



ESNIFADO DE RED CON WIRESHARK

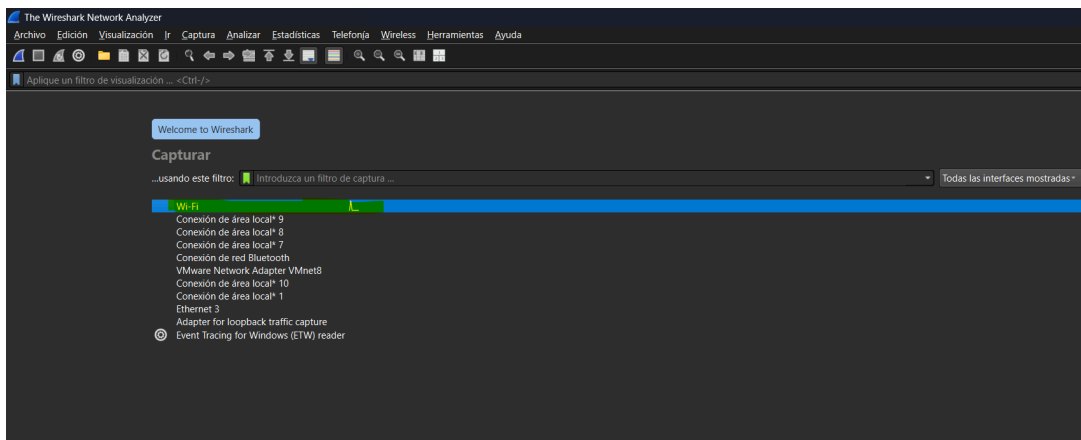
SANTIAGO PEÑARANDA MEJIA

BUENO LO PRIMERO QUE VAMOS HACER ES ABRIR LA HERRAMIENTA **WIRESHARK**, POR SI NO SABES QUE ES **WIRESHARK** ESTA ES UNA HERRAMIENTA QUE SIRVE PARA ESNIFAR O CAPTURAR PAQUETES QUE VIAJAN POR LA RED, EN CUESTIÓN VER EL TRÁFICO DE LA RED.

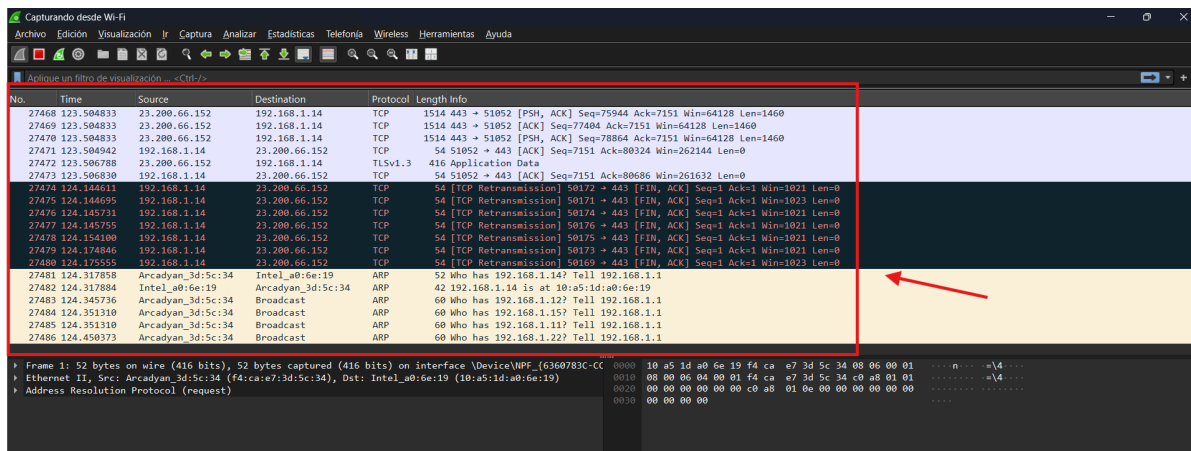
ESTA HERRAMIENTA ES MUY ÚTIL PARA NOSOTROS COMO **PENTESTERS** PORQUE NOS VA A PERMITIR ANALIZAR DATOS Y PROTOCOLOS QUE NOS VIENE MUY ÚTIL PARA LO QUE ESTEMOS HACIENDO SI SE TRATA DEL ÁMBITO DE LA RED, Y ADEMÁS ES UNA HERRAMIENTA MUY COMPLETA.



