


Notebook 2

02.1

1. SS command:

SA

Please find a single linux 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.



You can use the `netstat` command to list all TCP sockets in a LISTEN state on an IPv4 address and display the associated program. However, `netstat` is deprecated, and it's recommended to use `ss` (Socket Statistics) instead. Here's a single `ss` command that achieves this:

```
bash
ss -tln4
```

Copy code

Here's what each flag does:

- `-t`: Show TCP sockets.
- `-l`: Show only listening sockets.
- `-n`: Show numerical addresses (IP addresses and port numbers) instead of resolving hostnames.
- `-4`: Display only IPv4 sockets.

This command will provide a list of all TCP sockets in a LISTEN state on IPv4 addresses along with the associated program or service using them.

KHOD2

Output:

```
khod2@course-vm:~$ sudo ss -tln4
State  Recv-Q  Send-Q  Local Address:Port  Peer Address:Port  Process
LISTEN  0        4096    127.0.0.53%lo:53      0.0.0.0:*
LISTEN  0        4096      127.0.0.1:42789      0.0.0.0:*
LISTEN  0         128      0.0.0.0:22           0.0.0.0:*
khod2@course-vm:~$
```

This didn't get me what I wanted and I didn't want to screenshot every prompt I could find, so I changed the command to `ss -tlp4` to get the process names.

```
khod2@course-vm:~$ sudo ss -tlp4
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
LISTEN     0            4096        127.0.0.53%lo:domain     0.0.0.0:*               users: (("systemd-resolve",pid=356,fd=14))
LISTEN     0            4096        127.0.0.1:42789          0.0.0.0:*               users: (("containerd",pid=3042,fd=10))
LISTEN     0            128         0.0.0.0:ssh              0.0.0.0:*               users: (("sshd",pid=766,fd=3))
khod2@course-vm:~$
```

The service that can be contacted from any interface on the machine is the “ssh” service, with 0.0.0.0:ssh. The other two services only listen on the loopback interface and can only be contacted by local processes, “systemd-resolve” and “containerd”.

Output of command in linux.cs.pdx.edu:

```
khod2@ada:~$ ss -tlp4
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
LISTEN     0            511        127.0.0.1:45549          0.0.0.0:*
LISTEN     0            5          127.0.0.1:5902           0.0.0.0:*
LISTEN     0            511        127.0.0.1:39503          0.0.0.0:*
LISTEN     0            511        127.0.0.1:39507          0.0.0.0:*
LISTEN     0            128        127.0.0.1:6100           0.0.0.0:*
LISTEN     0            4096       127.0.0.53%lo:domain     0.0.0.0:*
LISTEN     0            128         0.0.0.0:ssh              0.0.0.0:*
LISTEN     0            128        127.0.0.1:ipp            0.0.0.0:*
LISTEN     0            100        127.0.0.1:smtp           0.0.0.0:*
LISTEN     0            5          127.0.0.1:5984           0.0.0.0:*
LISTEN     0            5          127.0.0.1:5953           0.0.0.0:*
LISTEN     0            128        127.0.0.1:1222           0.0.0.0:*
khod2@ada:~$
```

This machine provides only the ssh service (0.0.0.0) for external access.

2. ChatGPT lsof command to list all TCP sockets listening:

SA I don't want to target a specific address, I want to list all tcp sockets with their ipv4 addresses that are listening

I apologize for the misunderstanding. To list all TCP sockets in a LISTEN state along with their associated IPv4 addresses, you can use the following **lsof** command:

```
bash
lsof -iTCP -sTCP:LISTEN -n -P
```

Here's what each flag does:

KHOD2

- * **-iTCP**: Specifies to list TCP sockets.
- * **-sTCP:LISTEN**: Filters the results to show only sockets in the LISTEN state.
- * **-n**: Displays numerical addresses (IP addresses and port numbers) instead of resolving hostnames.
- * **-P**: Prevents **lsof** from resolving port names.

This command will provide a list of all TCP sockets in a LISTEN state, along with their associated IPv4 addresses, without targeting a specific address.

Output:

```
khod2@course-vm:~$ sudo lsof -iTCP -sTCP:LISTEN -n -P
COMMAND  PID    USER      FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
systemd-r 356  systemd-resolve 14u  IPv4  16753      0t0  TCP  127.0.0.53:53 (LISTEN)
sshd      766    root       3u   IPv4  19057      0t0  TCP  *:22 (LISTEN)
sshd      766    root       4u   IPv6  19060      0t0  TCP  *:22 (LISTEN)
container 3042   root      10u  IPv4  25381      0t0  TCP  127.0.0.1:42789 (LISTEN)
xrdp-sesm 8887   root       7u   IPv6  37104      0t0  TCP  [::1]:3350 (LISTEN)
xrdp      8901   xrdp      11u  IPv6  36471      0t0  TCP  *:3389 (LISTEN)
khod2@course-vm:~$
```

