Wireshark(守夜人Jaden-吴老板)

一、介绍

Wireshark（前身是Ethereal）是一个网络封包分析软件，目前是全球使用最广泛的开源抓包软件，别名小鲨鱼或者鲨鱼鳍。网络封包分析软件的功能是截取网卡进出的网络封包，并尽可能显示出最为详细的网络封包资料，它能够检测并解析各种协议，包括以太网、WIFI、TCP/IP和HTTP协议等等。Wireshark使用LibPCAP、WinPCAP(现在普遍使用的是Npcap)作为驱动程序，他们提供了通用的抓包接口，直接与网卡进行数据报文交换，WinPCAP本身就是抓包分析工具，Wireshark通过对他进行整合加工丰富出来的产物，所以安装Wireshark的时候会提示我们安装WinPCAP。

网络管理员使用Wireshark来检测网络问题、网络故障情况，比如网络连接没问题就是上不了网、获取不到IP地址之类的工具都可以登录，但是就打不开网页，网络安全工程师使用Wireshark来检查网络安全相关问题，比如网络真实流量、攻击流量分析、对黑客的渗透攻击进行快速定位并找到攻击源等等，开发者使用Wireshark来为新的通讯协定排错，普通使用者使用Wireshark来学习网络协定的相关知识。当然，有的人也会"居心叵测"的用它来寻找一些敏感信息，大多数的黑客仅仅为了探测内部网上的主机并取得控制权，只有那些"雄心勃勃"的黑客，为了控制整个网络才会安装特洛伊木马和后门程序等等。他们经常使用的手法是安装嗅探抓包工具。在内部网上，黑客要想迅速获得大量的账号（包括用户名和密码），最为有效的手段是使用嗅探抓包工具程序。这种方法要求运行嗅探抓包程序的主机和被监听的主机必须在同一个以太网段上，故而在外部主机上运行嗅探抓包工具是没有效果的。再者，必须以管理员的身份使用嗅探抓包工具程序，才能够监听到以太网段上的数据流，所以说做网络安全相关工作Wireshark等嗅探抓包工具是必须要学会的。
Wireshark不是入侵侦测系统（Intrusion Detection System,IDS）。对于网络上的异常流量行为，Wireshark不会产生警示或是任何提示。然而，仔细分析Wireshark截取的封包能够帮助使用者对于网络行为有更清楚的了解。Wireshark不会对网络封包产生内容的修改，它只会反映出流通的封包资讯。Wireshark本身也不会送出封包至网络上。市面上有很多的流量检测安全设备相当于内置了Wireshark，其实也是基于WinPCAP等工具开发出来的，比如IDS入侵检测设备、态势感知设备等等，我们使用这些安全设备的时候，经常会看到pcap格式的数据包。

声明：为了安全考虑，wireshark只能查看封包，而不能修改封包的内容，或者发送封包。wireshark能获取HTTP，也能获取HTTPS，但是不能解密HTTPS，所以wireshark看不懂HTTPS中的内容。如果是处理HTTP，HTTPS还是用Fiddler、Burpsuite、Charles等，其他协议比如TCP、UDP 就用wireshark。除了Wireshark之外，还有Sniffer Pro(Windows平台)、Tcpdump(Linux平台)、WinDump(Windows平台)，其中Wireshark具备跨平台性。其实Sniffer是嗅探的意思，上面说到的这些工具都可以称之为Sniffer工具。

二、下载安装

吴老板的操作系统是win11，安装的wireshark我们下载最新的安装。

下载地址：https://www.wireshark.org/download.html

学习网址：https://www.chappell-university.com/books

WIKI：https://wiki.wireshark.org/Home

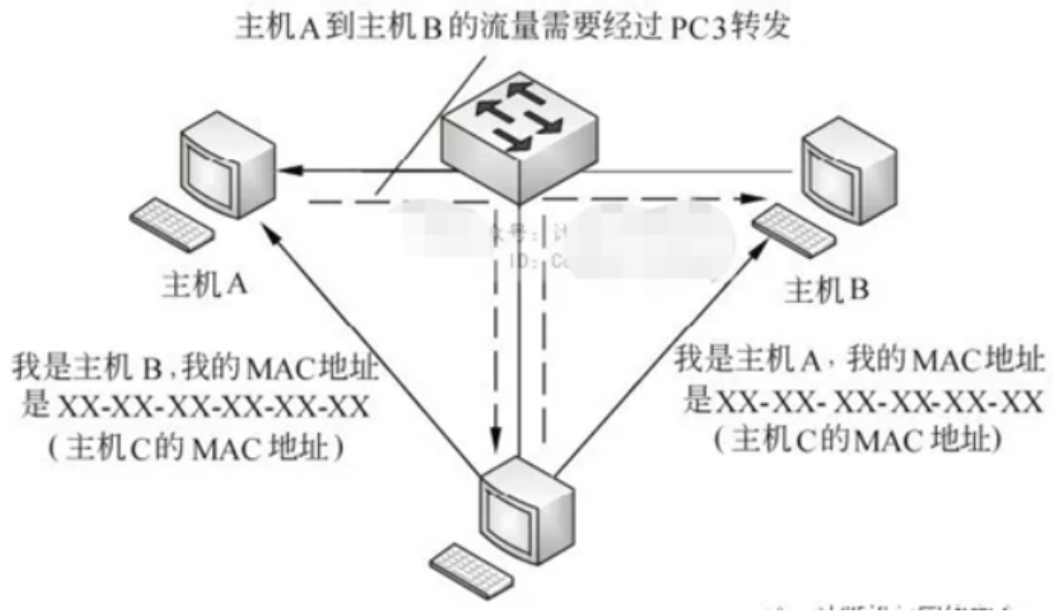
三、抓包场景

抓包的场景其实有很多种，我简单说几个：

单机抓包：直接抓取本机网卡的进出流量数据包

局域网抓包：通过端口镜像技术在交换机上设置端口流量复制转发到安装了Wireshark的主机网卡上，来进行整个局域网的数据包的抓取分析。因为现在的交换机基本都具备MAC表，可以做到MAC地址和交换机端口的对应关系记录，这样的话就很难接收到其他主机的通讯数据包了。

黑客ARP欺骗抓包：就是通过一些ARP攻击软件或者技术手法对宿主机进行ARP欺骗攻击，将其他主机的流量数据欺骗到宿主机的网卡上来，进行局域网的各个主机的数据包抓取。



