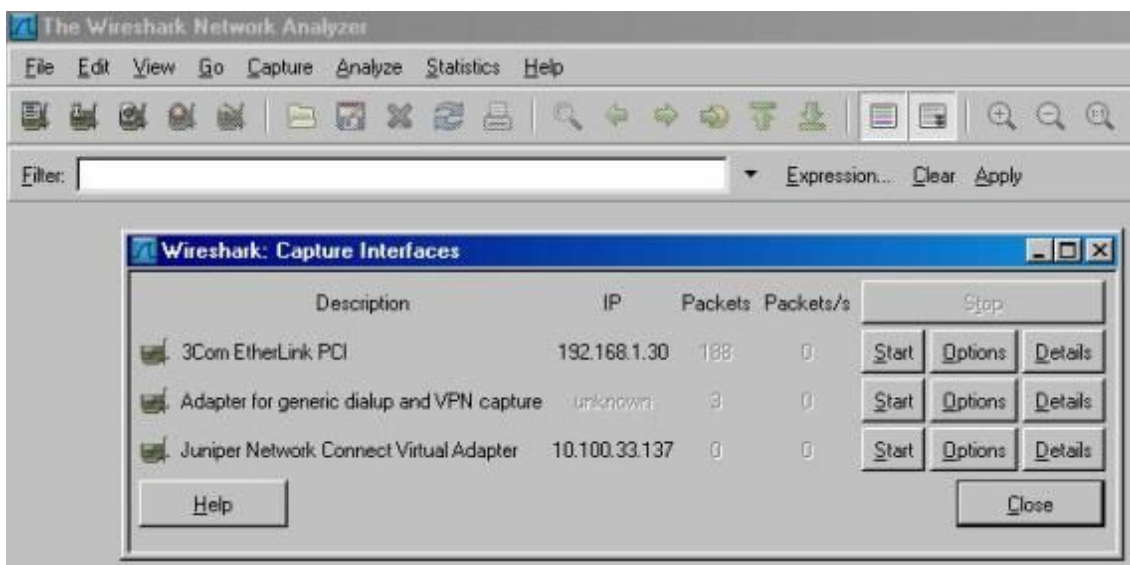


Wireshark

Wireshark es una herramienta de código abierto, multiplataforma, de análisis de red.

Nos permite capturar y analizar el tráfico que llega a nuestro sistema. Funciona al igual que lo puede hacer cualquier otro sniffer tal como Windump, TCPDump ó dsniff. Pero, al contrario de estos, lo hace mostrando los datos a través de un entorno gráfico y de forma más amigable y entendible.

Antes que nada, tras arrancar Wireshark, el menú Capture > Interfaces.... nos muestra la siguiente pantalla:



Solo tendremos que pulsar en Start para capturar a través de la interface que nos interese. Inmediatamente Wireshark comienza a capturar.

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
19	20.973509	192.168.1.1	224.0.0.1	IGMP	v2 Membership Query
20	17.053535	Dell_5f:a9:25	Broadcast	ARP	who has 192.168.1.12? Tell 192.1
21	17.482221	192.168.1.97	239.255.255.250	IGMP	v2 Membership Report
22	17.530214	192.168.1.30	224.0.0.9	IGMP	v2 Membership Report
23	18.192513	217.126.75.222	192.168.1.30	TCP	[TCP segment of a reassembled PDI
24	18.192808	217.126.75.222	192.168.1.30	TCP	[TCP segment of a reassembled PDI

Frame 23 (673 bytes on wire (673 bytes captured))
Arrival Time: Feb 14, 2008 11:00:21.113714000
[Time delta from previous captured frame: 0.662299000 seconds]
[Time delta from previous displayed frame: 0.662299000 seconds]
[Time since reference or first frame: 18.192513000 seconds]
Frame Number: 23
Frame Length: 673 bytes
Capture Length: 673 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:tcp:http]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80]

Ethernet II, Src: Dell_5f:a9:25 (00:14:22:5f:a9:25), Dst: 3Com_ed:89:c3 (00:04:75:ed:89:c3)
Destination: 3Com_ed:89:c3 (00:04:75:ed:89:c3)
Source: Dell_5f:a9:25 (00:14:22:5f:a9:25)
Type: IP (0x0800)

Internet Protocol, Src: 217.126.75.222 (217.126.75.222), Dst: 192.168.1.30 (192.168.1.30)

