一、安装

直接点击下一步即可。

二、使用

Wireshark是非常流行的网络封包分析软件，可以截取各种网络数据包，并显示数据包详细信息。常用于开发测试过程各种问题定位。本文主要内容包括：

1、Wireshark软件下载和安装以及Wireshark主界面介绍。

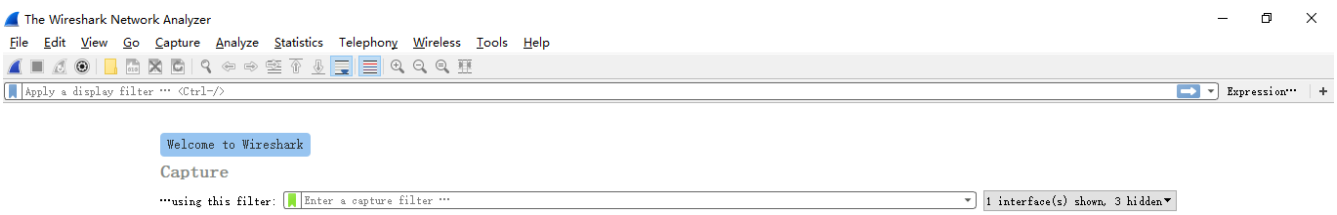
2、WireShark简单抓包示例。通过该例子学会怎么抓包以及如何简单查看分析数据包内容。

3、Wireshark过滤器使用。过滤器包含两种类型，一种是抓包过滤器，就是抓取前设置过滤规则。另外一种是显示过滤器，就是在数据包分析时进行过滤数据使用。通过过滤器可以筛选出想要分析的内容。包括按照协议过滤、端口和主机名过滤、数据包内容过滤。具体规则和实例可以查看正文。

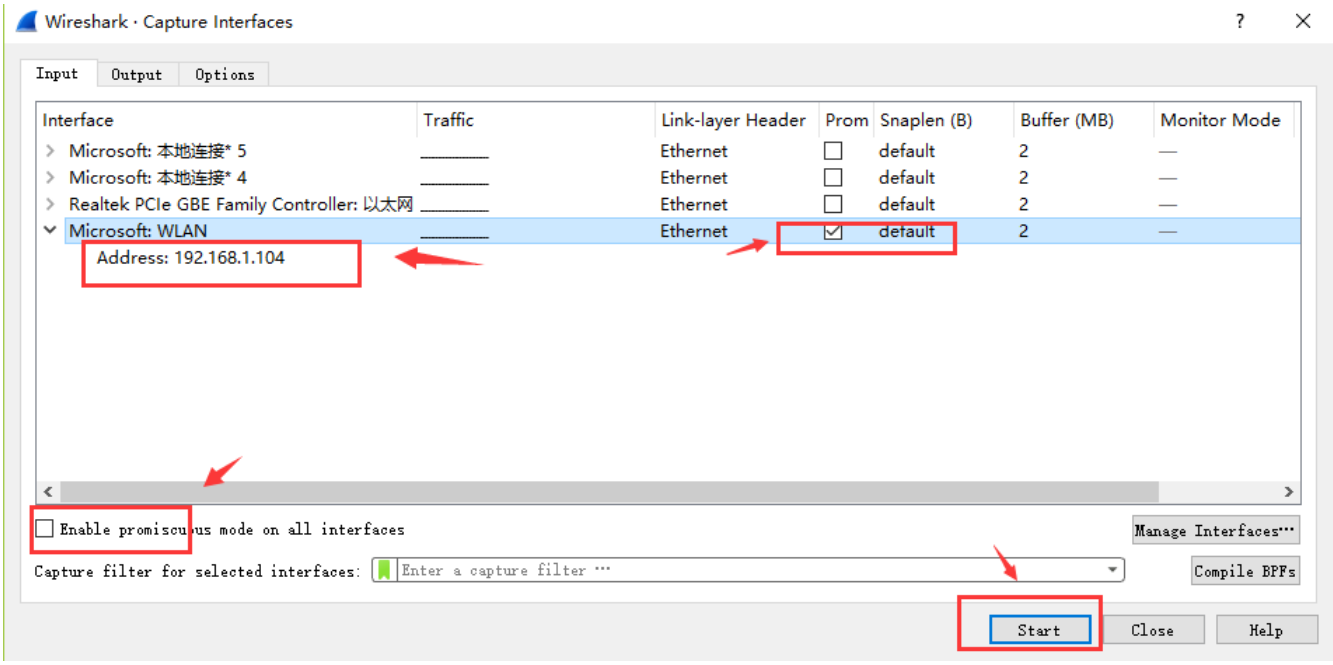
Wireshark 开始抓包示例

先介绍一个使用wireshark工具抓取ping命令操作的示例，让读者可以先上手操作感受一下抓包的具体过程。

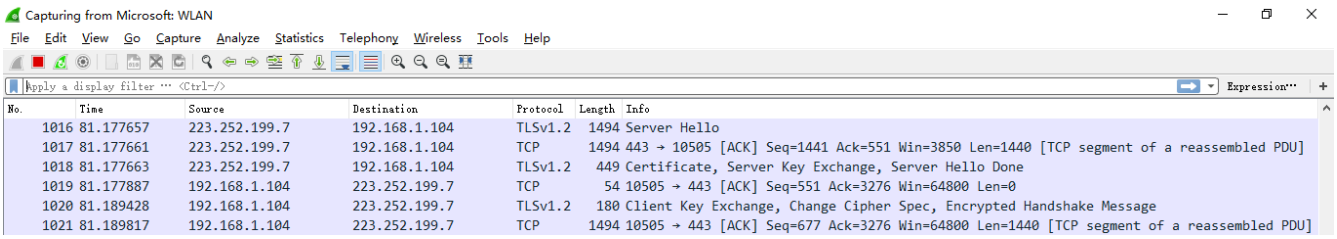
1、打开wireshark 2.6.5，主界面如下：



2、选择菜单栏上Capture -> Option，勾选WLAN网卡（这里需要根据各自电脑网卡使用情况选择，简单的办法可以看使用的IP对应的网卡）。点击Start。启动抓包。