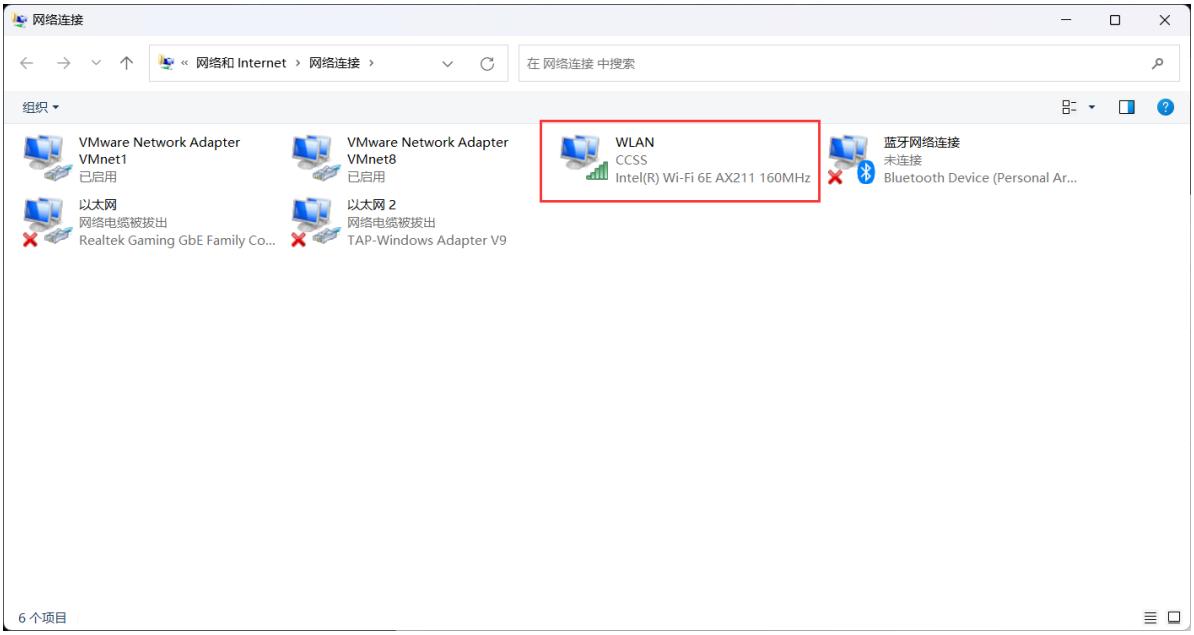
四、界面介绍

4.1 初始界面介绍

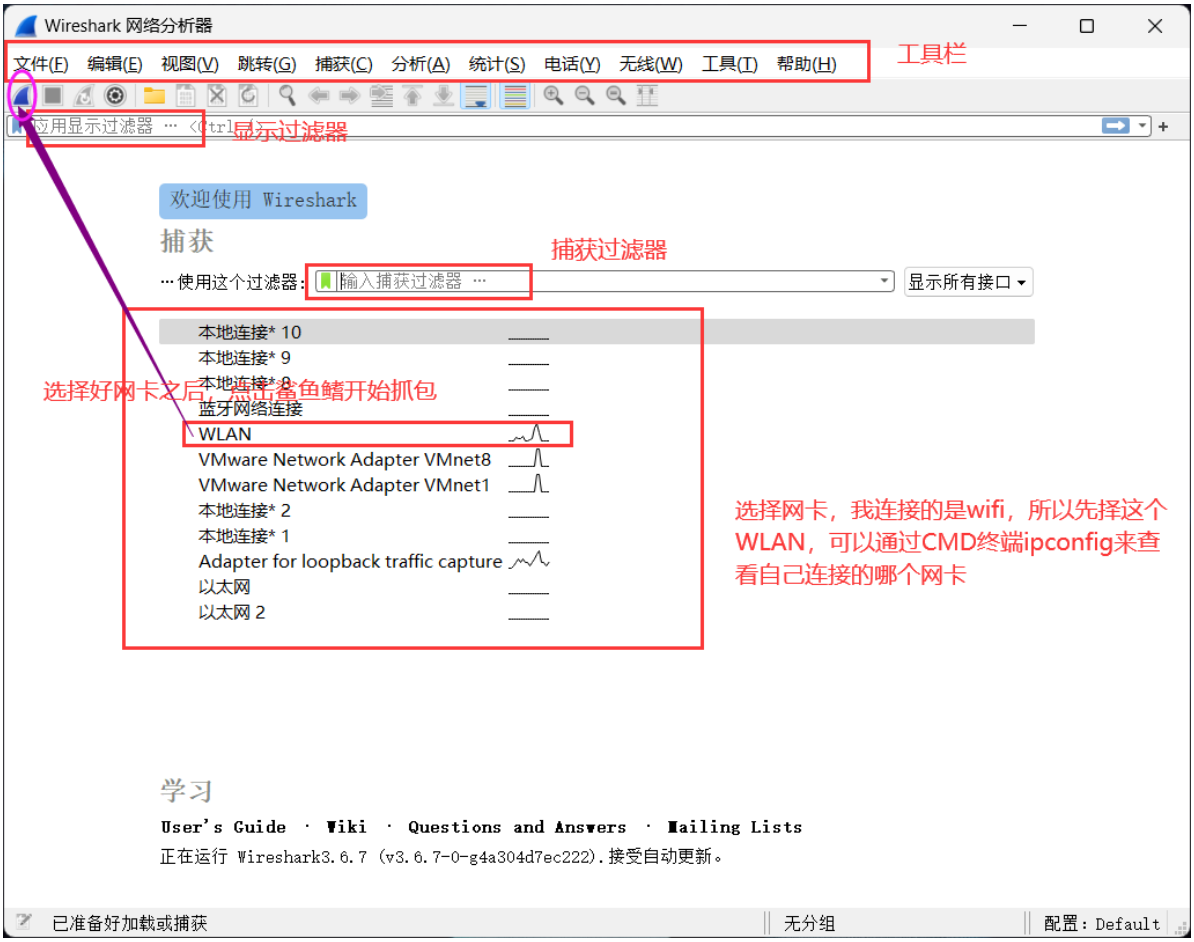
首先查看自己目前是用哪块网卡上网的，win键+r键 -- > 运行窗口-->输入cmd，打开终端窗口，输入`ipconfig`进行查看。

```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . : Home
   IPv4 地址 . . . . . . . . . . . . : 192.168.2.110
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.2.1

以太网适配器 蓝牙网络连接:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

以太网适配器 以太网:

   媒体状态 . . . . . . . . . . . . : 媒体已断开连接
   连接特定的 DNS 后缀 . . . . . . . :

C:\Users\ls198>
```

