**WireShark使用教程 TCP/UDP**

2017年10月20日 17:01:00 [sinat\_32176267](https://me.csdn.net/sinat_32176267) 阅读数：10660

抓包应该是每个技术人员掌握的基础知识，无论是技术支持运维人员或者是研发，多少都会遇到要抓包的情况，用过的抓包工具有fiddle、wireshark，作为一个不是经常要抓包的人员，学会用Wireshark就够了，毕竟它是功能最全面使用者最多的抓包工具。

Wireshark（前称Ethereal）是一个网络封包分析软件。网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。

网络封包分析软件的功能可想像成 "电工技师使用电表来量测电流、电压、电阻" 的工作 - 只是将场景移植到网络上，并将电线替换成网络线。在过去，网络封包分析软件是非常昂贵的，或是专门属于营利用的软件。Ethereal的出现改变了这一切。在GNUGPL通用许可证的保障范围底下，使用者可以以免费的代价取得软件与其源代码，并拥有针对其源代码修改及客制化的权利。Ethereal是目前全世界最广泛的网络封包分析软件之一。

wireshark的官方下载网站： <http://www.wireshark.org/>

wireshark是非常流行的网络封包分析软件，功能十分强大。可以截取各种网络封包，显示网络封包的详细信息。

wireshark是开源软件，可以放心使用。 可以运行在Windows和Mac OS上。

**Wireshark不能做的**

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 VS Fiddler**

Fiddler是在windows上运行的程序，专门用来捕获HTTP，HTTPS的。

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

**同类的其他工具**

微软的network monitor

sniffer

**什么人会用到wireshark**

1. 网络管理员会使用wireshark来检查网络问题

2. 软件测试工程师使用wireshark抓包，来分析自己测试的软件

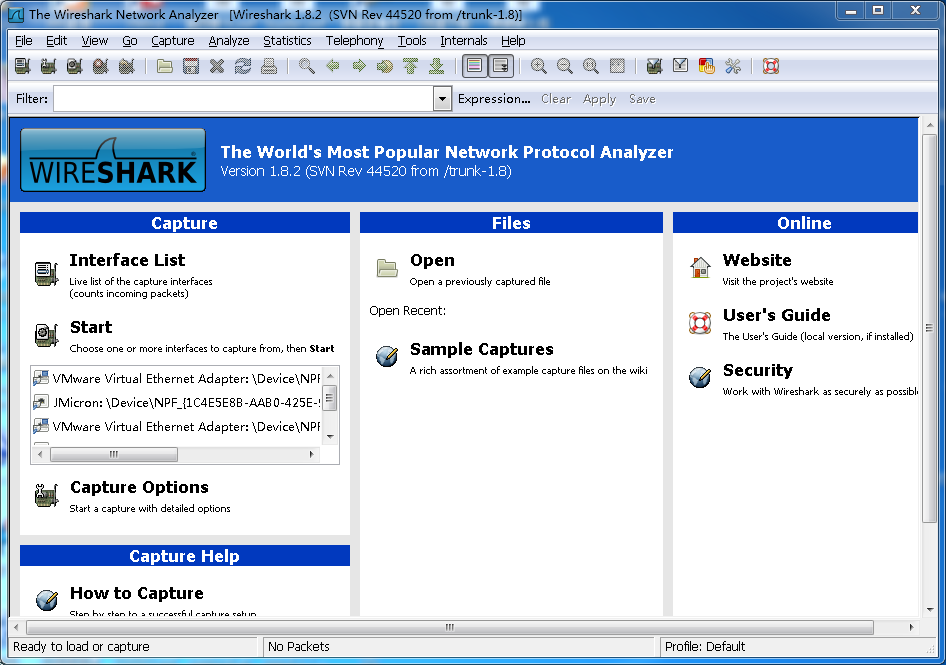
3. 从事socket编程的工程师会用wireshark来调试

4. 听说，华为，中兴的大部分工程师都会用到wireshark。

总之跟网络相关的东西，都可能会用到wireshark.

