Telefonica

# WIRESHARK

## Paquetes capturados

# Trabajando con paquetes capturados

## Paquetes capturados

# Trabajar con paquetes capturados

## Paquetes capturados

Una vez que hayamos capturado tráfico o se haya abierto un archivo de captura, podemos acceder a los detalles de cada paquete seleccionando el paquete en la lista de paquetes

También podemos ver detalles de un paquete en una captura en tiempo real (**Preferences → Capture → Update list of packets in real time**)

Podemos ver el paquete en una nueva ventana haciendo doble click o seleccionando con el botón derecho del ratón **Show packet in new window**

# Trabajar con paquetes capturados

## Paquetes capturados

Podemos marcar paquetes en el panel lista de paquetes, este, se mostrará con fondo negro.

Las marcas de paquetes, no se almacenan en el archivo de captura y se perderan cuan do el archivo de captura sea cerrado

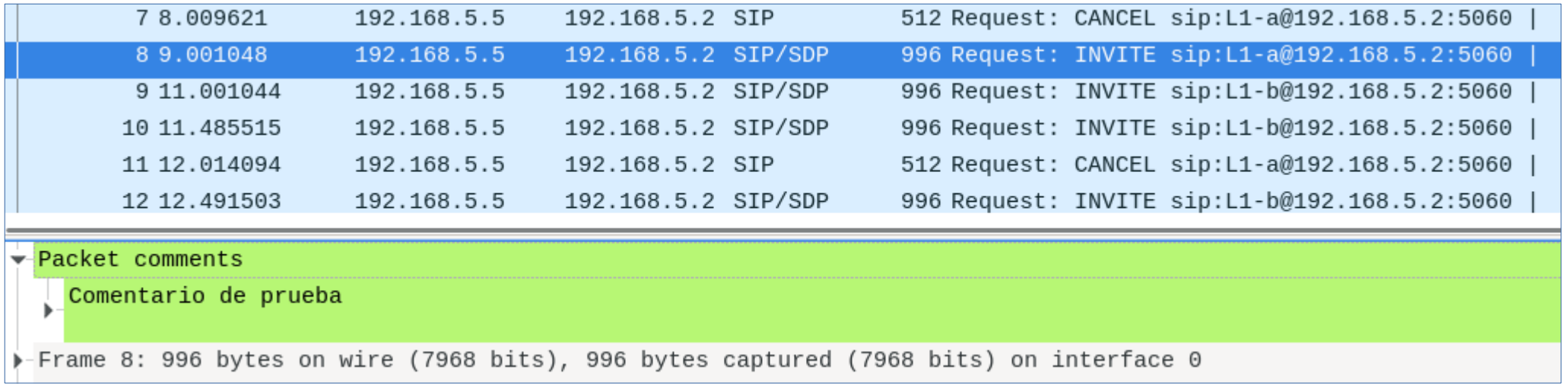
Podemos seleccionar los paquetes marcados con un filtro

```
frame.marked == True
```

# Trabajar con paquetes capturados

## Paquetes capturados

Podemos agregar un comentario a un sólo paquete, nos aparecerá en el detalle del paquete una entrada con el comentario que hemos escrito



# Trabajando con paquetes capturados

## Filtrado de paquetes

# Trabajar con paquetes capturados

## Filtrado de paquetes

Los filtros de visualización, se aplican después de capturar datos y nos permite, mediante un potente lenguaje, construir expresiones para seleccionar ciertos paquetes que cumplan con la regla de filtrado