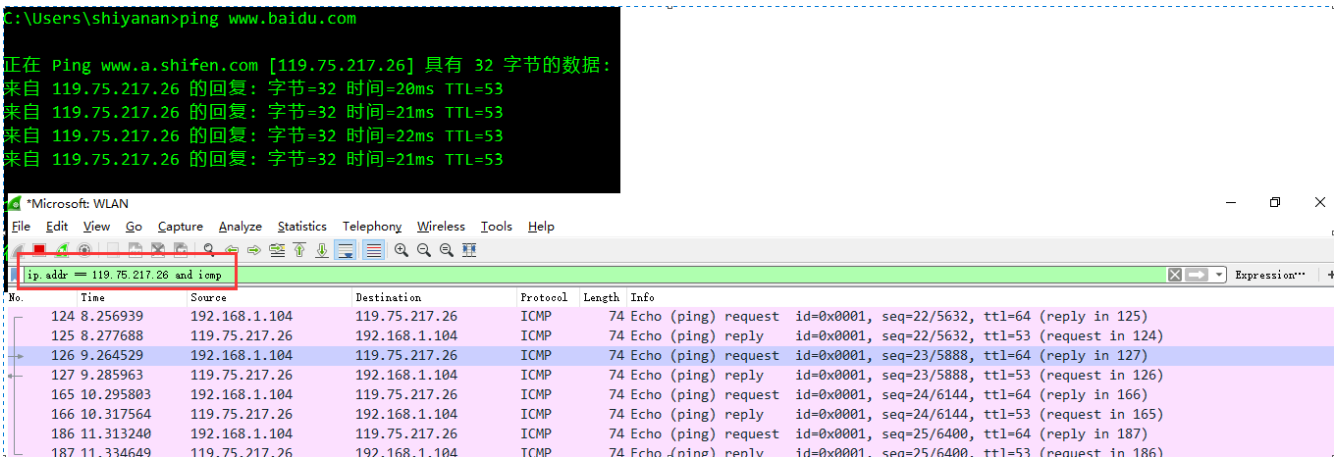


3、wireshark启动后，wireshark处于抓包状态中。



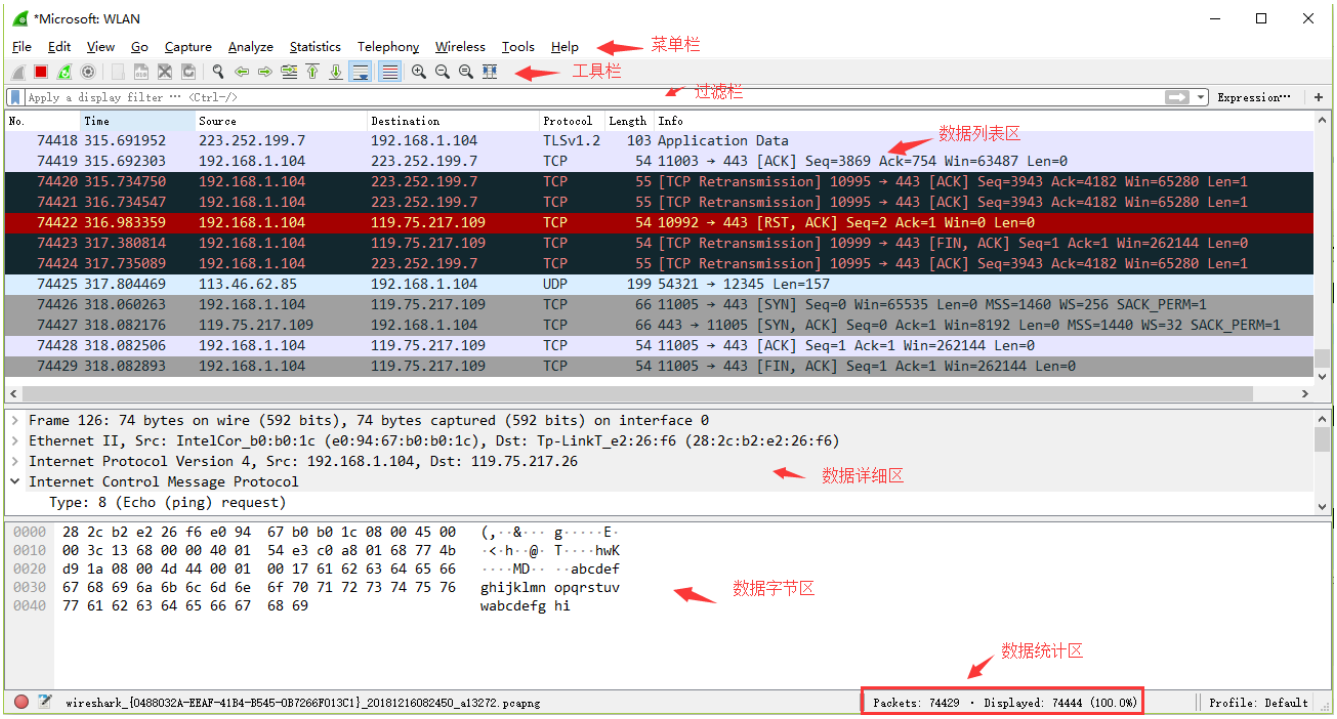
4、执行需要抓包的操作，如在cmd窗口下执行ping www.baidu.com。

5、操作完成后相关数据包就抓取到了。为避免其他无用的数据包影响分析，可以通过在过滤栏设置过滤条件进行数据包列表过滤，获取结果如下。说明：ip.addr == 119.75.217.26 and icmp 表示只显示ICPM协议且源主机IP或者目的主机IP为119.75.217.26的数据包。说明：协议名称icmp要小写。

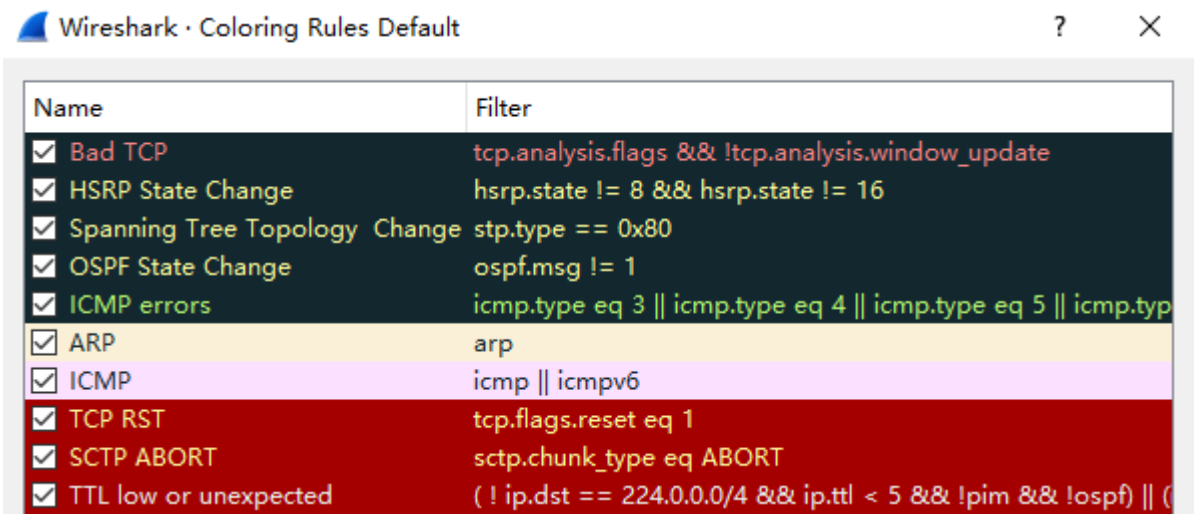


5、wireshark抓包完成，就这么简单。关于wireshark显示过滤条件、抓包过滤条件、以及如何查看数据包中的详细内容在后面介绍。

Wireshakr抓包界面介绍



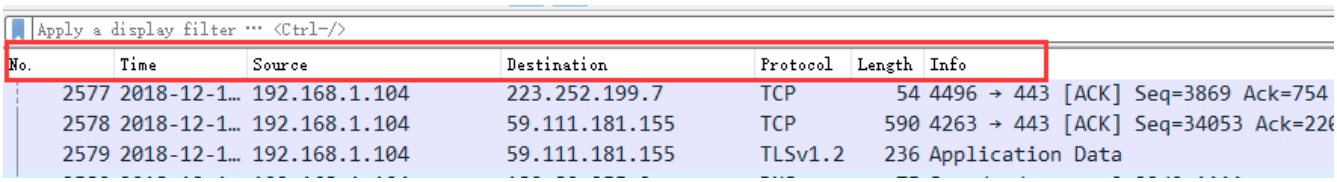
说明：数据包列表区中不同的协议使用了不同的颜色区分。协议颜色标识定位在菜单栏View --> Coloring Rules。如下所示



WireShark 主要分为这几个界面

\1. Display Filter(显示过滤器)， 用于设置过滤条件进行数据包列表过滤。菜单路径：Analyze --> Display Filters。

\2. Packet List Pane(数据包列表)， 显示捕获到的数据包，每个数据包包含编号，时间戳，源地址，目标地址，协议，长度，以及数据包信息。 不同协议的数据包使用了不同的颜色区分显示。



\3. Packet Details Pane(数据包详细信息), 在数据包列表中选择指定数据包，在数据包详细信息中会显示数据包的所有详细信息内容。数据包详细信息面板是最重要的，用来查看协议中的每一个字段。各行信息分别为