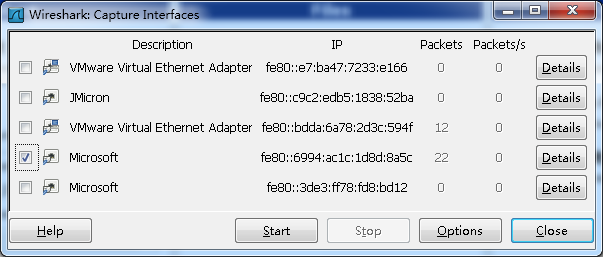
**wireshark 开始抓包**

开始界面

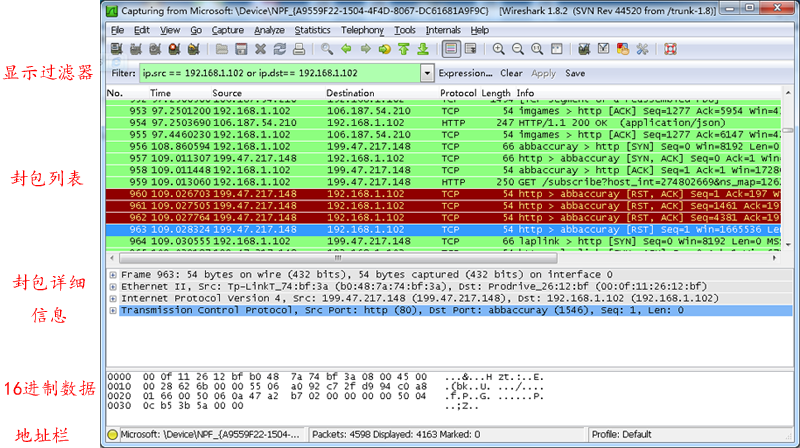


wireshark是捕获机器上的某一块网卡的网络包，当你的机器上有多块网卡的时候，你需要选择一个网卡。

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)

**Wireshark 显示过滤**

http://images2015.cnblogs.com/blog/150046/201608/150046-20160802202004372-179986181.png

使用过滤是非常重要的， 初学者使用wireshark时，将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。

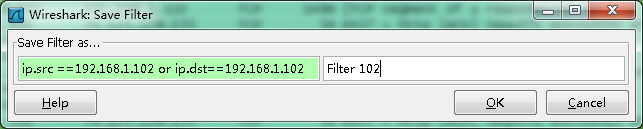
过滤器有两种，

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://images2015.cnblogs.com/blog/150046/201608/150046-20160802202005497-535988035.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

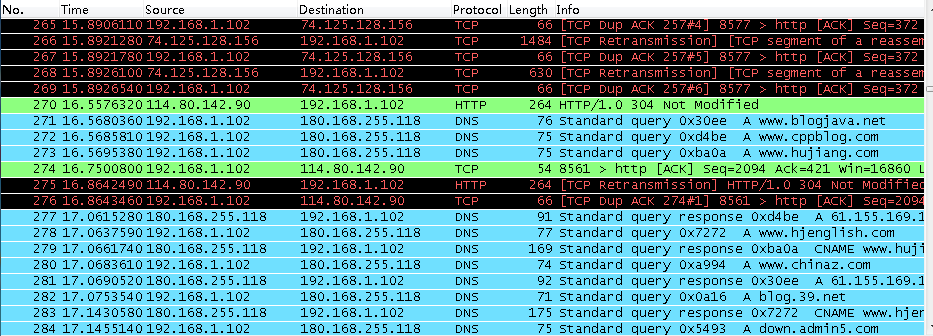
常用的过滤表达式

|  |  |
| --- | --- |
| 过滤表达式 | 用途 |
| http | 只查看HTTP协议的记录 |
| ip.src ==192.168.1.102 or ip.dst==192.168.1.102 | 源地址或者目标地址是192.168.1.102 |
|  |  |
|  |  |