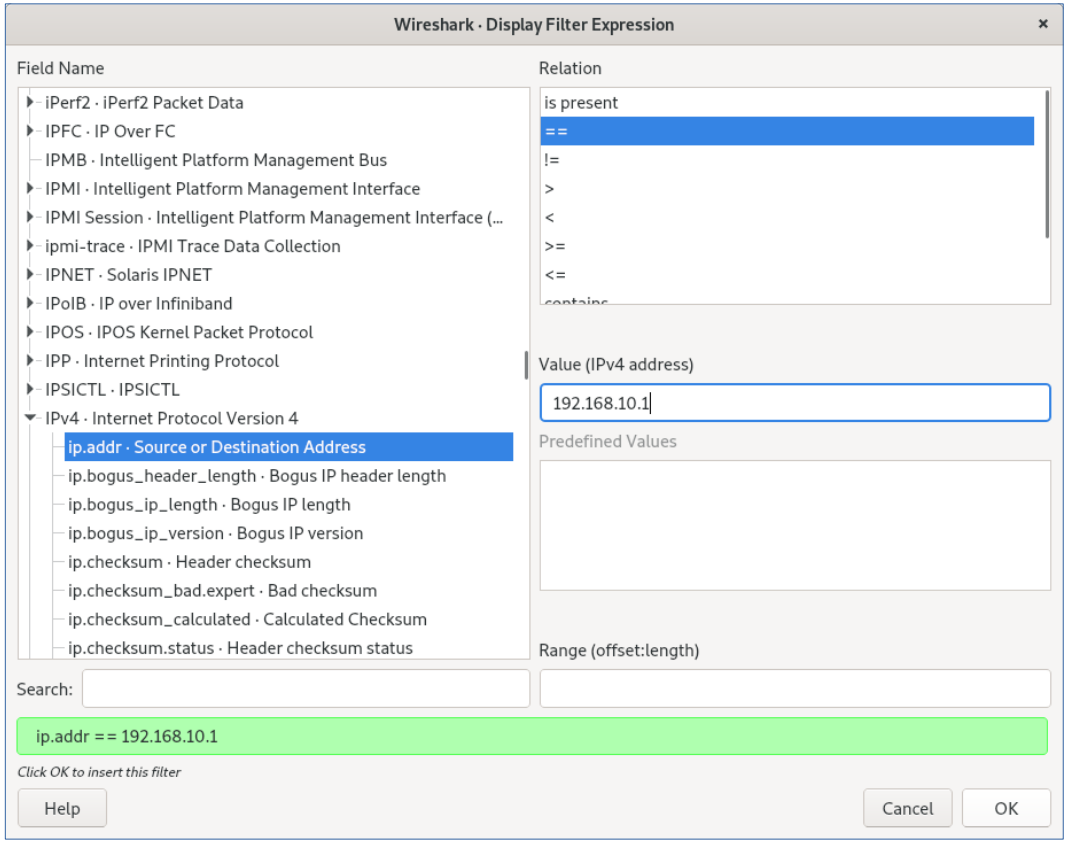
- Parámetros, como la dirección IP, una red concreta, puertos, etc..
- Protocolos y datos de diversas aplicaciones
- La presencia de un campo o un valor concreto
- Condiciones, como la longitud de un paquete, existencia de flags
- Fenómenos, como retransmisiones TCP, ACKs duplicados, etc...

# Trabajar con paquetes capturados

## Filtrado de paquetes

A la hora de configurar filtros, podemos elegir entre varias opciones

- Elegir desde el menú los filtros guardados
- Escribir la sintaxis directamente en la entrada de filtrado
- Elegir un parámetros en el panel de detalle del paquete y definirlo como filtro
- Construir un filtro mediante el botón **Expression**



# Trabajar con paquetes capturados

## Filtrado de paquetes

En general, un filtro de visualización, es una expresión formada por primitivas, conectadas por operadores

```
[not] Expresión [and|or] [not] Expresión
ip.addr == 10.15.100.1 and sip
```

| | |
|---|---|
| && and | ip.src == 10.10.60.1 && tcp.port == 80 |
| \|\| or | tcp.port == 80 \|\| tcp.port == 443 |
| ! not | !arp && not icmp |