4. Available bandwidth from us-west1-b VM:

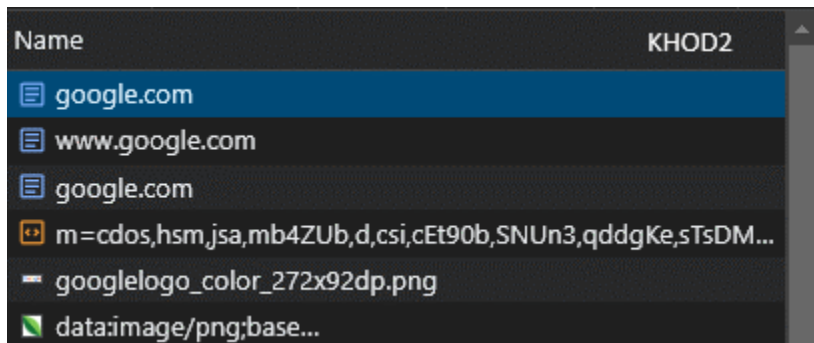
```

khod2@vm-us-west1-b:~$ iperf -c 10.152.0.4 -p 80
-----
Client connecting to 10.152.0.4, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[  1] local 10.138.0.7 port 55768 connected with 10.152.0.4 port 80
[ ID] Interval           Transfer     Bandwidth
[  1] 0.0000-10.2025 sec   173 MBytes  143 Mbits/sec
khod2@vm-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 10.138.0.7 port 57344 connected with 10.132.0.2 port 80
[ ID] Interval           Transfer     Bandwidth
[  1] 0.0000-10.2251 sec   191 MBytes  157 Mbits/sec
khod2@vm-us-west1-b:~$ iperf -c 10.142.0.3 -p 80
-----
Client connecting to 10.142.0.3, TCP port 80
TCP window size: 85.0 KByte (default)
-----
[  1] local 10.138.0.7 port 44046 connected with 10.142.0.3 port 80
[ ID] Interval           Transfer     Bandwidth
[  1] 0.0000-10.0841 sec   425 MBytes  354 Mbits/sec
khod2@vm-us-west1-b:~$ █

```

It makes sense that the bandwidth is somewhat similar for the first two servers (Australia-southeast and Europe-west) since those servers are relatively far from our us-west server, while the us-east server is obviously much closer. The bandwidth decreases with long distances.

5. Initial 3 requests for google:



Google.com:

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

The status code is 307, indicating an internal redirect.

Name	KHOD2	Status	Protocol
google.com		307	http/1.1

Google.com

The URL being requested is <https://google.com/>.

The status code is 301, indicating a permanent redirection.

google.com	KHOD2	307	http/1.1
google.com		301	h2

www.google.com

The URL being requested is <https://www.google.com/>.

google.com	KHOD2	301	h2
www.google.com		200	h3

The first redirection sends the browser to <https://google.com/>.

The second redirection finally sends the browser to <https://www.google.com/>.

Cookie:

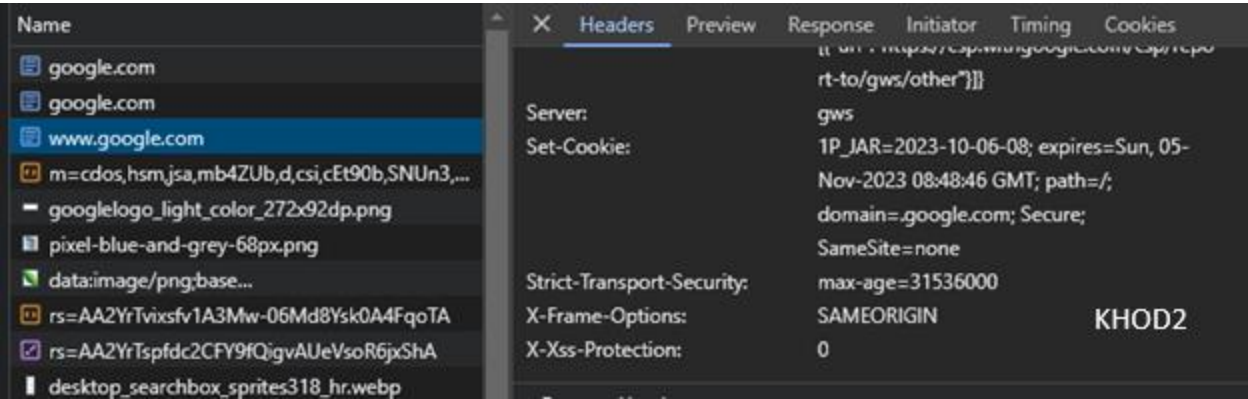
Name	Headers	Preview	Response	Initiator	Timing	Cookies
google.com			rt-to/gws/other)))			
google.com			Server: gws			
www.google.com			X-Frame-Options: SAMEORIGIN			KHOD2
m=cDos,hsM,jSA,mb4ZUb,d,csi,cEt90b,SNU3,...			X-Xss-Protection: 0			
googlelogo_light_color_272x92dp.png						
pixel-blue-and-grey-68px.png						
data:image/png;base...						
rs=AA2YrTviXsfv1A3Mw-06Md8Ysk0A4FqoTA						
rs=AA2YrTspfdC2CFY9fQigvAUeVsoR6jxShA						
desktop_searchbox_sprites318_hr.webp						
gen_204?s=webhp&at=aft&atyp=csi&ei=7skfZ...						
m=lvPZ6d?xjs=s1						
rs=ACT90oGF5A4NiQ1wBeY_lqW1gy8AtQ9Yvw						
search?q&cp=0&client=gws-wiz&xssi=t&gs_...						
client_204?atyp=i&biw=806&bih=1144&dpr=...						
callout?prid=19037050&pgid=19037049&pui...						
cb=gapi.loaded_0						
m=sy6q,syde,sydz,sym5,MkHyGd?xjs=s3						
m=ws9Tlc?xjs=s3						
m=x8cHvb?xjs=s3						
m=sy6r,sys9,x4FYXe?xjs=s3						
m=syef,syeg,kHVSUb?xjs=s3						

