计网实验笔记(一)

原创

0verWatch 于 2018-07-20 12:03:58 发布 5871 收藏 28

分类专栏： 计算机网络 文章标签： 计算机网络

前言

总算有时间把计网实验的学习过程记录一下了。也当做是一个复习的过程，过程太长，分两篇博客写吧。

正文

这一篇是对协议报文的分析，这里面有对Wireshark的使用，只能说我对于Wireshark这款软件的使用还是渣。各种ip地址的过滤以及如何去利用Wireshark强大的搜索功能这里就不多说了。个人感觉最好的方法就是在CTF里面的流量分析题目里学，在CTF的流量分析题目里面你会经常触碰到那种要导出文件图片，导进秘钥解密https的各种类型题，感觉学的会更快。

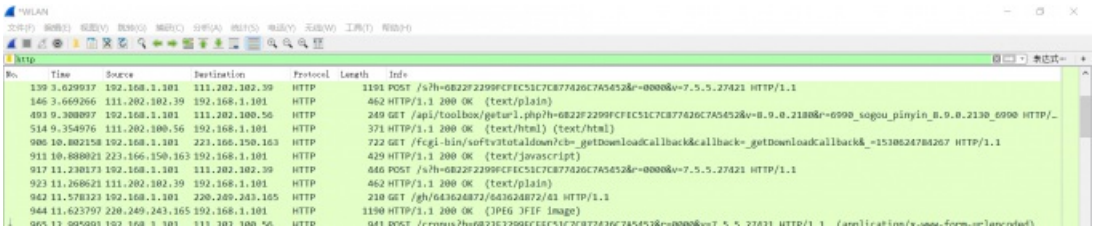
不多说废话下面上课实验的内容

```
（1） 运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；过滤捕获和过滤显示不同条件的数据包。
（2）分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义；要求分别获取WWW服务、Email服务、QQ通信和迅雷文件下载四种不同网络服务过程中的数据包。
（3）运用抓包工具，连续获取面向连接的互联网访问情形下的本机网卡数据包；对连续获取的数据包找到执行面向连接过程的报文，给出实现面向连接过程（TCP三次握手）的详细分析。
```

实验内容第一点

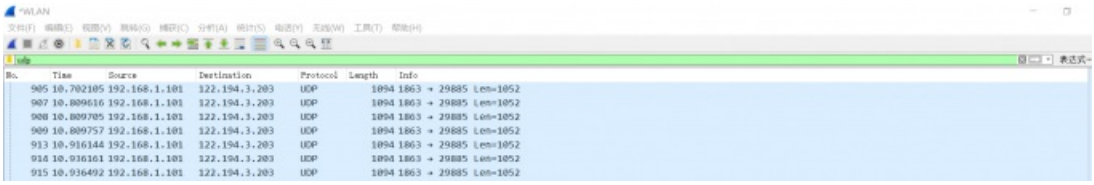
http协议

关于这个协议，个人理解是万维网的服务协议，默认端口是80端口，运输层使用的是可靠的TCP，不过现在也有它的加密版本https，也就是在http的基础上增加了ssl去加密，默认端口是443，所以在我们需要解密含有https的报文的时候，想查看更多的信息，应该追踪http流而不应该追踪tcp，到运输层的时候已经加密了，看不出啥东西的，我记得DDCTF里面有一道流量分析的题目就是这样的，里面可以导出一个图片然后运用的是ocr把秘钥提取出来。确实有点烦，ocr这东西不准。对于Wireshark里面的分组，我们可以直接过滤http就可以获得啦，前提是记得访问http的页面，别访问https的页面，233333，不然你只能顾虑的是https了。



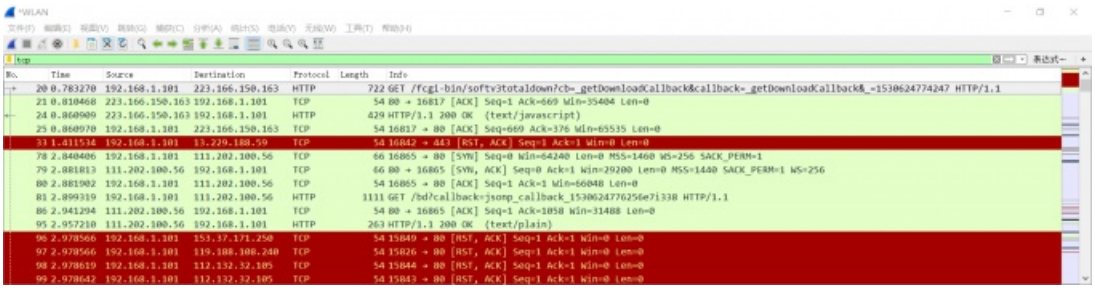
UDP协议

UDP协议是面向报文的一种协议，如果你观察他的封装，你会发现客户给他什么数据他就直接封装网下层IP层送去，不像TCP一样会有时候会对报文的进行分组处理，所以一般UDP都是不可靠，无连接的，除非上层的应用程序承担了检错的任务，所以UDP协议追求的是效率，一般都用在视频语音上了。我们在抓包的时候直接使用udp就可以过滤网络中的udp用户数据报文了。