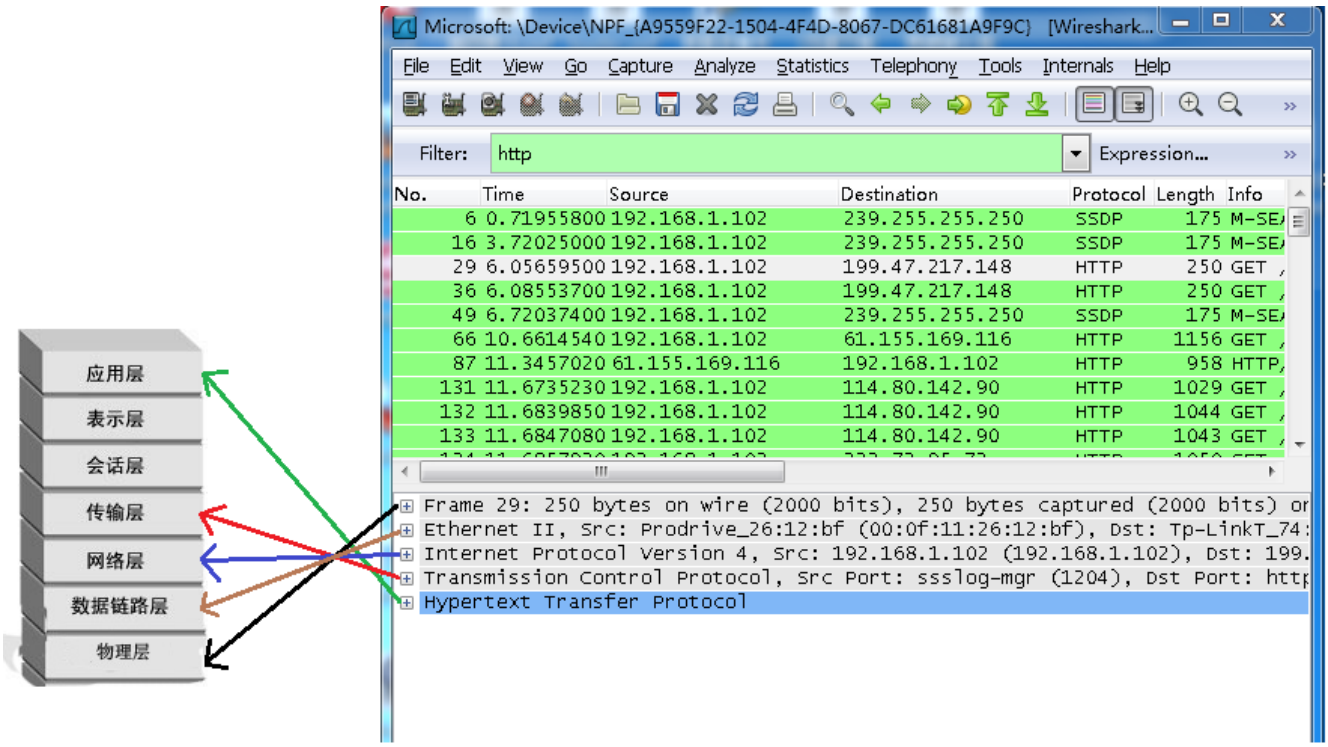
（1）Frame: 物理层的数据帧概况

（2）Ethernet II: 数据链路层以太网帧头部信息

（3）Internet Protocol Version 4: 互联网层IP包头部信息

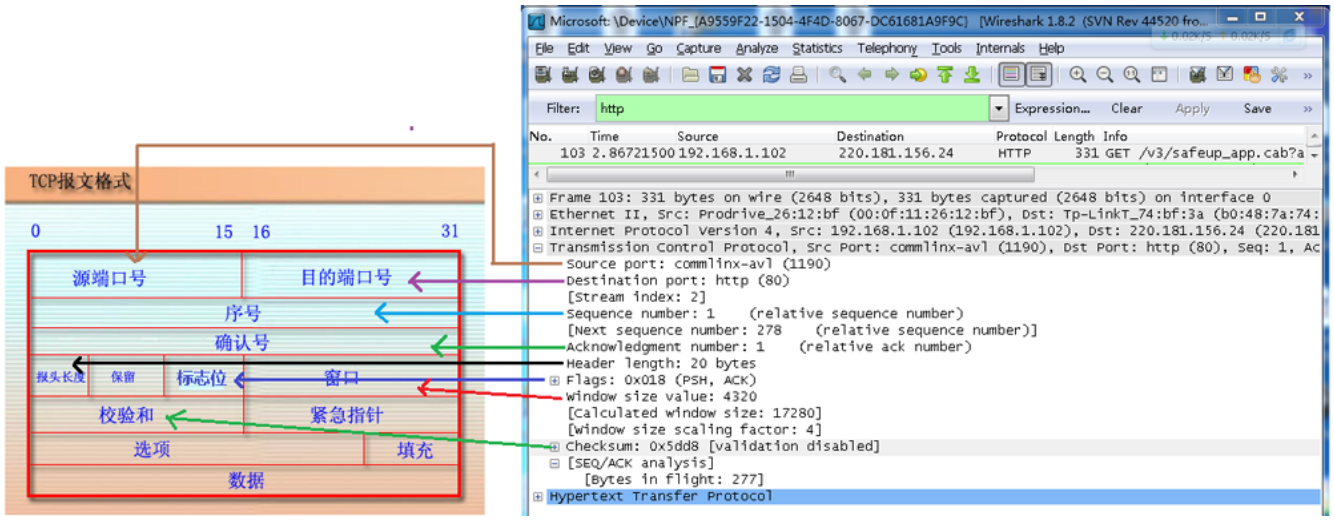
（4）Transmission Control Protocol: 传输层T的数据段头部信息，此处是TCP

（5）Hypertext Transfer Protocol: 应用层的信息，此处是HTTP协议



TCP包的具体内容

从下图可以看到wireshark捕获到的TCP包中的每个字段。



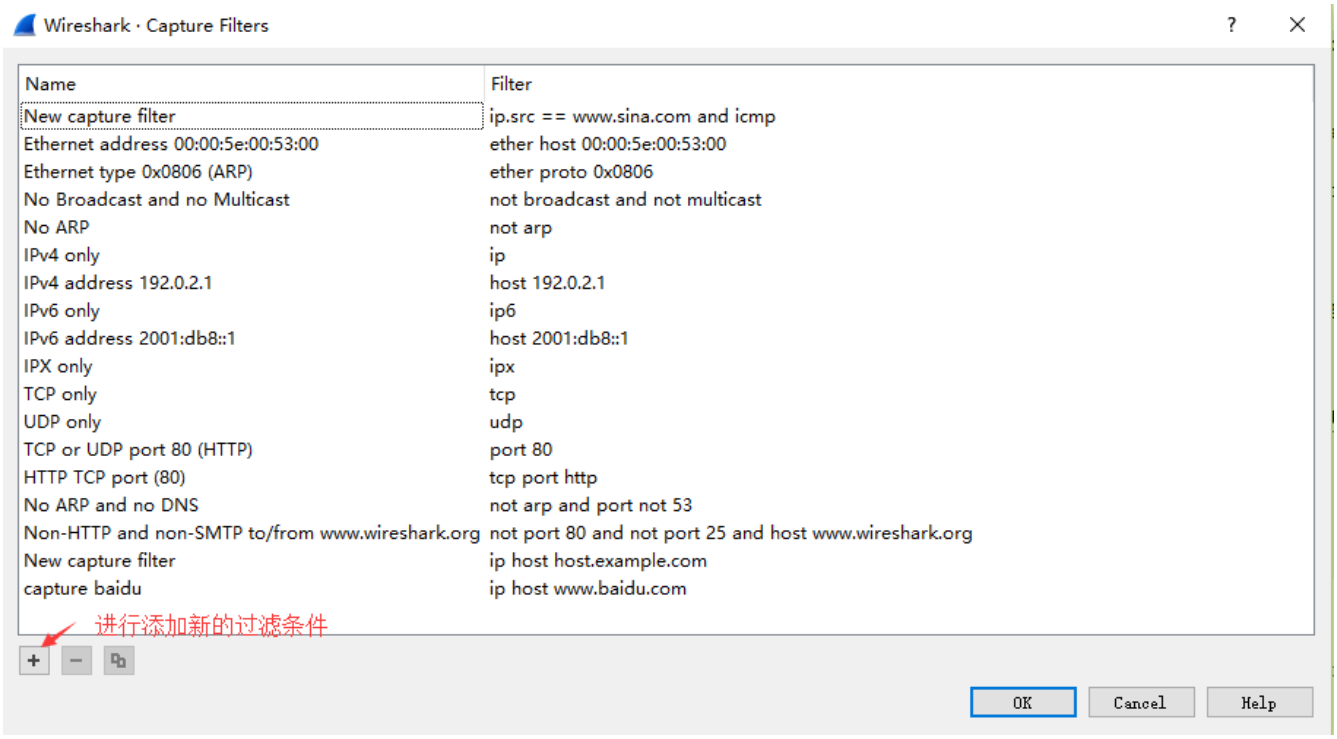
\4. Dissector Pane(数据包字节区)。

Wireshark过滤器设置

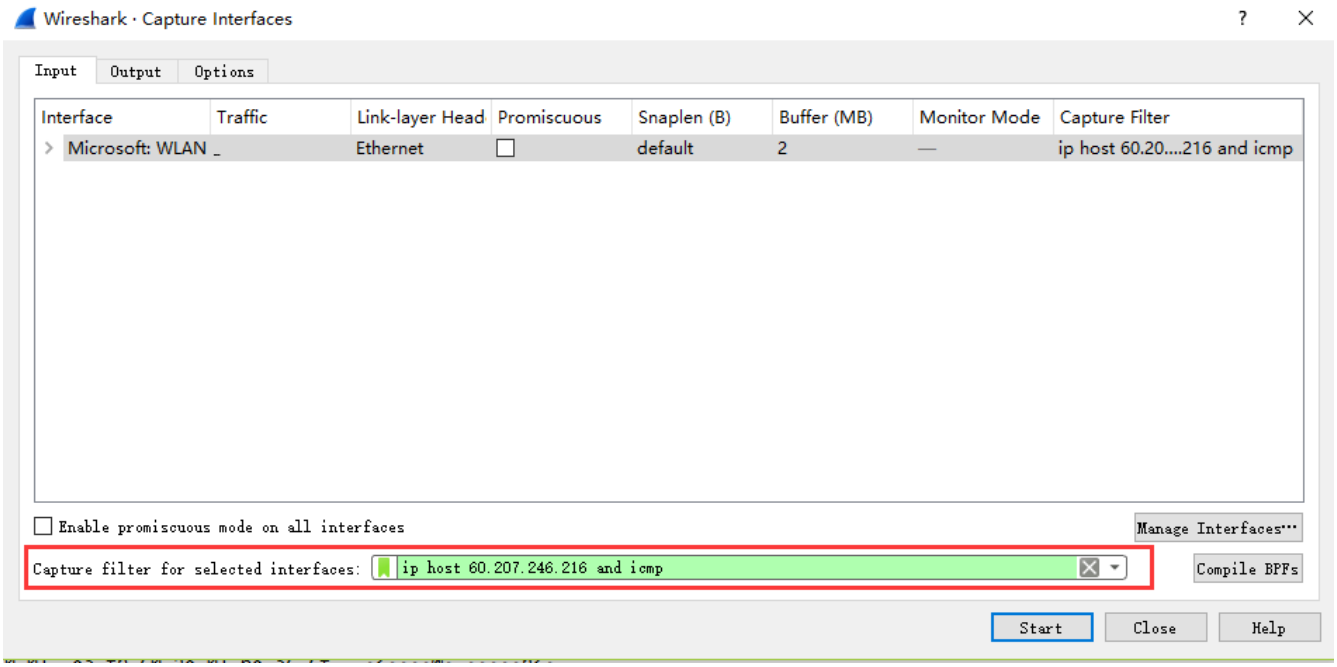
初学者使用wireshark时，将会得到大量的冗余数据包列表，以至于很难找到自己需要抓取的数据包部分。wireshark工具中自带了两种类型的过滤器，学会使用这两种过滤器会帮助我们在大量的数据中迅速找到我们需要的信息。

