**密 级：内部公开**

**版 本 号：V1.0**

**抓包步骤以及抓包分析样列**

|  |  |
| --- | --- |
| 编制：梁浩 | 创建日期：2015-8-06 |
| 审核： | 批准： |

目录

[目录 2](#_Toc429320287)

[1 文档概述 3](#_Toc429320288)

[2 抓包介绍 3](#_Toc429320289)

[2.1 Wireshark抓包 3](#_Toc429320290)

[2.1.1 抓包方法 3](#_Toc429320291)

[2.1.2 无线网络共享抓包方式 4](#_Toc429320292)

[2.2 TCPdump抓包 5](#_Toc429320293)

[2.2.1 Tcpdump命令详解 5](#_Toc429320294)

[2.2.2 实例展示 6](#_Toc429320295)

[3 Wireshark抓包文件分析 7](#_Toc429320296)

[3.1 抓包界面介绍 7](#_Toc429320297)

[3.2 Wireshark过滤规则 9](#_Toc429320298)

[3.3 封包列表(Packet List Pane) 10](#_Toc429320299)

[3.4 实例展示 11](#_Toc429320300)

[3.5 典型样列 14](#_Toc429320301)

[3.5.1 500错误-代码逻辑错误 14](#_Toc429320302)

[3.5.2 404报错 15](#_Toc429320303)

[3.5.3 页面返回之后焦点未记忆 16](#_Toc429320304)

[3.5.4 用户session丢失 16](#_Toc429320305)

[3.5.5 EPG视频无法播放 17](#_Toc429320306)

[4 小结 17](#_Toc429320307)

# 文档概述

工作中经常出现无法快速的定位的问题，此时可通过抓包并分析抓包文件来协助判断问题以及定位，本文档主要介绍了如何使用抓包的工具以及简单的分析。

# 抓包介绍

目前主要介绍两种方式:Wireshark、tcpdump

Wireshark是一个网络协议检测工具，支持Windows平台和linux平台，我一般只在Windows平台下使用Wireshark抓起机顶盒的包，如果是Linux的话，则可直接使用tcpdump。

tcpdump是基于Unix系统的命令行式的数据包嗅探工具。如果要使用tcpdump抓取其他主机MAC地址的数据包，必须开启网卡混杂模式，所谓混杂模式，用最简单的语言就是让网卡抓取任何经过它的数据包，不管这个数据包是不是发给它或者是它发出的，

可参考【<http://en.wikipedia.org/wiki/Promiscuous_mode>】获取更多有关混杂模式的资料。

## Wireshark抓包

### 抓包方法

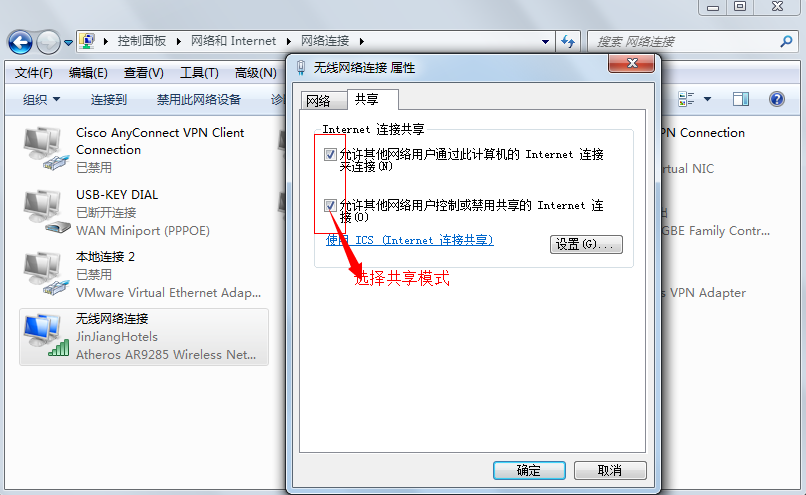
使用Wireshark抓包，主要有两种方式，一使用HUB（集线器）抓包，二是直接使用本机的无线网络共享抓包，此次重点介绍无线网络共享抓包。