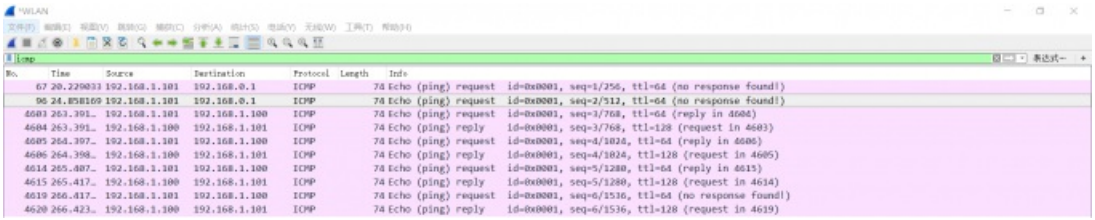
TCP协议

TCP协议是是一种面向字节流，可靠，面向连接的协议，他对于用户传送的报文可以进行分组形成报文段，其中协议中如何保证传输的正确性，如何进行拥塞控制，如何进行连接的建立以及释放都是重点，为了复习，还是回顾一下吧，其中保证传输的正确性窗口(窗口的概念就不多讲了)的作用至关重要，其中最为重要的是其中的ARQ协议(自动重传协议)，对于拥塞控制，最重要的还是那四个算法，慢开始，拥塞避免，快重传，快恢复，关键点就在拥塞窗口值以及门限值的变化并且在不同情况下使用的算法也不一样，慢开始是使用在一开始的时候以及发现超时的时候使用的，而快重传，快恢复是在在超时计数器结束前连续收到三个响应报文使用的。在过滤的时候我们只需要用tcp命令就可把TCP报文给滤出来了。



ICMP协议

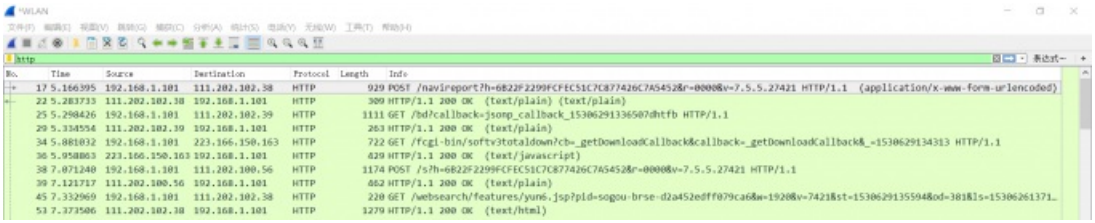
ICMP协议是一种用于报告错误信息的协议，位于IP层，我们平常使用的ping，tracert命令都是属于这一协议里面的，这两个命令都十分常用，ping用于及钠盐网络是否连通，而是用tracert命令我们可以探测出网络拓扑结构，所以要抓到icmp的报文，只需要ping一下然后在Wireshark里面用icmp这一个命令过滤出来即可。