Una vez ya tengamos abierta la herramienta, es hora de seleccionar la interfaz de red que vamos a usar para hacer el esnifado, en mi caso voy a usar la “**Wifi**”

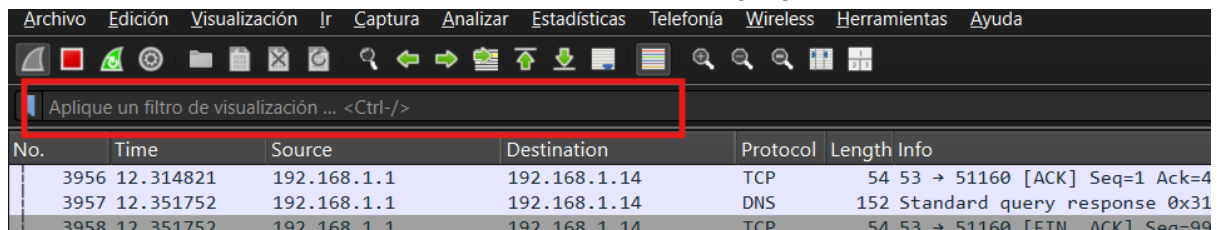


Al darle click a nuestra interfaz de red seleccionada, el programa automáticamente iniciará con el sniffado de aquella red, capturando así todo los paquetes que viajan por ella.



Ahora lo que podemos hacer es aplicar una serie de filtros, estos filtros nos sirven para poder buscar y analizar detalladamente la información que necesitamos y no tener que buscar uno por uno entre todos los paquetes capturados ya que así nos tardamos muchísimo tiempo más y por ende seríamos menos eficientes.

Bueno ahora si, una vez ya hecho el escaneo de la red para poder filtrar la información lo que hacemos es darle a el buscador de aqui que dice ***“Aplique un filtro de visualización”***



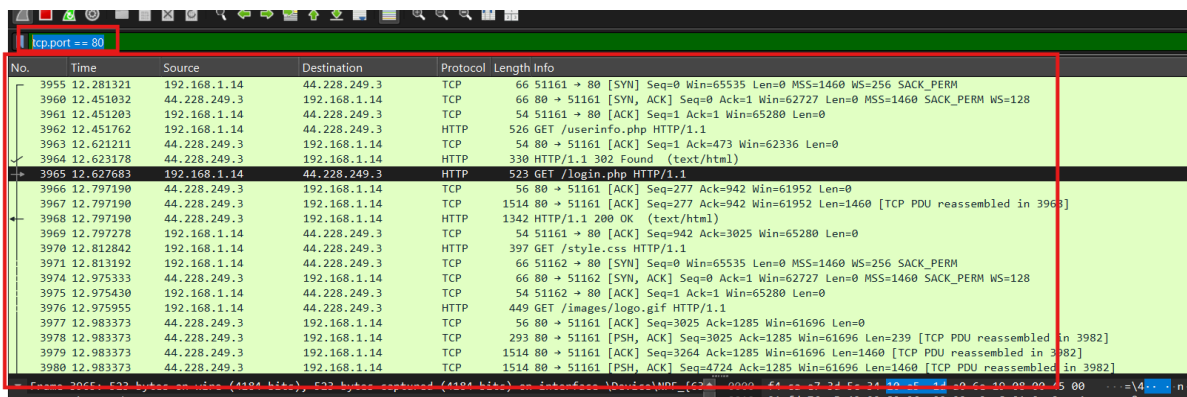
Y ahora la solo queda ir buscando y poner ciertos filtros dependiendo la información y lo que queramos buscas, como por ejemplo:

Este filtro nos sirve para mostrar solo el tráfico TCP que utiliza el puerto 80



Usar este filtro permite ver solicitudes y respuestas HTTP, como:

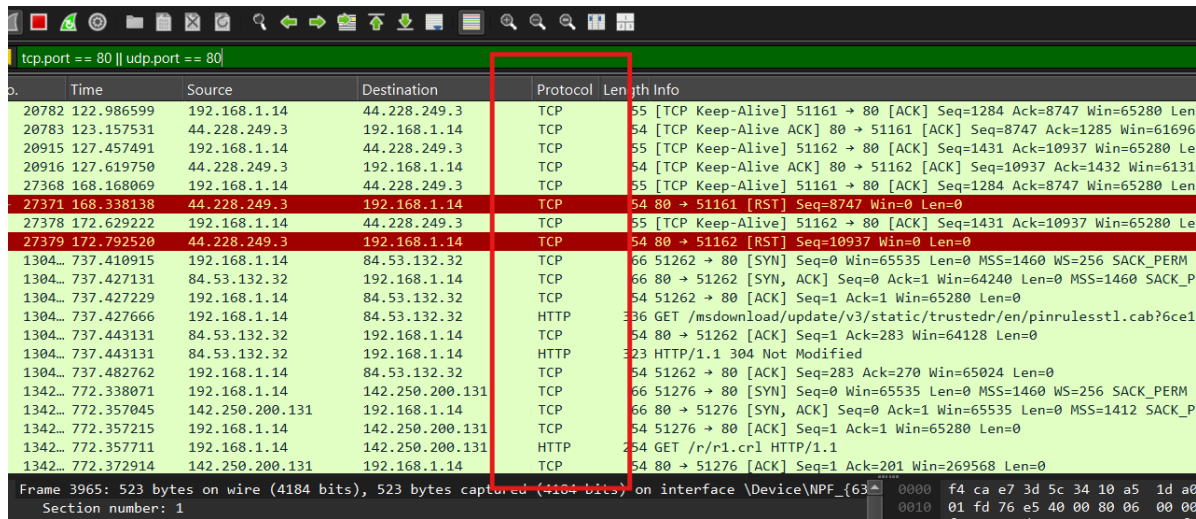
- Solicitudes GET, POST, PUT, DELETE.
- Respuestas HTTP con códigos de estado (200 OK, 404 Not Found, etc.).



También podrías combinar ciertos parámetros y así ir jugando con lo que nos venga bien y según la información que queramos filtrar.

```
tcp.port == 80 || udp.port == 80
```