或者在网络适配器的位置查看



选择网卡界面：



4.2 功能界面介绍

分三个窗口

1号窗口：显示所有进出数据包，也叫做数据包列表

time：数据包进出的当时时间

Source：源ip地址

Destination：目标ip地址

Protocol：协议

length：数据包长度

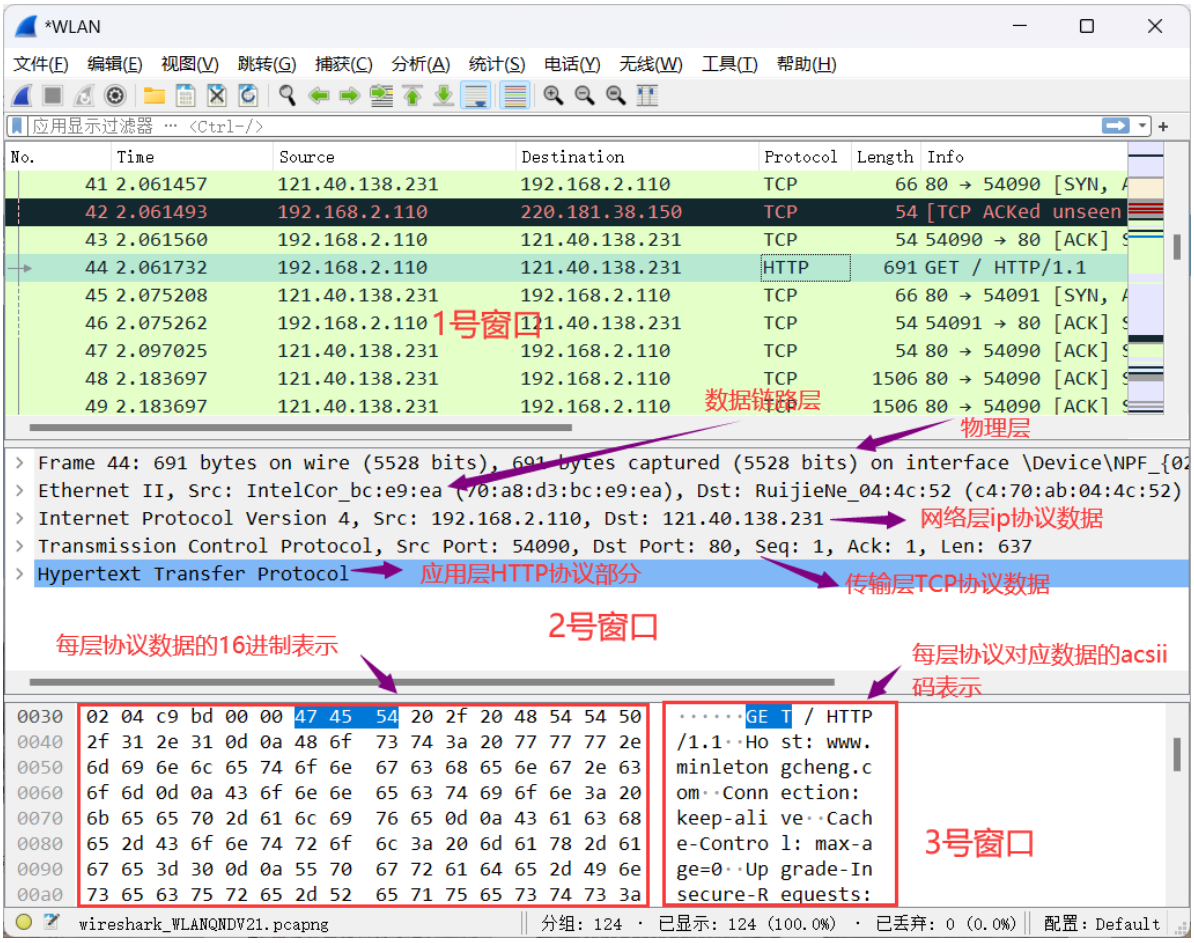
info：数据包的一个简要描述，不能看到具体数据

对列可以进行增加、修改、删除、隐藏等操作，但是默认的这几列就够用

2号窗口：数据包详情，数据包的各层协议的详细数据，1号窗口每点击一个数据包，那么2号和3号窗口就显示这个数据包的详情信息。

3号窗口：数据包对应的16进制表示和ascii类型数据显示，随着2号窗口点击不同协议部分，3号窗口对应数据部分会高亮显示。

看图，我抓取的一个HTTP协议的请求，比如访问这个网站：`http://www.minletongcheng.com/`



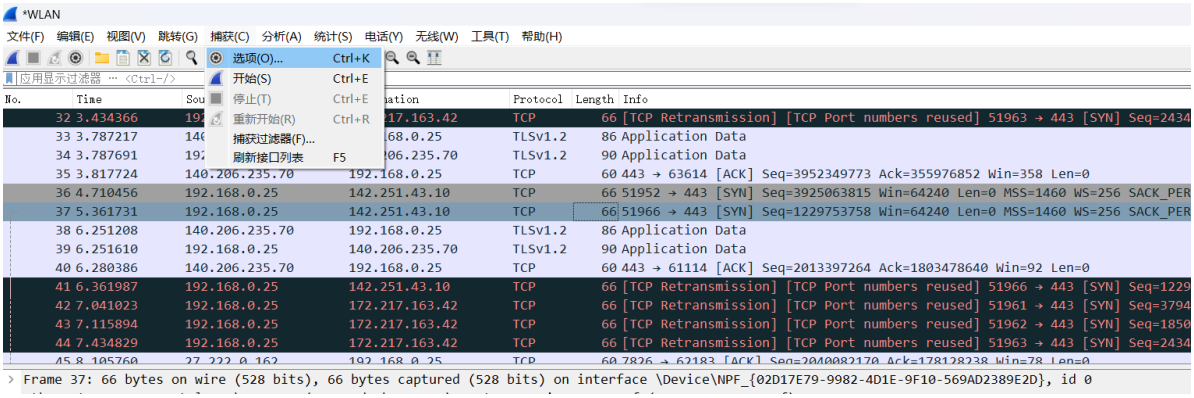
五、混杂模式

混杂模式：接收所有经过网卡的数据包，包括不是发送给主机的包，即不验证MAC地址。

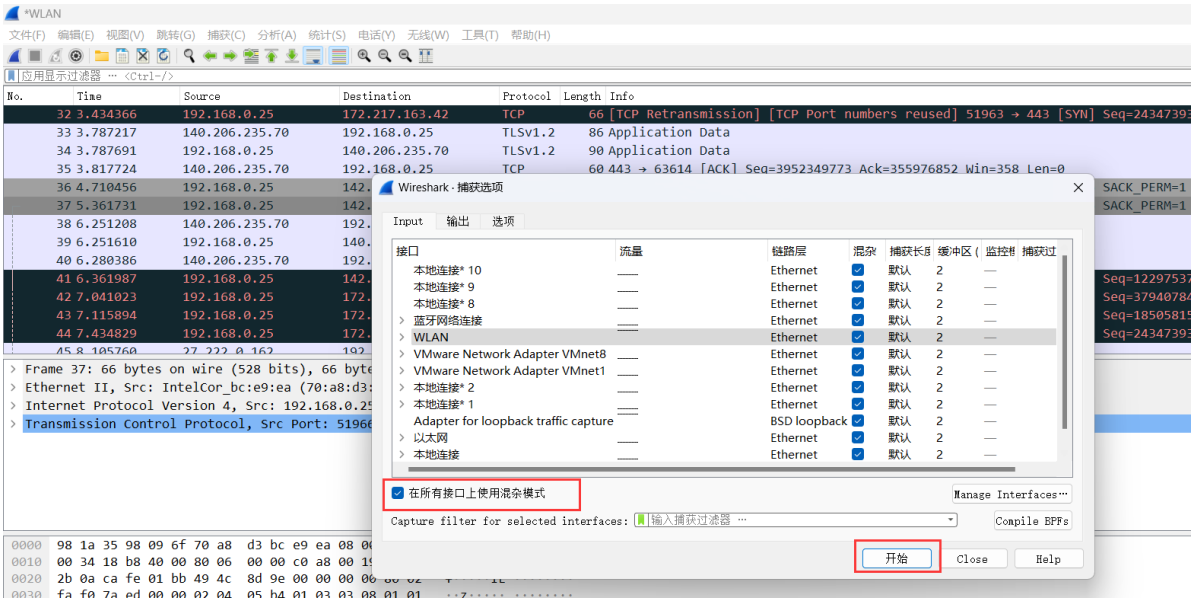
普通模式：网卡只接收发送给本机的包（包括广播包）传递给上层程序，其它包一律丢弃。

通过上面的介绍大家应该就清楚了，我们抓包一定要开启混杂模式，不然你访问其他主机的数据包是抓不到的。并且混杂模式也不会影响到网卡的正常工作，多在网络监听工具上使用，比如wireshark。

开启混杂模式，菜单栏，捕获-->选项



如下：勾选在所有接口上使用混杂模式即可。

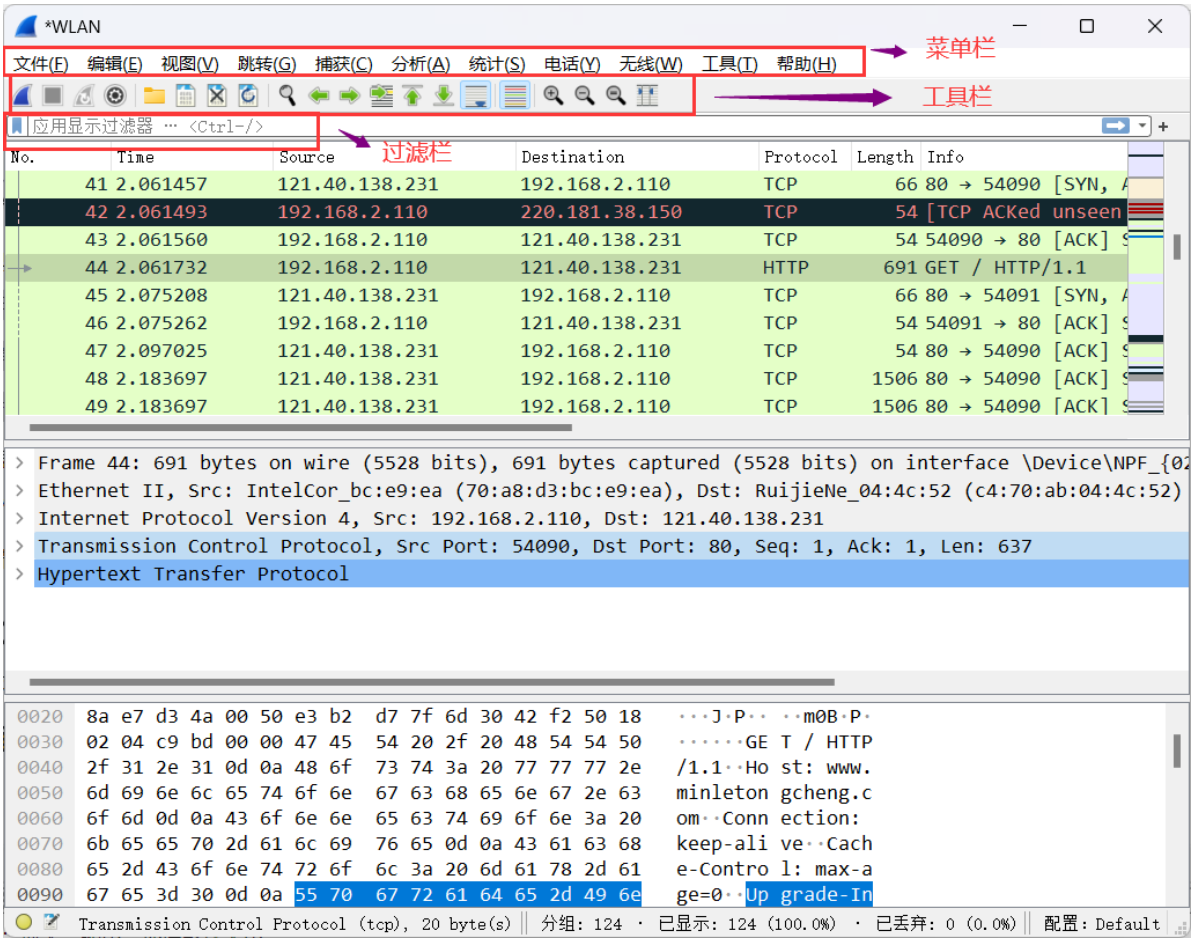


六、基础操作

wireshark在功能界面帮我们提供了三个操作栏：

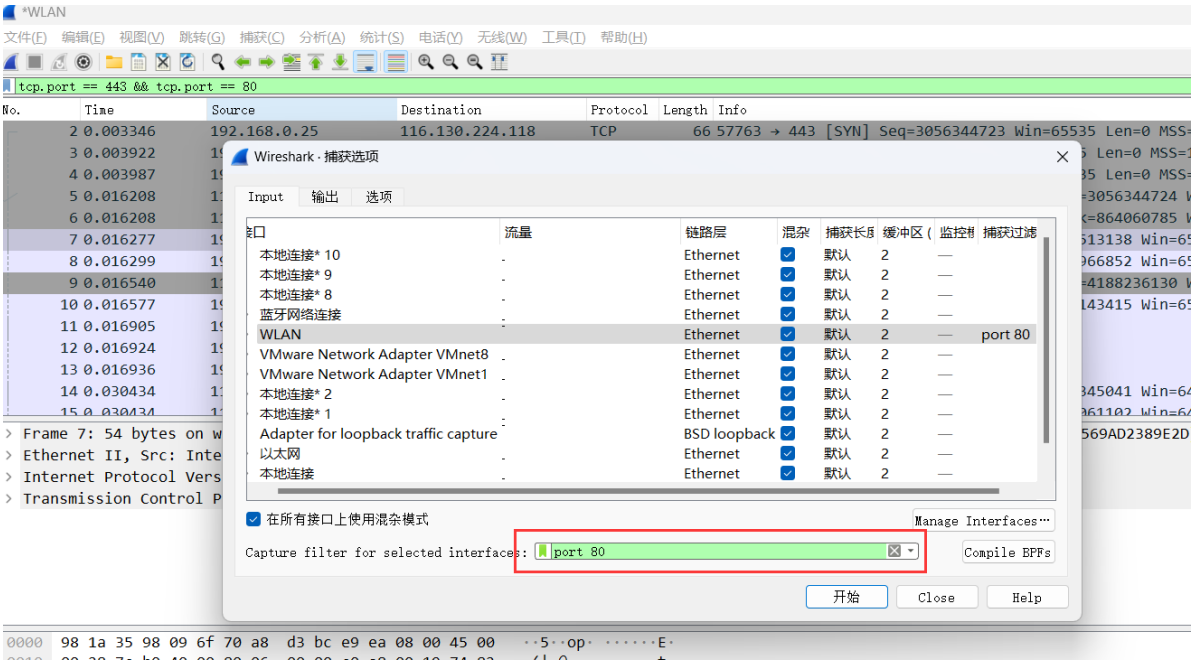
菜单栏：用于调试、配置
工具栏：常用功能的快捷方式
过滤栏：指定过滤条件，过滤数据包，过滤器有两种：抓包过滤器和展示过滤器

