.255.132)  4.920 ms
8  142.250.228.154 (142.250.228.154)  10.178 ms  216.239.57.168 (216.239.57.168)  10.688 ms  172.253.76.192 (172.253.76.192)  12.449 ms
9  142.250.213.69 (142.250.213.69)  51.691 ms  142.250.213.63 (142.250.213.63)  51.505 ms  72.14.237.1 (72.14.237.1)  53.221 ms
10 192.178.81.228 (192.178.81.228)  65.927 ms  192.178.81.238 (192.178.81.238)  66.417 ms  192.178.81.228 (192.178.81.228)  64.746 ms
11 142.250.289.246 (142.250.289.246)  64.420 ms  142.251.244.134 (142.251.244.134)  66.341 ms  66.086 ms
12 216.239.62.193 (216.239.62.193)  65.240 ms  142.250.235.93 (142.250.235.93)  66.087 ms  142.250.211.25 (142.250.211.25)  66.110 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  ww-in-f147.1e100.net (142.251.167.147)  64.183 ms  64.063 ms  63.897 ms
metens@ada:~$ ]
```

## 6/7. Wireshark Lab #3, -

E\_DIR not set, defaulting to '/tmp/runtime-root'  
nl80211 not found.  
\*\* (wireshark:2203) 01:26:22.725455 [Capture MESSAGE] -- Capture Start ...

Capturing from ens4 (icmp)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000000	10.138.0.3	74.125.142.104	ICMP	98 Echo (ping) request id=0x5fb8, seq=1/256, ttl=64 (reply in 2)
2	0.002233750	74.125.142.104	10.138.0.3	ICMP	98 Echo (ping) reply id=0x5fb8, seq=1/256, ttl=118 (request in 1)
3	0.947112097	52.81.237.148	10.138.0.3	ICMP	54 Echo (ping) request id=0x277b, seq=0/0, ttl=239 (reply in 4)
4	0.947152736	10.138.0.3	52.81.237.148	ICMP	54 Echo (ping) reply id=0x277b, seq=0/0, ttl=64 (request in 3)
5	1.001931375	10.138.0.3	74.125.142.104	ICMP	98 Echo (ping) request id=0x5fb8, seq=2/512, ttl=64 (reply in 6)
6	1.002206026	74.125.142.104	10.138.0.3	ICMP	98 Echo (ping) reply id=0x5fb8, seq=2/512, ttl=118 (request in 5)
7	2.033724776	10.138.0.3	74.125.142.104	ICMP	98 Echo (ping) request id=0x5fb8, seq=3/768, ttl=64 (reply in 8)
8	2.034140371	74.125.142.104	10.138.0.3	ICMP	98 Echo (ping) reply id=0x5fb8, seq=3/768, ttl=118 (request in 7)

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ens4  
Ethernet II, Src: 42:01:0a:8a:00:03 (42:01:0a:8a:00:03), Dst: 42:01:0a:8a:00:03  
Destination: 42:01:0a:8a:00:01 (42:01:0a:8a:00:01)  
Source: 42:01:0a:8a:00:03 (42:01:0a:8a:00:03)  
Type: IPv4 (0x0800)  
Internet Protocol Version 4, Src: 10.138.0.3, Dst: 74.125.142.104  
Internet Control Message Protocol

Destination Hardware Address (eth.dst), 6 byte(s)

Packets: 8 · Displayed: 8 (100.0%) · Profile: Default

```

metens@course-vml:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc mq state UP group default qlen 1000
    link/ether 42:01:0a:8a:00:03 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 10.138.0.3/32 metric 100 scope global dynamic ens4
        valid_lft 86176sec preferred_lft 86176sec
        inet6 fe80::4001:aff:fe8a:3/64 scope link
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:0f:03:0f:56 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
metens@course-vml:~$ ip route show
default via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.3 metric 100
10.138.0.1 dev ens4 proto dhcp scope link src 10.138.0.3 metric 100
169.254.169.254 via 10.138.0.1 dev ens4 proto dhcp src 10.138.0.3 metric 100
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
metens@course-vml:~$ ping -c 3 www.google.com
PING www.google.com (74.125.142.106) 56(84) bytes of data.
64 bytes from ie-in-f106.1e100.net (74.125.142.106): icmp_seq=1 ttl=118 time=1.90 ms
64 bytes from pv-in-f106.1e100.net (74.125.142.106): icmp_seq=2 ttl=118 time=0.410 ms
64 bytes from pv-in-f106.1e100.net (74.125.142.106): icmp_seq=3 ttl=118 time=0.385 ms

