



***Universidad Nacional
Autónoma de México
Facultad de Ciencias***



Facultad de Ciencias

Criptografía y Seguridad

Semestre: 2025-1

Equipo: Pingüicoders

**Práctica 2:
*Man in the Middle***

Arrieta Mancera Luis Sebastián - 318174116

Cruz Cruz Alan Josue - 319327133

García Ponce José Camilo - 319210536

Matute Cantón Sara Lorena - 319331622

Introducción:

La práctica trata sobre conocer un poco sobre los ataques Man in the Middle y algunas herramientas relacionadas a cosas de redes y sobre este tipo de ataques. Le confiamos a nuestras computadoras casi, sino es que toda nuestra información confidencial por lo que es muy importante tener conocimiento de estos tipos de ataques para poder intentar evitarlas o en el caso de que seamos una víctima tener un conocimiento básico de su funcionamiento, esto con el fin de poder minimizar daños. Otro aspecto importante es conocer la gran cantidad de riesgos o situaciones no favorables que podemos experimentar al conectarnos a una red.

Desarrollos:

- Alan:

Lo primero es instalar las herramientas iniciales de la primera parte

```
(kali㉿kali)-[~]
└─$ sudo apt install net-tools
[sudo] password for kali:
net-tools is already the newest version (2.10-1.1).
net-tools set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 766

(kali㉿kali)-[~]
└─$ sudo ifconfig -a
          interface: eth0             Link layer type:Ethernet
          HWaddr 08:00:27:75:3a:56      MTU:1500 Queueingdisc:bio
          Brdrgmt:255.255.255.0        broadcast:10.0.2.255
          Mask:255.255.255.0          broadcast:10.0.2.255
          inet 10.0.2.15 brd 10.0.2.255 netmask 255.255.255.0
          ether 08:00:27:75:3a:56      txqueuelen 1000 (Ethernet)
          RX packets 49098 bytes 44119017 (42.0 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 25338 bytes 12184007 (11.6 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

          interface: lo               Link layer type:Loopback
          HWaddr 00:00:00:00:00:00      MTU:65536 Queueingdisc:bio
          inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
          loop txqueuelen 1000 (Local Loopback)
          RX packets 8 bytes 480 (480.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 8 bytes 480 (480.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

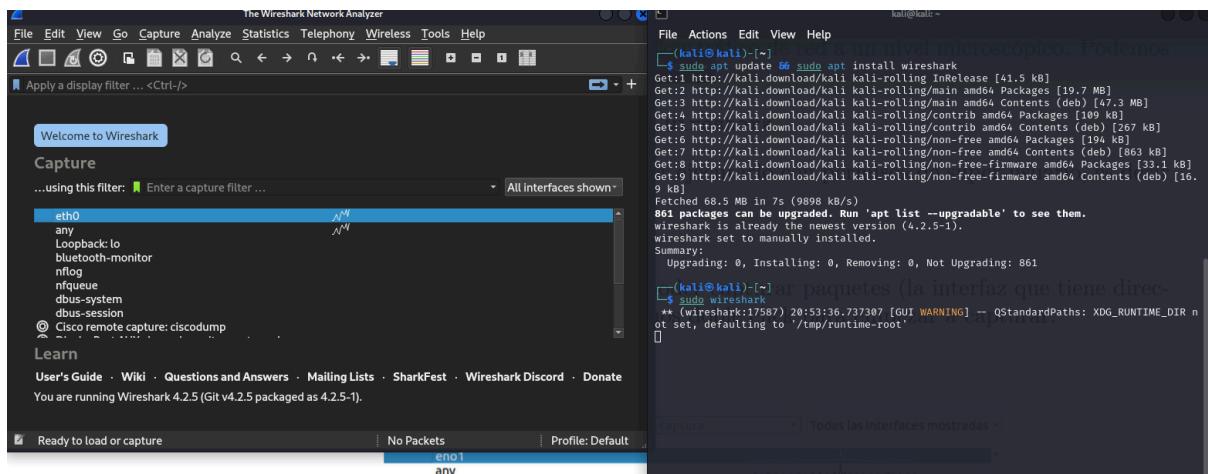
Luego el comando que muestra mis interfaces de red

```
(kali㉿kali)-[~]
└─$ sudo ifconfig -a
[sudo] password for kali:
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.2.15 brd 10.0.2.255  netmask 255.255.255.0
      ether 08:00:27:75:3a:56  txqueuelen 1000 (Ethernet)
      RX packets 27854 bytes 32285307 (30.7 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 7398 bytes 3366251 (3.2 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1 brd 127.0.0.1  netmask 255.0.0.0
      loop txqueuelen 1000 (Local Loopback)
      RX packets 8 bytes 480 (480.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 8 bytes 480 (480.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
```

Luego instalé Wireshark y ejecuté el programa. Posteriormente elegí eth0 para comenzar con el rastreo de la red.



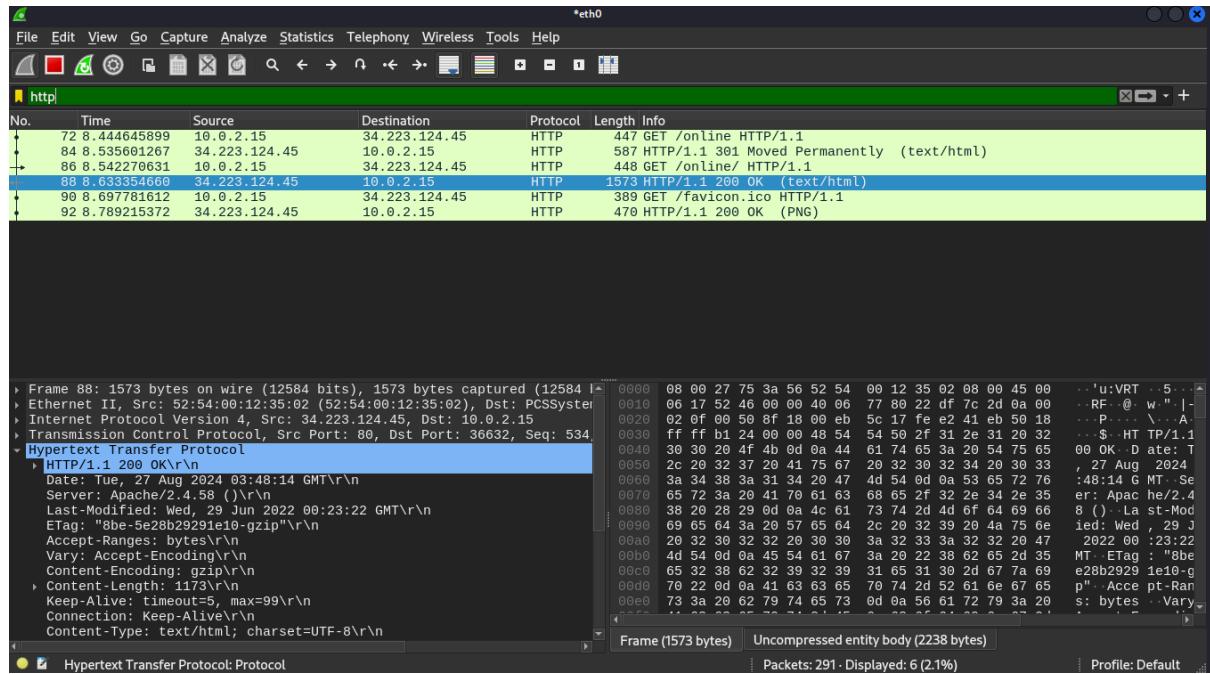
Luego comencé a ir a sitios donde tengo la sesión iniciada, aquí entré al dashboard de classroom y el programa estaba captando lo siguiente

The screenshot shows a Kali Linux desktop environment. A browser window is open to 'https://classroom.google.com'. In the foreground, a Wireshark capture window is displaying network traffic from interface 'eth0'. The traffic list shows various frames, including Ethernet, IP, TCP, and UDP layers, with detailed hex and ASCII dump information.

Y aquí entré a Youtube

The screenshot shows a Kali Linux desktop environment. A browser window is open to 'https://www.youtube.com/?gpm=desktop&hl=es'. In the foreground, a Wireshark capture window is displaying network traffic from interface 'eth0'. The traffic list shows various frames, including Ethernet, IP, TCP, and QUIC layers, with detailed hex and ASCII dump information.

Luego filtré con http los datos al entrar a la página que nos indican en la práctica, neverssl.com, los datos se ven distintos porque mi máquina virtual se trabó y tuve que reiniciarla.



Para nmap primero lo instalé

```
(kali㉿kali)-[~]
$ sudo apt install nmap
Upgrading:
  nmap nmap-common

Summary:
  Upgrading: 2, Installing: 0, Removing: 0, Not Upgrading: 859
  Download size: 6170 kB
  Freed space: 3072 B
  Continue? [Y/n] y
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.94+git20230807.3be01efb1+dfsg-2+kali3 [4241 kB]
Get:1 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.94+git20230807.3be01efb1+dfsg-2+kali3 [1929 kB]
Fetched 6170 kB in 1s (6258 kB/s)
(Reading database ... 392850 files and directories currently installed.)
Preparing to unpack .../nmap_7.94+git20230807.3be01efb1+dfsg-2+kali3_amd64.deb ...
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-2+kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2) ...
Preparing to unpack .../nmap-common_7.94+git20230807.3be01efb1+dfsg-2+kali3_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-2+kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2) ...
Setting up nmap-common (7.94+git20230807.3be01efb1+dfsg-2+kali3) ...
Setting up nmap (7.94+git20230807.3be01efb1+dfsg-2+kali3) ...
Processing triggers for kali-menu (2023.4.7) ...
Processing triggers for man-db (2.12.1-1) ...
Processing triggers for wordlists (2023.2.0) ...
nmap done: 1 IP address (1 host up) scanned

```

Ejecuté el comando de sudo nmap -sn con la ip y submáscara 24 y la bandera -R. Honestamente no estoy muy seguro de cómo interpretar la información.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sn 10.0.2.15/24 -R
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-26 22:01 CST
Nmap scan report for 10.0.2.2
Host is up (0.000088s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.000085s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.00029s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 31.09 seconds

(kali㉿kali)-[~]
└─$
```

Luego sigue instalar Bettercap, seguí solamente los pasos de la práctica

```
→ $ sudo apt-get install build-essential ruby-dev libpcap-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10).
ruby-dev is already the newest version (1:3.1+mu1).
The following additional packages will be installed:
  libdbus-1-dev libpcap0.8-dev libpcapconf3 pkgconf pkgconf-bin
The following NEW packages will be installed:
  libdbus-1-dev libpcap-dev libpcap0.8-dev libpcapconf3 pkgconf pkgconf-bin
0 upgraded, 6 newly installed, 0 to remove and 859 not upgraded.
Need to get 640 kB of archives.
After this operation, 2178 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 libpcapconf3 amd64 1.8.1-3 [36.2 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 pkgconf amd64 1.8.1-3 [26.1 kB]
Get:4 http://http.kali.org/kali kali-rolling/main amd64 libdbus-1-dev amd64 1.14.10-4+b1 [242 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 libpcap0.8-dev amd64 1.10.4-5 [277 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 libpcap-dev amd64 1.10.4-5 [28.9 kB]
Get:2 https://mirror.cedia.org.ec/kali kali-rolling/main amd64 pkgconf-bin amd64 1.8.1-3 [29.9 kB]
Fetched 640 kB in 2s (420 kB/s)
Selecting previously unselected package libpcapconf3:amd64.
(Reading database ... 392849 files and directories currently installed.)
Preparing to unpack .../0-libpcapconf3_1.8.1-3_amd64.deb ...
Unpacking libpcapconf3:amd64 (1.8.1-3) ...
Selecting previously unselected package pkgconf-bin.
Preparing to unpack .../1-pkgconf-bin_1.8.1-3_amd64.deb ...
Unpacking pkgconf-bin (1.8.1-3) ...
Selecting previously unselected package pkgconf:amd64.
```

```
(kali㉿kali)-[~]
↳ sudo gem install bettercap ; gem update bettercap
Building native extensions. This could take a while...
Successfully installed pcaprub-0.13.3
Successfully installed packetfu-1.1.13
Building native extensions. This could take a while...
Successfully installed network_interface-0.0.4
Successfully installed net-dns-0.20.0
Building native extensions. This could take a while...
Successfully installed eventmachine-1.2.7
Successfully installed em-proxy-0.1.9
Successfully installed colorize-0.8.1
Successfully installed bettercap-1.6.2
Parsing documentation for pcaprub-0.13.3
Installing ri documentation for pcaprub-0.13.3
Parsing documentation for packetfu-1.1.13
Installing ri documentation for packetfu-1.1.13
Parsing documentation for network_interface-0.0.4
Installing ri documentation for network_interface-0.0.4
Parsing documentation for net-dns-0.20.0
Installing ri documentation for net-dns-0.20.0
Parsing documentation for eventmachine-1.2.7
Installing ri documentation for eventmachine-1.2.7
Parsing documentation for em-proxy-0.1.9
Installing ri documentation for em-proxy-0.1.9
Parsing documentation for colorize-0.8.1
Installing ri documentation for colorize-0.8.1
Parsing documentation for bettercap-1.6.2
Installing ri documentation for bettercap-1.6.2
Done installing documentation for pcaprub, packetfu, network_interface, net-dns, eventmachine, em-proxy, colorize, bettercap after 5 seconds
8 gems installed
```

y verifiqué la instalación

```

(kali㉿kali)-[~] $ bettercap --check-updates
flag provided but not defined: -check-updates
Usage of bettercap:
  -autostart string
    Comma separated list of modules to auto start. (default "events.stream")
  -caplet string
    Read commands from this file and execute them in the interactive session.
  -caplets-path string
    Specify an alternative base path for caplets.
  -cpu-profile file
    Write cpu profile file.
  -debug
    Print debug messages.
  -env-file string
    Load environment variables from this file if found, set to empty to disable environment persistence.
  -eval string
    Run one or more commands separated by ; in the interactive session, used to set variables via command line.
  -gateway-override string
    Use the provided IP address instead of the default gateway. If not specified or invalid, the default gateway will be used.
  -iface string
    Network interface to bind to, if empty the default interface will be auto select ed.
  -mem-profile file
    Write memory profile to file.
  -no-colors
    Disable output color effects.
  -no-history
    Disable interactive session history file.
  -pcap-buf-size int
    PCAP buffer size, leave to 0 for the default value. (default -1)
  -script string
    Load a session script.
  -silent
    Suppress all logs which are not errors.
  -version
    Print the version and exit.

(kali㉿kali)-[~] $ bettercap --version
bettercap v2.32.0 (built for linux amd64 with go1.22.3)

```

Ahora iniciamos el ataque MitM, como no tenía muchos dispositivos a la mano tuve que probarlo con mi misma máquina pero hubo un problema y creo que era muy obvio (je).

