
WIRESHARK

Contenido

Wireshark

1. Descripcion

Wireshark es el analizador de protocolos de red más importante y usado en el mundo. Es el estándar en muchas empresas comerciales y sin fines de lucro, agencias gubernamentales e instituciones educativas, siendo utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica.

Las mayores ventajas frente a otros analizadores de protocolos es tener una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red estableciendo la configuración en modo promiscuo (captura del tráfico).

Permite examinar datos de una red en uso o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete.

Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos Unix y compatibles, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Android y Mac OS X, así como en Microsoft Windows.

El desarrollo de Wireshark prospera gracias a las contribuciones voluntarias de expertos en redes en todo el mundo y es la continuación del proyecto Ethereal, iniciado por Gerald Combs en 1998.

Para asegurarnos que la captura que realicemos sea completa, deberemos ejecutar Wireshark como súper usuario, también se puede configurar para que la captura la realicen usuarios normales del sistema, pero para asegurarnos al 100% es recomendable ejecutar como administrador.

Si quisiéramos guardar directamente el trafico analizado, podemos usar una herramienta que viene incorporada en Wireshark llamada tcpdump, la cual tiene el mismo funcionamiento que Wireshark, pero almacenando los datos en un fichero para luego ser analizados.

También existe otra herramienta llamada Tshark, la cual es una versión de Wireshark, pero en linea de comandos

2. Instalación

Wireshark está disponible para múltiples sistemas operativos, en su página oficial podemos encontrar los instaladores para Windows de 64 y 32 bits, un portable de 32 bits para Windows, un instalador para macOS y el código fuente, el cual puede ser utilizado para distribuciones Linux, adicionalmente también encontramos la documentación:

https://www.wireshark.org/download.html

En Linux, además podemos instalar directamente desde el terminal desde los repositorios oficiales con el siguiente comando:

```
sudo apt install wireshark
```

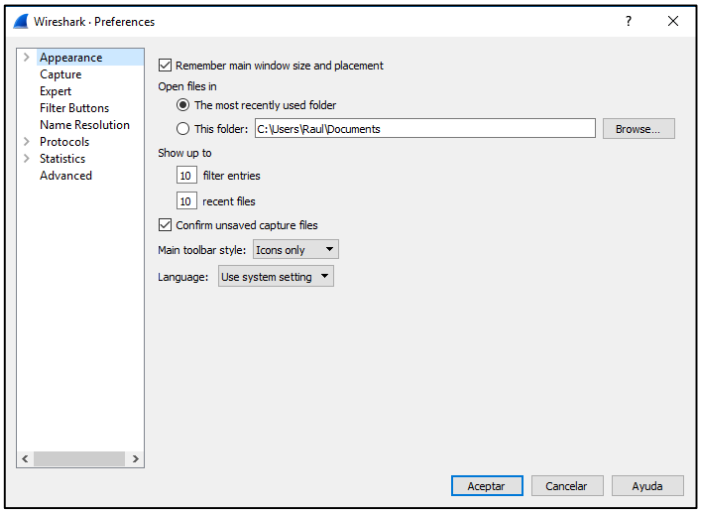
Si no queremos instalarlo de los repositorios oficiales, podemos instalarlo con los comandos:

```
sudo add-apt-repository ppa:wireshark-dev/stable
sudo apt update
sudo apt install wireshark
```

Otra forma de instalarlo en Ubuntu, es mediante el centro de software del sistema.

3. Configuración de Wireshark y menús

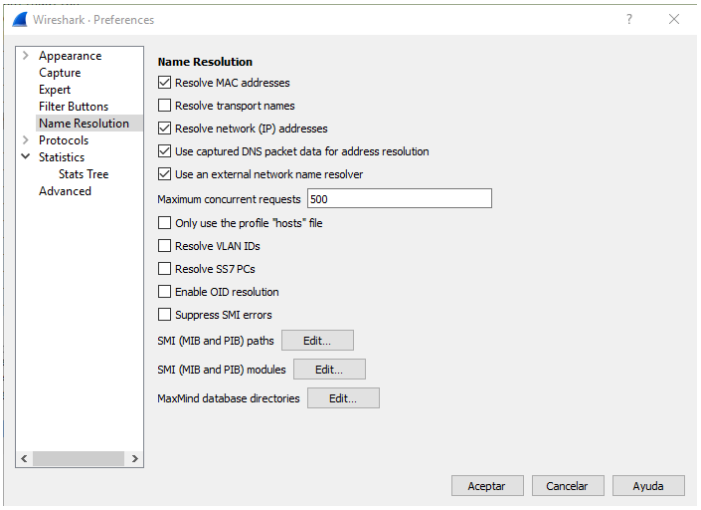
Antes de empezar a usar Wireshark, hay algunas cosas que debemos configurar para una correcta captura del tráfico, esto lo hacemos desde la pestaña “Edit” en la opción “Preferences”, donde veremos la siguiente imagen:



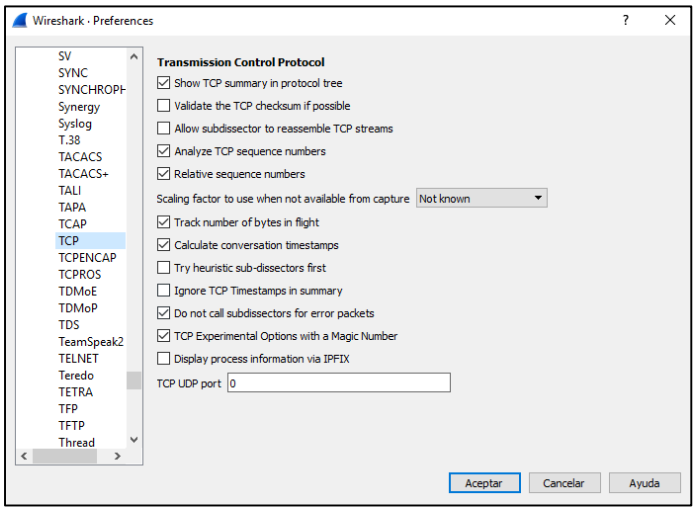
En el menú lateral izquierdo, podemos ver las diferentes ventanas que podremos configurar, las más importantes son:

- Name Resolution
- Protocols
- Statistics

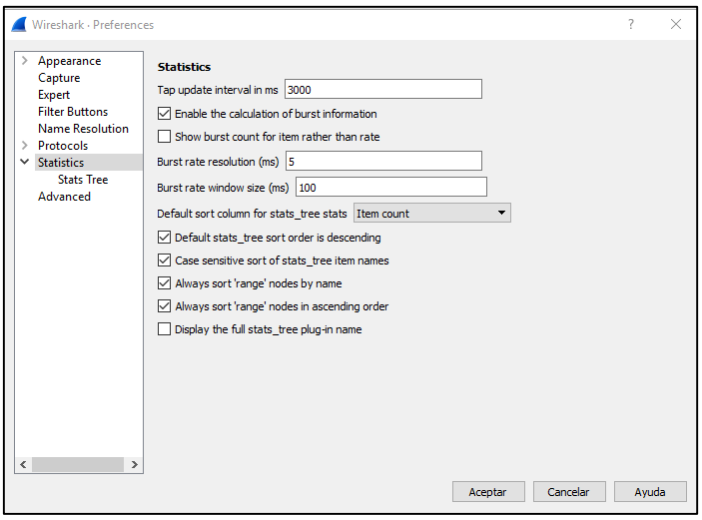
En la ventana de “Name Resolution”, podremos configurar si queremos que Wireshark resuelva el nombre del dominio y nos lo muestre en vez de la direccion IP si está disponible, también podemos marcar opciones para que resuelva los identificadores de las Virtual LAN o las direcciones MAC.



