分析Ping命令的执行过程

实验环境

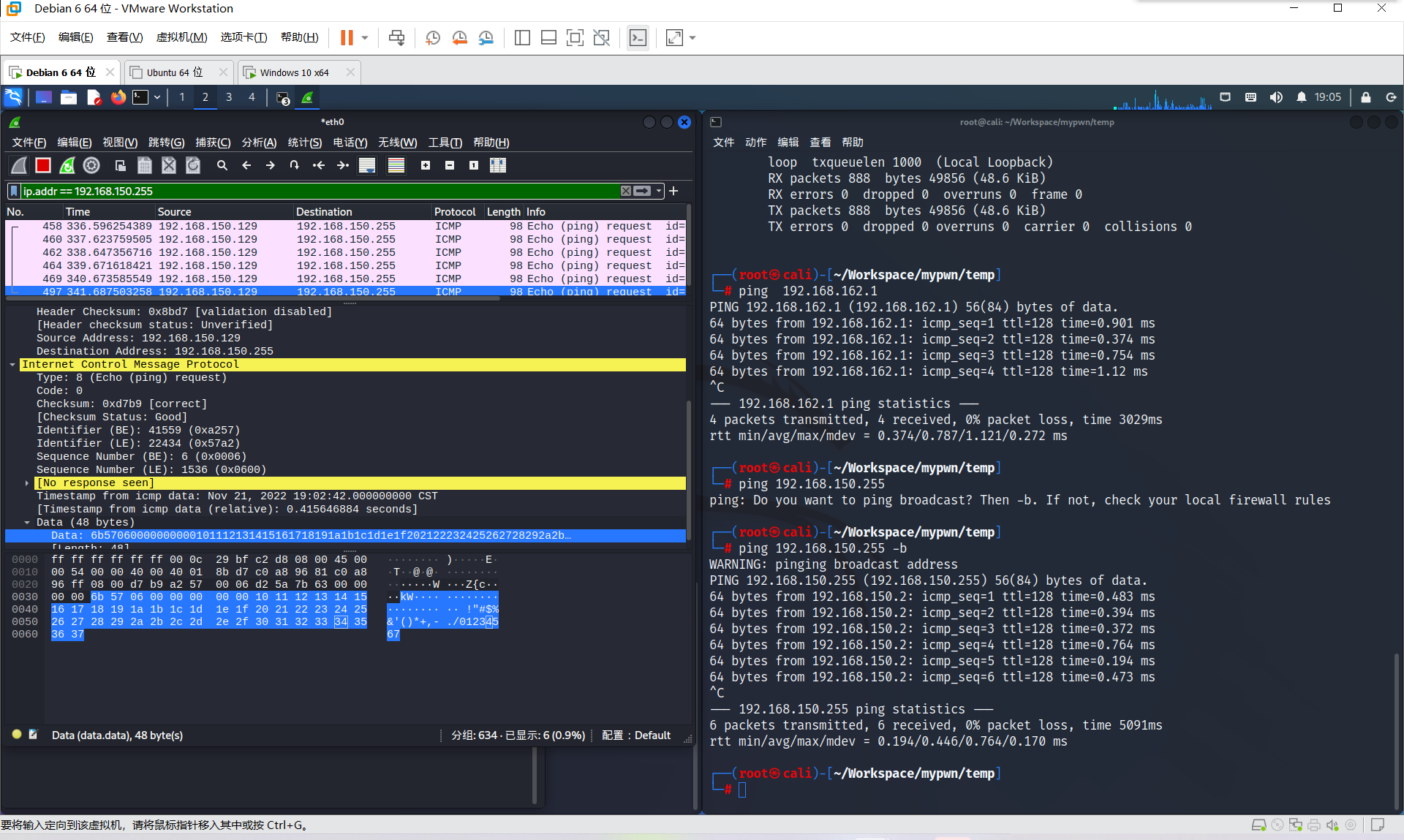
PC1 kali ip:192.168.150.129

PC2 win11 ip:192.168.162.1

默认网关192.168.150.255

使用PC1向网关进行ping，使用wirshark进行抓包

Ping使用ICMP报文



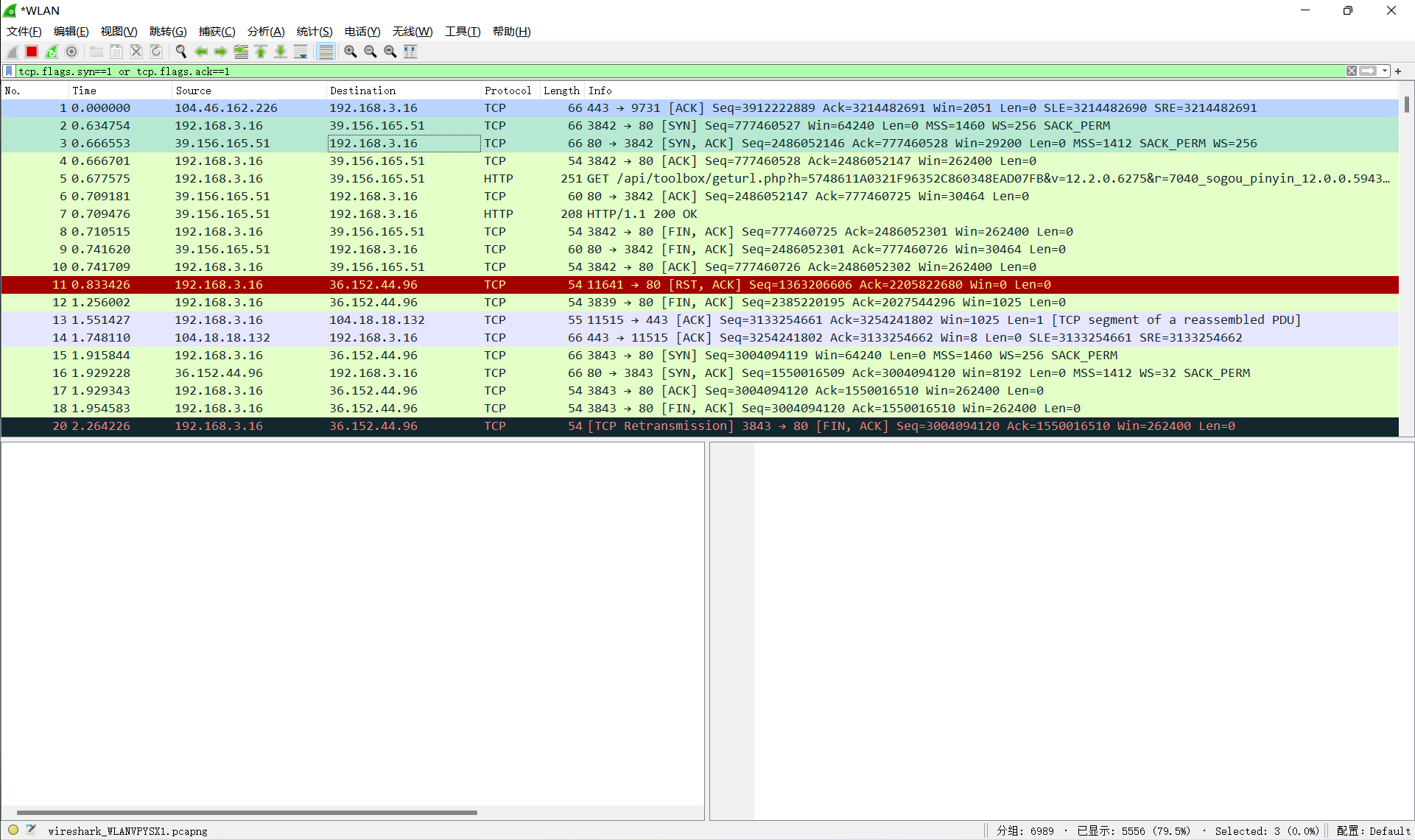
IP 数据报首部信息如下

|  |  |  |
| --- | --- | --- |
| 字段 | 报文信息 | 说明 |
| 版本 | 4 | IP使用版本号 |
| 首部长度 | 20 | IP头的长度 |