Headers	Preview	Response	Initiator	Timing	Cookies
X-Frame-Options: SAMEORIGIN					
X-Xss-Protection: 0					KHOD2
Request Headers					
authority:	google.com				
method:	GET				
path:	/				
scheme:	https				
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7				
Accept-Encoding:	gzip, deflate, br				
Accept-Language:	en-US,en;q=0.9				
Cookie:	1P_JAR=2023-10-06-08; AEC=Ackid1RFi59xecy8SGYkxtktu3wslA1nc8IN7ebPXquAuFFQmZPDFWqXwcQ; NID=511=OaarPz1u5aYU5001r2Cb4w_VyD9rc-E0b78oXpabmejekMFWz7hfahE9SNOUO2bF284nF4yudbq_P5Eh2gU-H7RJVOVHc-oRW2RR33ugBLb276F8KKcQaybPqjcbijc4M5DKI_DySoJyH300n_jSv80rksrW85o8tavXbnyFHf4cvD_4toy18A				

Name	Headers	Preview	Response	Initiator	Timing	Cookies
google.com						
google.com						
www.google.com						
m=cDos,hsM,jSA,mb4ZUb,d,csi,cEt90b,SNU3,...						
googlelogo_light_color_272x92dp.png						
pixel-blue-and-grey-68px.png						
data:image/png;base...						
rs=AA2YrTviXsfv1A3Mw-06Md8Ysk0A4FqoTA						
rs=AA2YrTspfdC2CFY9fQigvAUeVsoR6jxShA						
desktop_searchbox_sprites318_hr.webp						
gen_204?s=webhp&at=aft&atyp=csi&ei=7skfZ...						
m=lvPZ6d?xjs=s1						
rs=ACT90oGF5A4NiQ1wBeY_lqW1gy8AtQ9Yvw						
search?q&cp=0&client=gws-wiz&xssi=t&gs_...						
client_204?atyp=i&biw=806&bih=1144&dpr=...						
callout?prid=19037050&pgid=19037049&pui...						
cb=gapi.loaded_0						
m=sy6q,syde,sydz,sym5,MkHyGd?xjs=s3						
m=ws9Tlc?xjs=s3						
m=x8cHvb?xjs=s3						
m=sy6r,sys9,x4FYXe?xjs=s3						
m=syef,syeg,kHVSUb?xjs=s3						

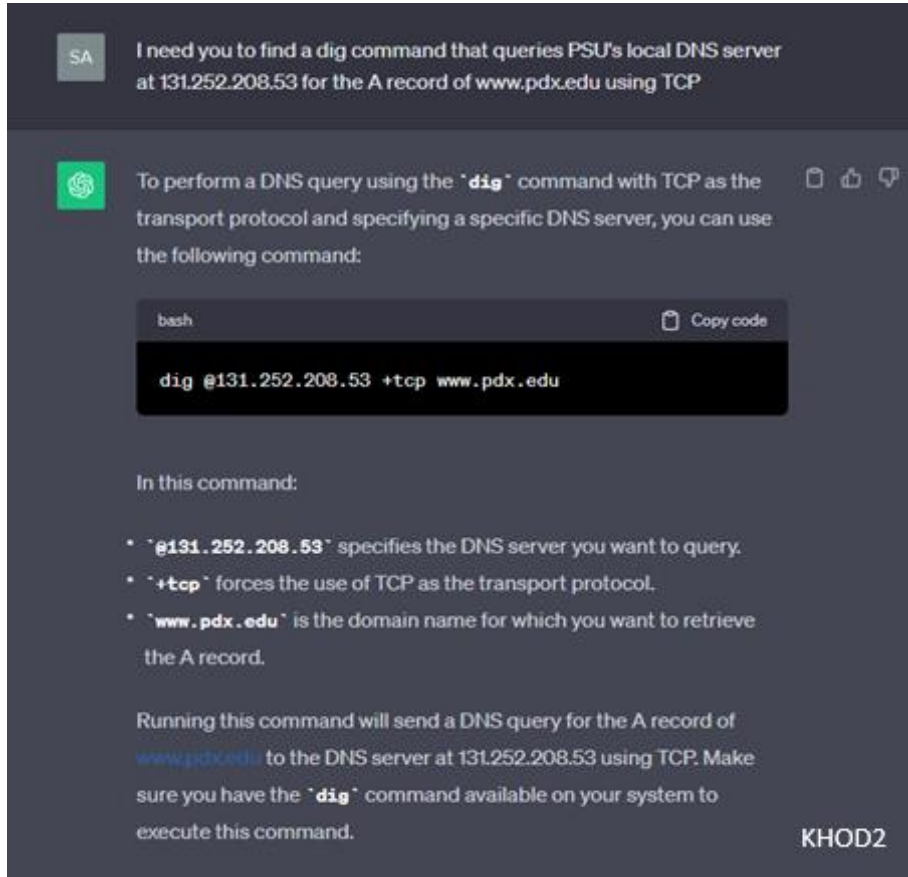
Headers	Preview	Response	Initiator	Timing	Cookies
X-Frame-Options: SAMEORIGIN					
X-Xss-Protection: 0					KHOD2
Request Headers					
authority:	www.google.com				
method:	GET				
path:	/				
scheme:	https				
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7				
Accept-Encoding:	gzip, deflate, br				
Accept-Language:	en-US,en;q=0.9				
Cookie:	1P_JAR=2023-10-06-08; AEC=Ackid1RFi59xecy8SGYkxtktu3wslA1nc8IN7ebPXquAuFFQmZPDFWqXwcQ; DV=Ux2sZ-mLUJ4aEM4dSZxmxyqurRCs8g; NID=511=OaarPz1u5aYU5001r2Cb4w_VyD9rc-E0b78oXpabmejekMFWz7hfahE9SNOUO2bF284nF4yudbq_P5Eh2gU-H7RJVOVHc-oRW2RR33ugBLb276F8KKcQaybPqjcbijc4M5DKI_DySoJyH300n_jSv80rksrW85o8tavXbnyFHf4cvD_4toy18A				