1、HUB抓包清单：3根网线、一个HUB、相关的机顶盒、以及笔记本（已安装wireshark）；

2、无线网络共享抓包清单：一个网线，笔记本（已安装wireshark），相关机顶盒。

### 无线网络共享抓包方式

1. 笔记本连上无线网络，相关的网卡设置成共享模式；



1. 设置本地网关，

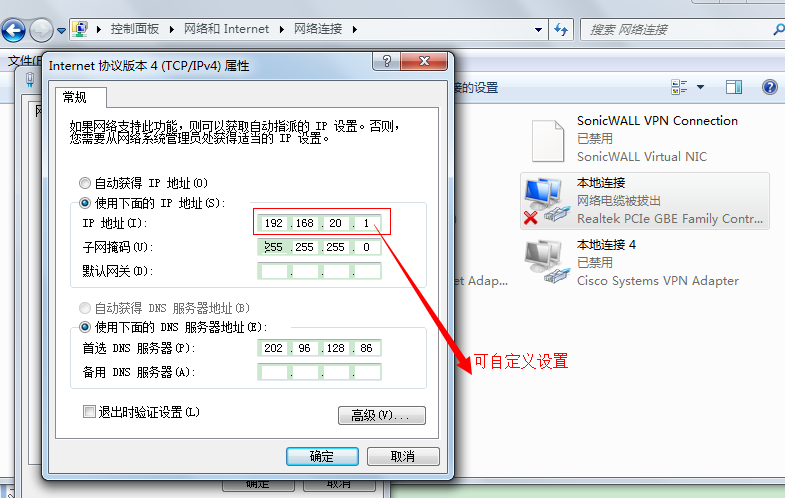
原理：将通过无线共享，将网络共享出来，给机顶盒上网，所以需保证机顶盒与共享网络在同一网段内。

步骤：

（1）：使用网线直了连机顶盒与笔记本；

（2）：打开笔记本 相应网卡的本地连接，-设置静态IP以及网关，

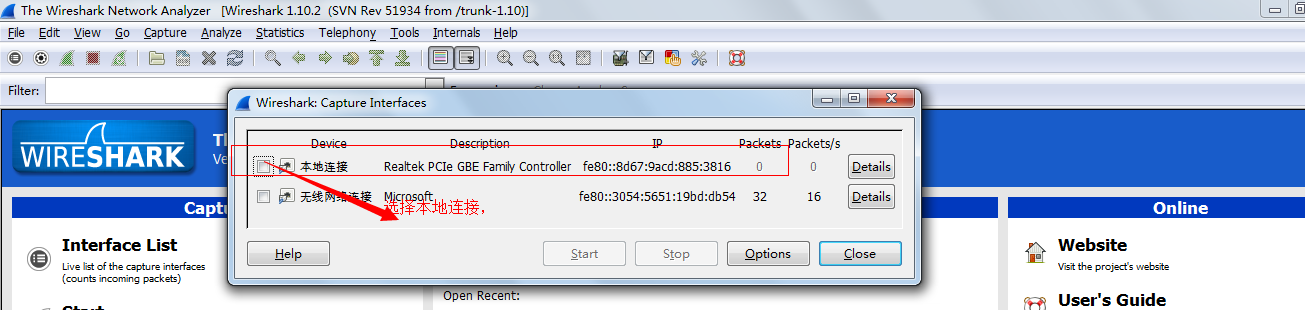
（3）：设置机顶盒IP与网关，需与PC端上一步设置的一致



3、抓包

此时STB可正常访问页面

进入Wireshark，选择监听端口，选择相应的网卡，点击start，进入抓包



## TCPdump抓包

如前文所说，tcpdump抓包运用于抓取服务器上发出的网络包，tcpdump可以将网络中传送的数据包的“头”完全截获下来提供分析。它支持针对网络层、协议、主机、网络或端口的过滤，并提供and、or、not等逻辑语句来帮助你去掉无用的信息。

