项目 1

网络封包分析工具 Wireshark

视频讲解

1.1 Wireshark 简介

Wireshark 是目前流行的网络封包分析工具，可以帮助我们获得网络连接中的各项数据。以前上网、访问网页对于人们来说，只是一个抽象的概念，我们并不知道到底是如何浏览那些网络信息的，而利用 Wireshark 可以将这些概念实体化，各项数据直观地展现了网络连接、网页访问的全过程。这个强大的工具可以捕捉网络中的数据，并为用户提供关于网络和上层协议的各种信息。与其他很多网络工具一样，Wireshark 也使用 Npcap 来进行封包捕捉，并可破解局域网内 QQ、邮箱、MSN 等账号密码。

网络管理员使用 Wireshark 来检测网络问题，网络安全工程师使用 Wireshark 来检查安全的相关问题，开发者使用 Wireshark 来为新的通信协议找错，普通使用者使用 Wireshark 来学习网络协议的相关知识。当然，有的人也会“居心叵测”地用它来寻找一些敏感信息。

1.2 Wireshark 工作流程

(1) 确定 Wireshark 的位置。如果没有一个正确的位置，启动 Wireshark 后会花费很长的时间捕获一些与自己无关的数据。

(2) 选择捕获接口。一般都是选择连接到 Internet 网络的接口，这样才可以捕获到与网络相关的数据；否则，捕获到的其他数据对自己也没有任何帮助。

(3) 使用捕获过滤器。通过设置捕获过滤器，可以避免产生过大的捕获文件。这样用户在分析数据时，也不会受其他数据的干扰，而且，还可以为用户节约大量的时间。

(4) 使用显示过滤器。通常使用捕获过滤器过滤后的数据，往往还是很复杂。为了使过滤的数据包更细致，此时使用显示过滤器进行过滤。

(5) 使用着色规则。通常使用显示过滤器过滤后的数据，都是有用的数据包。如果想更加突出地显示某个会话，可以使用着色规则高亮显示。

(6) 构建图表。如果用户想要更明显地看出一个网络中数据的变化情况，使用图表的形式可以很方便地展现数据分布的情况。

(7) 重组数据。Wireshark 的重组功能，可以重组一个会话中不同数据包的信息，或者是重组一个完整的图片或文件。由于传输的文件往往较大，所以信息分布在多个数据

包中。为了能够查看到整个图片或文件,需要使用重组数据的方法来实现。

1.3 Wireshark 安装

可到官网下载 Wireshark,网址为 https://www.wireshark.org/download.html,网站界面如图 1.1 所示。

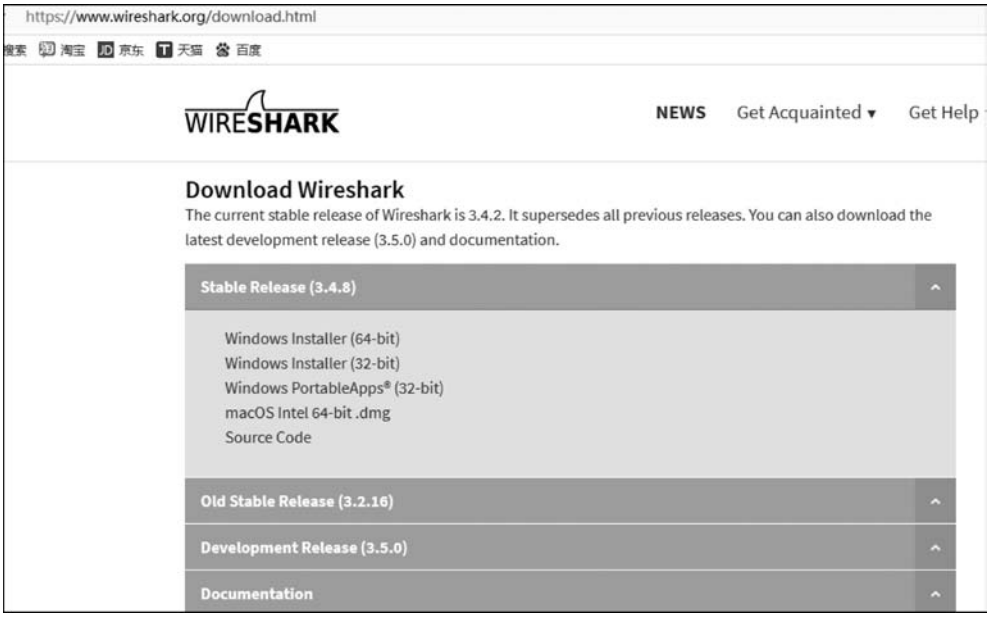


图 1.1 打开 Wireshark 的官网

选择 3.4.2 版本,单击 Windows Installer(64-bit)下载,可以下载到任意目录,下载的文件为 Wireshark-win64-3.4.2.exe,执行该文件,显示结果如图 1.2 所示。

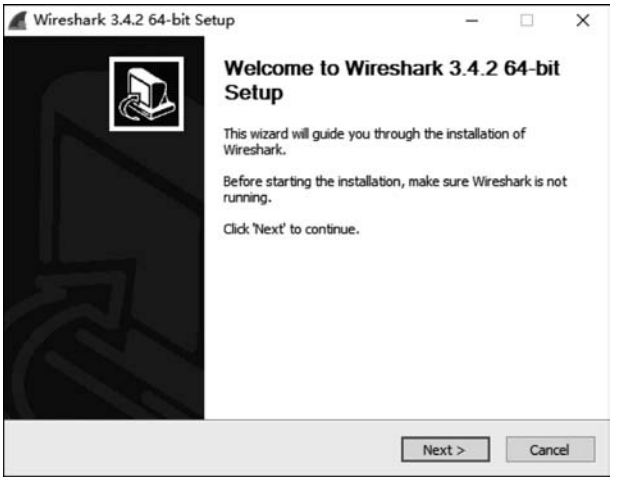


图 1.2 安装 Wireshark-win64-3.4.2.exe

单击 Next 按钮，显示许可协议，如图 1.3 所示。

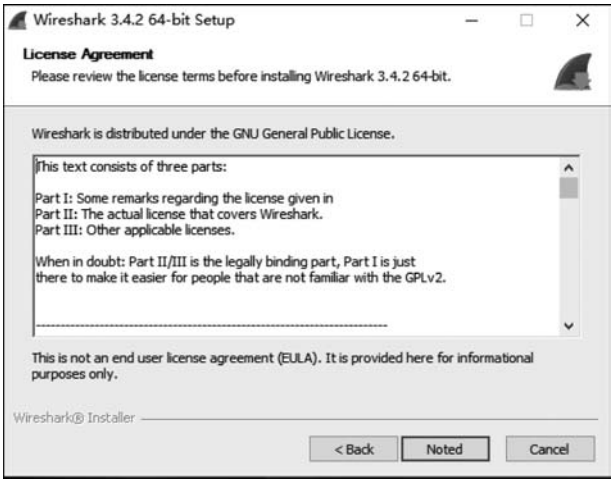


图 1.3　显示许可协议

单击 Noted 按钮，选择安装的组件，这里将所有的复选框都选中，如图 1.4 所示。

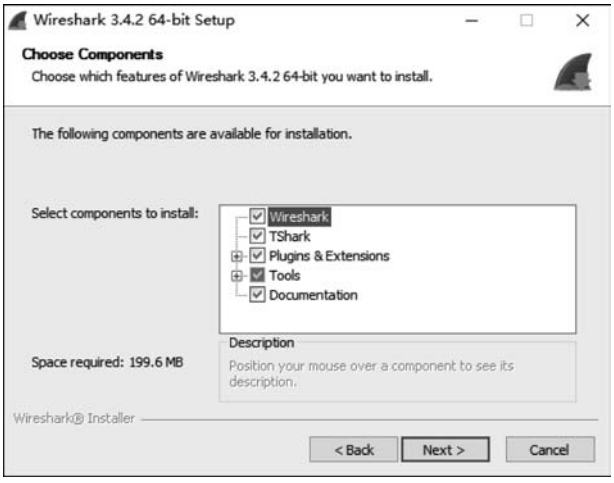


图 1.4　选择安装的组件

单击 Next 按钮，显示其他任务，默认安装，如图 1.5 所示。

单击 Next 按钮，选择安装的路径，如图 1.6 所示。

单击 Next 按钮，Wireshark 需要 Npcap 或 WinPcap 来捕获实时网络数据，安装 Npcap 和 WinPcap，继续安装，如图 1.7 所示。

