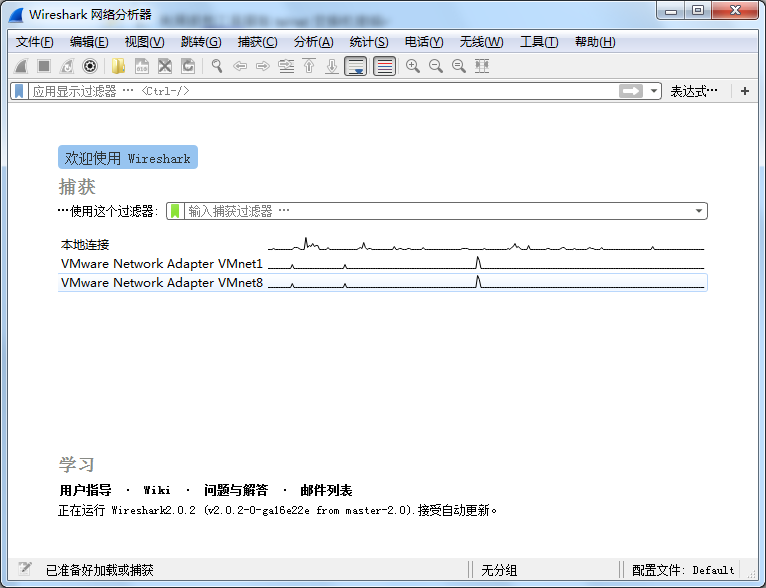
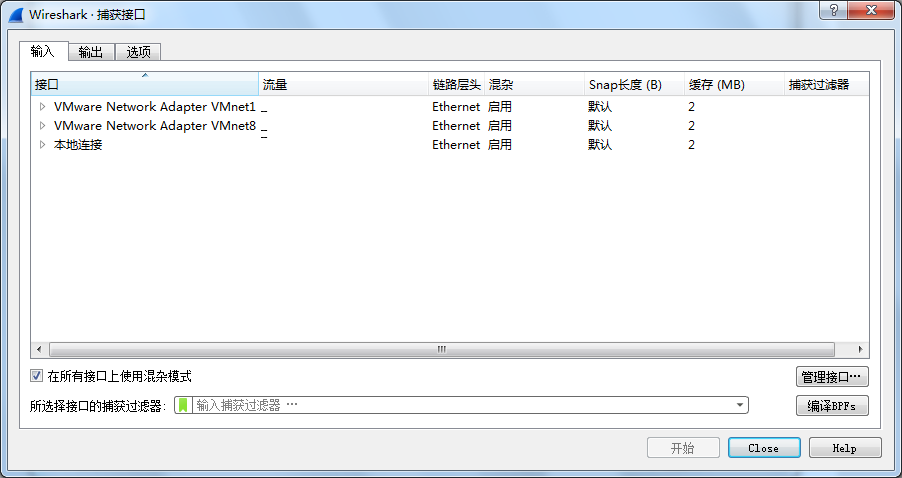
1. **利用抓包工具获取telnet交换机密码**

首先，打开Wireshark软件，



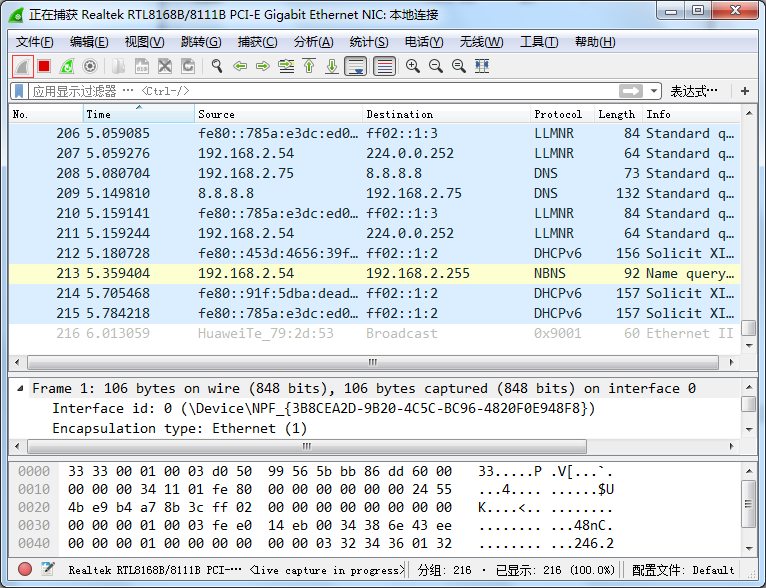
打开之后选择捕获选项，



这里会让选择是捕获哪个端口的信息，由于我机子装有虚拟机，所以会有两个虚拟网卡的接口，我们这里选择本地连接的接口。

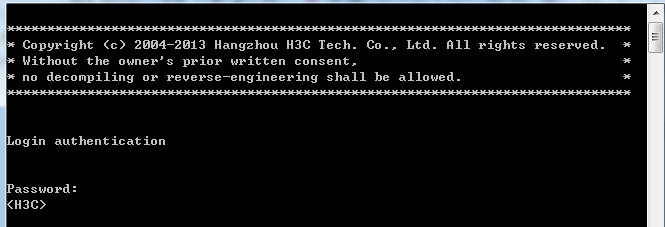
这里拓展一下，接口可以通过右下角的管理接口，选择或者取消掉不想要的接口，还可以在输出选项卡中，选择输出的文件路径，以及输出格式等其他选项，在选项的选项卡中，主要设置显示的信息，以及解析名称，及自动停止捕获等设置。