### Tcpdump命令详解

|  |
| --- |
| -A 以ASCII格式打印出所有分组，并将链路层的头最小化。  -c 在收到指定的数量的分组后，tcpdump就会停止。  -C 在将一个原始分组写入文件之前，检查文件当前的大小是否超过了参数file\_size 中指定的大小。如果超过了指定大小，则关闭当前文件，然后在打开一个新的文件。参数 file\_size 的单位是兆字节（是1,000,000字节，而不是1,048,576字节）。  -d 将匹配信息包的代码以人们能够理解的汇编格式给出。  -dd 将匹配信息包的代码以c语言程序段的格式给出。  -ddd 将匹配信息包的代码以十进制的形式给出。  -D 打印出系统中所有可以用tcpdump截包的网络接口。  -e 在输出行打印出数据链路层的头部信息。  -E 用spi@ipaddr algo:secret解密那些以addr作为地址，并且包含了安全参数索引值spi的IPsec ESP分组。  -f 将外部的Internet地址以数字的形式打印出来。  -F 从指定的文件中读取表达式，忽略命令行中给出的表达式。  -i 指定监听的网络接口。  -l 使标准输出变为缓冲行形式，可以把数据导出到文件。  -L 列出网络接口的已知数据链路。  -m 从文件module中导入SMI MIB模块定义。该参数可以被使用多次，以导入多个MIB模块。  -M 如果tcp报文中存在TCP-MD5选项，则需要用secret作为共享的验证码用于验证TCP-MD5选选项摘要（详情可参考RFC 2385）。  -b 在数据-链路层上选择协议，包括ip、arp、rarp、ipx都是这一层的。 -n 不把网络地址转换成名字。 -nn 不进行端口名称的转换。 -N 不输出主机名中的域名部分。例如，‘nic.ddn.mil‘只输出’nic‘。  -t 在输出的每一行不打印时间戳。  -O 不运行分组分组匹配（packet-matching）代码优化程序。  -P 不将网络接口设置成混杂模式。  -q 快速输出。只输出较少的协议信息。  -r 从指定的文件中读取包(这些包一般通过-w选项产生)。  -S 将tcp的序列号以绝对值形式输出，而不是相对值。  -s 从每个分组中读取最开始的snaplen个字节，而不是默认的68个字节。  -T 将监听到的包直接解释为指定的类型的报文，常见的类型有rpc远程过程调用）和snmp（简单网络管理协议；）。  -t 不在每一行中输出时间戳。  -tt 在每一行中输出非格式化的时间戳。  -ttt 输出本行和前面一行之间的时间差。  -tttt 在每一行中输出由date处理的默认格式的时间戳。  -u 输出未解码的NFS句柄。  -v 输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息。  -vv 输出详细的报文信息。  -w 直接将分组写入文件中，而不是不分析并打印出来 |