单击 Next 按钮，安装 USB Capture，如图 1.8 所示。

单击 Install 按钮，开始安装，如图 1.9 所示。

安装时跳出的窗口如图 1.10 所示，开始安装 Npcap 插件，Npcap 是 WinPcap 的改进版。

单击 I Agree 按钮，继续安装，如图 1.11 所示。

单击 Install 按钮，显示如图 1.12 所示。

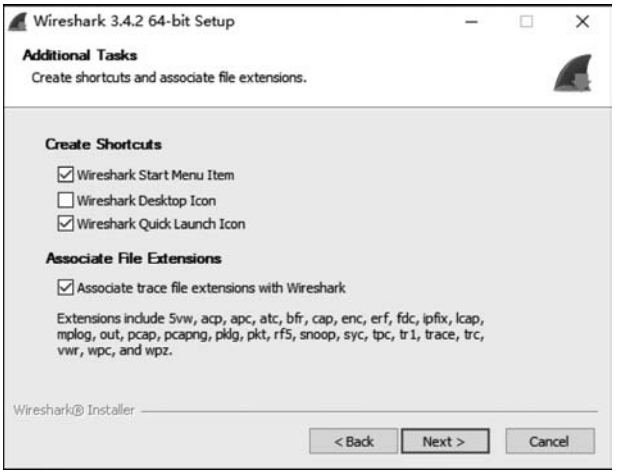


图 1.5 安装的任务

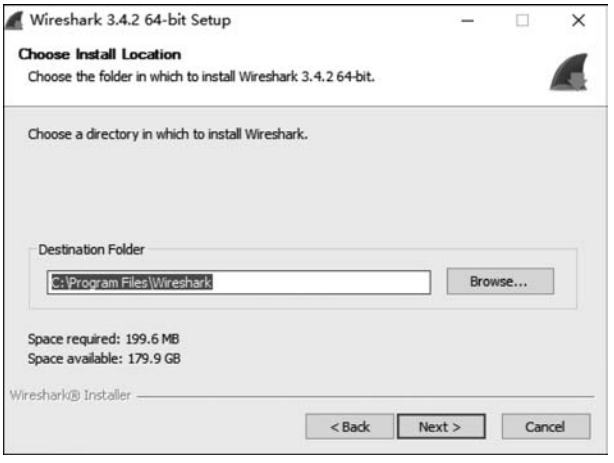


图 1.6 选择安装的路径

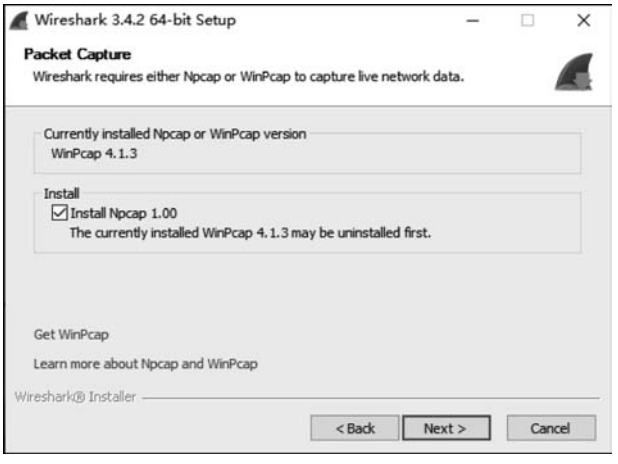


图 1.7 继续安装

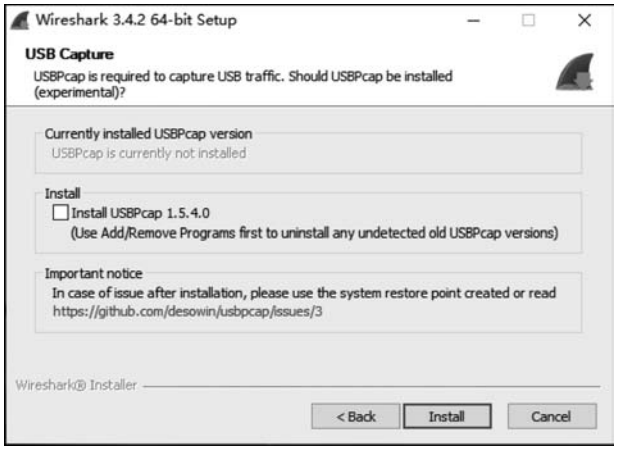


图 1.8　安装 USB Capture

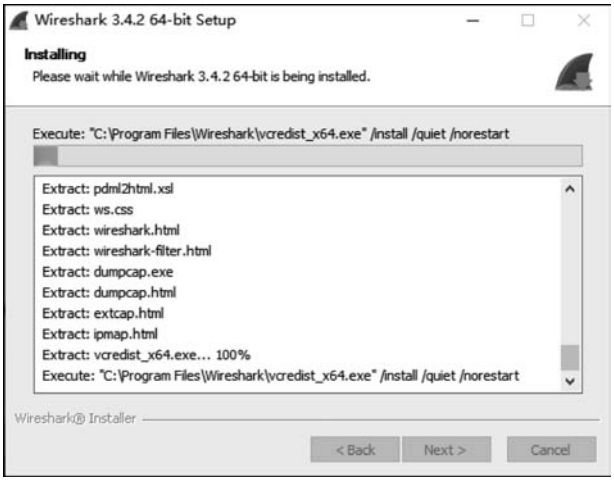


图 1.9　开始安装

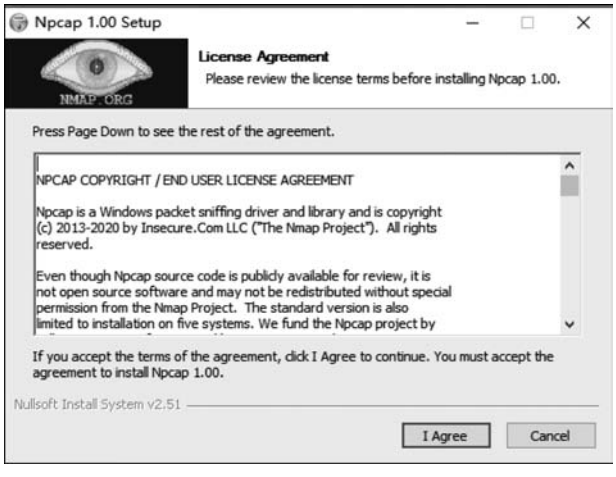


图 1.10　安装 Npcap 插件

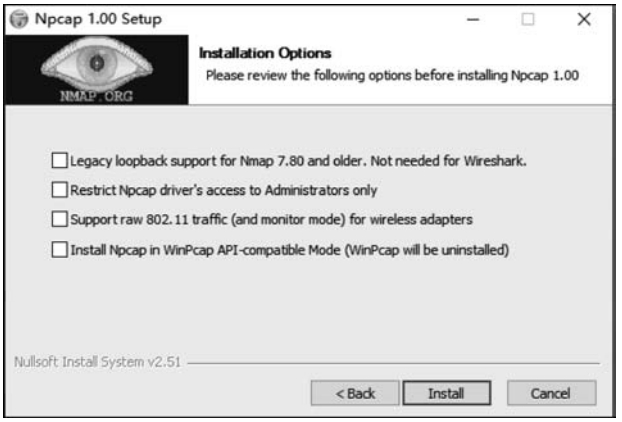


图 1.11 继续安装 Npcap 插件

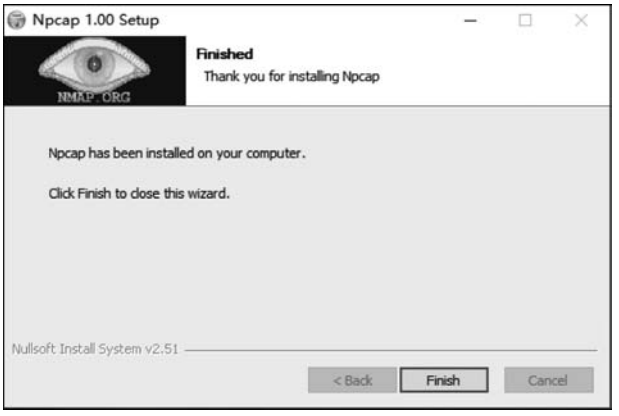


图 1.12 完成 Npcap 插件的安装

单击 Finish 按钮,完成 Npcap 的安装。如图 1.13 所示,显示安装完成。

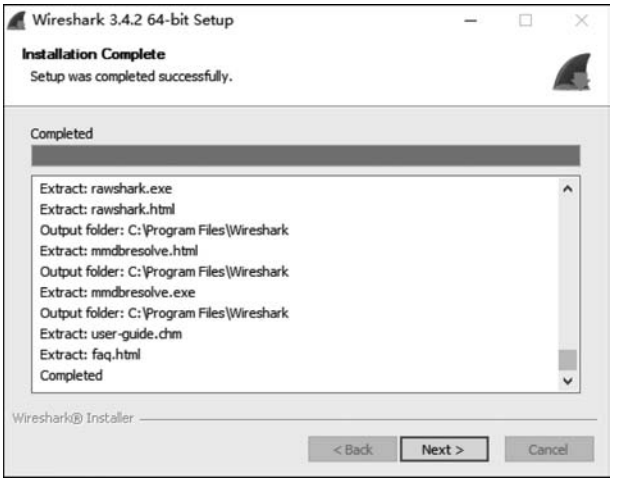


图 1.13 显示安装完成

单击 Next 按钮，Wireshark 安装完成，重启系统，如图 1.14 所示。

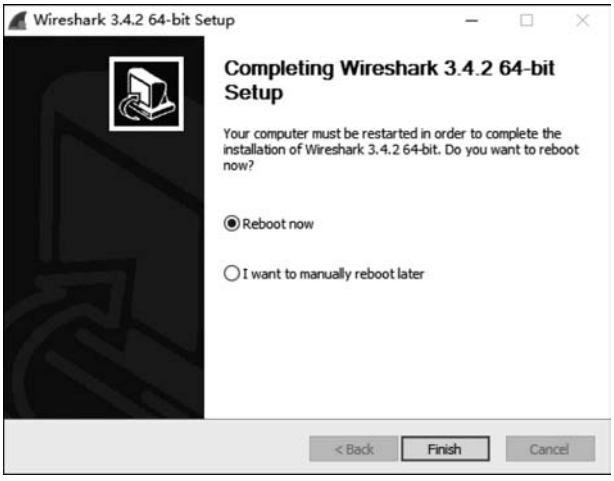


图 1.14　安装完成，重启系统

1.4　Wireshark 基本应用

1. 界面介绍

打开 Wireshark 程序，Wireshark 3.4.2 启动界面如图 1.15 所示。

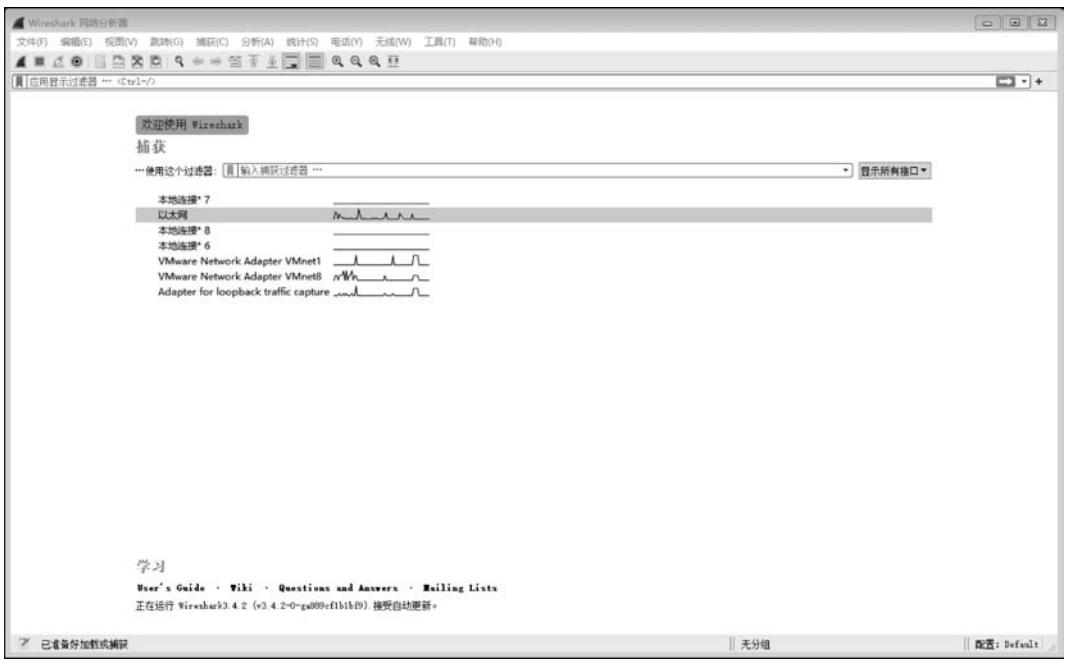


图 1.15　Wireshark 启动界面

首先选择要抓包的网卡,这里选择以太网。然后单击左上角蓝色的按钮,开始捕获分组,如图 1.16 所示。

1 号窗口是数据包列表,显示捕获的数据包,第一列为捕获的序列号,第二列为捕获的时间,第三列为源 IP 地址,第四列为目标 IP 地址,第五列为协议,第六列为长度,第七列为说明信息。

2 号窗口是数据包详细信息,在 1 号窗口数据包列表中选择指定数据包,在 2 号窗口数据包详细信息中会显示数据包的所有详细信息内容。数据包详细信息是非常重要的,可用来查看协议中的每一个字段。各行信息分别说明如下。

(1) Frame:物理层的数据帧概况。

(2) Ethernet Ⅱ:数据链路层以太网帧头部信息。

(3) Internet Protocol Version 4:网络层 IP 包头部信息。

(4) Transmission Control Protocol:传输层的数据段头部信息,此处是 TCP。

(5) Hypertext Transfer Protocol:应用层的信息,此处是 HTTP。

3 号窗口是 1 号窗口中选定的数据包字节区,其中左侧是十六进制表示,右侧是 ASCII 码表示。另外,在 2 号窗口中选中某层或某字段,3 号窗口对应位置也会被高亮。

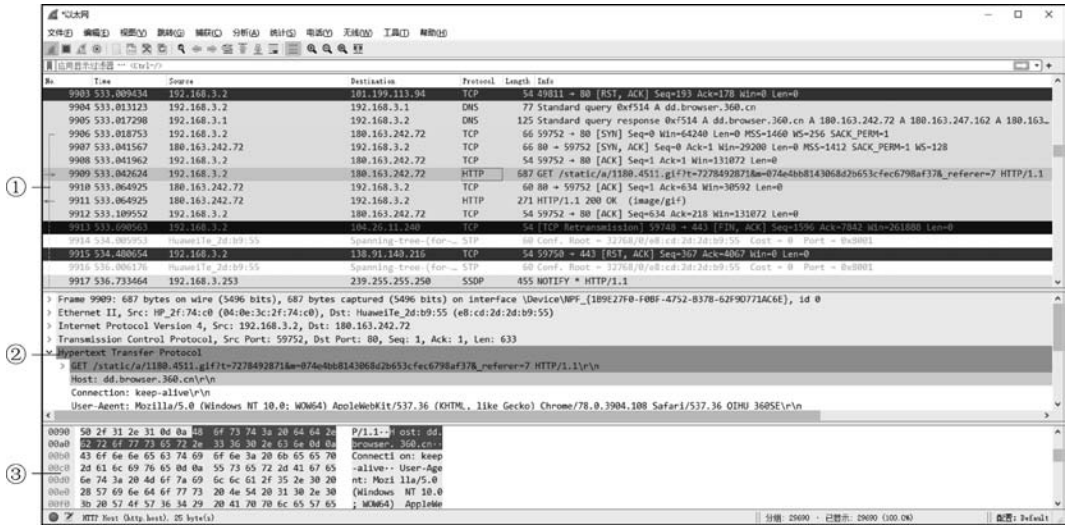


图 1.16 抓包结果

2. 设置数据列表颜色

从图 1.16 看到,数据包列表中不同的协议功能使用了不同的颜色区分。如需更改颜色,选择菜单栏“视图(V)”→“着色规则”,选择某个规则,使用下面的按钮更改前景色和背景色,如图 1.17 所示。

3. 设置网卡

如需重新选择网卡,应选择菜单栏“捕获(C)”→“选项”,打开“捕获选项”对话框,在 Input 选项卡中选择网卡后,单击“开始”按钮,开始抓包,如图 1.18 所示。