设置好之后点击开始，就可以开始抓包了

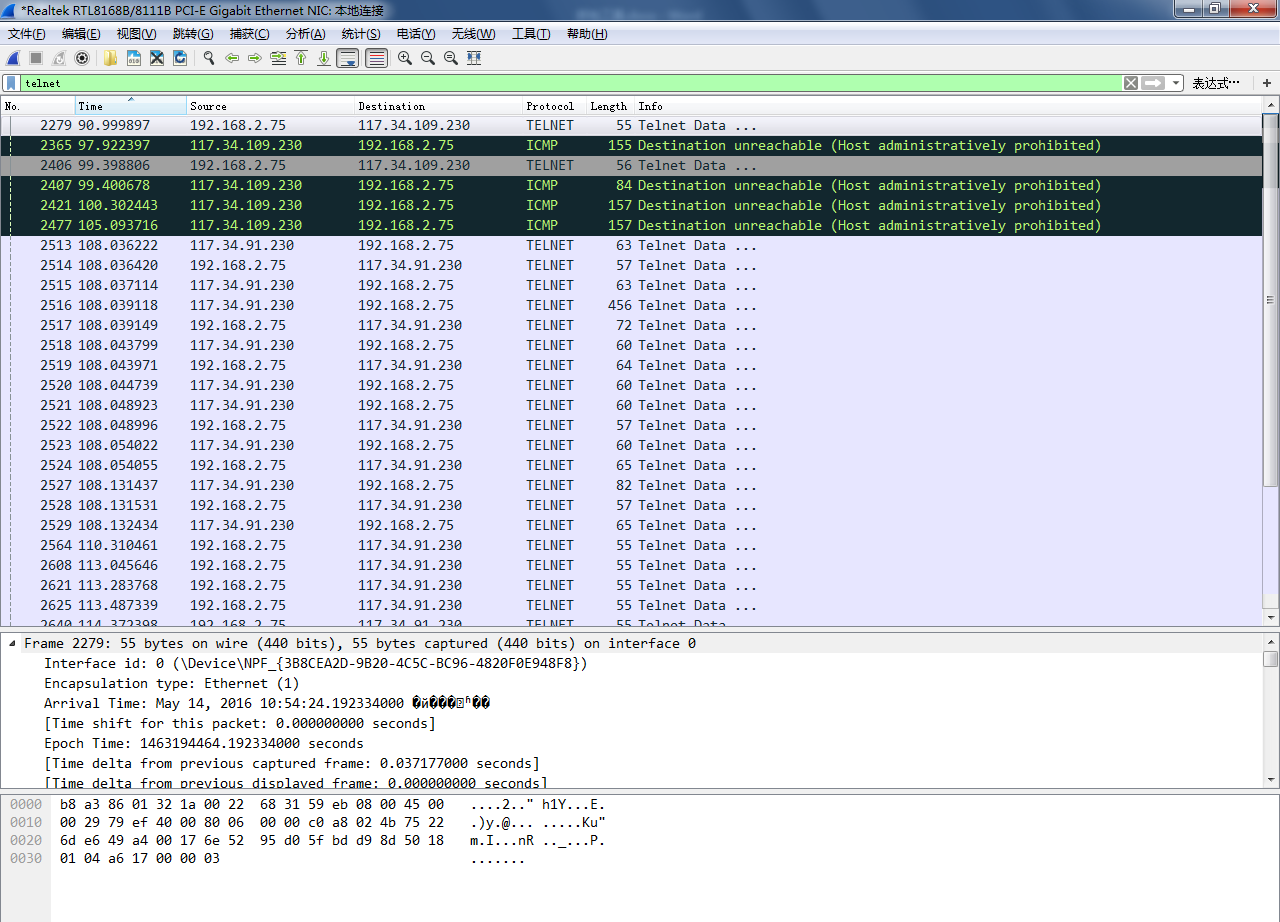


开始telnet登录交换机

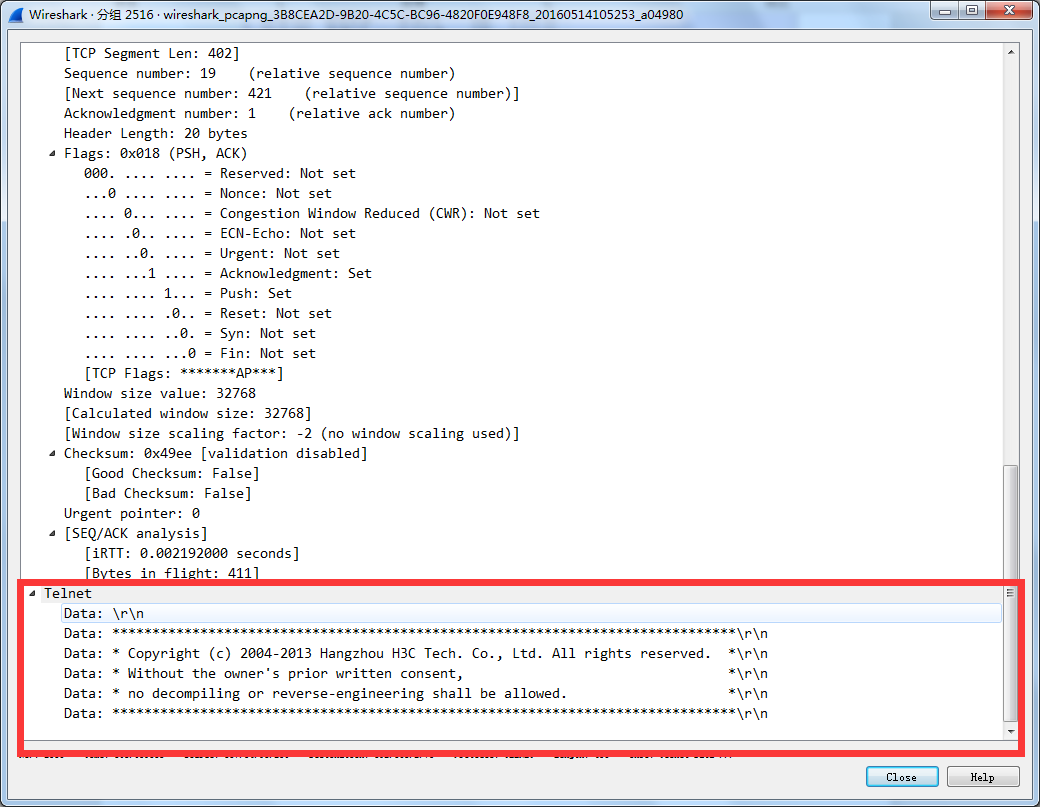
开始——运行——cmd，telnet + IP，输入正确的用户名密码登录



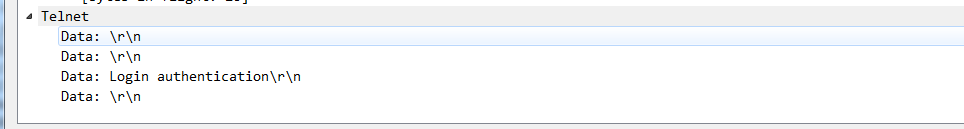
然后切换到抓包工具，停止抓包，在过滤工具栏中输入telnet，表示过滤telnet协议的包，



这里可以看到，抓包的信息包括，时间，源IP地址，目标IP地址，端口/协议，数据包长度，数据包信息，我们可以从数据包信息这列看，telnet dada的数据包，就是包含有telnet登录信息的数据包，我们可以通过双击进入该数据包，查看数据包详细信息。



我们可以看到，红框的地方，是输出的我们登录交换机的交换机回传数据，接着往下，



这里提示登录验证，



这里提示输入登录密码，



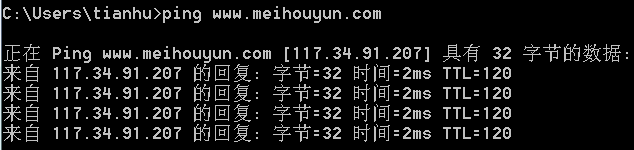
从这个包开始往下，每个包会携带密码信息的1到2个字段，这样，就可以获取到完整的交换机登录用户名密码。

