Wireshark 部分教程及作业

1.简介

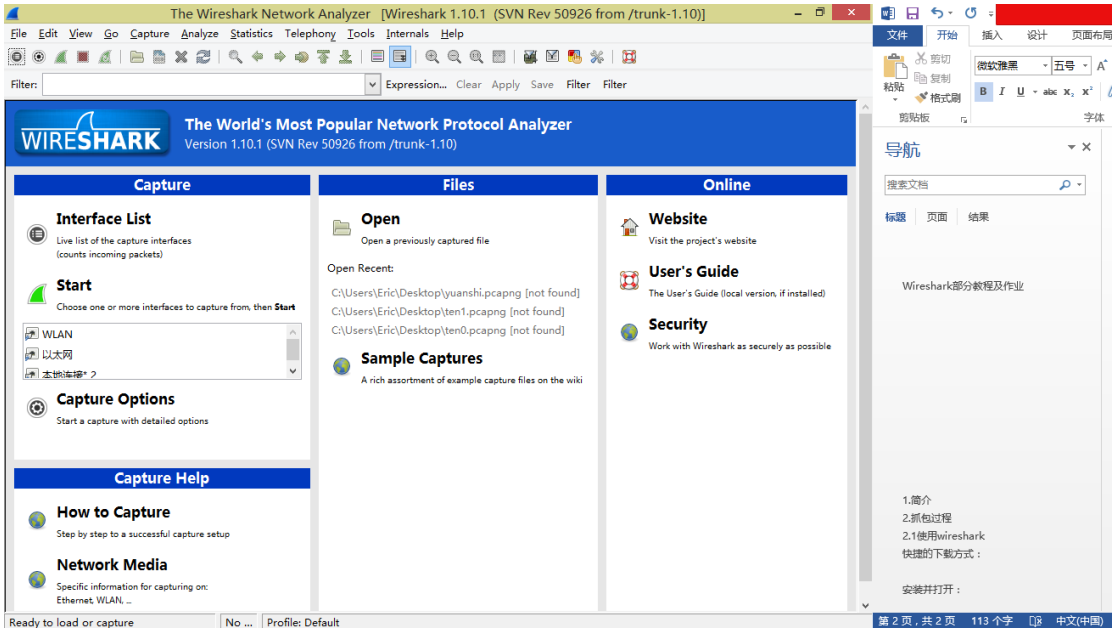
Wireshark（前称 Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark 使用 WinPCAP 作为接口，直接与网卡进行数据报文交换。