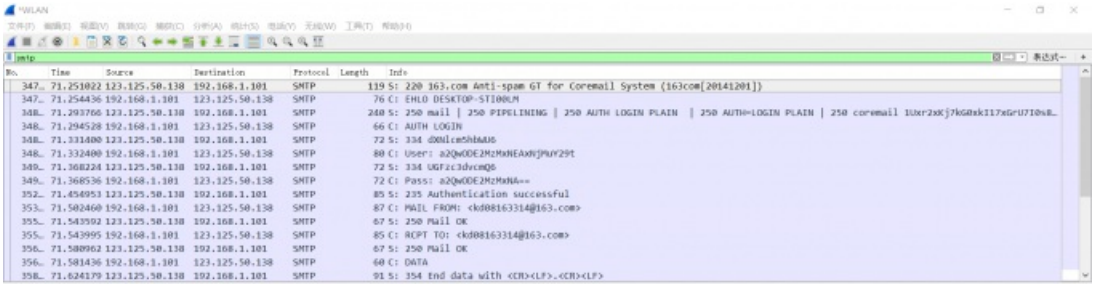
实验内容第二点

先把各部分的包的先过滤出来

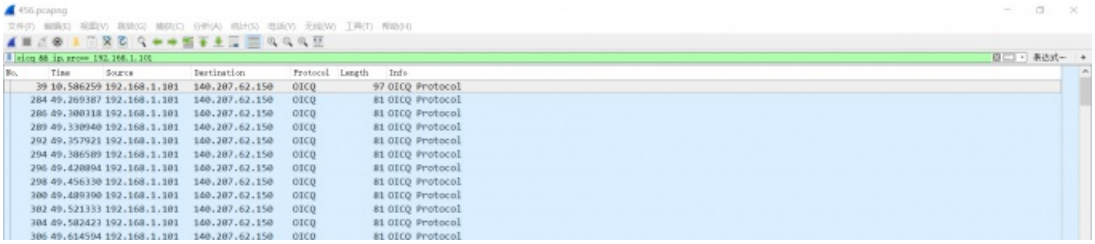
WWW服务数据包：



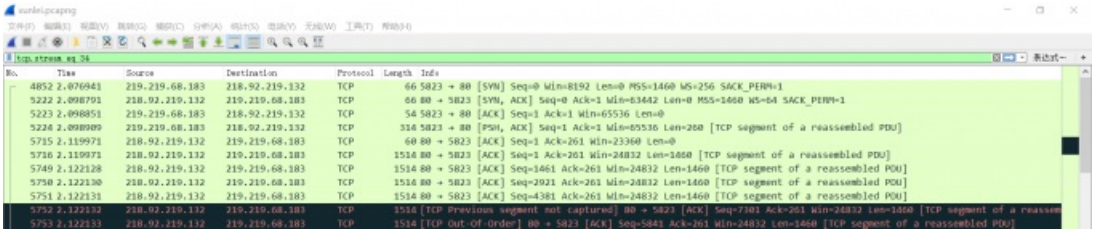
Email服务数据包：



QQ通信数据包：



迅雷文件下载服务数据包：

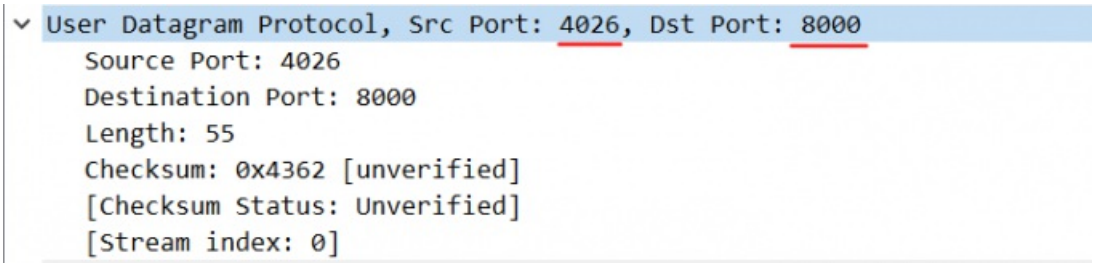


不同互联网访问情形下的数据包进行逐层分析，先Packet Details Pane(封包详细信息)有大致理解, 显示封包中的字段，各行信息说明如下：

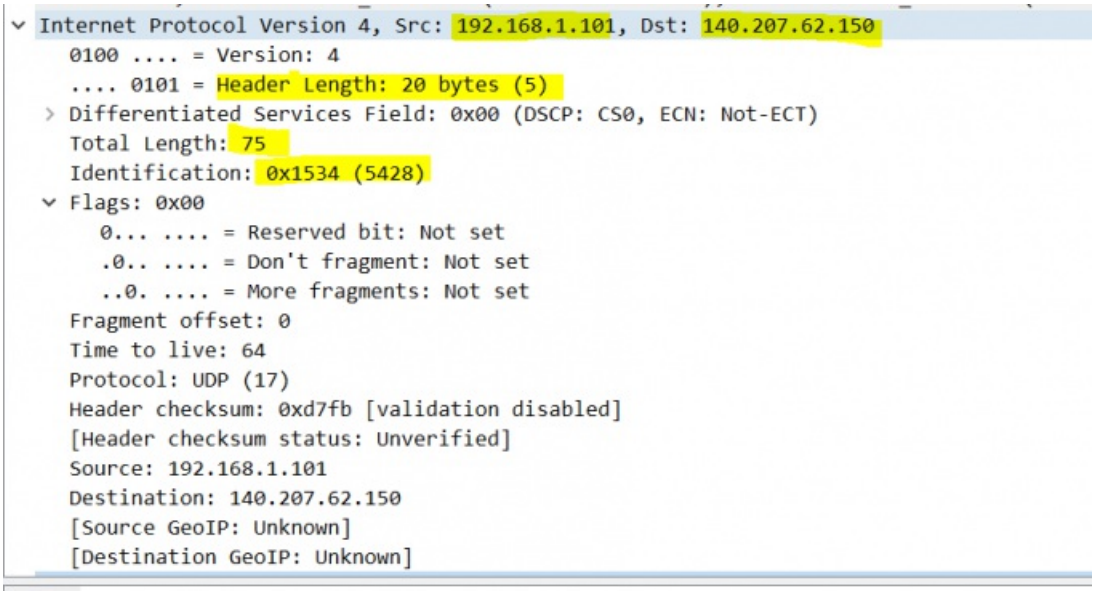
1. Frame: 物理层的数据帧概况(这是最底层的，一般以比特流传送，看不懂)
2. Ethernet II: 数据链路层以太网帧头部信息，
3. Internet Protocol Version 4: 互联网层IP包头部信息
4. Transmission Control Protocol: 传输层T的数据段头部信
5. Hypertext Transfer Protocol: 应用层的信息

UDP数据包详细分析：

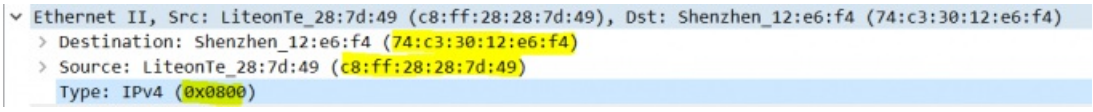
这个数据包的详细信息里面包含了物理层(Frame那一行)，数据链路层(Ethernet那一行)，网路层(Internet Protocol那一行)，传输层(User Datagram Protocol那一行)还有应用层数据，层次结构十分明显。
从UDP的报文格式可以得到UDP的首部信息，源端口为4026，目的端口为8000，UDP长度为55，检验和为0x4362