1. **如何获取网站登录信息**

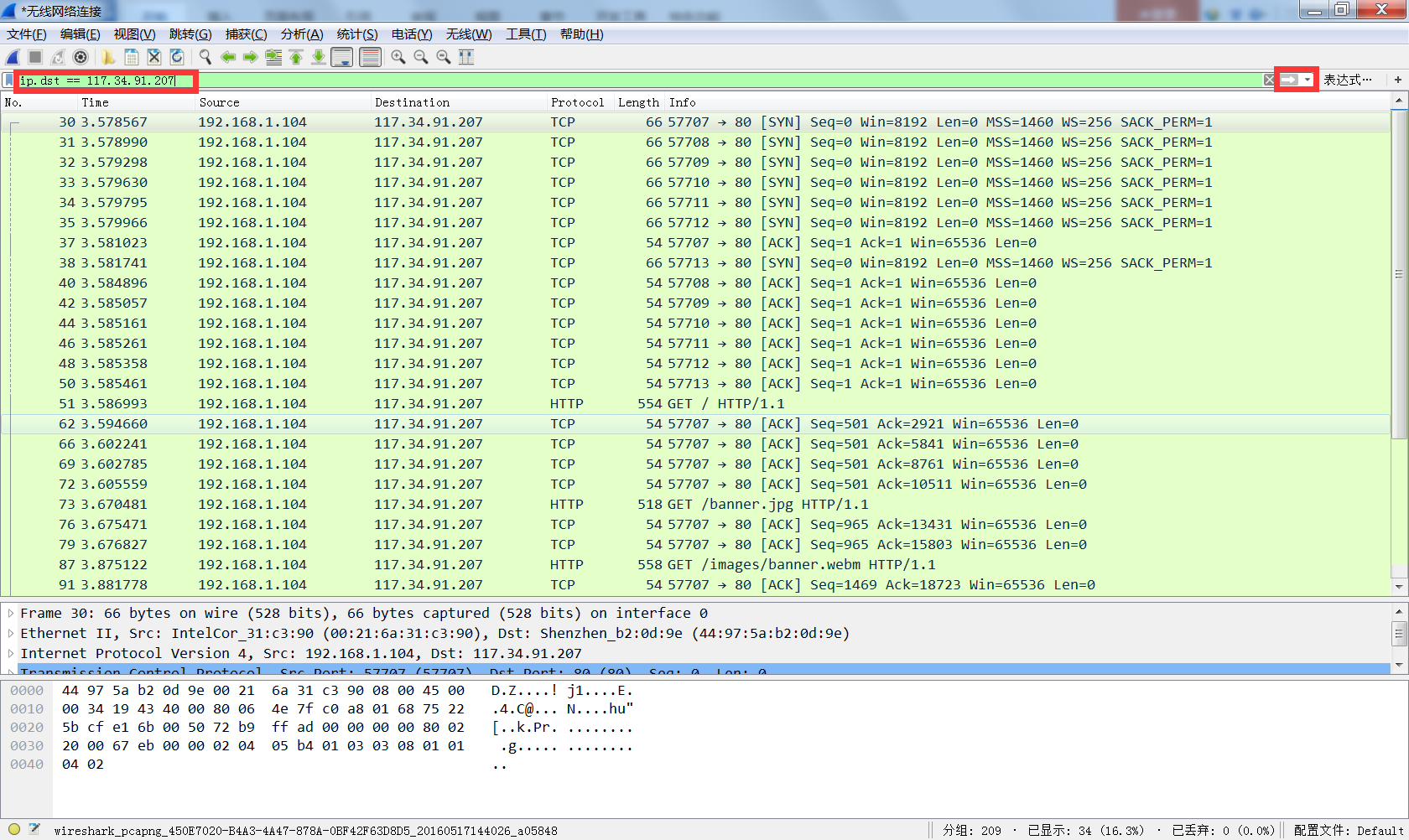
首先，打开抓包软件，和上面步骤一样，选好连接外网的网卡，之后开始抓包，然后打开要登录的网站，输入登录信息，点击登录