| 区分服务 | 0 | 优先级标志位和服务类型标志位，被路由器用来进行流量的优先排序 |
| 总长度 | 84 | IP头与数据包中数据的长度 |
| 标识 | 0 | 一个唯一的标识数字，用来识别一个数据包或者被分片数据包的次序 |
| 标志 | 0x40 | 用来标记一个数据包是否是一组分片数据包的一部分 |
| 片偏移 | 0 | 一个数据包是一个分片，这个域中的值就会被用来将数据包以正确的顺序重新组装 |
| 生存时间 | 64 | 用来定义数据包的生存周期，已经过路由器的跳数/秒数进行描述 |
| 协议 | 1 | 用来识别在数据包序列中上层协议数据包的类型 |
| 首部检验和 | 0x8db7 | 一个错误检测机制，用来确认IP头的内容没有被损坏或者篡改 |
| 源地址 | 192.168.150.129 | 发出数据包的主机的IP地址 |
| 目的地址 | 192.168.150.255 | 数据包目的地址IP地址 |

分析TCP三次握手过程

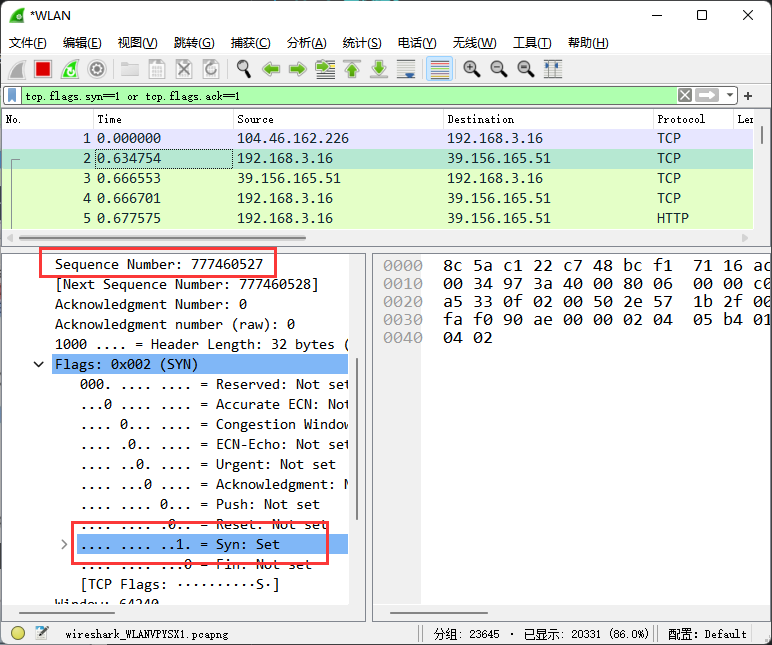
打开Wireshark，选择网络接口捕获流量

在过滤器中输tcp.flags.syn==1过滤出所有SYN包，使用tcp.flags.ack==1过滤出ACK包，可以看到TCP三次握手的过程——客户端发送的 SYN 包，然后服务器回复 SYN-ACK 包，最后客户端回复 ACK 包。且可以看到在三次TCP后出现HTTP报文，说明http是有TCP协议建立连接的。