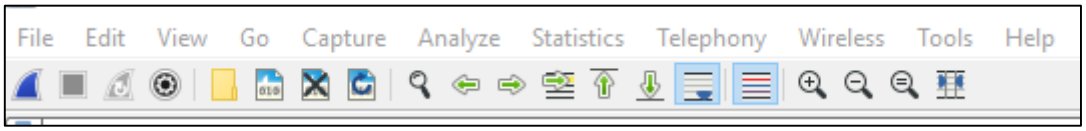
En la ventana “Protocols”, podemos ver una lista de todos los protocolos de los que dispone Wireshark, que no son pocos, por ejemplo, en el protocolo TCP, podemos hacer que no permita re ensamblar las transmisiones TCP.



Y la ventana “Statistics”, donde podremos configurar como se nos muestra la información:



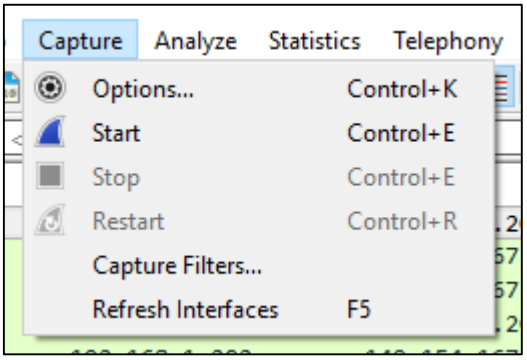
Además, Wireshark dispone de una serie de menús en la parte superior, donde podremos manipular y cambiar la forma en la que tratamos el trafico capturado.

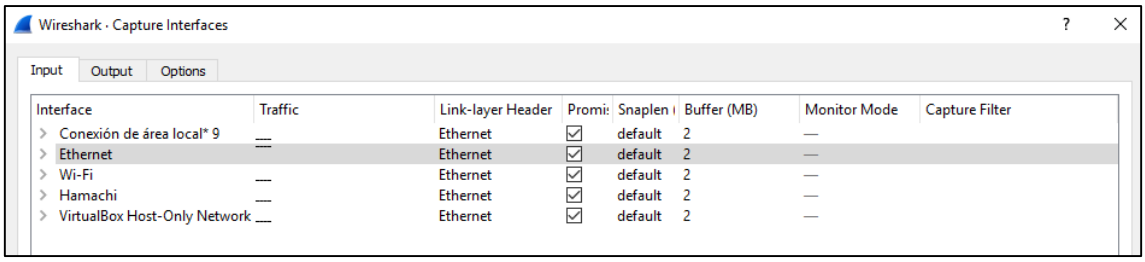


Algunos de los menús que consideramos que son los más importantes son los siguientes:

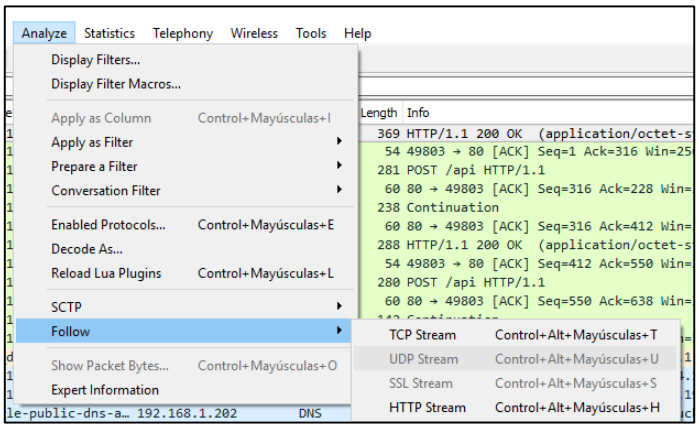
- Capture
- Analyze
- Statistics
- Tools
- Telephony
- Wireless

El menú “Capture” nos permite configurar la forma en la que capturamos el tráfico, permitiendo elegir la red en la que capturar, iniciar la captura, pararla o reiniciarla, y capturar según unos filtros dados.

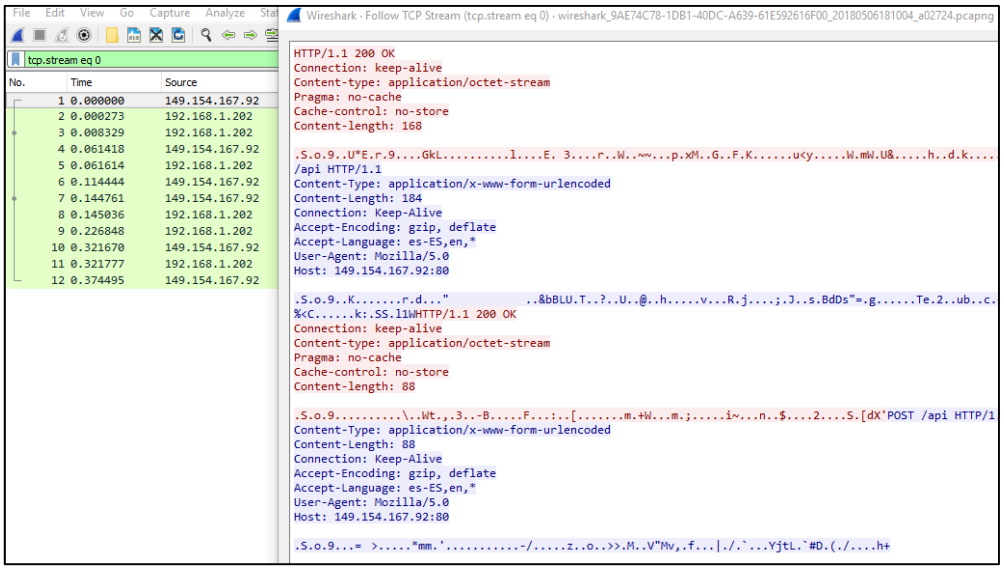




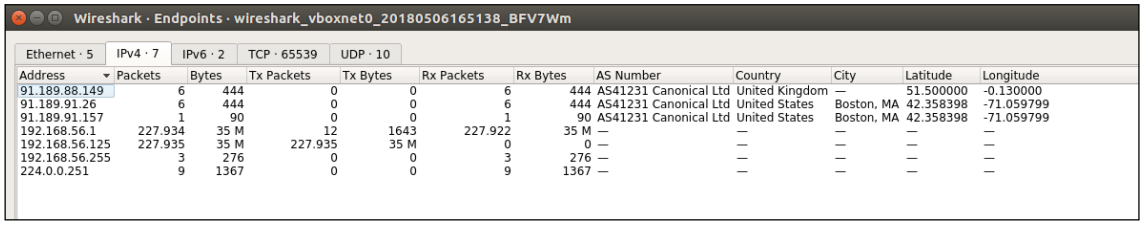
El menú “Analyze”, nos permite analizar el trafico capturado de forma más individual, así podemos centrarnos en lo que realmente queremos analizar sin desviarnos por capturas intermedias, como alguna captura DNS esporádica.