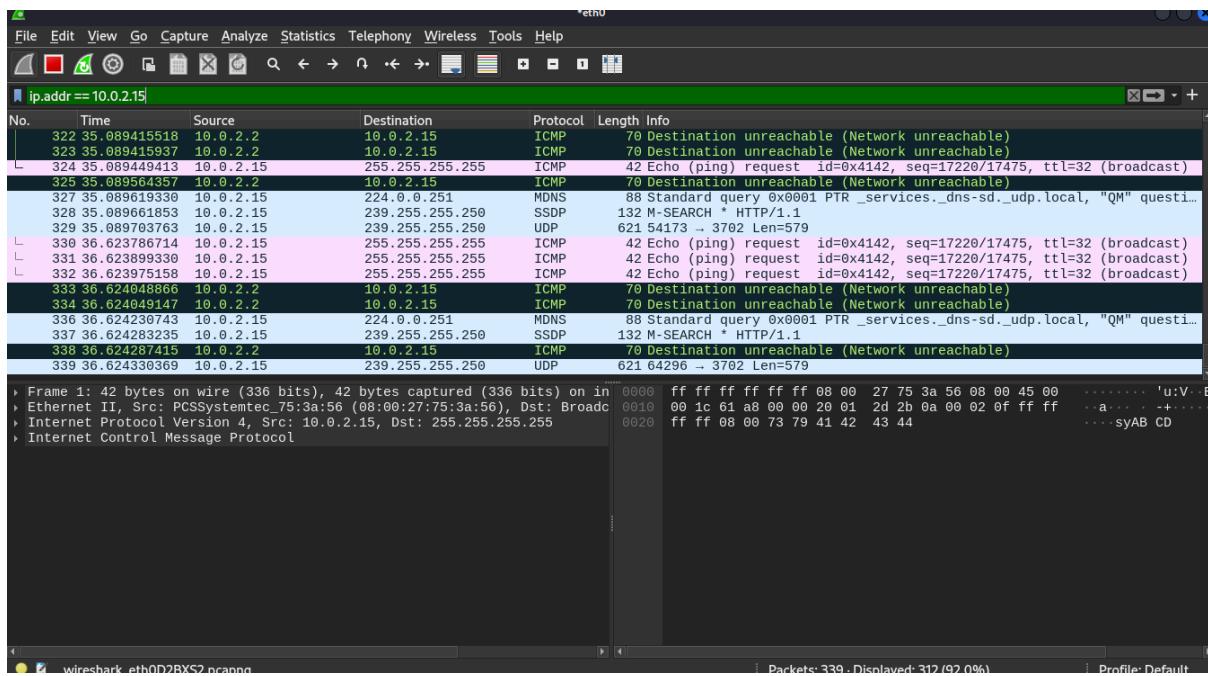
```

(kali㉿kali)-[~] $ sudo bettercap --interface eth0 --sniffer --target 10.0.2.15
[sudo] password for kali:
[!] Starting [ spoofing:v discovery:x sniffer:v tcp-proxy:x udp-proxy:x http-proxy:x https-proxy:x sslstrip:x http-server:x dns-server:x ] ...
[!] [eth0] 10.0.2.15 : 08:00:27:75:3A:56 / eth0 ( PCS Systemtechnik GmbH )
[!] [GATEWAY] 10.0.2.2 : 52:54:00:12:35:02 ( ??? )

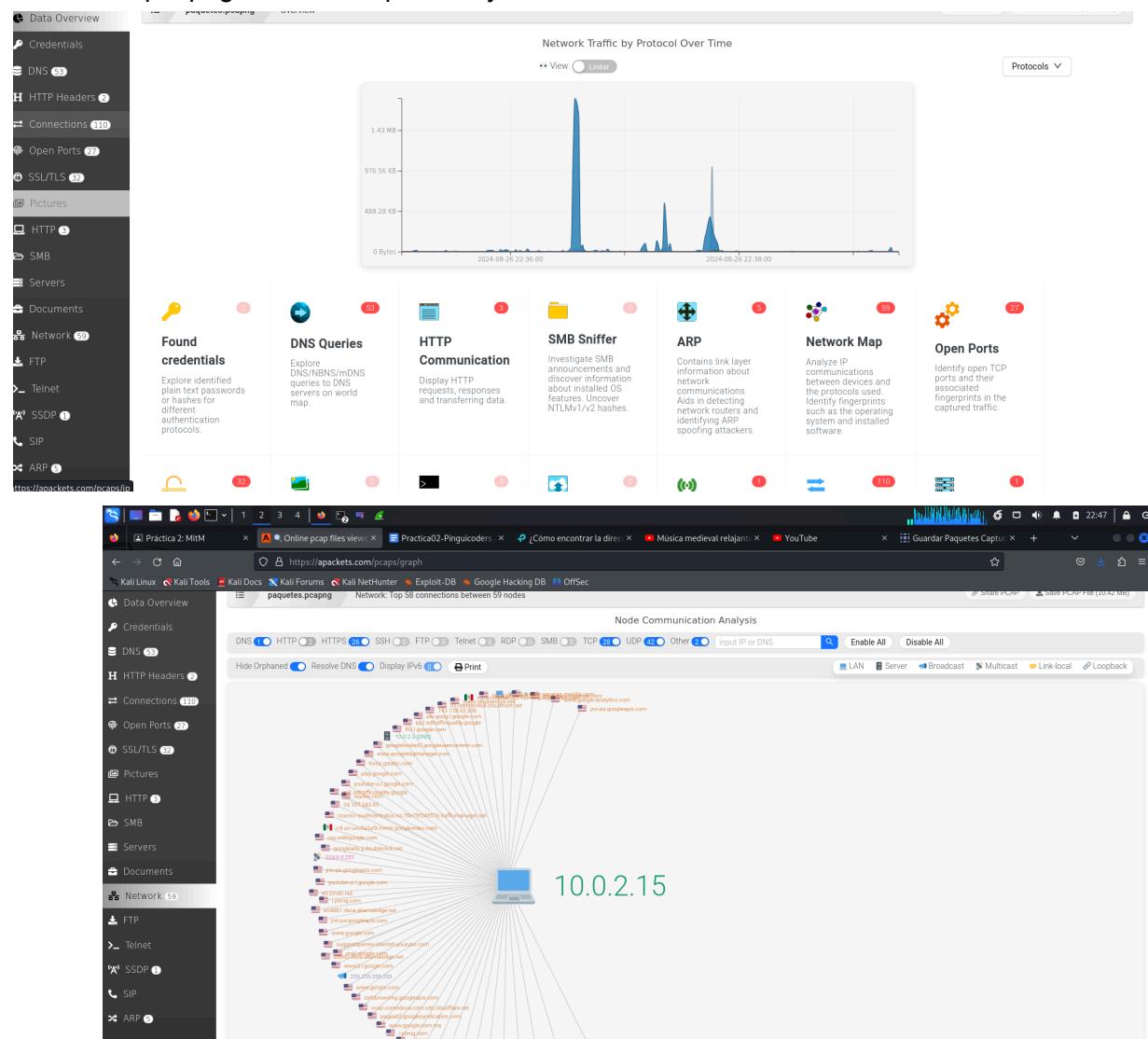
```

Es que al parecer no estaba reconociendo el proceso, supongo que el problema es que me estoy espiando a mí mismo en el sentido de que estoy escuchando el mismo sistema en el que estoy pero para el tiempo en que hice esta parte no pude encontrar una alternativa ya que lo estaba usando en mi máquina virtual.

Para lo de wireshark puse la dirección ip de antes y me captó los paquetes así que decidí guardarlos y seguir con la práctica.



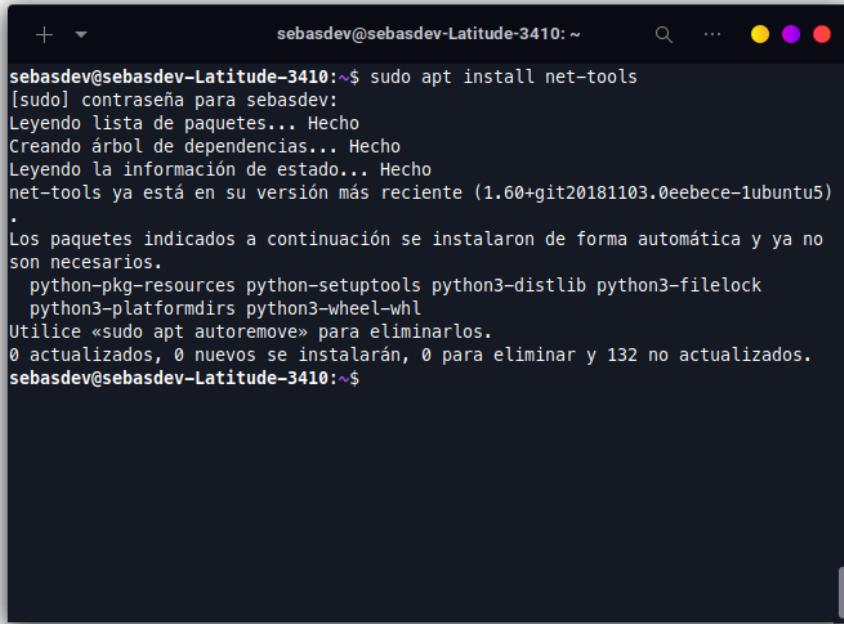
El archivo .pcapng. lo subí a a-packets y este fue el resultado.



Me aparecen sitios que ni tengo idea de qué son pero ahí dicen que son de Estados Unidos, particularmente son sitios relacionados con google porque no uso la máquina virtual más que para cosas de la materia.

- Sebastian

Instalación de net-tools



```
sebasdev@sebasdev-Latitude-3410:~$ sudo apt install net-tools
[sudo] contraseña para sebasdev:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
net-tools ya está en su versión más reciente (1.60+git20181103.0eebece-1ubuntu5)
.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  python-pkg-resources python-setuptools python3-distlib python3-filelock
  python3-platformdirs python3-wheel-whl
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 132 no actualizados.
sebasdev@sebasdev-Latitude-3410:~$
```

prueba sudo ifconfig -a

```
sebasdev@sebasdev-Latitude-3410:~$ sudo ifconfig -a
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
                ether 02:42:43:41:82:53 txqueuelen 0 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        ether b0:7b:25:3b:e7:c9 txqueuelen 1000 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                    loop txqueuelen 1000 (Bucle local)
                    RX packets 1701 bytes 177927 (177.9 KB)
                    RX errors 0 dropped 0 overruns 0 frame 0
                    TX packets 1701 bytes 177927 (177.9 KB)
                    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.73 netmask 255.255.255.0 broadcast 192.168.1.255
                inet6 fe80::8d14:c0f9:1f80:b973 prefixlen 64 scopeid 0x20<link>
                inet6 2806:104e:18:ad45:aa3c:a211:8f84:bca2 prefixlen 64 scopeid 0x0<global>
                inet6 2806:104e:18:ad45:cfdc:99b8:470f:95f0 prefixlen 64 scopeid 0x0<global>
                ether 18:26:49:6b:29:7a txqueuelen 1000 (Ethernet)
                RX packets 178595 bytes 248839662 (248.8 MB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 54176 bytes 7760216 (7.7 MB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

sebasdev@sebasdev-Latitude-3410:~$
```

De la interfaz de red wlp0s20f3 podemos extraer los siguientes valores:

- + **IP:** 192.168.1.73
- + **MAC:** 18:26:49:6b:29:7a
- + **Submáscara de red:** 255.255.255.0

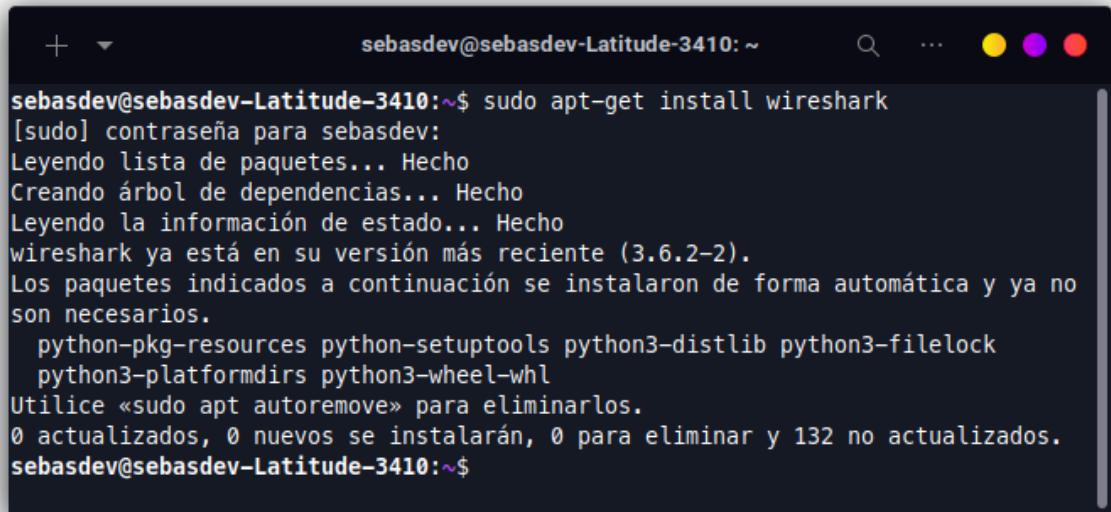
Wireshark

Con los comandos del pdf de la práctica no se pudo instalar wireshark, ya que aparecía el siguiente error:

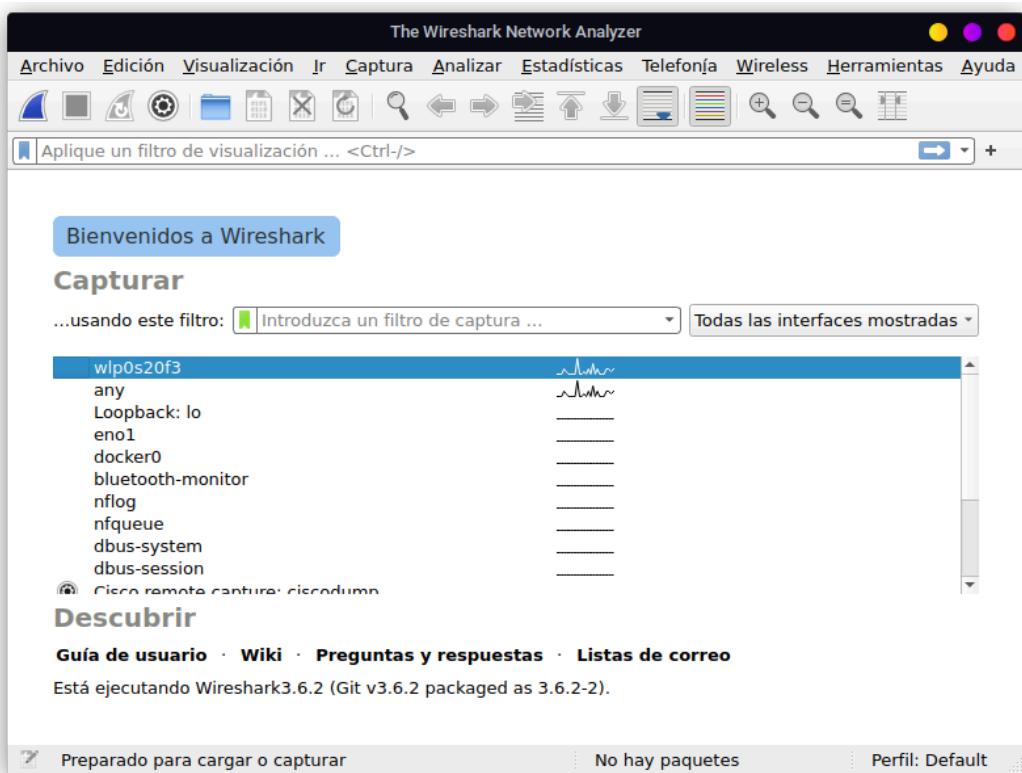
E: El repositorio «https://ppa.launchpadcontent.net/ys/emoji-one-picker/ubuntu jammy Release» no tiene un fichero de Publicación

Sin embargo, encontré una página donde recomendaban instalar wireshark con el comando:

```
sudo apt-get install wireshark
```



```
sebasdev@sebasdev-Latitude-3410:~$ sudo apt-get install wireshark
[sudo] contraseña para sebasdev:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
wireshark ya está en su versión más reciente (3.6.2-2).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  python-pkg-resources python-setuptools python3-distlib python3-filelock
  python3-platformdirs python3-wheel-whl
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 132 no actualizados.
sebasdev@sebasdev-Latitude-3410:~$
```



Notemos que wireshark nos enlista la interfaz de red **wlp0s20f3** que visualizamos con el comando ifconfig. Capturamos el tráfico de esta interfaz de red y navegamos por páginas de internet como youtube, amazon, facebook e instagram. Aplicamos los filtros para revisar de manera específica los paquetes relacionados a los sitios que se visitan.

*wlp0s20f3

Archivo Edición Visualización Ir Captura Analizar Estadísticas Teléfono Wireless Herramientas Ayuda

frame contains youtube

No.	Time	Source	Destination	Protocol	Length Info
5763	11.327347152	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	126 Standard query 0xa32c AAAA suggestqueries-clients6.youtube.com OPT
5764	11.327518692	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	126 Standard query 0x113a A suggestqueries-clients6.youtube.com OPT
5765	11.327575653	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	126 Standard query 0xc9a3 HTTPS suggestqueries-clients6.youtube.com OPT
5824	11.344368863	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	142 Standard query response 0x113a A suggestqueries-clients6.youtube.com A 192.178.52.192
5831	11.354579514	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	154 Standard query response 0xa32c AAAA suggestqueries-clients6.youtube.com AAAA 2607:1880:6000:1:1:1:1:1
5833	11.354631189	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	183 Standard query response 0xc9a3 HTTPS suggestqueries-clients6.youtube.com SOA ns1.google.com.
5884	11.414248306	2806:104e:18:ad45:a...	2607:fbb0:4012:81e:...	TLSv1.3	1863 Client Hello
9530	15.122225699	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	102 Standard query 0x0207 AAAA youtube.com OPT
9541	15.122305954	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	102 Standard query 0x99f0 A youtube.com OPT
9541	15.122305954	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	102 Standard query 0x660d HTTPS youtube.com OPT
9551	15.149232344	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	130 Standard query response 0x0207 AAAA youtube.com AAAA 2607:fbb0:4012:81e::200e OPT
9551	15.149232270	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	118 Standard query response 0x99f0 A youtube.com A 172.217.3.142 OPT
9551	15.149232984	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	161 Standard query response 0x660d HTTPS youtube.com HTTPS A 192.178.52.174 AAAA 2607:fbb0:4012:81e:200e OPT
9551	15.180035416	2806:104e:18:ad45:a...	2607:fbb0:4012:81e:...	TLSv1.3	1839 Client Hello
9575	15.25337279	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	111 Standard query 0x14fc AAAA accounts.youtube.com OPT
9576	15.254143690	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	111 Standard query 0x34ac A accounts.youtube.com OPT
9577	15.254231089	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	111 Standard query 0x82ce HTTPS accounts.youtube.com OPT
9587	15.281642343	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	155 Standard query response 0x34ac A accounts.youtube.com CNAME www3.l.google.com A 14.217.217.3.142 OPT
9588	15.281629907	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	189 Standard query response 0x82ce HTTPS accounts.youtube.com CNAME www3.l.google.com A 14.217.217.3.142 OPT
9599	15.281631115	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	167 Standard query response 0x14fc AAAA accounts.youtube.com CNAME www3.l.google.com A 14.217.217.3.142 OPT

```
> Frame 9595: 167 bytes on wire (1336 bits), 167 bytes captured (1336 bits) on interface wlp0s20f3, id 0
> Ethernet II, Src: DpNetTec_d9:33:73 (98:3b:67:d9:33:73), Dst: IntelCor_6b:29:7a (18:26:49:60:29:7a)
> Internet Protocol Version 6, Src: 2806:1040:ffff:3::e, Dst: 2806:104e:18:ad45:aa3c:a211:8f84:bca2
> User Datagram Protocol, Src Port: 53, Dst Port: 42476
> Domain Name System (response)
```

```
0000 18 26 49 6b 29 7a 98 3b 67 d9 33 73 86 dd 60 00 -&IKz; g:3s-
0010 00 00 00 71 11 3d 28 06 10 4e ff ff 00 03 00 00 ...q=(<@...
0020 00 00 00 00 08 28 06 10 4e 00 18 ad 45 aa 3c .....( N- E<
0030 a2 11 8f 84 bc a2 06 35 a5 ec 00 71 64 6c 14 fc .....5 ..qd1..
0040 81 80 00 01 02 00 00 00 01 01 08 61 63 63 75 .....accou...
0050 6e 74 73 07 79 6f 75 74 75 62 65 03 63 6f 00 nts.youtube.com...
0060 00 01 00 01 c8 00 05 00 00 01 00 00 01 19 00 10 .....
0070 04 77 77 77 33 01 6c 06 67 6f 67 6c 65 01 cd www3.l.google.com...
0080 c0 32 00 01 00 01 00 00 01 19 00 10 26 07 f8 b0 ..2 .....&...
0090
```

