
Wireshark 网络分析实例集锦

(内部资料)

A large, light gray watermark of the Wireshark logo is centered on the page. It features a stylized shark fin above the word "WIRESHARK" in a bold, sans-serif font.

WIRESHARK

大学霸

www.daxueba.net



前言

由于网络广泛广泛，与网络相关的安全问题也就变的非常重要。为了更好的分析整个网络的情况，人们开始使用各种专业的数据包分析工具。**Wireshark** 是一款最知名的开源网络封包分析软件。它可以抓取网络封包，并以最为详细的方式，显示封包的数据。

Wireshark 应用非常广泛。例如，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** 检查数据传输的安全问题；开发者使用它为新的通信协议排错；普通人使用 **Wireshark** 来学习网络协议的相关知识。

由于 **Wireshark** 工具的广泛使用及市场的需求，笔者编写了这本书。本书按照网络分析专业流程，一步步地介绍了 **Wireshark** 各项功能的使用。本书还介绍了命令行下数据捕获的方法，以满足了在命令行下操作的用户。希望各位读者能在本书的带领下熟练地掌握 **Wireshark**，并且成为数据包的分析高手。

1.学习所需的系统和设备

本书所讲解的内容基于 Windows 7 和 Red Hat Enterprise Linux 6.4。读者在学习的时候，也可以采用其他操作系统。如果为了方便抓取各种数据，建议读者安装 VM ware，以虚拟各种其他系统或者服务。

2.学习建议

大家学习之前，可以致信到 wireshark@daxueba.net，获取相关的资料和软件。如果大家在学习过程遇到问题，也可以将问题发送到该邮箱。我们尽可能给大家解决。

目 录

| | | |
|-------|--------------------------------|----|
| 第 1 章 | Wireshark 的基础知识..... | 1 |
| 1.1 | Wireshark 的功能..... | 1 |
| 1.1.1 | Wireshark 主窗口界面..... | 1 |
| 1.1.2 | Wireshark 的作用..... | 2 |
| 1.2 | 安装 Wireshark..... | 3 |
| 1.2.1 | 获取 Wireshark..... | 3 |
| 1.2.2 | 安装 Wireshark..... | 5 |
| 1.3 | Wireshark 捕获数据..... | 11 |
| 1.5 | 认识数据包 | 12 |
| 1.6 | 捕获 HTTP 包 | 14 |
| 1.7 | 访问 Wireshark 资源..... | 18 |
| 1.8 | Wireshark 快速入门..... | 21 |
| 1.9 | 分析网络数据 | 28 |
| 1.9.1 | 分析 Web 浏览数据..... | 28 |
| 1.9.2 | 分析后台数据 | 30 |
| 1.10 | 打开其它工具捕获的文件 | 31 |
| 第 2 章 | 设置 Wireshark 视图..... | 33 |
| 2.1 | 设置 Packet List 面板列 | 33 |
| 2.1.1 | 添加列 | 33 |
| 2.1.2 | 隐藏、删除、重新排序及编辑列 | 35 |
| 2.2 | Wireshark 分析器及 Profile 设置..... | 41 |
| 2.2.1 | Wireshark 分析器..... | 41 |
| 2.2.2 | 分析非标准端口号流量 | 43 |
| 2.2.3 | 设置 Wireshark 显示的特定数据类型..... | 45 |
| 2.2.4 | 使用 Profile 定制 Wireshark..... | 50 |
| 2.2.5 | 查找关键的 Wireshark Profile..... | 52 |
| 2.3 | 数据包时间延迟 | 54 |
| 2.3.1 | 时间延迟 | 54 |
| 2.3.2 | 检查延迟问题 | 55 |
| 2.3.3 | 检查时间差延迟问题 | 57 |
| 第 3 章 | 捕获过滤器技巧 | 61 |
| 3.1 | 捕获过滤器简介 | 61 |
| 3.2 | 选择捕获位置 | 62 |
| 3.3 | 选择捕获接口 | 62 |
| 3.4.1 | 判断那个适配器上的数据 | 63 |
| 3.4.2 | 使用多适配器捕获 | 63 |
| 3.4 | 捕获以太网数据 | 64 |
| 3.5 | 捕获无线数据 | 65 |

| | | |
|-------|-------------------------------------|-----|
| 3.5.1 | 捕获无线网络数据方式 | 65 |
| 3.5.2 | 使用 AirPcap 适配器 | 66 |
| 3.6 | 处理大数据 | 66 |
| 3.6.1 | 捕获过滤器 | 66 |
| 3.6.2 | 捕获文件集 | 68 |
| 3.7 | 处理随机发生的问题 | 70 |
| 3.8 | 捕获基于 MAC/IP 地址数据 | 72 |
| 3.8.1 | 捕获单个 IP 地址数据 | 72 |
| 3.8.2 | 捕获 IP 地址范围 | 74 |
| 3.8.3 | 捕获广播或多播地址数据 | 76 |
| 3.8.4 | 捕获 MAC 地址数据 | 77 |
| 3.9 | 捕获端口应用程序数据 | 80 |
| 3.9.1 | 捕获所有端口号的数据 | 80 |
| 3.9.2 | 结合基于端口的捕获过滤器 | 81 |
| 3.10 | 捕获特定 ICMP 数据 | 82 |
| 第 4 章 | 显示技巧 | 86 |
| 4.1 | 显示过滤器简介 | 86 |
| 4.2 | 使用显示过滤器 | 87 |
| 4.2.1 | 显示过滤器语法 | 87 |
| 4.2.2 | 检查语法错误 | 89 |
| 4.2.3 | 识别字段名 | 91 |
| 4.2.4 | 比较运算符 | 92 |
| 4.2.5 | 表达式过滤器 | 93 |
| 4.2.6 | 使用自动补全功能 | 94 |
| 4.2.7 | 手动添加显示列 | 96 |
| 4.3 | 编辑和使用默认显示过滤器 | 98 |
| 4.4 | 过滤显示 HTTP | 100 |
| 4.5 | 过滤显示 DHCP | 102 |
| 4.6 | 根据地址过滤显示 | 103 |
| 4.6.1 | 显示单个 IP 地址或主机数据 | 103 |
| 4.6.2 | 显示一个地址范围的数据 | 106 |
| 4.6.3 | 显示一个子网 IP 的数据 | 107 |
| 4.7 | 过滤显示单一的 TCP/UDP 会话 | 108 |
| 4.8 | 使用复杂表达式过滤 | 112 |
| 4.8.1 | 使用逻辑运算符 | 112 |
| 4.8.2 | 使用括号 | 114 |
| 4.8.3 | 使用关键字 | 116 |
| 4.8.4 | 使用通配符 | 118 |
| 4.9 | 发现通信延迟 | 119 |
| 4.9.1 | 时间过滤器 (frame.time_delta) | 119 |
| 4.9.2 | 基于 TCP 的时间过滤 (tcp.time_delta) | 120 |

| | | |
|--------|--------------------------|-----|
| 4.10 | 设置显示过滤器按钮 | 123 |
| 4.10.1 | 创建显示过滤器表达式按钮 | 123 |
| 4.10.2 | 编辑、添加、删除显示过滤器按钮 | 124 |
| 4.10.3 | 编辑 preferences 文件 | 125 |
| 第 5 章 | 着色规则和数据包导出 | 128 |
| 5.1 | 认识着色规则 | 128 |
| 5.2 | 禁用着色规则 | 129 |
| 5.2.1 | 禁用指定类型数据包彩色高亮 | 129 |
| 5.2.2 | 禁用所有包彩色高亮 | 131 |
| 5.3 | 创建用户着色规则 | 132 |
| 5.3.1 | 创建时间差着色规则 | 132 |
| 5.3.2 | 快速查看 FTP 用户名密码着色规则 | 133 |
| 5.3.3 | 创建单个会话着色规则 | 136 |
| 5.4 | 导出数据包 | 138 |
| 5.4.1 | 导出显示包 | 138 |
| 5.4.2 | 导出标记包 | 140 |
| 5.4.3 | 导出包的详细信息 | 141 |
| 第 6 章 | 构建图表 | 147 |
| 6.1 | 数据统计表 | 147 |
| 6.1.1 | 端点统计 | 147 |
| 6.1.2 | 网络会话统计 | 149 |
| 6.1.3 | 快速过滤会话 | 150 |
| 6.1.4 | 地图化显示端点统计信息 | 152 |
| 6.2 | 协议分层统计 | 155 |
| 6.3 | 图表化显示带宽使用情况 | 156 |
| 6.3.1 | 认识 IO Graph | 156 |
| 6.3.2 | 应用显示过滤器 | 157 |
| 6.4 | 专家信息 | 161 |
| 6.5 | 构建各种网络错误图表 | 162 |
| 6.5.1 | 构建所有 TCP 标志位包 | 163 |
| 6.5.2 | 构建单个 TCP 标志位包 | 164 |
| 第 7 章 | 重组数据 | 166 |
| 7.1 | 重组 Web 会话 | 166 |
| 7.1.1 | 重组 Web 浏览会话 | 166 |
| 7.1.2 | 导出 HTTP 对象 | 171 |
| 7.2 | 重组 FTP 会话 | 174 |
| 7.2.1 | 重组 FTP 数据 | 174 |
| 7.2.2 | 提取 FTP 传输的文件 | 176 |
| 第 8 章 | 添加注释 | 180 |
| 8.1 | 捕获文件注释 | 180 |
| 8.2 | 包注释 | 180 |

| | | |
|-------|--|-----|
| 8.2.1 | 添加包注释 | 181 |
| 8.2.2 | 查看包注释 | 181 |
| 8.3 | 导出包注释 | 183 |
| 8.3.1 | 使用 Export Packet Dissections 功能导出..... | 183 |
| 8.3.2 | 使用复制功能导出包 | 185 |
| 第 9 章 | 捕获、分割、合并数据 | 189 |
| 9.1 | 将大文件分割为文件集 | 189 |
| 9.1.1 | 添加 Wireshark 程序目录到自己的位置..... | 189 |
| 9.1.2 | 使用 Capinfos 获取文件大小和包数 | 189 |
| 9.1.3 | 分割文件 | 190 |
| 9.2 | 合并多个捕获文件 | 195 |
| 9.3 | 命令行捕获数据 | 196 |
| 9.3.1 | Dumpcap 和 Tshark 工具..... | 196 |
| 9.3.2 | 使用捕获过滤器 | 199 |
| 9.3.3 | 使用显示过滤器 | 199 |
| 9.4 | 导出字段值和统计信息 | 200 |
| 9.4.1 | 导出字段值 | 200 |
| 9.4.2 | 导出数据统计 | 202 |

