BPF语法介绍(Berkeley Packet Filter)

此文是伯克利包过滤的语法，该语法适用于wireshark, tcpdump等使用libpcap截获数据包的应用程序。想知道libpcap如何设置该语法，可以参考我之前写的文章<<libpcap编程简单介绍>>。

一、BPF的作用是什么？

使用BPF过滤规则，你可以确定该获取和检查哪些流量，忽略哪些流量。BPF让你能够通过比较第2、3、4层协议中各个数据字段值的方法对流量进行过滤。BPF中内置了一些“基元”来指代一些常用的协议字段。可以用“host”、”port"之类的基元写出非常简洁的BPF过滤规则，也可以检测位于指定偏移量上的字段（甚至可以是一个位）的值。BPF过滤器也可以由详尽的条件链和嵌套的逻辑“与”、“或”操作组成。

二、BPF根据过滤表达式过滤数据包

BPF过滤表达式由BPF基元组成

BPF基元可以理解为由一个或者多个限定符组成的一个id、名称或者序号。可以用来指定host, port, 协议、协议元素或者其他抓包规则。

限定符有：

type限定符：规定了id名或id序号指的是哪种类型的数据，可能的type有host、net、port和portrange

dir限定符：规定了流量是从id流进还是流出的（或两种兼有）。可能的dir有src、dst。

Proto限定符：规定了所匹配的协议。可能的proto有：ether、fddi、tr、wlan、ip、ip6、arp、rarp、decnet、tcp和udp

最常用的BPF基元要数“host id”，它是用来过滤与某台主机相关的流量的，其中id一栏中应该填上一个地址或主机名。输入“tcp and host 10.10.10.10”这样的过滤规则，将值获取流入/流出得做10.10.10.10的TCP流量，其他的所有帧都会被过滤掉。

举例说明：假设我们需要过滤主机10.0.10.240上端口1024, 1025上的tcp协议数据，那么过滤语法就如下：

"host 10.0.10.240 and ( port 1024 or port 1025 ) and tcp"

如使用tcpdump，则可以输入命令行如下：

sudo tcpdump -i eth0 "host 10.0.10.240 and ( port 1024 or port 1025 ) and tcp"

目前许可的基元有：

* dst host id , src host id, host id
* ether dst id, ether src id, ether host id
* gateway id

• net id , dst net id , src net id

* port id , dst port id , src port id
* dst portrange port1-port2, src portrange port1-port2, portrange port1-port2

• ip proto id , ether proto id

• tcp, udp, icmp, arp

• vlan id

三、BPF也可以根据字节的值过滤数据包

BPF语言也可以用来检查帧内任意一个单字节字段的值是不是规定值。下面是一些例子：

ip[8]<64

这个过滤规则规定要抓取的是：所有自ip头偏移1个字节的那个单字节字段的值小于64的IP包。被检查的这个字段表示的是“包的存活时间”或称“TTL”。

也可以检查连续几个字节的值:

tcp[0:2] = 31337

这个语句检查了一个多字节字段，它检查的是：TCP头部偏移0个字节起的一个多字节字段（2个字节,右侧2表示字节数），该字段表示的是TCP源端口。所以这个表达式就等价于BPF基元“src port 31337”

根据位（bit）的值过滤数据包

BPF语言还提供了一种方法让我们能检查更小的字段或精度更高的偏移量。具体做法是：我们先引用相关的字节，或多个字节，然后再用“位掩码”逐位地把我们需要检查的位分离出来。

假设要过滤所有IP头部中可选字段被启用的包（就是IP头的长度大于20个字节的包）。IP头部的低半个字节表示的是IP头的长度，以“word”(字)为单位(一个word等于4个字节)。我们只需要找出这半个字节的值大于5（5word\*4个字节/word=20个字节）的包就等于找出所有IP头部大于20个字节的包了。具体做法是用位掩码“00001111”（或者0x0F）创建一个BPF过滤规则，通过逻辑“与”运算提取目标值。

ip[0] & 0x0F > 5

四、BPF过滤表达式可以是混合表达式

比如过滤tcp端口为80、8080或者8088，且tcp包含有数据的数据包

tcp && ( port 80 || port 8088 || port 8080 ) and (((ip[2:2] - ((ip[0]&0xf)<<2)) - ((tcp[12]&0xf0)>>2)) != 0)

附:ip协议与tcp协议的协议图