Set-cookie:



02.2

1. Dig command given by ChatGPT:



SA I need you to find 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 perform a DNS query using the **"dig"** command with TCP as the transport protocol and specifying a specific DNS server, you can use the following command:

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

In this command:

- **"@131.252.208.53"** specifies the DNS server you want to query.
- **"+tcp"** forces the use of TCP as the transport protocol.
- **"www.pdx.edu"** is the domain name for which you want to retrieve the A record.

Running this command will send a DNS query for the A record of www.pdx.edu to the DNS server at 131.252.208.53 using TCP. Make sure you have the **"dig"** command available on your system to execute this command.

KHOD2

A record returned:

```
khod2@ada:~$ dig @131.252.208.53 +tcp www.pdx.edu

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> @131.252.208.53 +tcp www.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63860
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 1a5ece274d3ae9010000000652242cf52813b92ac826b4b (good)
;; QUESTION SECTION:
;www.pdx.edu.                IN      A

;; ANSWER SECTION:
www.pdx.edu.                 60      IN      A      108.138.94.58
www.pdx.edu.                 60      IN      A      108.138.94.13
www.pdx.edu.                 60      IN      A      108.138.94.27
www.pdx.edu.                 60      IN      A      108.138.94.85

;; Query time: 59 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Sat Oct 07 22:49:03 PDT 2023
;; MSG SIZE rcvd: 132

khod2@ada:~$ |
```

MX record returned:

```
khod2@ada:~$ dig @131.252.208.53 +tcp MX pdx.edu

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> @131.252.208.53 +tcp MX pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9457
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 3e1a560677b3292901000000652244ac74035c83a945508d (good)
;; QUESTION SECTION:
;pdx.edu.                IN      MX

;; ANSWER SECTION:
pdx.edu.                 59800   IN      MX      1 aspmx.l.google.com.
pdx.edu.                 59800   IN      MX      5 alt1.aspmx.l.google.com.
pdx.edu.                 59800   IN      MX      5 alt2.aspmx.l.google.com.
pdx.edu.                 59800   IN      MX      10 alt3.aspmx.l.google.com.
pdx.edu.                 59800   IN      MX      10 alt4.aspmx.l.google.com.

;; ADDITIONAL SECTION:
aspmx.l.google.com.     238     IN      A        172.253.117.26
aspmx.l.google.com.     39      IN      AAAA     2607:f8b0:400e:c06::1a

;; Query time: 0 msec
;; SERVER: 131.252.208.53#53(131.252.208.53) (TCP)
;; WHEN: Sat Oct 07 22:57:00 PDT 2023
;; MSG SIZE rcvd: 226

khod2@ada:~$ |
```

It looks like Amazon hosts the web site for www.pdx.edu.
Google handles mail for pdx.edu.