第 3 章 捕获过滤器技巧

捕获过滤器用于决定将什么样的信息记录在捕获文件中。在使用 Wireshark 捕获数据时，捕获过滤器是数据经过的第一层过滤器，它用来控制捕获数据的数量。通过设置捕获过滤器，可以避免产生过大的捕获文件。本章将介绍使用捕获过滤器技巧。

3.1 捕获过滤器简介

使用 Wireshark 的默认设置捕获数据时，会产生大量冗余信息，这样会导致用户很难找到自己需要的部分。这时就可以使用捕获过滤器来控制捕获数据的数量了。捕获过滤器的设置界面如图 3.1 所示。

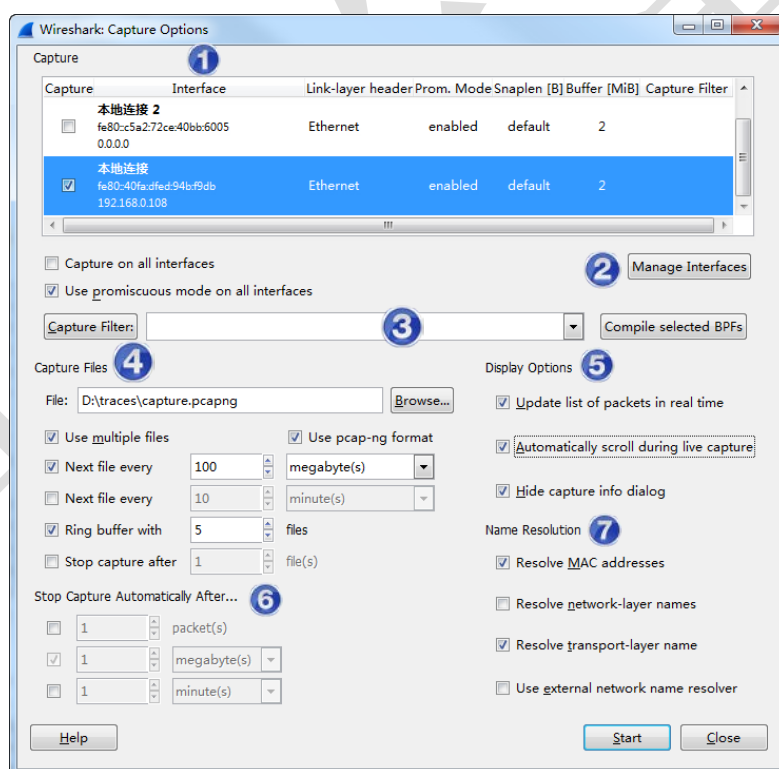


图 3.1 捕获选项

在该图中，每部分的含义如下所示：

- ❑ ①Interface 列表：选择一个或多个接口（捕获多个适配器）。
- ❑ ②Manage Interfaces 按钮：单击该按钮可以添加或删除接口。
- ❑ ③Capture Filter 下拉列表：显示被应用的捕获过滤器（双击可以修改、删除或添加捕获过滤器）。
- ❑ ④Capture File(s)选项框：设置保存多个文件、循环缓冲区大小和基于文件数量自动

停止的条件。

- ❑ ⑤Display Options 选项框：设置捕获数据时，自动滚动显示捕获的数据包。
- ❑ ⑥Stop Capture 选项框：设置自动停止条件，如基于包数、数据捕获的数量或运行时间。
- ❑ ⑦Name Resolution 选项框：为 MAC 地址、IP 地址和端口号启动/禁用名称解析。

当以上捕获选项设置完成后，就可以看到 Start 按钮捕获数据了。捕获数据保存中，Wireshark 的图标显示为绿色，如图 3.2 所示。

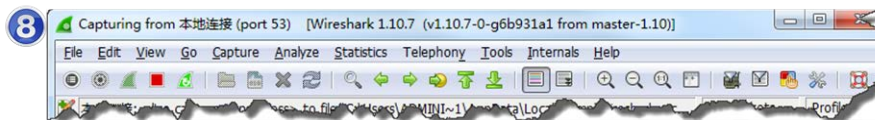


图 3.2 Wireshark 运行界面

3.2 选择捕获位置

使用 Wireshark 分析网络数据时，首先要确认 Wireshark 捕获数据的正确位置。如果没有在正确的位置启动 Wireshark，则导致用户可能花费很长的时间处理一些与自己无关的数据。所以在使用 Wireshark 之前，需要确认它的位置。

如图 3.3 所示，该图代表了一个简单的网络环境。在捕获过程中，可以检测到往返延迟时间、丢包、错误信息及其它主机之间传输的问题。如果在捕获过程中，发现访问网页速度慢，这说明 Wireshark 捕获工具可能是来自通客户端。

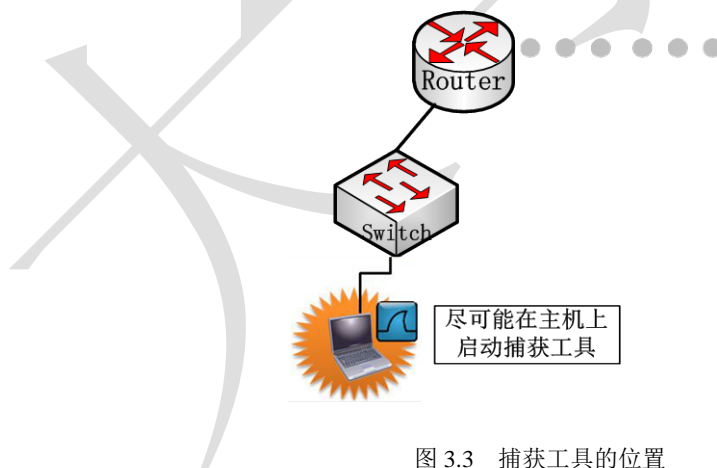


图 3.3 捕获工具的位置


当出现以上的情况时，就需要考虑将 Wireshark 捕获工具移动到其它位置。如发现大量丢包时，可以在路由器或交换机上开启 Wireshark 工具，以确定那个设备存在大量丢包。

3.3 选择捕获接口

在使用 Wireshark 捕获数据前，首先要选择捕获接口。在一台计算机上可能存在许多个

网卡，包括有线和无线网卡。Wireshark 可能无法检测到所有的本地接口和远程可用的网络接口，只能列出可用的网络接口。本节将介绍如何选择捕获接口。

3.4.1 判断那个适配器上的数据

在工具栏中单击  按钮或在菜单栏中依次选择 **Capture|Interfaces** 命令，可以快速的判断那个接口捕获数据和每个接口连接的网络。捕获接口界面，如图 3.4 所示。

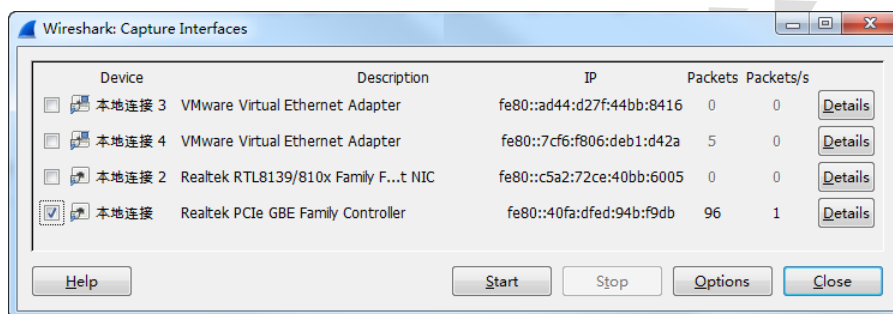


图 3.4 捕获接口

如果主机使用了双协议栈（IPv4 和 IPv6），Wireshark 默认将显示每个适配器的 IPv6 地址。如果存在 IPv4 地址，单击 IPv6 地址将可以看到 IPv4 地址（以本地连接为例），如图 3.5 所示。