# Trabajar con paquetes capturados

## Filtrado de paquetes

### Tipos de campos

- Unsigned integer, se pueden expresar enteros en decimal, octal o hexadecimal
- Booleanos
- Direcciones ethernet, 6 bytes separados por dos puntos
- Direcciones IP, también admite la notación CIDR para las redes

```
ip.len <= 1500
tcp.flag.syn
Eth.dst == ff:ff:ff:ff:ff:ff
ip.addr == 192.168.10.0/24
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

### Tipos de campos y II

- Cadenas de texto, se encierran entre comillas
- Slice operator, para seleccionar parte de un array
- Operador de pertenencia
- Funciones, disponemos de un número de funciones para convertir campos (upper, lower, string, len)

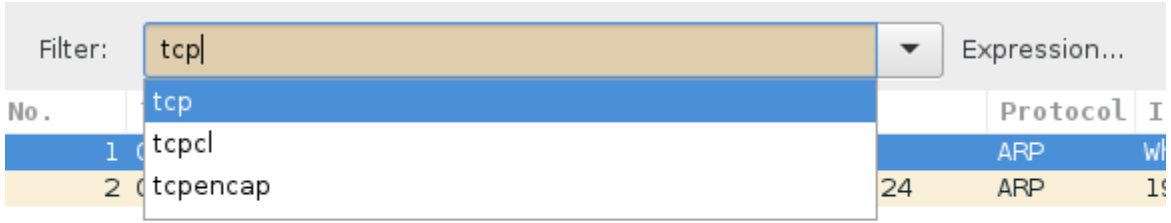
```
http.request.uri == “https://www.wireshark.org”
tcp.port in {80 443 8080}
lower(http.server) contains “apache”
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

Podemos escribir directamente la sintaxis del filtro en el cuadro de edición de filtros de la barra de herramientas

Cuando se teclea en el área de edición, wireshark, abre una ventana de autocompletado para las opciones de filtrado



# Trabajar con paquetes capturados

## Filtrado de paquetes

Cuando empezamos a escribir en el área de edición de filtros, se colorea el fondo del campo de edición

- Fondo rojo, indica que la sintaxis no es correcta
- Fondo verde, sintaxis correcta
- Fondo amarillo, sintaxis correcta, pero puede que no obtenga los resultados esperados

# Trabajar con paquetes capturados

## Filtrado de paquetes

Wireshark colorea el área de filtro amarillo, cada vez que se utiliza el operador **!=** en expresiones combinadas como **eth.addr**, **ip.addr**, **tcp.port**, etc.

```
ip.addr != 10.15.100.1
```

Esta expresión se evaluará como verdadera, cuando al menos una de las dos direcciones difiera de 10.15.100.1. Lo correcto, sería

```
!(ip.addr == 10.15.100.1)
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

Hay casos en los que la negación de un valor, dará un valor erróneo. Se interpreta como “mostrar todos los paquetes que no tengan un conjunto de bits SYN TCP a 1”, otros paquetes como ARP o UDP, cumplen con este filtro

```
!(tcp.flags.sym == 1)
```

Este filtro es correcto, sólo visualiza paquetes TCP cuyo flag SYN sea distinto de 1

```
tcp.flags.syn != 1
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

Podemos seleccionar los filtros de visualización que hemos configurado, desde el menú **Analyze → Display Filters** o pulsando el icono a la izquierda del cuadro de edición de filtros

Nos despliega una ventana con los filtros que se encuentran en el archivo **dfilters**.

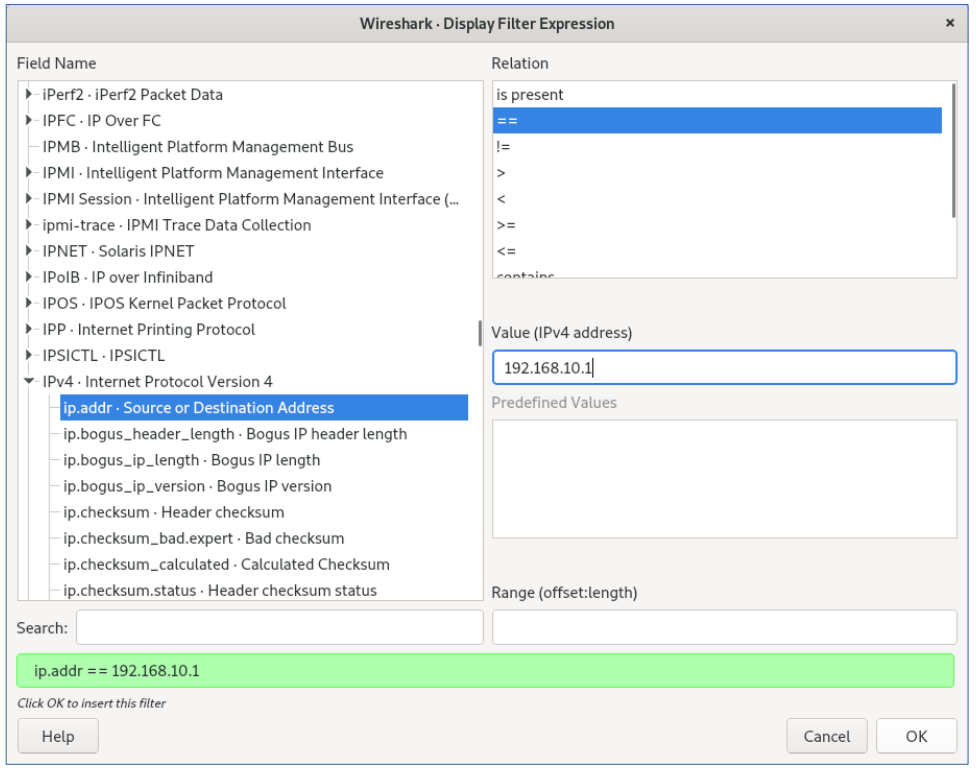
Desde esta ventana, accedemos a la opción de gestión **Manage Display Filters**