Lo que hace este filtro es capturar tráfico que usa el puerto 80, ya sea por **TCP** o por **UDP**.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---------|------------|-----------------|-----------------|----------|--------|-------------------------------------------------------------------------|
| 20782 | 122.986599 | 192.168.1.14 | 44.228.249.3 | TCP | 55 | [TCP Keep-Alive] 51161 → 80 [ACK] Seq=1284 Ack=8747 Win=65280 Len=0 |
| 20783 | 123.157531 | 44.228.249.3 | 192.168.1.14 | TCP | 54 | [TCP Keep-Alive ACK] 80 → 51161 [ACK] Seq=8747 Ack=1285 Win=61696 Len=0 |
| 20915 | 127.457491 | 192.168.1.14 | 44.228.249.3 | TCP | 55 | [TCP Keep-Alive] 51162 → 80 [ACK] Seq=1431 Ack=10937 Win=65280 Len=0 |
| 20916 | 127.619750 | 44.228.249.3 | 192.168.1.14 | TCP | 54 | [TCP Keep-Alive ACK] 80 → 51162 [ACK] Seq=10937 Ack=1432 Win=6131 Len=0 |
| 27368 | 168.168069 | 192.168.1.14 | 44.228.249.3 | TCP | 55 | [TCP Keep-Alive] 51161 → 80 [ACK] Seq=1284 Ack=8747 Win=65280 Len=0 |
| 27371 | 168.338138 | 44.228.249.3 | 192.168.1.14 | TCP | 54 | 80 → 51161 [RST] Seq=8747 Win=0 Len=0 |
| 27378 | 172.629222 | 192.168.1.14 | 44.228.249.3 | TCP | 55 | [TCP Keep-Alive] 51162 → 80 [ACK] Seq=1431 Ack=10937 Win=65280 Len=0 |
| 27379 | 172.792520 | 44.228.249.3 | 192.168.1.14 | TCP | 54 | 80 → 51162 [RST] Seq=10937 Win=0 Len=0 |
| 1304... | 737.410915 | 192.168.1.14 | 84.53.132.32 | TCP | 66 | 51262 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 1304... | 737.427131 | 84.53.132.32 | 192.168.1.14 | TCP | 66 | 80 → 51262 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_P |
| 1304... | 737.427229 | 192.168.1.14 | 84.53.132.32 | TCP | 54 | 51262 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 1304... | 737.427666 | 192.168.1.14 | 84.53.132.32 | HTTP | 36 | GET /msdownload/update/v3/static/trusted/en/pinrulesstl.cab?6ce1 |
| 1304... | 737.443131 | 84.53.132.32 | 192.168.1.14 | TCP | 54 | 80 → 51262 [ACK] Seq=1 Ack=283 Win=64128 Len=0 |
| 1304... | 737.443131 | 84.53.132.32 | 192.168.1.14 | HTTP | 323 | HTTP/1.1 304 Not Modified |
| 1304... | 737.482762 | 192.168.1.14 | 84.53.132.32 | TCP | 54 | 51262 → 80 [ACK] Seq=283 Ack=270 Win=65024 Len=0 |
| 1342... | 772.338071 | 192.168.1.14 | 142.250.200.131 | TCP | 66 | 51276 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM |
| 1342... | 772.357045 | 142.250.200.131 | 192.168.1.14 | TCP | 66 | 80 → 51276 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_P |
| 1342... | 772.357215 | 192.168.1.14 | 142.250.200.131 | TCP | 54 | 51276 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 1342... | 772.357711 | 192.168.1.14 | 142.250.200.131 | HTTP | 254 | GET /r/r1.cr1 HTTP/1.1 |
| 1342... | 772.372914 | 142.250.200.131 | 192.168.1.14 | TCP | 54 | 80 → 51276 [ACK] Seq=1 Ack=201 Win=269568 Len=0 |

Frame 3965: 523 bytes on wire (4184 bits), 523 bytes captured (4104 bits) on interface \Device\NPF_{63...} Section number: 1

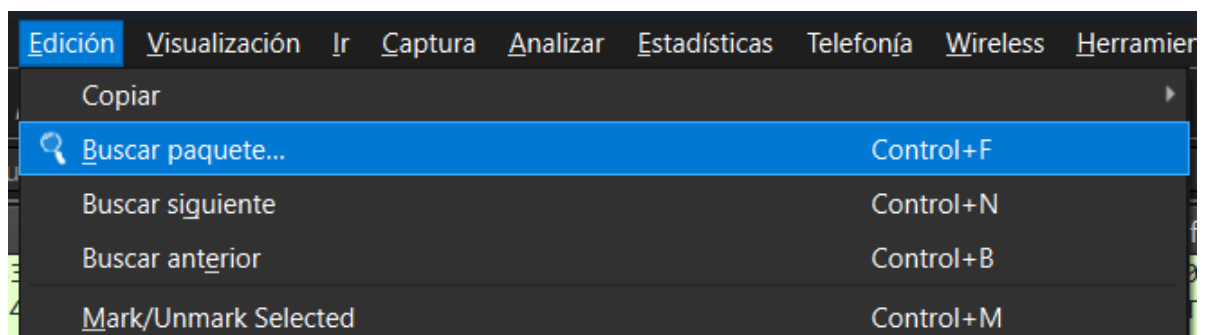
Este siguiente filtro nos sirve para analizar todo el tráfico de entrada y salida de una cierta dirección IP en concreto, esto nos permite ver desde los DNS hasta los UDP absolutamente todo lo que tenga que ver con la IP en cuestión.

```
ip.addr == 192.168.1.14
```