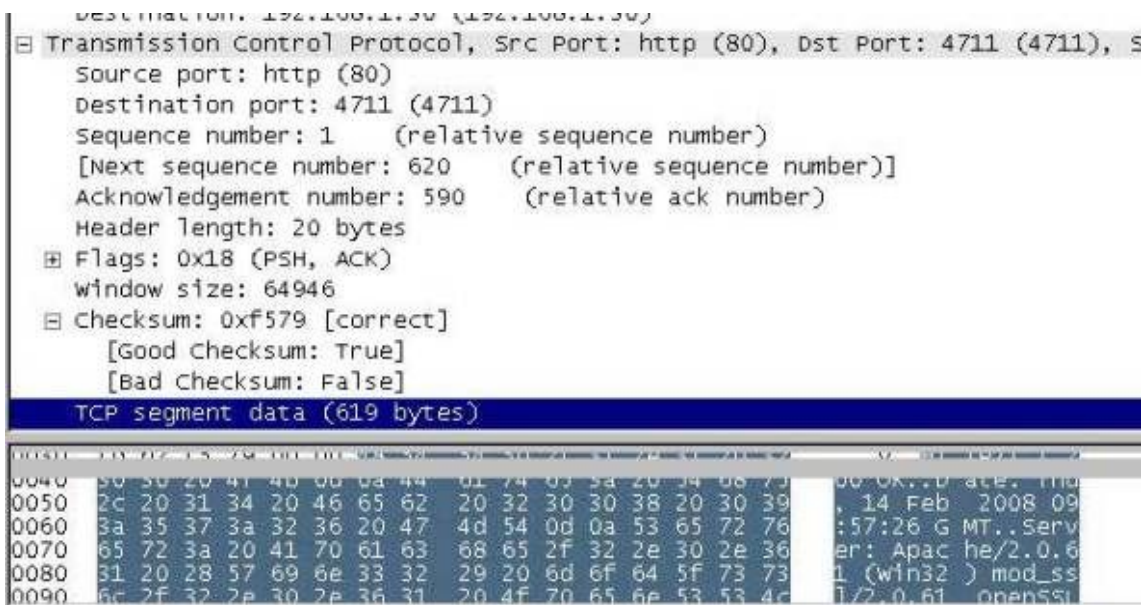
Transmission Control Protocol, Src Port: http (80), Dst Port: 4711 (4711), Seq: 1, Ack: 590, Len: 619
Source port: http (80)
Destination port: 4711 (4711)
Sequence number: 1 (relative sequence number)
[Next sequence number: 620 (relative sequence number)]
Acknowledgement number: 590 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
Window size: 64946
Checksum: 0x6570 [correct]

0000 00 04 75 ed 89 c3 00 14 22 5f a9 25 08 00 45 00
0010 02 93 e1 8f 00 00 80 06 6f b2 09 7e 4b 0a c0 a8
0020 01 1e 00 50 12 67 21 9e b2 a5 99 28 f1 c8 50 18
0030 fd b2 f5 79 00 00 48 54 54 50 2f 31 2e 31 20 32 ...P.g!...
0040 30 30 20 4f 4b 0d 0a 44 61 74 65 3a 20 54 68 75 ...y..HT TP/1.1.2
0050 2c 20 31 34 20 46 65 62 20 32 30 30 38 20 30 39 00 OK..D ate: Thu
0060 3a 35 37 3a 32 38 20 47 4d 54 0d 0a 53 65 72 76 , 14 Feb 2008 09
0070 65 72 3a 20 41 70 61 63 68 65 2f 32 2e 30 2e 36 :57:26 G MT..Serv
0080 31 20 28 57 69 6e 33 32 29 20 6d 6f 64 5f 73 73 er: Apac he/2.0.6
0090 6c 2f 32 2e 30 2e 36 31 20 4f 70 65 6e 53 53 4c 1 (Win32) mod_ss
00a0 2f 30 2e 39 2e 37 6d 20 50 48 50 2f 34 2e 33 2e /2.0.61 OpenSSL
00b0 31 31 0d 0a 58 2d 50 6f 77 65 72 65 64 2d 42 79 /0.9.7m PHP/4.3.
00c0 3a 20 50 48 50 2f 34 2e 34 2e 37 0d 0a 53 65 74 11..X-Po wered-By
00d0 2d 43 6f 6f 6b 69 65 3a 20 70 68 70 62 62 32 6d : PHP/4. 4.7..set
00e0 79 73 71 6c 5f 64 61 74 61 3d 61 25 33 41 32 25 -Cookie: phpb2m
00f0 33 41 25 37 42 73 25 33 41 31 31 25 33 41 25 32 ysq!_dat a=a%3A2%
0100 32 61 75 74 6f 6c 6f 67 69 6e 69 64 25 32 32 25 3AX7B%3A11%3A%2
0110 33 42 73 25 33 41 33 32 25 33 41 25 32 32 63 32 2autolog 1nid%22%
0120 65 61 36 30 30 30 66 38 36 36 32 64 64 33 37 63 3B%3A32 %3A%22c2
0130 63 33 63 35 38 63 34 65 30 63 37 61 66 61 25 32 ea6000f8 662dd37c
0140 32 25 33 42 73 25 33 41 36 25 33 41 25 32 32 75 c3c58c4e 0c7afa%2
0150 73 65 72 69 64 25 32 32 25 33 42 69 25 33 41 34 2%3B%3A 6%3A%22u
serid%22 %3B1%3A4