### 实例展示

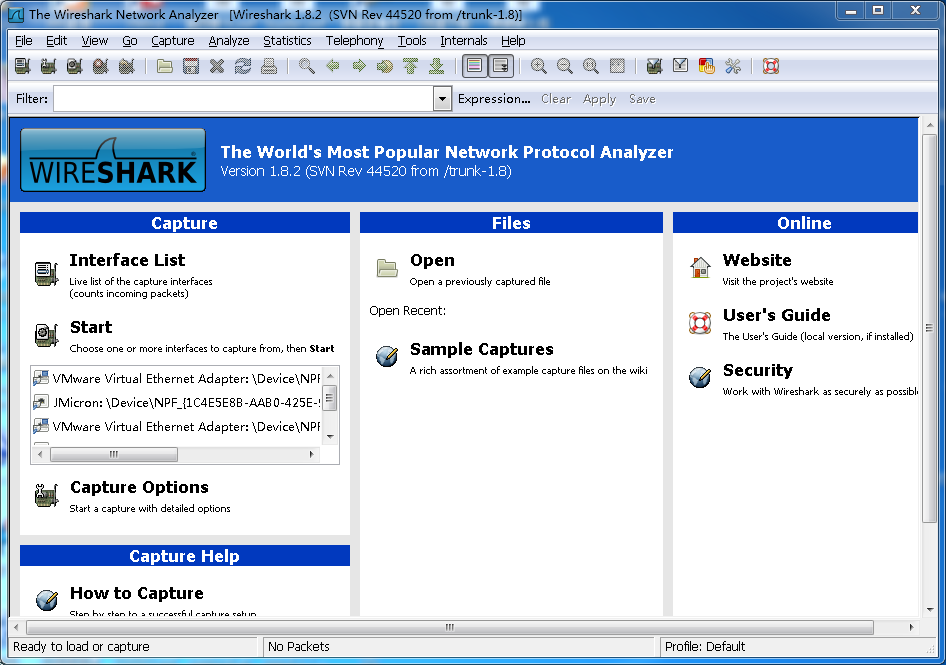
|  |
| --- |
| 下面的例子全是以抓取eth0接口为例，如果不加”-i eth0”是表示抓取所有的接口。 1、抓取包含10.10.10.122的数据包  # tcpdump -i eth0 -vnn host 10.10.10.122  2、抓取包含10.10.10.0/24网段的数据包  # tcpdump -i eth0 -vnn net 10.10.10.0/24  3、抓取包含端口22的数据包  # tcpdump -i eth0 -vnn port 22  4、抓取源ip是10.10.10.122数据包。  # tcpdump -i eth0 -vnn src host 10.10.10.122  5、抓取目的ip是10.10.10.122数据包  # tcpdump -i eth0 -vnn dst host 10.10.10.122  6、抓取源ip是10.10.10.2且目的端口是22，或源ip是10.10.10.65且目的端口是80的数据包。  # tcpdump –i eth0 -vnn -w /tmp/fil1 -c 100  7、从/tmp/fill记录中读取tcp协议的数据包  # tcpdump –i eth0 -vnn -r /tmp/fil1 tcp  8、从/tmp/fill记录中读取包含10.10.10.58的数据包  # tcpdump –i eth0 -vnn -r /tmp/fil1 host 10.10.10.58 |

# Wireshark抓包文件分析

## 抓包界面介绍

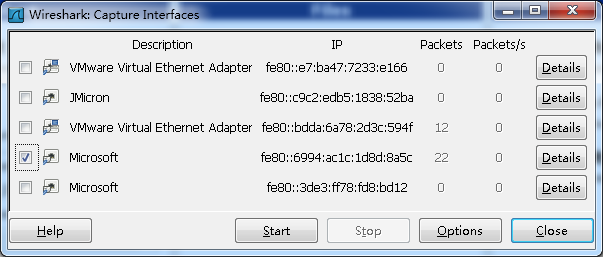
**wireshark 开始抓包**

开始界面

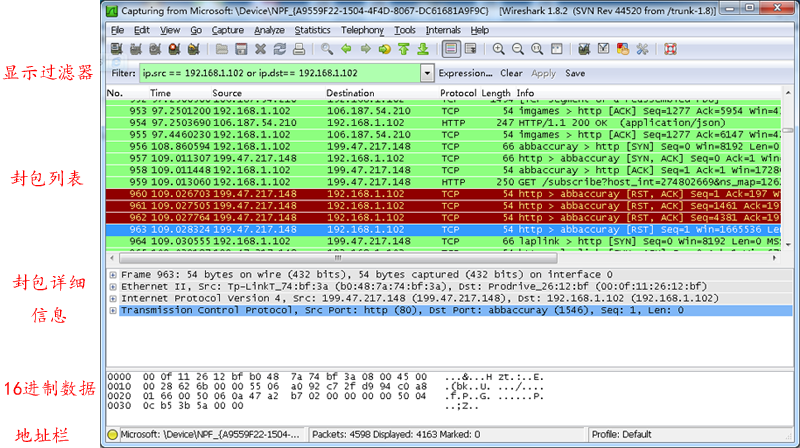


**wireshark是捕获机器上的某一块网卡的网络包**，当机器上有多块网卡的时候，需要选择一个网卡。

点击Caputre->Interfaces.. 出现下面对话框，选择正确的网卡。然后点击"Start"按钮, 开始抓包



Wireshark 窗口介绍



**WireShark 主要分为这几个界面**

1. Display Filter(显示过滤器)，  用于过滤

2. Packet List Pane(封包列表)， 显示捕获到的封包， 有源地址和目标地址，端口号。 颜色不同，代表

3. Packet Details Pane(封包详细信息), 显示封包中的字段

4. Dissector Pane(16进制数据)

5. Miscellanous(地址栏，杂项)

http://www.cr173.com/up/2013-5/2013050217125625854.png

## Wireshark过滤规则

使用wireshark时，会得到大量的冗余信息，在几千甚至几万条记录中，以至于很难找到自己需要的部分。搞得晕头转向。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。

