4.5.1 捕获过滤器

捕获过滤器用于进行数据包捕获的实际场合，使用它的一个主要原因就是性能。如果你并不需要分析某个类型的流量，则可以简单地使用捕获过滤器过滤掉它，从而节省那些会被用来捕获这些数据包的处理器资源。

当处理大量数据的时候，创建自定义的捕获过滤器是相当好用的。它可以让你专注于那些与你手头事情有关的数据包，从而加速分析过程。

举一个简单的例子，你在一台有多种角色的服务器上捕获流量时很可能会用到捕获过滤器，假设你正在解决一个运行于 262 端口网络服务的问题，如果你正在分析的那台服务器在许多端口运行了各种不同的网络服务，那么找到并分析只运行于 262 端口的流量本身可能就具有一定的工作量。你可以通过本章前面讨论过的 Capture Options 对话框到达目的，步骤如下所示。

（1） 选择 Capture -> Options 按钮打开捕获接口对话框。

（2） 选择你想进行数据包捕获的设备，然后在最右列选中捕获过滤器。

（3） 你可以单击该列以输入一个表达式来应用捕获过滤器。我们希望过滤器只显示出 262 端口的出站和入站流量，所以如图 4-14 所示，输入 port 262（我们将在下一区段中仔细地讨论关于过滤表达式的问题）。如果你输入的表达式合法，那么方格里的字颜色应该会变绿；否则会变红色。

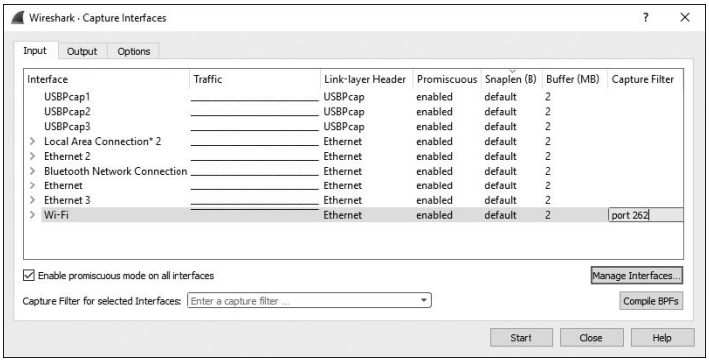

图 4-14　在 Capture Options 对话框中创建一个捕获过滤器

（4） 当你设定好过滤器之后，单击 Start 开始捕获。

当收集到足够多的样本之后，你应该只看见了端口 262 的流量，这样就能更有效率地分析这些数据了。

1．捕获过滤器的 BPF 语法

捕获过滤器应用于 libpcap/WinPcap，并使用 Berkeley Packet Filter（BPF）语法。这个语法被广泛用于多种数据包嗅探软件，主要因为大部分数据包嗅探软件都依赖于使用 BPF 的 libpcap/WinPcap 库。掌握 BPF 语法对你在数据包级别更深入地探索网络来说，非常关键。

使用 BPF 语法创建的过滤器被称为 expression（表达式），并且每个表达式包含一个或多个 primitives（原语）。每个原语包含一个或多个 qualifiers（限定词），然后跟着一个 ID 名字或者数字（见表 4-2），如图 4-15 所示。

表 4-2　BPF 限定词

限定词	说明	例子
Type	指出名字或数字所代表的意义	host、net、port
Dir	指明传输方向是前往还是来自名字或数字	src、dst
Proto	限定所要匹配的协议	Ether、ip、tcp、udp、http、ftp

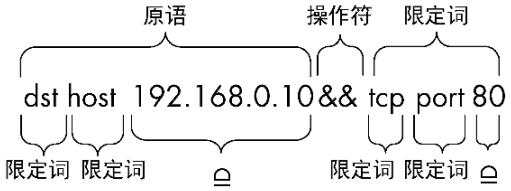

图 4-15　一个捕获过滤器样例

在给定表达式的组成部分中，一个 src 限定词和 192.168.0.10 组成了一个原语。这个原语本身就是表达式，可以用它只捕获那些源 IP 地址是 192.168.0.10 的流量。

你可以使用以下 3 种逻辑运算符，对原语进行组合，从而创建更高级的表达式：

- 连接运算符 与（&&）

- 选择运算符 或（||）
- 否定运算符 非（!）

举例来说，下面的这个表达式只对源地址是 192.168.0.10 和源端口或目标端口是 80 的流量进行捕获。

```
src host 192.168.0.10 && port 80
```

2．主机名和地址过滤器

你所创建的大多数过滤器都会关注于一个或一些特定的网络设备。根据这种情况，你可以根据设备的 MAC 地址、IPv4 地址、IPv6 地址或者 DNS 主机名配置过滤规则。

举例来说，假设你对一个正在和你网络中某个服务器进行交互的主机所产生的流量感兴趣，那么你可以在这台服务器上创建一个使用 host 限定词的过滤器，来捕获所有和那台主机 IPv4 地址相关的流量：

```
host 172.16.16.149
```

如果你在使用一个 IPv6 网络，则可能需要使用基于 IPv6 地址的 host 限定词，如下所示：

```
host 2001:db8:85a3:8a2e:370:7334
```

你同样可以使用基于一台设备的主机名 host 限定词，就像这样：

```
host testserver2
```

或者，如果你考虑到一台主机的 IP 地址可能会变化，则可以通过加入 ether 协议限定词，对它的 MAC 地址进行过滤：

```
ether host 00-1a-a0-52-e2-a0
```