# Trabajar con paquetes capturados

## Filtrado de paquetes

Si pulsamos el botón Expression, se abre una ventana de diálogo donde podemos ir construyendo el filtro a medida

- **Field name**, elegimos protocolo y parámetro
- **Relation**, operador que se ajustará al parámetro
- **Value**, valor para ese parámetro
- **Valores predefinidos**, para campos seleccionados, no todos los campos tienen esa ayuda
- **Range**, parámetros de longitud

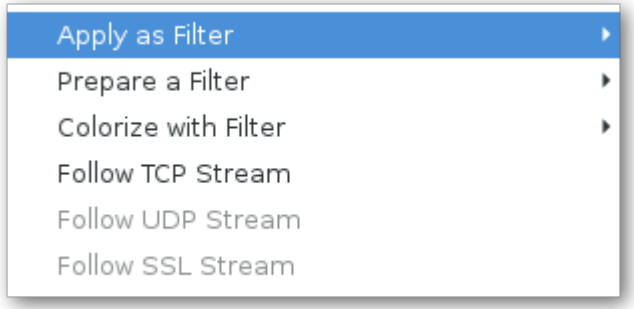


# Trabajar con paquetes capturados

## Filtrado de paquetes

También podemos seleccionar un campo en el panel de detalle del paquete y con el botón derecho, definirlo como un filtro

- Apply as filter, seleccionamos el factor booleano (selectedy not selected) y se ejecuta el filtro
- Prepare a filter, construimos el filtro y se queda en el campo de edición para seguir editándolo.