（1）抓包过滤器

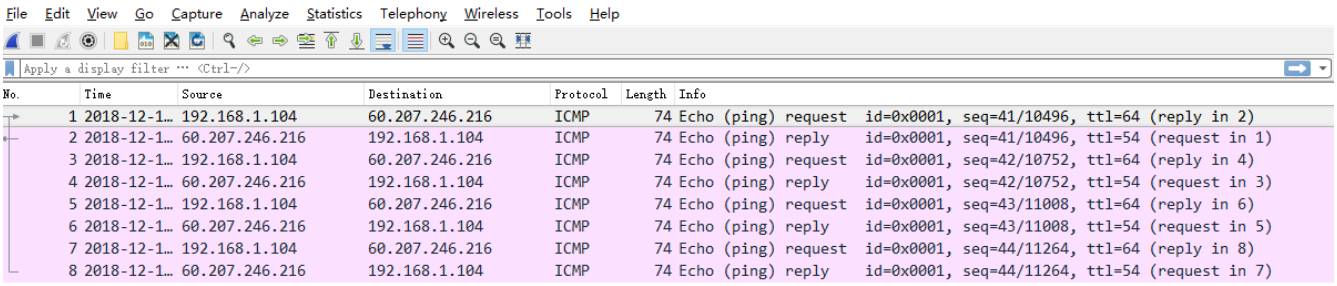
捕获过滤器的菜单栏路径为Capture --> Capture Filters。用于**在抓取数据包前设置。**



如何使用？可以在抓取数据包前设置如下。

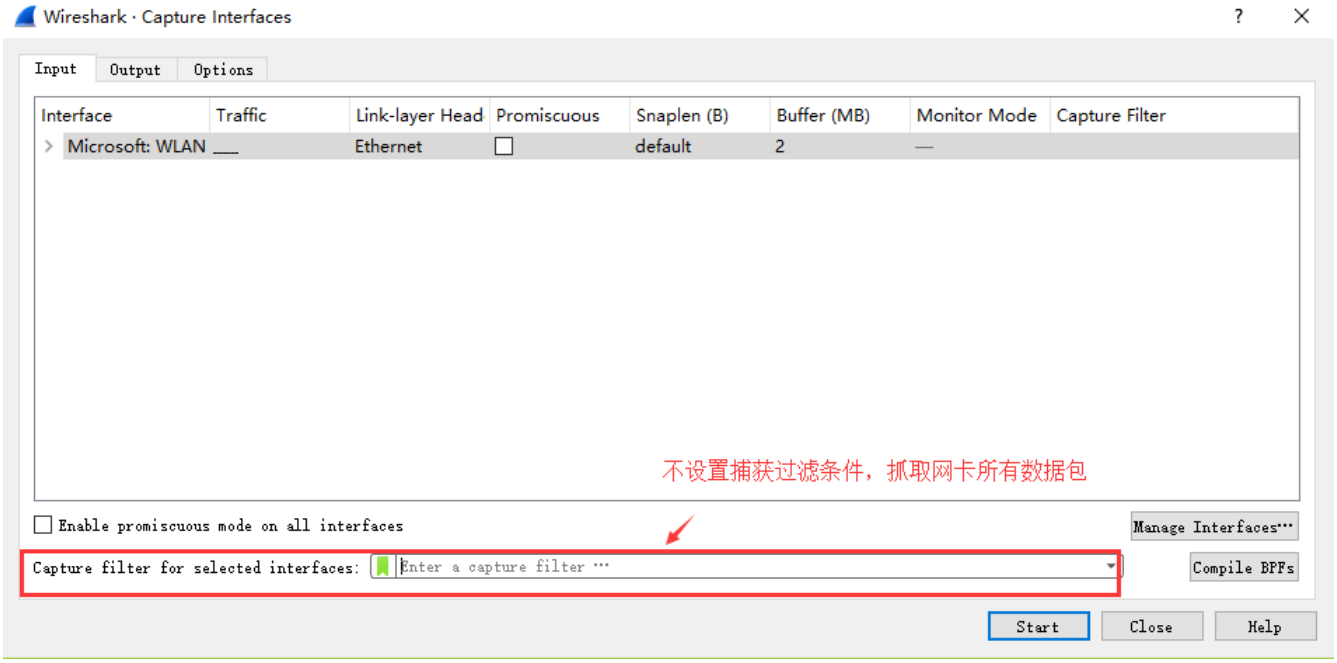


ip host 60.207.246.216 and icmp表示只捕获主机IP为60.207.246.216的ICMP数据包。获取结果如下：

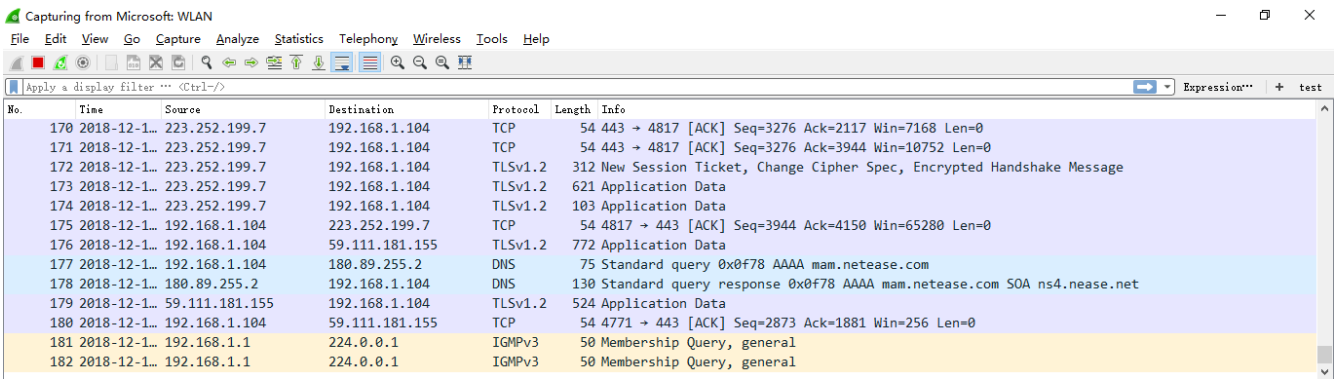


（2）显示过滤器

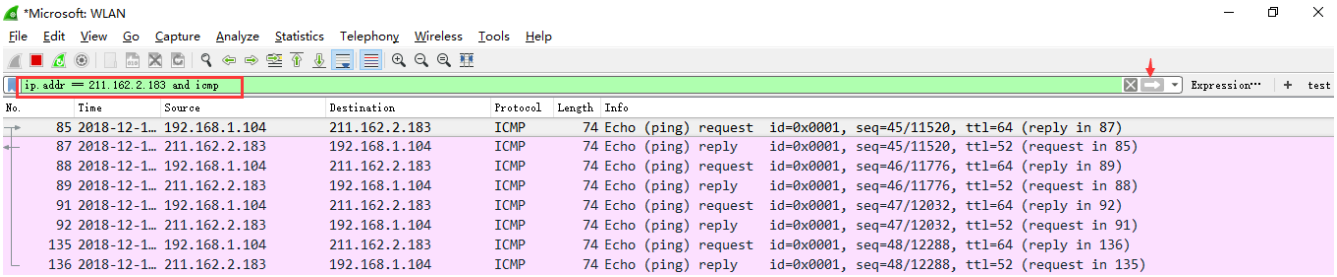
显示过滤器是用于在抓取数据包后设置过滤条件进行过滤数据包。通常是在抓取数据包时设置条件相对宽泛或者没有设置导致抓取的数据包内容较多时使用显示过滤器设置条件过滤以方便分析。同样上述场景，在捕获时未设置抓包过滤规则直接通过网卡进行抓取所有数据包，如下



执行ping www.huawei.com获取的数据包列表如下



观察上述获取的数据包列表，含有大量的无效数据。这时可以通过设置显示器过滤条件进行提取分析信息。ip.addr == 211.162.2.183 and icmp。并进行过滤。



上述介绍了抓包过滤器和显示过滤器的基本使用方法。**在组网不复杂或者流量不大情况下，使用显示器过滤器进行抓包后处理就可以满足我们使用。**下面介绍一下两者间的语法以及它们的区别。

wireshark过滤器表达式的规则

1、抓包过滤器语法和实例

抓包过滤器类型Type（host、net、port）、方向Dir（src、dst）、协议Proto（ether、ip、tcp、udp、http、icmp、ftp等）、逻辑运算符（&& 与、|| 或、！非）

