wirshark, filtros

Filtros en Wireshark

Filtros de Captura

Los filtros de captura se aplican a la hora de capturar el tráfico de red . Es decir vamos a filtrar todo el tráfico para quedarnos con la parte del tráfico de red que permitamos en el filtro. Los filtros se escriben usando una sintaxis llamada BPF (Berkeley Packet Filter) y tienen dos partes **calificador**+**primitiva**

Existen 3 tipos de calificadores:

- Type: host,net,port,portrange
- Dir: src,dst,src or dst,src and dst,ra,ta,addr1,addr2,addr3,addr4
- Proto:ether,fddi,tr,wlan,ip,ip6,arp,rarp,decnet,tcp,udp

Ejemplos:

- Capturar el tráficos desde/hacia una ip → **host 192.168.0.1**
- Capturar todo excepto el que va desde/hacia una ip → **not host 192.168.0.1**
- Capturar el trafico hacia una ip →**dst 192.168.0.1**
- Capturar el tráficos hacia una red → **net 192.168.0.0/24**
- Capturar el trafico TCP/UDP del puerto 53→ **port 53**
- Capturar el tráfico de destino hacia el puerto 80 → **tcp dst port 80**

Tambien podemos usar los operadores lógicos AND y OR para afinar más nuestro filtros Por ejemplo → **host 192.168.0.1 and port 80**

Filtros de Visualización

Los filtros de visualización se aplican sobre el tráfico capturado para visualizar los paquetes de interés dentro de una captura.

Sintáxis

calificador + operador + primitiva

El calificador puede tratarse de un protocolo, campo de la cabecera de un protocolo, o simplemente una característica de un protocolo.

Para visualizar los paquetes del un protocolo bastaría con poner en la línea de filtro el protocolo, así por ejemplo podrámos poner; arp, ip, tcp, udp, http, dns etc para visualizar los paquetes de ese protocolo.

Por ejemplo para visualizar los paquetes de una ip determinada → ip == 192.168.0.1

Los filtros de visualización utilizan una sintáxis diferente a la de los filtros de captura (BPF)

Es posible unir 2 o más filtros de visualización a través de concatenadores lógicos.

- &&: implica que ambos filtros deben cumplirse. Es como un operador lógico **AND**
- | |: implica que es suficiente con que uno de los filtros se cumpla. Es como el operador lógico **OR**

Ejemplos:

```
ip.src == 192.168.0.1 && tcp.port == 80
tcp.port == 80 || tcp.port == 443
```