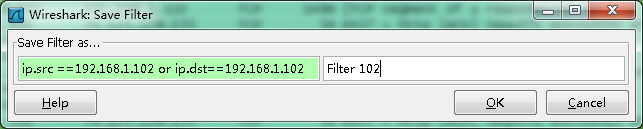
过滤器有两种，

一种是显示过滤器，就是主界面上那个，用来在捕获的记录中找到所需要的记录

一种是捕获过滤器，用来过滤捕获的封包，以免捕获太多的记录。 在Capture -> Capture Filters 中设置

保存过滤

在Filter栏上，填好Filter的表达式后，点击Save按钮， 取个名字。比如"Filter 102",



Filter栏上就多了个"Filter 102" 的按钮。

http://www.cr173.com/up/2013-5/2013050217125766695.png

**过滤表达式的规则**

表达式规则

 1. 协议过滤

比如TCP，只显示TCP协议。

2. IP 过滤

比如 ip.src ==192.168.1.102 显示源地址为192.168.1.102，

ip.dst==192.168.1.102, 目标地址为192.168.1.102

3. 端口过滤

tcp.port ==80,  端口为80的

tcp.srcport == 80,  只显示TCP协议的愿端口为80的。

4. Http模式过滤

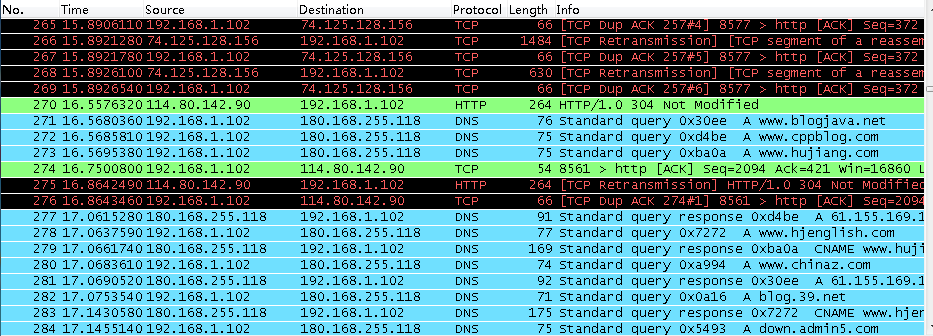
http.request.method=="GET",   只显示HTTP GET方法的。

5. 逻辑运算符为 AND/ OR

## 封包列表(Packet List Pane)

封包列表的面板中显示，编号，时间戳，源地址，目标地址，协议，长度，以及封包信息。 可以看到不同的协议用了不同的颜色显示。

也可以修改这些显示颜色的规则，  View ->Coloring Rules.



封包详细信息 (Packet Details Pane)