（1）协议过滤

比较简单，直接在抓包过滤框中直接输入协议名即可。

tcp，只显示TCP协议的数据包列表

http，只查看HTTP协议的数据包列表

icmp，只显示ICMP协议的数据包列表

（2）IP过滤

host 192.168.1.104

src host 192.168.1.104

dst host 192.168.1.104

（3）端口过滤

port 80

src port 80

dst port 80

（4）逻辑运算符&& 与、|| 或、！非

src host 192.168.1.104 && dst port 80 抓取主机地址为192.168.1.80、目的端口为80的数据包

host 192.168.1.104 || host 192.168.1.102 抓取主机为192.168.1.104或者192.168.1.102的数据包

！broadcast 不抓取广播数据包

2、显示过滤器语法和实例

（1）比较操作符

比较操作符有== 等于、！= 不等于、> 大于、< 小于、>= 大于等于、<=小于等于。

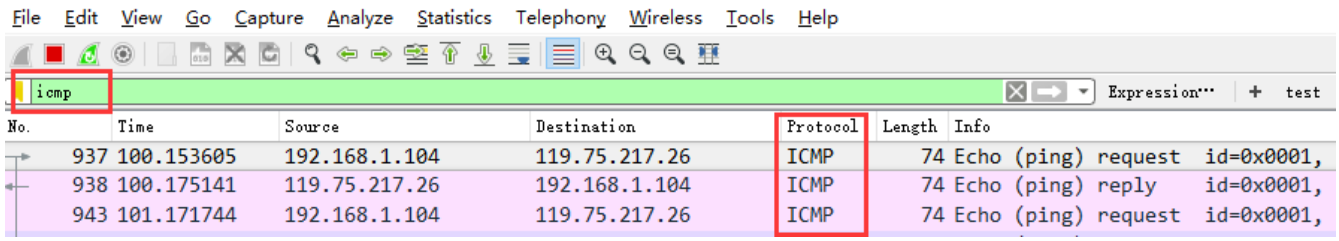
（2）协议过滤

比较简单，直接在Filter框中直接输入协议名即可。**注意：协议名称需要输入小写。**

tcp，只显示TCP协议的数据包列表

http，只查看HTTP协议的数据包列表

icmp，只显示ICMP协议的数据包列表

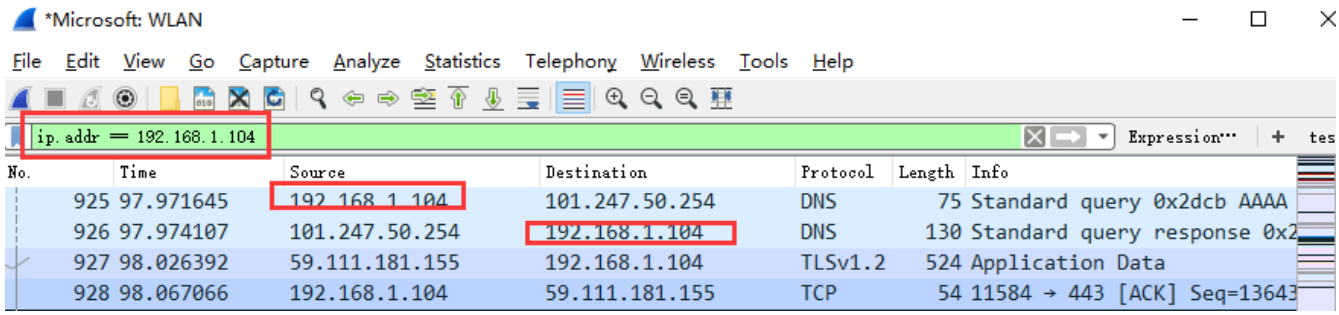


（3） ip过滤

ip.src ==192.168.1.104 显示源地址为192.168.1.104的数据包列表

ip.dst==192.168.1.104, 显示目标地址为192.168.1.104的数据包列表

ip.addr == 192.168.1.104 显示源IP地址或目标IP地址为192.168.1.104的数据包列表

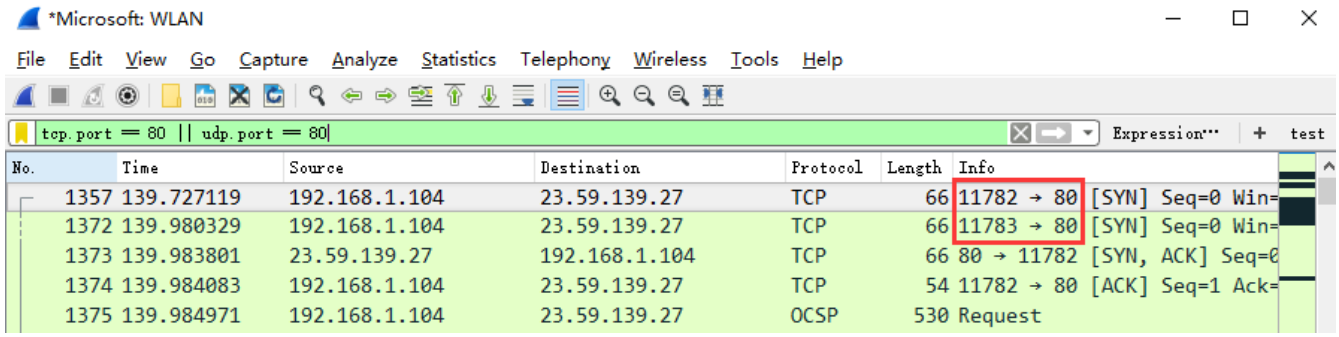


（4）端口过滤

tcp.port ==80, 显示源主机或者目的主机端口为80的数据包列表。

tcp.srcport == 80, 只显示TCP协议的源主机端口为80的数据包列表。

tcp.dstport == 80，只显示TCP协议的目的主机端口为80的数据包列表。

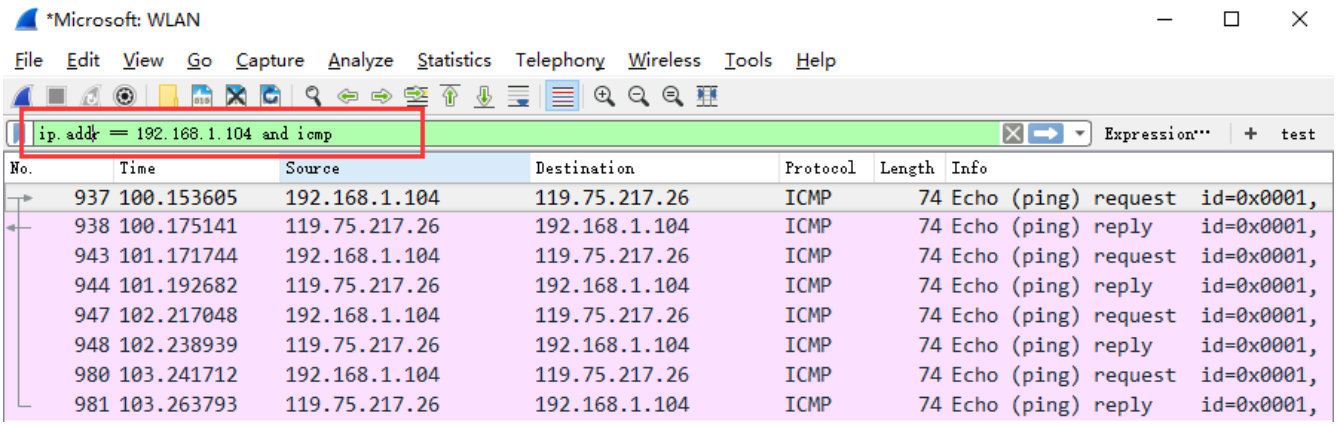


（5） Http模式过滤

http.request.method=="GET", 只显示HTTP GET方法的。

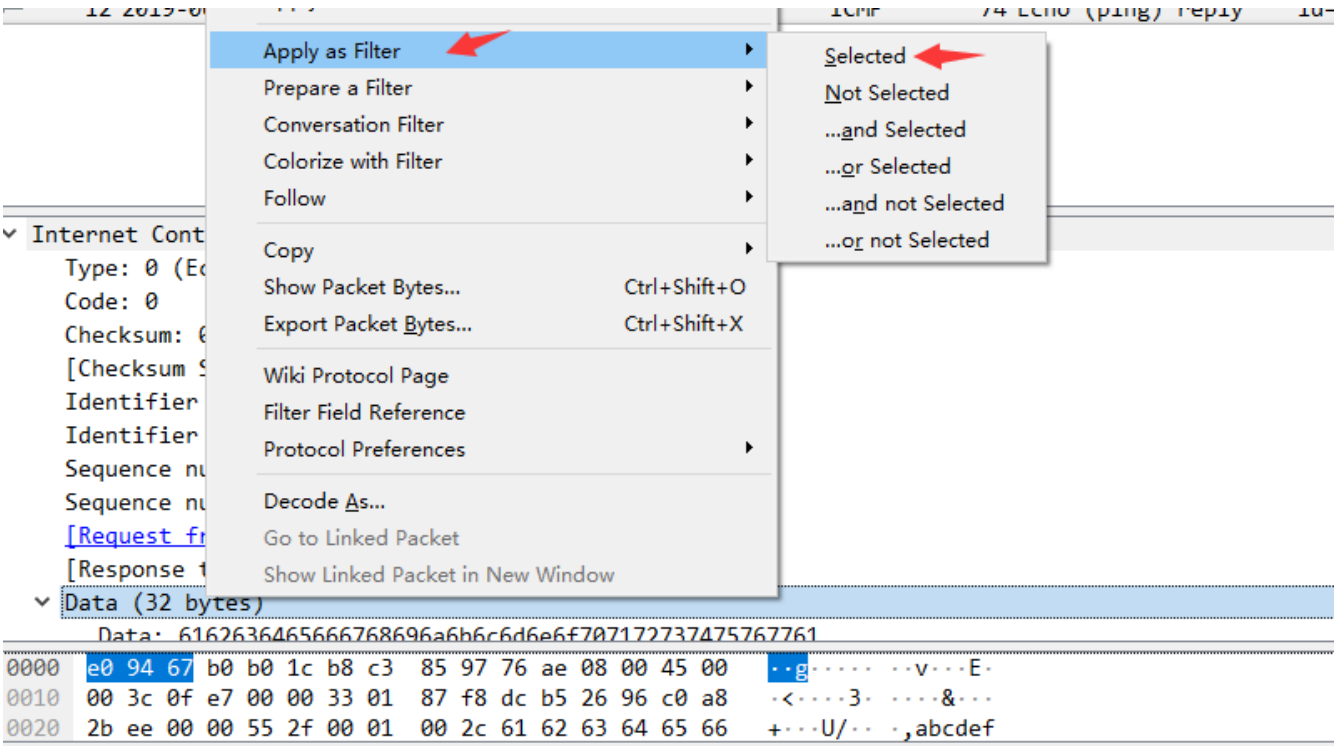
（6）逻辑运算符为 and/or/not

过滤多个条件组合时，使用and/or。比如获取IP地址为192.168.1.104的ICMP数据包表达式为ip.addr == 192.168.1.104 and icmp