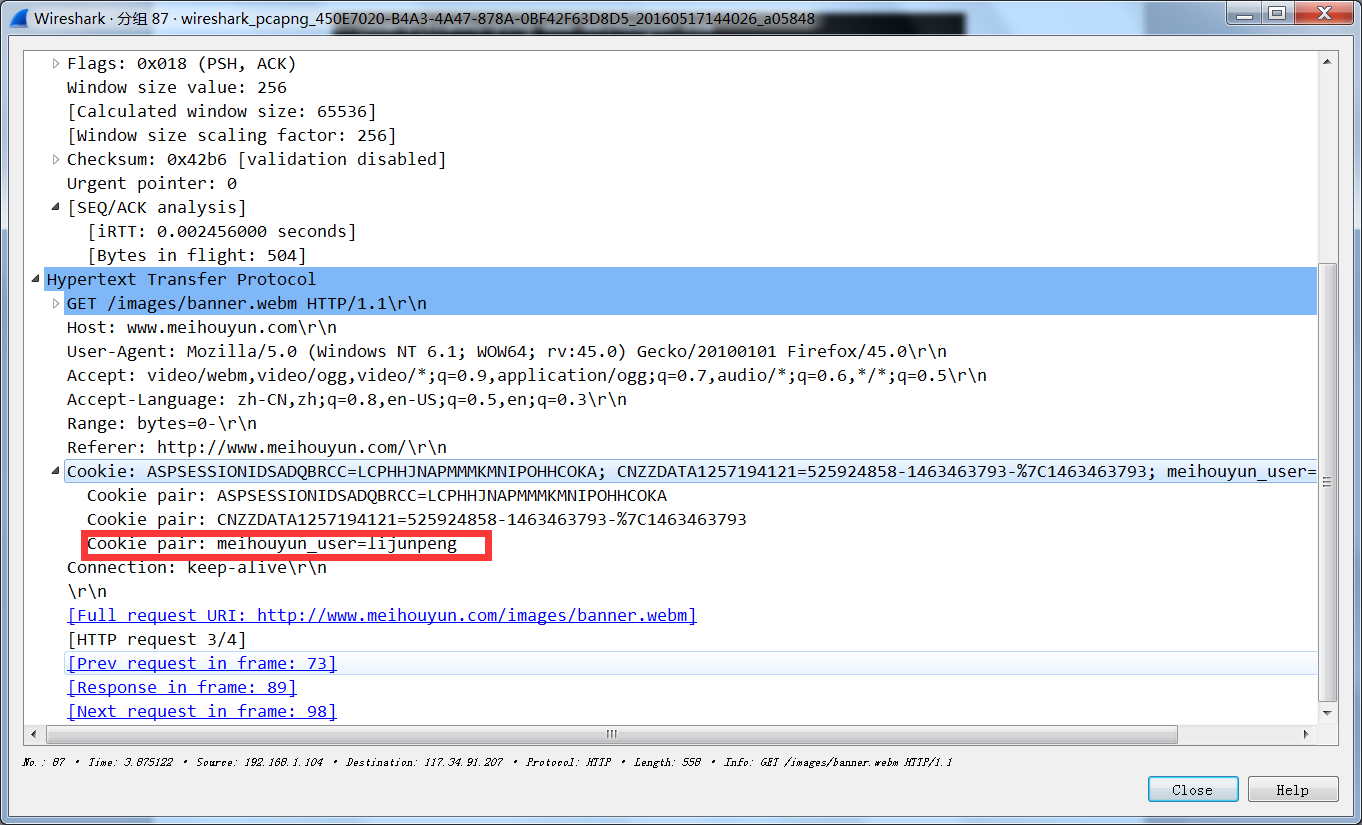
登录之后停止抓包，解析网站域名绑定的IP



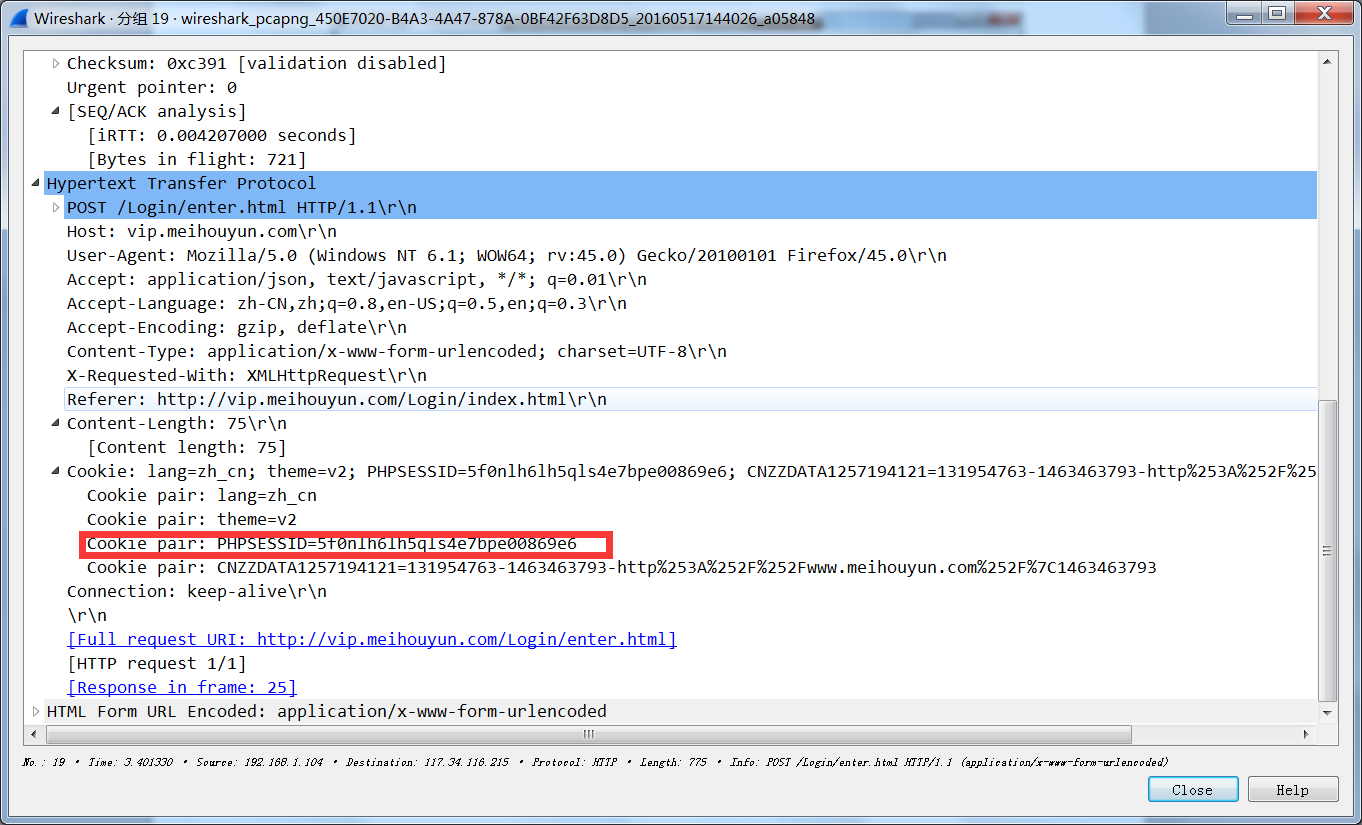
解析出IP之后，可以在Wireshark上过滤目标IP，因为平常情况下上网的时候数据包很多，要从中找出想要的数据包就需要用表达式过滤想要的包，



从其中找到HTTP协议的get请求，双击get请求数据包，可以在Cookie中看到登录的用户名



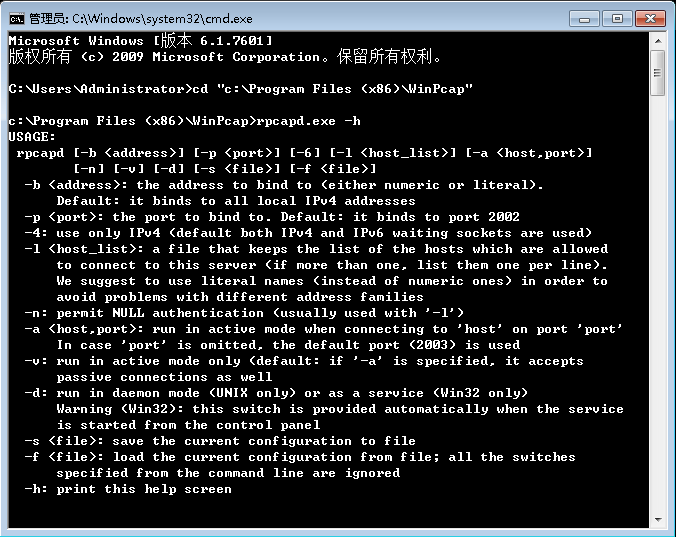
这里获取到用户名之后，我们再次过滤抓取到的数据包，一般的网站都是POST方法提交密码的，所以我们过滤post包，