Find the A record for mashimaro.cs.pdx.edu:

```
khod2@ada:~$ dig NS mashimaro.cs.pdx.edu

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> NS mashimaro.cs.pdx.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 956
;; 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.                    87      IN      SOA      walt.ee.pdx.edu. support.cat.pdx.edu. 2023100302 600 300 1209600 300

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Sat Oct 07 23:15:02 PDT 2023
;; MSG SIZE rcvd: 105

khod2@ada:~$ dig @walt.ee.pdx.edu. A mashimaro.cs.pdx.edu

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> @walt.ee.pdx.edu. A mashimaro.cs.pdx.edu
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64660
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: a70ad403a376dcf201000000652248ecfb49b407ea4e312e (good)
;; QUESTION SECTION:
;mashimaro.cs.pdx.edu.          IN      A

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

;; Query time: 0 msec
;; SERVER: 131.252.208.38#53(walt.ee.pdx.edu.) (UDP)
;; WHEN: Sat Oct 07 23:15:08 PDT 2023
;; MSG SIZE rcvd: 93

khod2@ada:~$ |
```

2. F root server:

f.root-servers.net. 4027 IN A 192.5.5.241

dig @192.5.5.241 +tcp +norecurse NS console.cloud.google.com

f.gtld-servers.net. 172800 IN A 192.35.51.30

dig @192.35.51.30 +tcp +norecurse NS console.cloud.google.com

ns2.google.com. 172800 IN A 216.239.34.10

dig @216.239.34.10 +tcp +norecurse NS console.cloud.google.com

Result:

```
khod2@ada:~$ dig @216.239.34.10 +tcp +norecurse NS console.cloud.google.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> @216.239.34.10 +tcp +norecurse NS console.cloud.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37208
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

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

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

;; AUTHORITY SECTION:
l.google.com.              60      IN      SOA      ns1.google.com. dns-admin.google.com. 571287796 900 900 1800 60

;; Query time: 11 msec
;; SERVER: 216.239.34.10#53(216.239.34.10) (TCP)
;; WHEN: Sat Oct 07 23:38:50 PDT 2023
;; MSG SIZE rcvd: 124

khod2@ada:~$ |
```

So, we end by finding the CNAME record for the console.cloud.google.com address. We can't go further, as far as I know?

3. ChatGPT output for dig on espn.go.com:

SA Please 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 the `dig` command in combination with `egrep` and `awk` to list all IPv4 addresses associated with `espn.go.com`. Here's a single command line to accomplish this:

```
bash
egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print}'
```

Here's a breakdown of what this command does:

1. `dig +short espn.go.com`: This part of the command uses `dig` to query the A records for `espn.go.com` and the `+short` option to display the results in a shorter format.
2. `egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$'`: The `egrep` command filters the output to only include lines that match the pattern of IPv4 addresses (in the form of four decimal numbers separated by periods).
3. `awk '{print}'`: Finally, `awk` is used to print each line that matches the pattern, which effectively lists all the IPv4 addresses associated with `espn.go.com`.

Running this command will give you a list of IPv4 addresses for `espn.go.com`.

KHOD2

Output:

```
khod2@ada:~$ dig +short espn.go.com | egrep '^[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+$' | awk '{print}'
18.65.229.14
18.65.229.61
18.65.229.26
18.65.229.93
khod2@ada:~$ |
```

I do not really understand why we would not just use `dig +short espn.go.com`. It gave me the exact same answer, which I can only assume is the correct one.

Result of DNS reverse lookup on espn.go.com Ips:


```
khod2@ada:~$ X=$(dig +short espn.go.com)
khod2@ada:~$ for i in `echo $X`; do dig -x $i +short; done
server-18-65-229-93.sea73.r.cloudfront.net.
server-18-65-229-14.sea73.r.cloudfront.net.
server-18-65-229-26.sea73.r.cloudfront.net.
server-18-65-229-61.sea73.r.cloudfront.net.
khod2@ada:~$ |
```



4. Result of car hosts enumeration:


```
khod2@ada:~/CS430P/lab_files/02.2$ cat 220hosts.txt | head -185 | 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.
khod2@ada:~/CS430P/lab_files/02.2$ |
```


5. Geographic locations for 131.252.208.53:


Geolocation data from **ipinfo.io** (Product: API, real-time)


**IP ADDRESS:** 131.252.208.53


**COUNTRY:** United States 


**REGION:** Oregon

**CITY:** Portland

**ISP:** [Portland State University](#)


**ORGANIZATION:** Portland State University ([pdx.edu](#))



**LATITUDE:** 45.5234


**LONGITUDE:** -122.6762


KHOD2


Geolocation data from **DB-IP** (Product: API, real-time)


**IP ADDRESS:** 131.252.208.53


**COUNTRY:** United States 


**REGION:** Oregon

**CITY:** Portland (North Portland)

**ISP:** [Portland State University](#)


**ORGANIZATION:** Portland State University


**LATITUDE:** 45.584


**LONGITUDE:** -122.728

Geographic locations for 198.82.247.66:


Geolocation data from **ipinfo.io** (Product: API, real-time)


**IP ADDRESS:** 198.82.247.66


**COUNTRY:** United States 


**REGION:** Virginia

**CITY:** Blacksburg

**ISP:** Virginia Polytechnic Institute and State Univ.

**ORGANIZATION:** Virginia Polytechnic Institute and State Univ. ([vt.edu](#))

**LATITUDE:** 37.2296

**LONGITUDE:** -80.4139

KHOD2


Geolocation data from **DB-IP** (Product: API, real-time)