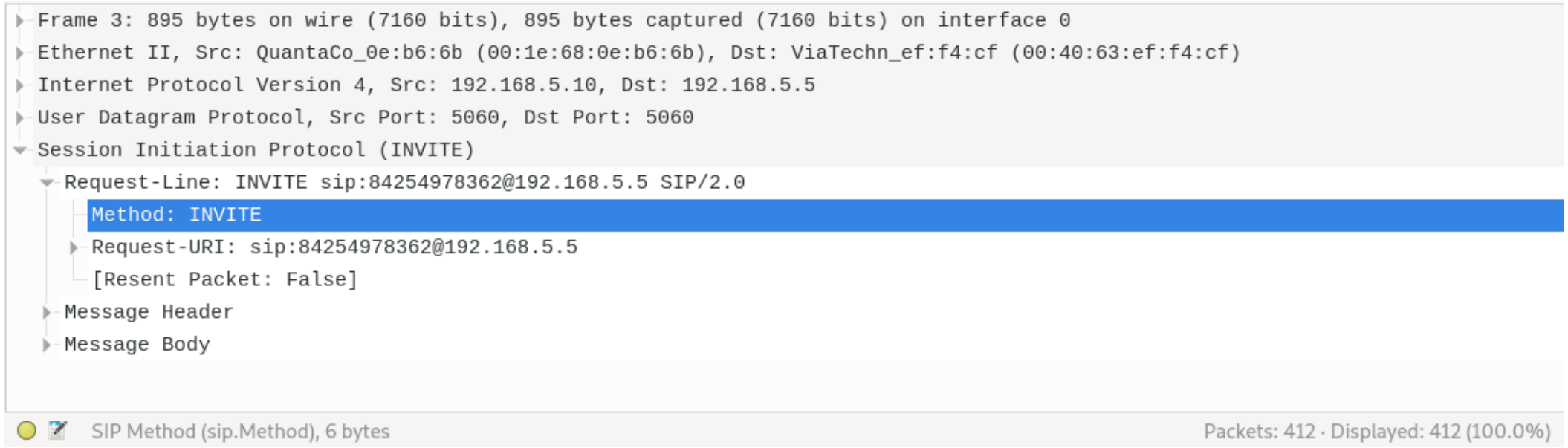

```
# Selected
sip.Method == “INVITE”

# Not selected
!(sip.Method == “INVITE”)
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

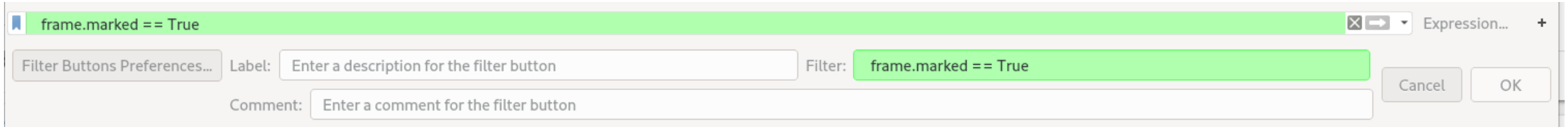
Una vez que hemos seleccionado un campo en el detalle de paquete, nos aparecerá en la barra de estado, la sintaxis del filtro correspondiente a ese campo



# Trabajar con paquetes capturados

## Filtrado de paquetes

Una vez seleccionado un filtro en el campo de edición de filtros, podemos crear un botón en la barra de herramientas seleccionando el icono **[+]**, nos aparecerá un cuadro de diálogo donde seleccionaremos una etiqueta que defina el botón




Podemos gestionar los botones desde **Edit → Preferences → Filter Buttons**, dónde gestionamos el archivo **dfilter_buttons**

# Trabajar con paquetes capturados

## Filtrado de paquetes

Para aplicaciones que usen TCP como protocolo de transporte, es más eficiente filtrar por número de puerto

```
http
tcp.port == 80
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

Aplicando filtros basados en direcciones IP,, rango de direcciones o subredes

| Una sola dirección IP |
| --- |
| ip.src o ip.dst<br>ip.addr o ip.host |
| **Un rango de direcciones** |
| ip.addr > 192.168.100.100 && ip.addr < 192.168.100.254 |
| **Una subnet** |
| ip.addr == 192.168.100.0/24 |

# Trabajar con paquetes capturados

## Filtrado de paquetes

Hay ocasiones en la que se quiere buscar una palabra en particular, por ejemplo “admin”

- Uso de **contains** para filtrar una trama entera
- Uso de **contains** para filtrar por el contenido de un campo
- Wireshark es case-sensitive, si queremos buscar “admin” o “Admin”, podemos utilizar expresiones regulares

```
http.server contains “apache”
ftp.request.arg matches “(A|a)dmin”
```

# Trabajar con paquetes capturados

## Filtrado de paquetes

Wireshark soporta expresiones regulares PCRE (Perl Compatible Regular Expression) en los filtros de visualización

- ^ y $ → principio y final de línea
- () → grupos de propuesta, | alternancia
- [] -> rangos
- *, +, ? → Coincide en 0 o más, 1 o mas y 1 o 0

# Filtrado

## Laboratorio 1

### Archivo de sesión HTTP

- Situar el nombre de host como columna
- Cómo podemos ver todas las peticiones HTTP que se realizan
- Qué filtro hay que utilizar si queremos buscar un host en particular
- Y si queremos ver las consultas DNS
- Guardar filtro en dfilters, crear un botón genérico y comprobar en archivos de configuración

# Filtrado

## Laboratorio 2

### Archivo de sesión HTTP

- Filtrar errores DNS o errores 404 Response
- Y si se quiere filtrar por cualquier error HTTP