这里我们可以看到密码的HASH值，通过HASH解密，就可以破解出网站的密码。

**3、如何利用Wireshark远程抓包：**

很多时候我们需要抓取远程端的数据包来分析远程端网络情况，首先需要在我们要被抓包的remote端安装WinPcap，WinPcap是一个网络底层的驱动，尽量找到和你这边版本一致的版本进行安装，默认路径安装，然后再remoted端进入命令行，切换到WinPcap路径下，如图：



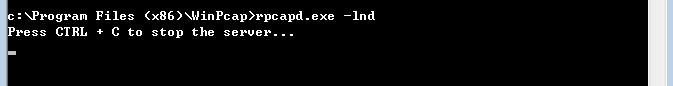
通过rpcapd -h可以查看各个参数的用法，我们常用的有3个参数，

**-l 允许远端可以连接本机，此参数必须**

**-d 允许rpcapd服务以daemon的方式运行，此参数必须**

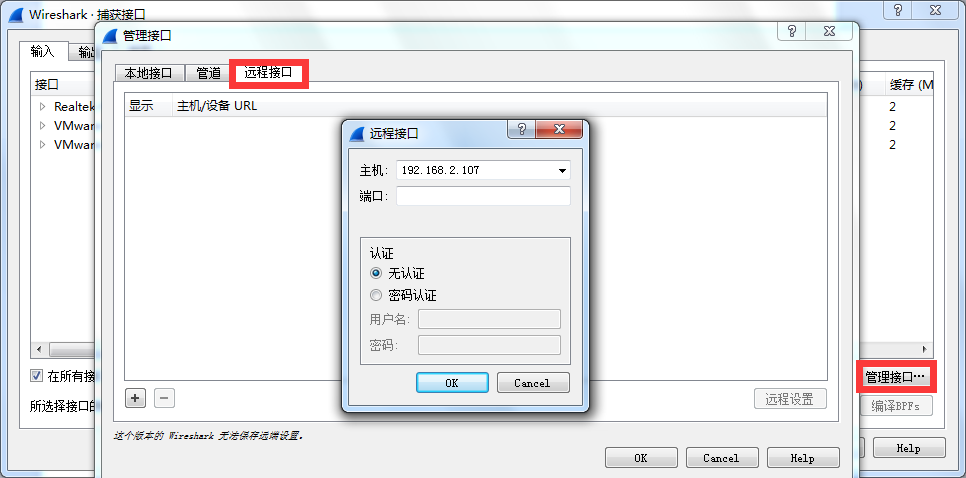
**-n 此参数加上可以让远程连接抓包时不需要输入用户名和密码**

所以在remote端我们允许rpcapd -lnd，



这样远端就设置好了，我们开始在本地远程抓取数据包，

首先我们打开Wireshark，设置捕获选项，点击管理接口，选择远程接口选项卡，

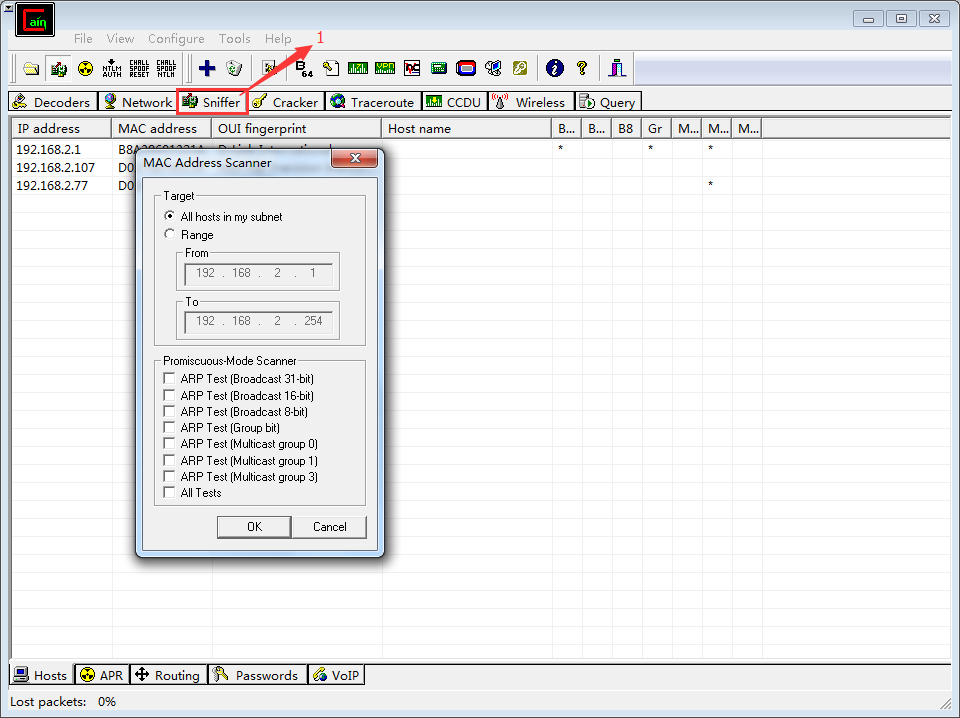