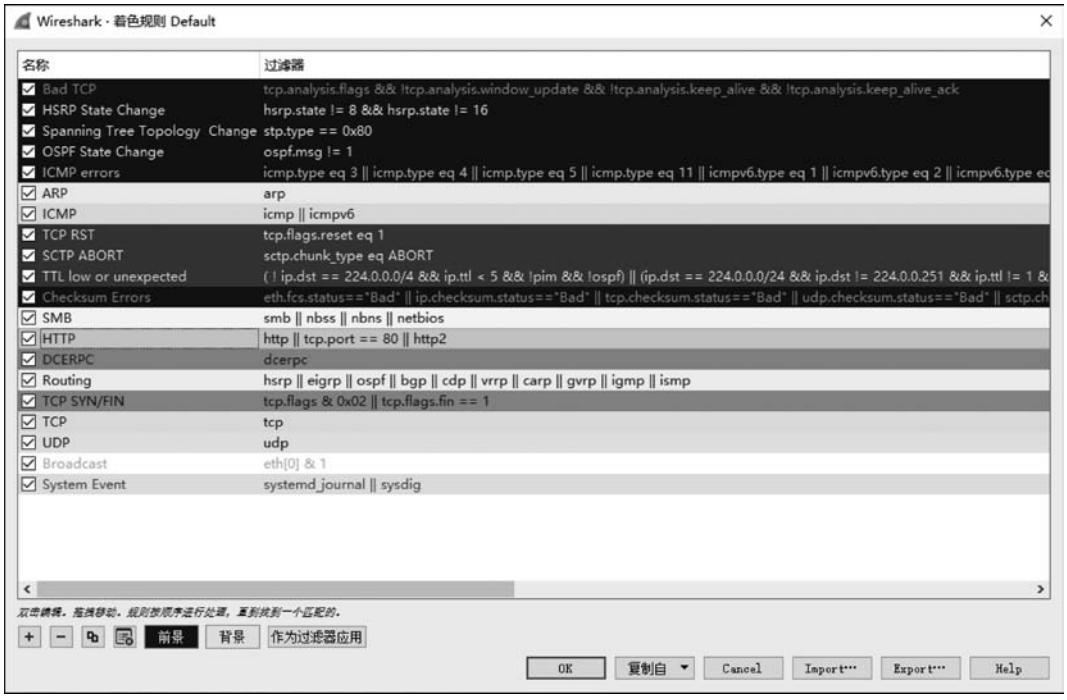


图 1.17　数据包列表中颜色的设置

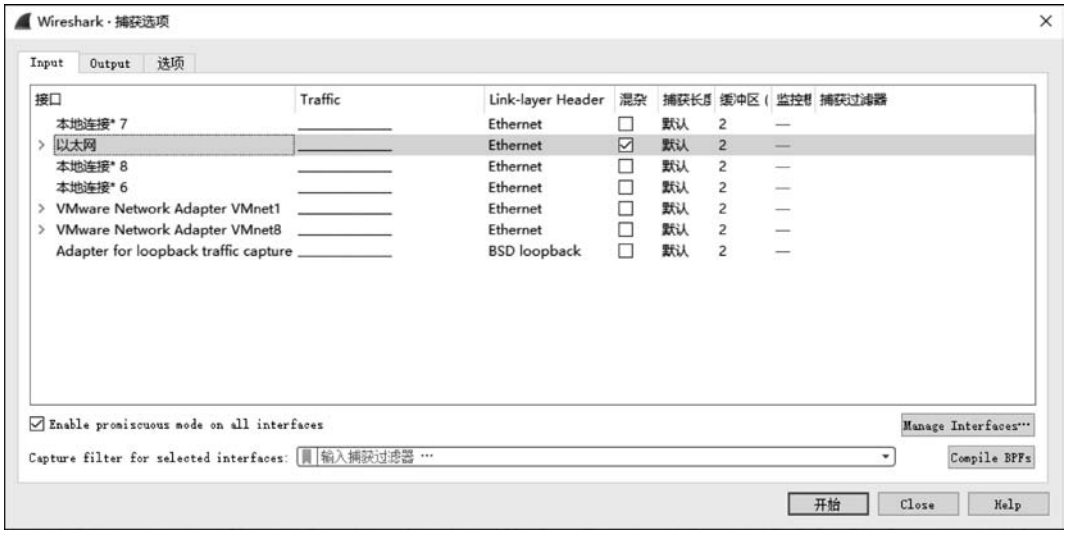


图 1.18　更换网卡抓包

4. 显示过滤器

在捕获时未设置捕获规则直接通过网卡抓取所有数据包，如图 1.19 所示，可看到抓取了所有协议的数据包。

显示过滤器用于在抓取数据包后设置过滤条件，从而过滤数据包。通常在抓取数据

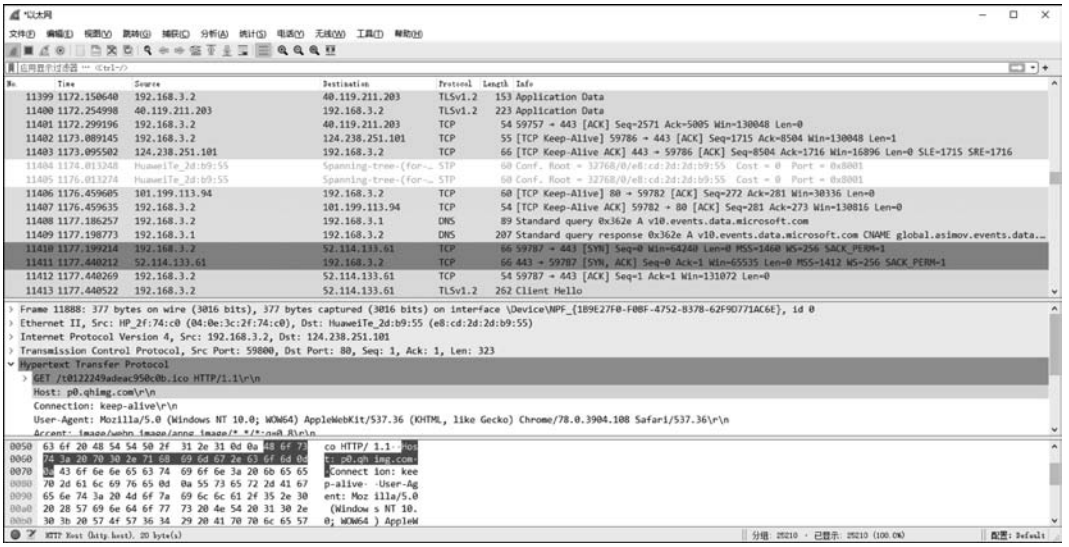


图 1.19　抓取所有数据包

包时设置条件相对宽泛,抓取的数据包内容较多时,使用显示过滤器设置条件过滤,以方便分析,如果只抓取 TCP 的数据包,则在显示过滤器中输入 TCP,如图 1.20 所示。

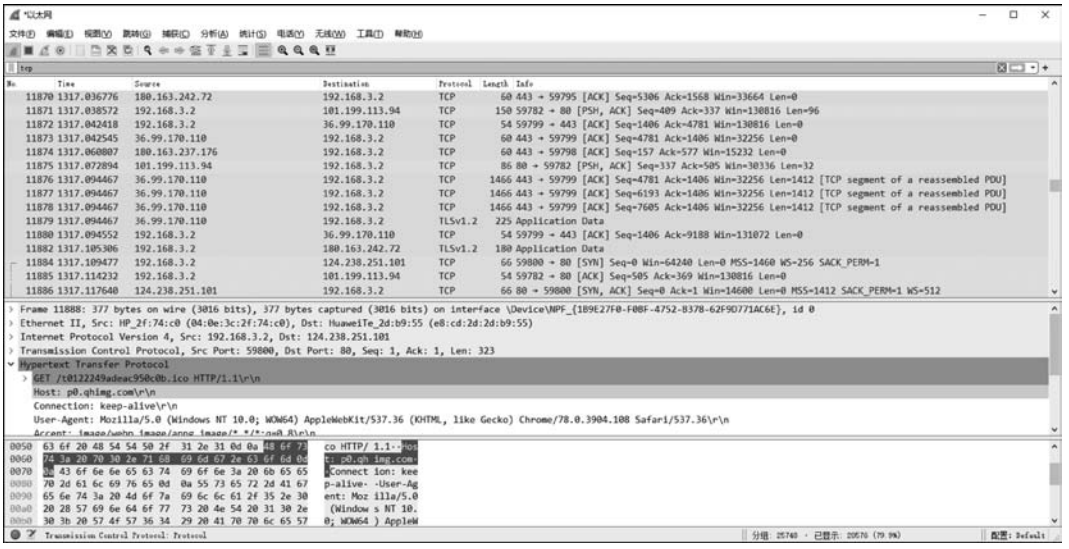


图 1.20　显示过滤器中输入 TCP 抓包

1.4.1　数据链路层过滤

按照 MAC 地址进行筛选:

格式:eth.src == MAC 地址

【例 1-1】 如筛选 MAC 地址为 e0:d5:5e:ac:eb:71 的数据包,则应该在显示过滤器中输入 eth.src == e0:d5:5e:ac:eb:71,然后单击右侧的箭头按钮,如图 1.21 所示。

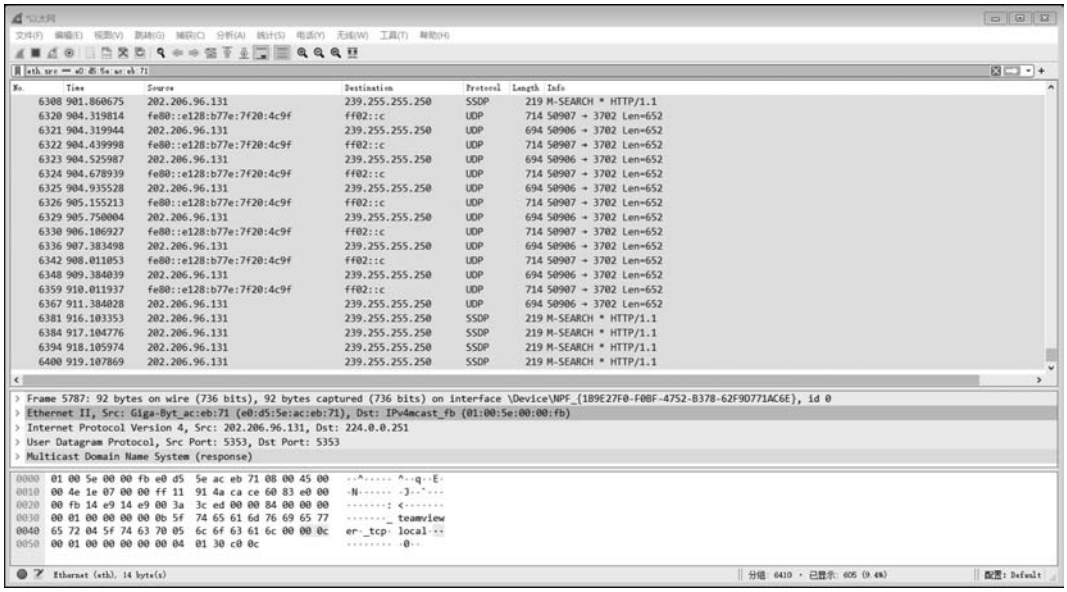


图 1.21　按照 MAC 地址筛选

注意：等号必须输入两个，如果输入一个等号则语法错误。输入框为红色表示错误，为绿色表示正确，可以执行过滤器。

1.4.2　网络层过滤

1. 按照 IP 地址筛选

格式：ip.addr == IP 地址

【例 1-2】 如想要过滤出目的地址或源地址为 202.206.96.52 的数据包，则应在过滤器中输入 ip.addr == 202.206.96.52，然后单击右侧的箭头按钮，如图 1.22 所示。

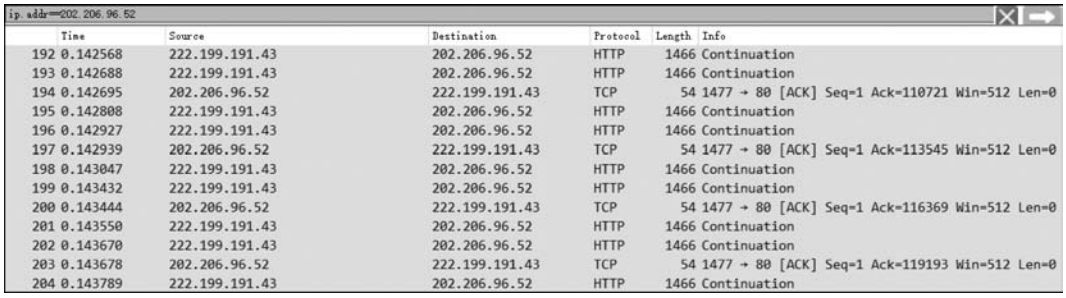


图 1.22　按照 IP 地址筛选

2. 按照源 IP 地址筛选

格式：ip.src == 源 IP 地址

【例 1-3】 如想要过滤源地址为 202.206.96.180 的数据包，根据语法规则，在过滤器中输入 ip.src == 202.206.96.180，然后单击右侧的箭头按钮，就可以进行过滤了，如