Se establecen 3 zonas de datos. La primera es la zona de listado de los paquetes capturados con información del Número de Frame, tiempo en segundos de la captura, Origen, Destino, protocolo involucrado y por último un campo de información extra que previamente Wireshark a decodificado.

La segunda zona muestra los datos del Frame capturado.

- ❑ Frame 23 (los numera secuencialmente). Nos da información sobre la hora de llegada el tamaño, etc.
- ❑ Ethernet II nos muestra la cabecera Ethernet II que a su vez pertenece a la capa de enlace de datos. Donde podemos identificar MAC origen y MAC destino.
- ❑ Transmission Control Protocol. (TCP): Puerto origen, puerto destino, flags, etc.
- ❑ TCP Segment Data, con todo el contenido del campo Data del segmento TCP.



Filtros

Wireshark contempla dos tipos de Filtros. Filtros de captura y Filtros de visualización. En Wireshark para los filtros de captura podemos hacer uso de los mismos filtros de TCPDump / Windump, ya que usa la misma librería pcap.

Los filtros de captura (Capture Filter) son los que se establecen para definir qué paquetes vamos a capturar.

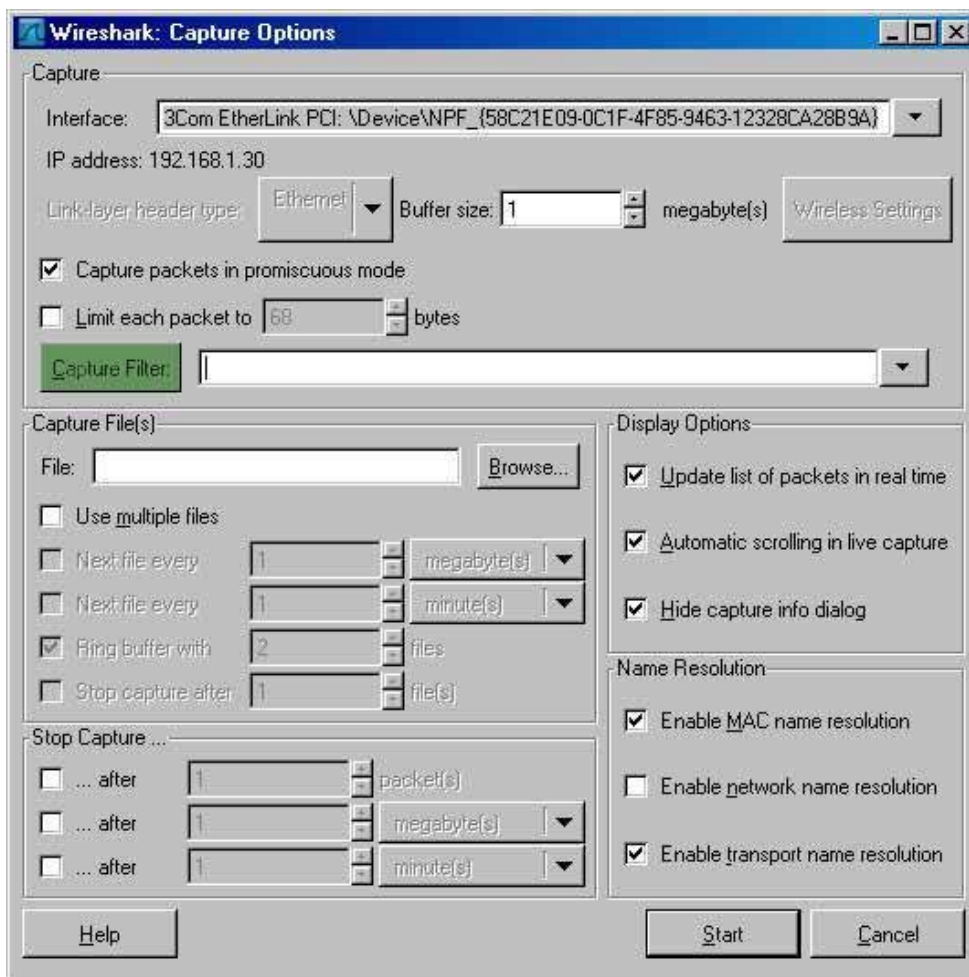
Los filtros de visualización (Display Filter) establecen un criterio de filtro sobre las paquetes capturados y que estamos visualizando en la pantalla principal de Wireshark. Estos filtros son más flexibles y potentes.