Paquetes: 27028 - Mostrado: 21 (0.1%) Perdido: 0 (0.0%) Perfil: Default

*wlp0s20f3

Archivo Edición Visualización Ir Captura Analizar Estadísticas Teléfono Wireless Herramientas Ayuda

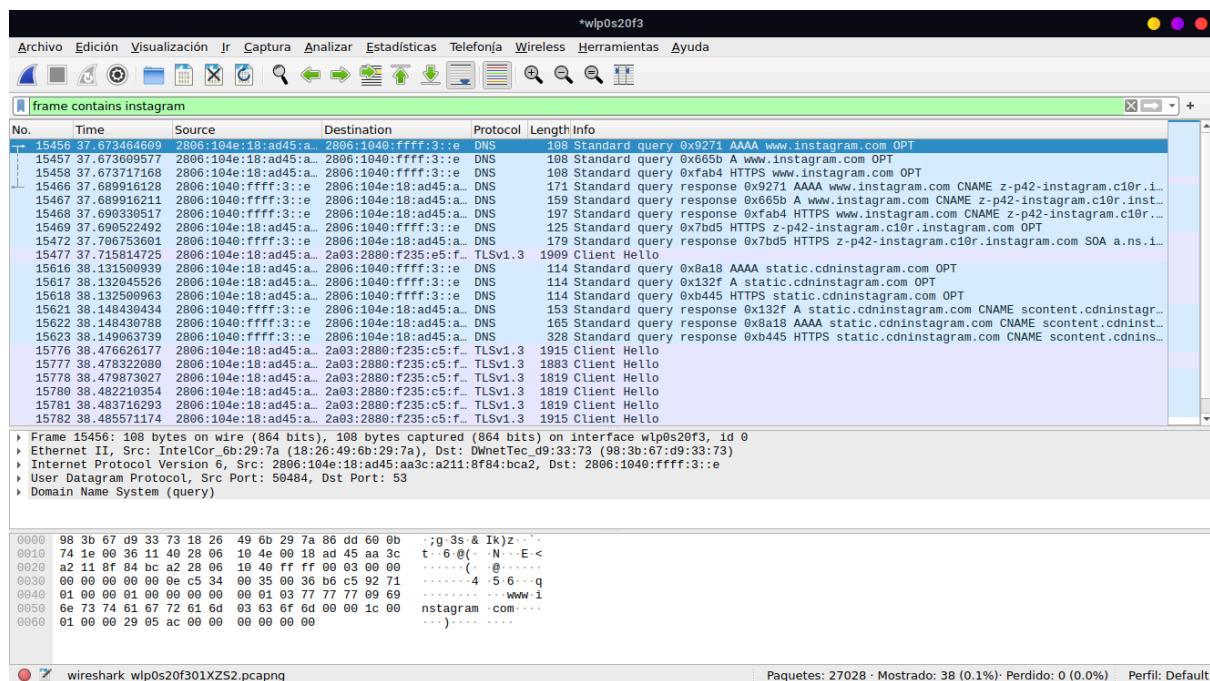
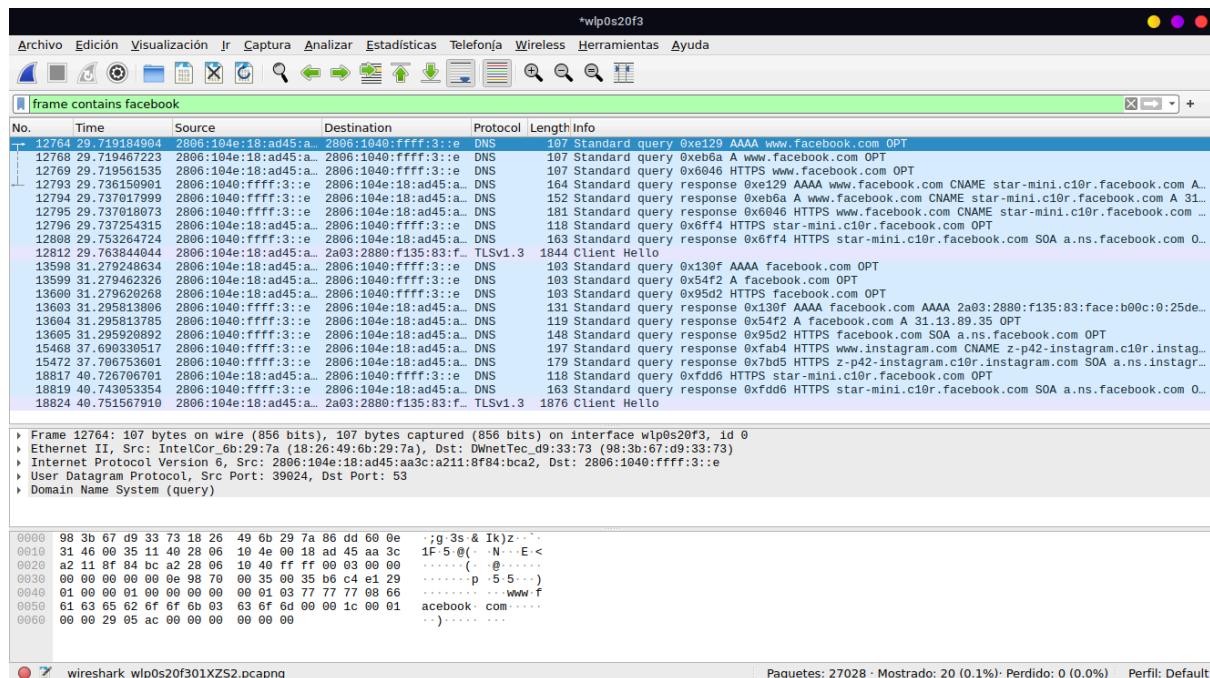
frame contains amazon

No.	Time	Source	Destination	Protocol	Length Info
18475	20.099508643	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	108 Standard query 0x7abd AAAA www.amazon.com.mx OPT
18476	20.099610164	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	108 Standard query 0x444e A www.amazon.com.mx OPT
18477	20.099696403	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	108 Standard query 0x9113 HTTPS www.amazon.com.mx OPT
18483	20.106506919	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	280 Standard query response 0x7abd AAAA www.amazon.com.mx CNAME tp.4e37fb303-frontier...
18484	20.107408791	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	288 Standard query response 0x9113 HTTPS www.amazon.com.mx CNAME tp.4e37fb303-frontier...
18485	20.107496805	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	293 Standard query response 0x444e A www.amazon.com.mx CNAME tp.4e37fb303-frontier.amazon.com...
18493	20.136116412	2806:104e:18:ad45:a...	2609:141c:e009:18a:...	TLSv1.3	2116 Client Hello
18794	20.769939436	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	122 Standard query 0xc586 AAAA images-na.ssl-images-amazon.com OPT
18797	20.770055763	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	122 Standard query 0x6f78 A images-na.ssl-images-amazon.com OPT
18798	20.770144997	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	122 Standard query 0xc4e2 HTTPS images-na.ssl-images-amazon.com OPT
18799	20.771131234	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	109 Standard query 0x5aaa AAAA m.media.amazon.com OPT
18800	20.771539829	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	109 Standard query 0x37e5 A m.media.amazon.com OPT
18801	20.771635192	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	109 Standard query 0x2dd8 HTTPS m.media.amazon.com OPT
18802	20.7717124462	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	112 Standard query 0x4ed8 AAAA completion.amazon.com OPT
18803	20.771820813	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	112 Standard query 0x1af0 A completion.amazon.com OPT
18804	20.771830046	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	112 Standard query 0xf6f8 HTTPS completion.amazon.com OPT
18806	20.787119123	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	302 Standard query response 0x4e2 HTTPS images-na.ssl-images-amazon.com CNAME m.media...
18807	20.787119276	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	272 Standard query response 0xc586 AAAA images-na.ssl-images-amazon.com CNAME m.media...
18808	20.787140812	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	295 Standard query response 0x5aaa AAAA m.media.amazon.com CNAME tp.c47710ee9-frontier...
18809	20.787140838	2806:1040:ffff:3::e	2806:104e:18:ad45:a...	DNS	260 Standard query response 0x6f78 A images-na.ssl-images-amazon.com CNAME m.media-ama...
18810	20.787795454	2806:104e:18:ad45:a...	2806:1040:ffff:3::e	DNS	118 Standard query 0xcb84 HTTPS media.amazon.map.fastly.net OPT

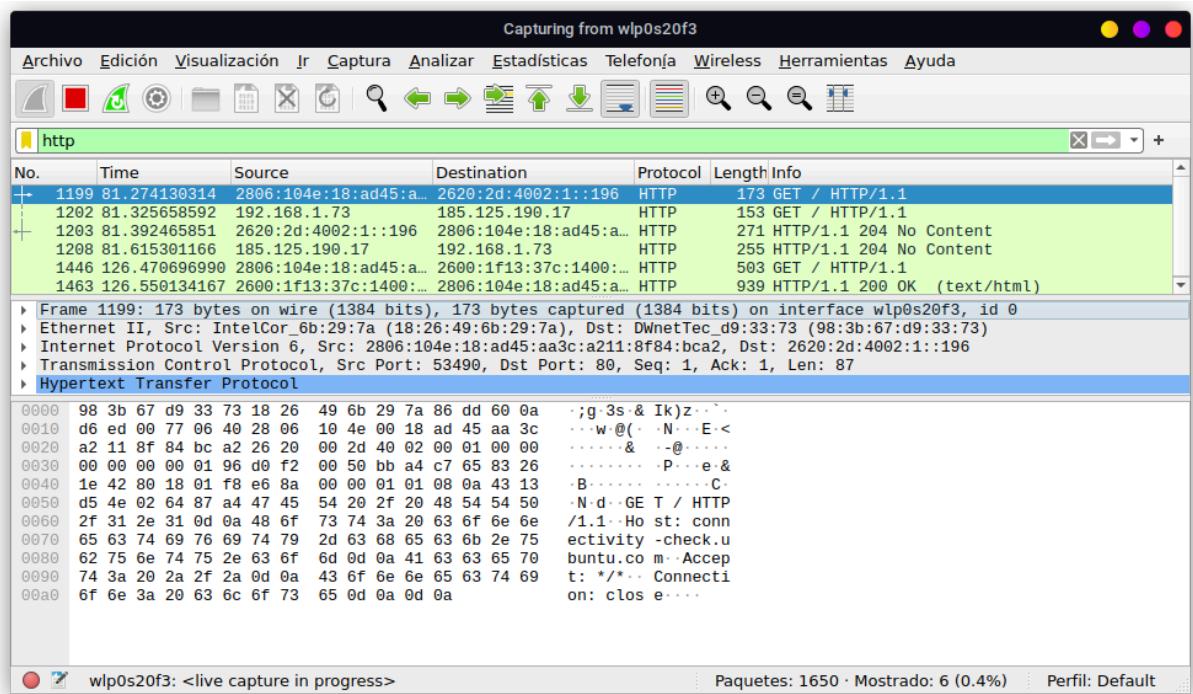
```
> Frame 10475: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface wlp0s20f3, id 0
> Ethernet II, Src: DpNetTec_d9:33:73 (98:3b:67:d9:33:73), Dst: IntelCor_6b:29:7a (18:26:49:60:29:7a)
> Internet Protocol Version 6, Src: 2806:1040:ffff:3::e, Dst: 2806:104e:18:ad45:aa3c:a211:8f84:bca2
> User Datagram Protocol, Src Port: 41402, Dst Port: 53
> Domain Name System (query)
```

```
0000 98 3b 67 d9 33 73 18 26 49 6b 29 7a 86 dd 60 02 -g:3s-&IKz:-
0010 e6 77 00 36 11 40 28 06 10 4e 00 18 ad 45 aa 3c ..w:6 @(. N- E<
0020 a2 11 8f 84 bc a2 06 10 40 ff ff 00 03 00 00 .....( @.....
0030 00 00 00 00 00 0e a1 ba 00 35 00 36 b6 c5 7a bd .....5 6 .z-
0040 01 00 00 01 00 00 00 00 01 03 77 77 00 61 .....www.a...
0050 6d 61 7a 6f 6e 03 63 6f 6d 02 6d 78 00 00 1c 00 mazon.co m:mx...
0060 01 00 00 29 05 ac 00 00 00 00 00 00 00 00 00 00 ..) .....
```

Paquetes: 27028 - Mostrado: 124 (0.5%) Perdido: 0 (0.0%) Perfil: Default



Accedemos a la página <http://neverssl.com> y aplicamos el filtro http.



Nmap

```
sebasdev@sebasdev-Latitude-3410:~$ sudo apt install nmap
[sudo] contraseña para sebasdev:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
nmap ya está en su versión más reciente (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1)
.
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  python-pkg-resources python-setuptools python3-distlib python3-filelock
  python3-platformdirs python3-wheel-whl
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 132 no actualizados.
sebasdev@sebasdev-Latitude-3410:~$
```

Realizamos un escaneo general de toda la red con sudo nmap 192.168.1.0/24 -v. Las partes que considero importante del escaneo son las siguiente:

```
Discovered open port 53/tcp on 192.168.1.254
Discovered open port 3306/tcp on 192.168.1.69
Discovered open port 80/tcp on 192.168.1.254
Discovered open port 9009/tcp on 192.168.1.69
Completed SYN Stealth Scan against 192.168.1.254 in 6.90s (1 host left)
Completed SYN Stealth Scan at 11:14, 6.93s elapsed (2000 total ports)
```

```
Nmap scan report for 192.168.1.69
Host is up (0.0050s latency).
Not shown: 998 filtered ports
PORT STATE SERVICE
3306/tcp open mysql
9009/tcp open pichat
MAC Address: 74:56:3C:5D:6E:B3 (Unknown)
Nmap scan report for _gateway (192.168.1.254)
Host is up (0.0028s latency).
Not shown: 990 filtered ports
PORT STATE SERVICE
53/tcp open domain
80/tcp open http
113/tcp closed ident
256/tcp closed fw1-secureremote
554/tcp closed rtsp
993/tcp closed imaps
1720/tcp closed h323q931
1723/tcp closed pptp
3306/tcp closed mysql
5060/tcp closed sip
MAC Address: 98:3B:67:D9:33:73 (Unknown)
Initiating SYN Stealth Scan at 11:14
Scanning sebasdev-Latitude-3410 (192.168.1.73) [1000 ports]
Discovered open port 80/tcp on 192.168.1.73
Completed SYN Stealth Scan at 11:14, 0.03s elapsed (1000 total ports)
Nmap scan report for sebasdev-Latitude-3410 (192.168.1.73)
Host is up (0.0000090s latency).
Not shown: 999 closed ports
PORT STATE SERVICE
80/tcp open http
Read data files from: /usr/bin/../share/nmap
Nmap done: 256 IP addresses (3 hosts up) scanned in 14.73 seconds
Raw packets sent: 5504 (234.048KB) | Rcvd: 2023 (84.936KB)
```

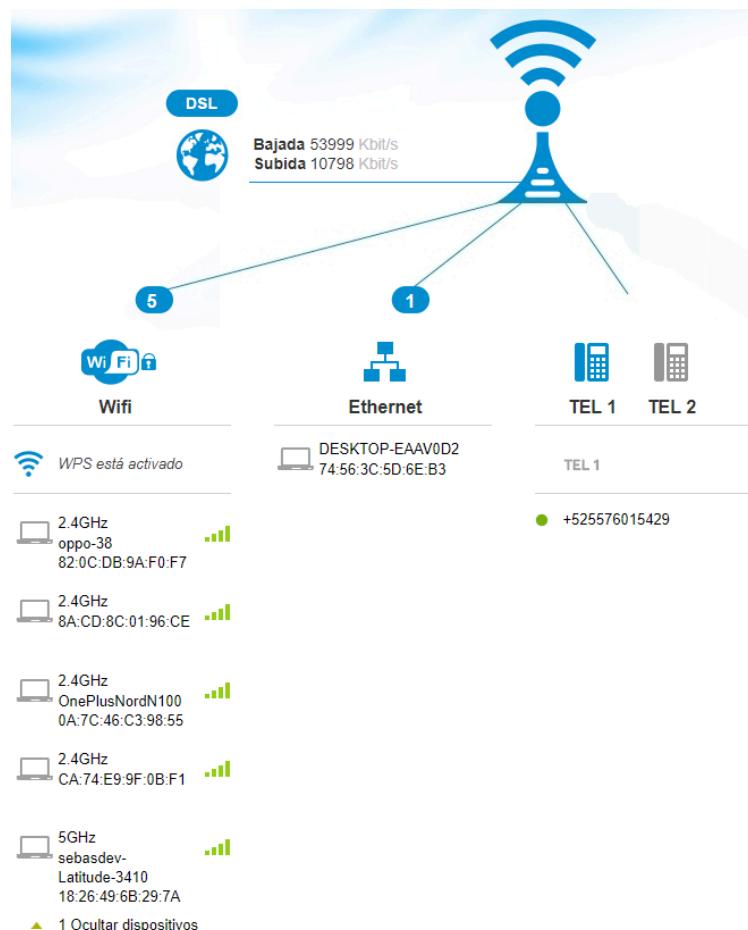
Ahora realizamos el escaneo rápido con el siguiente comando:

```
sudo nmap -sn 192.168.1.0/24 -R
```