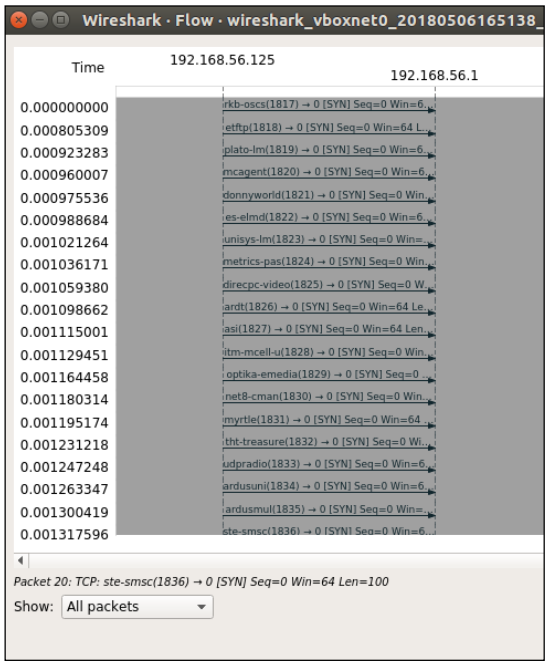


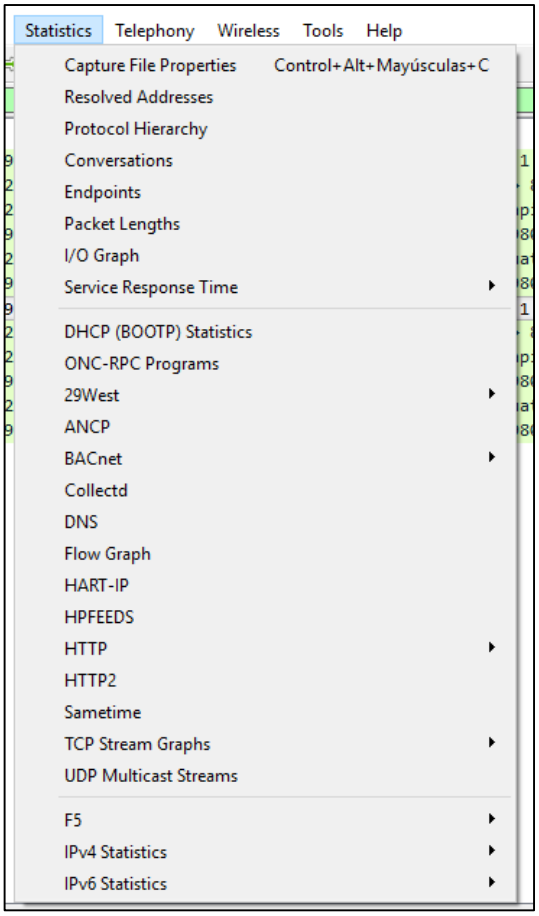
La opción más interesante que hemos encontrado en este menú, ha sido la de “TCP Stream” y la de “HTTP Stream”, las cuales nos crean un filtrado de la opción elegida, y abren una ventana con información de la traza como podemos ver en la imagen inferior:



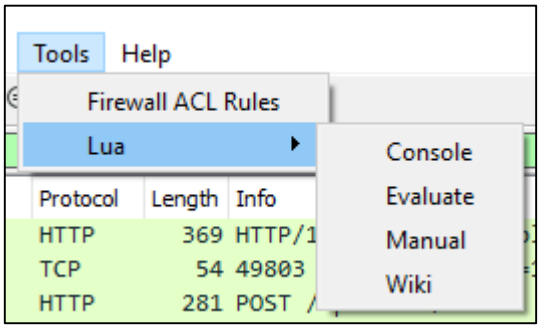
El siguiente menú es “Statistics”, este menú nos permite ver el tráfico de distintas formas, ya sea como diagrama de flujo, gráficos de I/O, puntos finales o incluso mirar las estadísticas de IPv4 o IPv6.



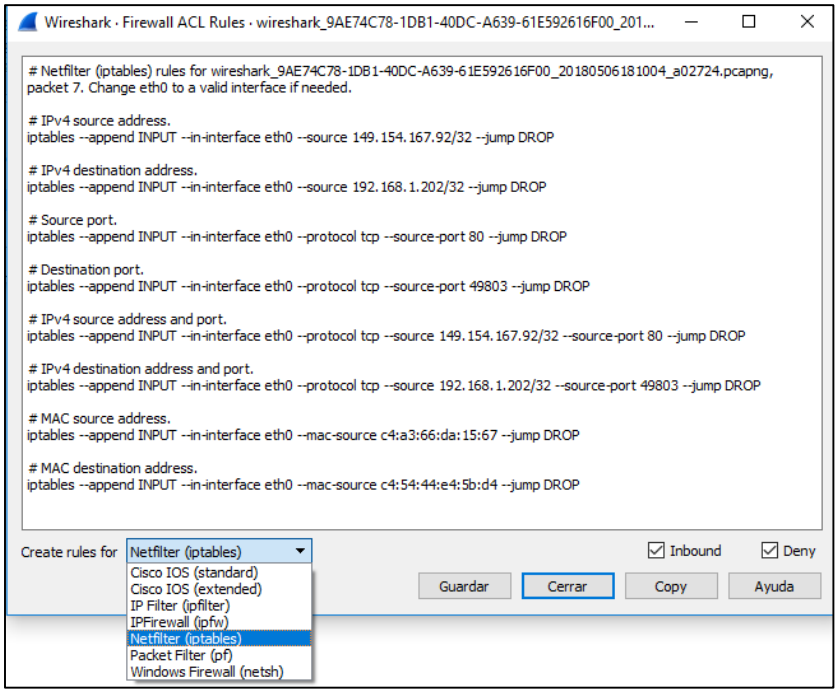




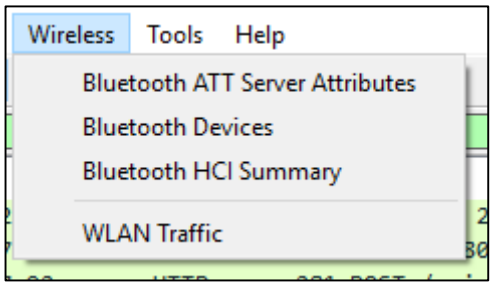
El ultimo menú importante a nuestro parecer, es “Tools”, en el cual, podemos establecer las reglas del firewall que usa Wireshark para analizar el tráfico.

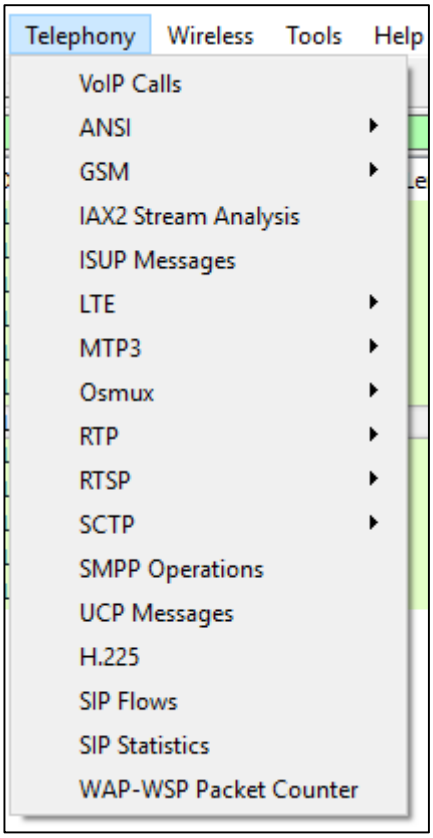


En la siguiente imagen podemos ver las reglas iptables que genera para un perfil propio:



Finalmente, creemos que debemos mencionar estos dos menús: “Telephony” y “Wireless”, los cuales nos permiten analizar el tráfico generado en Llamadas o las conexiones mediante Bluetooth en el menú Wireless.





4. Demos

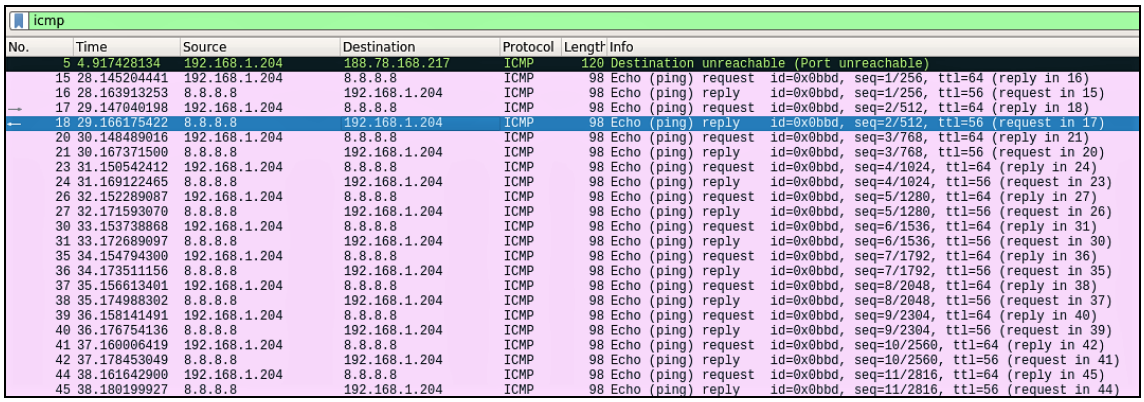
En esta parte, vamos a realizar una serie de pruebas en las que veremos el funcionamiento de Wireshark y cómo interpretar le trafico asociado a cada evento.

