willingtolove

一念成了佛，一念成了魔~

昵称： willingtolove
园龄： 7年9个月
粉丝： 78
关注： 32
+加关注

Wireshark过滤器写法总结

目录

#Wireshark提供了两种过滤器：

回到顶部

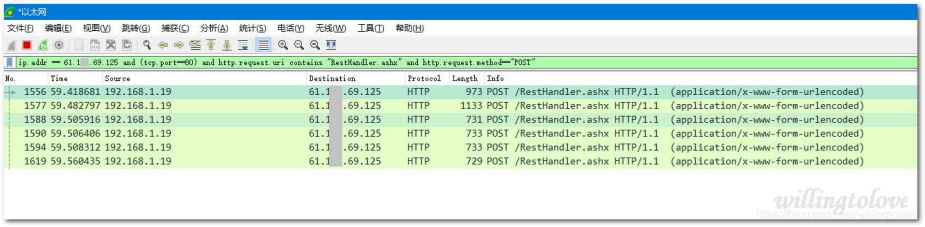
1、捕获过滤器

捕获过滤器：在抓包之前就设定好过滤条件，然后只抓取符合条件的数据包。

回到顶部

2、显示过滤器

显示过滤器：在已捕获的数据包集合中设置过滤条件，隐藏不想显示的数据包，只显示符合条件的数据包。



注意：这两种过滤器所使用的语法是完全不同的，想想也知道，捕捉网卡数据的其实并不是Wireshark,而是WinPcap,当然要按WinPcap的规则来，显示过滤器就是Wireshark对已捕捉的数据进行筛选。

使用捕获过滤器的主要原因就是性能。如果你知道并不需要分析某个类型的流量，那么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，使用捕获过滤器是相当好用的。

2020年9月(1)
2020年7月(5)
2020年6月(5)
2020年5月(2)
2020年3月(2)
2020年2月(4)
2020年1月(6)
2019年12月(5)
2019年11月(10)
2019年10月(1)
2019年9月(1)
2019年8月(5)
2019年7月(11)
2019年6月(8)
更多

阅读排行榜
1. 【JS】JS数组添加元素的三种方法(271494)
2. 【SQL】sql update 多表关联更新方法总结(162054)
3. js中Date与timestamp（时间戳）的相互转换(128901)
4. 【.net】未在本地计算机上注册“microsoft.ACE.oledb.12.0”提供程序解决办法(124647)
5. 【C#】C#获取文件夹下的所有文件(120320)

评论排行榜
1. 【C#】判断字符串中是否包含指定字符串,contains与indexof方法效率问题(18)
2. .Net防sql注入的方法总结(6)
3. C#实现在foreach遍历中删除集合中的元素(方法总结)(5)
4. 【SQL】sql update 多表关联更新方法总结(5)
5. 【.net】未在本地计算机上注册“microsoft.ACE.oledb.12.0”提供程序解决办法(5)

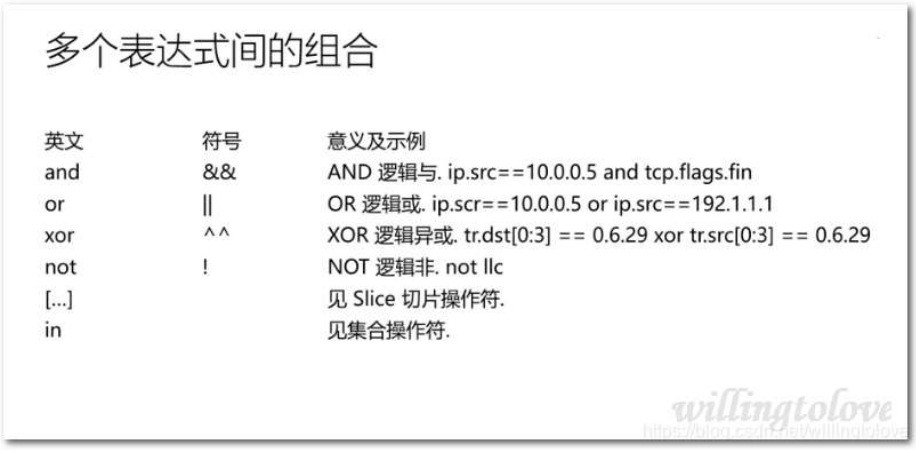
Wireshark拦截通过网卡访问的所有数据，前提是没有设置任何代理。
Wireshark不能拦截本地回环访问的请求，即127.0.0.1或者localhost。

#过滤器具体写法

回到顶部

#显示过滤器写法

1、过滤值比较符号及表达式之间的组合



2、针对ip的过滤

- 对源地址进行过滤

```
ip.src == 192.168.0.1
```

- 对目的地址进行过滤

```
ip.dst == 192.168.0.1
```

- 对源地址或者目的地址进行过滤

```
ip.addr == 192.168.0.1
```

- 如果想排除以上的数据包，只需要将其用括号囊括，然后使用 "!" 即可

```
!(ip.addr == 192.168.0.1)
```

3、针对协议的过滤

- 获某种协议的数据包，表达式很简单仅仅需要把协议的名字输入即可

5 0

```
http
```

注意：是否区分大小写？答：区分， 只能为小写

- 捕获多种协议的数据包

```
http or telnet
```

- 排除某种协议的数据包

```
not arp   或者   !tcp
```

4、针对端口的过滤（视传输协议而定）

- 捕获某一端口的数据包（以tcp协议为例）

```
tcp.port == 80
```

- 捕获多端口的数据包，可以使用and来连接，下面是捕获高于某端口的表达式（以udp协议为例）

```
udp.port >= 2048
```

5、针对长度和内容的过滤

- 针对长度的过虑（这里的长度指定的是数据段的长度）

```
udp.length < 20
http.content_length <=30
```

- 针对uri 内容的过滤

```
http.request.uri matches "user" （请求的uri中包含“user”关键字的）
```

注意： matches 后的关键字是 不区分大小写 的！

```
http.request.uri contains "User" （请求的uri中包含“user”关键字的）
```