El resultado fue el siguiente:

```
sebasdev@sebasdev-Latitude-3410:~$ sudo nmap -sn 192.168.1.0/24 -R
Starting Nmap 7.80 ( https://nmap.org ) at 2024-08-24 12:24 CST
Nmap scan report for 192.168.1.67
Host is up (0.079s latency).
MAC Address: 82:0C:DB:9A:F0:F7 (Unknown)
Nmap scan report for 192.168.1.69
Host is up (0.0088s latency).
MAC Address: 74:56:3C:5D:6E:B3 (Unknown)
Nmap scan report for _gateway (192.168.1.254)
Host is up (0.0042s latency).
MAC Address: 98:3B:67:D9:33:73 (Unknown)
Nmap scan report for sebasdev-Latitude-3410 (192.168.1.73)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 7.72 seconds
sebasdev@sebasdev-Latitude-3410:~$
```

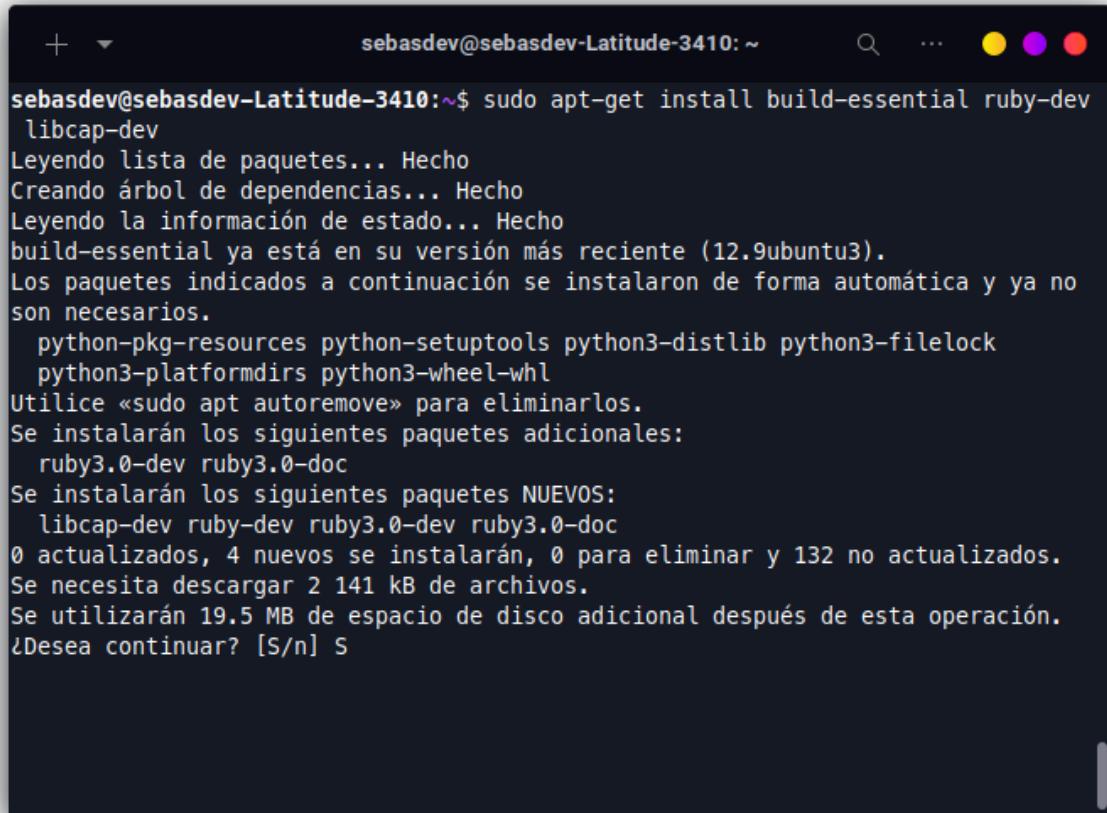
Quiero hacer notar que esta información me recuerda a la que se muestra en la configuración del modem:



Bettercap

Instalamos ruby con el siguiente comando:

```
sudo apt-get install build-essential ruby-dev libcap-dev
```



```
sebasdev@sebasdev-Latitude-3410:~$ sudo apt-get install build-essential ruby-dev libcap-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
build-essential ya está en su versión más reciente (12.9ubuntu3).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  python-pkg-resources python-setuptools python3-distlib python3-filelock
  python3-platformdirs python3-wheel-whl
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  ruby3.0-dev ruby3.0-doc
Se instalarán los siguientes paquetes NUEVOS:
  libcap-dev ruby-dev ruby3.0-dev ruby3.0-doc
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 132 no actualizados.
Se necesita descargar 2 141 kB de archivos.
Se utilizarán 19.5 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

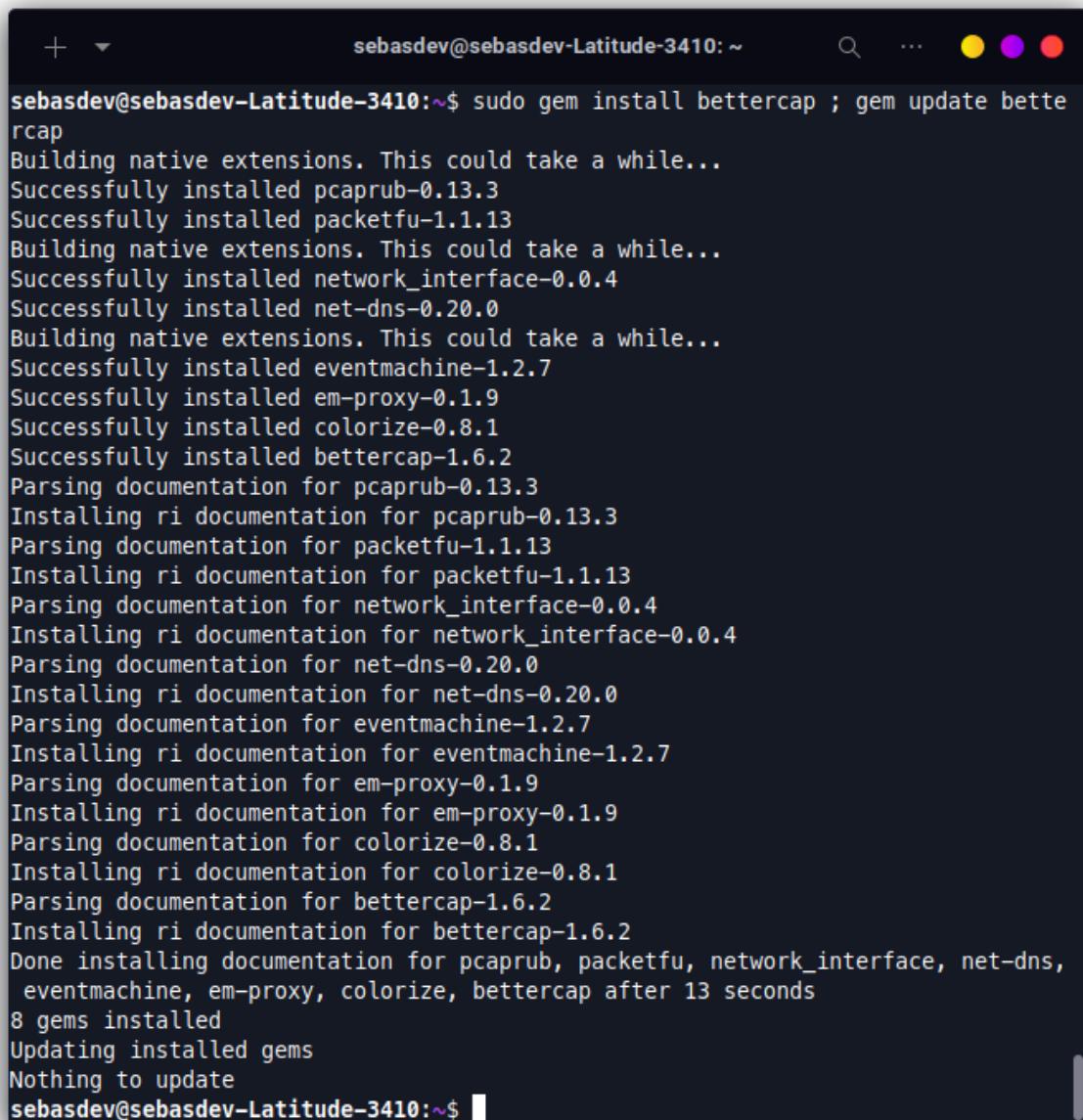
Al ejecutar el comando sudo gem install bettercap ; gem update bettercap se obtuvo el siguiente error:

```
pcaprub.c:11:10: fatal error: pcap.h: No existe el archivo o el directorio
11 | #include <pcap.h>
| ^~~~~~
compilation terminated
```

Investigando un poco encontré un foro donde solucionaban el error, se ejecutó el siguiente comando:

```
apt install -y libpcap-dev
```

Se volvió a ejecutar el comando para la instalación de bettercap, los errores se solucionaron



A screenshot of a terminal window titled "sebasdev@sebasdev-Latitude-3410: ~". The window shows the command "sudo gem install bettercap ; gem update bettercap" being run. The output of the command is displayed, showing the installation of various gems including pcaprub, packetfu, network_interface, net-dns, eventmachine, em-proxy, colorize, and bettercap itself. The process includes building native extensions, parsing documentation, and updating ri documentation for each component. The terminal concludes with a message about documentation being installed for all dependencies after 13 seconds, and a summary of 8 gems installed.

```
sebasdev@sebasdev-Latitude-3410:~$ sudo gem install bettercap ; gem update bettercap
Building native extensions. This could take a while...
Successfully installed pcaprub-0.13.3
Successfully installed packetfu-1.1.13
Building native extensions. This could take a while...
Successfully installed network_interface-0.0.4
Successfully installed net-dns-0.20.0
Building native extensions. This could take a while...
Successfully installed eventmachine-1.2.7
Successfully installed em-proxy-0.1.9
Successfully installed colorize-0.8.1
Successfully installed bettercap-1.6.2
Parsing documentation for pcaprub-0.13.3
Installing ri documentation for pcaprub-0.13.3
Parsing documentation for packetfu-1.1.13
Installing ri documentation for packetfu-1.1.13
Parsing documentation for network_interface-0.0.4
Installing ri documentation for network_interface-0.0.4
Parsing documentation for net-dns-0.20.0
Installing ri documentation for net-dns-0.20.0
Parsing documentation for eventmachine-1.2.7
Installing ri documentation for eventmachine-1.2.7
Parsing documentation for em-proxy-0.1.9
Installing ri documentation for em-proxy-0.1.9
Parsing documentation for colorize-0.8.1
Installing ri documentation for colorize-0.8.1
Parsing documentation for bettercap-1.6.2
Installing ri documentation for bettercap-1.6.2
Done installing documentation for pcaprub, packetfu, network_interface, net-dns,
eventmachine, em-proxy, colorize, bettercap after 13 seconds
8 gems installed
Updating installed gems
Nothing to update
sebasdev@sebasdev-Latitude-3410:~$
```

Al comprobar la instalación de bettercap se obtuvo el siguiente error:

```
<internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:85:in `require'
from <internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:85:in `re
from /var/lib/gems/3.0.0/gems/bettercap-1.6.2/lib/bettercap.rb:24:in <top (required)>
from <internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:85:in `re
from <internal:/usr/lib/ruby/vendor_ruby/rubygems/core_ext/kernel_require.rb>:85:in `re
from /var/lib/gems/3.0.0/gems/bettercap-1.6.2/bin/bettercap:19:in <top (required)>
from /usr/local/bin/bettercap:25:in `load'
from /usr/local/bin/bettercap:25:in <main>
```

Teníamos que haber instalado webrick con el siguiente comando:

sudo gem install webrick

Volvemos a comprobar la instalación de bettercap:

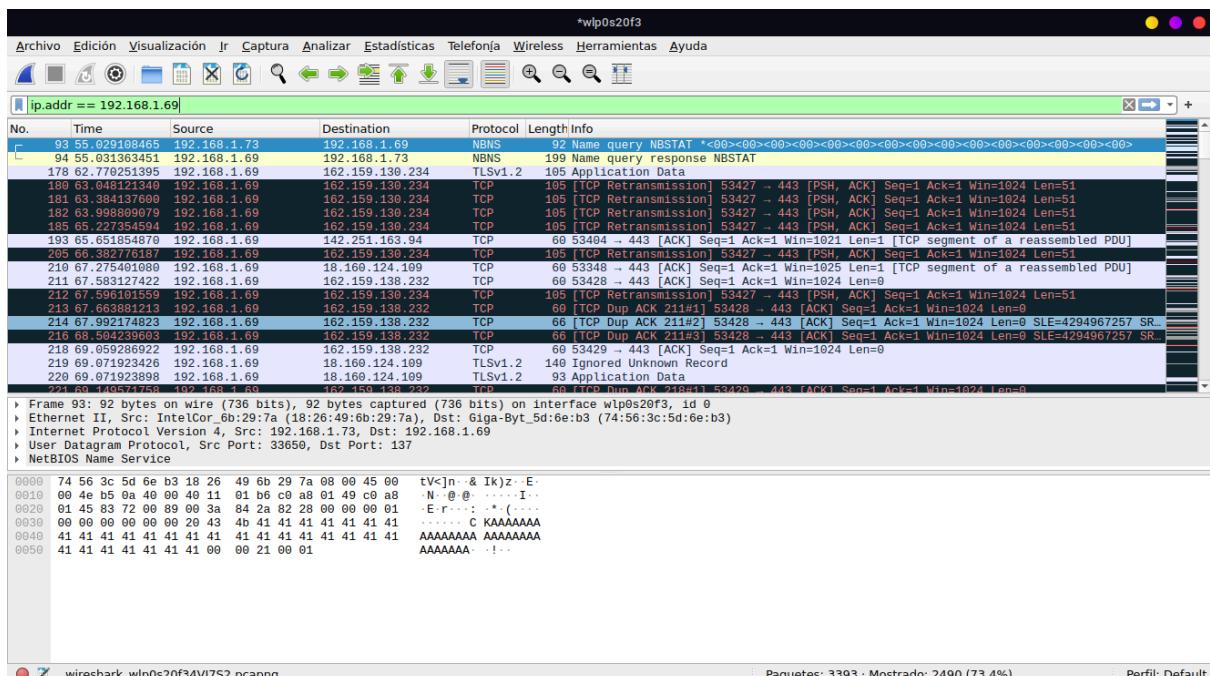
Main in the Middle (MitM)

Consideremos la interfaz de red **wlp0s20f3** junto con alguno de los dispositivos mostrados cuando ejecutamos el comando que realiza un escaneo rápido con nmap, tomemos como víctima el dispositivo con dirección ip **192.168.1.69**, lo incluimos en el siguiente comando:

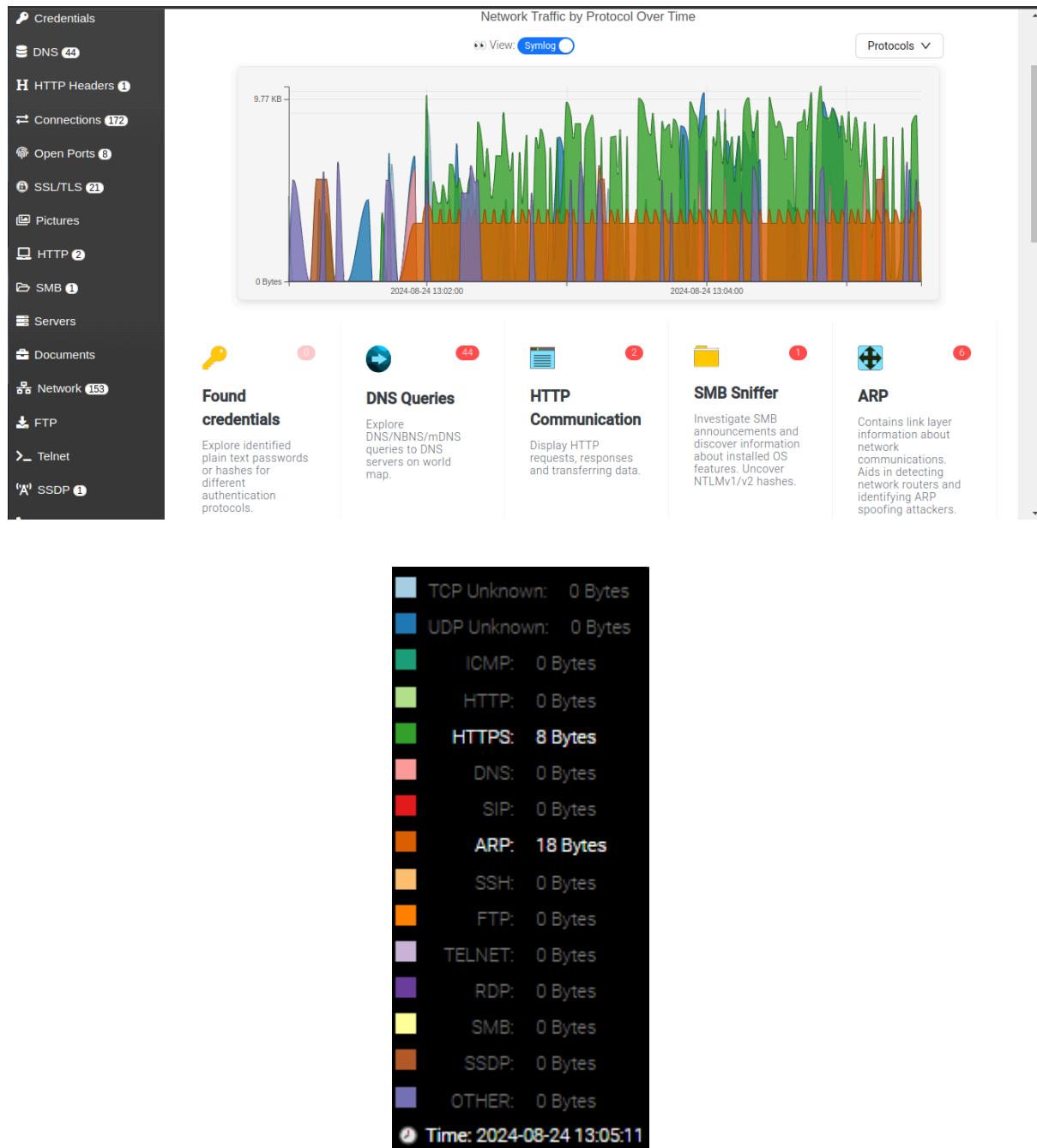
```
sudo bettercap --interface wlp0s20f3 --sniffer --target 192.168.1.69
```

