过滤器的区别

- 捕捉过滤器（CaptureFilters）：用于决定将什么样的信息记录在捕捉结果中。需要在开始捕捉前设置。
- 显示过滤器（DisplayFilters）：在捕捉结果中进行详细查找。他们可以在得到捕捉结果后随意修改。

两种过滤器的不同点：

- 捕捉过滤器是数据经过的第一层过滤器，它用于控制捕捉数据的数量，以避免产生过大的日志文件。
- 显示过滤器是一种更为强大（复杂）的过滤器。它允许您在日志文件中迅速准确地找到所需要的记录。

两种过滤器使用的语法是完全不同的。

捕捉过滤器

语法： Protocol Direction Host(s) Value Logical Operations Other expression
例子： tcp dst 10.1.1.1 80 and tcp dst 10.2.2.2 3128

Protocol（协议）:
可能的值: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp.
如果没有特别指明是什么协议，则默认使用所有支持的协议。

Direction（方向）:
可能的值: src, dst, src and dst, src or dst
如果没有特别指明来源或目的地，则默认使用 “src or dst” 作为关键字。
例如，”host 10.2.2.2″与”src or dst host 10.2.2.2″是一样的。

Host(s):
可能的值： net, port, host, portrange.
如果没有指定此值，则默认使用”host”关键字。
例如，”src 10.1.1.1″与”src host 10.1.1.1″相同。

Logical Operations（逻辑运算）:
可能的值：not, and, or.
否(“not”)具有最高的优先级。或(“or”)和与(“and”)具有相同的优先级，运算时从左至右进行。
例如，
“not tcp port 3128 and tcp port 23″与”(not tcp port 3128) and tcp port 23″相同。
“not tcp port 3128 and tcp port 23″与”not (tcp port 3128 and tcp port 23)”不同。

例子：

tcp dst port 3128
显示目的 TCP 端口为 3128 的封包。

ip src host 10.1.1.1
显示来源 IP 地址为 10.1.1.1 的封包。

host 10.1.2.3
显示目的或来源 IP 地址为 10.1.2.3 的封包。

src portrange 2000-2500
显示来源为 UDP 或 TCP，并且端口号在 2000 至 2500 范围内的封包。

not imcp

显示除了 icmp 以外的所有封包。（icmp 通常被 ping 工具使用）

src host 10.7.2.12 and not dst net 10.200.0.0/16
显示来源 IP 地址为 10.7.2.12，但目的地不是 10.200.0.0/16 的封包。

(src host 10.4.1.12 or src net 10.6.0.0/16) and tcp dst portrange 200-10000 and dst net 10.0.0.0/8
显示来源 IP 为 10.4.1.12 或者来源网络为 10.6.0.0/16，目的地 TCP 端口号在 200 至 10000 之间，并且目的位于网络 10.0.0.0/8 内的所有封包。

src net 192.168.0.0/24
src net 192.168.0.0 mask 255.255.255.0
显示来源 IP 地址为 10.1.1.1 的封包。

注意事项：

当使用关键字作为值时，需使用反斜杠“\”。
“ether proto \ip” (与关键字”ip”相同).
这样写将会以 IP 协议作为目标。

“ip proto \icmp” (与关键字”icmp”相同).
这样写将会以 ping 工具常用的 icmp 作为目标。

可以在”ip”或”ether”后面使用”multicast”及”broadcast”关键字。
当您想排除广播请求时，”no broadcast”就会非常有用。

Protocol（协议）:

您可以使用大量位于 OSI 模型第 2 至 7 层的协议。点击”Expression…”按钮后，您可以看到它们。
比如：IP，TCP，DNS，SSH

String1, String2 (可选项):

协议的子类。
点击相关父类旁的”+”号，然后选择其子类。

Comparison operators （比较运算符）:

可以使用 6 种比较运算符：

英文写法： C 语言写法： 含义：
eq == 等于
ne != 不等于
gt > 大于

lt < 小于
ge >= 大于等于
le <= 小于等于

Logical expressions（逻辑运算符）:

英文写法： C 语言写法： 含义：
and && 逻辑与
or || 逻辑或
xor ^^ 逻辑异或
not ! 逻辑非

显示过滤器

语法： Protocol . String 1 . String 2 Comparison operator Value Logical Operations Other expression
例子： ftp passive ip == 10.2.3.4 xor icmp.type

例子：

snmp || dns || icmp 显示 SNMP 或 DNS 或 ICMP 封包。
ip.addr == 10.1.1.1
显示来源或目的 IP 地址为 10.1.1.1 的封包。

ip.src != 10.1.2.3 or ip.dst != 10.4.5.6
显示来源不为 10.1.2.3 或者目的不为 10.4.5.6 的封包。
换句话说，显示的封包将会为：
来源 IP：除了 10.1.2.3 以外任意；目的 IP：任意
以及
来源 IP：任意；目的 IP：除了 10.4.5.6 以外任意

ip.src != 10.1.2.3 and ip.dst != 10.4.5.6
显示来源不为 10.1.2.3 并且目的 IP 不为 10.4.5.6 的封包。
换句话说，显示的封包将会为：
来源 IP：除了 10.1.2.3 以外任意；同时须满足，目的 IP：除了 10.4.5.6 以外任意

tcp.port == 25 显示来源或目的 TCP 端口号为 25 的封包。
tcp.dstport == 25 显示目的 TCP 端口号为 25 的封包。
tcp.flags 显示包含 TCP 标志的封包。
tcp.flags.syn == 0×02 显示包含 TCP SYN 标志的封包。