图 1.23 所示。

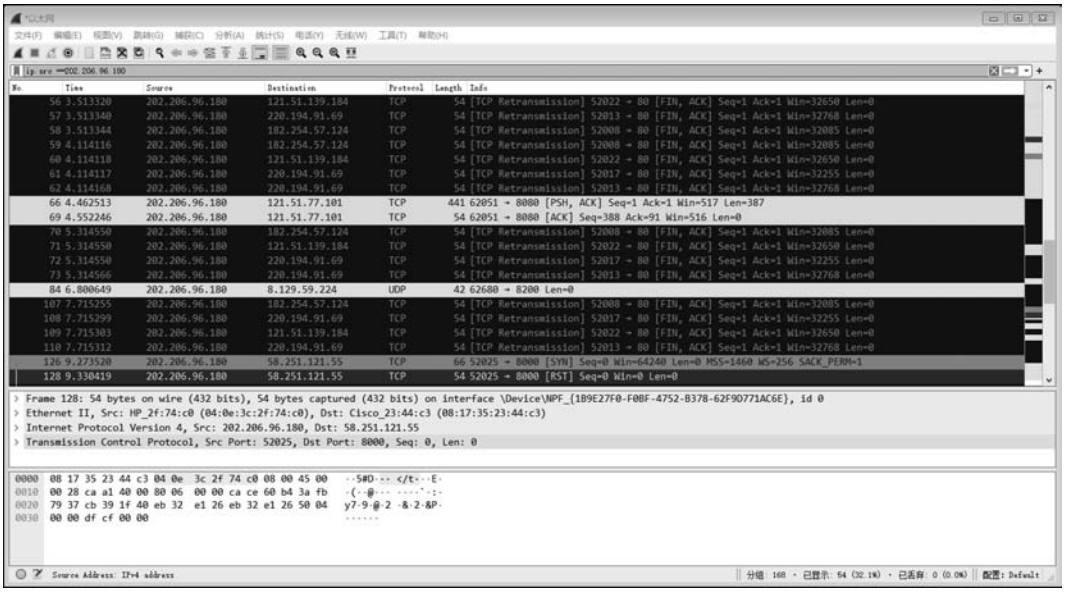


图 1.23　按照源 IP 地址筛选

3. 按照目的 IP 地址筛选

格式：ip.dst == IP 地址

【例 1-4】 如果要过滤目的地址为 202.206.96.180 的数据包，则根据语法规则在过滤器中输入 ip.dst == 202.206.96.180，然后单击右侧的箭头按钮，就可以进行过滤了，结果如图 1.24 所示。

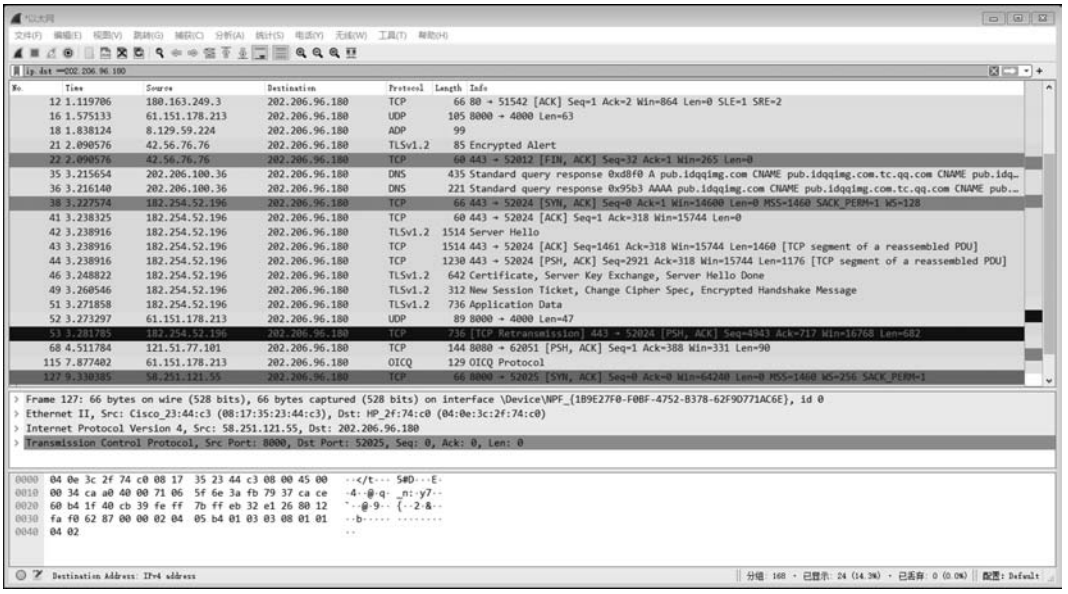


图 1.24　按照目的地址筛选

4. 按照指定的源地址和目的地址进行筛选数据包

格式：ip.src == IP 地址 && IP.dst == IP 地址

【例 1-5】 如果想要筛选源地址为 202.206.96.180，目的地址为 58.205.218.18 的数据包，只需要根据语法规则在过滤器中输入 ip.src == 202.206.96.180 && ip.dst == 58.205.218.18，然后单击右侧的箭头按钮，就可以进行过滤了，结果如图 1.25 所示。