```
[I] Starting [ spoofing: discovery: sniffer: tcp-proxy: udp-proxy: http-proxy: https-proxy: sslstrip: http-server: dns-server: ] ...
[I] [wlp0s20f3] 192.168.1.73 : 18:26:49:6B:29:7A / wlp0s20f3 ( ??? )
[I] [GATEWAY] 192.168.1.254 : 98:3B:67:D9:33:73 ( ??? )
[I] [TARGET] 192.168.1.69 : 74:56:3C:5D:6E:B3 ( ??? )
[I] Found hostname _gateway for address 192.168.1.254
[I] Found NetBIOS name 'DESKTOP-EAAV0D2' for address 192.168.1.69
[DESKTOP-EAAV0D2/192.168.1.69 > 52.112.87.18:https] [HTTPS] https://pub-ent-uswe-10-t.trouter.teams.microsoft.com/
[DESKTOP-EAAV0D2/192.168.1.69 > 3.209.71.109:https] [HTTPS] https://fls-na.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 54.83.83.234:https] [HTTPS] https://f-log-extension.grammarly.io/
[DESKTOP-EAAV0D2/192.168.1.69 > 3.209.71.109:https] [HTTPS] https://fls-na.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 54.83.83.234:https] [HTTPS] https://f-log-extension.grammarly.io/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.116.37:https] [HTTPS] https://completion.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 3.209.71.109:https] [HTTPS] https://fls-na.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 54.83.83.234:https] [HTTPS] https://f-log-extension.grammarly.io/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.116.37:https] [HTTPS] https://completion.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 54.83.83.234:https] [HTTPS] https://f-log-extension.grammarly.io/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.137.64:https] [HTTPS] https://unagi.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 3.209.71.109:https] [HTTPS] https://fls-na.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 54.83.83.234:https] [HTTPS] https://f-log-extension.grammarly.io/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.116.37:https] [HTTPS] https://completion.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.137.64:https] [HTTPS] https://unagi.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 34.233.93.53:https] [HTTPS] https://fls-na.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.215.116.37:https] [HTTPS] https://completion.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.194.59.28:https] [HTTPS] https://es.duolingo.com/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.199.181.222:https] [HTTPS] https://unagi.amazon.com.mx/
[DESKTOP-EAAV0D2/192.168.1.69 > 44.194.59.28:https] [HTTPS] https://es.duolingo.com/
```

En mi computadora víctima comencé a navegar por algunas páginas de internet. Sin embargo, no se mostraron capturados todos los sitios en bettercap o algunos tardaban en aparecer. Considero importante resaltar que mi red se volvió bastante lenta, al grado de que las páginas tardaban mucho en cargar o incluso no lo hacían. Al mismo tiempo hice la captura del tráfico de red con wireshark, aplicando un filtro de dirección ip:

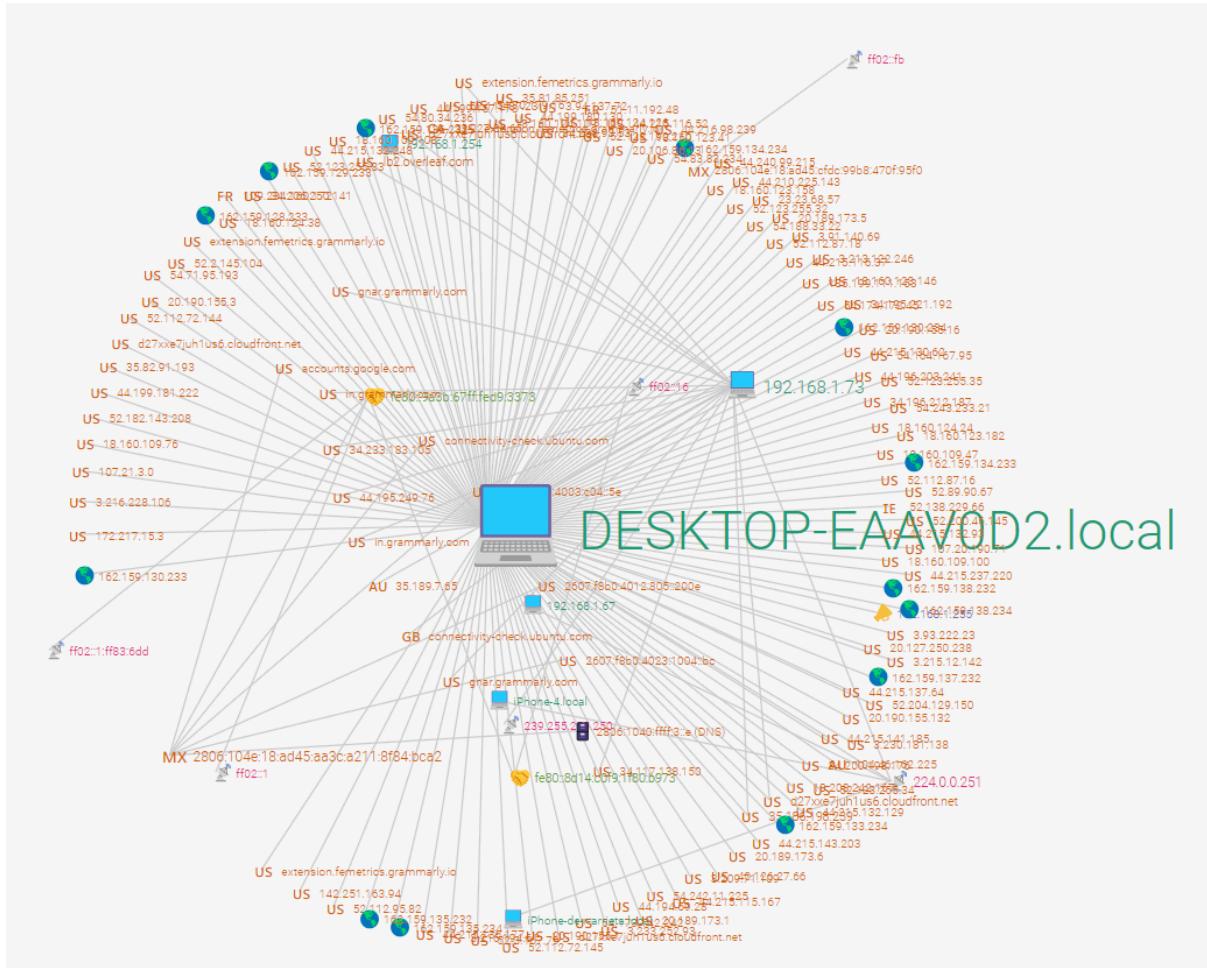


Guardamos la captura de tráfico en un archivo pcapng y lo cargamos en la página [A-Packets](#) y este fue el resultado:



 0 Found credentials <p>Explore identified plain text passwords or hashes for different authentication protocols.</p>	 44 DNS Queries <p>Explore DNS/NBNS/mDNS queries to DNS servers on world map.</p>	 2 HTTP Communication <p>Display HTTP requests, responses and transferring data.</p>	 1 SMB Sniffer <p>Investigate SMB announcements and discover information about installed OS features. Uncover NTLMv1/v2 hashes.</p>	 6 ARP <p>Contains link layer information about network communications. Aids in detecting network routers and identifying ARP spoofing attackers.</p>
 153 Network Map <p>Analyze IP communications between devices and the protocols used. Identify fingerprints such as the operating system and installed software.</p>	 8 Open Ports <p>Identify open TCP ports and their associated fingerprints in the captured traffic.</p>	 21 SSL/TLS <p>Retrieve information about SSL/TLS sessions, including client/server hello messages and certificate chains.</p>	 0 Images <p>View images discovered in HTTP data.</p>	 0 Telnet <p>Show Telnet sessions data.</p>
 0 FTP <p>Show FTP sessions data.</p>	 1 SSDP Announcements <p>Contains announcements of services running on network devices using the SSDP protocol.</p>	 172 Connections <p>Visualize IP connections, displaying endpoints and data volume transfer on a world map.</p>	 1 DNS, DHCP and LDAP Servers <p>Detect DNS, DHCP and LDAP servers from intercepted network traffic.</p>	 6 Ethernet Devices <p>Identify Ethernet devices and detect the used Ethernet broadcast addresses.</p>
 0 WiFi <p>View information about access points, clients, connection requests, and discovered WPA2 handshakes.</p>	 0 SIP <p>Explore details of SIP communications and authentication data.</p>	 0 Documents <p>Found office documents in PDF, MS Word, MS Excel, RTF and other formats.</p>		

Visualizamos el archivo con la opción de Network



- Camilo:

Primero ejecute el comando “sudo dnf install net-tools” para tener las herramientas necesarias

```
camilo@wowi:~$ sudo dnf install net-tools
[sudo] password for camilo:
Last metadata expiration check: 0:41:08 ago on Fri 23 Aug 2024 05:35:00 PM CST.
Package net-tools-2.0-0.69.20160912git.fc40.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
camilo@wowi:~$
```

Después el comando “sudo ifconfig -a” mostrando la información de las interfaces de red en mi sistema. Se mostraron 3 diferentes, pero la más importante es “wlo1”, mi interfaz de red inalámbrica, por lo tanto es la que usaré

```

Complete:
camilo@wowi:~$ sudo ifconfig -a
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
        ether 02:42:64:d7:5e:5c txqueuelen 0 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 466 bytes 457667 (446.9 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 466 bytes 457667 (446.9 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.241 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 2806:2a0:416:9534::118f prefixlen 128 scopeid 0x0<global>
        inet6 fe80::8c75:8f24:ee02:1fd7 prefixlen 64 scopeid 0x20<link>
          ether 2e:34:ba:c4:0e:65 txqueuelen 1000 (Ethernet)
            RX packets 1841288 bytes 2672004565 (2.4 GiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 331192 bytes 71741366 (68.4 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

camilo@wowi:~$ 

```

Luego ejecute el comando “sudo dnf check-update && sudo dnf install wireshark” para poder instalar la herramienta Wireshark

```

Installed:
  bcg729-1.1.1-9.fc40.x86_64
  qt6-qt5compat-6.7.2-1.fc40.x86_64
  qt6-qtquicktimeline-6.7.2-1.fc40.x86_64
  wireshark-1:4.2.6-1.fc40.x86_64

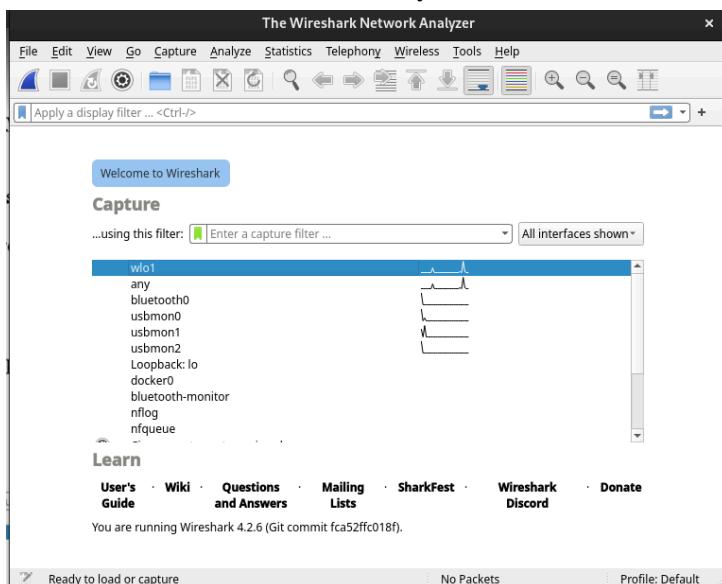
  libsmi-0.4.8-39.fc40.x86_64
  qt6-qtmultimedia-6.7.2-1.fc40.x86_64
  qt6-qtshadertools-6.7.2-1.fc40.x86_64
  wireshark-cli-1:4.2.6-1.fc40.x86_64

minizip-ng-compat-3.0.10-7.fc40.x86_64
qt6-qtquick3d-6.7.2-3.fc40.x86_64
spandsp-0.0.6-18.fc40.x86_64

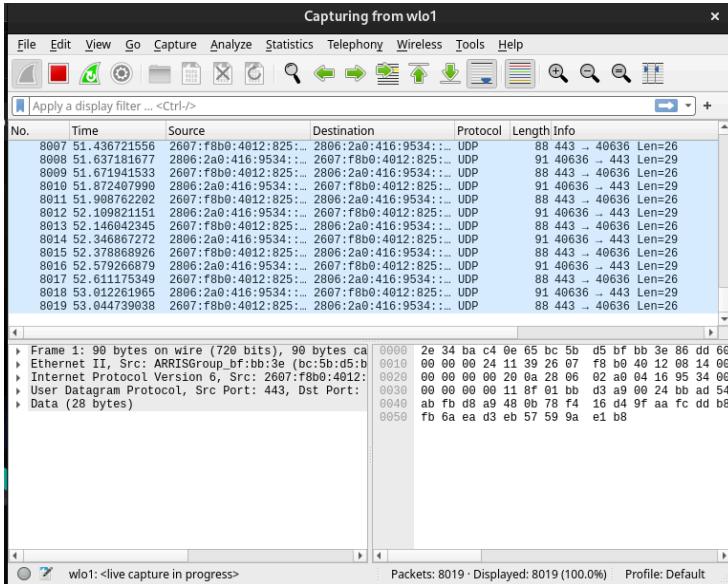
Complete!
camilo@wowi:~$ 

```

Posteriormente abrir Wireshark y seleccione la interfaz “wlo1”



Me puse a visitar las siguientes páginas: GitHub, Google Classroom y YouTube



Despues entre a neverssl.com y en Wireshark filtre por http, lo cual me saco muchos datos pero entre todos un text/html de 131 líneas (este caso me costo algo de trabajo, primero porque se mueve muy rápido lo de que aparece en Wireshark, luego no sabia bien como filtrar y por ultimo no sabia bien la estructura del paquete, entonces tuve que ver todo en búsqueda del html)

Luego ejecute el comando “sudo dnf install nmap” para instalar nmap

```
Running transaction
Preparing : 
Installing  : nmap-4:7.92-2.fc40.x86_64
Running scriptlet: nmap-4:7.92-2.fc40.x86_64

Installed:
nmap-4:7.92-2.fc40.x86_64

Complete!
camilo@wowl:~$
```

Posteriormente ejecute el comando “nmap -sn 192.168.0.0/24 -R”, encontrado 54 dispositivos conectados a la red, los cuales son mi computadora, el router y 3 más (creo que

uno es mi iPad, mi teléfono y el teléfono de mi papá o mamá). Entonces el dispositivo a interceptar será 192.168.0.90 que es mi teléfono

```
camilo@wofi:~$ nmap -sn 192.168.0.0/24 -R
Starting Nmap 7.92 ( https://nmap.org ) at 2024-08-23 20:24 CST
Nmap scan report for _gateway (192.168.0.1)
Host is up (0.0073s latency).
Nmap scan report for 192.168.0.90
Host is up (0.043s latency).
Nmap scan report for 192.168.0.99
Host is up (0.034s latency).
Nmap scan report for 192.168.0.116
Host is up (0.012s latency).
Nmap scan report for wofi (192.168.0.241)
Host is up (0.000059s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 19.54 seconds
camilo@wofi:~$
```

Luego ejecute el comando “sudo apt-get install build-essential ruby-dev libpcap-dev” para instalar Ruby (para este paso me pase a una máquina virtual con Debian, debido a que intente usar Bettercap en Fedora pero no logré que funcionara como quería)

```
camilo@debian:~$ sudo apt-get install build-essential ruby-dev libpcap-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9).
ruby-dev is already the newest version (1:3.1).
libpcap-dev is already the newest version (1.10.3-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
camilo@debian:~$
```

Despues ejecute el comando “sudo gem install bettercap ; gem update bettercap” para instalar Bettercap

```
Installing ri documentation for bettercap 1.6.2
Done installing documentation for hitimes, timers, nio4r, celluloid, celluloid-io, rubydns,
pcaprub, packetfu, network_interface, net-dns, eventmachine, em-proxy, colorize, bettercap
after 19 seconds
14 gems installed
Updating installed gems
Nothing to update
camilo@debian:~$
```

Posteriormente ejecute los comandos “sudo bettercap --check-updates” y “sudo bettercap --version”

```
camilo@debian:~$ sudo bettercap --check-updates
[!] Checking for updates ...
[!] You are running the latest version.