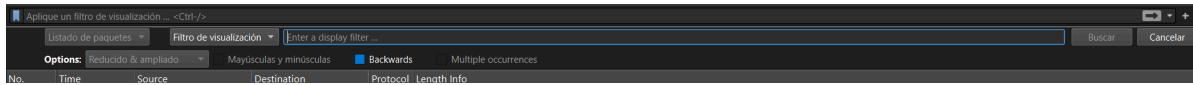
| ip.addr == 192.168.1.14 | | | | | |
|-------------------------|-----------|-----------------|-----------------|----------|-----------------------------------------------------------------|
| No. | Time | Source | Destination | Protocol | Length Info |
| 4083 | 17.515086 | 192.168.1.14 | 142.250.200.106 | TCP | 54 51165 → 443 [ACK] Seq=2566 Ack=7440 Win=65280 Len=0 |
| 4084 | 17.517007 | 192.168.1.14 | 142.250.200.106 | QUIC | 205 Protected Payload (KP0), DCID=eccccc1f823c41e9 |
| 4085 | 17.517155 | 192.168.1.14 | 142.250.200.106 | TLSv1.3 | 93 Application Data |
| 4086 | 17.535988 | 142.250.200.106 | 192.168.1.14 | QUIC | 1018 Protected Payload (KP0) |
| 4087 | 17.535988 | 142.250.200.106 | 192.168.1.14 | QUIC | 163 Protected Payload (KP0) |
| 4088 | 17.536319 | 192.168.1.14 | 142.250.200.106 | QUIC | 73 Protected Payload (KP0), DCID=eccccc1f823c41e9 |
| 4089 | 17.539610 | 142.250.200.106 | 192.168.1.14 | TCP | 56 443 → 51165 [ACK] Seq=7440 Ack=2605 Win=268032 Len=0 |
| 4090 | 17.557457 | 142.250.200.106 | 192.168.1.14 | QUIC | 66 Protected Payload (KP0) |
| 4097 | 22.696838 | 192.168.1.14 | 44.228.249.3 | TCP | 66 [TCP Retransmission] 51157 → 443 [SYN] Seq=0 Win=65535 Len=0 |
| 4098 | 22.946393 | 192.168.1.14 | 44.228.249.3 | TCP | 66 [TCP Retransmission] 51158 → 443 [SYN] Seq=0 Win=65535 Len=0 |
| 4099 | 23.512843 | 192.168.1.14 | 216.58.215.174 | UDP | 71 60691 → 443 Len=29 |
| 4100 | 23.531586 | 216.58.215.174 | 192.168.1.14 | UDP | 68 443 → 60691 Len=26 |
| 4101 | 26.851546 | 192.168.1.14 | 212.232.108.140 | UDP | 1287 59909 → 443 Len=1245 |
| 4102 | 26.851672 | 192.168.1.14 | 212.232.108.140 | UDP | 1292 59909 → 443 Len=1250 |
| 4103 | 26.851713 | 192.168.1.14 | 212.232.108.140 | UDP | 759 59909 → 443 Len=717 |
| 4104 | 26.856442 | 212.232.108.140 | 192.168.1.14 | UDP | 193 443 → 59909 Len=151 |
| 4105 | 26.857840 | 192.168.1.14 | 212.232.108.140 | UDP | 84 59909 → 443 Len=42 |
| 4106 | 26.857906 | 212.232.108.140 | 192.168.1.14 | UDP | 1287 443 → 59909 Len=1245 |
| 4107 | 26.857906 | 212.232.108.140 | 192.168.1.14 | UDP | 1292 443 → 59909 Len=1250 |
| 4108 | 26.857906 | 212.232.108.140 | 192.168.1.14 | UDP | 1292 443 → 59909 Len=1250 |