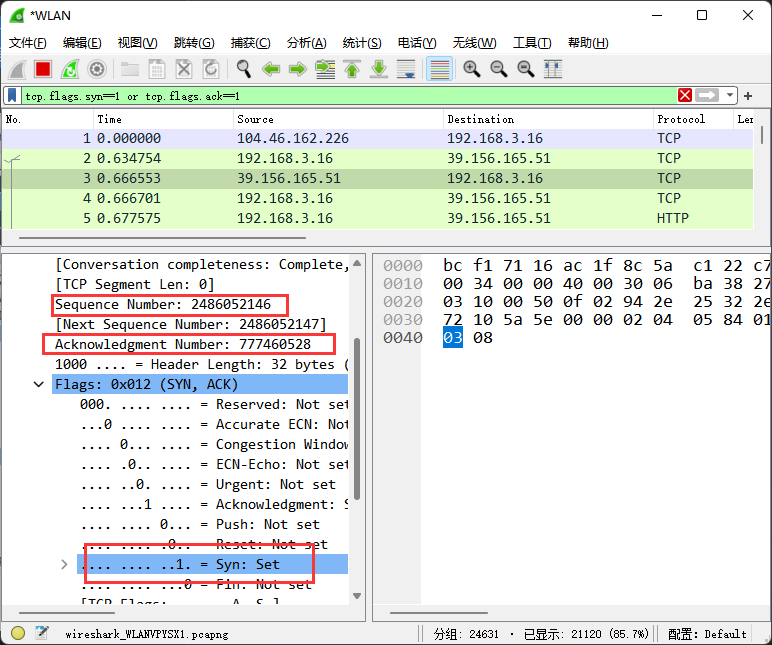


专门观察这三次包，

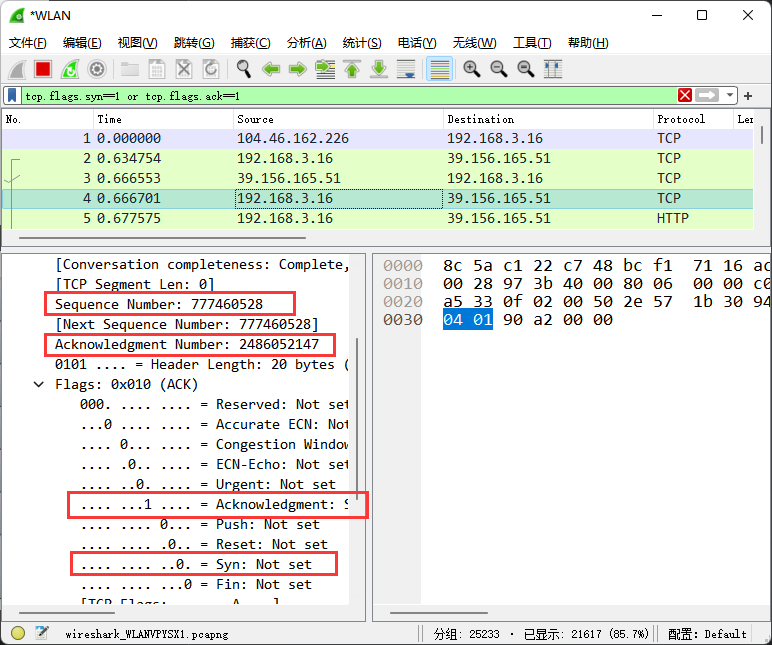
第一次握手，客户端向服务器发送连接请求包，标志位SYN为1，Sequence number为777460527



第二次握手： 服务器收到客户端发过来报文，由SYN=1知道客户端要求建立联机。向客户端发送一个SYN和ACK都置为1的TCP报文，设置初始序号为2486052146，将确认序号(Acknowledgement Number)设置为客户的序列号加1，