如下



捕获过滤器

也叫做抓包过滤器。



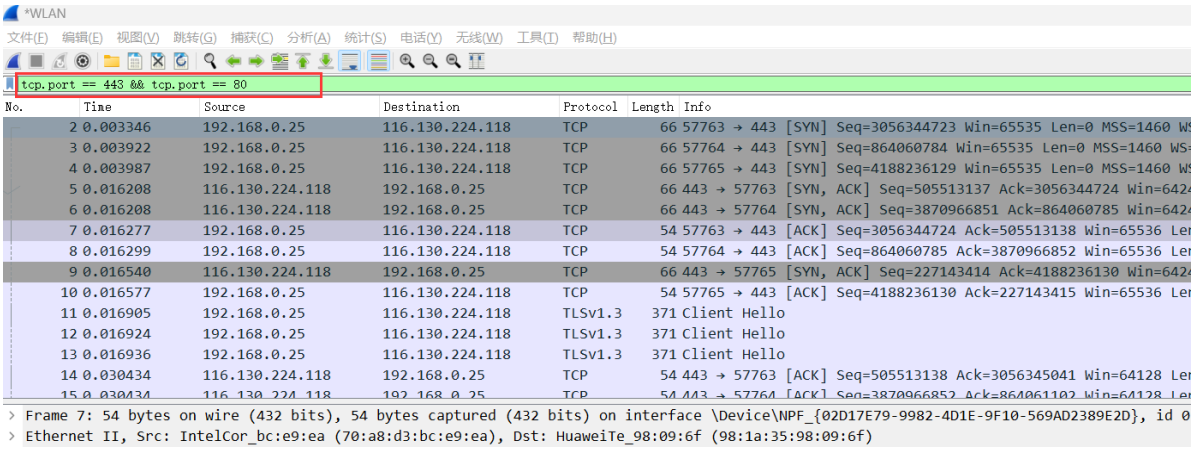
常用过滤指令

如果你自己有很多你想要过滤的条件，百度一下全都有。

```
host、port、src、dst、ip、tcp、http、ftp等
逻辑运算符：||、&&、！    #或与非
举例：
    src host 192.168.2.11 && dst port 80  #抓取源地址为192.168.2.11，并且目的端口为80的流量
    host 192.168.2.11 || host 192.168.2.22 #抓取192.168.2.11或者192.168.2.22的流量数据
    !broadcast   #不抓取广播包
    host ip地址  #通过ip地址过滤
    ether host mac地址 #通过mac地址过滤  比如，ether host 98:1a:35:98:09:6f
    src ether host 98:1a:35:98:09:6f  #过滤出源mac地址为98:1a:35:98:09:6f的数据包
```

显示过滤器

对已经抓取到的数据包进行过滤，查找自己想看的数据包。

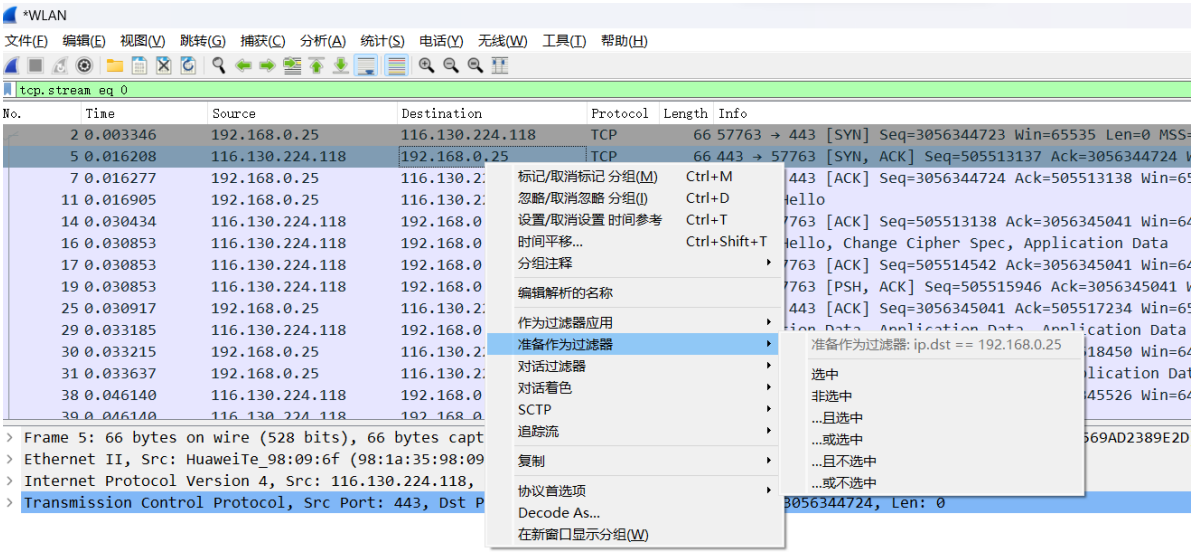


常用过滤指令

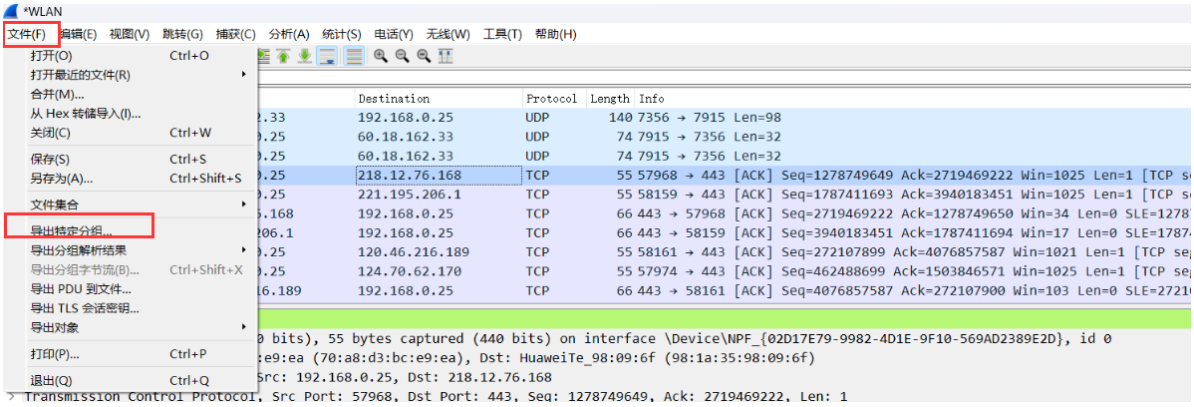
如果你自己有很多你想要过滤的条件，百度一下全都有。

```
http、tcp  #按照协议搜索
ip.src_host=192.168.2.16  #src_host按照源ip地址进行搜索
ip.src_host=192.168.2.16 or ip.dst_host=192.168.2.1 # or是或者的关系，dst_host是目标主机ip地址

tcp.flags.ack == 0 and tcp.flags.syn == 1  # and是并且的关系，这是过滤出，你发送的tcp连接请求三次握手建立连接的交互数据包中，ack标记为0，syn标记为1，这是第一次给目标主机发送的请求建立连接的第一个握手包的标识。
ip.addr=192.168.2.33  # 是要ip地址为192.168.2.33的全部展示出来，不管是目标地址还是源地址。
tcp.srcport == 443  # 源端口为443的数据包
not arp  # 不获取arp数据
tcp.port == 443 # 过滤端口443的数据包
```