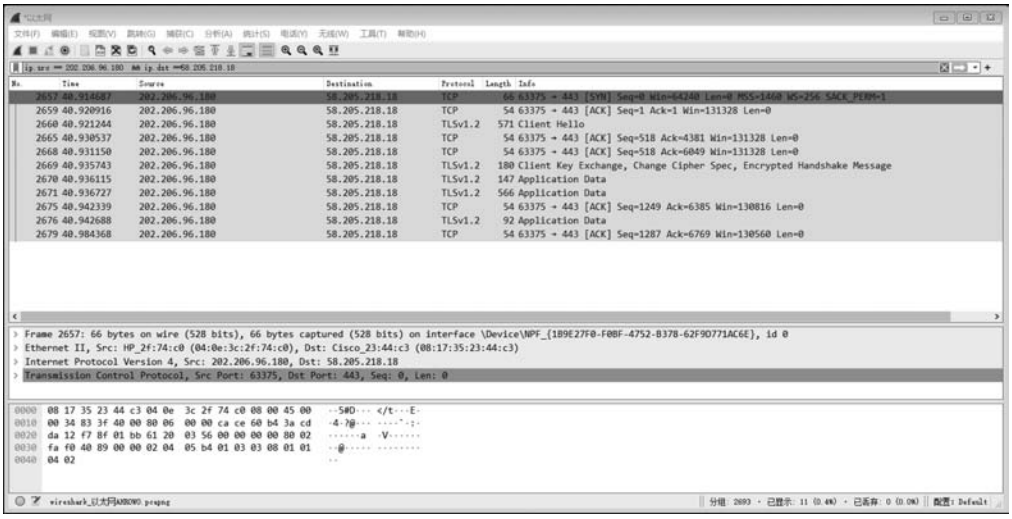


图 1.25　按照指定的源地址和目的地址筛选

1.4.3　传输层过滤

1. 筛选 TCP 的数据包

格式：tcp

【例 1-6】 在过滤器中输入规则 TCP，这样就可以过滤出所有协议为 TCP 的数据包，如图 1.26 所示。

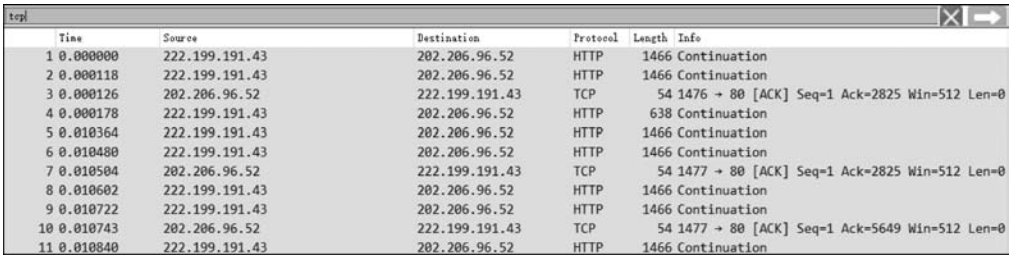


图 1.26　筛选 TCP 的数据包

2. 筛选不是 TCP 的数据包

格式：!tcp

【例 1-7】 在过滤器中输入规则！tcp，过滤出所有不是 TCP 的数据包，如图 1.27 所示。

注意：在英文状态下输入感叹号。

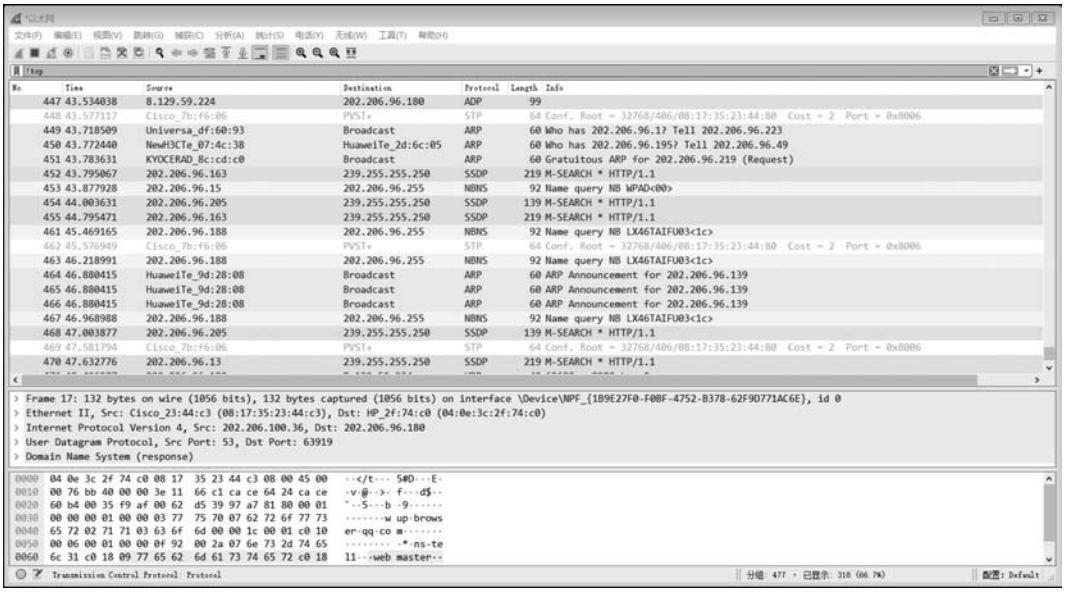


图 1.27　筛选不是 TCP 的数据包

3. 筛选端口是 80 的数据包

格式：tcp.port == 80

【例 1-8】 输入规则 tcp.port == 80，过滤出所有经过 80 端口的数据包，如图 1.28 所示。

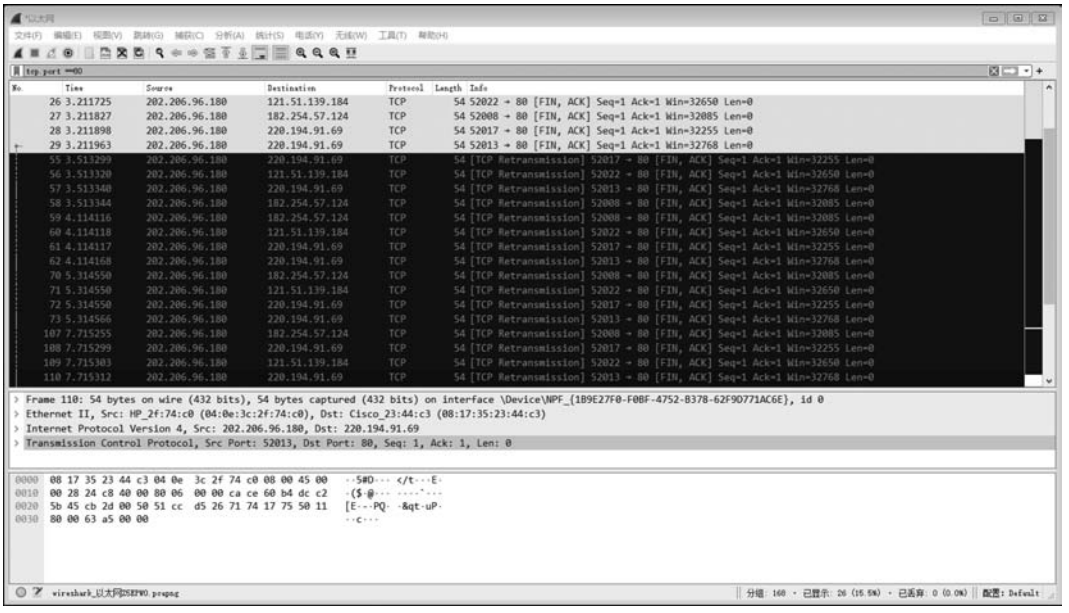
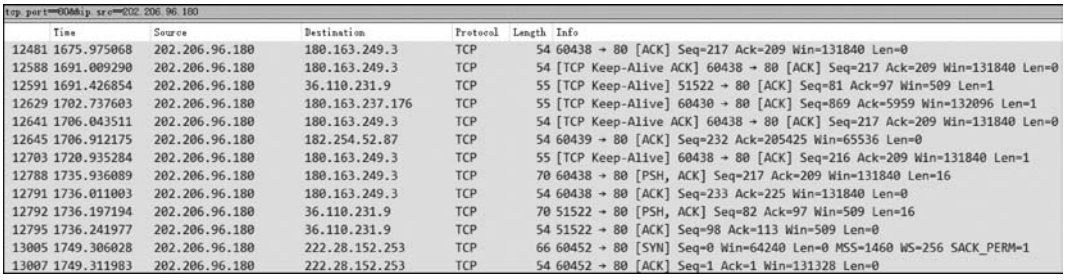


图 1.28　筛选端口是 80 的数据包

4. 筛选指定的源 IP 地址并且端口是 80 的数据包

格式：tcp.port == 80 && IP.src == 源 IP 地址

【例 1-9】 输入规则 tcp.port == 80 && ip.src == 202.206.96.180，筛选源 IP 地址为 202.206.96.180 并且端口是 80 的数据包，如图 1.29 所示。


tcp.port==80&&ip.src==202.206.96.180

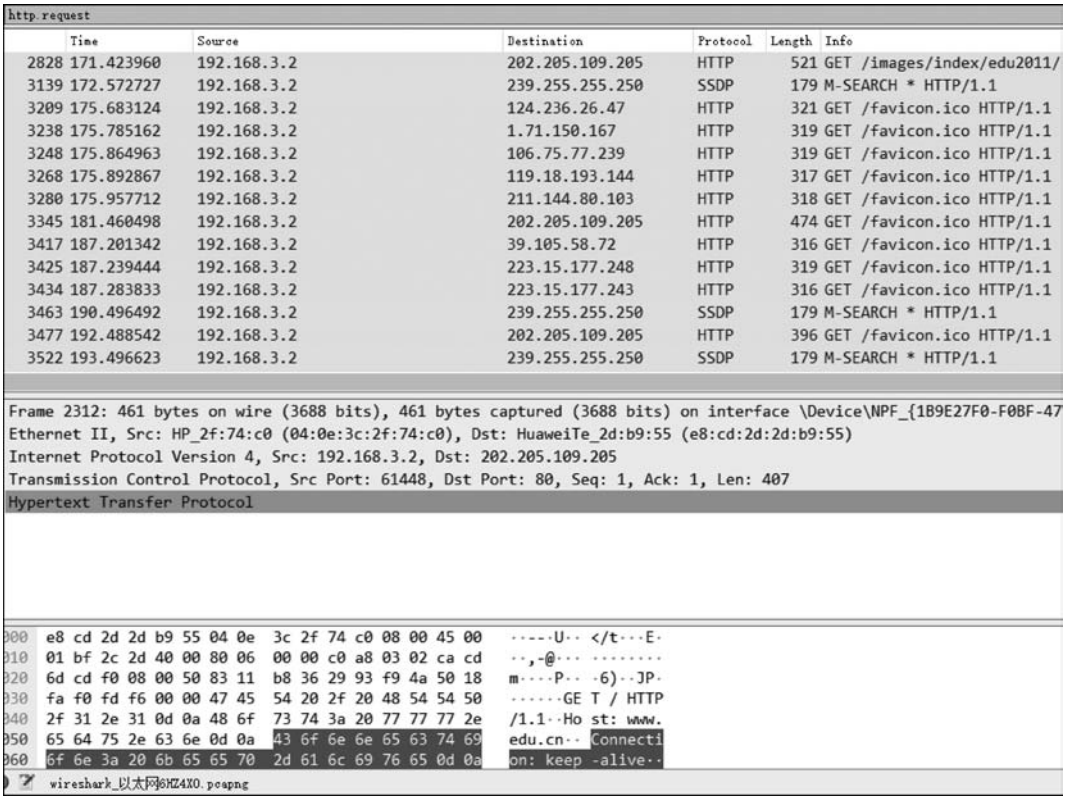
	Time	Source	Destination	Protocol	Length	Info
12481	1675.975068	202.206.96.180	180.163.249.3	TCP	54	60438 → 80 [ACK] Seq=217 Ack=209 Win=131840 Len=0
12588	1691.009290	202.206.96.180	180.163.249.3	TCP	54	[TCP Keep-Alive ACK] 60438 → 80 [ACK] Seq=217 Ack=209 Win=131840 Len=0
12591	1691.426854	202.206.96.180	36.110.231.9	TCP	55	[TCP Keep-Alive] 51522 → 80 [ACK] Seq=81 Ack=97 Win=509 Len=1
12629	1702.737603	202.206.96.180	180.163.237.176	TCP	55	[TCP Keep-Alive] 60430 → 80 [ACK] Seq=869 Ack=5959 Win=132096 Len=1
12641	1706.043511	202.206.96.180	180.163.249.3	TCP	54	[TCP Keep-Alive ACK] 60438 → 80 [ACK] Seq=217 Ack=209 Win=131840 Len=0
12645	1706.912175	202.206.96.180	182.254.52.87	TCP	54	60439 → 80 [ACK] Seq=232 Ack=205425 Win=65536 Len=0
12703	1720.935284	202.206.96.180	180.163.249.3	TCP	55	[TCP Keep-Alive] 60438 → 80 [ACK] Seq=216 Ack=209 Win=131840 Len=1
12788	1735.936089	202.206.96.180	180.163.249.3	TCP	70	60438 → 80 [PSH, ACK] Seq=217 Ack=209 Win=131840 Len=16
12791	1736.011003	202.206.96.180	180.163.249.3	TCP	54	60438 → 80 [ACK] Seq=233 Ack=225 Win=131840 Len=0
12792	1736.197194	202.206.96.180	36.110.231.9	TCP	70	51522 → 80 [PSH, ACK] Seq=82 Ack=97 Win=509 Len=16
12795	1736.241977	202.206.96.180	36.110.231.9	TCP	54	51522 → 80 [ACK] Seq=98 Ack=113 Win=509 Len=0
13005	1749.306028	202.206.96.180	222.28.152.253	TCP	66	60452 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
13007	1749.311983	202.206.96.180	222.28.152.253	TCP	54	60452 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0

图 1.29　筛选指定的源 IP 地址并且端口是 80 的数据包

注意：逻辑运算符“&&”表示与，“||”表示或，“!”表示非，用于规则之间的连接。

1.4.4　应用层过滤

（1）在过滤器中输入 http.request 表示请求，从图 1.30 看到源 IP 地址都为 192.168.3.2。


http.request

	Time	Source	Destination	Protocol	Length	Info
2828	171.423960	192.168.3.2	202.205.109.205	HTTP	521	GET /images/index/edu2011/
3139	172.572727	192.168.3.2	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
3209	175.683124	192.168.3.2	124.236.26.47	HTTP	321	GET /favicon.ico HTTP/1.1
3238	175.785162	192.168.3.2	1.71.150.167	HTTP	319	GET /favicon.ico HTTP/1.1
3248	175.864963	192.168.3.2	106.75.77.239	HTTP	319	GET /favicon.ico HTTP/1.1
3268	175.892867	192.168.3.2	119.18.193.144	HTTP	317	GET /favicon.ico HTTP/1.1
3280	175.957712	192.168.3.2	211.144.80.103	HTTP	318	GET /favicon.ico HTTP/1.1
3345	181.460498	192.168.3.2	202.205.109.205	HTTP	474	GET /favicon.ico HTTP/1.1
3417	187.201342	192.168.3.2	39.105.58.72	HTTP	316	GET /favicon.ico HTTP/1.1
3425	187.239444	192.168.3.2	223.15.177.248	HTTP	319	GET /favicon.ico HTTP/1.1
3434	187.283833	192.168.3.2	223.15.177.243	HTTP	316	GET /favicon.ico HTTP/1.1
3463	190.496492	192.168.3.2	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
3477	192.488542	192.168.3.2	202.205.109.205	HTTP	396	GET /favicon.ico HTTP/1.1
3522	193.496623	192.168.3.2	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1

