WireShark过滤语法

1.过 滤IP，如来源IP或者目标IP等于某个IP  
例子:  
ip.src eq 192.168.1.107 or ip.dst eq 192.168.1.107  
或者  
ip.addr eq 192.168.1.107 // 都能显示来源IP和目标IP

2.过滤端 口  
例子:  
tcp.port eq 80 // 不管端口是来源的还是目标的都显示  
tcp.port == 80  
tcp.port eq 2722  
tcp.port eq 80 or udp.port eq 80  
tcp.dstport == 80 // 只显tcp协议的目标端口80  
tcp.srcport == 80 // 只显tcp协议的来源端口80  
udp.port eq 15000  
过滤端口范围  
tcp.port >= 1 and tcp.port <= 80  
  
3.过 滤协议  
例子:  
tcp  
udp  
arp  
icmp  
http  
smtp  
ftp  
dns  
msnms  
ip  
ssl  
oicq  
bootp  
等 等  
排除arp包，如!arp  或者  not arp

4.过 滤MAC  
太以网头过滤  
eth.dst == A0:00:00:04:C5:84 //过滤目 标mac  
eth.src eq A0:00:00:04:C5:84 //过 滤来源mac  
eth.dst==A0:00:00:04:C5:84  
eth.dst==A0-00-00-04-C5-84  
eth.addr eq A0:00:00:04:C5:84 //过滤来 源MAC和目标MAC都等于A0:00:00:04:C5:84的  
less than 小于 < lt  
小于等于 le  
等 于 eq  
大于 gt  
大于等于 ge  
不等 ne

5.包长度过 滤  
例子:  
udp.length == 26 这个长度是指udp本身固定长度8加上udp下面那块数据包之和  
tcp.len >= 7  指的是ip数据包(tcp下面那块数据),不包括tcp本身  
ip.len == 94 除了以太网头固定长度14,其它都算是ip.len,即从ip本身到最后  
frame.len == 119 整个数据包长度,从eth开始到最后  
eth ---> ip or arp ---> tcp or udp ---> da

ta  
  
6.http 模式过滤  
  
例子:  
http.request.method == "GET"  
http.request.method == "POST"  
http.request.uri == "/img/logo-edu.gif"  
http contains "GET"  
http contains "HTTP/1."  
  
// GET包  
http.request.method == "GET" && http contains "Host: "http.request.method == "GET" && http contains "User-Agent: "  
// POST包  
http.request.method == "POST" && http contains "Host: "  
http.request.method == "POST" && http contains "User-Agent: "  
// 响应包  
http contains "HTTP/1.1 200 OK" && http contains "Content-Type: "  
http contains "HTTP/1.0 200 OK" && http contains "Content-Type: "  
一 定包含如下  
Content-Type:  
  
  
7.TCP参数过 滤  
tcp.flags 显示包含TCP标志的封包。  
tcp.flags.syn == 0x02    显示包含TCP SYN标志的封包。  
tcp.window\_size == 0 && tcp.flags.reset != 1  
  