2.抓包过程

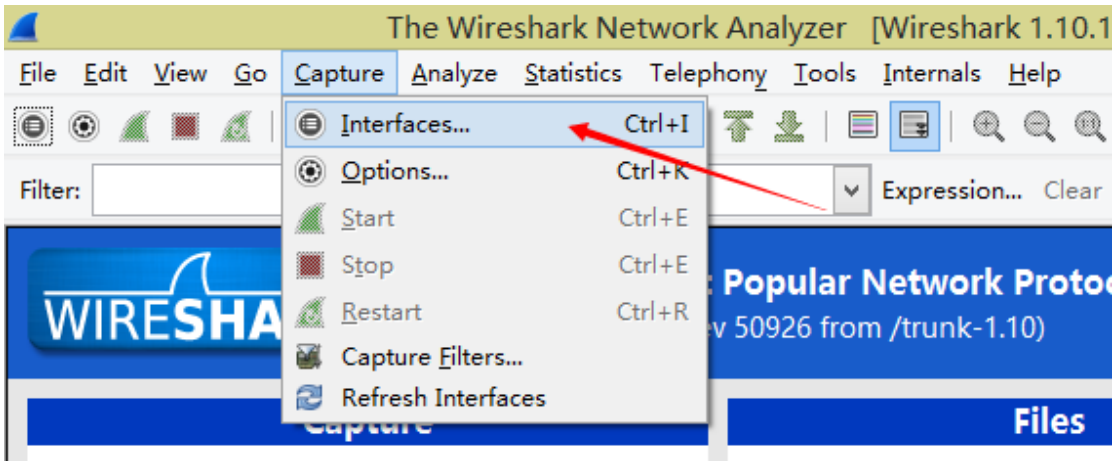
2.1 使用 wireshark

快捷的下载方式：

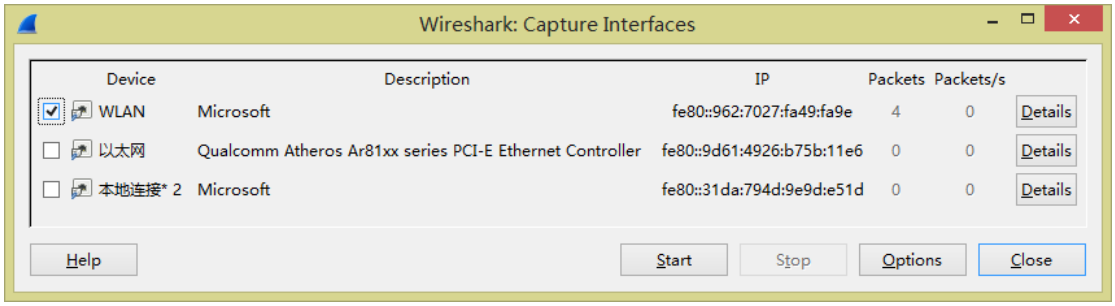
安装并打开：



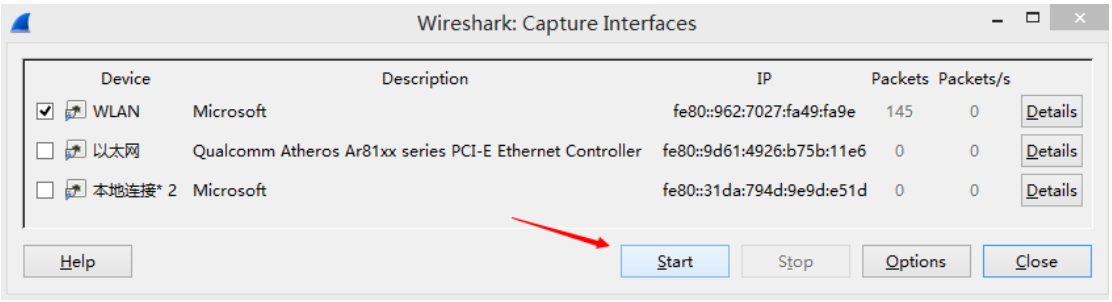
选择网卡：



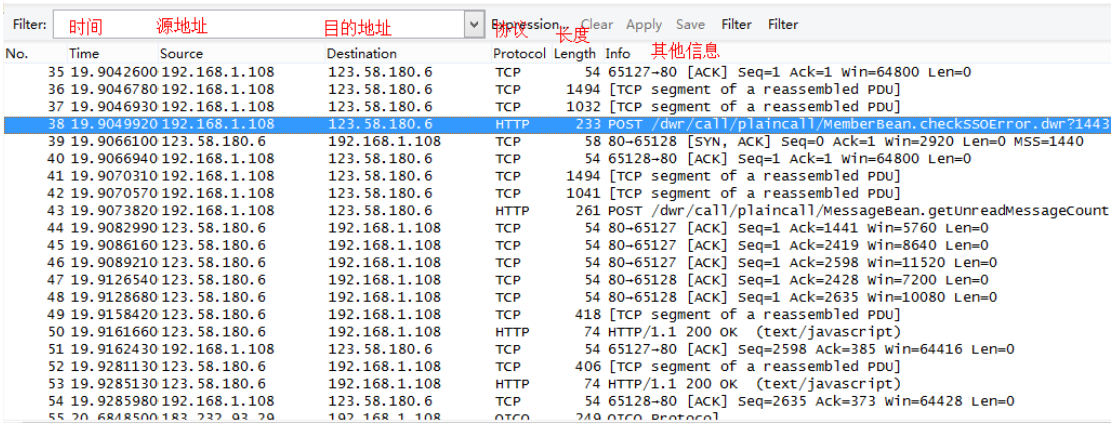
在联网状态下，选择有数据包的网卡即是当前应用网卡



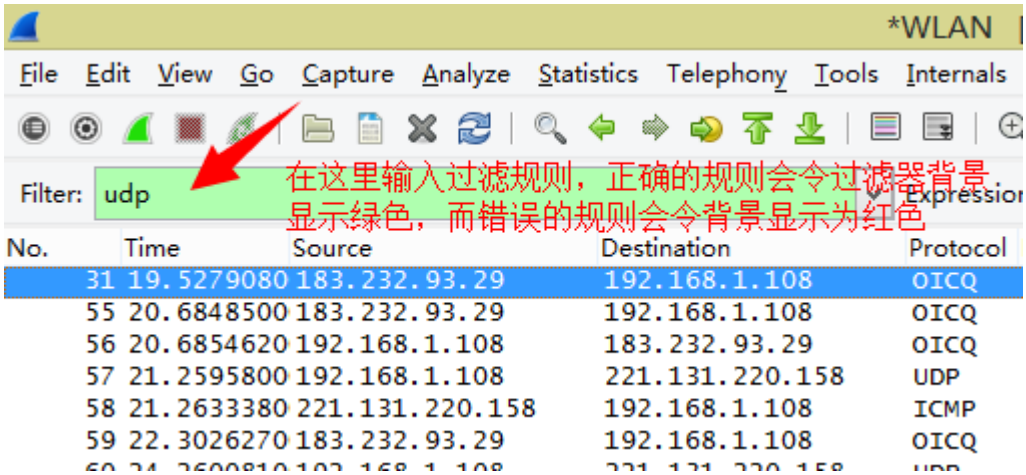
开始抓包：



wireshark 会将抓到的数据包用列表形式列在当前页面上，如图所示：



数据包过滤：