4.1. Ping

El programa ping usa el protocolo ICMP. Por tanto, para comprobar el uso de ping deberemos filtrar dicho protocolo.

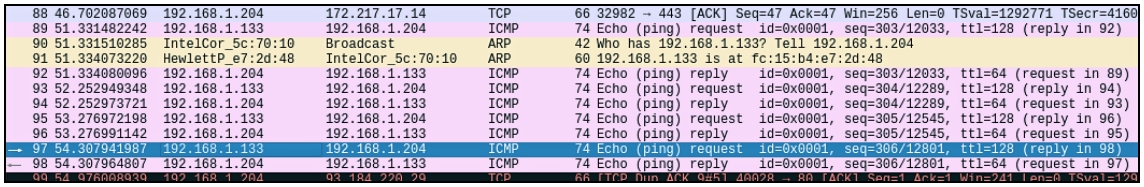
Lanzamos ping a Google y capturamos la red. Filtramos el protocolo ICMP.

Ahora podemos comprobar en la interfaz de Wireshark que nuestra máquina (192.168.1.204) manda una solicitud a Google (8.8.8.8), y a su vez, Google responde.



Ahora hacemos un ping desde otra máquina de la red (192.168.1.133) a la máquina que tiene Wireshark (192.168.1.204).

Al capturar la red y filtrar el protocolo ICMP podemos comprobar que efectivamente, la máquina con IP 192.168.1.133 manda una solicitud a la máquina con IP 192.168.1.204, y a su vez esta le responde.



Como podemos ver en la imagen superior, al seleccionar una linea, Wireshark nos indica donde están las filas relacionadas. En este ejemplo, al seleccionar el ping request, Wireshark nos marca a la izquierda donde está el ping reply asociado a ese ping request. Esto muy útil cuando tenemos que analizar grandes cantidades de tráfico, o donde una operación consta de varias partes, como puede ser una petición http.

4.2. Detección ataque DDoS

En este apartado vamos a ver cómo detectar un ataque DDoS con Wireshark. Un ataque DDoS (Denial of Service) es un ataque de denegación de servicio, este consiste en la saturación de los puertos con múltiples flujos de información, haciendo que un recurso o servicio sea inaccesible.

Para simular el ataque DDoS hemos utilizado la herramienta hping3 desde una máquina virtual con IP (192.168.56.101) a otra la cual está haciendo peticiones ping a google constantemente (192.168.56.102).

Para instalar la herramienta hping3 ejecutamos el siguiente comando en la terminal de Ubuntu:

```
sudo apt install hping3
```

Una vez instalado, podemos iniciar el ataque con el siguiente comando:

```
sudo hping3 -c 100 -d 100 -S -w 64 -p 80 --flood 192.168.56.102
```

Donde los parámetros dados corresponden con:

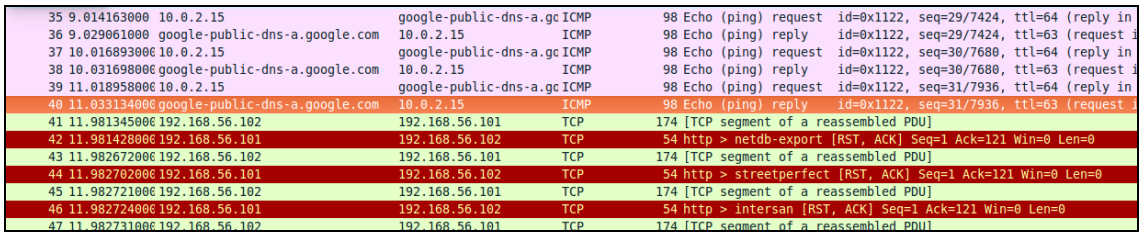
- -c --> Numero de paquetes a enviar.
- -d --> Longitud del mensaje a enviar.
- -S --> Bandera SYN.
- -w --> Tamaño paquete TCP.
- -p --> Puerto al que se envían los paquetes.
- --flood --> Los paquetes se envían en tiempo real de forma masiva.

Y acto seguido, en la maquina atacada, la cual está esnifando la red, empieza a aparecer una cantidad de trafico grandísima, ralentizando toda la máquina virtual, en la imagen siguiente podemos ver que, en casi 4 segundos, ha generado unos 75000 envíos.

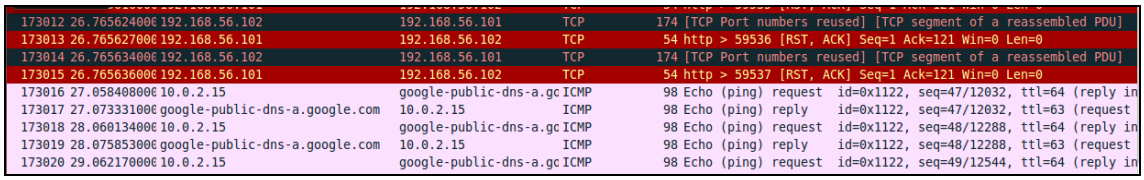
No.	Time	Source	Destination	Protocol	Length	Info
156020	3.883702000	192.168.56.101	192.168.56.102	TCP	54	http > 65531 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156021	3.883708000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156022	3.883710000	192.168.56.101	192.168.56.102	TCP	54	http > 65532 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156023	3.883766000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156024	3.883770000	192.168.56.101	192.168.56.102	TCP	54	http > 65533 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156025	3.883837000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156026	3.883842000	192.168.56.101	192.168.56.102	TCP	54	http > 65534 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156027	3.883852000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156028	3.883854000	192.168.56.101	192.168.56.102	TCP	54	http > 65535 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156029	3.883865000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156030	3.883867000	192.168.56.101	192.168.56.102	TCP	54	http > 0 [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156031	3.883876000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156032	3.883878000	192.168.56.101	192.168.56.102	TCP	54	http > tcpmux [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156033	3.883886000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156034	3.883888000	192.168.56.101	192.168.56.102	TCP	54	http > compressnet [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156035	3.883943000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]
156036	3.883947000	192.168.56.101	192.168.56.102	TCP	54	http > compressnet [RST, ACK] Seq=1 Ack=121 Win=0 Len=0
156037	3.884017000	192.168.56.102	192.168.56.101	TCP	174	[TCP Port numbers reused] [TCP segment of a reassembled PDU]

Como podemos ver en la imagen anterior, las filas rojas corresponden al envío de la conexión desde el atacante, y en oscuro la respuesta. Las líneas en rojo indican que la conexión ha sido cerrada [RST, ACK].

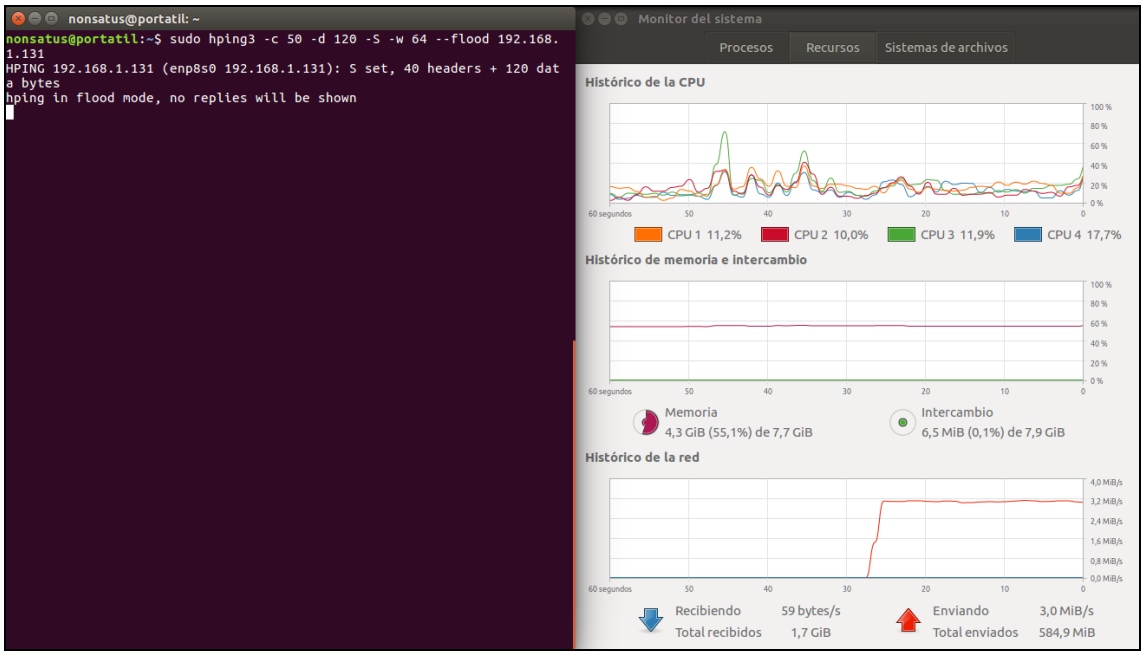
A continuación, vamos a ver un ataque DDoS mientras se ejecutan un ping a google en la maquina atacada. Como podemos ver en la siguiente imagen, cuando se empieza a realizar el ataque, los pings dejan de aparecer, para ser sustituidos por las trazas DDoS.