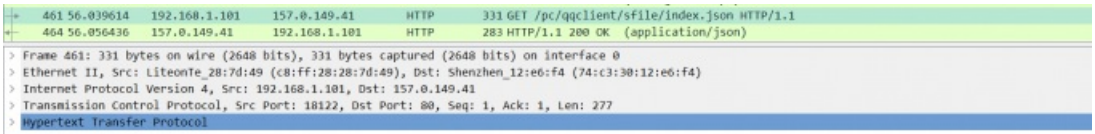
再看IP层的内容



可以清楚分析出他的首部信息，首部长为20字节，还有标识位5428，3位没有设置的标志位以及总长度偏移量，TTL为64，协议字段17，代表了上层使用UDP，下面就是源IP为192.168.1.101，目的IP为140.207.62.105
再往下分析就是数据链路层了，里面的信息也很清楚，把源MAC地址还有目的MAC地址显示出来，并且上层IP类型为IPv4源MAC地址:74:c3:30:12:e6:f4,目的MAC地址:c8:ff:28:28:7d:49

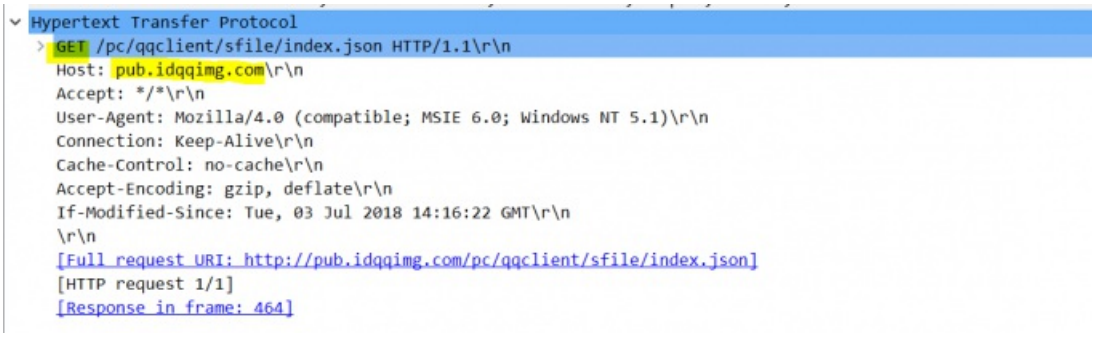


http数据包详细分析：

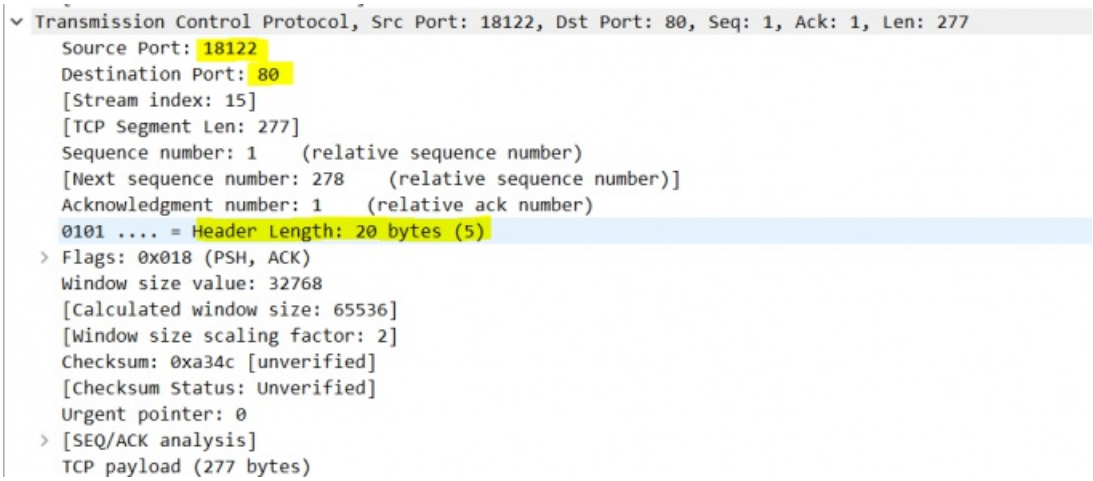


对于http数据进行分析，发现还是离不开五层协议，物理层，数据链路层，网路层，传输层，还有应用层，层次结构十分明显，

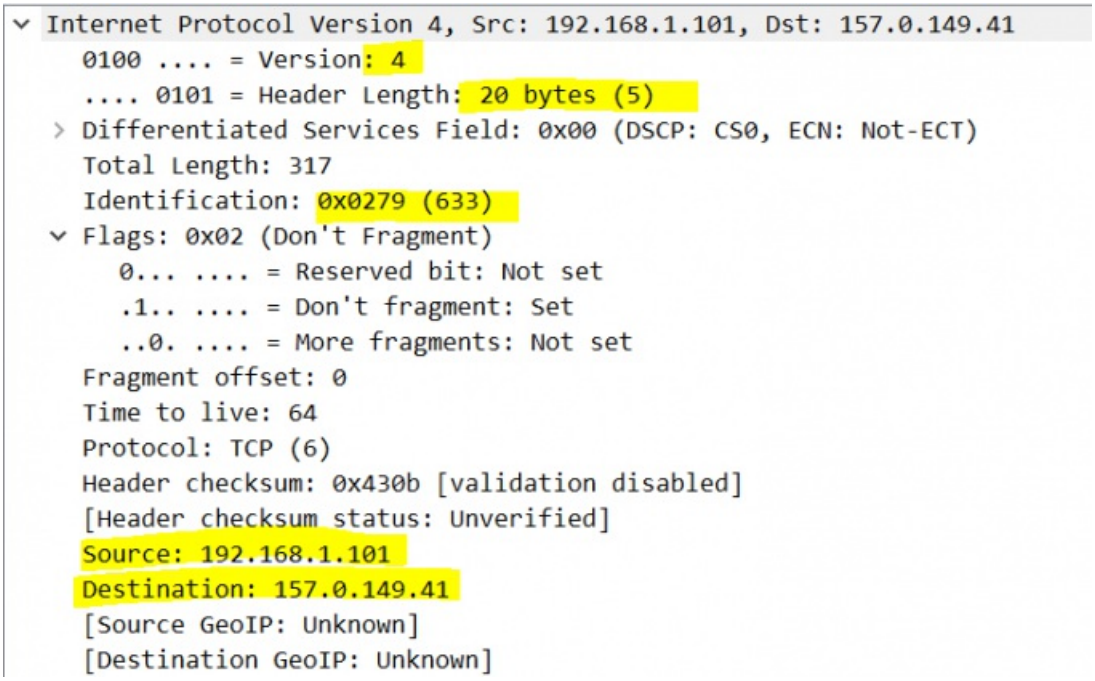
对于http数据进行分析，发现是高层并且应用层协议，物理层，数据链路层，网络层，传输层，还有应用层，层次结构十分明显，这一次应用层以http报文显示，里面传递的的东西很明确，通过GET方式访问网站资源，访问的主机名为pub.idqqimg.com,还有其他的一些浏览器相关的信息等等。



下面一层是TCP，可以看出TCP首部信息，我们也可以从另一个方面推出http协议使用的运输层协议是tcp，源端口为18122，目的端口为80，还有它的序号以及确认信号，还可以看到标志位Flags，窗口大小为32768，首部长度为20，检验和是0x0dfc,紧急指针Urgent pointer置0



再看IP层的内容，可以清楚分析出他的首部信息，版本号为4，首部长为20字节，还有标识633，3位没有设置的标志位Flags以及总长度偏移量，TTL为64，协议字段6，代表了上层使用TCP，下面就是源IP为192.168.1.101，目的IP为157.0.149.41



分析数据链路层，里面的信息也很清楚，把源MAC地址还有目的MAC地址显示出来，并且上层IP类型为IPv4，源MAC地

址:74:c3:30:12:e6:f4,目的MAC地址:c8:ff:28:28:7d:49

```
Ethernet II, Src: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49), Dst: Shenzhen_12:e6:f4 (74:c3:30:12:e6:f4)
  > Destination: Shenzhen_12:e6:f4 (74:c3:30:12:e6:f4)