**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。 你可以看到不同的协议用了不同的颜色显示。

你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

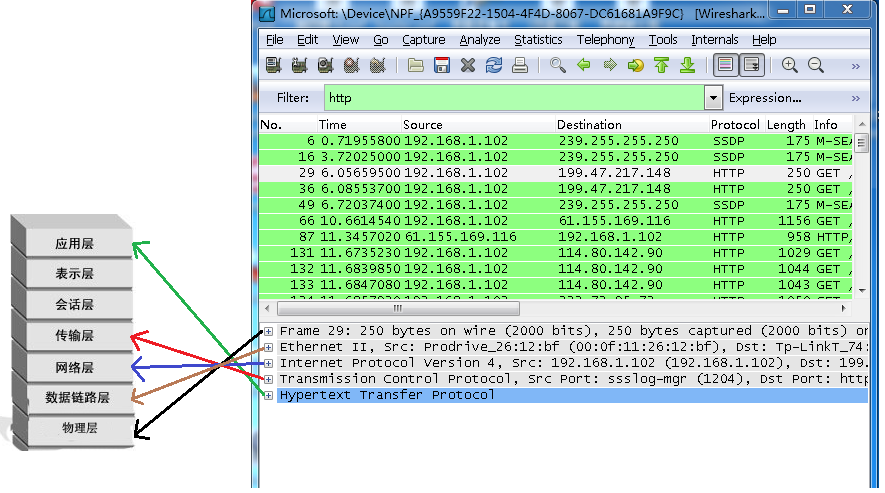
Ethernet II: 数据链路层以太网帧头部信息

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

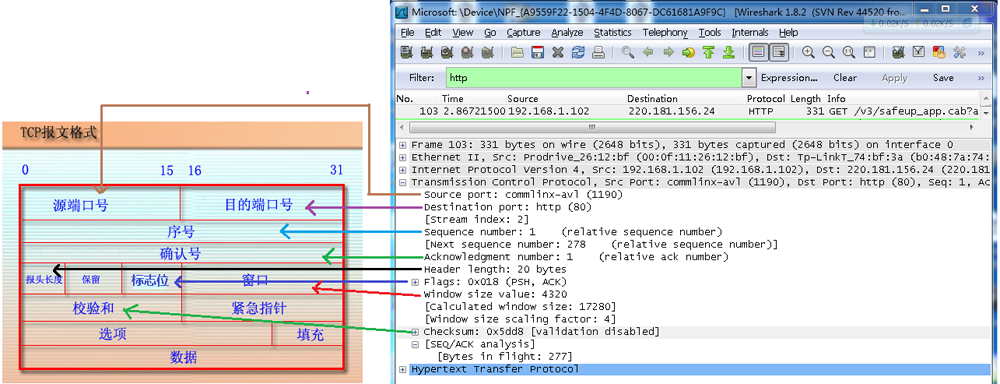
Hypertext Transfer Protocol:  应用层的信息，此处是HTTP协议