8.过滤内容  
tcp[20] 表示从20开始，取1个字符  
tcp[20:]表示从20开始，取1个字符以上  
tcp[20:8]表示从20开始，取8个字符  
tcp[offset,n]  
udp[8:3]==81:60:03 // 偏移8个bytes,再取3个数，是否与==后面的数据相等？  
udp[8:1]==32  如果我猜的没有错的话，应该是udp[offset:截取个数]=nValue  
eth.addr[0:3]==00:06:5B  
例 子:  
判断upd下面那块数据包前三个是否等于0x20 0x21 0x22  
我们都知道udp固定长度为8  
udp[8:3]==20:21:22  
判 断tcp那块数据包前三个是否等于0x20 0x21 0x22  
tcp一般情况下，长度为20,但也有不是20的时候  
tcp[8:3]==20:21:22  
如 果想得到最准确的，应该先知道tcp长度  
matches(匹配)和contains(包含某字符串)语法  
ip.src==192.168.1.107 and udp[8:5] matches "\\x02\\x12\\x21\\x00\\x22"  
ip.src==192.168.1.107 and udp contains 02:12:21:00:22  
ip.src==192.168.1.107 and tcp contains "GET"  
udp contains 7c:7c:7d:7d 匹配payload中含有0x7c7c7d7d的UDP数据包，不一定是从第一字节匹配。  
例子:  
得到本地qq登陆数据包(判断条 件是第一个包==0x02,第四和第五个包等于0x00x22,最后一个包等于0x03)  
0x02 xx xx 0x00 0x22 ... 0x03  
正确  
oicq and udp[8:] matches "^\\x02[\\x00-\\xff][\\x00-\\xff]\\x00\\x22[\\x00-\\xff]+\\x03$"  
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$" // 登陆包  
oicq and (udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x03$" or tcp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x03$")  
oicq and (udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$" or tcp[20:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$")  
  
不 单单是00:22才有QQ号码,其它的包也有,要满足下面条件(tcp也有，但没有做):  
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]+\\x03$" and !(udp[11:2]==00:00) and !(udp[11:2]==00:80)  
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]+\\x03$" and !(udp[11:2]==00:00) and !(udp[15:4]==00:00:00:00)  
说明:  
udp[15:4]==00:00:00:00 表示QQ号码为空  
udp[11:2]==00:00 表示命令编号为00:00  
udp[11:2]==00:80 表示命令编号为00:80  
当命令编号为00:80时，QQ号码为 00:00:00:00  
  
得到msn登陆成功账号(判断条件是"USR 7 OK ",即前三个等于USR，再通过两个0x20，就到OK,OK后面是一个字符0x20,后面就是mail了)  
USR xx OK mail@hotmail.com  
正确  
msnms and tcp and ip.addr==192.168.1.107 and tcp[20:] matches "^USR\\x20[\\x30-\\x39]+\\x20OK\\x20[\\x00-\\xff]+"  
  
9.dns 模式过滤  
  
  
10.DHCP  
以 寻找伪造DHCP服务器为例，介绍Wireshark的用法。在显 示过滤器中加入过 滤规则，  
显示所有非来自DHCP服务器并且bootp.type==0x02（Offer/Ack）的信息：  
bootp.type==0x02 and not ip.src==192.168.1.1  
  
11.msn  
msnms && tcp[23:1] == 20 // 第四个是0x20的msn数据包  
msnms && tcp[20:1] >= 41 && tcp[20:1] <= 5A && tcp[21:1] >= 41 && tcp[21:1] <= 5A && tcp[22:1] >= 41 && tcp[22:1] <= 5A  
msnms && tcp[20:3]=="USR" // 找到命令编码是USR的数据包  
msnms && tcp[20:3]=="MSG" // 找到命令编码是MSG的数据包  
tcp.port == 1863 || tcp.port == 80  
  
如何判断数据包是含 有命令编码的MSN数据包?  
1)端口为1863或者80,如:tcp.port == 1863 || tcp.port == 80  
2) 数据这段前三个是大写字母,如:  
tcp[20:1] >= 41 && tcp[20:1] <= 5A && tcp[21:1] >= 41 && tcp[21:1] <= 5A && tcp[22:1] >= 41 && tcp[22:1] <= 5A  
3)第四个为0x20,如:tcp[23:1] == 20  
4)msn是属于TCP协议的,如 tcp  
  
MSN Messenger 协议分析  
http://blog.csdn.net/Hopping/archive/2008/11/13/3292257.aspx  
  
MSN 协议分析  
http://blog.csdn.net/lzyzuixin/archive/2009/03/13/3986597.aspx  
  
更 详细的说明  
<<wireshark过 滤表达式实例介绍>>  
http://www.csna.cn/viewthread.php?tid=14614