传输方向限定词通常会和前面例子演示的那些过滤器一起使用，来捕获流向或者流出某台主机的流量。举例来说，如果想捕获来自某台主机的流量，则可以加入 src 限定词：

```
src host 172.16.16.149
```

如果只想捕获发往 172.16.16.149 服务器的流量，则可以使用 dst 限定词：

```
dst host 172.16.16.149
```

如果你在一个原语中没有指定一种类型限定符（host、net 或者 port），host 限定词将作为默认选择。所以上面的那个例子也可以写成没有类型限定符的样子：

```
dst 172.16.16.149
```

3．端口过滤器

不仅仅可以基于主机过滤，你还可以对基于每个数据包的端口进行过滤。端口过滤通常被用来过滤使用已知端口的服务和应用。举例来说，下面是一个只对 8080 端口进行流量捕获的简单过滤器的例子：

```
port 8080
```

如果想要捕获除 8080 端口外的所有流量，则如下所示：

```
!port 8080
```

端口过滤器可以和传输方向限定符一起使用。举例来说，如果希望只捕获访问标准 HTTP80 端口的 Web 服务器，则可以使用 dst 限定符：

```
dst port 80
```

4．协议过滤器

协议过滤器允许你基于特定协议进行数据包过滤。这通常被用于那些非应用层的不能简单地使用特定端口进行定义的协议。所以如果你只想看一看 ICMP 流量，则可以使用下面这个过滤器：

```
icmp
```

如果你想看除了 IPv6 之外的所有流量，则下面这个方式能够满足要求。

```
!ip6
```

5．协议域过滤器

BPF 语法提供给我们的一项强大的功能，就是可以通过检查协议头中的每一个字节来创建基于那些数据的特殊过滤器。在本节中我们将要讨论的这些高级的过滤器，通过它们你可以匹配一个数据包中从某一个特定位置开始的一定数量的字节。

举例来说，假设我们想要基于 ICMP 过滤器的类型域进行过滤，而类型域位于数据包的最开头也就是偏移量为 0 的位置，那么我们可以通过在协议限定符后输入由方括号括起的字节偏移量，在这个例子中就是 icmp[0]，来指定我们想在一个数据包内进行检查的位置。这样将返回一个 1 字节的整型

值用于比较。如果只想要得到代表目标不可达（类型 3）信息的 ICMP 数据包，则我们可在过滤器表达式中令其等于 3，如下所示：

```
icmp[0]==3
```

如果只想要检查代表 echo 请求（类型 8）或 echo 回复（类型 0）的 ICMP 数据包，则可以使用带有 OR 运算符的两个原语：

```
icmp[0]==8||icmp[0]==0
```

虽然这些过滤器很好用，但是它们只能基于数据包头部的 1 个字节进行过滤。当然，你也可以在方括号中偏移值的后面以冒号分隔加上一个字节长度，来指定你希望返回给过滤器表达式的数据长度。

举例来说，假设我们要创建一个过滤器，该过滤器捕获所有 ICMP 目标——不可访问、主机不可达的数据包（类型 3，代码 1）。这些是 1 字节的字段，它们与偏移量为 0 的数据包头部相邻。为此，我们创建了一个过滤器，它检查从数据包头部的偏移量 0 处开始的 2 字节数据，并与十六进制值 0301（类型 3，代码 1）进行比较，如下所示：

```
icmp[0:2]==0x0301
```

一个常用的场景就是只捕获带有 RST 标志的 TCP 数据包。我们将在第 8 章深入讲述 TCP 的相关内容，而现在你只需要知道 TCP 数据包的标志位在偏移为 13 字节的地方。有趣的是，虽然整个标志位加在一起是 1 字节，但是这个字节中每一比特位都是一个标志。在一个 TCP 数据包中，多个标志可以被同时设置，因此多个值可能都代表 RST 位被设置，我们不能只通过一个 tcp[13]的值来进行有效过滤。我们必须通过在当前的原语中加入一个单一的&符号，来指定我们希望在这个字节中检查的比特位置。在这个字节中 RST 标志所在的比特位代表数字 4，也就是说这个比特位被设置成 4，就代表这个标志被设置了。过滤器看上去是这个样子的：

```
tcp[13]&4==4
```

如果希望看到所有被设置了 PSH 标志（比特位代表数字 8）的数据包，那么我们的过滤器应该会将其相应位置替换成这样：

```
tcp[13]&8==8
```

6．捕获过滤器表达式样例

你将会发现分析的成败很多时候取决于你能否编写出恰当的过滤器。表 4-3 给出了一些我经常使用的捕获过滤器。

表 4-3　常用捕获过滤器

过滤器	说明
tcp[13]&32==32	设置了 URG 位的 TCP 数据包
tcp[13]&16==16	设置了 ACK 位的 TCP 数据包
tcp[13]&8==8	设置了 PSH 位的 TCP 数据包
tcp[13]&4==4	设置了 RST 位的 TCP 数据包
tcp[13]&2==2	设置了 SYN 位的 TCP 数据包
tcp[13]&1==1	设置了 FIN 位的 TCP 数据包
tcp[13]==18	TCP SYN-ACK 数据包
ether host 00:00:00:00:00:00（替换为你的 MAC）	流入或流出你 MAC 地址的流量
!ether host 00:00:00:00:00:00（替换为你的 MAC）	不流入或流出你 MAC 地址的流量
broadcast	仅广播流量
icmp	ICMP 流量
icmp[0:2]	ICMP 目标不可达、主机不可达
ip	仅 IPv4 流量

过滤器	说明
ip6	仅 IPv6 流量
udp	仅 UDP 流量