# Trabajando con paquetes capturados

## Reglas de coloreado

# Trabajar con paquetes capturados

## Reglas de coloreado

Wireshark colorea los paquetes capturados basándose en una regla que aplica un filtro de visualización

Hay dos tipos de reglas de coloreado

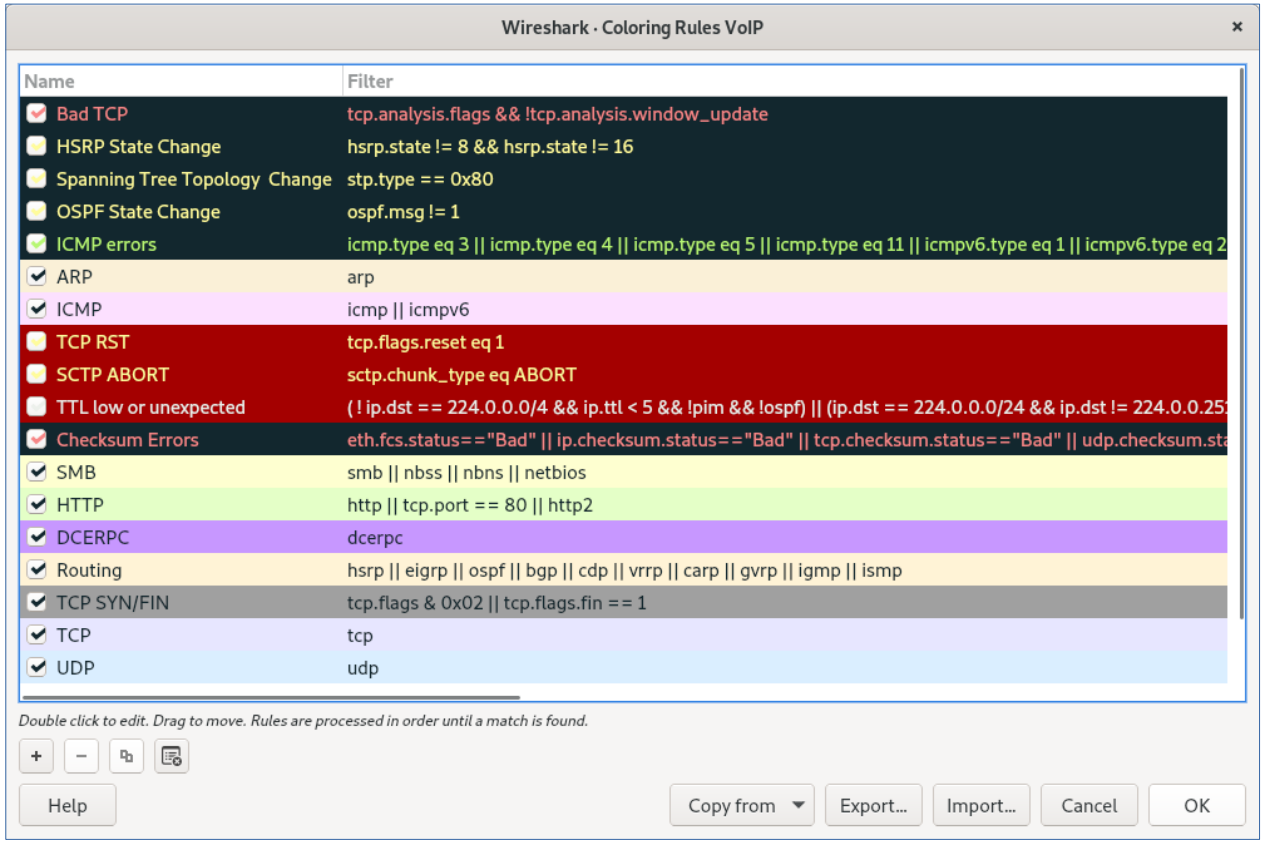
- Reglas temporales que sólo están vigentes hasta que se salga del programa
- Reglas permanentes que se guardan en un archivo de preferencias

# Trabajar con paquetes capturados

## Reglas de coloreado

Para colorear permenentemente los paquetes, seleccione el menú V **View → Coloring Rules**, donde podremos editar, crear y eliminar reglas de coloreado