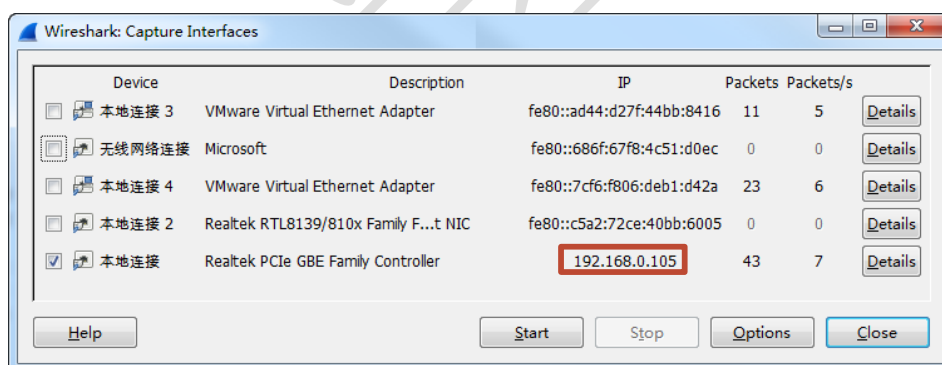


图 3.5 IPv4 地址显示

从该界面可以看到本地连接接口的 IP 由 IPv6 的地址（fe80::40fa:dfed:94b:f9db）变成了 IPv4 地址（192.168.0.105）。

如果想捕获某个接口上的数据时，只需将图 3.4 中设备前面的复选框勾上，然后单击 Start 按钮，将开始捕获该接口上的数据。

3.4.2 使用多适配器捕获

从 Wireshark 1.8 开始，可以同时捕获两个或更多个接口。如果想要同时捕获有线和无线网络数据，这个功能是有用的。例如，如果用户正试图解决在网络上的 WLAN 客户端的问题，可以同时捕获客户端的 WLAN 适配器和无线网络，如图 3.6 所示。

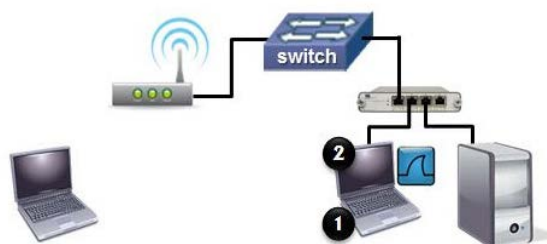


图 3.6 同时捕获有线和无线数据

3.4 捕获以太网数据

用户可以使用多种方法来捕获以太网上的数据。尽管有多种方法，但并不都是最有效的方法。最有效的捕获方法有三种，分别是直接在主机上捕获数据、映射主机的交换端口和设置一个测试访问点。下面将分别介绍这三种方法。

第 1 种：直接在主机上捕获数据

如果在主机上安装捕获工具，这可能是最好的选择。这样用户可以不用安装 Wireshark，使用一个简单的包捕获工具（如 tcpdump）就可以了，如图 3.7 所示。

第 2 种：端口映射

如图 3.8 所示，在该图中的交换机支持端口映射，并且用户有权配置交换机、设置交换机来复制所有数据到用户交换端口下的 Wireshark 端口。然而，需要注意的一个问题是交换机不会向链路层发送错误数据包，所以可以不看性能相关的所有数据。

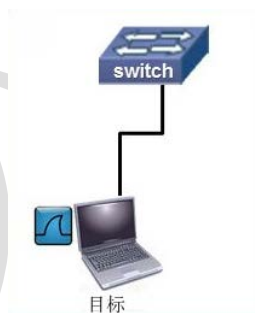


图 3.7 在主机捕获数据

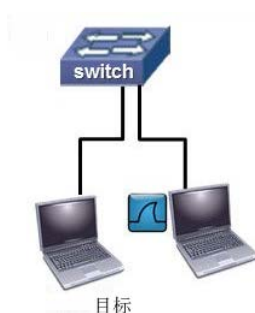


图 3.8 映射主机的交换端口

第 3 种：设置一个测试访问点（TAP）

测试访问点是全双工设备，它安装在主机和交换机之间，如图 3.9 所示。默认情况下，测试访问点向前发送所有网络数据，包括链路层错误。尽管测试访问点可能是昂贵的，如果用户想监听所有流量或来自一个主机的流量，它们可以节约大量的时间。



图 3.9 设置测试访问点

3.5 捕获无线数据

使用 Wireshark 捕获无线网络数据，可以帮助用户了解无线网络怎样工作和分析家庭网络性能慢的原因。如果要捕获无线网络数据，捕获之前需要做些准备工作。例如，确定无线局域网适配器是否正运行在 Wireshark 上。本节将介绍捕获无线网络数据。

3.5.1 捕获无线网络数据方式

无线网络数据捕获方式类似于以太网数据捕获，只是端口选择不同。下面简要介绍一下。

【实例 3-1】捕获无线局域网适配器数据。具体操作步骤如下所示：

(1) 在工具栏中依次选择 Capture|Interfaces 命令，将显示如图 3.10 所示的界面。

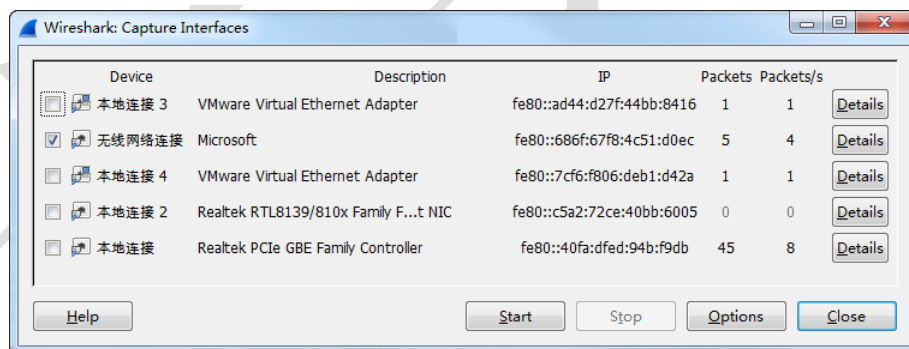


图 3.10 捕获接口

(2) 从该界面可以看到有一个无线网络适配器，在该界面选择无线网络连接接口的复选框，如图 3.10 所示。然后单击 Start 按钮，将开始捕获数据，如图 3.11 所示。

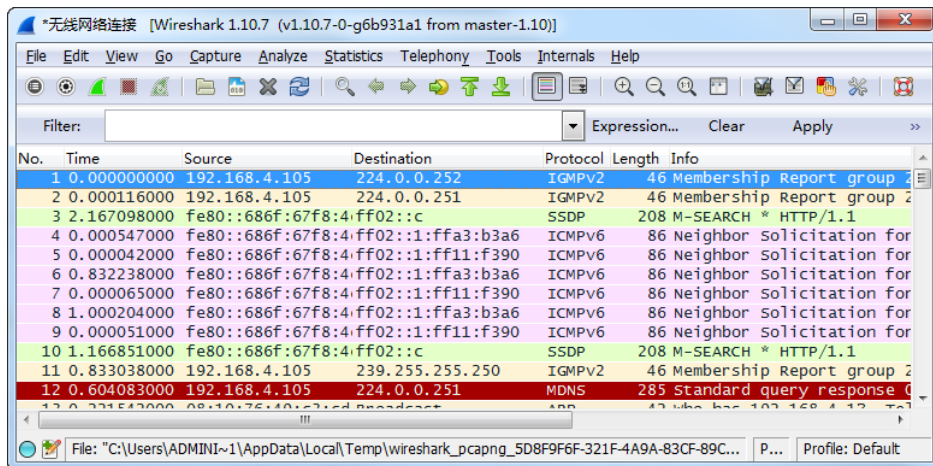


图 3.11 无线网络数据

(3) 该捕获文件中的捕获的数据都是来自无线接口上的数据。

3.5.2 使用 AirPcap 适配器

AirPcap 适配器是专门设计用于捕获所有类型的 WLAN 数据，应用 WLAN 解密密钥，并添加捕获数帧的元数据。AirPcap 适配器可以捕获 802.11 控制、管理和数据帧。此外，这些适配器运行在监听模式（也称为射频监控或 RFMON 模式），使适配器来捕获所有数据，而不必结合特定的访问点。这意味着 AirPcap 适配器可以捕获任何 802.11 网络流量，而不仅仅是一个本地主机接口上的数据。

3.6 处理大数据

在 Wireshark 的默认设置情况下，将会捕获各种协议的数据。当用户分析时，这样的大数据将会带来很大的困扰。本节将介绍如何处理这些大数据。

3.6.1 捕获过滤器

捕获过滤器是数据经过的第一层过滤器，它用于控制捕捉数据的数量，可以避免产生过大的捕获文件。这样在使用 Wireshark 捕获之前，就可以通过指定捕获过滤器获取到自己需要的数据。下面将介绍捕获过滤器的使用。