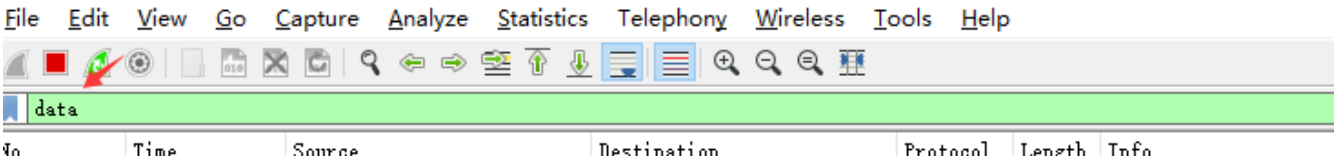


（7）按照数据包内容过滤。假设我要以IMCP层中的内容进行过滤，可以单击选中界面中的码流，在下方进行选中数据。如下

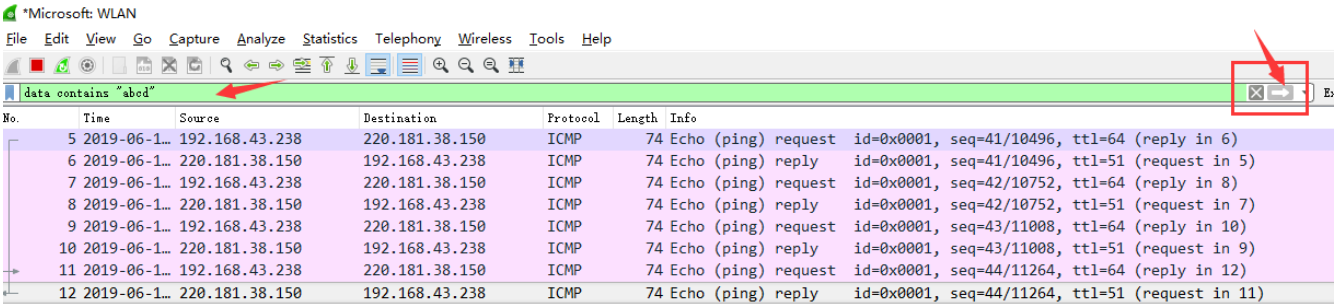
右键单击选中后出现如下界面



选中Select后在过滤器中显示如下



后面条件表达式就需要自己填写。如下我想过滤出data数据包中包含"abcd"内容的数据流。**包含的关键词是contains 后面跟上内容。**



看到这， 基本上对wireshak有了初步了解。

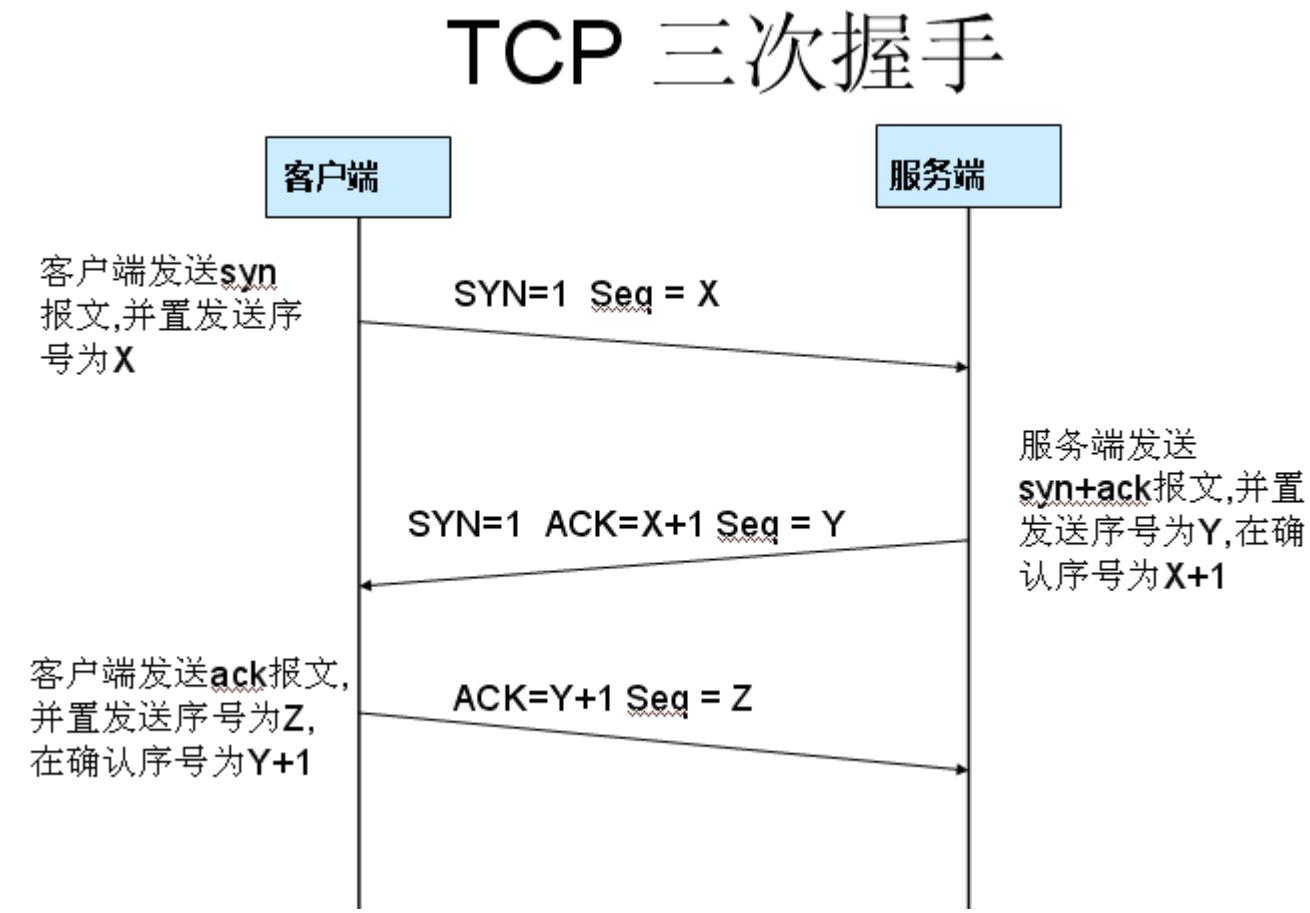
Wireshark抓包分析TCP三次握手

（1）TCP三次握手连接建立过程

Step1：客户端发送一个SYN=1，ACK=0标志的数据包给服务端，请求进行连接，这是第一次握手；

Step2：服务端收到请求并且允许连接的话，就会发送一个SYN=1，ACK=1标志的数据包给发送端，告诉它，可以通讯了，并且让客户端发送一个确认数据包，这是第二次握手；

Step3：服务端发送一个SYN=0，ACK=1的数据包给客户端端，告诉它连接已被确认，这就是第三次握手。TCP连接建立，开始通讯。



（2）wireshark抓包获取访问指定服务端数据包

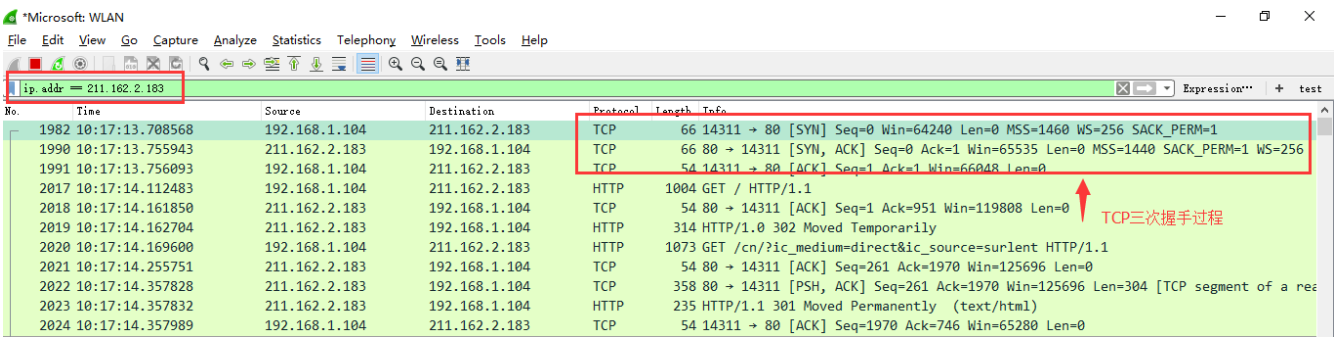
Step1：启动wireshark抓包，打开浏览器输入www.huawei.com。