  > Source: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49)
    Type: IPv4 (0x0800)
```

TCP包详细分析：

```
Transmission Control Protocol, Src Port: 18122, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 18122
    Destination Port: 80
    [Stream index: 15]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 65535
    [Calculated window size: 65535]
    Checksum: 0x16eb [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
```

分析TCP同样可以看出TCP首部信息，源端口为18122，目的端口为80，还有它的序号以及确认信号，还可以看到标志位Flags，窗口大小为65535，首部长度为20，检验和是0x16eb,紧急指针Urgent pointer置0

IP层的内容，可以清楚分析出他的首部信息，版本号为4，首部长为20字节，还有标识631，3位没有设置的标志位Flags以及总长度偏移量，TTL为64，协议字段6，代表了上层使用TCP，下面就是源IP为192.168.1.101，目的IP为157.0.149.41

```
Internet Protocol Version 4, Src: 192.168.1.101, Dst: 157.0.149.41
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0277 (631)
  > Flags: 0x02 (Don't Fragment)
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x4416 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.101
    Destination: 157.0.149.41
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

数据链路层，里面的信息也很清楚，把源MAC地址还有目的MAC地址显示出来，并且上层IP类型为IPv4，源MAC地址:74:c3:30:12:e6:f4,目的MAC地址:c8:ff:28:28:7d:49