在菜单栏中依次选择 Capture|Options...命令，打开捕获选项窗口。打开界面，如图 3.12 所示。

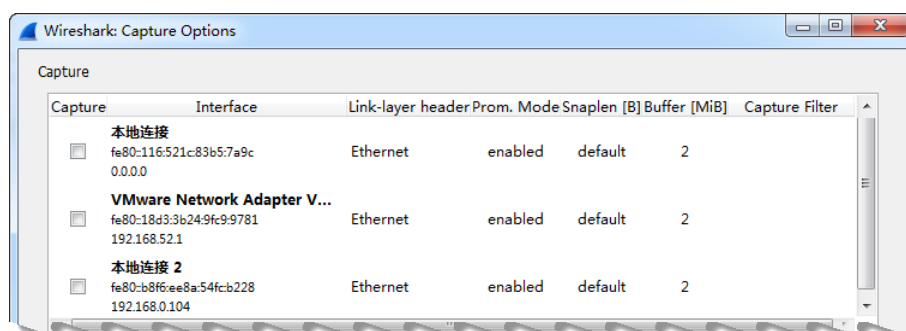


图 3.12 捕获窗口

在该界面可以看到捕获过滤器列是空白的。这是因为默认没有使用任何的过滤器。此时，双击选择接口行的任何一处，启动编辑接口设置窗口，如图 3.13 所示。

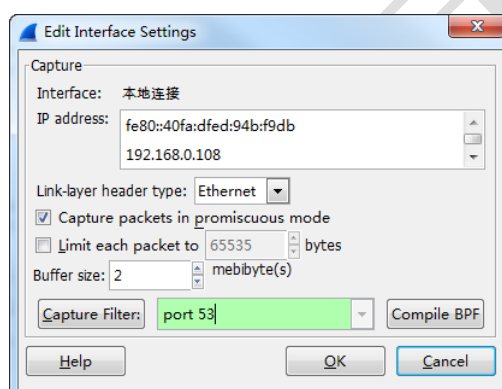


图 3.13 编辑接口设置

在该界面单击 **Capture Filter** 按钮，可以查看并选择捕获过滤器。这里选择 **port 53**，如图 3.13 所示。从该界面可以看到设置捕获过滤器后，背景颜色为绿色。通过该背景色可以判断使用的语法是否正确，如果语法错误，则背景为红色；如果正确，背景为绿色。然后单击 **OK** 按钮，将显示如图 3.14 所示的界面。

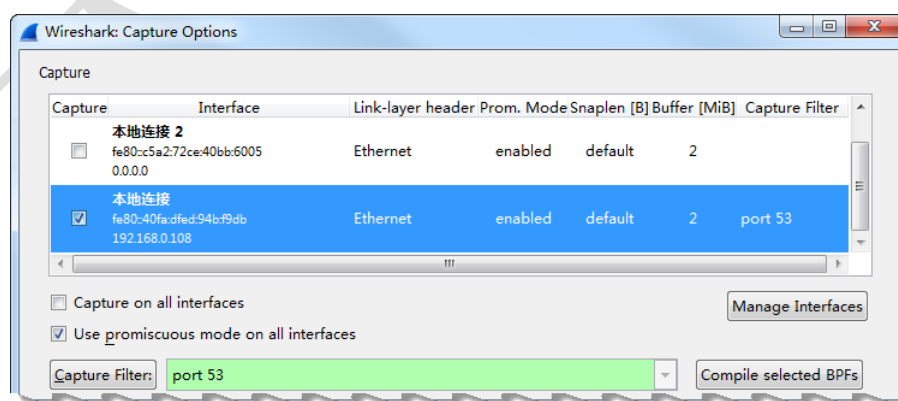


图 3.14 捕获选项

在该界面的 **Capture filter** 列可以看到，设置的捕获过滤器为 **port 53**。Wireshark 捕获过滤器使用的是伯克利数据包过滤器（Berkeley Packet Filtering）语法。用户也可以直接在捕获过滤器区域，输入捕获过滤器的语法。然后单击 **Start** 按钮，开始捕获数据。

3.6.2 捕获文件集

文件集就是多个文件的组合。在 Wireshark 中，使用文件集的方法可以将一个大文件分成好几个小文件。在捕获选项窗口中，可以设置每个文件的大小及每隔多长时间保存一个文件。这样也可以帮助用户，快速的处理数据。下面将介绍捕获文件集。

【实例 3-2】捕获文件集。具体操作步骤如下所示：

(1) 在主菜单栏中单击 （显示捕获选项）按钮，将打开如图 3.15 所示的界面。

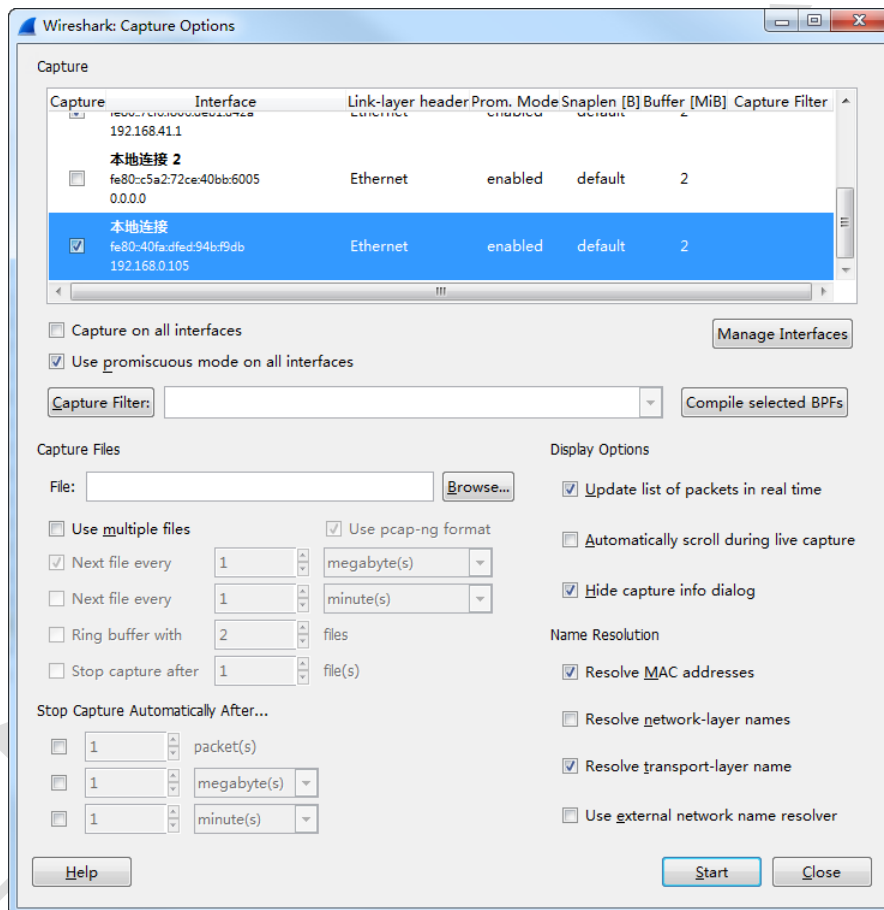


图 3.15 捕获选项界面

(2) 在该界面的 Capture 选项框中，选择连接到 Internet 网络适配器前的复选框。这里选择“本地连接”接口。

(3) 在 Capture Files 部分，单击 Browse 按钮选择保存捕获文件的路径和文件名。这里设置文件名为 capture.pcapng，如图 3.16 所示。然后单击 OK 按钮，将返回到捕获选项界面。

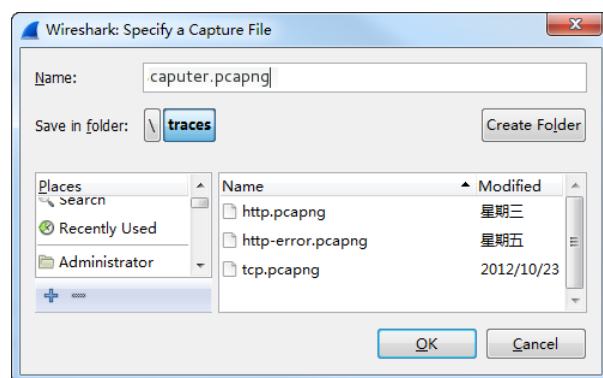


图 3.16 保存的文件

(4) 在捕获界面的 Capture Files 部分，将看到上面指定的捕获文件的路径和文件名，如图 3.17 所示。在该界面选择启用 Use multiple files 选项，并定义生成的捕获文件每个为 1MB、每 10 秒生成一个文件及捕获 4 个文件后自动停止捕获。以上信息设置完后，单击 Start 按钮，将开始数据捕获。

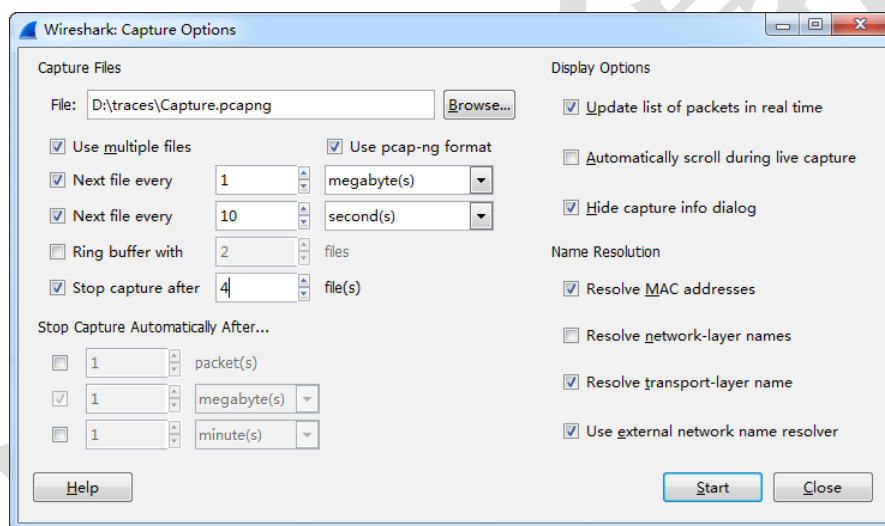


图 3.17 设置文件集

注意：在捕获选项窗口中的 Stop Capture after 选项，在某些版本中存在 Bug。在 1.10.7 中，选择该选项后，无法发挥它的作用。

(5) 现在通过访问 www.openoffice.org 网站，产生流量。大概访问几秒，然后返回到 Wireshark 查看状态栏的文件区域。将会看到文件名发送了变化，文件后面添加了文件编号（本例中是_00004）、时间和时间戳，如图 3.18 所示。