Filtros de Captura

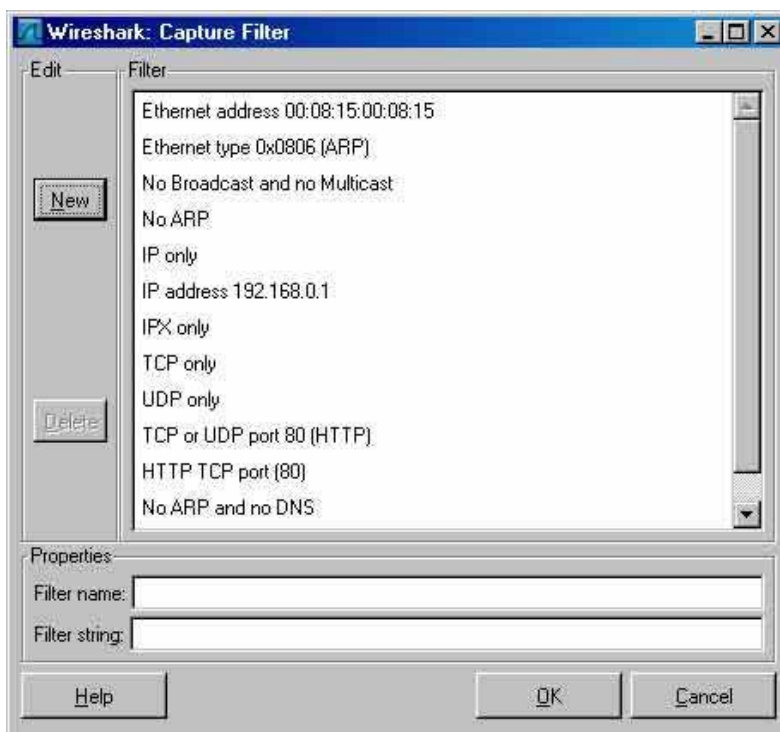
Estos filtros están basados en las librerías pcap. Los filtros son los mismos que podemos aplicar para Windump / TCPDump.

Si no establecemos ninguno, Wireshark capturaré todo el tráfico y lo presentará en la pantalla principal. Aún así podremos establecer filtros de visualización (display filter) para que nos muestre solo el tráfico deseado.

Se aplican en Capture > Options:



En el campo Capture Filter introducimos el filtro o pulsamos el botón Capture Filter para filtros predefinidos:



Combinación de Filtros.

Podemos combinar las primitivas de los filtros de la siguiente forma:

- ☐ Negación: ! ó not
- ☐ Unión o Concatenación: && ó and
- ☐ Alternancia: || ó or