**wireshark与对应的OSI七层模型**



**TCP包的具体内容**

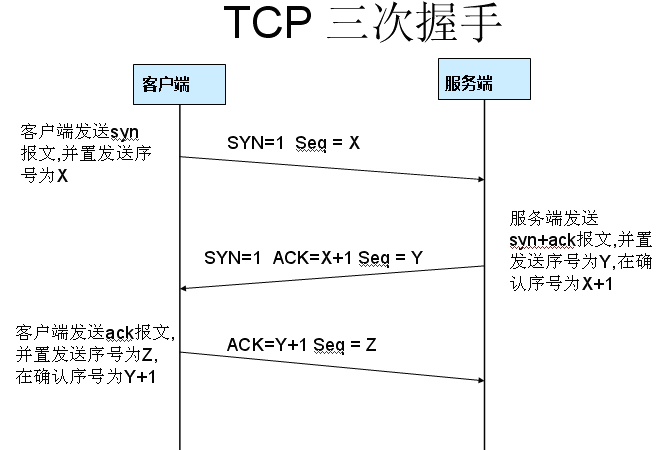
 从下图可以看到wireshark捕获到的TCP包中的每个字段。



**实例分析TCP三次握手过程**

看到这， 基本上对wireshak有了初步了解， 现在我们看一个TCP三次握手的实例

 三次握手过程为

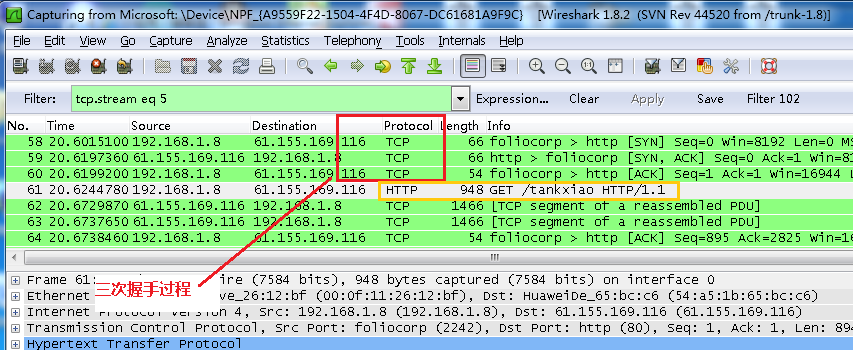


这图我都看过很多遍了， 这次我们用wireshark实际分析下三次握手的过程。

打开wireshark, 打开浏览器输入 http://www.cnblogs.com/tankxiao

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",

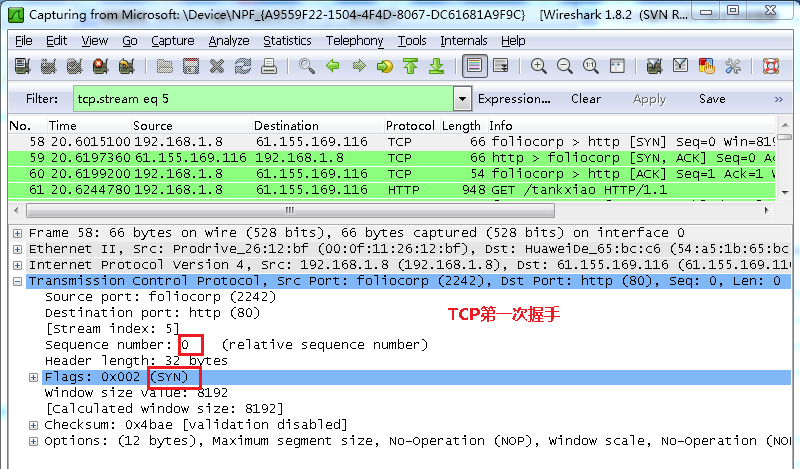
这样做的目的是为了得到与浏览器打开网站相关的数据包，将得到如下图



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。

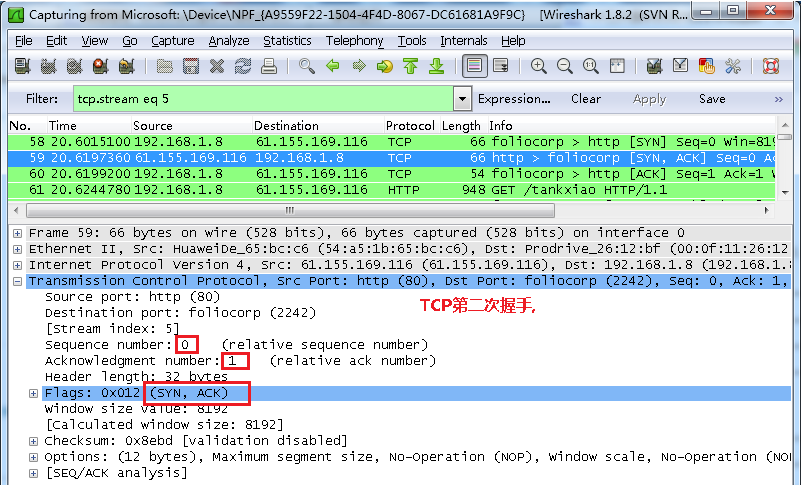
第一次握手数据包

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



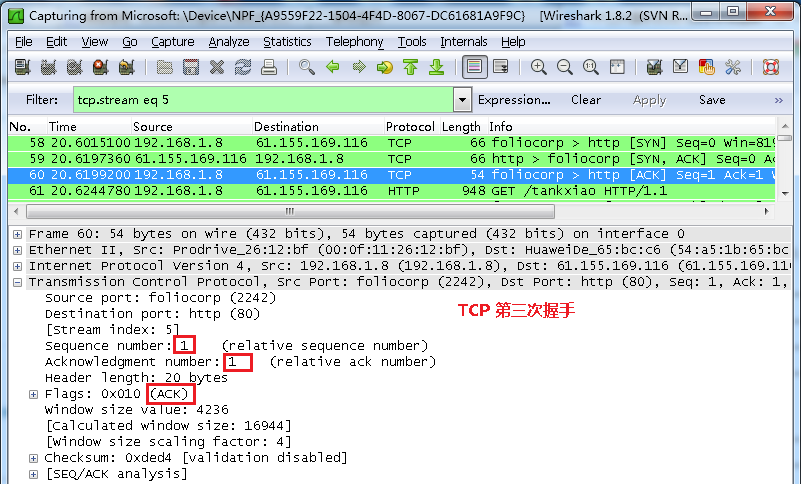
第二次握手的数据包

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