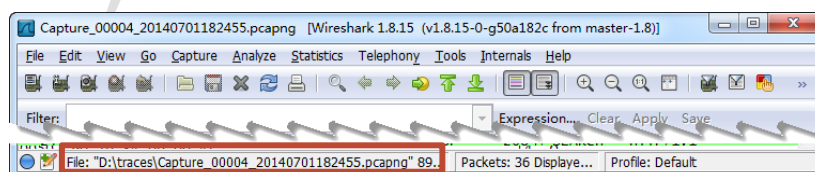


图 3.18 文件名变化

(6) 用户也可以通过在工具栏中依次选择 File|File Set|List Files 命令，查看文件集中的所有文件，如图 3.19 所示。

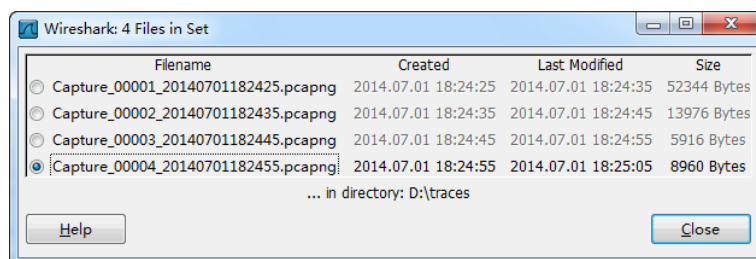


图 3.19 文件集

(7) 从该界面可以看到生成的 4 个小文件。

3.7 处理随机发生的问题

在捕获数据时，用户可能遇到一些特殊的问题。但是这些问题，并不是在每次捕获数据时都可以捕获到。所以，通常这些随机发生的问题给用户常常带来一定的困扰。在 Wireshark 中有一些特殊的功能，可以捕获到这些烦人的、难以捉摸的数据包。本节将介绍处理这些随机发生的问题。

在 Wireshark 中可以通过设置使用文件集，并且使用循环缓冲区的功能处理随机发生的问题。设置该功能后，Wireshark 会持续的捕获数据，直到问题再次出现。下面介绍设置循环缓冲区的方法。

在菜单栏中依次选择 Capture|Options 命令，打开捕获选项窗口。在该界面即可设置缓冲区文件，如图 3.20 所示。

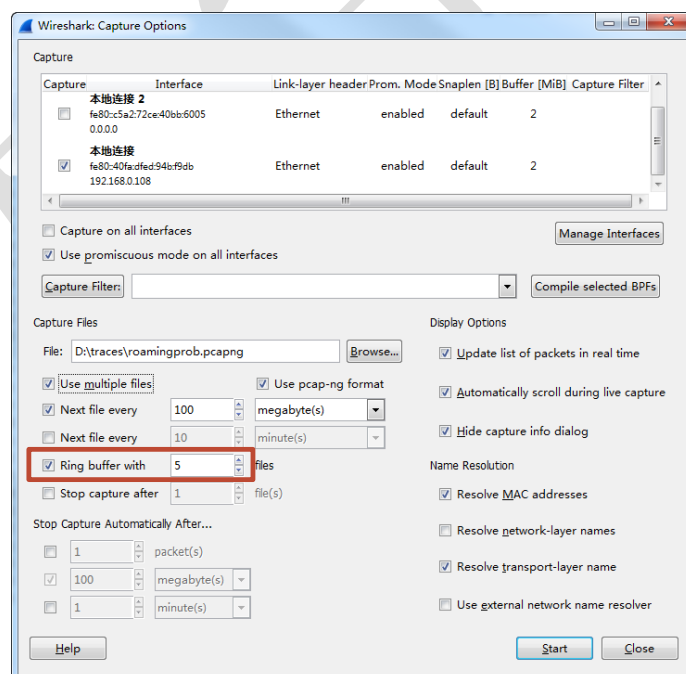


图 3.20 设置缓冲区

以上的设置表示当 Wireshark 完成捕获第 5 个 100MB 的文件后，将删除第一个 100MB

的文件，并创建第六个文件，使 Wireshark 继续运行。

【实例 3-3】使用设置循环缓冲区的方法节约磁盘空间。具体操作步骤如下所示：

(1) 在主菜单栏中单击 （显示捕获选项）按钮。

(2) 在该界面的 Capture 部分选择，连接到 Internet 网络适配器前的复选框。这里选择“本地连接”接口。

(3) 在 Capture Files 部分，单击 Browse 按钮选择保存捕获文件的路径和文件名。这里设置文件名为 capturese.pcapng，如图 3.9 所示。然后单击 OK 按钮，将返回到捕获选项界面。

(4) 在捕获界面的 Capture Files 部分，将看到上面指定的捕获文件的路径和文件名。在该界面选择启用 Use multiple files 选项，设置生成文件集中的每个文件大小为 10MB，每 30 秒生成一个文件、缓冲区最多保存 3 个文件，如图 3.21 所示。以上信息设置完后，单击 Start 按钮，将开始数据捕获。

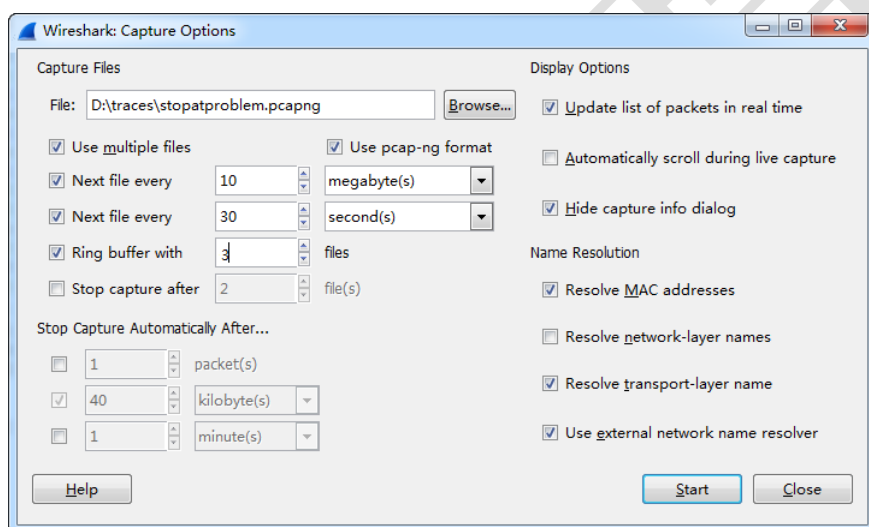



图 3.21 捕获选项

(5) 此时打开浏览器，访问 www.wireshark.org 网站产生流量。大概访问 30 秒该网站。然后再访问一下 www.chappellu.com/nothere.html 网站，将会出现 404 错误，因为该网站不存在。当出现 404 错误后，快速返回到 Wireshark 界面，单击 （停止捕获）按钮。

(7) 查看 Wireshark 的状态栏的文件区域，将看到许多文件编号已经被分配。当查看保存捕获文件目录或查看文件集时，仅能看到 3 个文件，如图 3.22 所示。因为在前面的循环缓存区设置了仅保存最后三个文件。

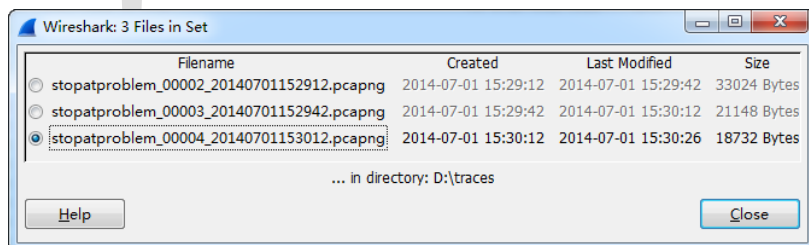


图 3.22 文件集

(8) 从该界面从文件名的编号（_00002、_00003、_00004）可以看出，目前保存的三个文件。由于缓存文件设置仅能保存 3 个文件，所以第一个文件（编号为_00001）被删除了。这样就可以节约磁盘空间。现在单击 Close 按钮，返回到 Wireshark 主界面。将能够快速的找出 404 的错误信息，如图 3.23 所示。