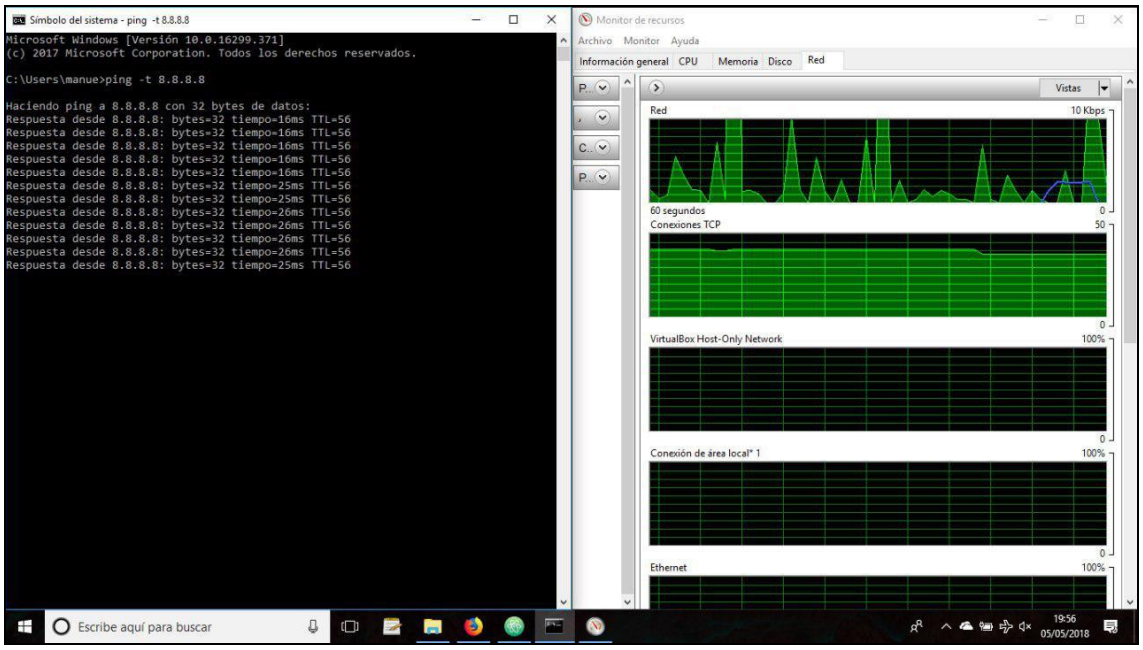
En el momento en el que se acaba el ataque, podemos ver como el ping continúa haciéndose.



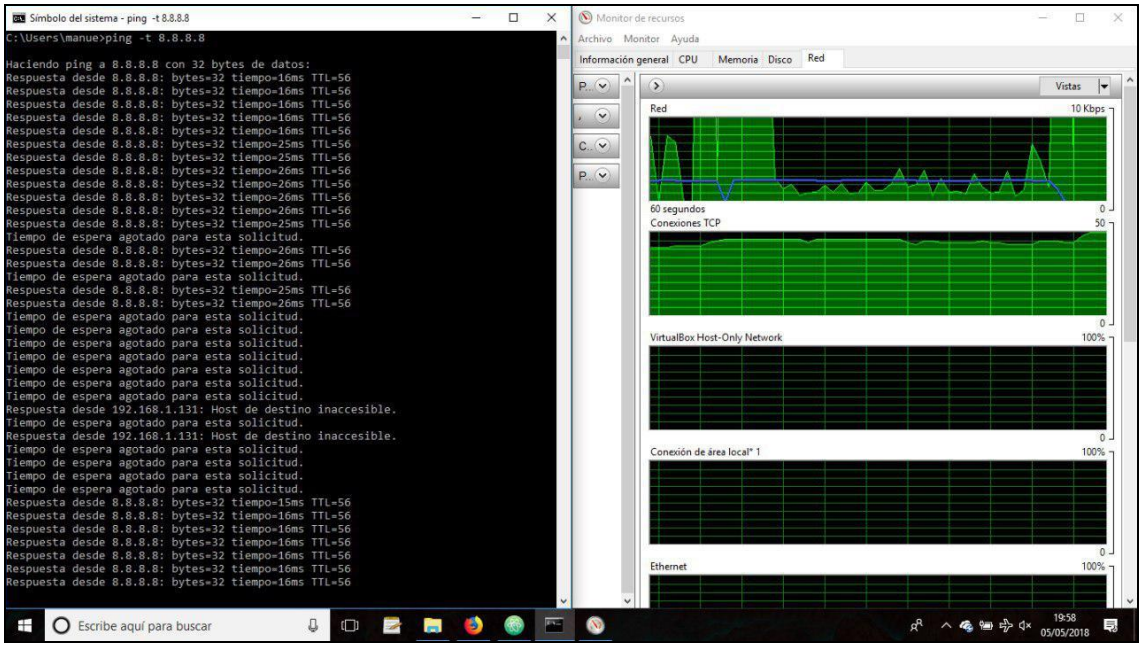
Para ver mejor el ataque, hemos simulado un ataque desde Ubuntu a una maquina Windows, a continuación, vemos la ejecución del comando hping3 y la cantidad de datos enviados en ese momento, que como podemos observar, es de 3MB/s.



En la maquina atacada, podemos ver como el ping que se está realizando, deja de ser funcional hasta que el ataque para.