点击OK，就可以连接到远端网卡，选择远端网卡，就可以开始抓包了。

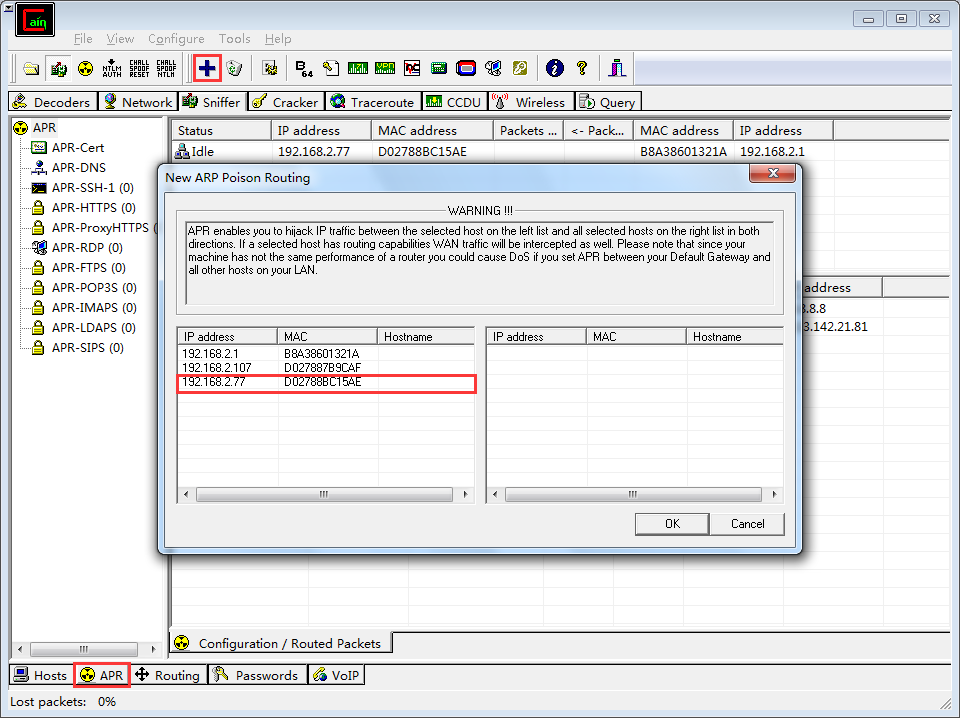
**4、如何通过Wireshark抓包判断ARP欺骗**

首先我们模拟arp攻击，

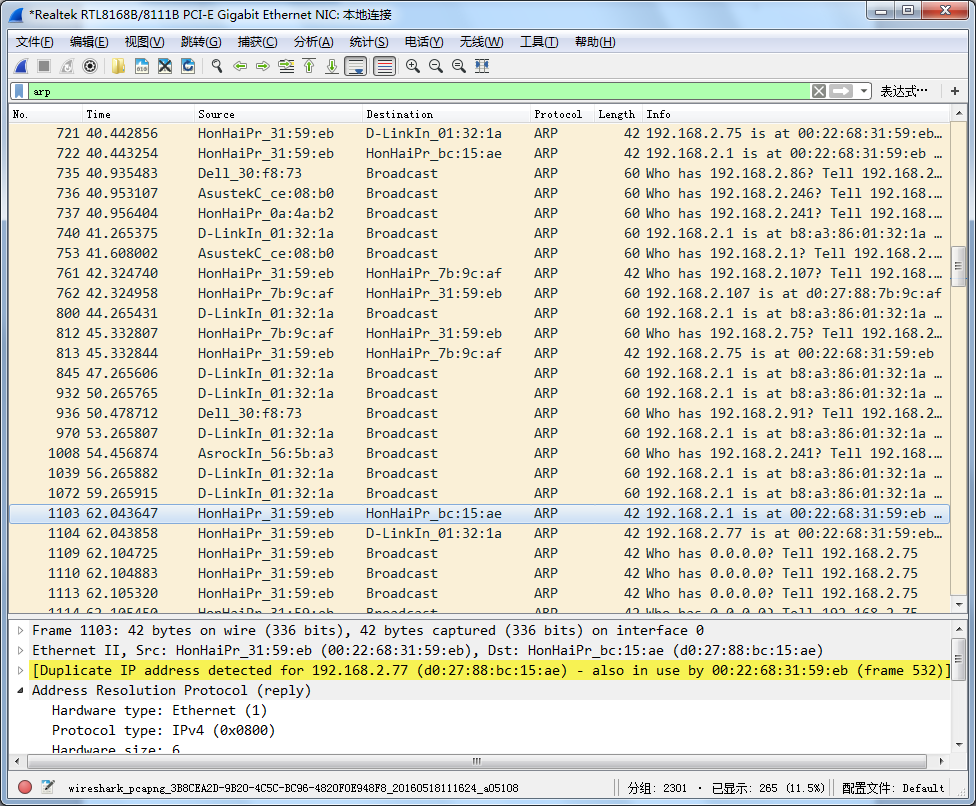
我们使用的工具是cain，软件安装很简单，为了不影响软件正常运行，在开始使用前退出所有杀毒软件。打开cain软件，



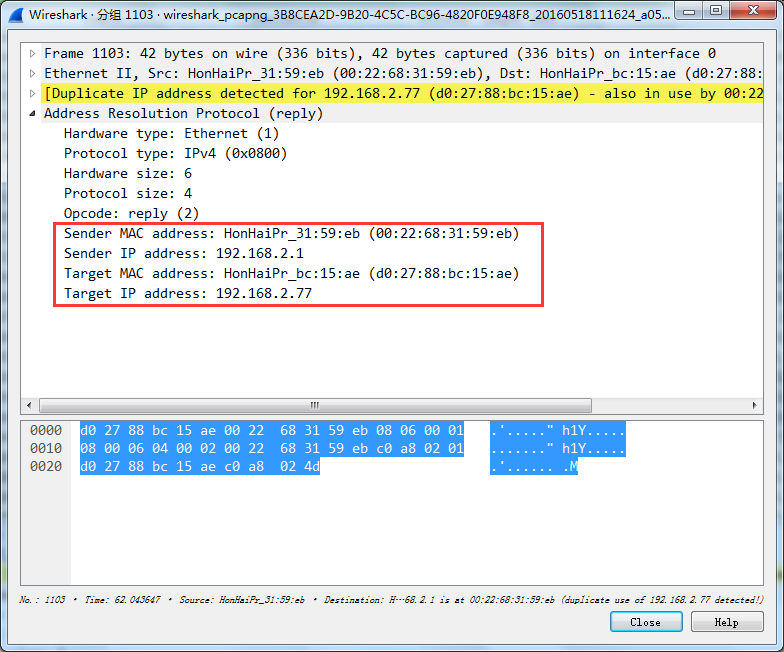
选择Sniffer，在空白处点击右键选择Scan MAC address，然后可以选择整段扫描或者是扫描某一段IP，选好之后点击OK，就会开始扫描，扫描完之后会在列表中显示扫描的详细结果，然后选择下面的ARP选项卡，选择上面的加号，在弹出来的对话框左侧选择要攻击的目标，在右侧窗口中会出现该攻击目标的上游IP，选择，这样就设置好了。



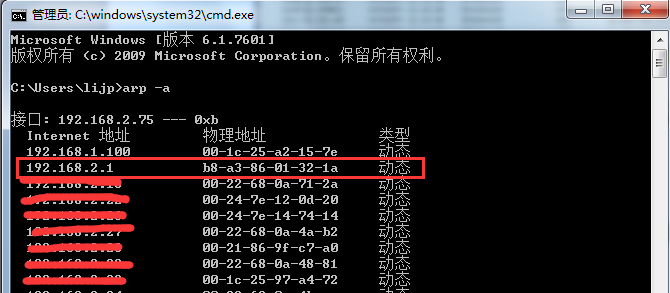
点击工具栏中的放射性图标，即ARP攻击开始/停止按钮，开始发起ARP攻击，这时查看Wireshark，