<http://www.cnblogs.com/TankXiao/archive/2012/10/10/2711777.html>

1.设置过滤条件

【1】http and ip.addr == 192.168.1.106 and tcp.port == 8080

【a】http：指定网络协议

【b】ip.addr == 192.168.1.106：指定服务器ip地址，请根据实际情况替换。

【c】tcp.port == 8080，指定端口号，请根据实际情况替换。

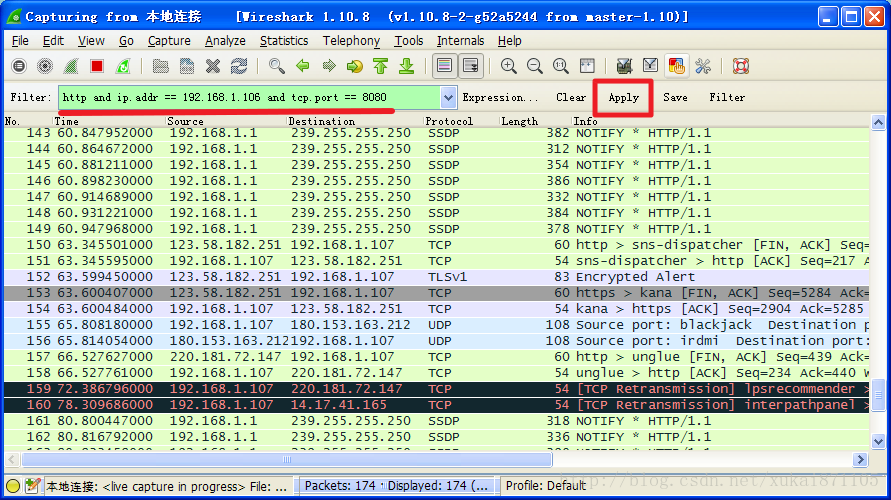


图1 过滤条件

【2】点击apply

    点击apply之后可过滤得到两个数据包，分别是HTTP请求和HTTP响应。

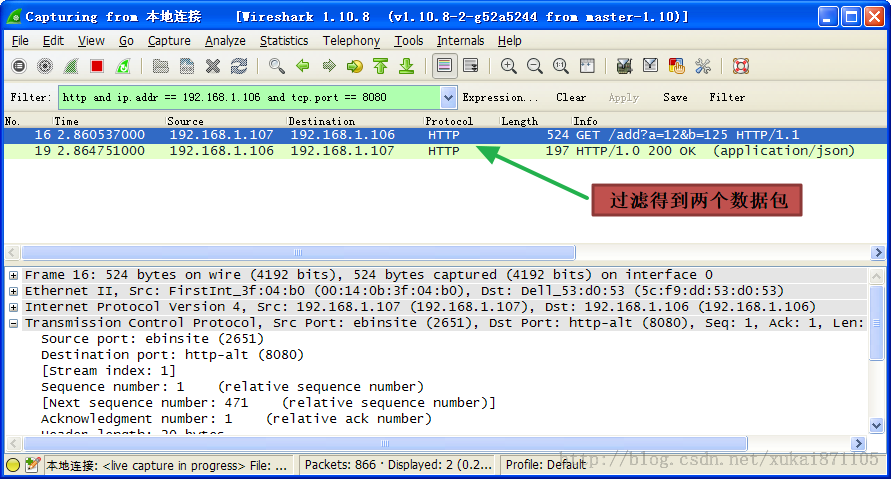


图2 HTTP请求和响应

2.查看TCP数据流——Follow TCP Stream

【1】在任意数据包上右击，选择Follow TCP Stream

    该步骤可以过滤出和该HTTP数据包有关的TCP数据包，包括TCP 3次握手，TCP分片和组装等。

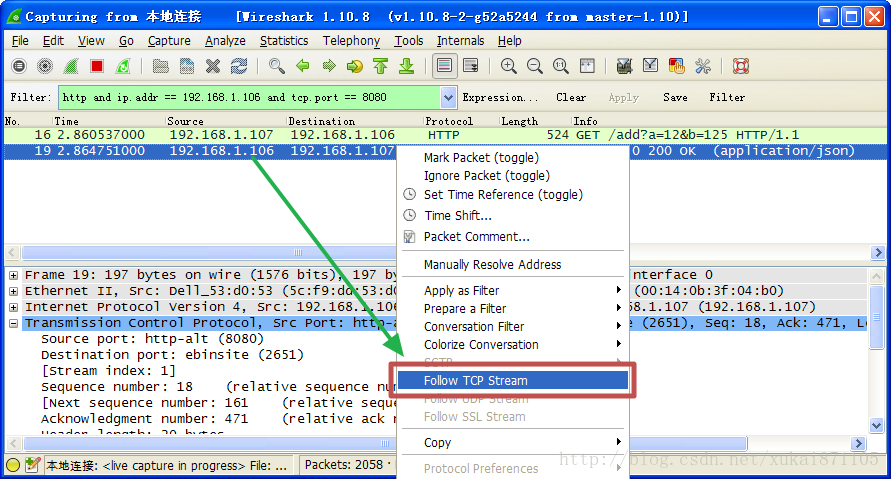


图3 选择Follow TCP Stream

【2】最终得到HTTP请求和响应

【a】红色背景字体为HTTP请求，蓝色背景字体为HTTP响应

【b】从User-Agent中可以看出，360浏览器兼容模式使用了IE8内核（该台计算机操作系统为XP，IE浏览器版本为8），这说明360浏览器使用了系统中的IE核。