这个面板是我们最重要的，用来查看协议中的每一个字段。

各行信息分别为

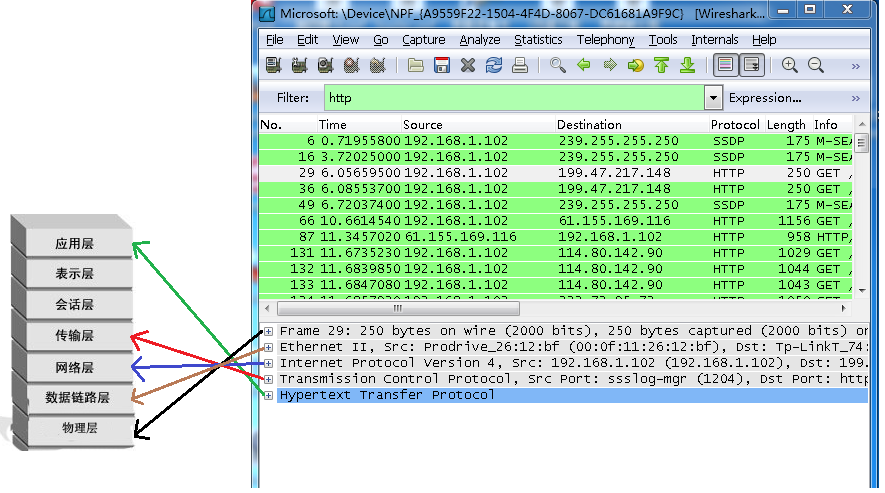
Frame:   物理层的数据帧概况

Ethernet II: 数据链路层以太网帧头部信息

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

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

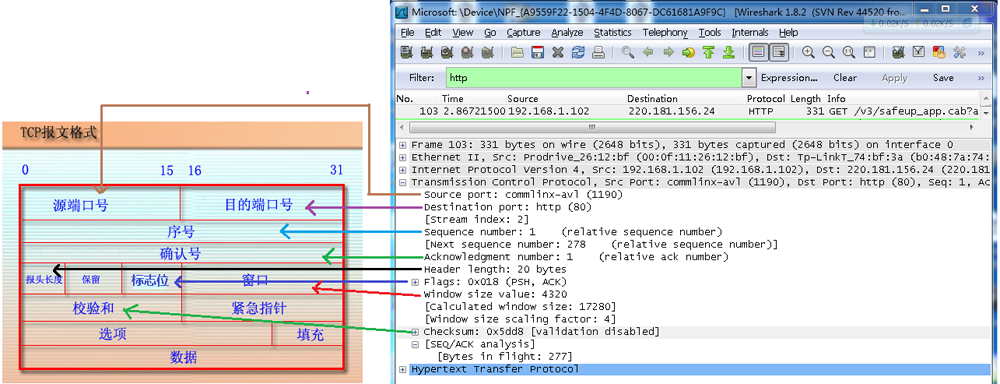
Hypertext Transfer Protocol:  应用层的信息，此处是HTTP协议