**IP ADDRESS:** 198.82.247.66


**COUNTRY:** United States 

**REGION:** Virginia

**CITY:** Blacksburg (Farmview - Ramble)

**ISP:** Virginia Polytechnic Institute and State Univ.

**ORGANIZATION:** Virginia Polytechnic Institute and State Univ.

**LATITUDE:** 37.2037

**LONGITUDE:** -80.4143

Resolutions for www.google.com from Portland:

:: ANSWER SECTION:

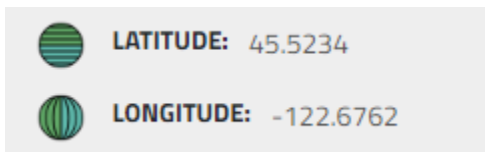
www.google.com. 269 IN A 142.250.217.100

Resolutions for www.google.com from Virginia:

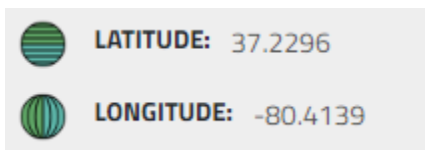
:: ANSWER SECTION:

www.google.com. 210 IN A 142.251.111.99

Geographic coordinates for the Portland DNS server, then the www.google.com server:



Geographic coordinates for the Virginia DNS server, then the Google server:



They are very close! Local servers.

Traceroute for the two DNS servers:

```

khod2@ada:~/CS430P/lab_files/02.2$ 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) 0.740 ms 0.596 ms 0.468 ms
khod2@ada:~/CS430P/lab_files/02.2$ 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) 9.166 ms 9.027 ms 9.042 ms
 2 CORE1.net.pdx.edu (131.252.5.142) 4.689 ms 4.580 ms 4.482 ms
 3 131.252.5.213 (131.252.5.213) 0.656 ms 0.557 ms 0.462 ms
 4 port-psu-pe-01.net.linkoregon.org (199.165.177.48) 0.491 ms 0.705 ms 2.036 ms
 5 eugn-oh-vpn-01.net.linkoregon.org (207.98.126.3) 10.314 ms 10.220 ms 10.123 ms
 6 bois-gtwy-pe-01.net.linkoregon.org (207.98.126.135) 10.320 ms 10.589 ms 9.991 ms
 7 bois-gtwy-pe-01-loren.net.linkoregon.org (163.253.5.65) 10.174 ms 10.048 ms 9.942 ms
 8 hundredge-0-0-0-24.703.core1.bois.net.internet2.edu (163.253.5.64) 11.896 ms 12.161 ms 11.989 ms
 9 fourhundredge-0-0-0-0.4079.core2.salt.net.internet2.edu (163.253.1.249) 74.574 ms 66.274 ms 66.143 ms
10 fourhundredge-0-0-0-23.4079.core1.salt.net.internet2.edu (163.253.1.32) 64.511 ms fourhundredge-0-0-0-22.4079.core1.salt.net.internet
2.edu (163.253.1.30) 66.220 ms fourhundredge-0-0-0-21.4079.core1.salt.net.internet2.edu (163.253.1.28) 66.100 ms
11 fourhundredge-0-0-0-0.4079.core2.kans.net.internet2.edu (163.253.1.251) 64.265 ms fourhundredge-0-0-0-0.4079.core1.denv.net.internet2
.edu (163.253.1.170) 64.349 ms fourhundredge-0-0-0-0.4079.core2.kans.net.internet2.edu (163.253.1.251) 64.039 ms
12 fourhundredge-0-0-0-0.4079.core1.kans.net.internet2.edu (163.253.1.243) 64.813 ms 64.686 ms 66.445 ms
13 fourhundredge-0-0-0-3.4079.core2.chic.net.internet2.edu (163.253.1.244) 66.349 ms 66.224 ms 66.127 ms
14 fourhundredge-0-0-0-3.4079.core2.eqch.net.internet2.edu (163.253.2.19) 66.015 ms 65.842 ms 65.272 ms
15 fourhundredge-0-0-0-0.4079.core2.clev.net.internet2.edu (163.253.2.16) 63.347 ms 65.017 ms 64.361 ms
16 fourhundredge-0-0-0-3.4079.core2.ashb.net.internet2.edu (163.253.1.138) 65.782 ms 65.427 ms 65.256 ms
17 192.122.175.14 (192.122.175.14) 63.244 ms 64.553 ms 64.394 ms
18 vtacs-1.msap.cns.vt.edu (192.70.187.18) 66.170 ms 66.012 ms 65.863 ms
19 isb-core.xe-7-0-0-0.cns.vt.edu (128.173.0.202) 68.025 ms 67.872 ms 68.143 ms
20 cas-core.lo0.2000.cns.vt.edu (198.82.1.143) 67.246 ms 67.312 ms 66.947 ms
21 jeru.cns.vt.edu (198.82.247.66) 67.825 ms 67.680 ms 67.815 ms
khod2@ada:~/CS430P/lab_files/02.2$ |