```
Ethernet II, Src: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49), Dst: Shenzhen_12:e6:f4 (74:c3:30:12:e6:f4)
  > Destination: Shenzhen_12:e6:f4 (74:c3:30:12:e6:f4)
  > Source: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49)
    Type: IPv4 (0x0800)
```

ICMP包详细分析：

使用icmp过滤相关的报文，分析根据报文格式分析icmp包里面的内容，Type显示8表明这是一个请求报文

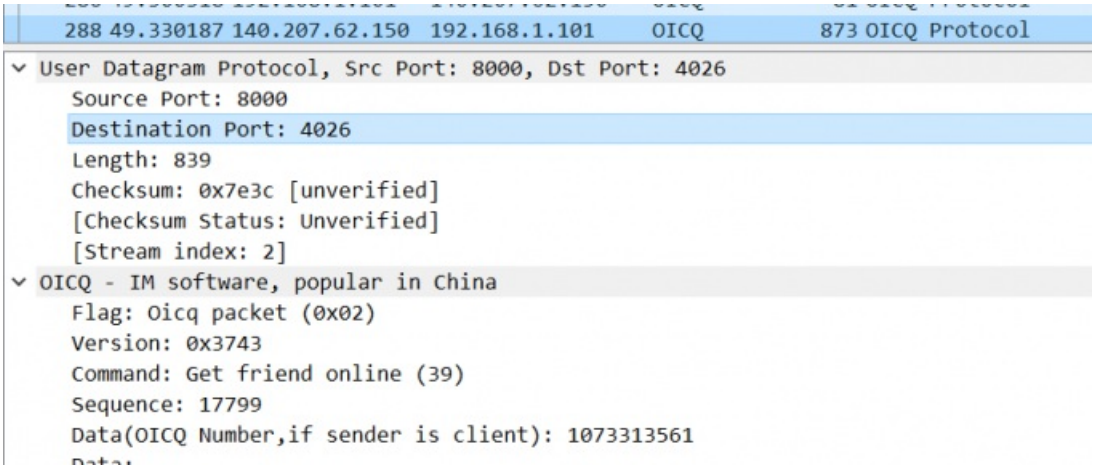
```
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d55 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 6 (0x0006)
    Sequence number (LE): 1536 (0x0600)
    [Response frame: 4620]
```

分析网络层和数据链路层跟前面几次没什么区别，源IP为192.168.1.101，目的IP为192.168.1.100，IP首部为20字节，IP类型为IPv4，源MAC地址:74:c3:30:12:e6:f4,目的MAC地址:c8:ff:28:28:7d:49

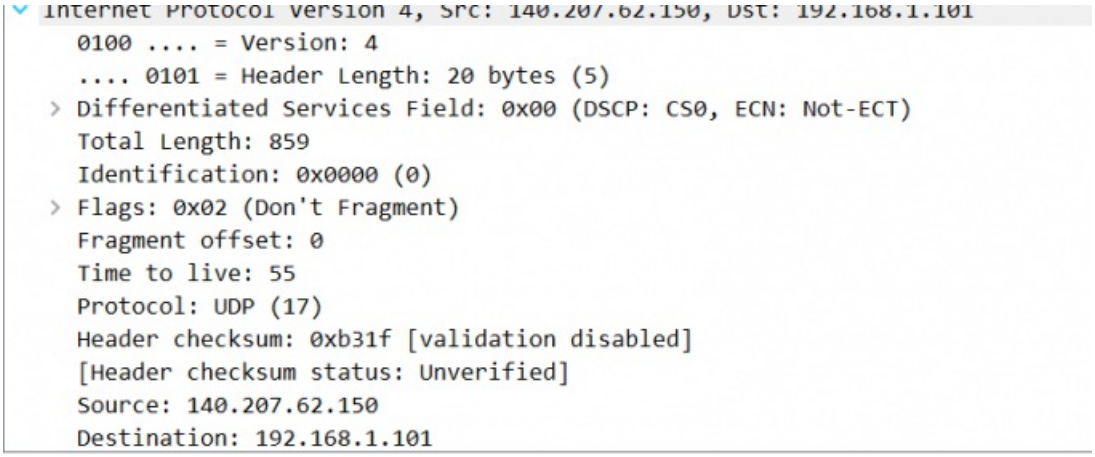
```
Internet Protocol Version 4, Src: 192.168.1.101, Dst: 192.168.1.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x37b4 (14260)
  > Flags: 0x00
    Fragment offset: 0
    Time to live: 64
    Protocol: ICMP (1)
    Header checksum: 0xbef3 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.101
    Destination: 192.168.1.100
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
```