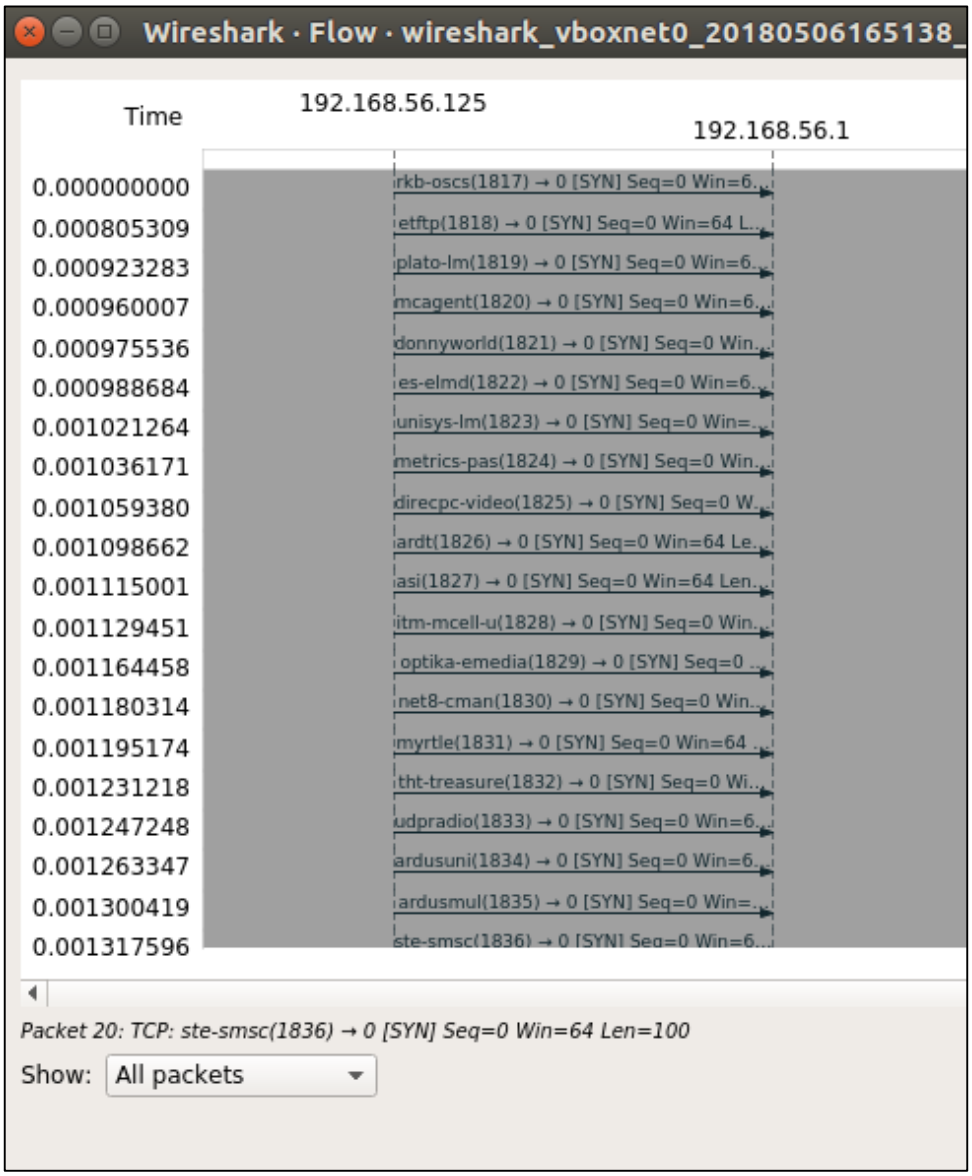


En el monitor del sistema podemos ver como al terminar el ataque, el uso de red cae.



En la siguiente imagen, podemos ver, el diagrama de flujo de un ataque DDoS, en la imagen, podemos ver dos direcciones IP:

- 192.168.56.125 (atacante)
- 192.168.56.1 (atacado) (corresponde con la maquina anfitrión)



En la imagen podemos ver como para un periodo MUY corto de tiempo (de 0 segundos a 0.0013 milisegundos) se han generado muchos intentos de conexión (20 conexiones) y podemos observar que se abren muchas conexiones [SYN] sin respuesta, lo que hace que la maquina se sature.

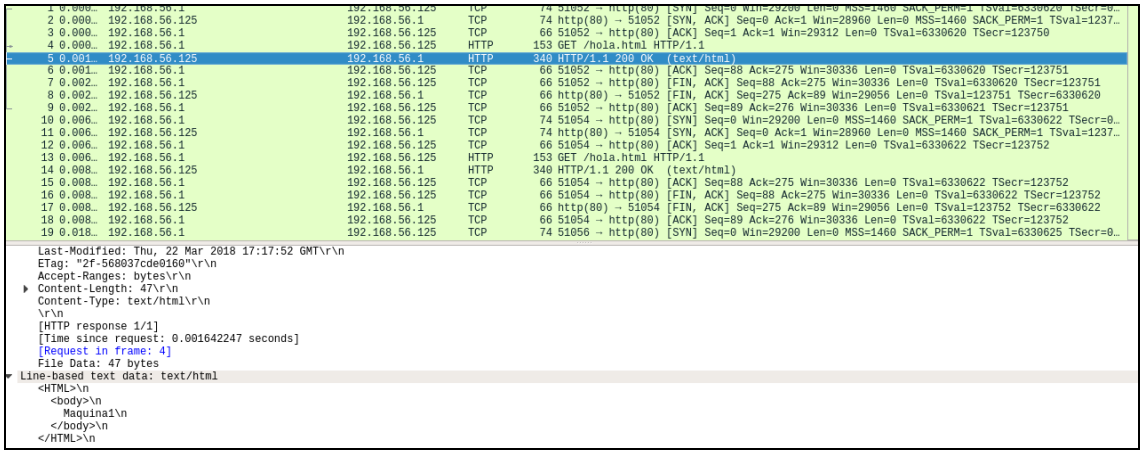
4.3. Comprobar balanceo

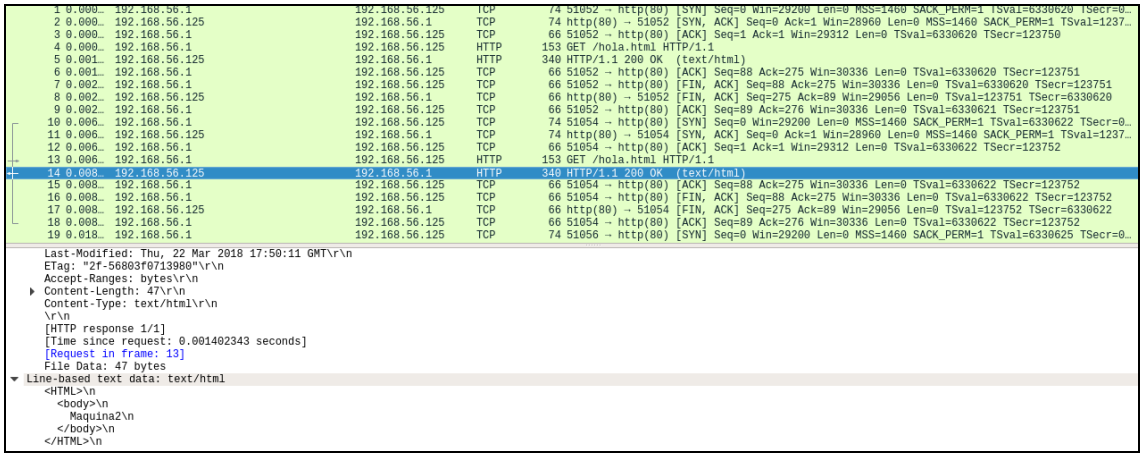
Ahora vamos a ver el funcionamiento del balanceador con Wireshark. Para esto, usamos una máquina virtual con Wireshark en la misma red en la que el balanceador y las maquinas servidoras están funcionando.

Primero, desde la maquina cliente (192.168.56.101) hacemos dos peticiones curl a la maquina balanceadora (192.168.56.125) y en Wireshark usamos un filtrado por http, ya que nos interesa ver los archivos que está balanceando la maquina balanceadora.