## 实例展示

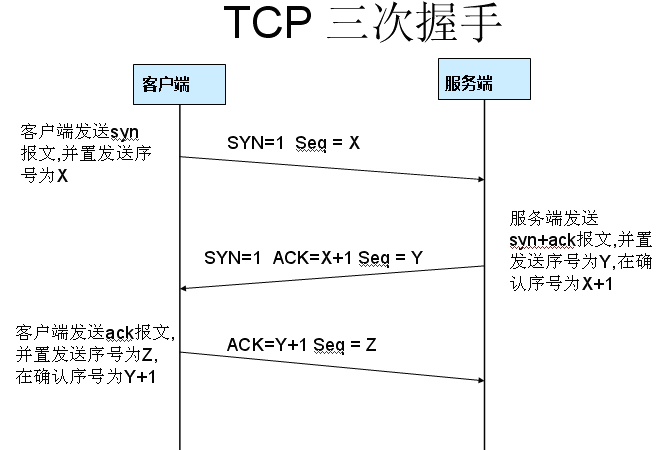
**TCP包的具体内容**

 从下图可以看到wireshark捕获到的TCP包中的每个字段。



看到这， 基本上对wireshak有了初步了解，

TCP三次握手的实例，三次握手过程如下



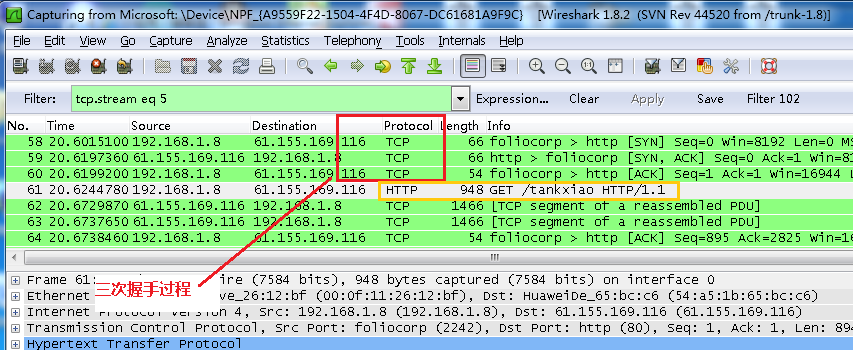
三次握手， 使用wireshark实际分析下三次握手的过程。

1、打开wireshark, 打开浏览器输入 http://www.cr173.com

在wireshark中输入http过滤， 然后选中GET /tankxiao HTTP/1.1的那条记录，

右键然后点击"Follow TCP Stream",

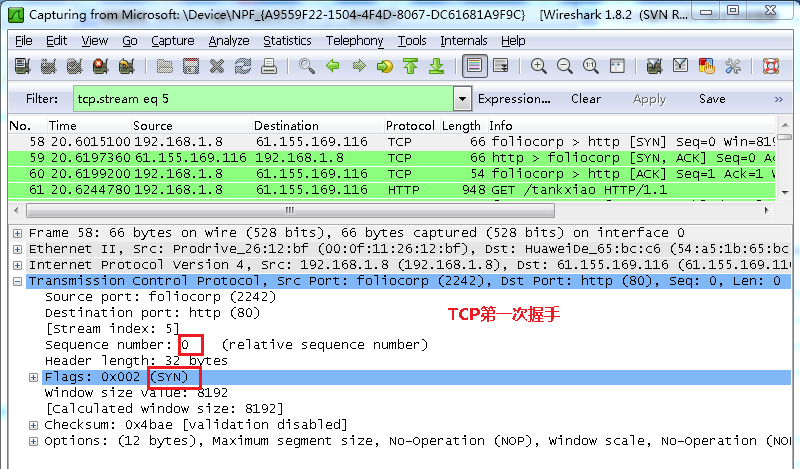
这样做的目的是为了得到与浏览器打开网站相关的数据包，将得到如下图



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。

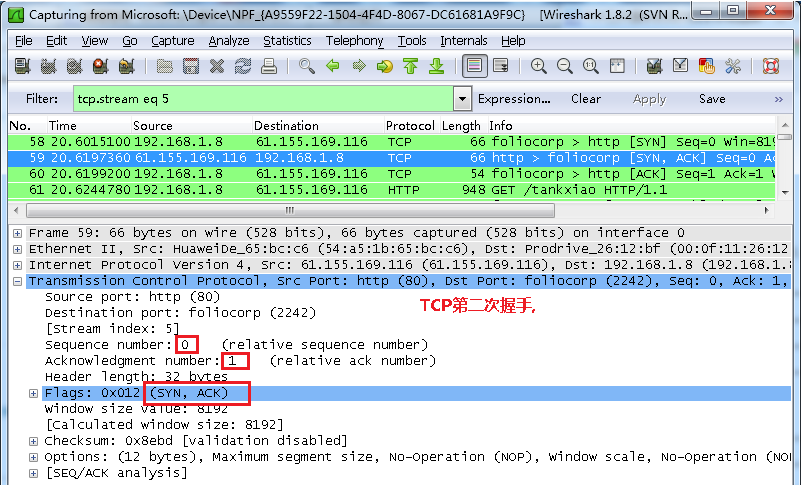
**第一次握手数据包**

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



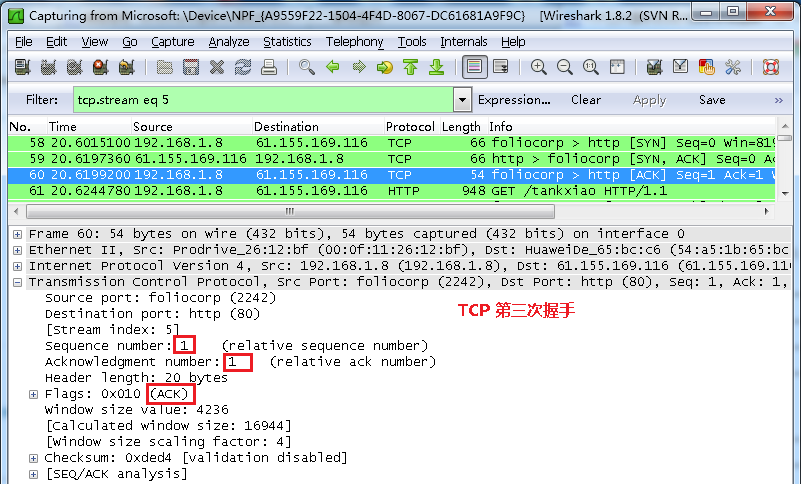
**第二次握手的数据包**

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.并且在数据段放写ISN的+1, 如下图:



 就这样通过了TCP三次握手，建立了连接，成功生成相应的http页面

## 典型样列

日常工作中经常遇到如500、404、2024等报错，或者播放不出、播放黑屏等错误，使用wireshark抓的包，可使用公司轻量级工具SmartWork，将抓包文件上传。

### 500错误-代码逻辑错误

错误描述：南传标清厂商EPG点击直播频道，界面报错，由于无法看到服务器日志，通过查看。500一般都是内部代码错误，需要找到出错的代码位置。

如下面抓包文件，Smartwork分析之后界面如下，



序号2的响应码为500，点开之后可查看到信息如下



点击响应内容，可查看到代码中有报错



|  |
| --- |
| An error occurred at line: 172 **in** the jsp file: /jsp/tzsmchd/en/play/hd\_channel\_play.jsp  Syntax error, insert "Finally" to complete TryStatement  169:  170: **function** **isNeedOrderNow**() {  