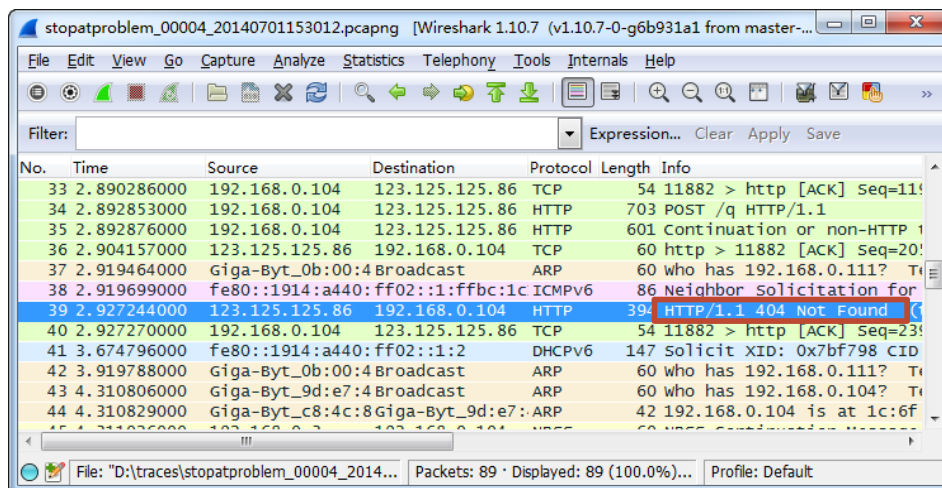


图 3.23 404 错误信息

(5) 从该界面可以看到，39 帧中的数据中包含 404 Not Found 信息。

3.8 捕获基于 MAC/IP 地址数据

在使用 Wireshark 捕获过滤器时，可以设置只捕获 MAC/IP 地址数据的过滤器。本节将介绍捕获基友 MAC/IP 地址数据的方法。

3.8.1 捕获单个 IP 地址数据

IP 地址是 IP 协议提供了一种统一的地址格式。它为互联网上的每一个网络和每一台主机分配了一个逻辑地址。通常 IP 地址分为 IPv4 和 IPv6 两大类。现在大部分都使用的是 IPv4 地址，该地址是一个 32 位的二进制数。通常在捕获数据时，用户会通过 IP 地址的方式来判断是那台主机上的数据。下面将介绍捕获单个 IP 地址数据的方法。

下面看几个 IP 地址捕获过滤器的例子。

- ☐ host 10.3.1.1: 捕获到/来自 10.3.1.1 主机的数据。
- ☐ host 2406:da00:ff00::6b16:f02d: 捕获到/来自 IPv6 地址 2406:da00:ff00::6b16:f02d 的数据。
- ☐ not host 10.3.1.1: 捕获除了到/从 10.3.1.1 的所有数据。
- ☐ src host 10.3.1.1: 捕获来自 10.3.1.1 主机上的数据。
- ☐ dst host 10.3.1.1: 捕获到达 10.3.1.1 主机上的数据。
- ☐ host 10.3.1.1 or host 10.3.1.2: 捕获到达/来自 10.3.1.1 上的数据，和到达/来自 10.3.1.2 主机的数据。
- ☐ host www.espn.com: 捕获解析 www.espn.com 的 IP 地址上的数据。

【实例 3-4】仅捕获到达/来自 192.168.0.112 主机的数据包。具体操作步骤如下所示：

(1) 在工具栏中单击 按钮，打开捕获选项界面，如图 3.24 所示。

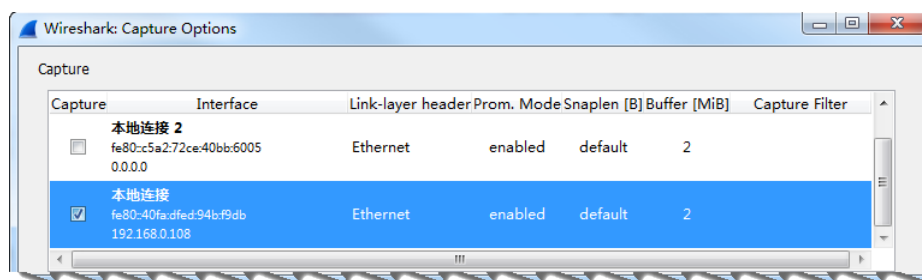


图 3.24 捕获选项

(2) 在该界面的捕获区域，选择捕获数据的接口（本地连接）的复选框。在这个捕获区域双击选择接口行的任何一处，启动编辑接口设置窗口，如图 3.25 所示。

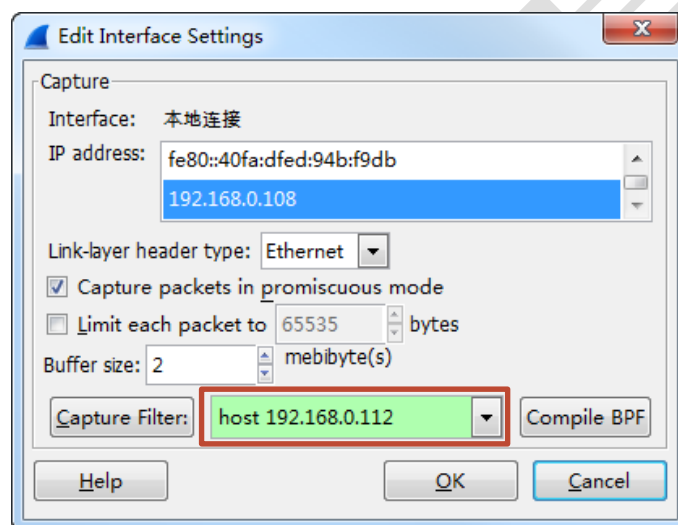


图 3.25 编辑接口设置

(3) 从该界面可以看到本地连接接口的 IP 地址，此时就可以根据该地址信息创建相应的捕获过滤器。在该界面的捕获过滤器区域，输入 host x.x.x.x (x.x.x.x 表示指定捕获的 IP 地址，本例中使用的地址是 192.168.0.112) 来过滤 IPv4 地址的数据。如果捕获 IPv6 地址的话，输入 host xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx。然后单击 OK 按钮，将看到如图 3.26 所示的界面。

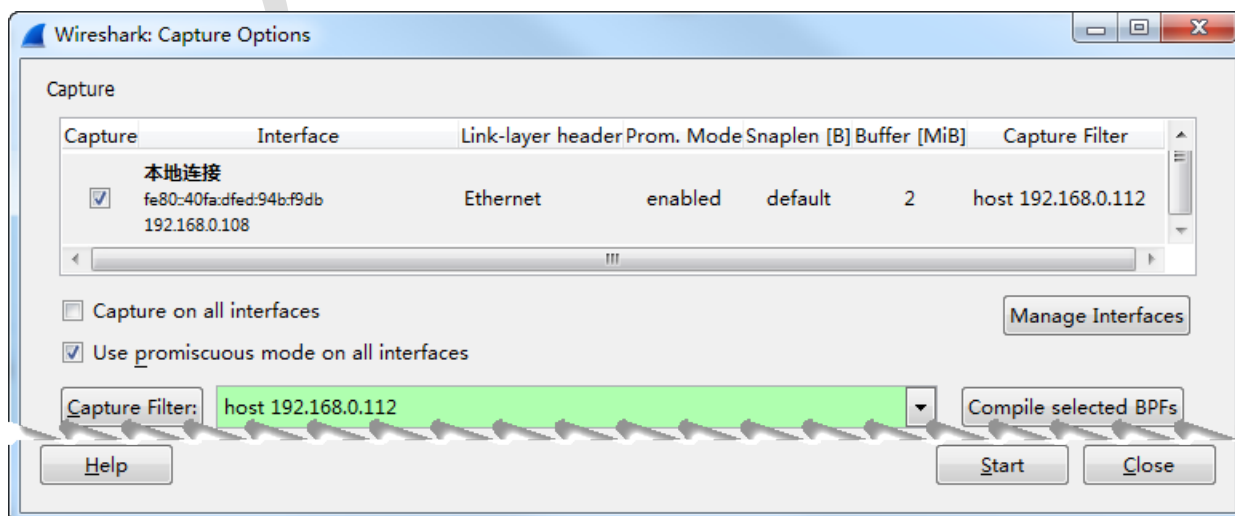


图 3.26 捕获过滤器

(4) 从该界面可以看到，本地连接接口的捕获过滤器中显示了一条信息。在该界面不要启动 Use multiple files，此时就可以捕获数据了。单击 Start 按钮，将开始捕获过程，如图 3.27 所示。

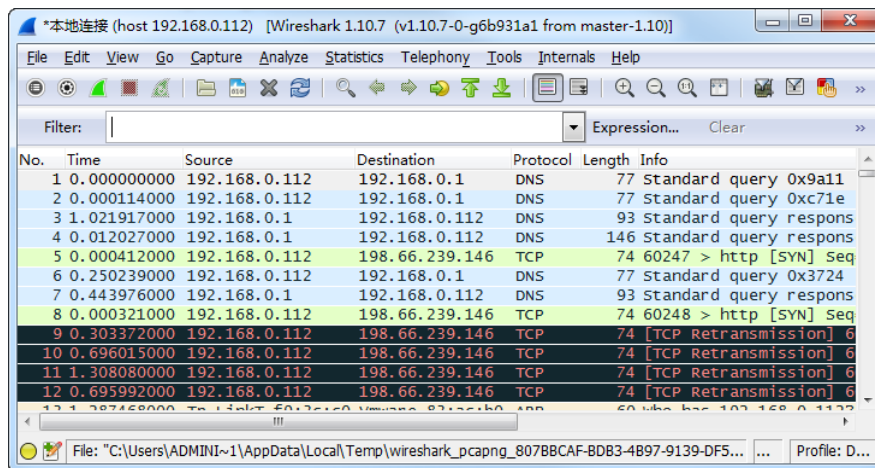


图 3.27 捕获数据过程

(5) 在 IP 地址为 192.168.0.112 主机上访问网络（如访问网站 www.chappellu.com），以产生数据。所有的数据，都将显示在图 3.27 中。

(6) 返回到 Wireshark 中查看捕获文件，所有的数据都是来自/到达 192.168.0.112 主机。

3.8.2 捕获 IP 地址范围

IP 地址一共分为 A、B、C、D、E 五类。其中，最常用的是前三类地址。IP 地址根据网络位和主机位，将其分为五类。为了节约地址，CIDR（Classless Interdomain Routing）将好几个 IP 网络结合在一起。通过使用掩码值，表示了一个 IP 地址范围。