camilo@debian:~$
```

Después ejecute el comando “sudo bettercap --interface enp0s3 --sniffer --target 192.168.0.90”, solo que ahora es otro interfaz de red, todo ya que ahora estoy en Debian (otra cosas que pasó es que al intentar entrar a páginas en mi teléfono fue algo lento o no cargaban las páginas y en la terminal solo salían cosas de DHCP)

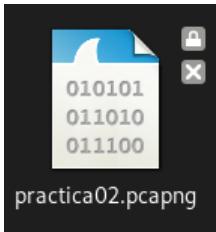
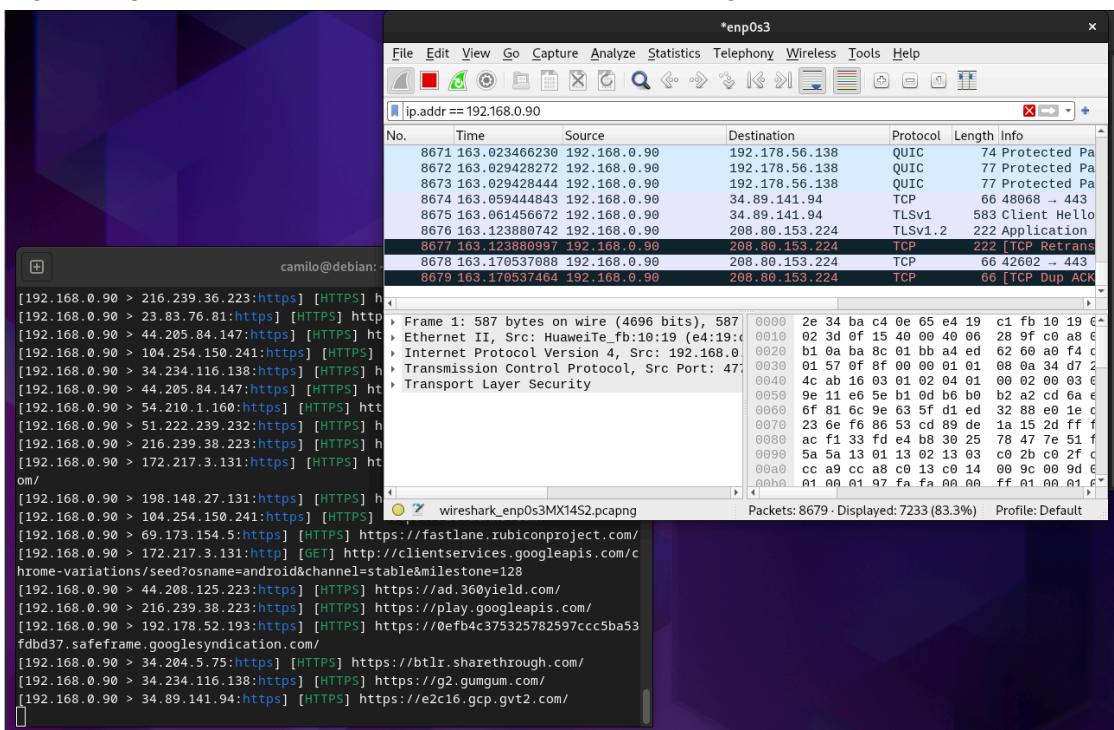
```
camilo@debian: ~ $ sudo ifconfig -a
[sudo] password for camilo:
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
          inet 192.168.0.209  netmask 255.255.255.0  broadcast 192.168.0.255
              inet6 fe80::a00:27ff:fe8c:f456  prefixlen 64  scopeid 0x20<link>
                ether 08:00:27:c8:f4:56  txqueuelen 1000  (Ethernet)
                  RX packets 148  bytes 30049 (29.3 KiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 125  bytes 15822 (15.4 KiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
          inet 127.0.0.1  netmask 255.0.0.0
              inet6 ::1  prefixlen 128  scopeid 0x10<host>
                loop  txqueuelen 1000  (Local Loopback)
                  RX packets 28  bytes 2832 (2.7 KiB)
                  RX errors 0  dropped 0  overruns 0  frame 0
                  TX packets 28  bytes 2832 (2.7 KiB)
                  TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

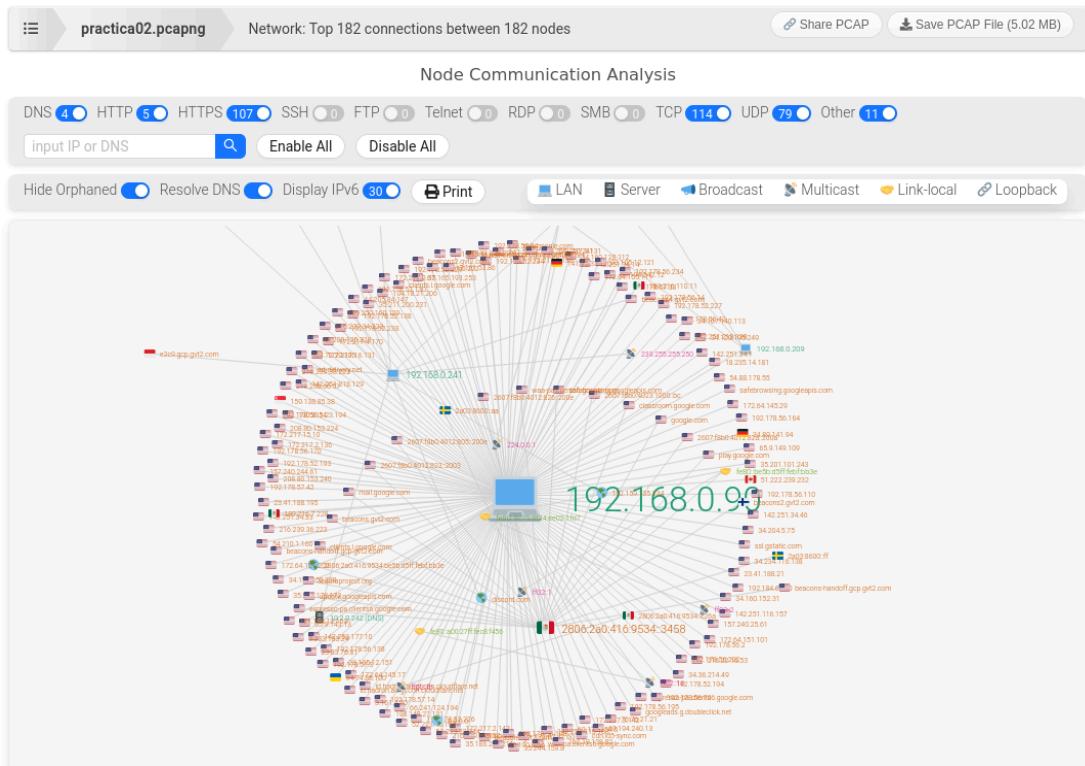
```
camilo@debian: ~ $ sudo nmap -sn 192.168.0.0/24 -R
Starting Nmap 7.93 ( https://nmap.org ) at 2024-08-24 17:29 CST
Nmap scan report for 192.168.0.1
Host is up (0.0059s latency).
MAC Address: BC:5B:D5:BF:BB:3E (Arris Group)
Nmap scan report for 192.168.0.90
Host is up (2.5s latency).
MAC Address: E4:19:C1:FB:10:19 (Huawei Technologies)
Nmap scan report for 192.168.0.103
Host is up (2.7s latency).
MAC Address: C8:3A:6B:F9:22:3F (Roku)
Nmap scan report for 192.168.0.200
Host is up (0.71s latency).
MAC Address: 94:44:44:0C:C2:94 (LG Innotek)
Nmap scan report for 192.168.0.241
Host is up (0.00076s latency).
MAC Address: 2E:34:BA:C4:0E:65 (Unknown)
Nmap scan report for 192.168.0.209
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 26.62 seconds
camilo@debian: ~ $
```

```
[192.168.0.90 > 3.233.183.24:https] [HTTPS] https://tlx.3lift.com/
[192.168.0.90 > 104.254.150.241:https] [HTTPS] https://ib.adnxs.com/
[192.168.0.90 > 192.178.56.195:https] [HTTPS] https://id.google.com/
[192.168.0.90 > 54.88.178.55:https] [HTTPS] https://rp.liadm.com/
[192.168.0.90 > 216.239.34.223:https] [HTTPS] https://play.googleapis.com/
[192.168.0.90 > 208.80.153.224:https] [HTTPS] https://es.wikipedia.org/
[192.168.0.90 > 192.178.56.170:https] [HTTPS] https://people-pa.googleapis.com/
[192.168.0.90 > 66.241.124.194:https] [HTTPS] https://api.pokedoku.com/
[192.168.0.90 > 3.233.183.24:https] [HTTPS] https://tlx.3lift.com/
[192.168.0.90 > 192.178.52.234:https] [HTTPS] https://www.googleapis.com/
[192.168.0.90 > 216.239.32.116:https] [HTTPS] https://beacons4.gvt2.com/
[192.168.0.90 > 66.241.124.194:https] [HTTPS] https://api.pokedoku.com/
[192.168.0.90 > 192.178.56.195:https] [HTTPS] https://id.google.com/
[192.168.0.90 > 192.178.56.170:https] [HTTPS] https://people-pa.googleapis.com/
[192.168.0.90 > 208.80.153.224:https] [HTTPS] https://es.wikipedia.org/
[192.168.0.90 > 66.241.124.194:https] [HTTPS] https://api.pokedoku.com/
[192.168.0.90 > 172.217.3.131:https] [HTTPS] https://clientservices.googleapis.c
```

Posteriormente visite algunos sitios con mi celular y puse Wireshark para ver qué paquetes llegaban, guarde los paquetes capturados en un .pcapng



Por ultimo entre a <https://apackets.com/> para subir el archivo generado y usar la opción Network para ver una representación gráfica



- Sara Lorena :

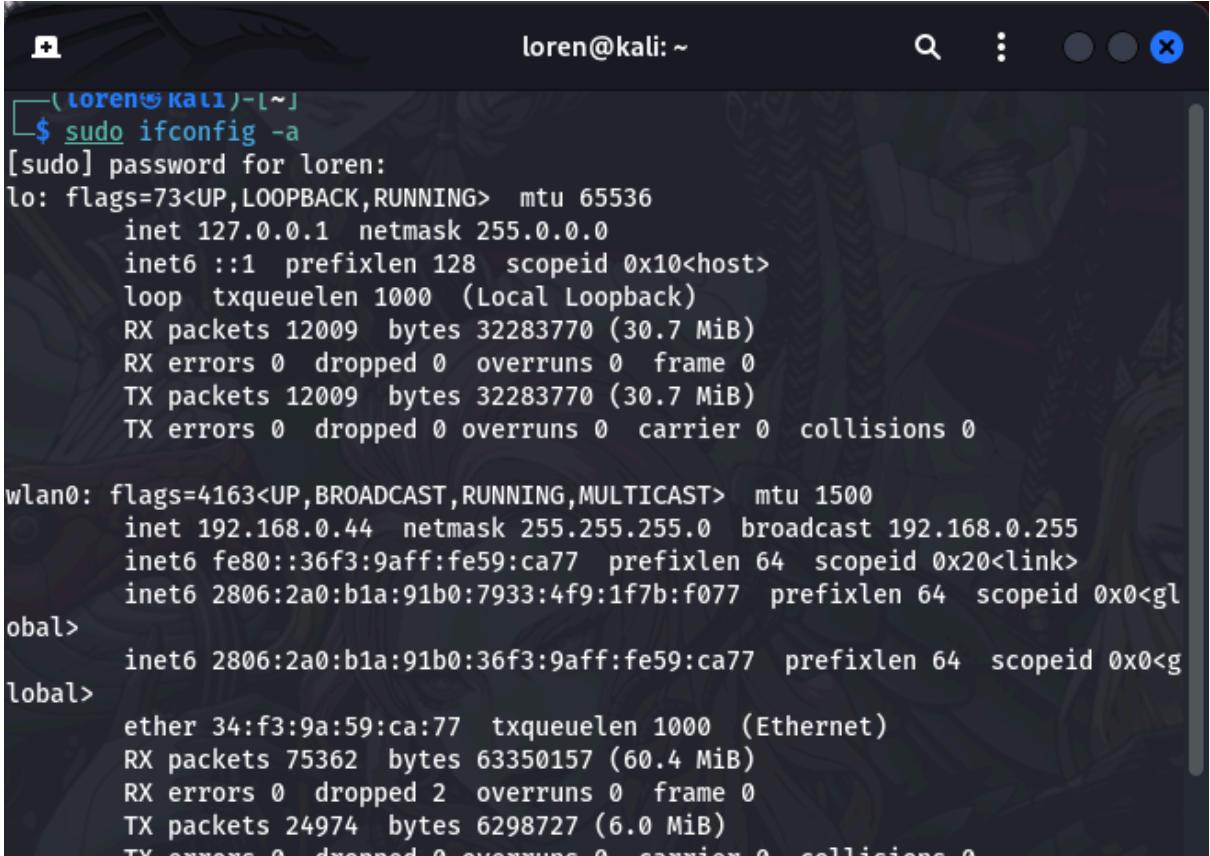
```

loren@kali: ~
[loren@kali]-[~]
$ sudo apt install net-tools
[sudo] password for loren:
net-tools is already the newest version (2.10-1.1).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 943

[loren@kali]-[~]
$ 

```

Desconozco cuando la instalé pero ya tenía instalada net-tools por lo cual este primer paso acabó en segundos.



```
loren@kali: ~
$ sudo ifconfig -a
[sudo] password for loren:
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 12009 bytes 32283770 (30.7 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 12009 bytes 32283770 (30.7 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

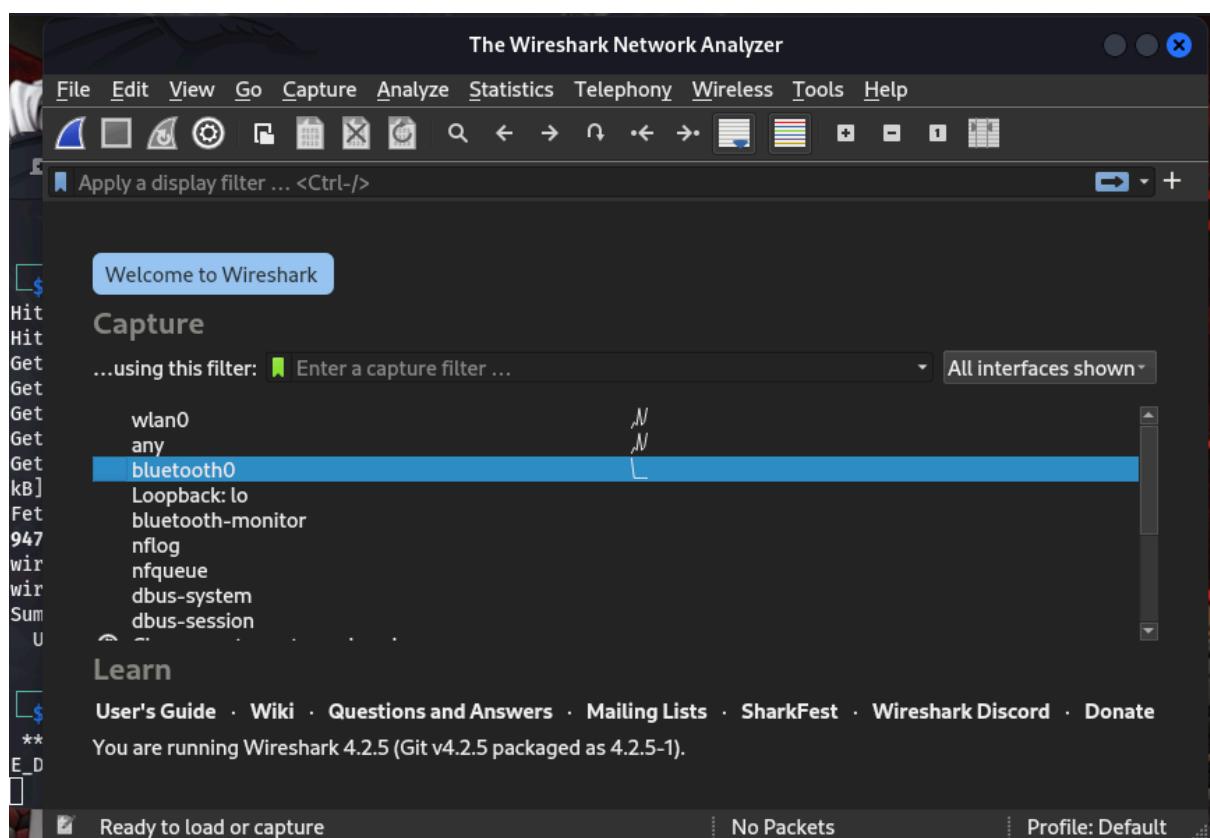
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.44 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::36f3:9aff:fe59:ca77 prefixlen 64 scopeid 0x20<link>
        inet6 2806:2a0:b1a:91b0:7933:4f9:1f7b:f077 prefixlen 64 scopeid 0x0<global>
        inet6 2806:2a0:b1a:91b0:36f3:9aff:fe59:ca77 prefixlen 64 scopeid 0x0<global>
            ether 34:f3:9a:59:ca:77 txqueuelen 1000 (Ethernet)
            RX packets 75362 bytes 63350157 (60.4 MiB)
            RX errors 0 dropped 2 overruns 0 frame 0
            TX packets 24974 bytes 6298727 (6.0 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

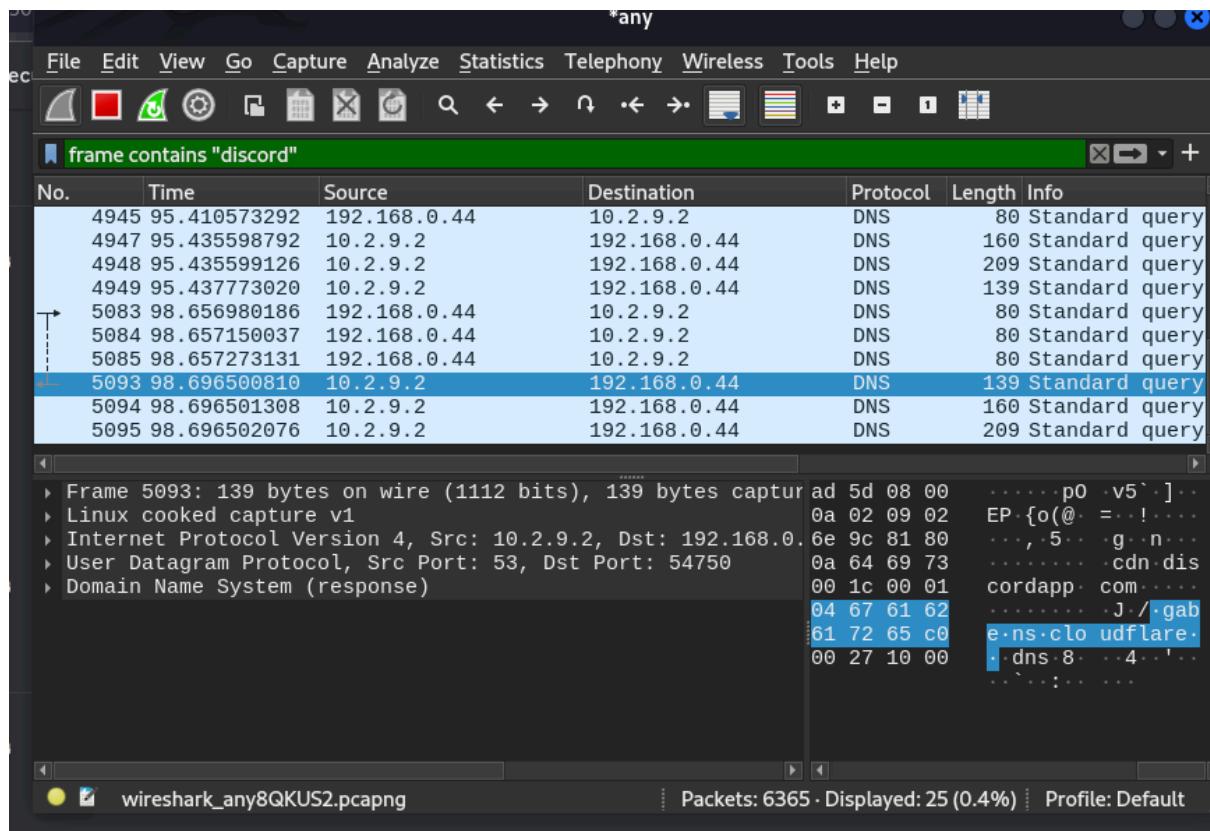
Igual, no hubo ningún problema a la hora de realizar el comando ifconfig -a y pude ver las interfaces disponibles en mi maquina. En este punto la práctica menciona que se usará el puerto eth0, cosa que no tengo. Esto me preocupó porque no sabía qué interfaz usar ni entendía a que se refería la información que se nos estaba dando. Tras una búsqueda en google determine que la interfaz que debía usar es wlan0, que es mi conexión inalámbrica.