```

Traceroute for both google servers:

```

khod2@ada:~/CS430P/lab_files/02.2$ 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) 1.097 ms 0.894 ms 0.927 ms
 2 CORE1.net.pdx.edu (131.252.5.142) 5.322 ms 5.218 ms 5.071 ms
 3 131.252.5.213 (131.252.5.213) 0.596 ms 0.478 ms 0.345 ms
 4 google.nwax.net (198.32.195.34) 6.773 ms 4.141 ms 3.933 ms
 5 108.170.245.97 (108.170.245.97) 4.965 ms 4.849 ms 5.295 ms
 6 142.251.55.201 (142.251.55.201) 4.414 ms 142.251.55.203 (142.251.55.203) 4.369 ms 142.251.55.201 (142.251.55.201) 4.504 ms
 7 sea09s30-in-f4.1e100.net (142.250.217.100) 4.044 ms 3.819 ms 3.718 ms
khod2@ada:~/CS430P/lab_files/02.2$ traceroute 142.251.111.99
traceroute to 142.251.111.99 (142.251.111.99), 30 hops max, 60 byte packets
 1 radiant.seas.pdx.edu (131.252.208.212) 0.942 ms 0.944 ms 0.961 ms
 2 CORE1.net.pdx.edu (131.252.5.142) 2.374 ms 2.270 ms 2.158 ms
 3 131.252.5.213 (131.252.5.213) 0.453 ms 0.347 ms 0.356 ms
 4 google.nwax.net (198.32.195.34) 3.820 ms 4.100 ms 4.082 ms
 5 108.170.245.124 (108.170.245.124) 4.665 ms 108.170.245.123 (108.170.245.123) 4.933 ms 74.125.243.198 (74.125.243.198) 4.557 ms
 6 216.239.57.194 (216.239.57.194) 11.826 ms 216.239.43.88 (216.239.43.88) 15.192 ms 142.250.228.152 (142.250.228.152) 11.523 ms
 7 192.178.74.222 (192.178.74.222) 50.891 ms 192.178.74.212 (192.178.74.212) 56.461 ms 192.178.74.214 (192.178.74.214) 93.410 ms
 8 192.178.72.205 (192.178.72.205) 59.105 ms 192.178.72.195 (192.178.72.195) 58.903 ms 192.178.72.203 (192.178.72.203) 57.975 ms
 9 192.178.81.238 (192.178.81.238) 75.018 ms * 192.178.81.232 (192.178.81.232) 71.813 ms
10 142.250.209.110 (142.250.209.110) 71.103 ms 142.251.70.30 (142.251.70.30) 72.621 ms 142.250.209.254 (142.250.209.254) 74.129 ms
11 142.251.68.9 (142.251.68.9) 70.882 ms 142.251.227.135 (142.251.227.135) 72.705 ms 142.251.66.227 (142.251.66.227) 69.734 ms
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * * *
20 * * *
21 bk-in-f99.1e100.net (142.251.111.99) 72.251 ms 71.838 ms 71.501 ms
khod2@ada:~/CS430P/lab_files/02.2$ |

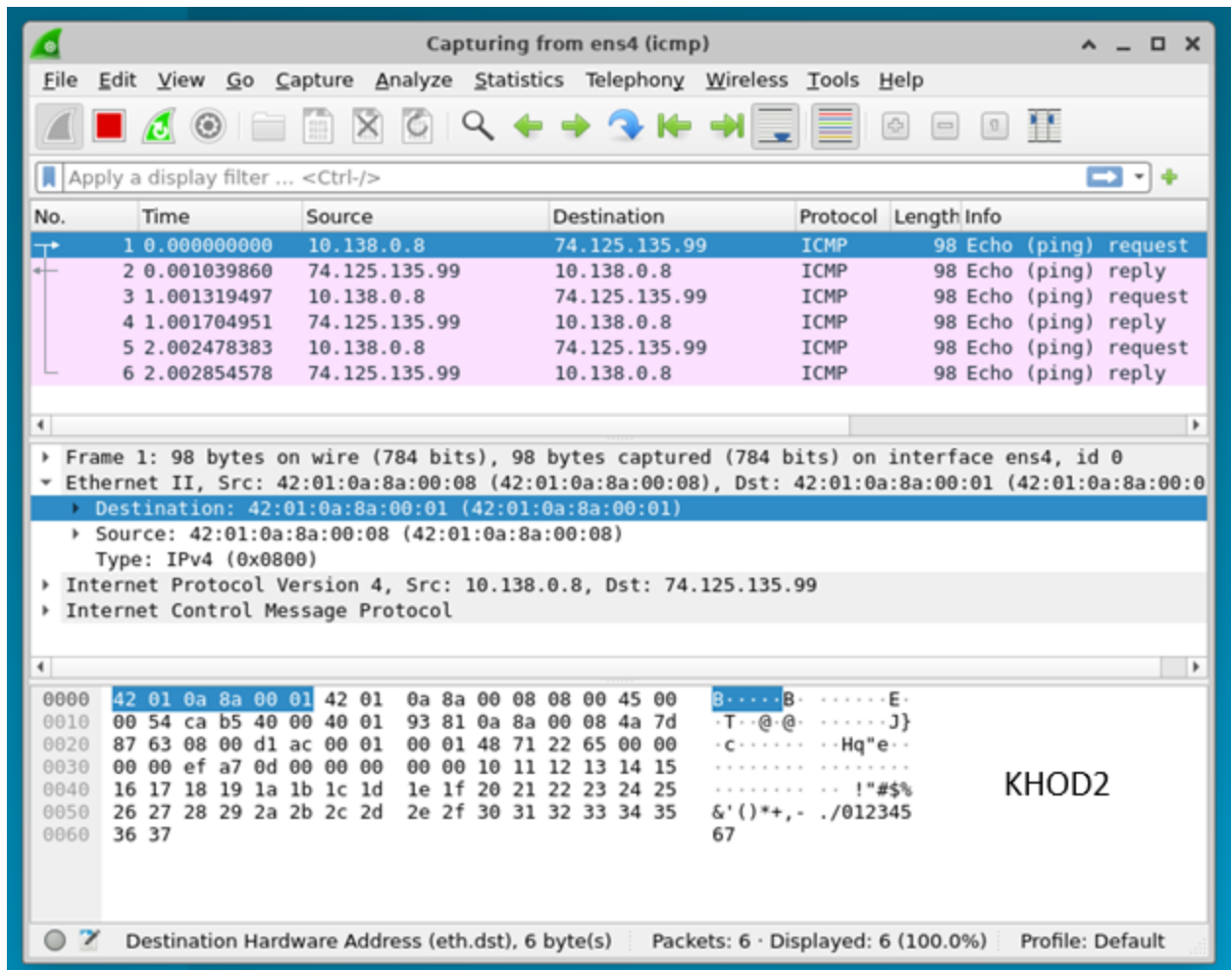
```