注意： contains 后的关键字是 区分大小写 的！

5、针对http请求的一些过滤实例。

- 过滤出请求地址中包含“user”的请求，不包括域名；

```
http.request.uri contains "User"
```

- 精确过滤域名

```
http.host==baidu.com
```

- 模糊过滤域名

```
http.host contains "baidu"
```

- 过滤请求的content_type类型

```
http.content_type =="text/html"
```

- 过滤http请求方法

```
http.request.method=="POST"
```

- 过滤tcp端口

```
tcp.port==80
```

```
http && tcp.port==80 or tcp.port==5566
```

- 过滤http响应状态码

```
http.response.code==302
```

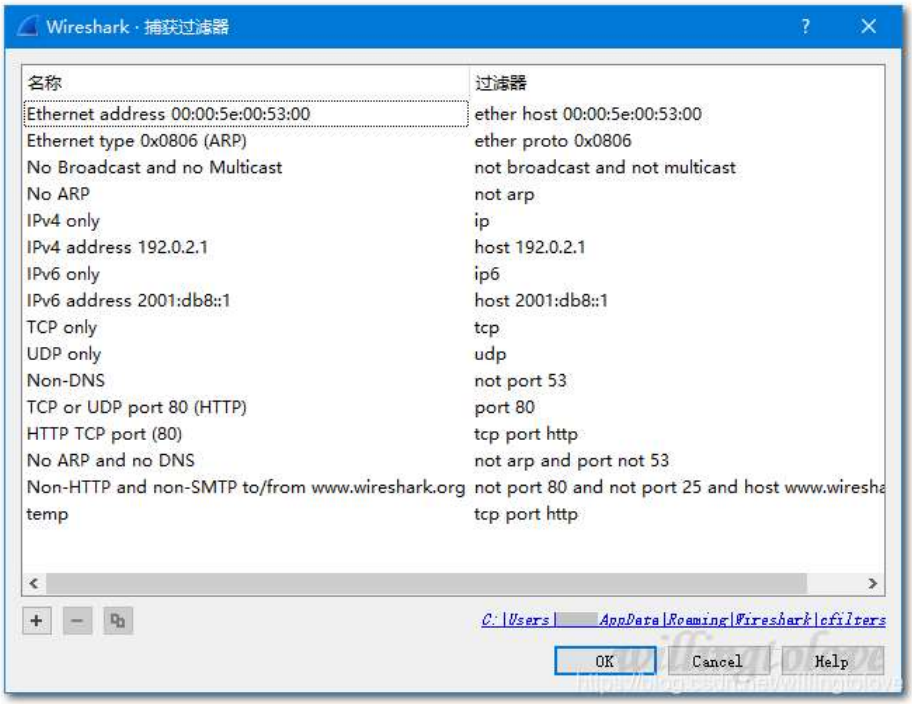
- 过滤含有指定cookie的http数据包

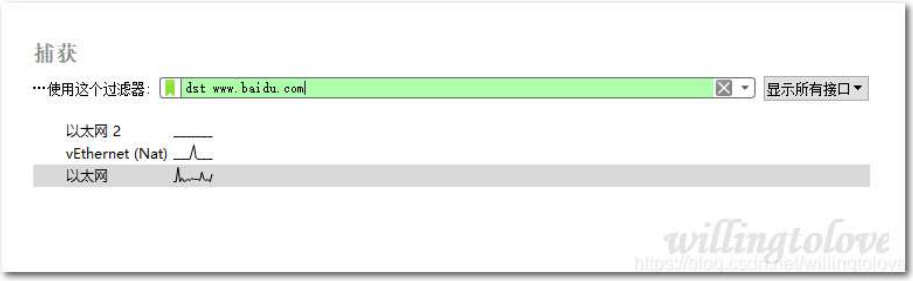
```
http.cookie contains "userid"
```

回到顶部

#捕捉过滤器写法

在wireshark的工具栏中点击 捕获 → 捕获过滤器 ，可以看到一些过滤器的写法，如下图：





1、比较符号

```
与: &&或者and
或: ||或者or
非: ！或者not
```

实例：

```
src or dst portrange 6000-8000 && tcp or ip6
```

2、常用表达式实例

- 源地址过滤

```
src www.baidu.com
```

- 目的地址过滤

```
dst www.baidu.com
```

- 目的地址端口过滤

```
dst post 80
```

- 协议过滤

```
udp
```


分类: 【抓包工具 相关】

« 上一篇： ASP.NET Core中添加MIME 类型
» 下一篇： .net intallutil.exe 安装服务报错：未能加载文件或程序集（异常来自 HRESULT:0x80131515）

posted @ 2020-03-18 19:13 willingtolove 阅读(13161) 评论(1) 编辑 收藏 举报