Step2：使用ping www.huawei.com获取IP。

```
C:\Users\shiyanan>ping www.huawei.com

正在 Ping huawei.dtwscache.ourwebcdn.com [211.162.2.183] 具有 32 字节的数据:
来自 211.162.2.183 的回复: 字节=32 时间=57ms TTL=51
```

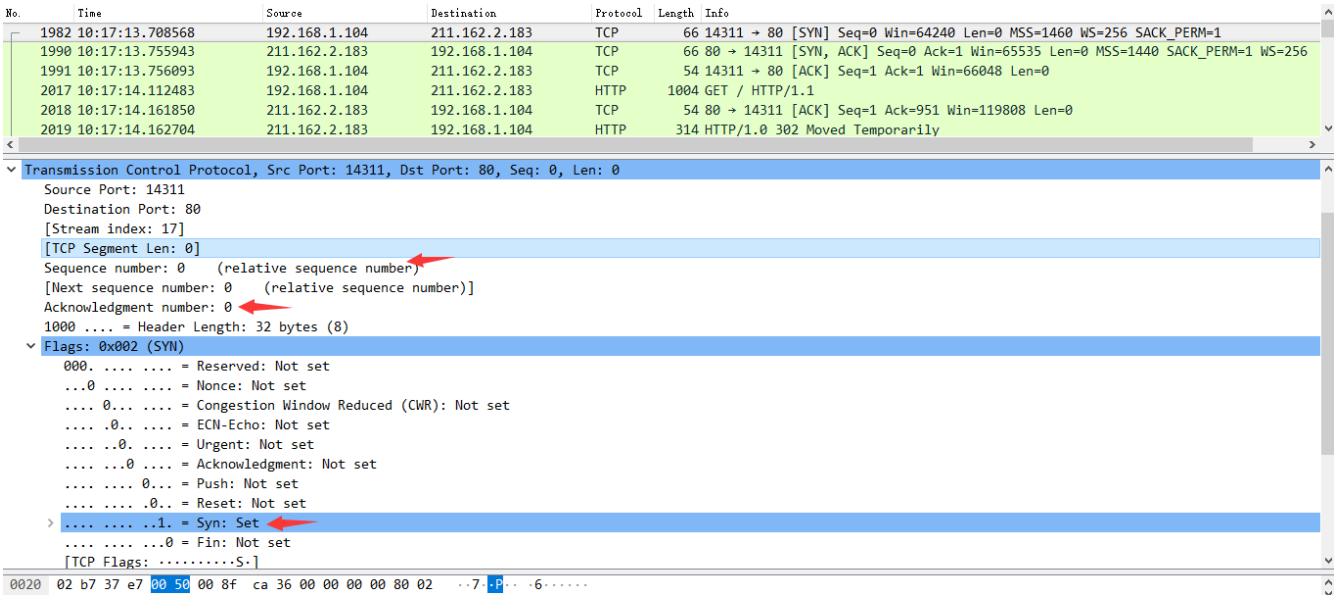
Step3：输入过滤条件获取待分析数据包列表 ip.addr == 211.162.2.183



图中可以看到wireshark截获到了三次握手的三个数据包。第四个包才是HTTP的， 这说明HTTP的确是使用TCP建立连接的。

第一次握手数据包

客户端发送一个TCP，标志位为SYN，序列号为0， 代表客户端请求建立连接。 如下图。



数据包的关键属性如下：

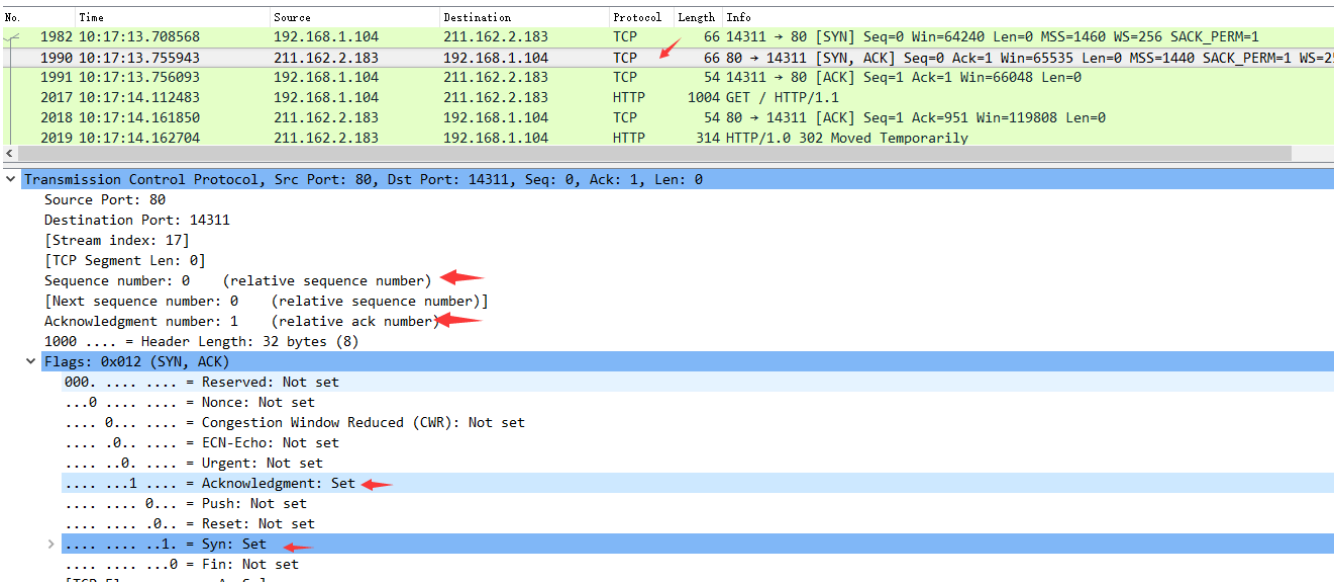
SYN ： 标志位，表示请求建立连接

Seq = 0 ： 初始建立连接值为0，数据包的相对序列号从0开始，表示当前还没有发送数据

Ack =0：初始建立连接值为0，已经收到包的数量，表示当前没有接收到数据

第二次握手的数据包

服务器发回确认包, 标志位为 SYN,ACK. 将确认序号(Acknowledgement Number)设置为客户的I S N加1以.即0+1=1, 如下图



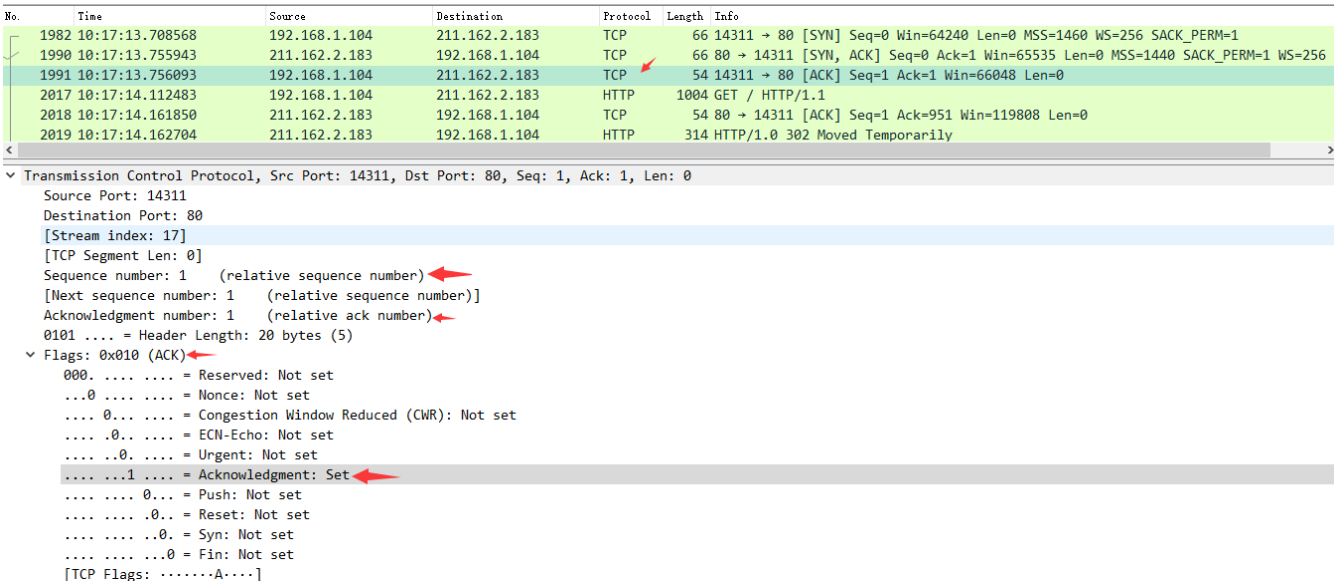
数据包的关键属性如下：

Seq = 0 ： 初始建立值为0，表示当前还没有发送数据

Ack = 1：表示当前端成功接收的数据位数，虽然客户端没有发送任何有效数据，确认号还是被加1，因为包含SYN或FIN标志位。（并不会对有效数据的计数产生影响，因为含有SYN或FIN标志位的包并不携带有效数据）

第三次握手的数据包

客户端再次发送确认包(ACK) SYN标志位为0,ACK标志位为1.并且把服务器发来ACK的序号字段+1,放在确定字段中发送给对方.并且在数据段放写ISN的+1, 如下图:



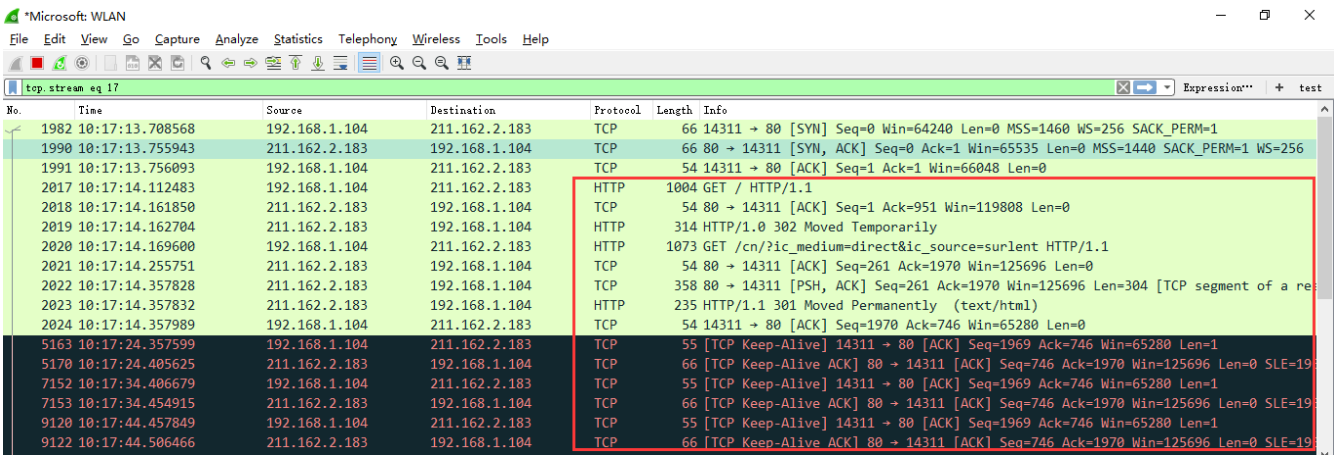
数据包的关键属性如下:

ACK： 标志位，表示已经收到记录

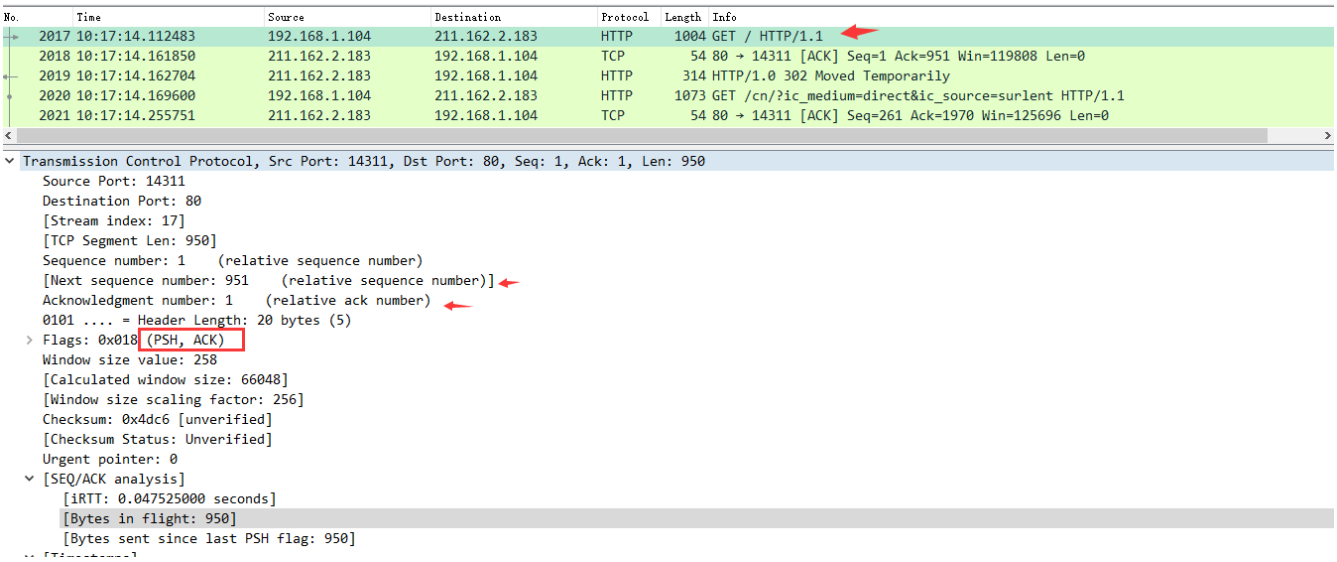
Seq = 1 ： 表示当前已经发送1个数据

Ack = 1 : 表示当前端成功接收的数据位数，虽然服务端没有发送任何有效数据，确认号还是被加1，因为包含SYN或FIN标志位（并不会对有效数据的计数产生影响，因为含有SYN或FIN标志位的包并不携带有效数据)。

就这样通过了TCP三次握手，建立了连接。开始进行数据交互



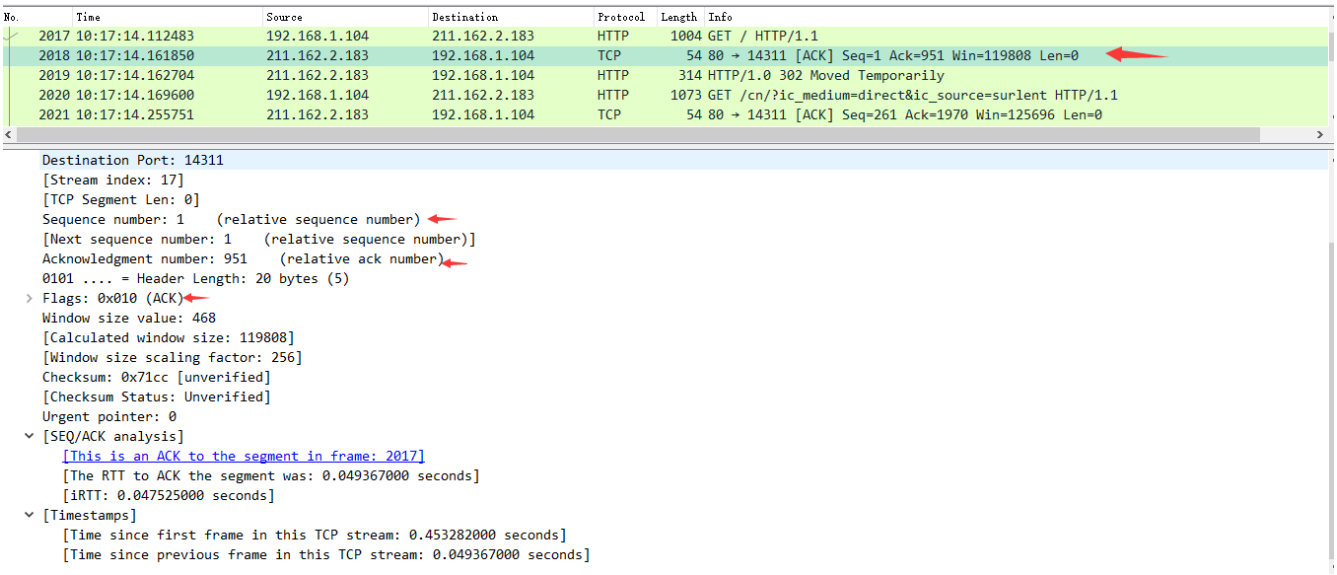
下面针对数据交互过程的数据包进行一些说明:



数据包的关键属性说明

Seq: 1

Ack: 1: 说明现在共收到1字节数据



Seq: 1 Ack: 951: 说明现在服务端共收到951字节数据

在TCP层，有个FLAGS字段，这个字段有以下几个标识：SYN, FIN, ACK, PSH, RST, URG。如下

```
Flags: 0x012 (SYN, ACK)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ·····A··S·]
```

其中，对于我们日常的分析有用的就是前面的五个字段。它们的含义是：SYN表示建立连接，FIN表示关闭连接，ACK表示响应，PSH表示有DATA数据传输，RST表示连接重置。

Wireshark分析常用操作

调整数据包列表中时间戳显示格式。调整方法为View -->Time Display Format --> Date and Time of Day。调整后格式如下：

No.	Time	Source	Destination	Protocol	Length	Info
1986	2018-12-16 11:37:44.583026	211.162.2.183	192.168.1.104	TCP	54	443 → 2206 [FIN, ACK] Seq=73867 Ack=3410 Win=137472 Len=0
1987	2018-12-16 11:37:44.583302	192.168.1.104	211.162.2.183	TCP	54	2201 → 443 [ACK] Seq=3435 Ack=64910 Win=66048 Len=0
1988	2018-12-16 11:37:44.583441	192.168.1.104	211.162.2.183	TCP	54	2201 → 443 [FIN, ACK] Seq=3435 Ack=64910 Win=66048 Len=0
1989	2018-12-16 11:37:44.583460	192.168.1.104	211.162.2.183	TCP	54	2206 → 443 [ACK] Seq=3410 Ack=73868 Win=66048 Len=0
1990	2018-12-16 11:37:44.583904	192.168.1.104	211.162.2.183	TCP	54	2206 → 443 [FIN, ACK] Seq=3410 Ack=73868 Win=66048 Len=0
1991	2018-12-16 11:37:44.587146	211.162.2.183	192.168.1.104	TLSv1.2	85	Encrypted Alert

参考文档：

Wireshark User's Guide