Frame 2312: 461 bytes on wire (3688 bits), 461 bytes captured (3688 bits) on interface \Device\NPF_{1B9E27F0-F0BF-47
Ethernet II, Src: HP_2f:74:c0 (04:0e:3c:2f:74:c0), Dst: HuaweiTe_2d:b9:55 (e8:cd:2d:2d:b9:55)
Internet Protocol Version 4, Src: 192.168.3.2, Dst: 202.205.109.205
Transmission Control Protocol, Src Port: 61448, Dst Port: 80, Seq: 1, Ack: 1, Len: 407
Hypertext Transfer Protocol

```
000  e8 cd 2d 2d b9 55 04 0e  3c 2f 74 c0 08 00 45 00   ··--·U·· </t···E·
010  01 bf 2c 2d 40 00 80 06  00 00 c0 a8 03 02 ca cd   ··,-@··· ········
020  6d cd f0 08 00 50 83 11  b8 36 29 93 f9 4a 50 18   m····P·· ·6)··JP·
030  fa f0 fd f6 00 00 47 45  54 20 2f 20 48 54 54 50   ·····GE T / HTTP
040  2f 31 2e 31 0d 0a 48 6f  73 74 3a 20 77 77 77 2e   /1.1··Ho st: www.
050  65 64 75 2e 63 6e 0d 0a  43 6f 6e 6e 65 63 74 69   edu.cn·· Connecti
060  6f 6e 3a 20 6b 65 65 70  2d 61 6c 69 76 65 0d 0a   on: keep -alive··
```

wireshark_以太网6MZ4X0.pcapng

图 1.30　http.request 请求

注意：从图 1.30 可以看到，协议(protocol)列显示 HTTP 和 SSDP，SSDP 是简单服务发现协议，此协议为网络客户提供一种无须任何配置、管理和维护网络设备服务的机制，设备查询通过 HTTP 协议扩展 M-SEARCH 方法实现。