Cada regla, está compuesta de un literal, un color de fondo, un color de primer plano y una regla de filtrado

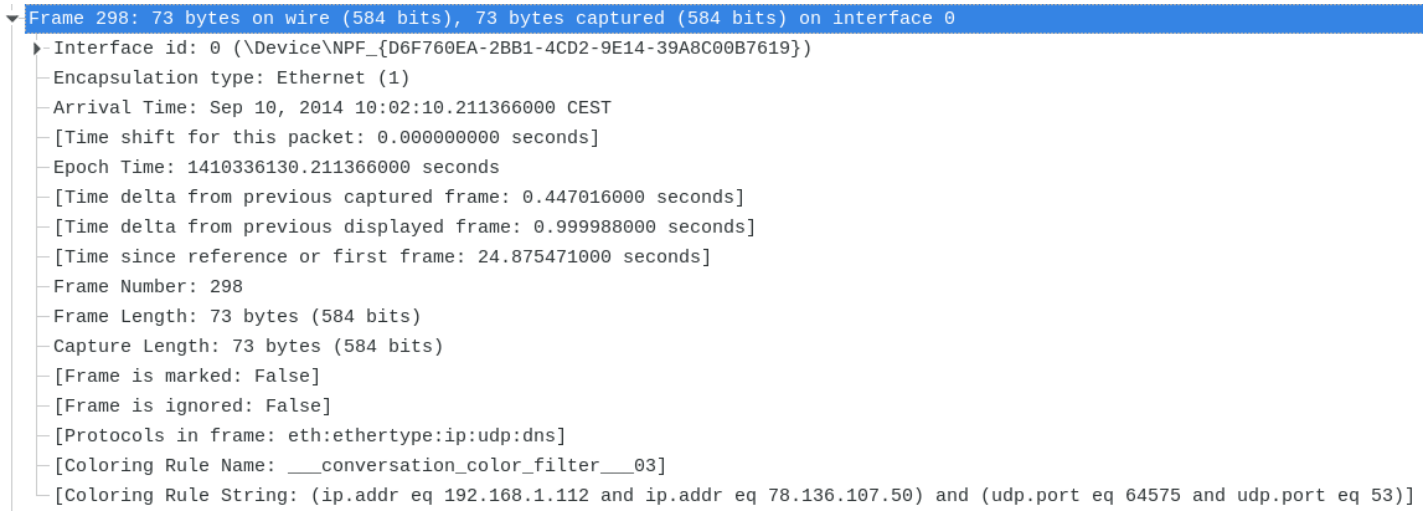


# Trabajar con paquetes capturados

## Reglas de coloreado

En el detalle del paquete, aparece una variable con la regla de coloreado que se ha aplicado

- Coloring rule name, el nombre que se aplica a la regla
- Coloring rule string, la regla de filtrado que se ejecuta



# Trabajar con paquetes capturados

## Reglas de coloreado

Por lo general, se deben enumerar las reglas más específicas antes que las reglas más generales. Por ejemplo, si tenemos una regla UDP antes que una DNS, la regla DNS puede que no se aplique, ya que normalmente

# Trabajar con paquetes capturados

## Reglas de coloreado

Se pueden realizar reglas de coloreado para separar visiblemente conversaciones en la lista de paquetes

Para colorear temporalmente una conversación Tcp, seleccione paquete y con el botón derecho del ratón, seleccione Colorize **Conversation → TCP → Color x**

# Filtrado

## Laboratorio 3

### Archivo FTP

- Construya una regla de coloreado para resaltar nombres de usuario y password en FTP
- Colorear el filtro con fondo rojo y con primer plano en blanco y si el usuario es administrador, cambiar el color del texto