```
Ethernet II, Src: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49), Dst: Vmware_a0:f8:e4 (00:0c:29:a0:f8:e4)
  > Destination: Vmware_a0:f8:e4 (00:0c:29:a0:f8:e4)
  > Source: LiteonTe_28:7d:49 (c8:ff:28:28:7d:49)
    Type: IPv4 (0x0800)
```

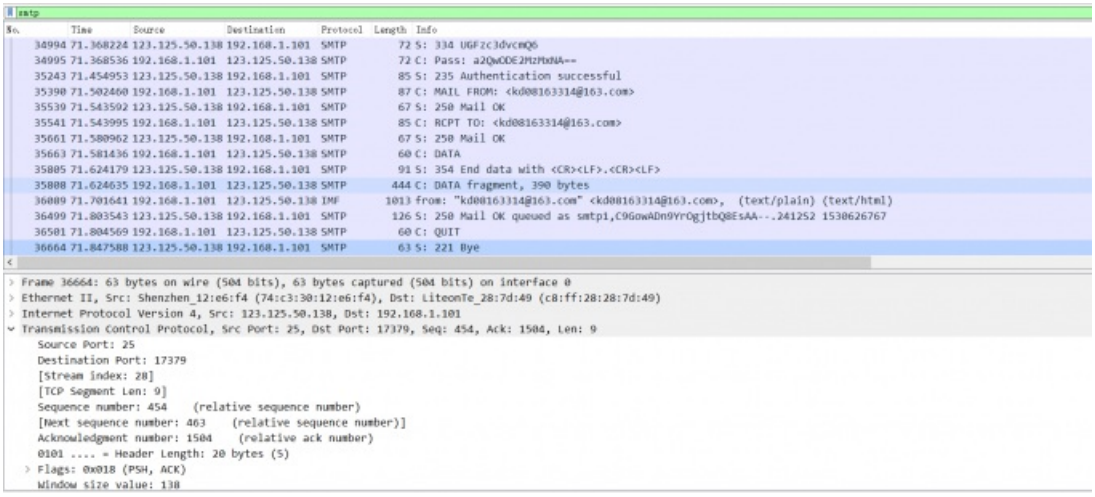
分析QQ的报文：



对于QQ的包可以得出来应用层是使用了OICQ的协议，里面包含了QQ号的信息以及加密过了的数据，使用的是UDP的传输协议，源端口是8000，目的端口是4026
对于网络层以及数据链路层是一样的分析过程，源IP是140.207.62.150，目的IP是192.168.1.101，传输层使用的是UDP，TTL是55



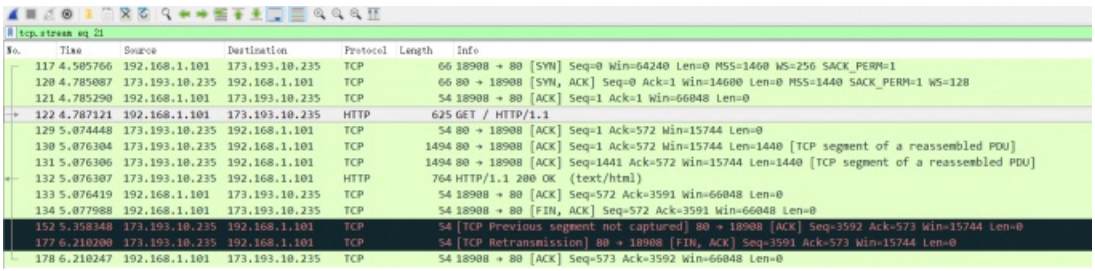
对邮件包分析：

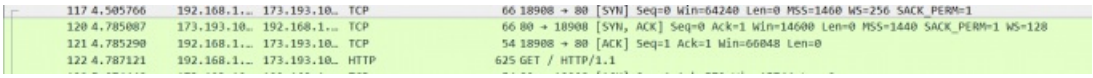


可以看得出来邮件传送有一个建立连接的过程，从哪里发的并且在哪里接收都有对应的邮箱，建立连接的密码用base64加密过了，看下层运输层是使用TCP建立连接的，目的端口是17379，源端口号25，窗口值138，序号是454，IP版本是IPv4，源IP是192.168.1.101， 目的IP是123.125.50.138