(2) 输入 http.response 表示响应，如图 1.31 所示。从图 1.31 中可以看到，目的 IP 地址都为 192.168.3.2 接收响应。

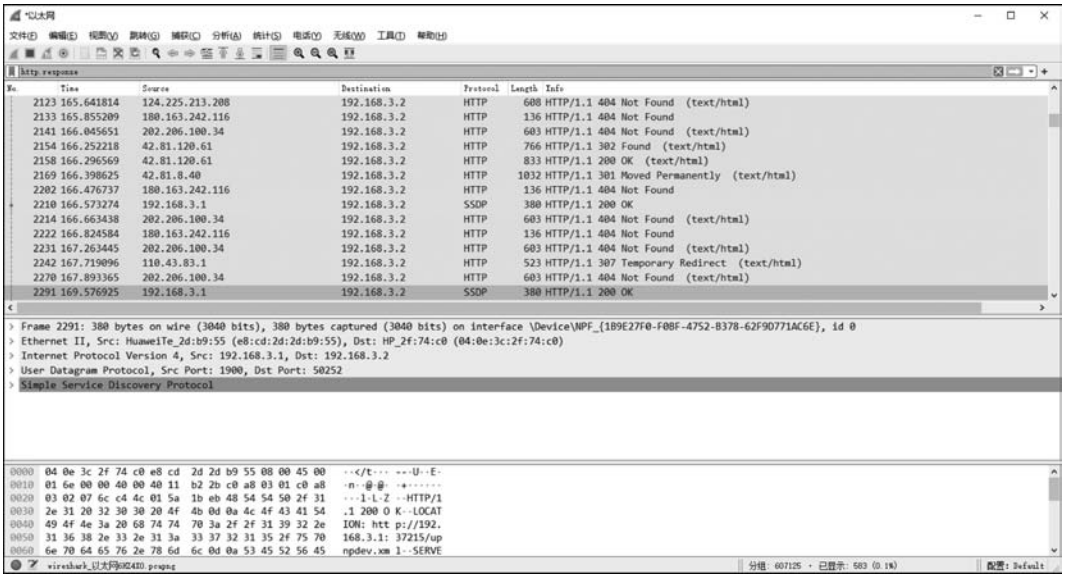


图 1.31 http.response 响应

(3) 输入 http.request.method==="GET"，请求指定的页面信息，显示 HTTP GET 方法的请求，如图 1.32 所示。

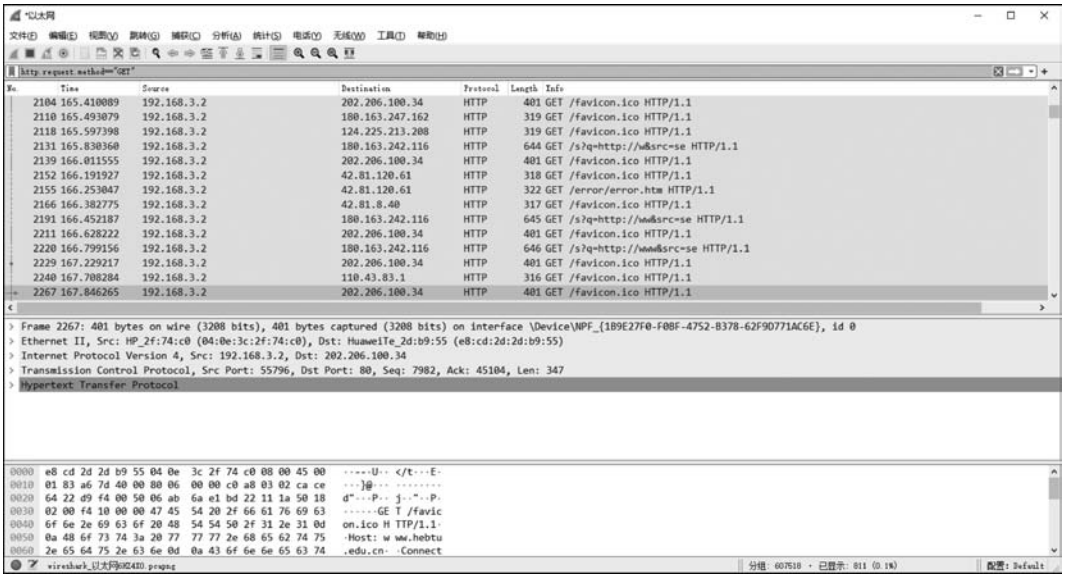


图 1.32 HTTP GET 方法的请求

（4）在过滤器输入 http.request.uri contains ".php"，筛选 URL HTTP 中包含.php 的数据包，如图 1.33 所示。

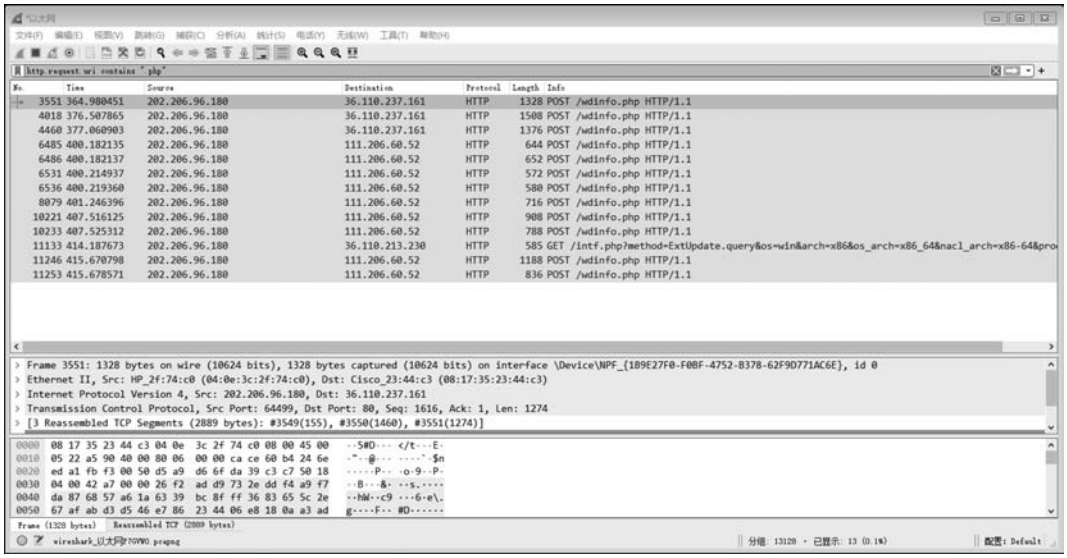


图 1.33　筛选包含.php 的数据包

1.4.5　抓包实例

【例 1-10】 使用 ICMP 协议并抓取百度的数据包，步骤如下。

（1）在 cmd 下执行 ping www.baidu.com -t 命令，如图 1.34 所示，显示百度的 IP 地址为 182.61.200.7。

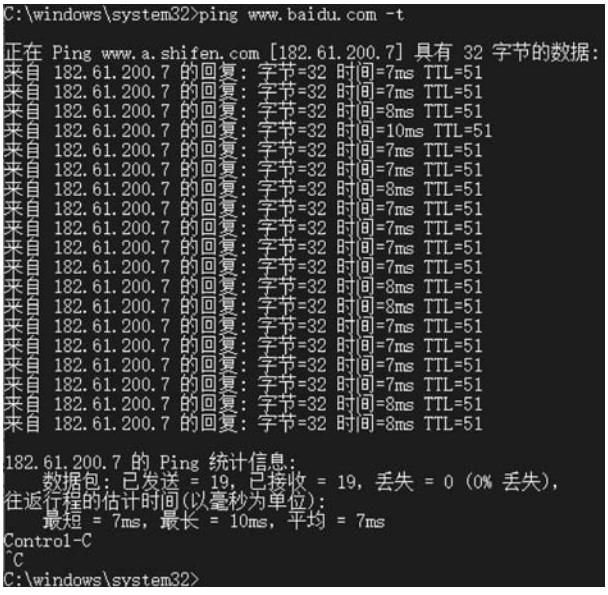


图 1.34　执行 ping 操作

(2) 回到桌面,在过滤器中,输入 ip.addr == 182.61.200.7 and icmp,如图 1.35 所示。

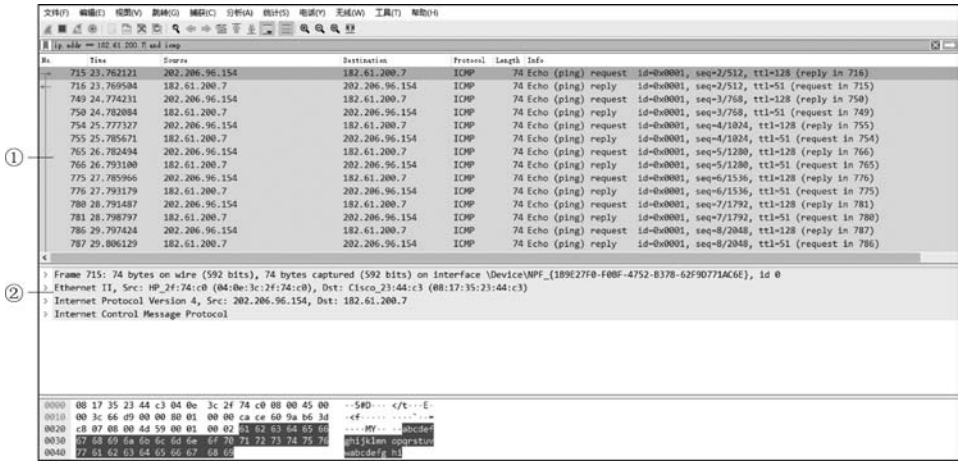


图 1.35 抓取百度数据包

(3) 选中图 1.35 中 1 号窗口的第一行的数据包,在 2 号窗口数据包详细信息中显示这个数据包的所有详细信息内容,包括 Frame、Ethernet Ⅱ、Internet Protocol Version 4、Internet Control Message Protocol。单击 2 号窗口左侧三角,可显示抓到的数据包的详细信息,如图 1.36 所示。

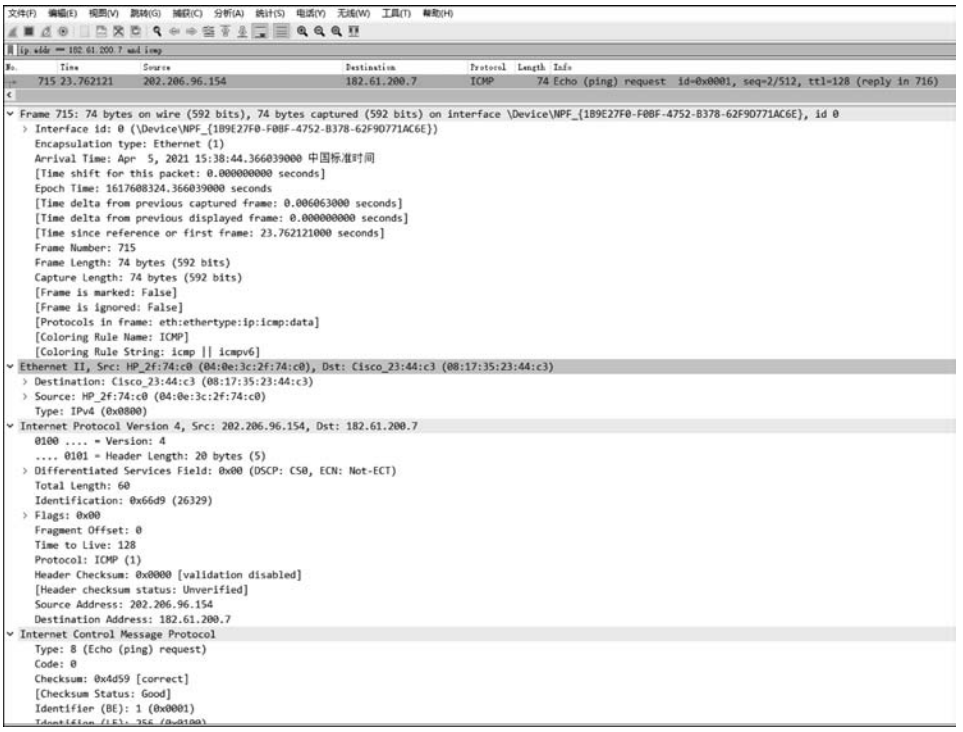


图 1.36 数据包的详细信息

1.5　网络安全分析实例

视频讲解

1.5.1　ARP 欺骗原理

ARP 协议是 Address Resolution Protocol(地址解析协议)的缩写,在以太网环境中,数据的传输所依赖的是 MAC 地址而非 IP 地址,而将已知 IP 地址转换为 MAC 地址的工作是由 ARP 协议来完成的。