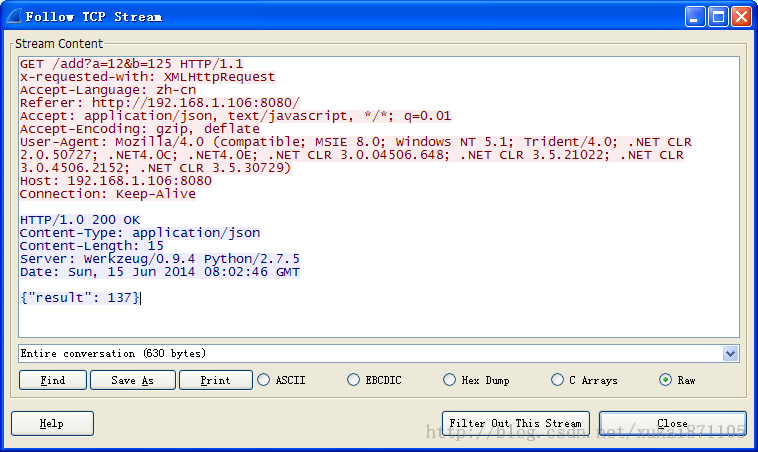


图4 HTTP请求和响应具体内容

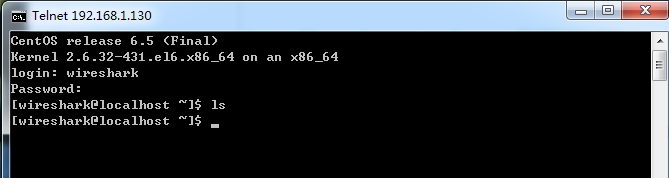
网络中明码传输的危险性

    通过明码传输的protocol和工具相当多，典型的就是telnet,ftp,http。我们拿telnet做这次实验。假设我以telnet方式登录到我的linux服务器，然后通过wireshark抓包，以抓取账号和密码信息。

1、首先启动wireshark，并处于Capture状态。然后通过telnet远程登录我们的linux服务器。

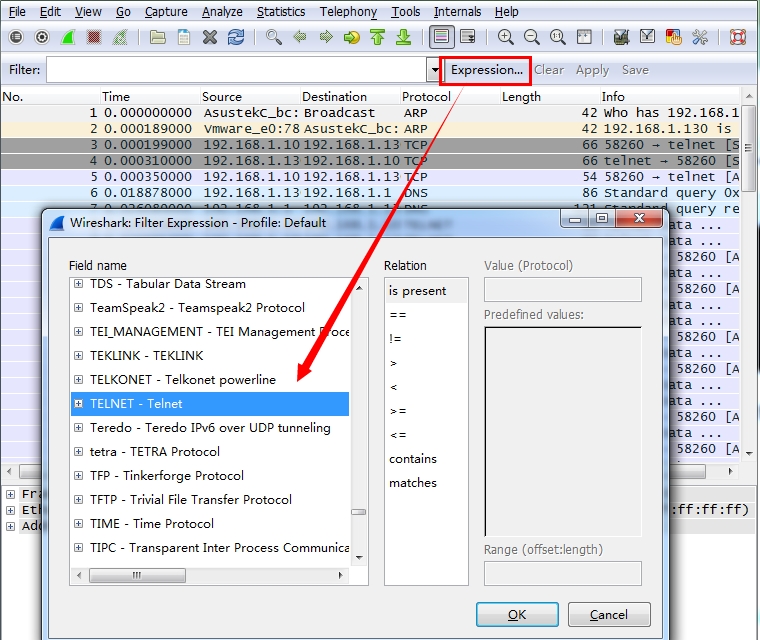
[](http://s3.51cto.com/wyfs02/M02/46/13/wKiom1PtxDOQimlAAACdwiZqfeE162.jpg)

进入登录界面后，输入账号和密码登入系统。

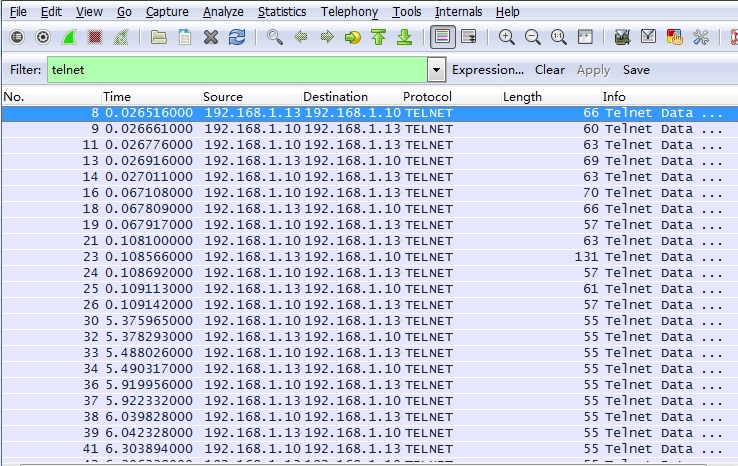
[](http://s3.51cto.com/wyfs02/M01/46/15/wKioL1PtxUvT0yksAACzkwdoI_Y013.jpg)