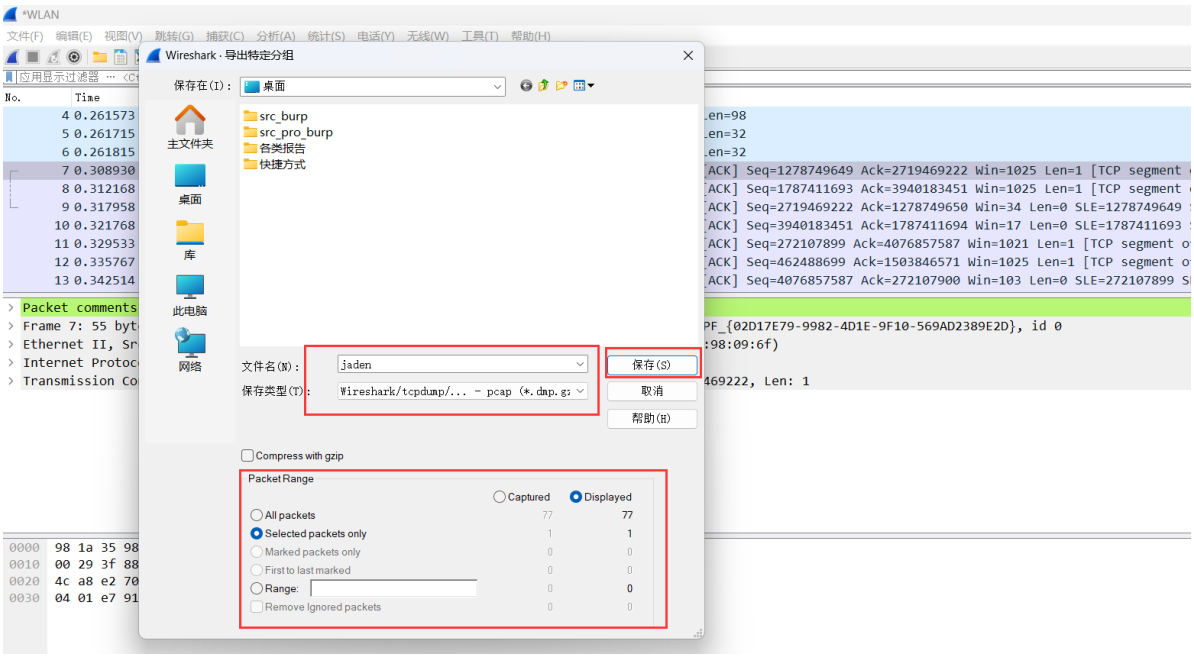
右键选中过滤器



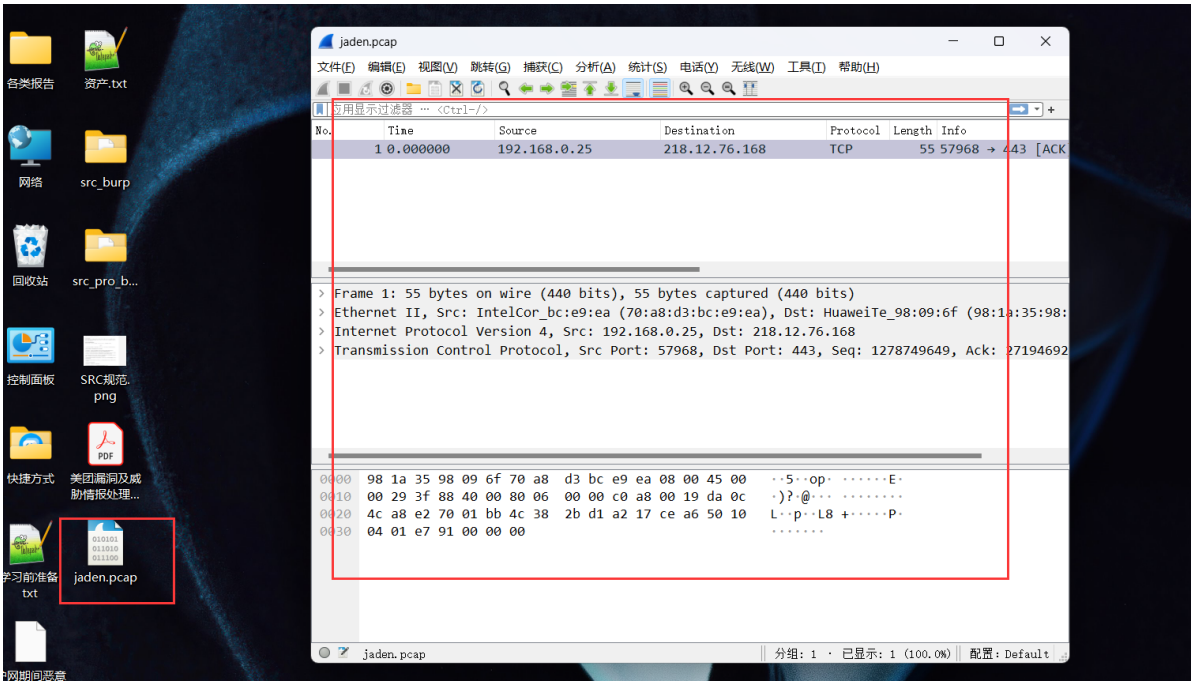
七、保存pcap数据包并打开分析



如下



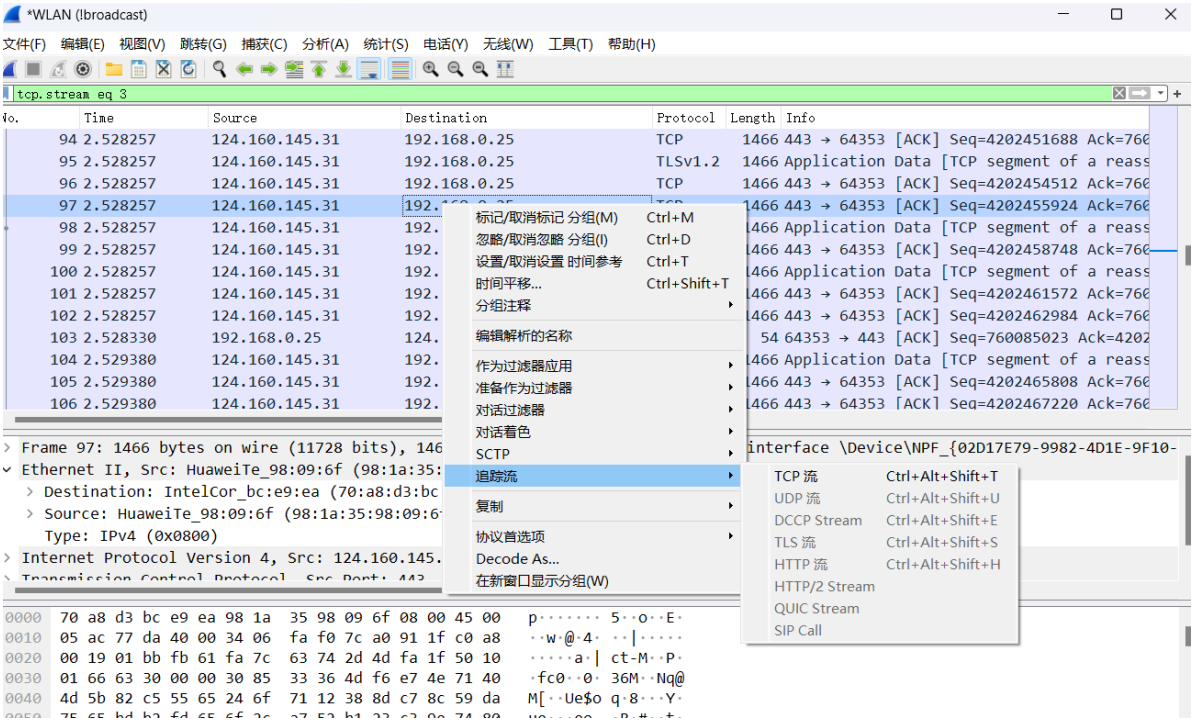
如下



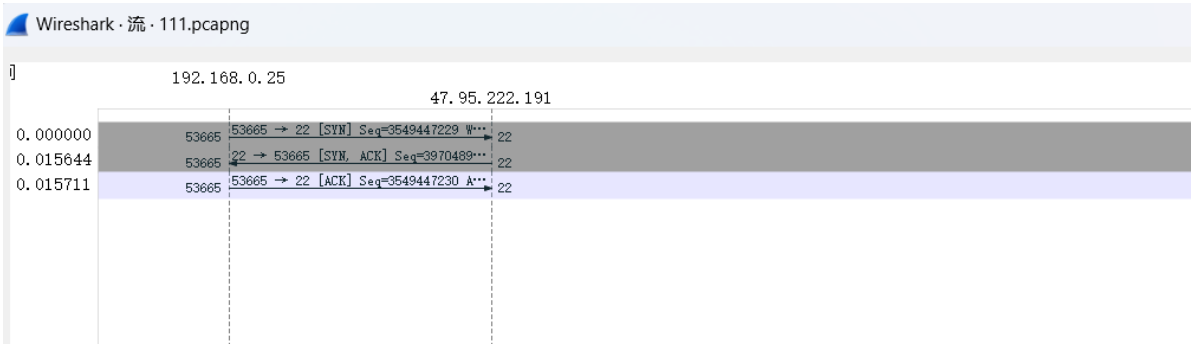
八、高级功能

8.1 追踪流

追踪流这个功能其实就是将多个数据包以连贯的方式呈现出来。



8.2 流量图

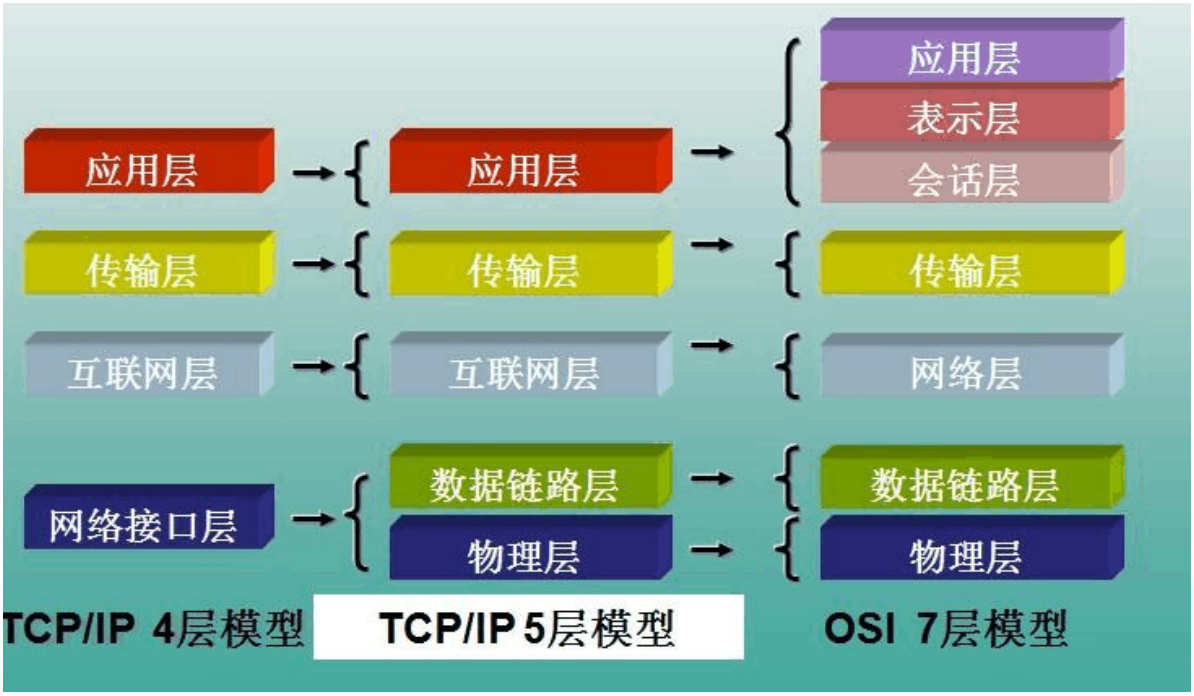


九、协议数据包详解

常见协议包：

- ARP协议
- ICMP协议
- TCP协议
- UDP协议
- DNS协议
- HTTP协议

OSI七层模型



各层协议数据包

物理层

```
Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{02D17E79-9982-4D1E-9F10-569AD2389
  Interface id: 0 (\Device\NPF_{02D17E79-9982-4D1E-9F10-569AD2389E2D})
  Encapsulation type: Ethernet (1)      以太网协议 --- 封装类型
  Arrival Time: Dec  6, 2023 11:49:56.299047000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1701834596.299047000 seconds
  [Time delta from previous captured frame: 0.000067000 seconds]
  [Time delta from previous displayed frame: 0.000067000 seconds]
  [Time since reference or first frame: 0.015711000 seconds]
  Frame Number: 3       数据包列表中的第几个数据包
  Frame Length: 54 bytes (432 bits)
  Capture Length: 54 bytes (432 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]     上层协议
  [Coloring Rule Name: TCP]
  [Coloring Rule String: tcp]
```