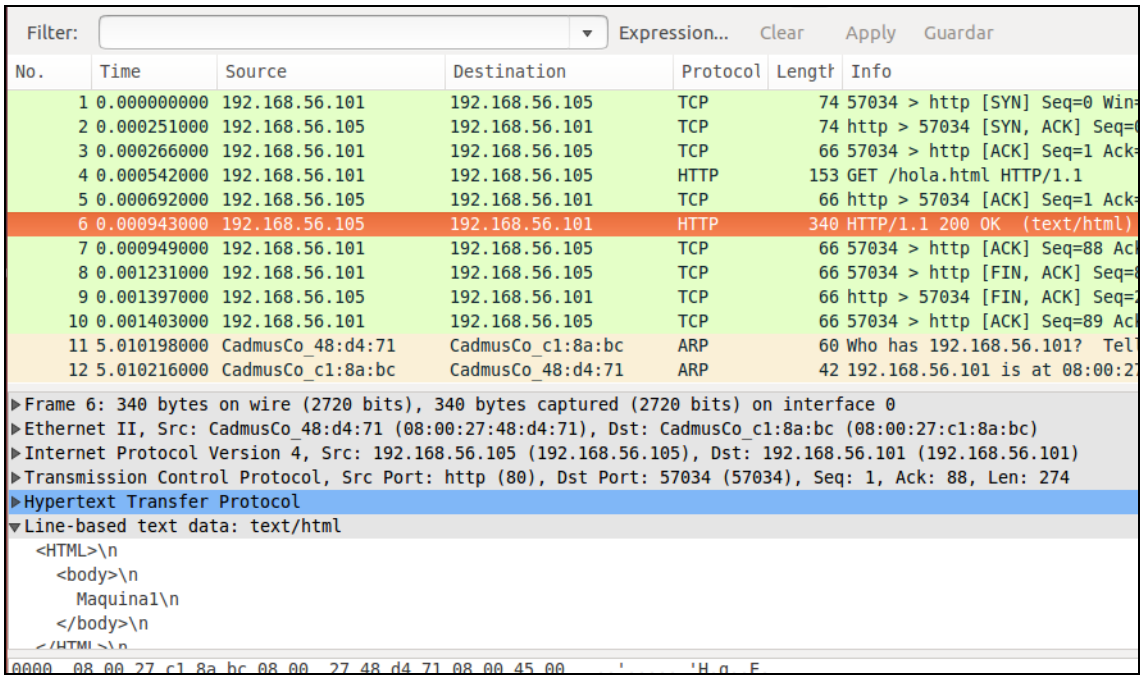
Observando las trazas, podemos ver que desde la maquina cliente, se hacen dos peticiones al balanceador mediante un método GET sobre el archivo hola.html y a continuación de dichas peticiones, obtenemos una respuesta de la maquina balanceadora con el archivo que hemos pedido en texto plano.

Ahora, si miramos la información interna de las respuestas del balanceador, veremos que se nos proporcionan los archivos de cada máquina servidora (maquina1 y maquina2) a través del balanceador.





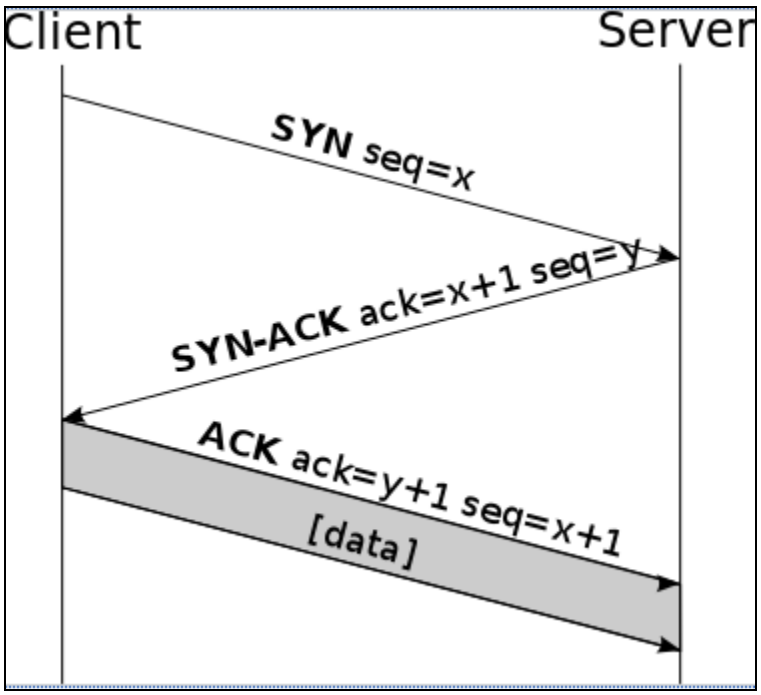
Además, si no filtramos por protocolo HTML, podemos ver todas trazas que se realizan para el curl.



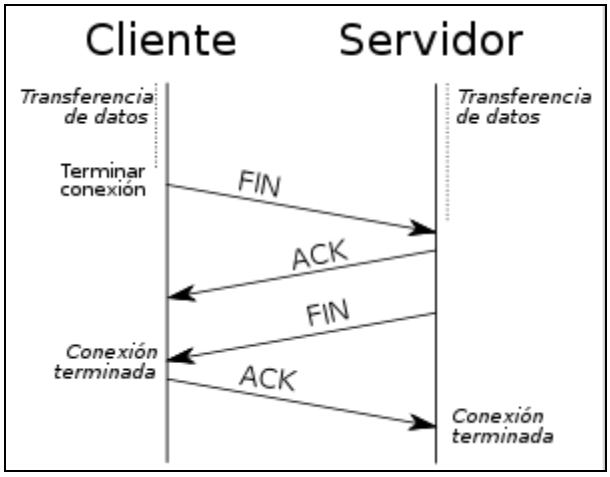
En la imagen superior, podemos ver mucha información:

- Líneas 1, 2 y 3: Inicio de la negociación en tres pasos:
  - 1: [SYN] (Cliente 192.168.56.101 -> Server 192.168.56.105)
  - 2: [SYN, ACK] (Server 192.168.56.105 -> Cliente 192.168.56.101)
  - 3: [ACK] (Cliente 192.168.56.101 -> Server 192.168.56.105)
- Líneas 4 y 6: Petición y envió del curl:
  - 4: Get /hola.html HTTP/1.1 (Petición del cliente 192.168.56.101 al servidor 192.168.56.105)
  - 6: HTTP/1.1 200 OK (text/html) (Recibo del fichero HTML solicitado desde la maquina servidora 192.168.56.105 al cliente 192.168.56.101)
- Líneas 8 y 9: Finalización de la negociación en tres pasos:
  - 8: Fin de la conexión del cliente al servidor.
  - 9: Fin de la conexión del servidor al cliente.

Inicio 3-way-handshake



Fin 3-way-handshake



4.4. Envío SCP

A continuación, vamos a ver el funcionamiento de la herramienta scp al transferir archivos de una maquina a otra. En esta ocasión, se va a transferir un archivo desde la maquina anfitrión a una maquina servidora de Virtual Box.

Wireshark está instalado ahora en la maquina anfitrión, por lo que, en las capturas, esta aparece con la ip correspondiente a la puerta de enlace 192.168.56.1, mientras que la máquina que recibe tiene la ip 192.168.56.105.

Scp o Secure Copy es una herramienta que se utiliza para la transferencia de archivos de forma segura, usando el protocolo SSH o Secure Shell, por tanto, no vamos a poder ver la información que se transfiere, pero sí de donde a donde y como.

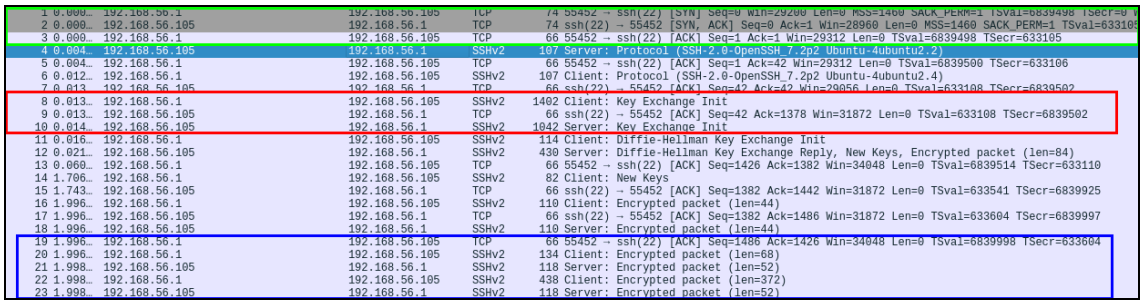
Para transferir un archivo, debemos escribir el siguiente comando:

```
sudo scp usuario@host:directorio/ArchivoOrigen ArchivoDestino
ó
scp ArchivoOrigen usuario@host:directorio/ArchivoDestino
```

En este caso, como el archivo se envía desde la maquina anfitrión, usaremos el siguiente comando, el cual transferirá el archivo al directorio de usuario de la maquina receptora:

```
sudo scp file rauldpm@192.168.56.105:/home/rauldpm/
```

Al ejecutar el comando, vemos lo siguiente con Wireshark:



En la imagen anterior, podemos ver 3 recuadros de colores.