实验内容第三点

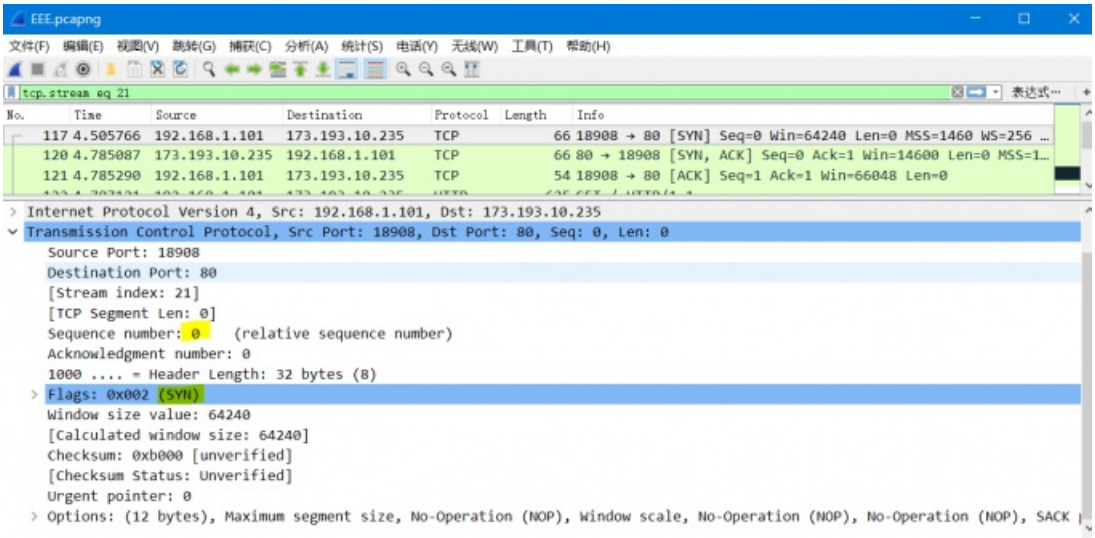
访问网站`www.souci.li`,过滤http然后追踪tcp流就可以看见http协议工作的大概过程(在这之前应该还得通过DNS去解析域名，你会发现这个包过滤DNS会发现把域名转换成IP的过程)，然后就是封装http请求数据包，封装成tcp包，建立tcp连接（三报文握手在HTTP工作开始 之前，客户机（Web浏览器）首先要通过网络与服务器建立连接，该连接是通过TCP来完成的，客户端发送请求，服务器响应，服务器关闭tcp连接





分析三报文握手的过程：

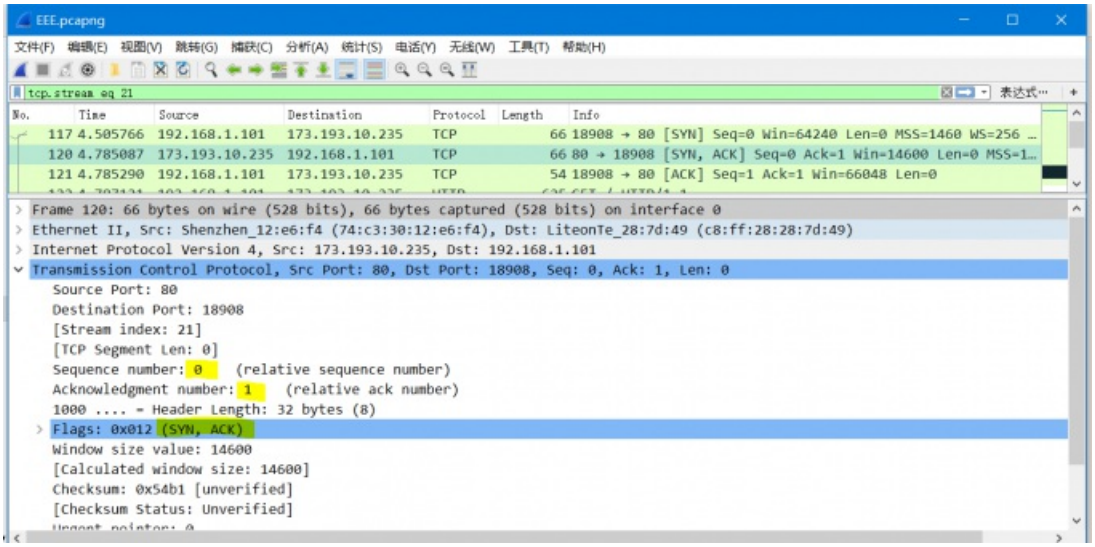
第一次报文握手



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

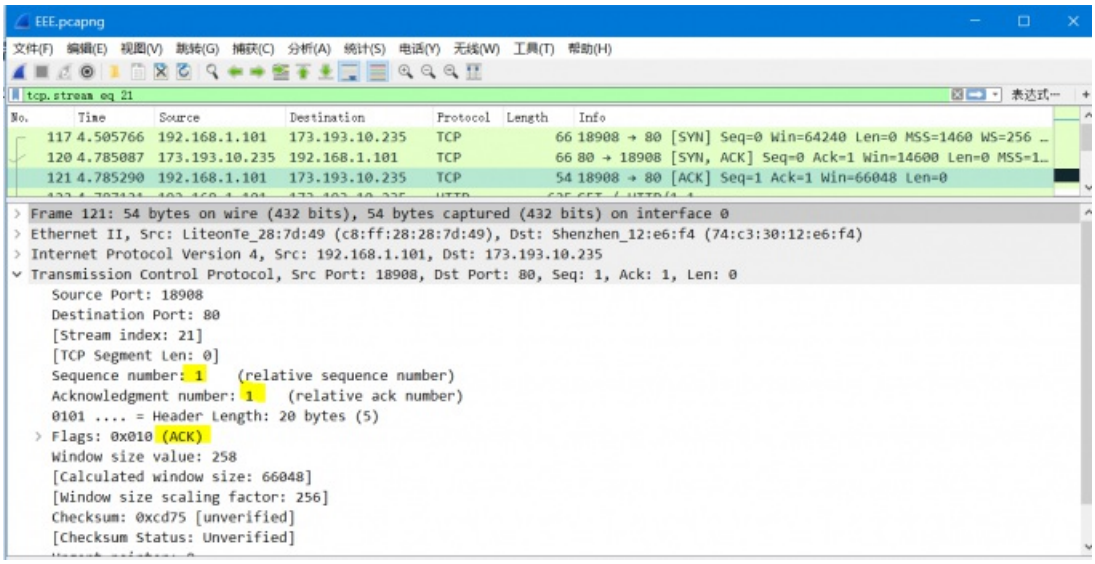
第二次握手的数据包

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



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, 如下图:



就这样通过了TCP三报文握手，建立了连接

小小回顾，感觉废话太多，emmm。。。

欢迎大家多来踩踩我的博客:https://0verwatch.top