时间

数据长度

着色规则

数据链路层

```
Ethernet II, Src: IntelCor_bc:e9:ea (70:a8:d3:bc:e9:ea), Dst: HuaweiTe_98:09:6f (98:1a:35:98:09:6f)
  Destination: HuaweiTe_98:09:6f (98:1a:35:98:09:6f)
  Source: IntelCor_bc:e9:ea (70:a8:d3:bc:e9:ea)
  Type: IPv4 (0x0800)
```

目标mac地址，mac地址前三位是厂商编号，wireshark可以自行识别一些厂商并提示

上层协议

源mac

网络层

IP协议

```
Internet Protocol Version 4, Src: 192.168.0.25, Dst: 47.95.222.191
  0100 .... = Version: 4     ip协议版本 ipv4
  .... 0101 = Header Length: 20 bytes (5)     IP协议数据包头部数据长度
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0x2a3f (10815)
  Flags: 0x40, Don't fragment     表示数据包不分片
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128     TTL 生存周期，没经过一个网络设备就会自动减1
  Protocol: TCP (6)     上层协议
  Header Checksum: 0x0000 [validation disabled]     头部校验和
  [Header checksum status: Unverified]     校验和之后的状态，wireshark不进行校验
  Source Address: 192.168.0.25
  Destination Address: 47.95.222.191
```

差分服务字段

对应的IP报文

ARP协议

windows上通过`arp -d`清空一下主机上的arp缓存表，就会触发arp请求

```
C:\Windows\System32>arp -d

C:\Windows\System32>
```

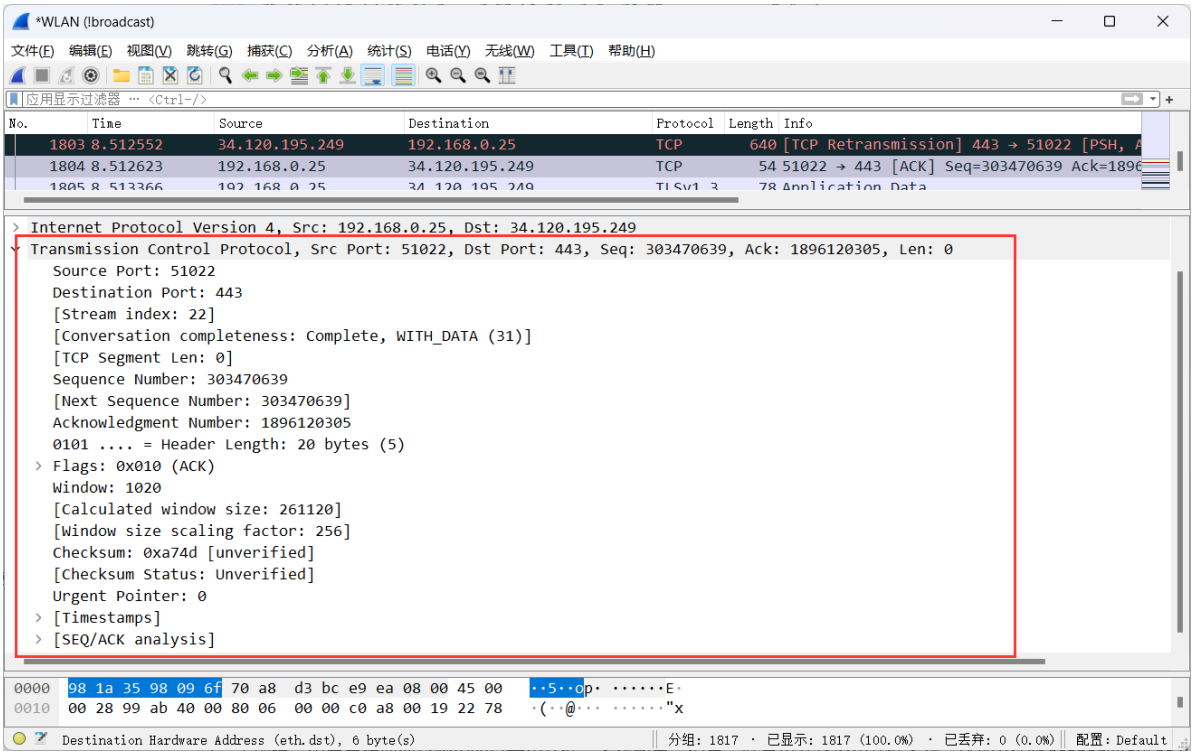
抓包：

```
> Frame 2: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF_{02D17E79-9982-4D1E-9F10-569AD2389E2D}, id 0
> Ethernet II, Src: HuaweiTe_98:09:6f (98:1a:35:98:09:6f), Dst: IntelCor_bc:e9:ea (70:a8:d3:bc:e9:ea)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)   硬件类型
    Protocol type: IPv4 (0x0800)        网络协议类型
    Hardware size: 6  硬件地址长度，mac地址
    Protocol size: 4  网络协议地址长度，ip地址
    Opcode: reply (2)  Opcode标识操作码，reply(2)表示响应，request(1)表示请求
    Sender MAC address: HuaweiTe_98:09:6f (98:1a:35:98:09:6f)
    Sender IP address: 192.168.1.1
    Target MAC address: IntelCor_bc:e9:ea (70:a8:d3:bc:e9:ea)
    Target IP address: 192.168.0.25
```