下面看几个 IP 地址范围捕获过滤器的例子。

- ☐ net 192.168.0.0/24: 捕获到/来自 192.168.0.0 网络中任何主机的数据。
- ☐ net 192.168.0.0 mask 255.255.255.0: 捕获到/来自 192.168.0.0 网络中任何主机的数据。
- ☐ ip6 net 2406:da00:ff00::/64: 捕获到/来自 2406:da00:ff00:0000 (IPv6)网络中任何主机的数据。
- ☐ not dst net 192.168.0.0/24: 捕获除目的 IP 地址是 192.168.0.0 网络外的所有数据。
- ☐ dst net 192.168.0.0/24: 捕获到达 IP 地址为 192.168.0.0 网络内的所有数据。
- ☐ src net 192.168.0.0/24: 捕获来自 IP 地址为 192.168.0.0 网络内的所有数据。

【实例 3-5】捕获 192.168.0.0 网络中所有主机的数据。具体操作步骤如下所示：

(1) 启动 Wireshark 捕获工具。在该界面的菜单栏中依次选择 Capture|Options 按钮，打开捕获选项窗口，如图 3.28 所示。

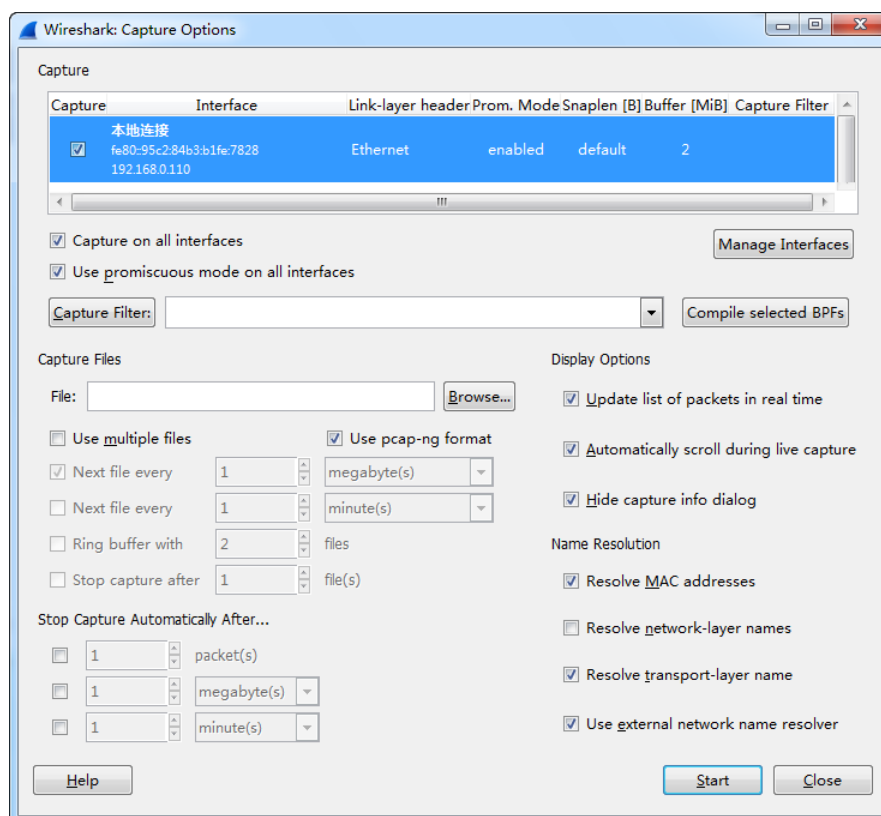


图 3.28 捕获选项窗口

(2) 在该界面过滤器区域输入捕获过滤器 `net 192.168.0.0/24`。如果要保存该捕获文件，单击 **Browse** 按钮选择保存捕获文件的位置和文件名。设置完后，如图 3.29 所示。

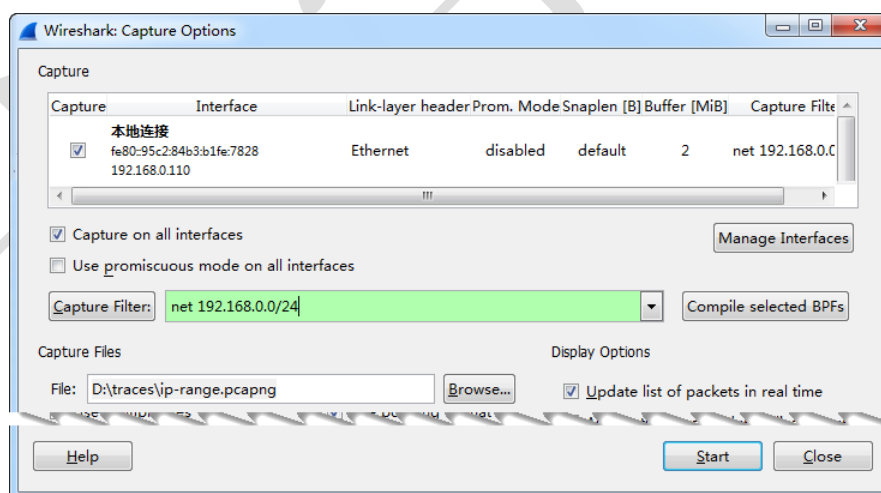


图 3.29 设置过滤器

(3) 从该界面可以看到目前设置的过滤器及文件保存位置。然后单击 **Start** 按钮，将开始捕获数据，如图 3.30 所示。

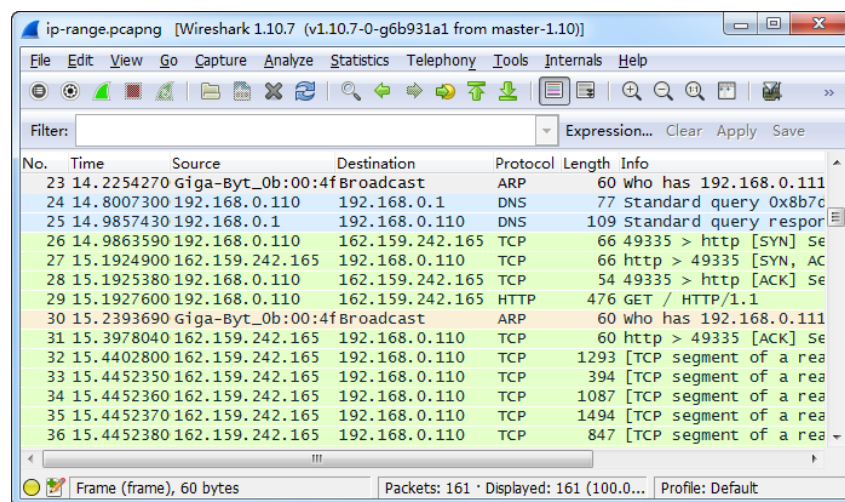


图 3.30 捕获的数据包

（4）该捕获文件中所有的数据包，都是 192.168.0.0/24 网络中主机的数据。

3.8.3 捕获广播或多播地址数据

广播地址就是当 IP 地址的网络和主机位全为 1 时，就是广播地址 255.255.255.255。该地址应用于网络内的所有主机。该地址通常用在向局域网内所有主机发送广播包时，其目的地址就是广播地址。

多播地址即组播地址，是一组主机的表示符。在以太网中，多播地址是一个 48 位的标示符。在 IPv4 中，它历史上被叫做 D 类地址，它的范围是 224.0.0.0 到 239.255.255.255。广播地址全 1 的 48 位地址，也属于多播地址。

通过监听广播和多播，可以在 Wireshark 中了解到关于网络上主机的数据。下面列出几个常用的例子，如下所示：

- ❑ ip broadcast: 捕获到 255.255.255.255 的数据。
- ❑ ip multicast: 捕获通过 239.255.255.255 到 224.0.0.0 的数据。
- ❑ dst host ff02::1: 捕获所有主机到 IPv6 多播地址的数据。
- ❑ dst host ff02::2: 捕获所有路由到 IPv6 多播地址的数据。