```
loren@kali: ~
RX errors 0 dropped 2 overruns 0 frame 0
TX packets 24974 bytes 6298727 (6.0 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(loren@kali)-[~]
$ sudo apt update && sudo apt install wireshark
Hit:2 https://dl.google.com/linux/chrome/deb stable InRelease
Hit:3 https://packages.microsoft.com/repos/code stable InRelease
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Packages [19.7 MB]
Get:5 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.3 MB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [194 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [863 kB]
Fetched 68.1 MB in 22s (3027 kB/s)
947 packages can be upgraded. Run 'apt list --upgradable' to see them.
wireshark is already the newest version (4.2.5-1).
wireshark set to manually installed.
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 947

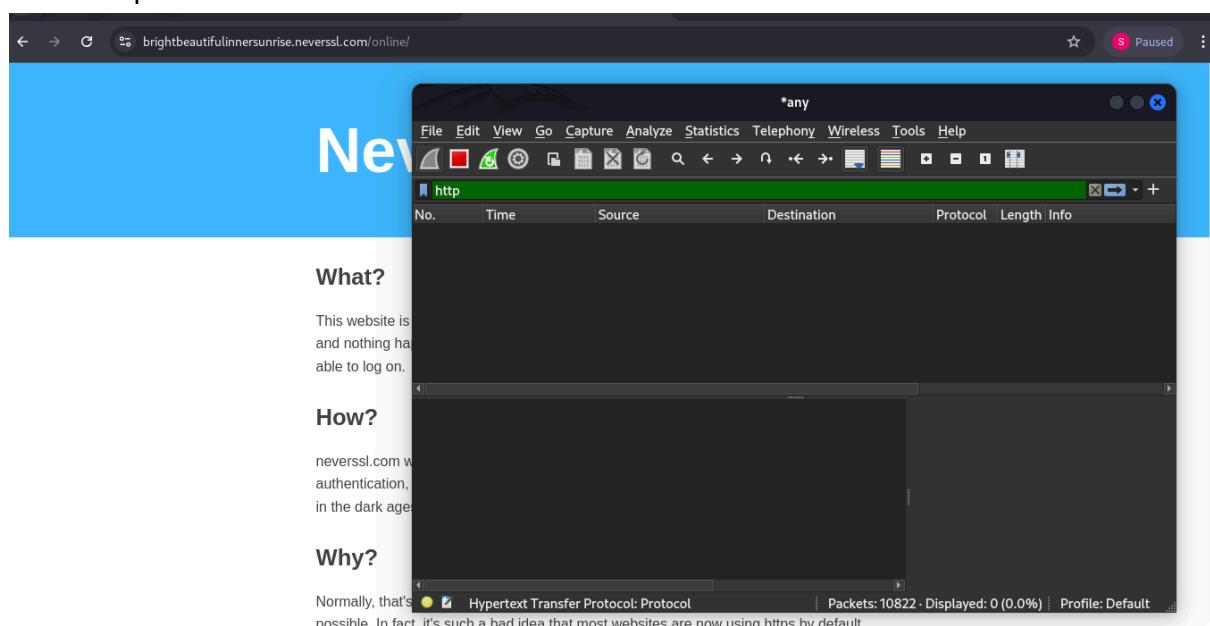
(loren@kali)-[~]
```



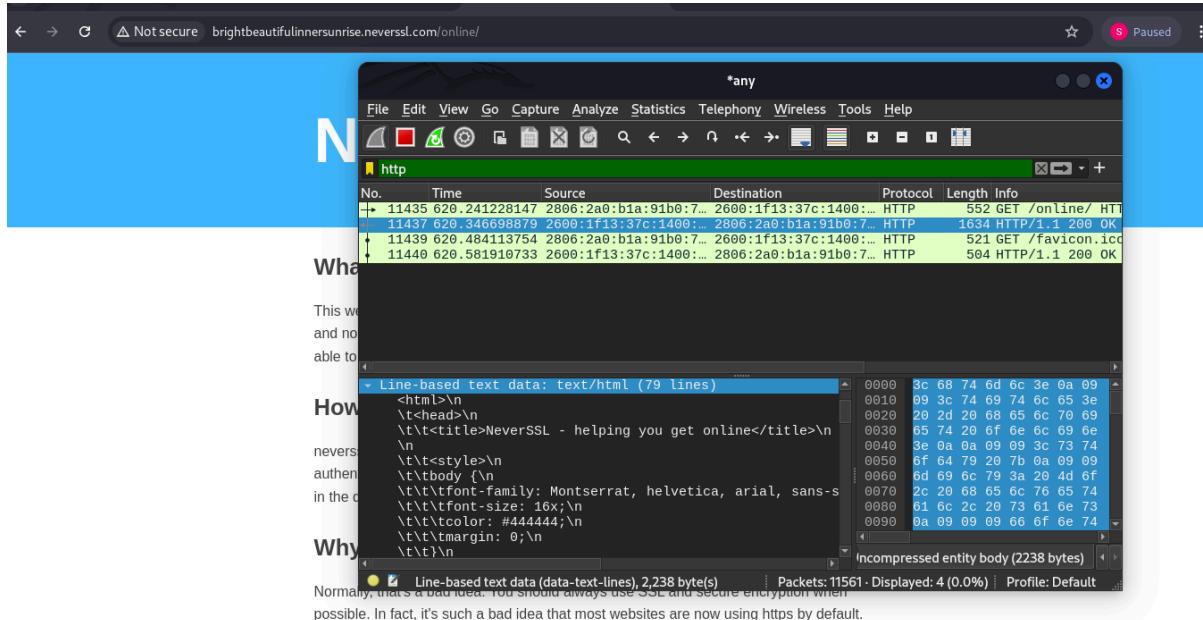
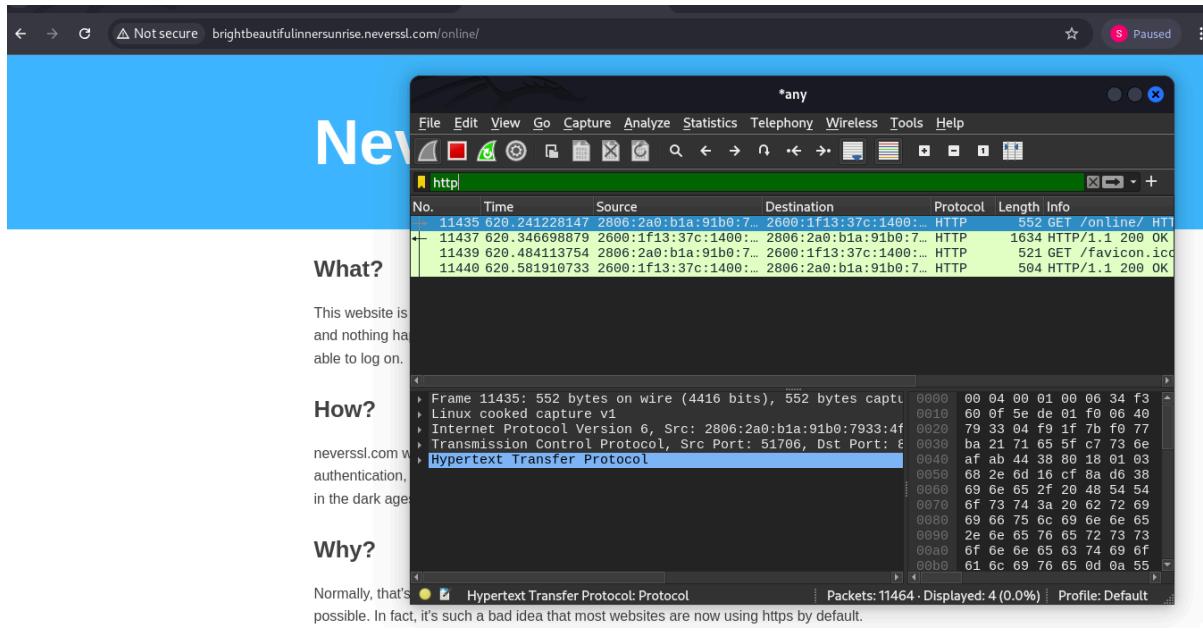


Al darle doble click a mi interfaz e iniciar la captura de paquetes fue sumamente abrumador la velocidad y la cantidad de cosas que aprecian en pantalla, sobre todo los diferentes colores de los cuales desconocía el significado.

Al hacer el ejercicio decidí visitar discord, github, google classroom y youtube. Al inicio no me funcionaba la búsqueda, me decía que no era una cadena válida y aparecía en rojo toda la sección de filtrar. Generalmente cuando eso pasa en otros lados es por la falta de comillas por lo que decidí ponerle y pareció funcionar al mostrarme el resultado de la búsqueda.



Una vez en la página me di cuenta que no me aparecía algo en la aplicación.



Cuando me di cuenta que estaba en la página con el protocolo https y no http. Esto se debe probablemente a como buscar la página, ya que en teoría al buscarla como www.neverssl.com debería enviar el primer paquete como http. Aquí pude ver todo el archivo html de la página.

```
loren@kali:~
```

```
(loren@kali)-[~]
$ sudo apt install nmap
[Upgrading:
  nmap nmap-common
  Downloads
Summary:
  Upgrading: 2, Installing: 0, Removing: 0, Not Upgrading: 945
  Download size: 6170 kB
  Freed space: 3072 B
Continue? [Y/n] Y
Get:1 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.94+git20230807.3be01efb1+dfsg-2+kali3 [1929 kB]
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.94+git20230807.3be01efb1+dfsg-2+kali3 [4241 kB]
Fetched 6170 kB in 2s (4096 kB/s)
(Reading database ... 425971 files and directories currently installed.)
Preparing to unpack .../nmap_7.94+git20230807.3be01efb1+dfsg-2+kali3_amd64.deb .
..
Unpacking nmap (7.94+git20230807.3be01efb1+dfsg-2+kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2+b1) ...
Preparing to unpack .../nmap-common_7.94+git20230807.3be01efb1+dfsg-2+kali3_all.deb ...
Unpacking nmap-common (7.94+git20230807.3be01efb1+dfsg-2+kali3) over (7.94+git20230807.3be01efb1+dfsg-2+kali2+b1)
```

```
loren@kali:~
```

```
(loren@kali)-[~]
$ sudo nmap -sn 192.168.0.0/24 -R
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-25 19:33 CST
Nmap scan report for 192.168.0.1
Host is up (0.0071s latency).
MAC Address: 70:4F:B8:76:35:60 (Arris Group)
Nmap scan report for 192.168.0.3
Host is up (0.22s latency).
MAC Address: 68:9A:87:D1:3F:EF (Amazon Technologies)
Nmap scan report for 192.168.0.5
Host is up (0.22s latency).
MAC Address: 64:20:0C:D4:5D:91 (Apple)
Nmap scan report for 192.168.0.8
Host is up (0.22s latency).
MAC Address: EA:48:B8:07:6A:6F (Unknown)
Nmap scan report for 192.168.0.9
Host is up (0.22s latency).
MAC Address: 5C:41:5A:62:13:79 (Amazon.com)
Nmap scan report for 192.168.0.10
Host is up (0.0070s latency).
MAC Address: 80:3F:5D:DA:20:1C (Winstars Technology)
Nmap scan report for 192.168.0.19
Host is up (0.22s latency)
```

Esto me dio como resultado diversos dispositivos los cuales solo conozco unos cuantos. Desconozco cuales son los demás, considerando que en el momento en que hice la tarea toda mi familia estaba en la casa conectada a la misma red. Pero para la siguiente parte usaré mi otra computadora.

```
[+] Nmap scan report for 192.168.0.42
Host is up (0.16s latency).
MAC Address: 74:4C:A1:9F:1D:2F (Liteon Technology)
```

```
[(loren㉿kali)-[~]
$ sudo apt-get install build-essential ruby-dev libcap-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10).
build-essential set to manually installed.
ruby-dev is already the newest version (1:3.1+nmu1).
ruby-dev set to manually installed.
The following NEW packages will be installed:
  libcap-dev
0 upgraded, 1 newly installed, 0 to remove and 945 not upgraded.
Need to get 527 kB of archives.
After this operation, 2779 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kali.download/kali kali-rolling/main amd64 libcap-dev amd64 1:2.66-5 [527 kB]
Fetched 527 kB in 2s (314 kB/s)
Selecting previously unselected package libcap-dev:amd64.
(Reading database ... 425970 files and directories currently installed.)
Preparing to unpack .../libcap-dev_1%3a2.66-5_amd64.deb ...
Unpacking libcap-dev:amd64 (1:2.66-5) ...
Setting up libcap-dev:amd64 (1:2.66-5) ...
Processing triggers for man-db (2.12.1-1) ...
```

```
[(loren㉿kali)-[~]
$ sudo gem install bettercap ; gem update bettercap
Building native extensions. This could take a while...
Successfully installed pcaprub-0.13.3
Successfully installed packetfu-1.1.13
Building native extensions. This could take a while...
Successfully installed network_interface-0.0.4
Successfully installed net-dns-0.20.0
Building native extensions. This could take a while...
Successfully installed eventmachine-1.2.7
Successfully installed em-proxy-0.1.9
Successfully installed colorize-0.8.1
Successfully installed bettercap-1.6.2
Parsing documentation for pcaprub-0.13.3
Installing ri documentation for pcaprub-0.13.3
```

```
(loren@kali)-[~]
$ bettercap --check-updates
This software must run as root.
[!] BetterCap v1.6.2 - A Network Protocol Sniffer & MitM Framework
[!] http://bettercap.org/
[!] web and network security tool
```

```
[I] Checking for updates ...
[I] You are running the latest version.
```

```
(loren@kali)-[~]
$ 
```

```
[I] Checking for updates ...
[I] You are running the latest version.
[!] BetterCap v1.6.2 - A Network Protocol Sniffer & MitM Framework
[!] http://bettercap.org/
[!] web and network security tool
```

```
bettercap 1.6.2
```

```
(loren@kali)-[~]
$ 
```

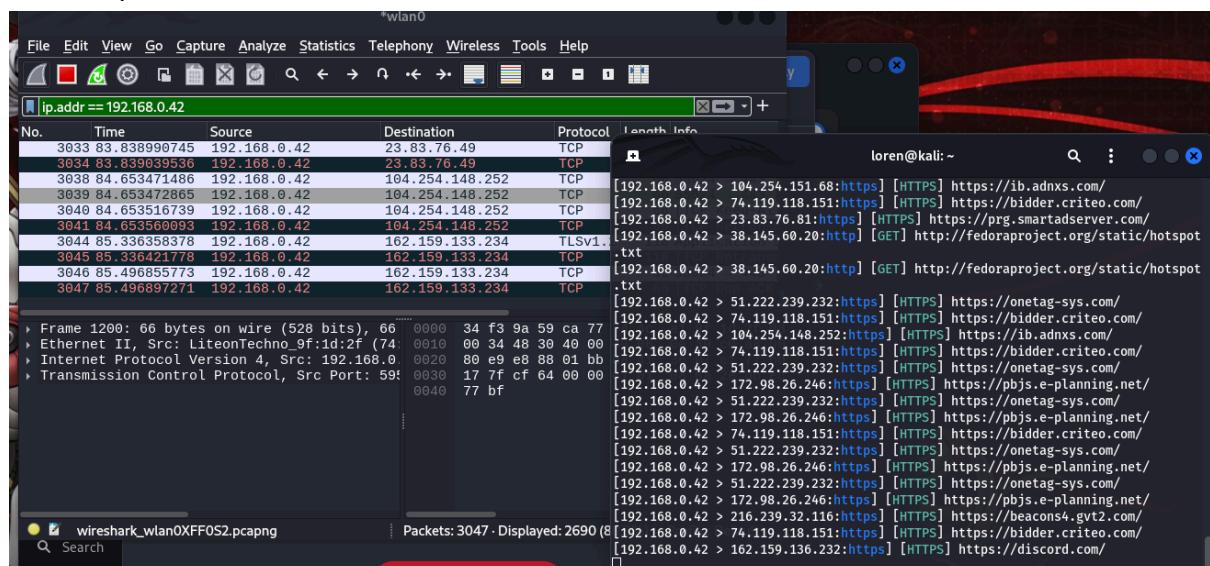
Una vez instalado revise que esté en efecto esté correctamente instalado. Lo cual parece que en efecto, todo bien.