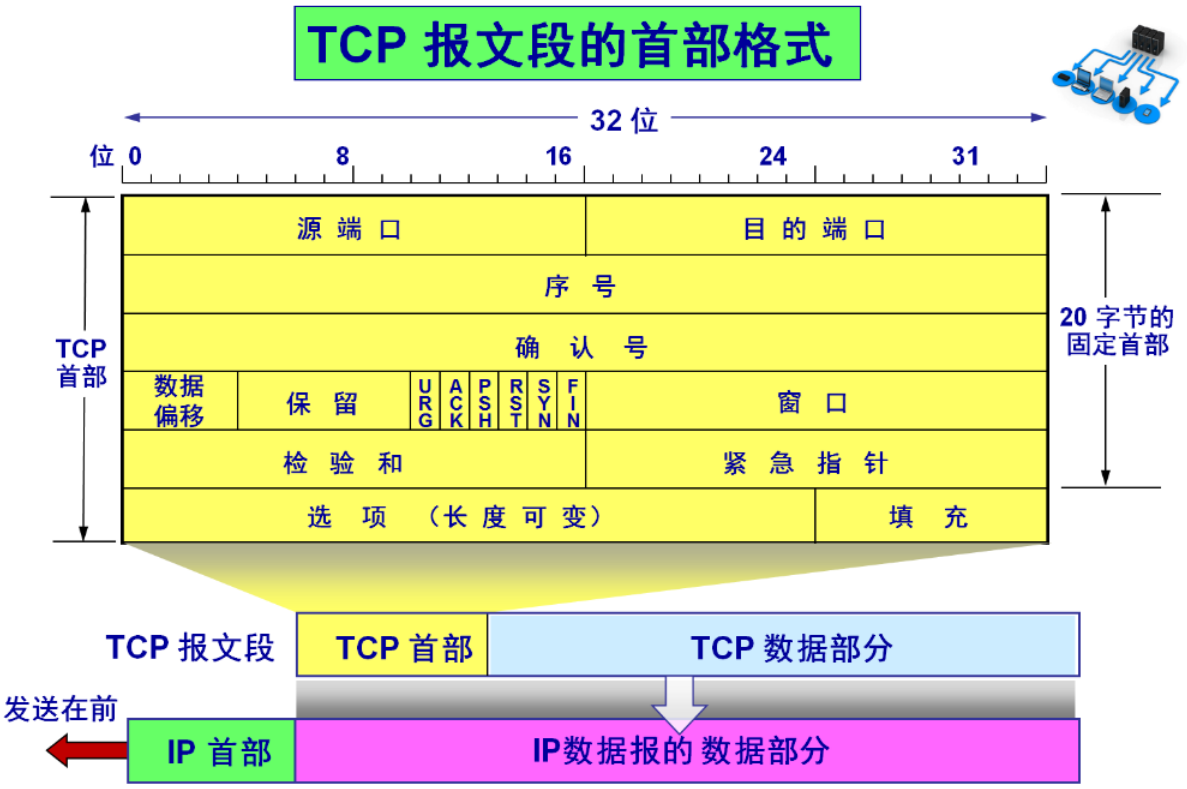
传输层

TCP报文格式解析

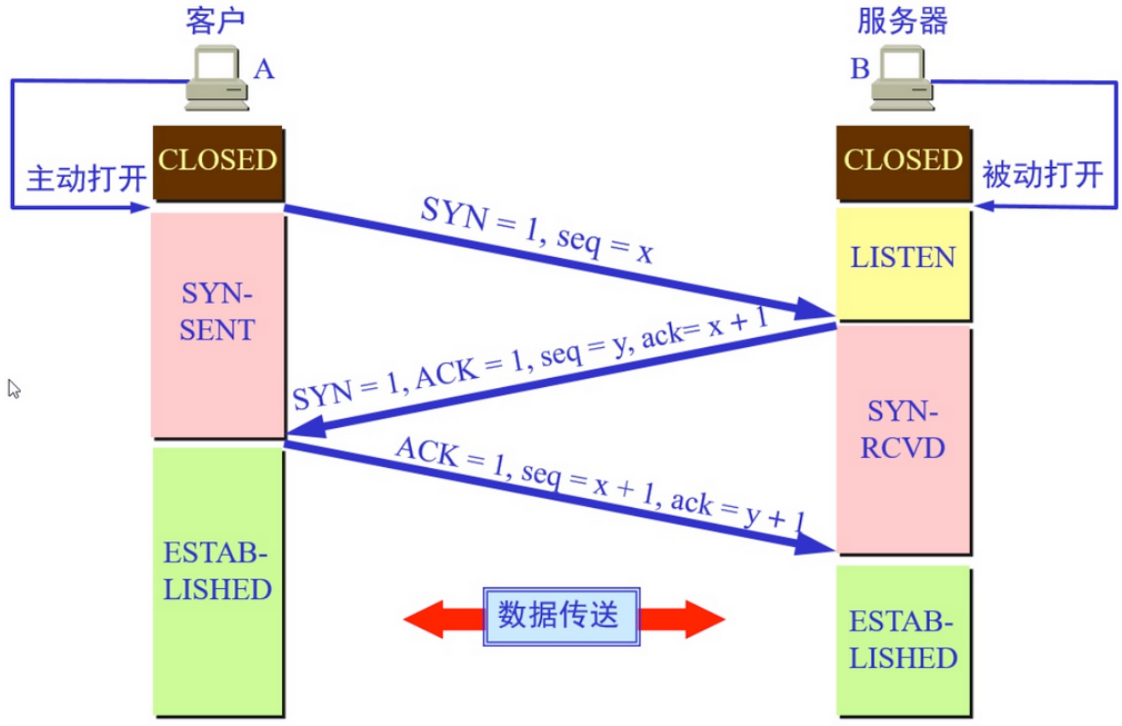
随便抓一个tcp的数据包，如下



整体格式：



TCP三次握手



下面我们分析 TCP 报文段的结构和各个字段的含义

```
Source Port: 源端口号
Destination Port: 目标端口号
Stream index：组序号，用来表示多个数据包是否属于同一组的，可以理解为Stream index值相同的是同一组数据包，是和同一个目标通信的数据包。可以用来进行目标请求数据包的筛选。
```

[Conversation completeness: Incomplete, ESTABLISHED (7)] 会话完整性的标记，是通过不同的状态值计算出来的结果，所以显示结果不同，上面的数字7是1 (SYN) + 2 (SYN/ACK) + 4 (ACK) = 7。表示一个仅包含标准 TCP 三次握手的会话。

```
1 : SYN
2 : SYN-ACK
4 : ACK
8 : DATA
16 : FIN
32 : RST
```

TCP Segment：当传输数据较大时，一个完整消息会被分割成多个请求包，每个请求包中包含的数据大小就是TCP Segment的值。Wireshark 为了能标识出哪些 TCP Segment 需要被重新组装(Reassembled)，会将除了最后一个 Segment 之外的其他 Segment 都打上「TCP segment of a reassembled PDU」标记。

Sequence Number:序列号(seq的值)，表示本次传输数据的起始字节在整个数据流中的位置，用于数据的重组和接收方确认使用

[Next Sequence Number: 3970489943] # 表示下一个序列化，是Sequence Number+1，但是如果Sequence Number已经是最后一个值了，那么他俩的值会是相同的。

Acknowledgment Number: 确认序号(ack的值)，值为期望收到下一包的序号，用于确认已经收到数据的偏移序号；

0101 .... = Header Length: 20 bytes (5) # 0101是数据偏移，占4位，....保留位，Header Length体现的是tcp报头长度

Flages:标识符

ACK–为1表示确认号字段有效

PSH–为1表示是带有PUSH标志的数据，指示接收方应该尽快将这个报文段交给应用层而不用等待缓冲区装满。

RST–为1表示出现严重差错。可能需要重新创建TCP连接。还可以用于拒绝非法的报文段和拒绝连接请求。

SYN–为1表示这是连接请求或是连接接受请求，用于创建连接和使顺序号同步

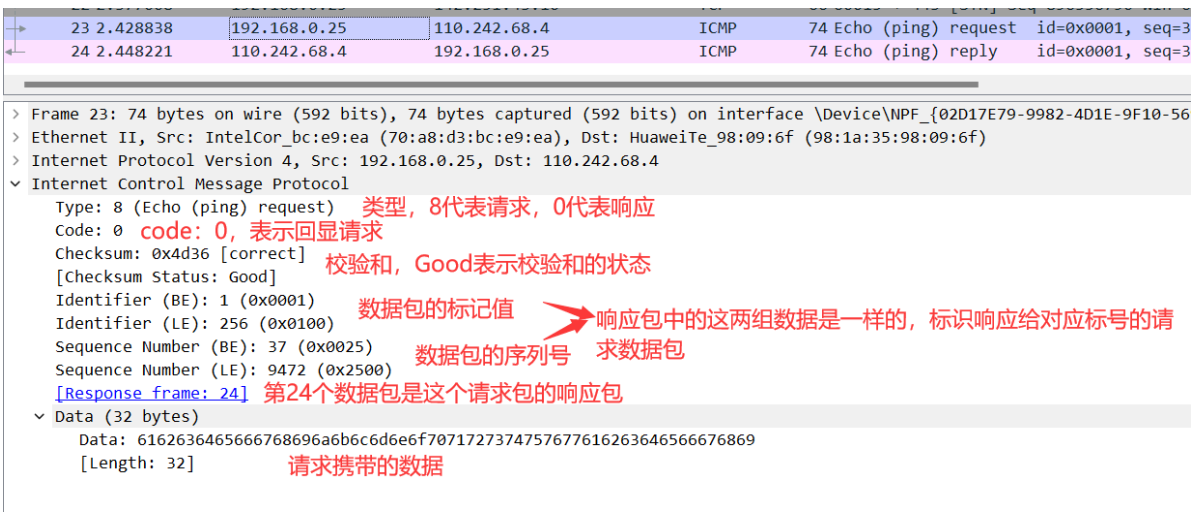
FIN–为1表示发送方没有数据要传输了，要求释放连接。

Window: 窗口，表示从确认号开始，本报文的发送方可以接收的字节数，即接收窗口大小。用于流量控制。

Checksum: # 校验和，每个 TCP 包首部中都有两字节用来表示校验和，防止在传输过程中有损坏。如果收到一个校验和有差错的报文，TCP 不会发送任何确认直接丢弃它，等待发送端重传