# Trabajando con paquetes capturados

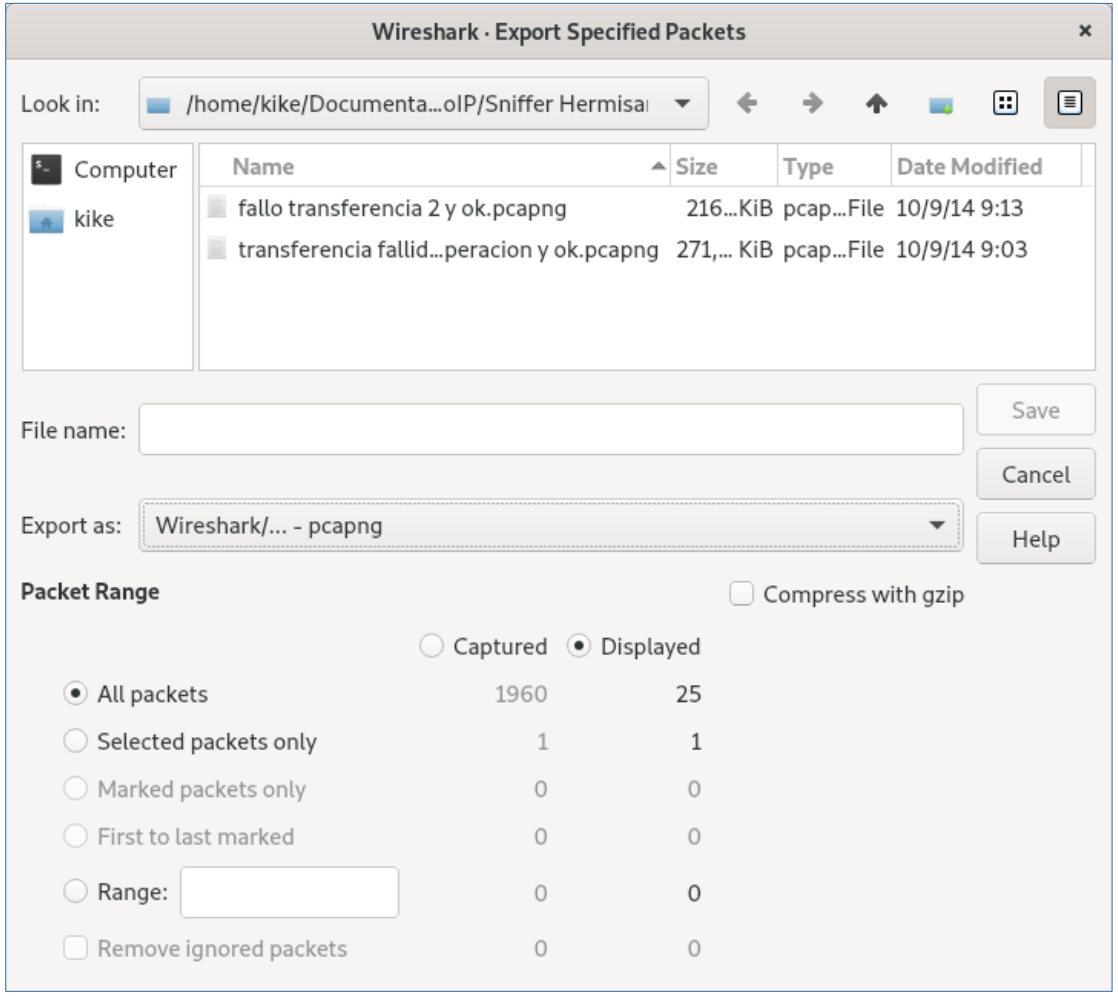
## Exportando paquetes

# Trabajar con paquetes capturados

## Exportando paquetes

Wireshark proporciona varias formas y formatos para exportar paquetes de datos

Desde **File → Export Especified Packets**, podemos exportar todo el archivo de captura o sólo los paquetes visualizados por un filtro seleccionado



# Trabajar con paquetes capturados

## Exportando paquetes

Cuando queremos exportar paquetes y cambiarlos de formato, elegimos File → Export Packets Dissections. Los formatos que podemos elegir

- Texto Plano
- Csv
- Xml
- JSON

# Filtrado

## Laboratorio 4

### Archivo HTTP

- Exportar una conversación TCP que se descargen archivos “.exe” y seleccionamos sólo esa conversación

# Filtrado

## Laboratorio 5

### Archivo HTTP

- Exportar lista de valores de campo HTTP que contenga el nombre de host, en formato CSV.
- Se puede utilizar cualquier herramienta para análisis posterior


Telefónica