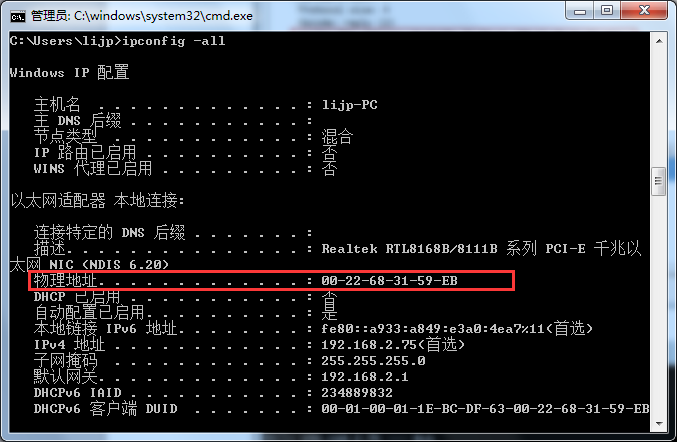
发现很多ARP包，详细查看每个包的详细信息，



本地查看真实的网关MAC



从抓到的数据包中分析看到向被攻击目标发送的欺骗MAC与真实网关MAC不一样，

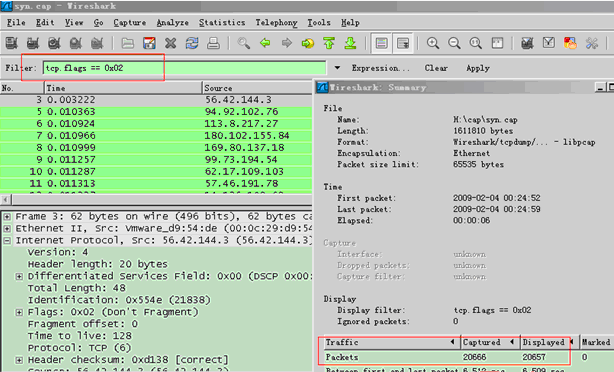


看一下发起攻击端MAC地址，正好是欺骗MAC,这就是ARP的欺骗。

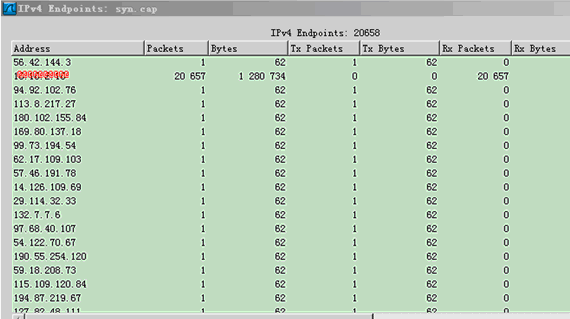
1. 利用Wireshark简单分析DDOS攻击数据包

SYN Flood

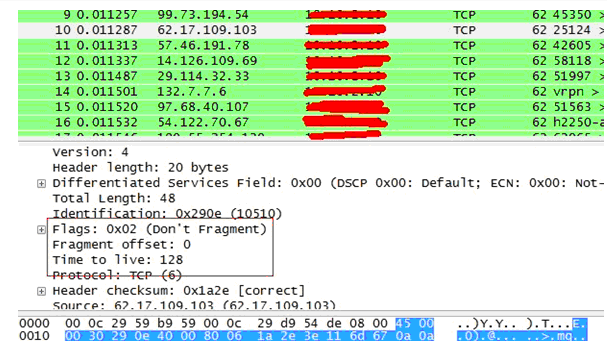
使用wireshark 过滤器tcp.flags==0x02 过滤检查数据包分布情况。如图所示，SYN Flood攻击发生时数据包分布发生明显改变，Syn包比例明显增加。



使用wireshark static->ipv4->endpoint分析数据包源地址分布。当使用伪造IP地址的DDoS攻击发生时，抓包文件中的数据包数目和源地址对应关系会发生明显变化。从图中实例可以发现，除了被攻击的目的IP意外，wireshark统计每个源地址对应的数据包数目较小，数据包大小字节数（Bytes）几乎一致。

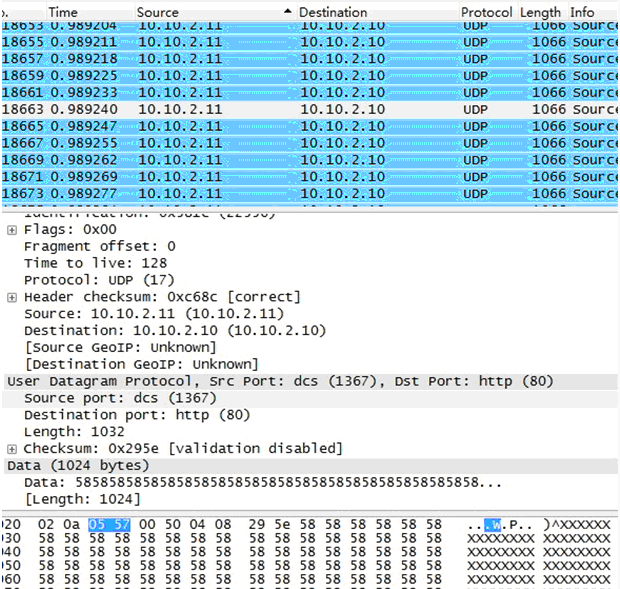


当使用随机源进行DDoS攻击时，虽然使用了伪造源地址进行攻击，但攻击者无法伪造攻击主机与目标主机之间的位置关系。下图所示的这个攻击程序并没有修改攻击数据包的TTL值，所有的攻击数据包使用相同的TTL值。所以，这个攻击数据包是由局域网内的一个windows计算机发出的。



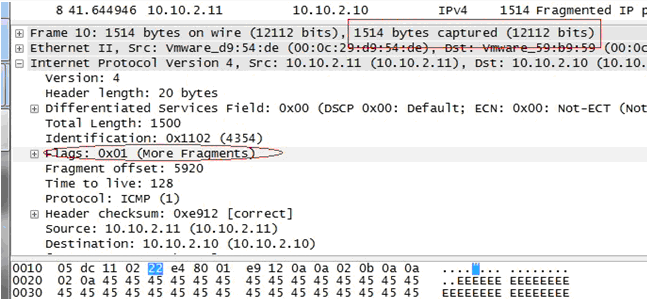
UDP FLOOD

UDP FLOOD攻击的主要目的是通过发送大量的UDP数据包来堵塞服务器的带宽。



ICMP FLOOD

与UDP FLOOD相同，ICMP FLOOD主要以阻塞服务器带宽为主。但与UDP FLOOD不同的是，ICMP通常不会承载数据业务，比较容易通过交换机ACL或者服务器的iptables等进行防御。同时，攻击实施者为了更加有效的利用手上的僵尸主机，会使用包大小较大的数据包和IP层分片的数据包进行攻击，这种攻击会绕过没有设置IP层分配的ACL，也会加重服务器分配重组的负担。