Urgent Pointer：紧急指针，紧急指针仅在标记位中的紧急位URG= 1时,Urgent Pointer值才有意义

ICMP报文格式解析

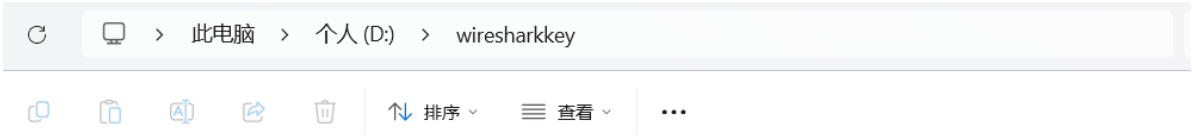


十、HTTPS数据解密分析

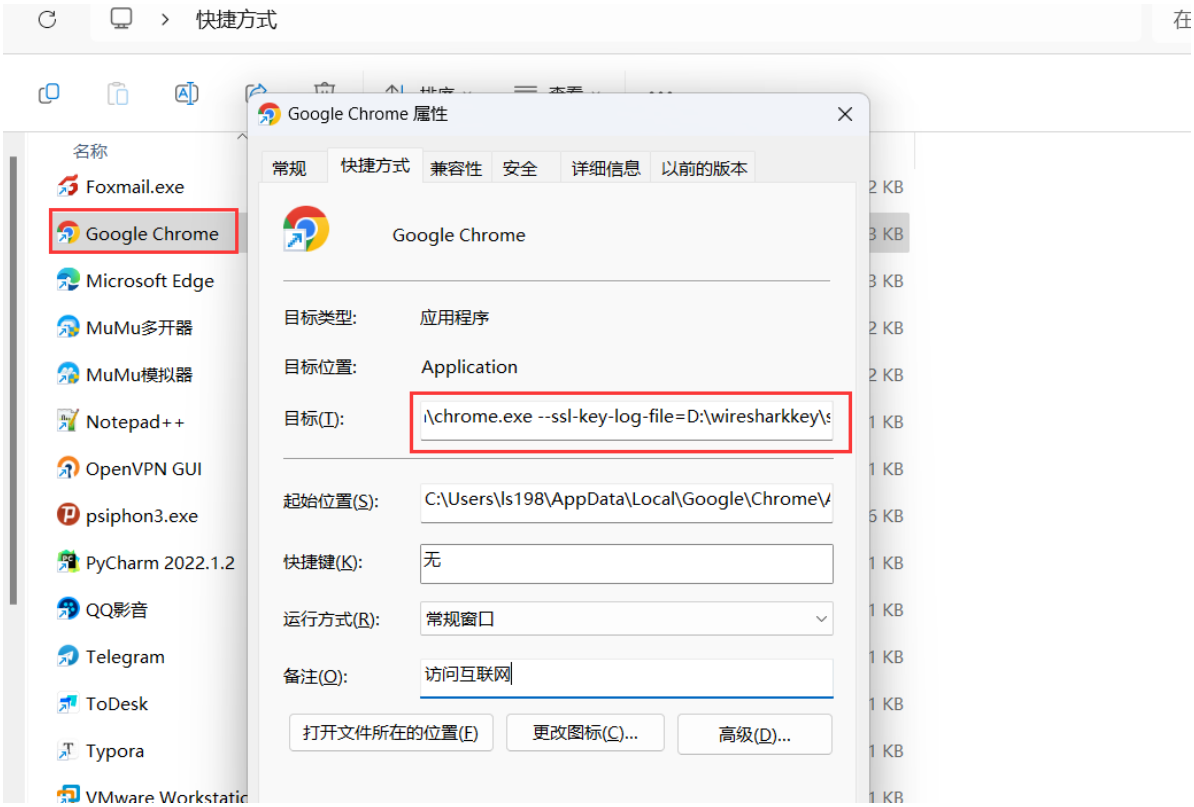
方式1、动态获取对称加密秘钥

以下是针对Chrome浏览器的配置：

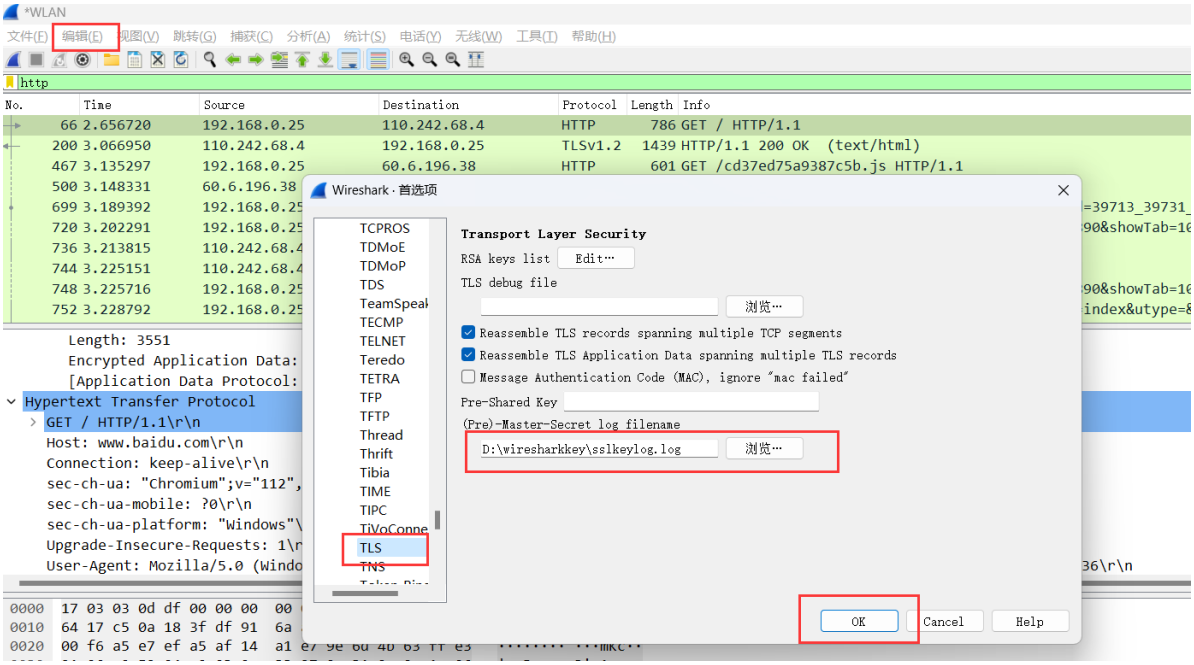
1、先创建一个自己想要保存密钥的目录和文件，名称随意，但是最好是英文路径和文件名称，比如`D:\wiresharkkey\sslkeylog.log`



2、先找到Chrome的快捷方式，如下，快捷方式 -->鼠标右键-->属性-->目标，设置如下，目标启动路径后面添加`--ssl-key-log-file=D:\wiresharkkey\sslkeylog.log`，我的设置之后是这个值`C:\Users\ls198\AppData\Local\Google\Chrome\Application\chrome.exe --ssl-key-log-file=D:\wiresharkkey\sslkeylog.log`，因为每个人的chrome.exe的安装目录不同，所以按照自己的安装目录路径来看。

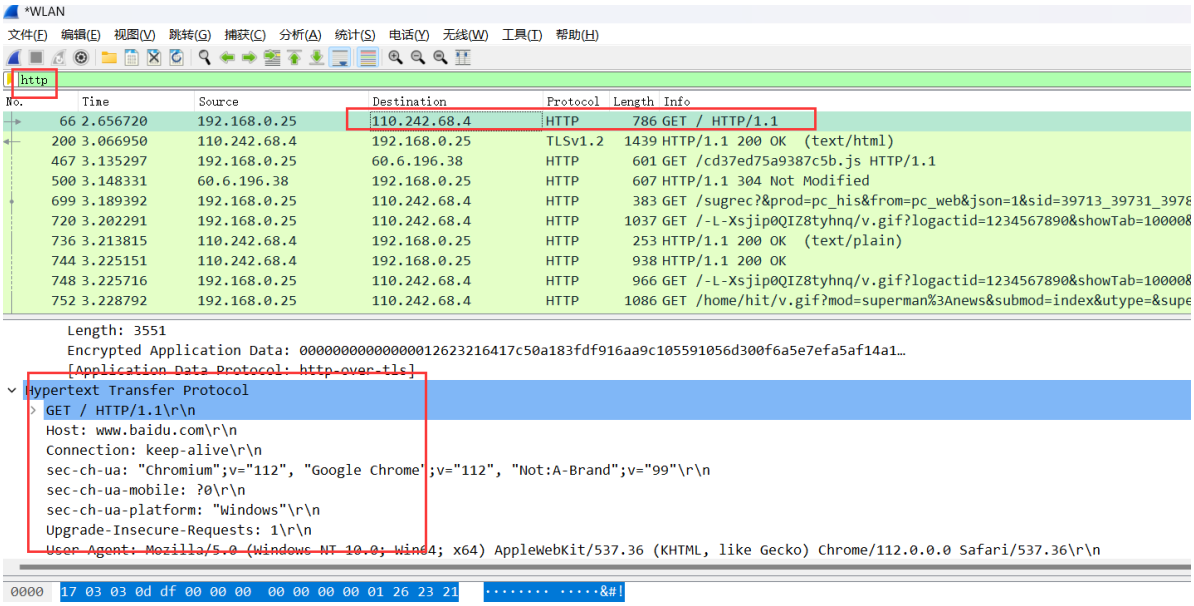


3、设置Wireshark解析http数据用到的动态密钥日志文件路径，如下`编辑--首选项--Protocols--TLS`，然后将我们创建的`sslkeylog.log`文件的路径配置到`(Pre)-Master-Secret log filename`项中即可



4、重新通过快捷方式打开Chrome浏览器，访问某个网站，比如百度。

结果：

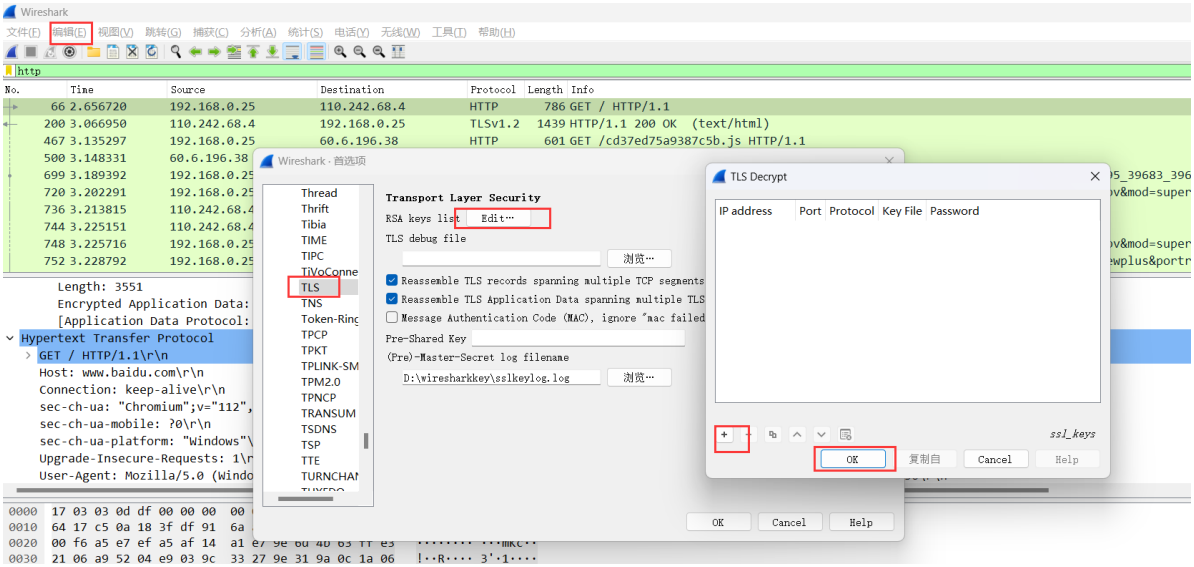


看到了HTTPS解密之后的数据包。

方式2、使用服务器证书方式解密

如果用户可以拿到服务器证书的话，则可以使用服务器证书方式解密HTTPS数据，这种方式不如第一种方式来的方便。

在菜单栏依次选择 编辑--首选项--Protocols--TLS 命令，打开TLS协议设置对话框。单击RSA keys list对应的Edit按钮，将打开TLS解密对话框，如下



该对话框中共包括5列，分别是IP address、Port、Protocol、Key File和Password。

IP address：服务器的IP地址。

Port：HTTPS监听的端口，一般为443。

Protocol：指定协议，一般为HTTP。

Key File：指定从服务器上获取到的RSA Key。这个RSA Key需要是一个解密后的PKCS#8PEM格式的（RSA）Key。

password：一般不填写。

十一、书籍推荐

《Wireshark数据包分析实战》、《Wireshark网络分析》、《TCP/IP协议栈详解》