6. For my notes, the IP address of the VM: ip addr - 10.138.0.8/32

Name of the local virtual ethernet interface: ip addr - ens4

IP address of the default router (gateway address of 0.0.0.0): netstat -rn - 10.138.0.1

7. Bytes in the packet dump window:



The destination MAC address in the **first packet** corresponds to an interface on the default router since it's the default router's MAC address.

The destination MAC address on the second packet corresponds to an interface on the VM, since it's the local ethernet interface's hardware address.

10. All packets returned within wireshark using khod2.oregonctf.org filter:

No.	Time	Source	Destination	Protocol	Length
585	18.623376	169.254.169.254	10.138.0.8	HTTP/J...	1699
588	18.623871	10.138.0.8	169.254.169.254	HTTP	281
790	24.629018	169.254.169.254	10.138.0.8	HTTP/J...	1699
792	24.629744	10.138.0.8	169.254.169.254	HTTP	282
1157	43.085340	10.138.0.8	169.254.169.254	DNS	119
1158	43.085513	10.138.0.8	169.254.169.254	DNS	119
1159	43.091523	169.254.169.254	10.138.0.8	DNS	135
1160	43.096919	169.254.169.254	10.138.0.8	DNS	208
1162	43.120122	42:01:0a:8a:00:08	Broadcast	ARP	42
1163	43.121232	42:01:0a:8a:00:01	42:01:0a:8a:00:08	ARP	42
1164	43.121248	10.138.0.8	169.254.169.254	DNS	90
1173	43.206792	169.254.169.254	10.138.0.8	DNS	172
1174	43.207702	10.138.0.8	35.233.233.233	TCP	74
1175	43.209278	35.233.233.233	10.138.0.8	TCP	74
1176	43.209389	10.138.0.8	35.233.233.233	TCP	66
1177	43.209533	10.138.0.8	35.233.233.233	HTTP	200
1178	43.210142	35.233.233.233	10.138.0.8	TCP	66
1179	43.210371	35.233.233.233	10.138.0.8	TCP	7106
1180	43.210371	35.233.233.233	10.138.0.8	HTTP	792
1181	43.210394	10.138.0.8	35.233.233.233	TCP	66
1182	43.210400	10.138.0.8	35.233.233.233	TCP	66
1185	43.212166	10.138.0.8	35.233.233.233	TCP	66
1186	43.212625	35.233.233.233	10.138.0.8	TCP	66
1187	43.212653	10.138.0.8	35.233.233.233	TCP	66

KHOD2

ARP:

Packets 1162 and 1163 are the result of the VM getting the hardware address of the default router.

The hardware address of the default router is 42:01:0a:8a:00:01

DNS:

Packets 1164 and 1173 are the result of the DNS request for the oregonctf.org website.

The IP address of the local DNS server being queried is 169.254.169.254.

TCP:

Packets 1174, 1175, and 1176 are part of the initial TCP handshake for a web request.

It takes 0.001687 seconds from the time that the first packet is recorded to the last. I used Delta Time Displayed to calculate this.

TCP	74	0.000910
TCP	74	0.001576
TCP	66	0.000111

HTTP:

Packets 1177 and 1180 are the HTTP request and response for the website.

It takes 0.000982 seconds (from the end of the handshake) to complete the HTTP request and response.

TCP	66	0.000111
HTTP	200	0.000144
TCP	66	0.000609
TCP	7106	0.000229
HTTP	792	0.000000