通过图中可以看到数据包大小为1514Byte，且使用了分片标识（Flags）。

以上都是常见的DDOS攻击数据包，通过分析可以有效的降低危害或减少攻击影响。

其他的抓包工具有snoop、tcpdump、fiddler等，每个工具的侧重有所不同，可以综合利用，达到想要的效果。

**附录**

**WireShark 过滤 语法**  
1. 过 滤 IP，如来源IP或者目标IP等于某个IP   
例子:   
ip.src eq 192.168.1.107 or ip.dst eq 192.168.1.107   
或者   
ip.addr eq 192.168.1.107 // 都能显示来源IP和目标IP   
  
2. 过滤 端 口   
例子:   
tcp.port eq 80 // 不管端口是来源的还是目标的都显示   
tcp.port == 80   
tcp.port eq 2722   
tcp.port eq 80 or udp.port eq 80   
tcp.dstport == 80 // 只显tcp协议的目标端口80   
tcp.srcport == 80 // 只显tcp协议的来源端口80   
udp.port eq 15000   
过滤 端口范围   
tcp.port >= 1 and tcp.port <= 80   
  
3. 过 滤 协议   
例子:   
tcp   
udp   
arp   
icmp   
http   
smtp   
ftp   
dns   
msnms   
ip   
ssl   
oicq   
bootp   
等 等   
排除arp包，如!arp  或者  not arp   
  
4. 过 滤 MAC   
太以网头 过滤   
eth.dst == A0:00:00:04:C5:84 // 过滤 目 标mac   
eth.src eq A0:00:00:04:C5:84 // 过 滤 来源mac   
eth.dst==A0:00:00:04:C5:84   
eth.dst==A0-00-00-04-C5-84   
eth.addr eq A0:00:00:04:C5:84 // 过滤 来 源MAC和目标MAC都等于A0:00:00:04:C5:84的   
less than 小于 < lt   
小于等于 le   
等 于 eq   
大于 gt   
大于等于 ge   
不等 ne   
  
5.包长度 过 滤   
例子:   
udp.length == 26 这个长度是指udp本身固定长度8加上udp下面那块数据包之和   
tcp.len >= 7  指的是ip数据包(tcp下面那块数据),不包括tcp本身   
ip.len == 94 除了以太网头固定长度14,其它都算是ip.len,即从ip本身到最后   
frame.len == 119 整个数据包长度,从eth开始到最后   
eth ---> ip or arp ---> tcp or udp ---> data   
  
6.http 模式 过滤   
例子:   
http.request.method == "GET"   
http.request.method == "POST"   
http.request.uri == "/img/logo-edu.gif"   
http contains "GET"   
http contains "HTTP/1."   
// GET包   
http.request.method == "GET" && http contains "Host: "   
http.request.method == "GET" && http contains "User-Agent: "   
// POST包   
http.request.method == "POST" && http contains "Host: "   
http.request.method == "POST" && http contains "User-Agent: "   
// 响应包   
http contains "HTTP/1.1 200 OK" && http contains "Content-Type: "   
http contains "HTTP/1.0 200 OK" && http contains "Content-Type: "   
一 定包含如下   
Content-Type:   
  
7.TCP参数 过 滤   
tcp.flags 显示包含TCP标志的封包。   
tcp.flags.syn == 0x02    显示包含TCP SYN标志的封包。   
tcp.window\_size == 0 && tcp.flags.reset != 1   
  
8. 过滤 内容   
tcp[20] 表示从20开始，取1个字符   
tcp[20:]表示从20开始，取1个字符以上   
tcp[20:8]表示从20开始，取8个字符   
tcp[offset,n]   
udp[8:3]==81:60:03 // 偏移8个bytes,再取3个数，是否与==后面的数据相等？   
udp[8:1]==32  如果我猜的没有错的话，应该是udp[offset:截取个数]=nValue   
eth.addr[0:3]==00:06:5B   
例 子:   
判断upd下面那块数据包前三个是否等于0x20 0x21 0x22   
我们都知道udp固定长度为8   
udp[8:3]==20:21:22   
判 断tcp那块数据包前三个是否等于0x20 0x21 0x22   
tcp一般情况下，长度为20,但也有不是20的时候   
tcp[8:3]==20:21:22   
如 果想得到最准确的，应该先知道tcp长度   
matches(匹配)和contains(包含某字符串)语法   
ip.src==192.168.1.107 and udp[8:5] matches "\\x02\\x12\\x21\\x00\\x22"   
ip.src==192.168.1.107 and udp contains 02:12:21:00:22   
ip.src==192.168.1.107 and tcp contains "GET"   
udp contains 7c:7c:7d:7d 匹配payload中含有0x7c7c7d7d的UDP数据包，不一定是从第一字节匹配。   
例子:   
得到本地qq登陆数据包(判断条 件是第一个包==0x02,第四和第五个包等于0x00x22,最后一个包等于0x03)   
0x02 xx xx 0x00 0x22 ... 0x03   
正确   
oicq and udp[8:] matches "^\\x02[\\x00-\\xff][\\x00-\\xff]\\x00\\x22[\\x00-\\xff]+\\x03$"   
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$" // 登陆包   
oicq and (udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x03$" or tcp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x03$")   
oicq and (udp[8:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$" or tcp[20:] matches "^\\x02[\\x00-\\xff]{2}\\x00\\x22[\\x00-\\xff]+\\x03$")   
不 单单是00:22才有QQ号码,其它的包也有,要满足下面条件(tcp也有，但没有做):   
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]+\\x03$" and !(udp[11:2]==00:00) and !(udp[11:2]==00:80)   
oicq and udp[8:] matches "^\\x02[\\x00-\\xff]+\\x03$" and !(udp[11:2]==00:00) and !(udp[15:4]==00:00:00:00)   
说明:   
udp[15:4]==00:00:00:00 表示QQ号码为空   
udp[11:2]==00:00 表示命令编号为00:00   
udp[11:2]==00:80 表示命令编号为00:80   
当命令编号为00:80时，QQ号码为 00:00:00:00   
得到msn登陆成功账号(判断条件是"USR 7 OK ",即前三个等于USR，再通过两个0x20，就到OK,OK后面是一个字符0x20,后面就是mail了)   
USR xx OK mail@hotmail.com   
正确   
msnms and tcp and ip.addr==192.168.1.107 and tcp[20:] matches "^USR\\x20[\\x30-\\x39]+\\x20OK\\x20[\\x00-\\xff]+"

9.DHCP   
以 寻找伪造DHCP服务器为例，介绍 Wireshark 的用法。在显 示 过滤 器中加入 过 滤 规则，   
显示所有非来自DHCP服务器并且bootp.type==0x02（Offer/Ack）的信息：   
bootp.type==0x02 and not ip.src==192.168.1.1