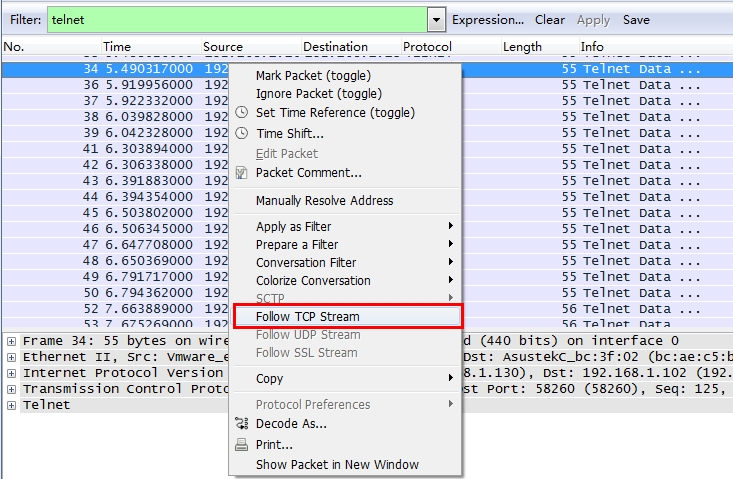
2、接下来停止wireshark的截取封包的操作，执行快捷方式的"Stop"即可。

    不过，捕获的信息非常多，这个时候可以利用Display Filter功能，过滤显示的内容，如下图所示，点击Expression,然后选择过滤表达式。这里，我们选择TELNET即可。

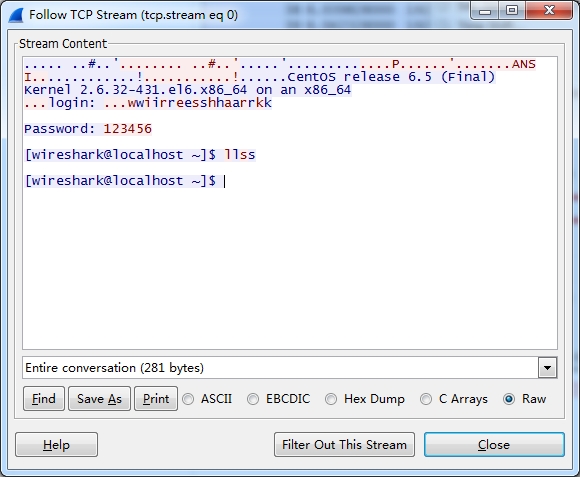
[](http://s3.51cto.com/wyfs02/M02/46/14/wKiom1PtxVXjVpp9AAU8cd7PEgE073.jpg)

表达式确定之后，选择"Apply",就可以过滤出只包含TELNET的封包

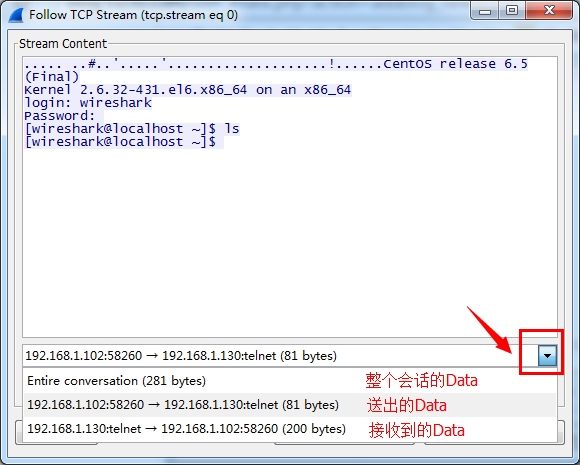
[](http://s3.51cto.com/wyfs02/M02/46/16/wKioL1Ptx42CAgvXAAYLelyRK58954.jpg)来，我们查看一下整个telnet会话的所有记录， wireshark可以记录会话记录（就像我们聊QQ时，"QQ聊天记录"一样），任意找到一个telnet封包，右键找到"Follow TCP Stream"，wireshark就会返回整个会话记录。

[](http://s3.51cto.com/wyfs02/M02/46/17/wKioL1PtyevSaTfJAASR__mwGZk771.jpg)

OK， 我们看到以下这些数据信息，红色的部分是我们发送出去的DATA，蓝色的部分是我们接收到的DATA。 [j_0061.gif](http://img.baidu.com/hi/jx2/j_0061.gif)， 告诉我， 你看到了什么

[](http://s3.51cto.com/wyfs02/M02/46/16/wKiom1PtyVriu__WAAINWJKx1S0346.jpg)