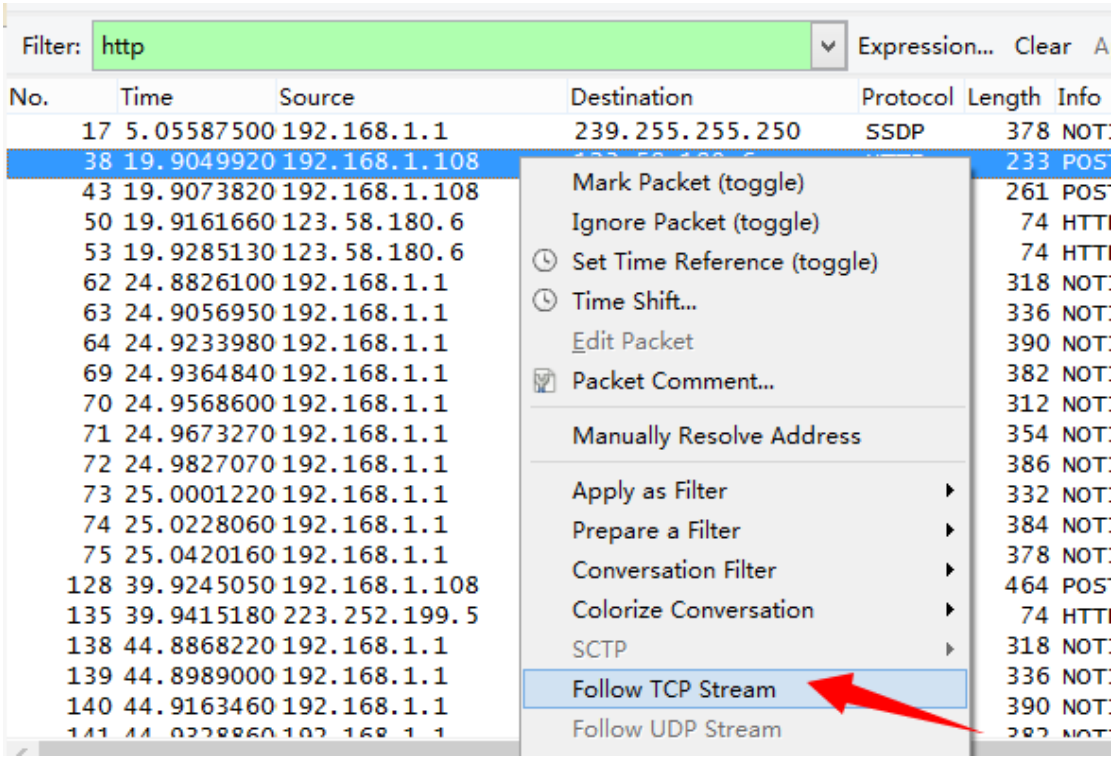
上图输入的过滤规则是：过滤所有使用 udp 协议的数据包，我们发现过滤出了一些 OICQ 协议的数据包，说明 QQ 的部分通信是基于 udp 协议而建立的。

更多过滤规则请看附件，过滤规则列表。

可以说，过滤与统计是 wireshark 的两大法宝，在数据包数量非常多的时候，我们可以通过输入适当的过滤规则巧妙地找到我们想分析的数据包。

跟踪 tcp 流：

找到 tcp 协议的数据包，右键点击跟踪 tcp 流可以让我们清晰地看到数据包所在的一次 tcp 数据交换中的所有流量，如下所示：



大家可以注意到，我在过滤器中输入的是 http 而非 tcp，这是因为 http 协议是基于 tcp 协议的，所以 http 数据包也支持 follow tcp stream 功能。