- El recuadro verde, corresponde con el 3-way-handshake
- El recuadro rojo, corresponde con el intercambio de claves entre la maquina anfitrión y la maquina receptora
- El recuadro morado, corresponde con el envío a través de ssh por el puerto 22 de los paquetes encriptados.

Finalmente, una vez que se ha realizado la transferencia, vemos como se cierra el 3-way-handshake:

```
49 3.672…  192.168.56.1      192.168.56.105   TCP   66 55452 → ssh(22) [FIN, ACK]
50 3.673…  192.168.56.105    192.168.56.1     TCP   66 ssh(22) → 55452 [ACK] Seq=
51 3.676…  192.168.56.105    192.168.56.1     TCP   66 ssh(22) → 55452 [FIN, ACK]
52 3.676…  192.168.56.1      192.168.56.105   TCP   66 55452 → ssh(22) [ACK] Seq=
```

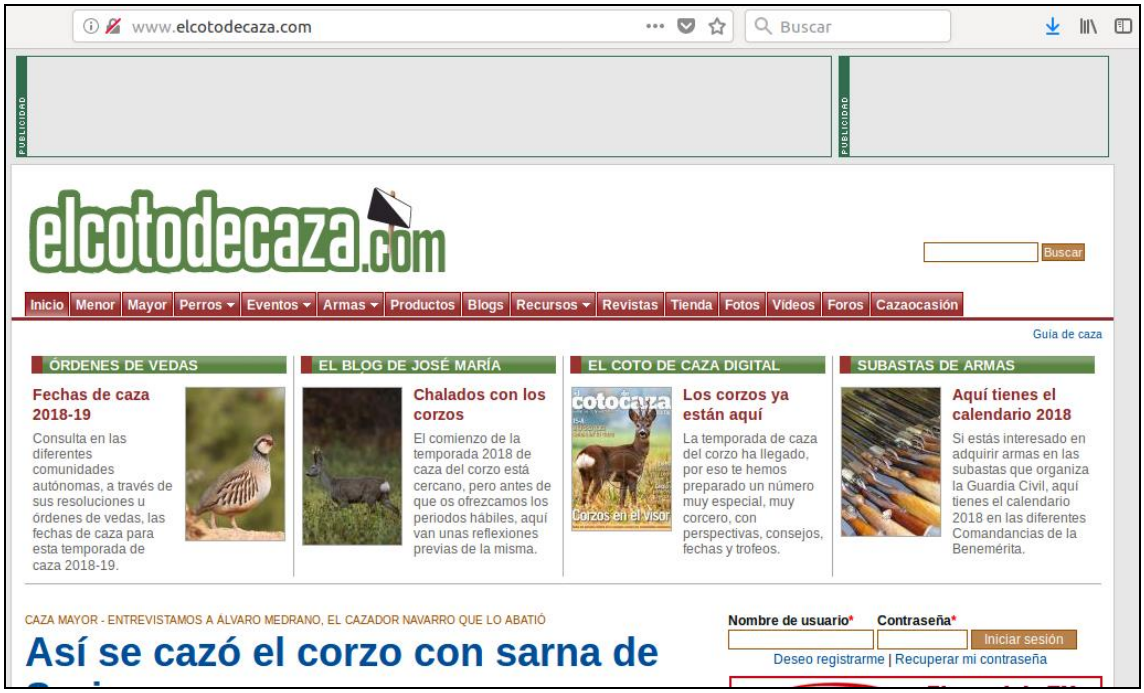
4.5. Obtener contraseña por HTTP

Finalmente, vamos a demostrar lo poco seguro que es el protocolo http cuando hay información personal de por medio.

El protocolo http (protocolo de transferencia de hipertexto) es un protocolo de comunicación que usa texto plano para la transferencia de información, pero que no está cifrada de ninguna forma, lo que lo hace muy vulnerable a diversos ataques como un MITM o un eavesdropping (ataques de escuchas telefónicas, por ejemplo).

Hoy en dia hay muy pocas webs que aun usen el protocolo http, debido a que su predecesor, el protocolo https, ofrece esa seguridad que http no tiene al cifrar la información.

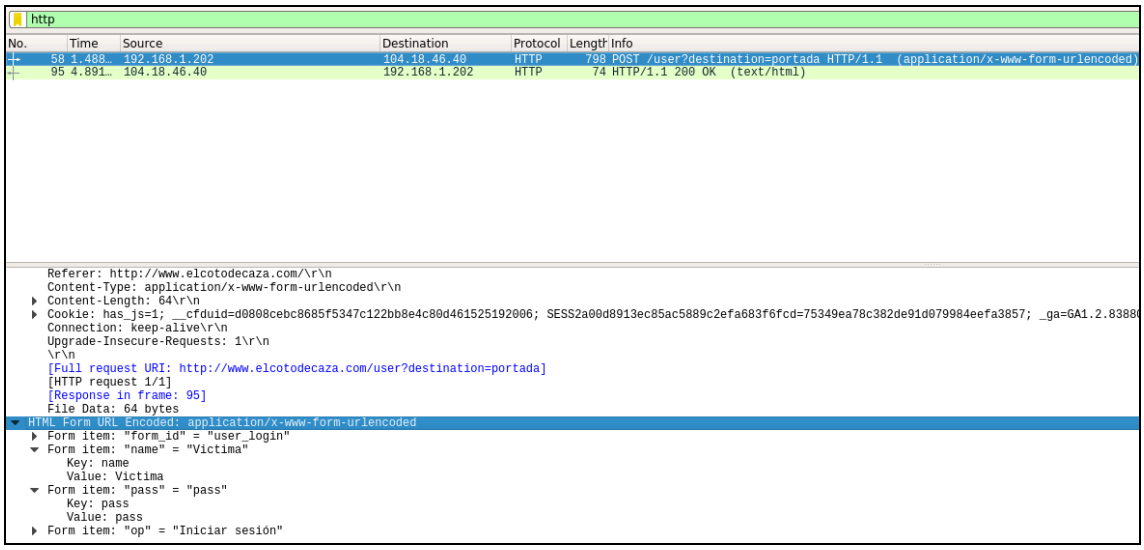
Para esta demostración, hemos usado como ejemplo, la página: http://elcotodecaza.com, una página orientada a la caza, pero que la cual dispone de un servicio de compra de productos de caza, cuya página principal, usa protocolo http, cosa más que suficiente para poder obtener información personal.



Ahora simplemente tenemos que observar Wireshark a la espera de que nuestra victima inicie sesión con sus credenciales, para esta prueba, vamos a ingresar la siguiente información:

- Nombre de usuario: Victima
- Contraseña: pass

Y al hacer click la víctima, en Wireshark nos aparecería, al igual que en las otras demos, una conexión por http con la información introducida mediante el método POST:



5. Bibliografía

- Instalación y uso de Wireshark
  - https://www.askmetutorials.com/2017/06/install-wireshark-227-in-ubuntu-1604.html
  - http://tuxylinux.com/instalar-y-configurar-wireshark-en-linux/
  - https://www.enlinux.org/instalar-wireshark-en-gnulinux-debian-ubuntu-server-ubuntu-desktop/

- Instalación y uso de hping3
  - https://www.redeszone.net/gnu-linux/hping3-manual-de-utilizacion-de-esta-herramienta-para-manipular-paquetes-tcp-ip/
  - https://kali-linux.net/article/hping3/

- Uso de Wireshark
  - https://comofriki.com/como-usar-wireshark-capturar-filtrar-analizar-paquetes/
  - https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/certinfseguridadanalisistraficowireshark.pdf
  - https://www.lifewire.com/wireshark-tutorial-4143298

- Ataques DDoS
  - https://es.wikipedia.org/wiki/Ataquededenegaci%C3%B3ndeservicio
  - https://www.lifewire.com/wireshark-tutorial-4143298
  - https://www.incibe.es/extfrontinteco/img/File/intecocert/EstudiosInformes/certinfseguridadanalisistraficowireshark.pdf