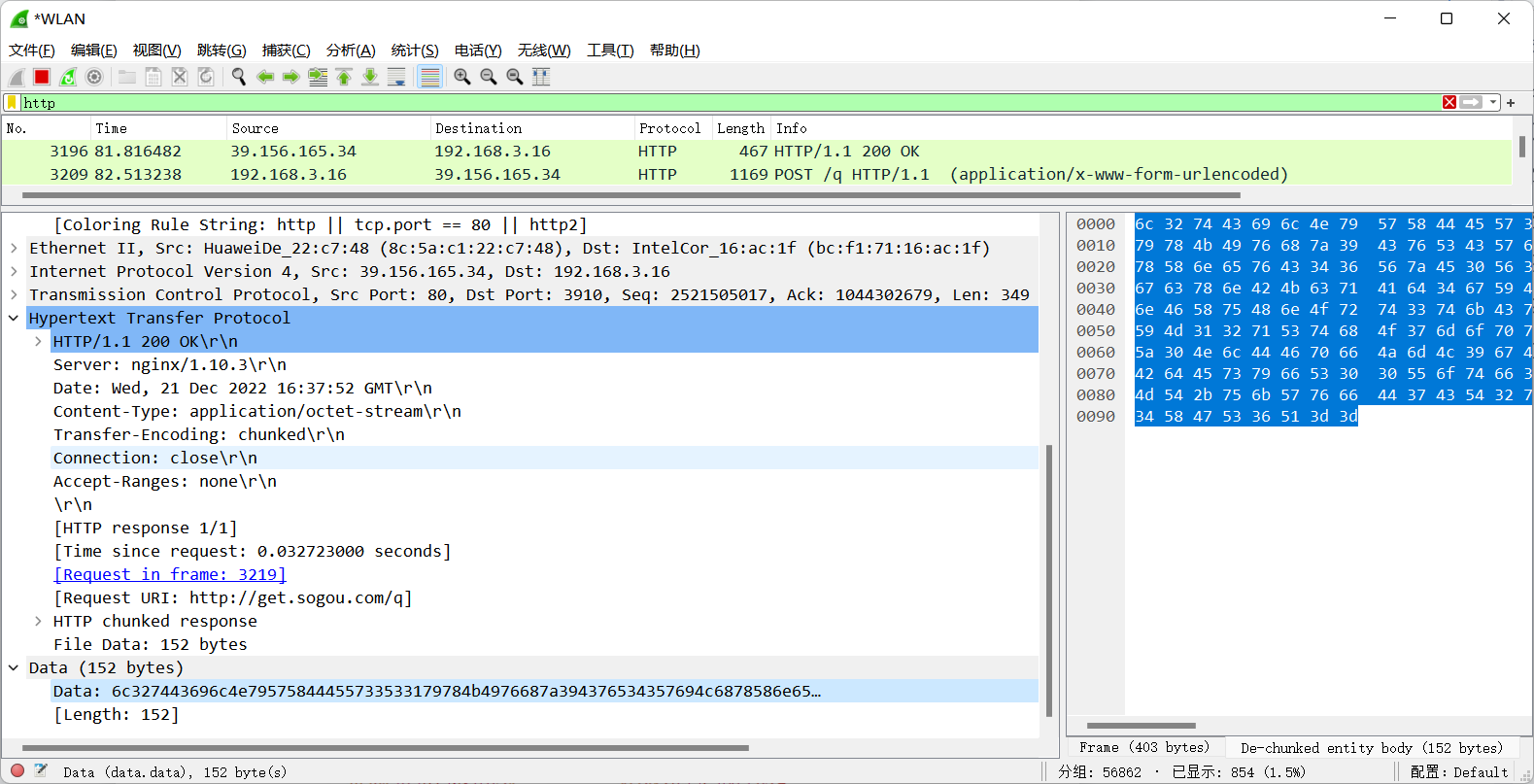


第三次握手： 客户端收到服务器发来的包后检查确认序号(Acknowledgement Number)是否正确，即第一次发送的序号加1。以及标志位ACK是否为1。若正确，客户端再次发送确认包，ACK标志位为1，SYN标志位为0。Acknowledgement Number加一变为2486052147，Sequence number加一变为777460528。服务器收到后确认序号值与ACK=1连接建立成功。