171: orderCode = **getCookie**("live\_channel\_order\_code");  172: **var** orderFlag = **getCookie**("live\_trailer\_need\_order");  173: **if**("1" == orderFlag) {  174: **clearCookie**("live\_trailer\_need\_order");  175: **clearCookie**("live\_channel\_order\_code"); |

此类错误，可将这段错误，截取给研发，或者也可按照信息，找到相应的代码进行查看。

### 404报错

404报错，一般两种情况：文件不存在、或者目录路径错误。如IE浏览器可查看时，直接点击查看，就可以看到相应的信息，如下图:



此错误为南传厂商EPG进入相应的详情页报错，URL中多了vod

### 页面返回之后焦点未记忆

当遇到此类错误时，如该EPG页面跳转使用的是传入参数的方式，则可以通过IE查看源代码，直接查看，如下。

点击进入到相应的页面-右击





直接查询到了相应的返回URL，将该url取出，直接使用浏览器点击，

Ip+port+/epg/biz/biz\_36089306.do?enterFocusId=amenu\_2

如可查看是否进入页面正确，且是否焦点记忆

### 用户session丢失

用户Session丢失的场景有多种，此实例只介绍一种，如下，客户报障，高清厂商EPG某个外链播放返回报用户信息都是丢失，抓包结果如下



此数据包中，链接IP+port出现了三种形式:

（1）<http://125.88.70.16:8082>

（2）<http://125.88.98.39:90>

（3）<http://125.88.98.39:8087>

这三种链接，正常链接是上面的两种，第一种厂商的页面，第二种为看吧的链接，且看吧的链接为nginx代理之后的VIP以及port。

错误原因：用户在使用外链链接时，使用了VIP加实际服务器的端口，为<http://125.88.98.39:8087>，而正确的应该是使用这样的链接<http://125.88.98.39:90>。

所以导致了用户信息丢信息。

Da

### EPG视频无法播放

遇到视频无法播放，首先确定视频源本身是否可播，如可以播放，接着查看播放的URL是否正确，如浏览器可以打开EPG页面，直接打开浏览器“审查元素”，在EPG相应的位置上，点击该内容进行播放，观察生成的播放URL

# 小结

抓包不一定可以解决所有问题。只是作为一种辅助手段来协助问题的定位，什么样问题需要使用抓包，什么时候需要抓包，都定位清楚。最重要的就是分析清楚问题的性质，这样才可以更快速，更精确的定位问题。