为了更准确的看清楚，我们再次仅筛选出我们发送出去的DATA。或者仅接收到的DATA。

[](http://s3.51cto.com/wyfs02/M02/46/18/wKioL1PtzA7D1_ewAAIm3rxMBZk668.jpg)

从这里，我们可以确切的抓到账号和密码信息。login:wireshark  Password:123456，除了这些，我们还可以更进一步知道别人在看什么网站，或是私人文件，隐私将毫无保障。

注：为了避免这些情况，防止有心人监测到重要信息，可以使用SSH,SSL,TSL,HTTPS等加密协议对重要数据进行加密，然后再到网络上传输，如果被人截取下来，看到的内容也是被加密的。

 <<http://skypegnu1.blog.51cto.com/8991766/1540728/>>

**常用到的过滤器规则：**

捕捉过滤器（CaptureFilters）：用于决定将什么样的信息记录在捕捉结果中。需要在开始捕捉前设置。  
显示过滤器（DisplayFilters）：在捕捉结果中进行详细查找。他们可以在得到捕捉结果后随意修改。

**捕捉过滤器**

Protocol（协议）:

可能的值: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp.

如果没有特别指明是什么协议，则默认使用所有支持的协议。

Direction（方向）:

可能的值: src, dst, src and dst, src or dst

如果没有特别指明来源或目的地，则默认使用 “src or dst” 作为关键字。

例如，”host 10.2.2.2″与”src or dst host 10.2.2.2″是一样的。

Host(s):

可能的值： net, port, host, portrange.

如果没有指定此值，则默认使用”host”关键字。

例如，”src 10.1.1.1″与”src host 10.1.1.1″相同。

Logical Operations（逻辑运算）:

可能的值：not, and, or.

否(“not”)具有最高的优先级。或(“or”)和与(“and”)具有相同的优先级，运算时从左至右进行。

例如，

“not tcp port 3128 and tcp port 23″与”(not tcp port 3128) and tcp port 23″相同。

“not tcp port 3128 and tcp port 23″与”not (tcp port 3128 and tcp port 23)”不同。

例子：

tcp dst port 3128  //捕捉目的TCP端口为3128的封包。

ip src host 10.1.1.1  //捕捉来源IP地址为10.1.1.1的封包。

host 10.1.2.3  //捕捉目的或来源IP地址为10.1.2.3的封包。

ether host e0-05-c5-44-b1-3c //捕捉目的或来源MAC地址为e0-05-c5-44-b1-3c的封包。如果你想抓本机与所有外网通讯的数据包时，可以将这里的mac地址换成路由的mac地址即可。

src portrange 2000-2500  //捕捉来源为UDP或TCP，并且端口号在2000至2500范围内的封包。

not imcp  //显示除了icmp以外的所有封包。（icmp通常被ping工具使用）

src host 10.7.2.12 and not dst net 10.200.0.0/16 //显示来源IP地址为10.7.2.12，但目的地不是10.200.0.0/16的封包。

(src host 10.4.1.12 or src net 10.6.0.0/16) and tcp dst portrange 200-10000 and dst net 10.0.0.0/8  //捕捉来源IP为10.4.1.12或者来源网络为10.6.0.0/16，目的地TCP端口号在200至10000之间，并且目的位于网络 10.0.0.0/8内的所有封包。

src net 192.168.0.0/24

src net 192.168.0.0 mask 255.255.255.0  //捕捉源地址为192.168.0.0网络内的所有封包。

**显示过滤器**

例子：

snmp || dns || icmp //显示SNMP或DNS或ICMP封包。

ip.addr == 10.1.1.1  //显示来源或目的IP地址为10.1.1.1的封包。

ip.src != 10.1.2.3 or ip.dst != 10.4.5.6  //显示来源不为10.1.2.3或者目的不为10.4.5.6的封包。

换句话说，显示的封包将会为：

来源IP：除了10.1.2.3以外任意；目的IP：任意

以及

来源IP：任意；目的IP：除了10.4.5.6以外任意

ip.src != 10.1.2.3 and ip.dst != 10.4.5.6  //显示来源不为10.1.2.3并且目的IP不为10.4.5.6的封包。

换句话说，显示的封包将会为：

来源IP：除了10.1.2.3以外任意；同时须满足，目的IP：除了10.4.5.6以外任意

tcp.port == 25  //显示来源或目的TCP端口号为25的封包。

tcp.dstport == 25  //显示目的TCP端口号为25的封包。

tcp.flags  //显示包含TCP标志的封包。

tcp.flags.syn == 0×02  //显示包含TCP SYN标志的封包。

如果过滤器的语法是正确的，表达式的背景呈绿色。如果呈红色，说明表达式有误。