Pero bueno, con Wireshark no solo se puede filtrar cosas desde aquí, también podemos filtrar cosas desde otra parte, como lo vamos a ver a continuación.

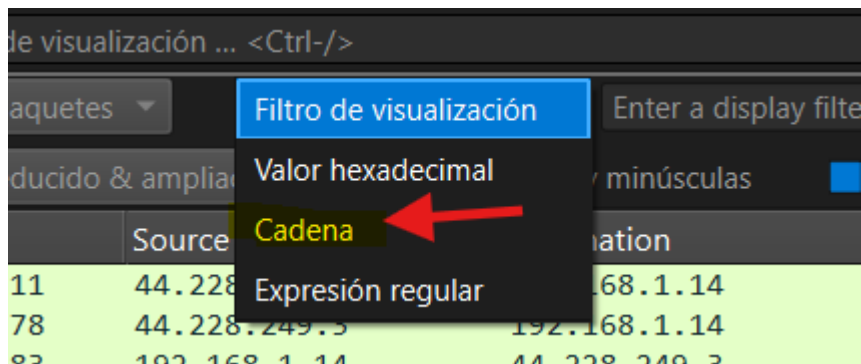
Vamos a la casilla “Edición”, que se encuentra en la parte superior izquierda, una vez estemos ahí le vamos a dar en “Buscar paquete” y le damos click como estamos viendo en pantalla



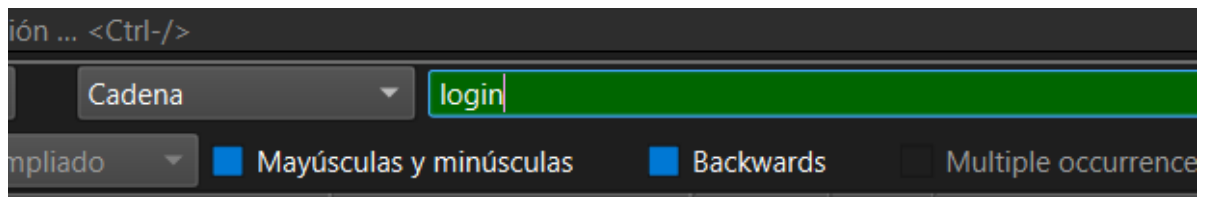
Una vez ya le hayamos dado click a “Buscar paquete” nos aparecerá una nueva barra, como estamos viendo en la imagen



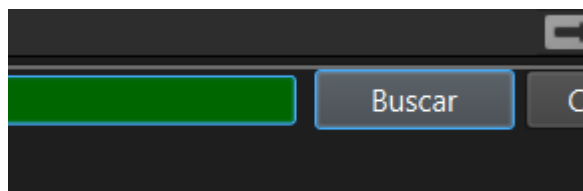
En esa barra vamos a ir a el apartado donde dice ”Filtro de visualización” y seleccionaremos Cadena



Lo que hace la cadena es que coje una palabra la que le pongamos y la busca entre todo lo capturado anteriormente para ver si tiene algo entre todos los paquetes que contenga nuestra palabra, esto es muy potentes ya que podemos poner cadenas como “**login**, **password**, **user**” o así que en barra que tenemos a el lado escribiremos una palabra que queramos que busque en este caso puse “**login**”



Y luego le damos a “**Buscar**”