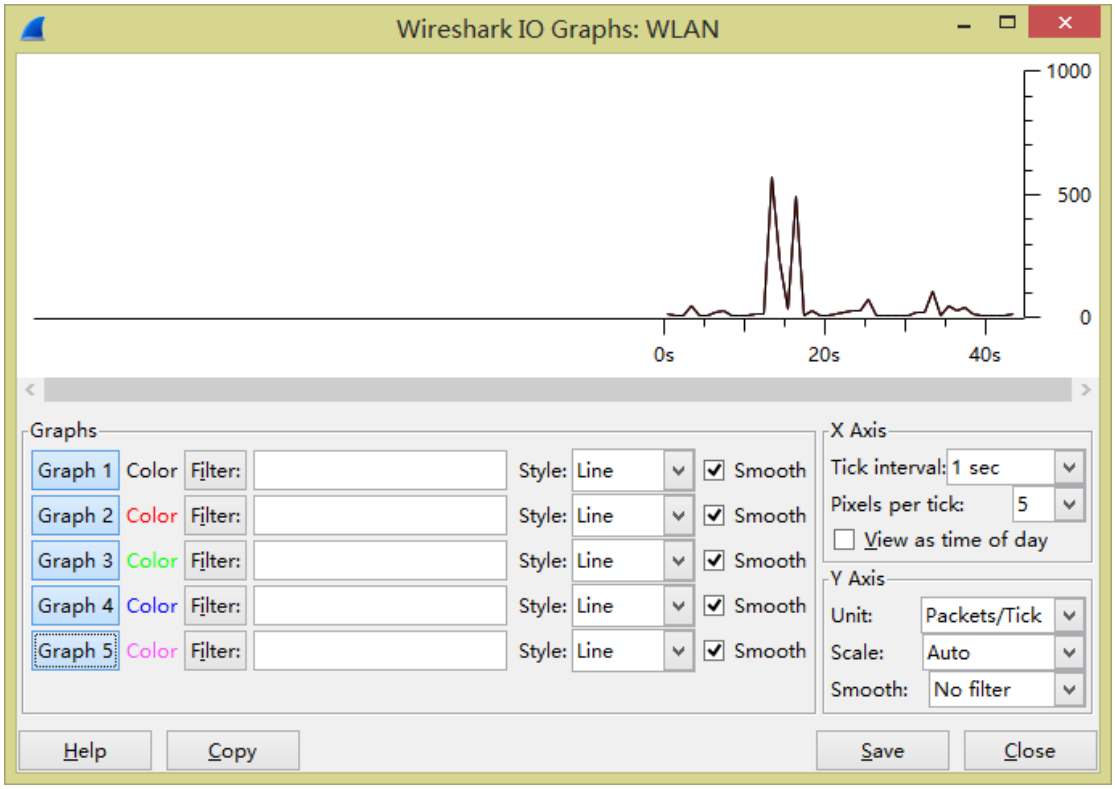
我们在 tcp 数据流中发现了我刚刚访问的是 study.163.com 也发现了其他和服务器交换的敏感数据，如 scriptsessionId 等信息，我们可以利用这些信息进行数据包伪造。

我们也看到了蓝色字体部分是服务器返回的信息，可以分析得到 163 使用 nginx 服务器等信息。

数据包统计：

statistics 是数据包统计选项，下面有多个小项用于数据包的统计分析。

如 io graph 选项如图：



有时针对大量数据包的统计分析可以快准狠的发现问题所在。

作业 1：

正确打开 wireshark，开始抓包后访问百度，然后过滤出目的地址为百度的所有数据包。（请参照附录中的过滤语法）

作业 2：

任意上网，抓取 3 次握手，4 次挥手的数据包。

作业 3：

使用 qq 聊天，然后过滤出 oicq 协议的数据包，根据源地址，目的地址分析两个人在聊天时是否能直接在两台电脑建立连接。如果不能直接链接，那么

你发给对方的数据包跑到了哪个城市的腾讯服务器？（提示：根据 ip 地址可以查询到服务器所在位置）

作业 4：

登陆 i.hdu.edu.cn。并从数据包中抓到自己的密码。（提示，follow tcp stream 中有 find 查找关键字功能）。

(提示：网页版数字杭电密码已经用 md5 加密过，可在 cmd5.com 尝试解谜)

附加作业：

使用 cain 等嗅探工具抓同局域网下（如寝室室友）的密码。

附件：过滤语法列表（部分）

```
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 ---> data

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