ARP 欺骗就是一种典型的中间人攻击方式,中间人攻击就是 A 机器和 B 机器在进行正常的网络通信时,Hacker 实施中间人攻击,监听 A 机器和 B 机器的通信,在这个过程中 Hacker 作为中间人会处理转发它们的数据信息,也就是说 A 机器和 B 机器之间依然可以正常通信,并且它们也不会发现通信过程中多了一个人。A 机器以为自己是在和 B 机器通信,然而实际上多了一个 Hacker 正在默默地监听 A 机器和 B 机器之间的通信。

ARP 协议的缺点是没有任何认证机制。当主机 A 收到一个发送方 IP 地址为 192.168.157.142 的 ARP 请求包时,主机 A 并不会对这个数据包做任何的真伪校验,无论这个数据包是否来自 192.168.157.142,它都会将其添加到 ARP 缓存表中,Hacker 正是利用这一点来冒充网关等主机的。

【例 1-11】 在 VMware 中分别创建两台虚拟机,Kali 机器的 IP 地址为 192.168.157.142,Windows 7 机器的 IP 地址为 192.168.157.170。

(1) 在 Kali 下执行"09-嗅探/欺骗"下的 Wireshark 程序,如图 1.37 所示。

说明:创建虚拟机和 Kali 系统的详细安装步骤可以参考项目 5。

图 1.37　在 Kali 下执行 Wireshark

(2) 在 Kali 下进入终端模式,执行 ping 192.168.157.170 命令,如图 1.38 所示。

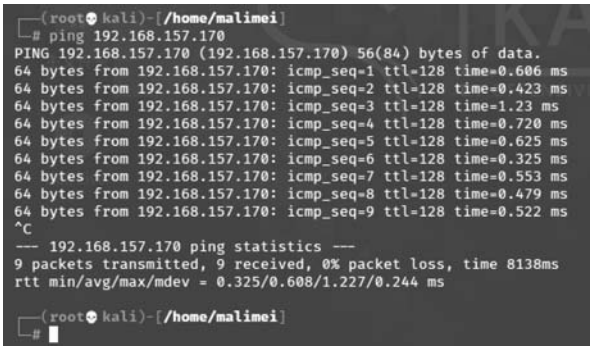


图 1.38 Kali 机器 ping Windows 7

(3) Kali 机器 192.168.157.142 和 Windows 7 机器 192.168.157.170 进行通信时,会先在 ARP 缓存表查找 192.168.157.170 对应的 ARP 表项(即 192.168.111.170 对应的 MAC 地址),如果没有找到则会发送 ARP 请求。IP 为 192.168.157.142 的主机首先会以广播方式发送一个 ARP 请求包获取 192.168.157.170 主机的 MAC 地址,其内容为 who has 192.168.157.170? Tell 192.168.157.142,如图 1.39 所示。

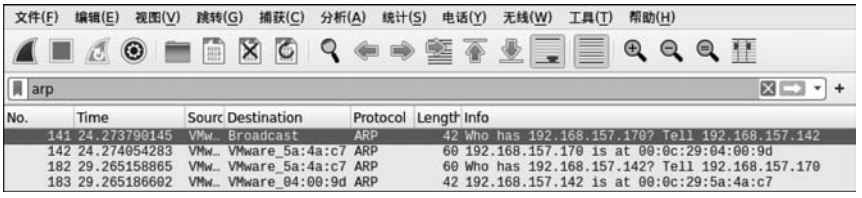


图 1.39 抓取到的广播内容

(4) 当 IP 为 192.168.157.170 的主机收到这个 ARP 请求包时并不会做任何的真伪校验,而是直接回复一个 ARP 响应包,把自己的 MAC 地址告诉 192.168.157.142 的主机,如图 1.40 所示。

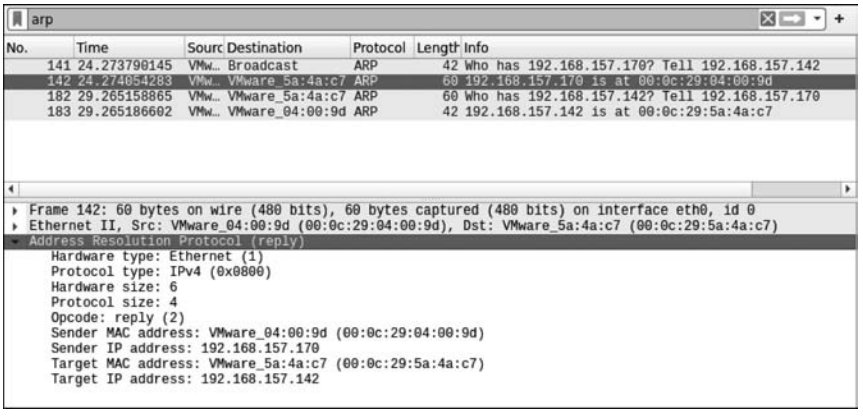


图 1.40 回复响应包