--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.385/0.899/1.903/0.709 ms
metens@course-vml:~$ ping -c 3 www.google.com
PING www.google.com (74.125.142.104) 56(84) bytes of data.
64 bytes from ie-in-f104.1e100.net (74.125.142.104): icmp_seq=1 ttl=118 time=2.26 ms
64 bytes from pv-in-f104.1e100.net (74.125.142.104): icmp_seq=2 ttl=118 time=0.306 ms
64 bytes from ie-in-f104.1e100.net (74.125.142.104): icmp_seq=3 ttl=118 time=0.453 ms

--- www.google.com ping statistics ---

```

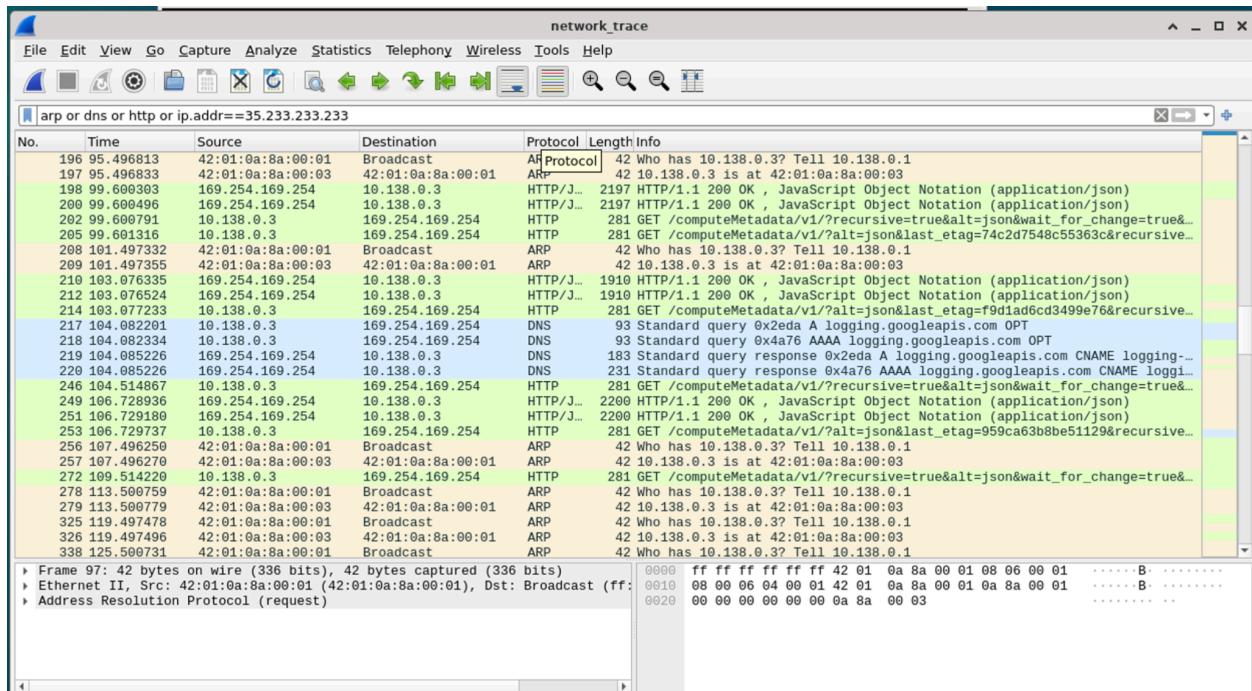
The VM IP address: 10.138.0.3/32.

The name of the local virtual ethernet interface is ens4.

The IP address of the default router is 10.138.0.1.

As seen on the packet dump in Wireshark, the destination MAC address corresponds to the MAC address of Google. For the reply packet, the destination MAC address corresponds to the MAC address of the VM. For the request packet, the internet IP address of the src is the VM IP address, and the destination IP address is Google's internet IP address as seen in the ping snapshot.

## 10. Analyze trace



### ARP

Packets No. 197, 209, and 257 are attempting to get the hardware address of the default router. The hardware address is 42:01:0a:8a:00:01.

### DNS

Packets No. 217 and 218 are requests for the website. The local DNS server being queried is 169.254.169.254.

### TCP

After including TCP in the filter, packets No. 1 and 2 are the initial handshake with the website. It took 0.000892 seconds to complete the handshake.

### HTTP

Packets No. 198, 200, 202, and 205 correspond to the HTTP request and response. It took about 99 seconds to process the HTTP request after the initial handshake in this snapshot.