```

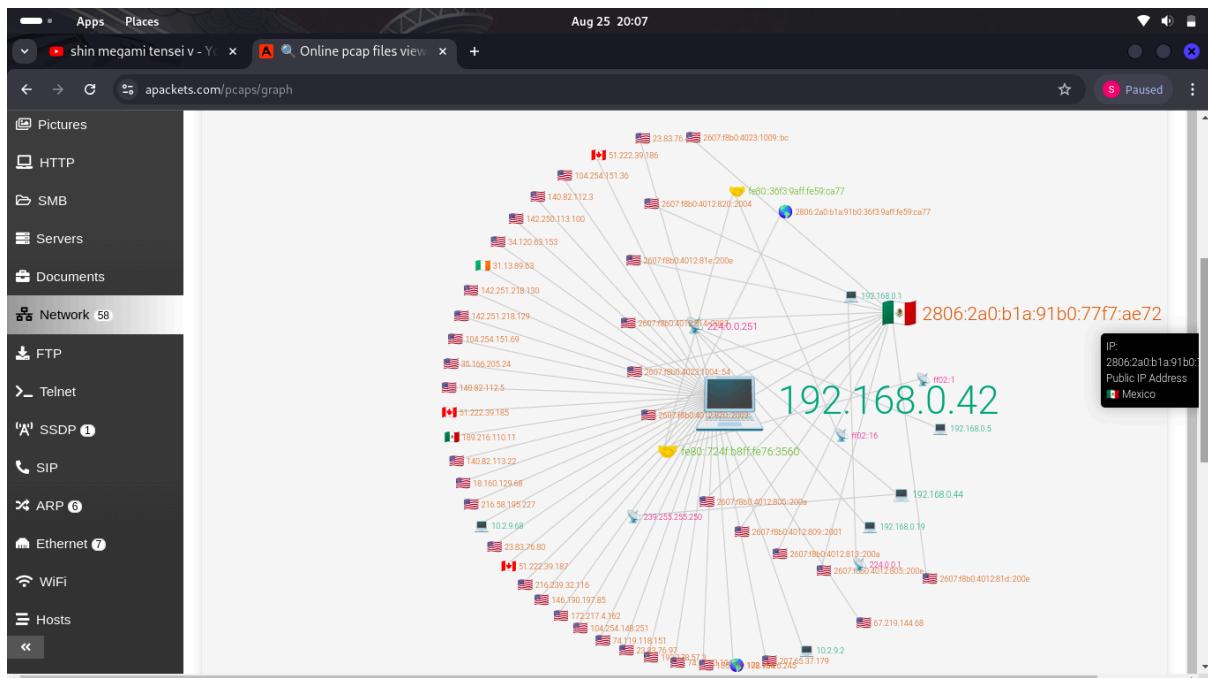
loren@kali:[~]
$ sudo bettercap --interface wlan0 --sniffer --target 192.168.0.42
[+] BetterCap v1.6.2
http://bettercap.org/
[I] Starting [ spoofing:✓ discovery:✗ sniffer:✓ tcp-proxy:✗ udp-proxy:✗ http-proxy:✗ https-proxy:✗ sslstrip:✗ http-server:✗ dns-server:✗ ] ...
[I] [wlan0] 192.168.0.44 : 34:F3:9A:59:CA:77 / wlan0 ( Intel Corporate )
[I] [GATEWAY] 192.168.0.1 : 70:4F:B8:76:35:60 ( ??? )
[I] [TARGET] 192.168.0.42 : 74:4C:A1:9F:1D:2F ( ??? )
[192.168.0.42 > 192.0.77.3:https] [HTTPS] https://64.media.tumblr.com/
[192.168.0.42 > 192.0.77.3:https] [HTTPS] https://va.media.tumblr.com/

```

Con la ip de mi laptop y mi interfaz inició el ataque. Lo cual me preocupo bastante ya que mi otra computadora seguía como si nada y podía ver en la terminal de mi otra computadora todos los sitios a los que me estaba metiendo, por ejemplo y como se muestra en la imagen a tumblr. Todo sin mayor problema y relativamente fácil de hacer. Me estaba espiando y mi otra computadora como si todo estuviera bien.



Me llamó la atención una vez más la velocidad y la cantidad de paquetes que estaba recopilando.



Al inicio hice la actividad dentro de mis cuentas, pero desconozco si eso iba a afectar o si contaba como información delicada así que repetí la actividad y ya con esos datos dejé que la página generará el gráfico.

Preguntas:

1. ¿Qué significa spoofing y sniffing? Explica la diferencia con un ejemplo.

Spoofing es el uso de técnicas que usa un atacante para suplantar la identidad de un individuo, una empresa, una página web o alguna otra entidad “confiable”. El propósito es obtener acceso a datos o información confidencial. El spam se puede realizar a través de sitios web, correos electrónicos, llamadas telefónicas, mensajes de texto, direcciones IP y servidores.

Snifing es una técnica que sirve para interceptar paquetes de datos que viajan a través de una red, el atacante usa sniffing para “escuchar” o “capturar” el contenido. Se utilizan rastreadores para capturar paquetes de datos que pueden contener datos confidenciales.

La diferencia principal es la forma de llevar a cabo los ataques, Sniffing solo escucha o captura los datos sin necesidad de interactuar con la víctima o fingir ser alguien confiable mientras que Spoofing usa esta falsa identidad para obtener los datos que el atacante necesita.

Por ejemplo digamos que quiero robar los datos de una cuenta de Cinépolis Klick. Con sniffing podemos utilizar un sniffer como wireshark que capture los datos del usuario cuando éste los introduce a la página (la real), como su contraseña, nombre de usuario y correo. Con Spoofing podríamos hacer una página falsa y anunciar promociones con un correo falso para enviárselo a usuarios de los cuales hay certeza que usan la página real, les adjuntamos el link de la página falsa y ahí introducirán sus datos, todo esto haciendo pasar nuestra dirección de la página como un sitio confiable.

2. ¿Qué pasa cuando hay 2 direcciones MAC iguales conectadas a la misma red?

Es muy poco probable que hayan dos dispositivos con la misma dirección MAC ya que es una dirección física única que se conforma a partir de la información del fabricante y la información del dispositivo. Pero en dado caso de que hubieran dos direcciones MAC conectadas a la misma red, esto provocaría pérdida de paquetes y de conectividad, ya que la red se confundiría sobre qué dispositivo debe recibir el paquete.

3.- Si estoy conectado al sitio fc.ciencias.unam, y tengo la dirección IP 192.168.0.13, y mi amigo tiene la dirección 192.168.0.13 al otro lado de la ciudad, ¿Cómo es que el servidor puede diferenciarme a mí de mi amigo, si ambos "tenemos la misma dirección"IP?

El servidor puede diferenciar ambos dispositivos gracias a que a pesar de que son la misma dirección ip, estas son direcciones ip privadas, las peticiones de estos dispositivos salen enmascaradas por la dirección ip pública correspondientes de cada módem de internet desde el que se hace la petición.

4 ¿Qué servicio nos permite dejar de preocuparnos por asignar direcciones IP a nuestros dispositivos?

El protocolo DHCP nos permite dejar de preocuparnos por asignar direcciones IP a nuestros dispositivos. Este protocolo asigna de forma dinámica direcciones IP a los dispositivos conectados en una red. Con asignación dinámica permite la reutilización de las direcciones IP, el administrador solo define el rango de las direcciones y cuando un dispositivo se conecte a la red se le asignará una dirección disponible (teniendo una lista donde están las direcciones a medida de que se quedan libres).

5. El Sniffing puede ser muy efectivo para interceptar información, pero ¿qué necesitamos hacer antes de poder efectuarlo?

Como sniffing es un ataque que se realiza a través de una red necesitamos una conexión en la que se están enviando los paquetes de datos que se quieren interceptar. Necesitamos un sniffer, es decir un programa que nos permita escuchar o captar el canal donde están pasando los datos, existen programas como Wireshark, Ettercap, Tcpdump que se especializan en este tipo de ataques. La red que se utiliza tiene que estar conectada con la víctima claramente. También hay protocolos que pueden identificar al atacante entonces también es bueno prevenir esto ocultando nuestra dirección IP o MAC.

6. Si la red fuera pública, y accedieras a un sitio https donde ingresaras tus credenciales, ¿podría un intruso conocer la información de tus credenciales? Explica por qué no, o si sí, cómo.

Si. El protocolo https encripta la información mediante el protocolo Transport Layer Security (TLS) antes conocido como Secure Socket Layer(ssl) usando dos llaves una pública y otra privada, es decir es un cifrado simétrico. Para que alguien pueda acceder a nuestra información, el atacante debe generar certificados falsos o llaves falsas tal que haga creer a su víctima que está en una conexión segura cuando realmente esto no es así. Para esto se puede explotar la debilidad descubierta por Moxie Marlinspike. Este tipo de ataque se puede hacer usando la herramienta sslStrip la cual convierte una página https a una página http, quitando la capa SSL haciendo creer al usuario que su información está encriptada bajo HTTPS, cuando realmente está usando el protocolo HTTP y por ende haciendo su información visible al atacante .

7. ¿Cómo es que funciona el bypass al protocolo HTTPS? Explica con la siguiente url <https://www.facebook.com/>.

El bypass al protocolo HTTPS o mejor conocido como un ataque de SSL. Esta vulnerabilidad fue descubierta por primera vez por el computólogo estadounidense Moxie Marlinspike en 2009. Este ataque se puede realizar con el apoyo de la herramienta SSLstrip. Como se mencionó en el inciso anterior El protocolo HTTPS encripta la información con el protocolo TLS/SSL, un ataque de SSL consiste en que alguien se posicione entre la víctima y el sitio al cual quieren acceder mediante algún otro método de MitM. Ahora el atacante crea una conexión segura con el sitio web al cual el usuario quiere acceder, pero en vez de darle al usuario la conexión segura le manda una conexión insegura de HTTP.

Es decir, digamos que A es la víctima de B y A quiere ingresar a su cuenta de facebook, pero desconoce que B ha logrado infiltrarse en su red.

A entra a <https://www.facebook.com/>, pero lo hace poniendo en su navegador www.facebook.com.

El navegador desconoce si esta página es compatible con el protocolo HTTPS, así que los primeros paquetes serán en HTTP.

B intercepta esto y él le manda al servidor correspondiente la petición obteniendo así un certificado válido. En pocas palabras B consiguió crear una conexión con <https://www.facebook.com>.

Ahora B le mandara a A la página que quiere, pero usando el protocolo HTTP, es decir A ahorita estaría realmente en <http://www.facebook.com> aunque tanto A como el servidor

crean que este en <https://www.facebook.com>, enviando a B toda la información en texto plano.

8 ¿En qué casos este tipo de ataques pueden ser útiles y para el bien de todos?

Menciona 3 ejemplos.

Puede ser útil en el desarrollo de aplicaciones web, para realizar diversas pruebas de seguridad y así poder desarrollar software de mejor calidad y que no ponga en riesgo a sus clientes.

Otro caso donde pueden ser útiles son en ambientes educativos (como esta práctica) donde se intenta enseñar sobre diversos riesgos que podemos experimentar al usar diferentes redes, además que comprender estrategias de como poder prevenirlos al observar cómo se comportan usando un ambiente controlado.

Y por último, los forenses de red (network forensics) pueden usar este tipo de ataques para capturar información o tráfico en una red y de esta manera poder obtener datos o pistas sobre la comunicación entre sistemas involucrados en algún caso o incidente

Conclusiones :

En un mundo donde cada vez dependemos de las computadoras tanto para almacenar nuestra información y como estudiantes en el campo de la computación, es importante conocer y entender cómo se llevan este tipos de ataques. En esta práctica conocimos diversas herramientas que la mayoría desconocía. Algo que nos llamó la atención, es que previamente sabíamos que el protocolo HTTP era inseguro pero, desconocíamos exactamente el porqué de esto, verlo con nuestros propios ojos nos ayudó bastante a entenderlo y comprender porque es necesario el protocolo HTTPS, todo para luego averiguar que aunque es mucho más seguro que HTTP igual puede ser “roto” si alguien realmente quisiera acceder a nuestra información. Al momento de usar redes públicas tenemos que ser muy cuidadosos (una opción muy útil al interactuar redes públicas es usar un VPN, para de esta forma estar de forma “anónima” y no comprometer nuestros datos). De igual manera aprendimos cómo estas herramientas nos pueden ayudar a detectar ataques MitM los cuales son complicados de detectar puesto que son muy sigilosos, pero una opción es usar las mismas herramientas para saber si estamos siendo vigilados; como usar wireshark para ver el tráfico de red y ver si hay alguna ip que no reconocemos o varios paquetes que no parecen coincidir con nuestro rastro en la red. Hay también software que detecta ataques sniffing, avast por ejemplo ofrece ese servicio.

Algunos de nosotros hemos tenido complicaciones antes con prácticas o tareas en las que las direcciones IP eran necesarias pero con esta práctica han quedado claras algunas dudas que hemos tenido. Aprendimos muchas cosas sobre la práctica y fueron cosas muy interesantes. La práctica está muy bien explicada y los pasos no fueron tan difíciles de

realizar, a excepción de algunos problemas que tuvimos con la distribución de Linux o con la máquina virtual.

Referencias:

- Avira (2024, 31 enero) *What's the difference between a public and private IP address?*
<https://www.avira.com/en/blog/public-vs-private-ip-address#:~:text=A%20public%20IP%20address%20is,connect%20securely%20to%20one%20another.>
- Burke, J. (2024, 23 julio). MAC address vs. IP address: What's the difference?
<https://www.techtarget.com/searchnetworking/answer/What-is-the-difference-between-an-IP-address-and-a-physical-address>
- Farrier, E. (2024, 1 agosto). Packet Sniffing Explained: Definition, Types, and Protection. Packet Sniffing Explained: Definition, Types, And Protection.
<https://www.avast.com/c-packet-sniffing>
- Invicti. (2024, 17 mayo). SSL hijacking.
<https://www.invicti.com/learn/mitm-ssl-hijacking/>
- Jadhav, N. (2021, 15 diciembre). BYPASSING HTTPS - Ninad Jadhav - Medium. Medium. https://medium.com/@stealth_fearzzz/bypassing-https-75bccdf88c49
- Keepcoding (s.f.) ¿Qué es la dirección MAC y para qué sirve?
<https://keepcoding.io/blog/que-es-la-direccion-mac-y-para-que-sirve#:~:text=La%20direcci%C3%B3n%20MAC%20se%20presenta,ser%20una%20direcci%C3%B3n%20MAC%20t%C3%ADpicamente.>
- Leather_Success2639 (2022) Could be there two identical MAC addresses? [Publicación de foro en línea]. Reddit.
https://www.reddit.com/r/networking/comments/10qpfiv/could_be_there_two_identical_mac_addresses/
- Manipulator-in-the-middle attack | OWASP. (n.d.). Owasp.org.
https://owasp.org/www-community/attacks/Manipulator-in-the-middle_attack
- Protocolo DHCP: cómo activarlo y desactivarlo, ventajas e inconvenientes. (n.d.). ADSLZone. <https://www.adslzone.net/como-se-hace/wifi/activar-dhcp/>
- Shivanshu. (2024, 10 abril). Sniffing and Spoofing: A Comprehensive Differentiation. Intellipaat. <https://intellipaat.com/blog/sniffing-and-spoofing/>
- Venafi. (s. f.). What are SSL stripping attacks? | Venafi. Venafi.
<https://venafi.com/blog/what-are-ssl-stripping-attacks/>
- What is sniffing ? How does sniffing work? | Lenovo US. (2023, 28 mayo).
https://www.lenovo.com/us/en/glossary/sniffing/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsItid=AfmB0ooVTc54S-WXRmoHUwhMBw7tAhpp7f69xDZdmXBqqsCx11jzzp_x
- What is HTTPS? (s. f.). Cloudflare. Recuperado 26 de agosto de 2024, de
<https://www.cloudflare.com/en-gb/learning/ssl/what-is-https/>