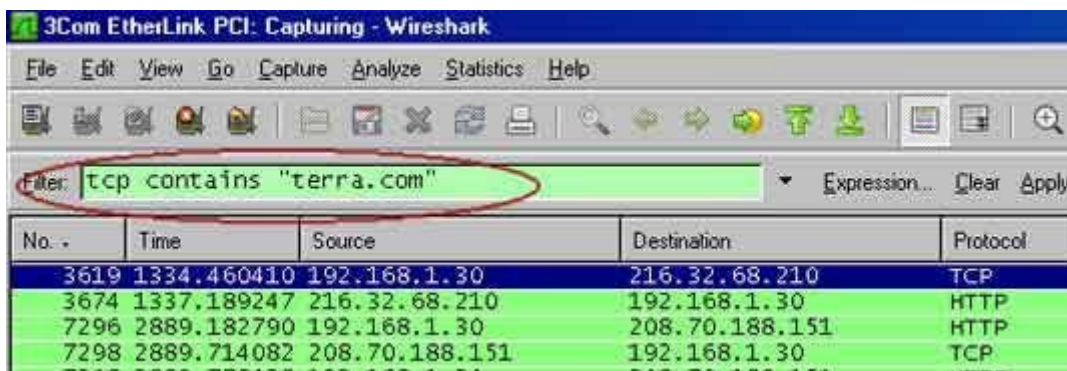
Vamos ahora a los filtros:

Filtros basados en hosts	
Ejemplo	Significado
host 192.168.1.20	Paquetes con origen o destino 192.168.1.20
src host 192.168.1.1	Paquetes con origen 192.168.1.1
dst host 192.168.1.1	Paquetes con destino 192.168.1.1
Filtros basados en puertos	
Ejemplo	Significado
port 21	Paquetes con puerto origen o destino 21
src port 21	Paquetes con puerto origen 21
not port 21 and not port 80	Paquetes excepto origen y destino puertos 21 y 80
portrange 1-1024	Paquetes con puerto origen y destino en un rango de puertos 1 a 1024
dst portrange 1-1024	Captura todos los paquetes con puerto destino en un rango de puertos 1 a 1024
Filtros protocolos Ethernet / IP	
Ejemplos	
ip	Captura todo el tráfico IP
tcp	Captura todos los segmentos TCP
arp	Captura todo el tráfico ARP
Filtros basados en red	
Ejemplos	
net 192.168.1.0	Tráfico con origen y destino subred 1.0
net 192.168.1.0/24	Tráfico para la subred 1.0 máscara 255.0
dst net 192.168.2.0	Tráfico con destino para la subred 2.0
net 192.168.2.0 and port 21	Tráfico origen y destino puerto 21 en subred 2.0
broadcast	Solo el tráfico broadcast
not broadcast and not multicast	Todo el tráfico excepto broadcast y el multicast

Filtros de Visualización (Display Filter)

Los filtros de visualización establecen un criterio de filtro sobre los paquetes que estamos visualizando en la pantalla principal de Wireshark. Al aplicar el filtro en la pantalla principal de Wireshark aparecerá solo el tráfico filtrado a través del filtro de visualización.

Comparación		
Forma 1	Forma 2	Significado
==	eq	Igual a
!=	neq	Distinto a
>	gt	Mayor a
<	lt	Menor a
>=	ge	Mayor o igual a
<=	le	Menor o igual a
Combinación		
!	not	Negación (no)
&&	and	Unión (y)
	or	Alternancia (uno u otro)
Otros operadores		
contains	Buscar una determinada cadena de caracteres	



Ejemplos	
Sintaxis	Significado
ip.addr == 192.168.1.40	Tráfico por host 192.168.1.40
ip.addr != 192.168.1.25	Todo excepto host 192.168.1.25
ip.dst == 192.168.1.30	Host destino 192.168.1.30

ip.src == 192.168.1.30	Host origen 192.168.1.30
ip	Todo el tráfico IP
tcp.port == 143	Origen o destino puerto TCP/143
ip.addr == 8.8.8.8 and tcp.port == 80	Origen y destino puerto TCP/80 y host 8.8.8.8
http contains "www.terra.com"	Paquetes HTTP que contienen www.terra.com
frame contains "@miempresa.es"	Correos con origen o destino al dominio miempresa.es.
icmp[0:1] == 08	Tráfico icmp de tipo echo request
ip.ttl == 1	Paquetes IP cuyo campo TTL sea igual a 1
tcp.window_size != 0	Paquetes cuyos campo Tamaño de Ventana del segmento TCP sea distinto de 0
ip.tos == x	Paquetes IP cuyo campo TOS sea igual a x
ip.flags.df == x	Paquetes IP cuyo campo DF sea igual a x
udp.port == 53	Visualiza todo el tráfico UDP puerto 53
tcp contains "terra.com"	Segmentos TCP conteniendo la cadena terra.com

Autor: Fabian Martinez Portantier

Fuentes:

- ❑ <http://incibe.es>
- ❑ <http://seguridadyredes.wordpress.com>
- ❑ <http://nmap.org>