Una vez le damos podemos ver que automáticamente nos arroja el paquete que contiene la cadena que le pusimos en este caso **“login”**

| Listado de paquetes | | | | |
|-------------------------------------------------------------------------------------|--------------|--------------|----------|----------------------------------------------------------------------------------------|
| Cadena | | | | |
| login | | | | |
| Options: Reducido & ampliado Mayúsculas y minúsculas Backwards Multiple occurrences | | | | |
| Time | Source | Destination | Protocol | Length Info |
| 3956 12.314821 | 192.168.1.1 | 192.168.1.14 | TCP | 54 53 → 51160 [ACK] Seq=1 Ack=41 Win=29248 Len=0 |
| 3957 12.351752 | 192.168.1.1 | 192.168.1.14 | DNS | 152 Standard query response 0x3149 HTTPS testphp.vulnweb.com SOA ns1.eurodns.com |
| 3958 12.351752 | 192.168.1.1 | 192.168.1.14 | TCP | 54 53 → 51160 [FIN, ACK] Seq=99 Ack=41 Win=29248 Len=0 |
| 3959 12.351853 | 192.168.1.14 | 192.168.1.1 | TCP | 54 51160 → 53 [RST, ACK] Seq=41 Ack=99 Win=0 Len=0 |
| 3960 12.451032 | 44.228.249.3 | 192.168.1.14 | TCP | 66 80 → 51161 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 SACK_PERM WS=128 |
| 3961 12.451203 | 192.168.1.14 | 44.228.249.3 | TCP | 54 51161 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 3962 12.451762 | 192.168.1.14 | 44.228.249.3 | HTTP | 526 GET /userinfo.php HTTP/1.1 |
| 3963 12.621211 | 44.228.249.3 | 192.168.1.14 | TCP | 54 80 → 51161 [ACK] Seq=1 Ack=473 Win=62336 Len=0 |
| 3964 12.623178 | 44.228.249.3 | 192.168.1.14 | HTTP | 330 HTTP/1.1 302 Found (text/html) |
| 3965 12.627683 | 192.168.1.14 | 44.228.249.3 | HTTP | 523 GET /login.php HTTP/1.1 |
| 3966 12.797190 | 44.228.249.3 | 192.168.1.14 | TCP | 56 80 → 51161 [ACK] Seq=277 Ack=942 Win=61952 Len=0 |
| 3967 12.797190 | 44.228.249.3 | 192.168.1.14 | TCP | 1514 80 → 51161 [ACK] Seq=277 Ack=942 Win=61952 Len=1460 [TCP PDU reassembled in 3968] |

| | | |
|-------------|------|-------------------------------------------------------------------|
| 92.168.1.1 | TCP | 54 51160 → 53 [RST, ACK] Seq=41 Ack=99 Win=0 Len=0 |
| 92.168.1.14 | TCP | 66 80 → 51161 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1460 S |
| 4.228.249.3 | TCP | 54 51161 → 80 [ACK] Seq=1 Ack=1 Win=65280 Len=0 |
| 4.228.249.3 | HTTP | 526 GET /userinfo.php HTTP/1.1 |
| 92.168.1.14 | TCP | 54 80 → 51161 [ACK] Seq=1 Ack=473 Win=62336 Len=0 |
| 92.168.1.14 | HTTP | 330 HTTP/1.1 302 Found (text/html) |
| 4.228.249.3 | HTTP | 523 GET /login.php HTTP/1.1 |
| 92.168.1.14 | TCP | 56 80 → 51161 [ACK] Seq=277 Ack=942 Win=61952 Len=0 |
| 92.168.1.14 | TCP | 1514 80 → 51161 [ACK] Seq=277 Ack=942 Win=61952 Len=1460 [TCP PDU |

Y así es cómo hacemos uso de los filtros para poder buscar información de forma eficiente de nuestro esnifado de red. CHAO Y GRACIAS 😊