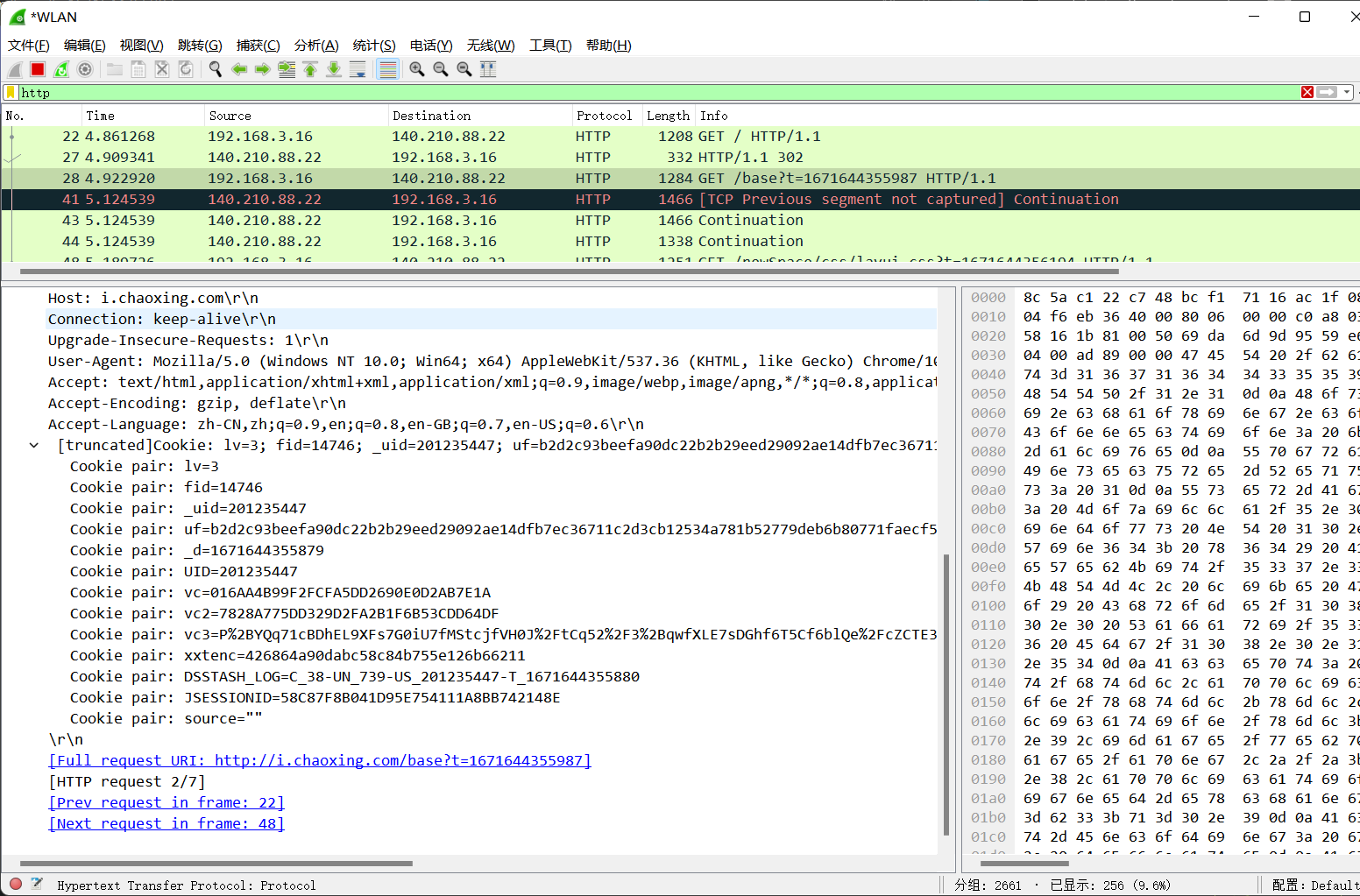
分析http报文长度

选择一个http数据包，查看fileData字段查看长度，说明传输152字节的数据



找出HTTP中的口令字段

打开超星进行登录同时抓包



可以看到有我在学系统中的uid，以及cookie