如果只想捕获所有 IP 或 IPv6 的数据，使用 ip 或 ipv6 捕获过滤器。

【实例 3-6】捕获广播地址数据。具体操作步骤如下所示：

- （1）启动 Wireshark 捕获工具。
- （2）在捕获窗口中设置捕获过滤器为 ip 255.255.255.255，如图 3.31 所示。

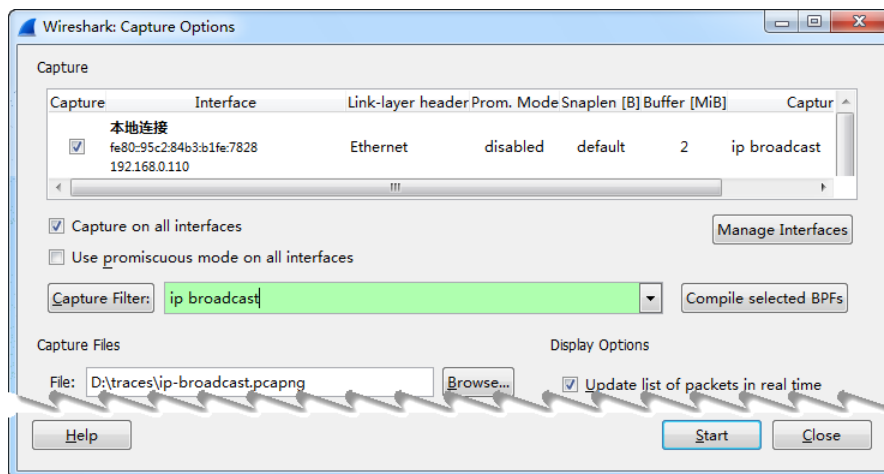


图 3.31 设置广播地址过滤器

(3) 从该界面可以看到指定的过滤器和捕获文件的位置。此时单击 **Start** 按钮，将开始捕获数据，如图 3.32 所示。

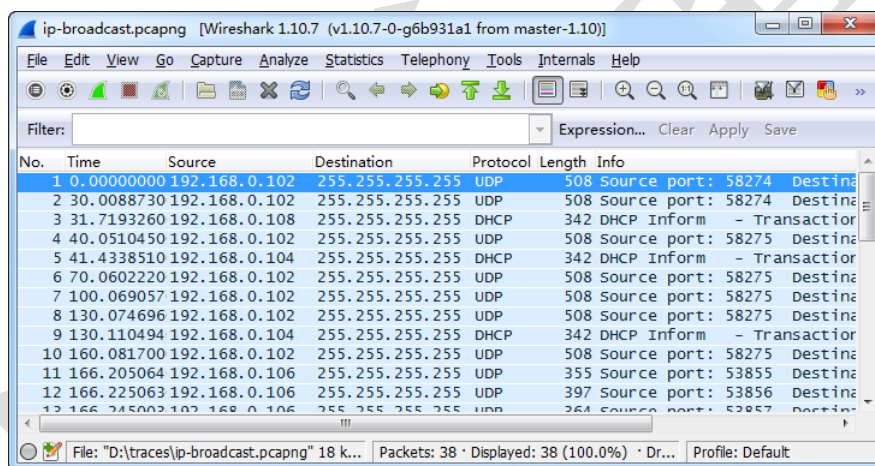


图 3.32 捕获的广播地址数据


(4) 从该界面可以看到所有数据包，都是发送给 255.255.255.255 主机的。

3.8.4 捕获 MAC 地址数据

当想要捕获到/来自一个主机 IPv4 或 IPv6 的数据时，可以创建一个基于主机的 MAC 地址捕获过滤器。自从 MAC 头部被剥去，并且通过沿着路由器的路径应用。这样确保了，网络片段和目标主机片段一样。

- ether host 00:08:15:00:08:15: 捕获到/来自 00:08:15:00:08:15 主机的数据。
- ether src 02:0A:42:23:41:AC: 捕获来自 02:0A:42:23:41:AC 主机的数据。
- ether dst 02:0A:42:23:41:AC: 捕获到达 02:0A:42:23:41:AC 主机的数据。
- not ether host 00:08:15:00:08:15: 捕获到/来自除了 00:08:15:00:08:15 的任何 MAC 地址的流量。

【实例 3-7】仅捕获到/来自其它 MAC 地址的数据。具体操作步骤如下所示：

- (1) 使用 `ipconfig` 或 `ifconfig` 命令，查看活跃接口的 MAC 地址。
- (2) 在工具栏中单击  按钮，打开捕获选项界面，如图 3.33 所示。

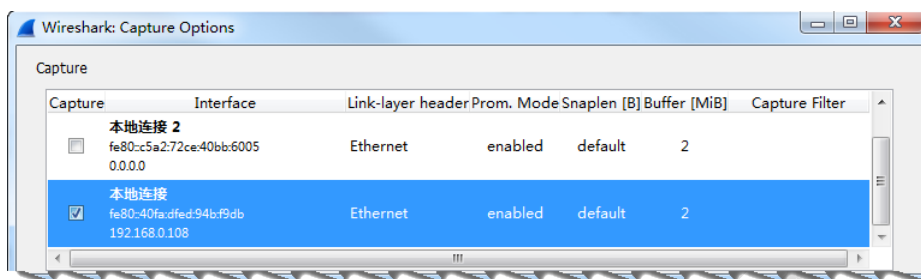


图 3.33 捕获选项

- (2) 在该界面的捕获区域，选择捕获数据的接口（本地连接）的复选框。在这个捕获区域双击选择接口行的任何一处，启动编辑接口设置窗口，如图 3.34 所示。

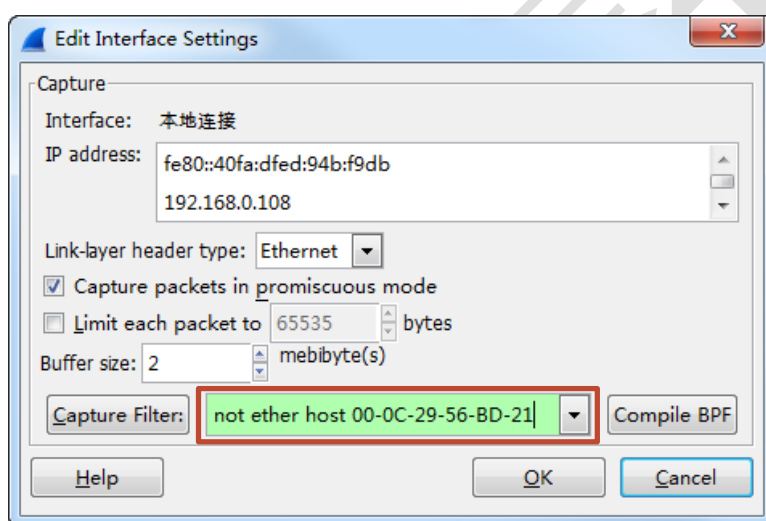


图 3.34 编辑接口设置

- (3) 在该界面输入 `not ether host xx.xx.xx.xx.xx.xx`（以太网地址），如图 3.34 所示。
- (4) 为了方便以后使用该过滤器，这里将保留此过滤器。单击 `Capture Filter` 按钮，将显示如图 3.35 所示的界面。

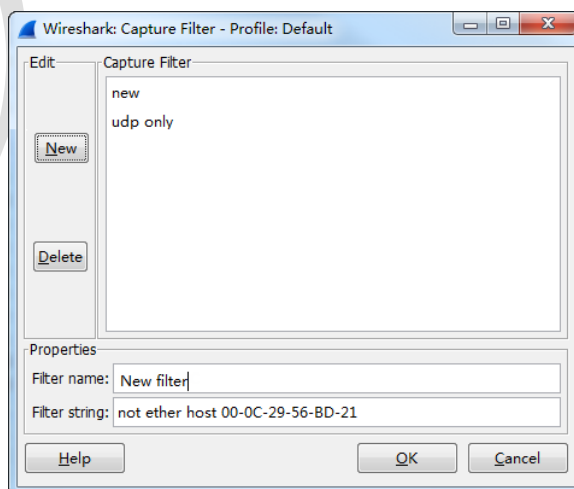


图 3.35 保存捕获过滤器

(5) 在该界面修改过滤器名字，设置为 NotMyMAC。然后单击 New 按钮，该过滤器创建成功，如图 3.36 所示。

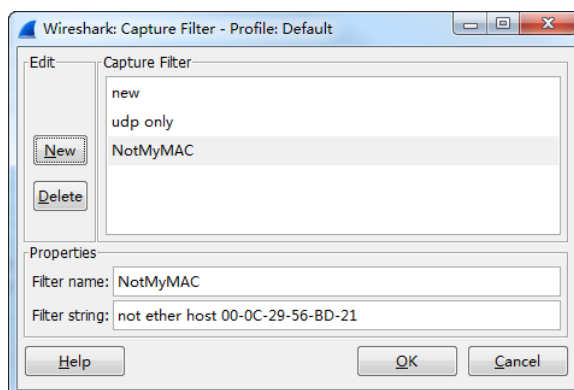


图 3.36 创建捕获过滤器

(6) 从该界面可以看到 NotMyMAC 捕获过滤器被成功的创建。此时单击 OK 按钮，将看到如图 3.37 所示的界面。

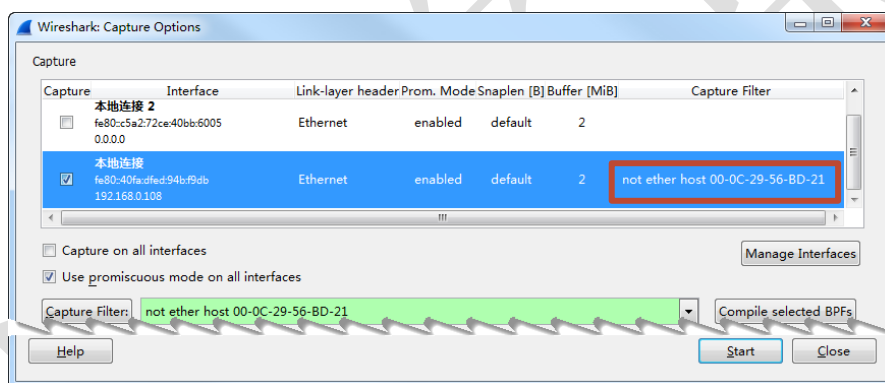



图 3.37 创建的捕获过滤器

(7) 从该界面可以看到新创建的捕获过滤器。现在单击 Start 按钮，将开始捕获。

(8) 此时，用户可以在非 MAC 地址为 00-0C-29-56-8D-21 的所有主机上进行操作。通过访问各种网站、登录服务器或发生邮件，产生主机间的数据流量。

(9) 返回到 Wireshark 主界面，单击 （停止捕获）按钮。捕获到的数据，如图 3.38 所示。

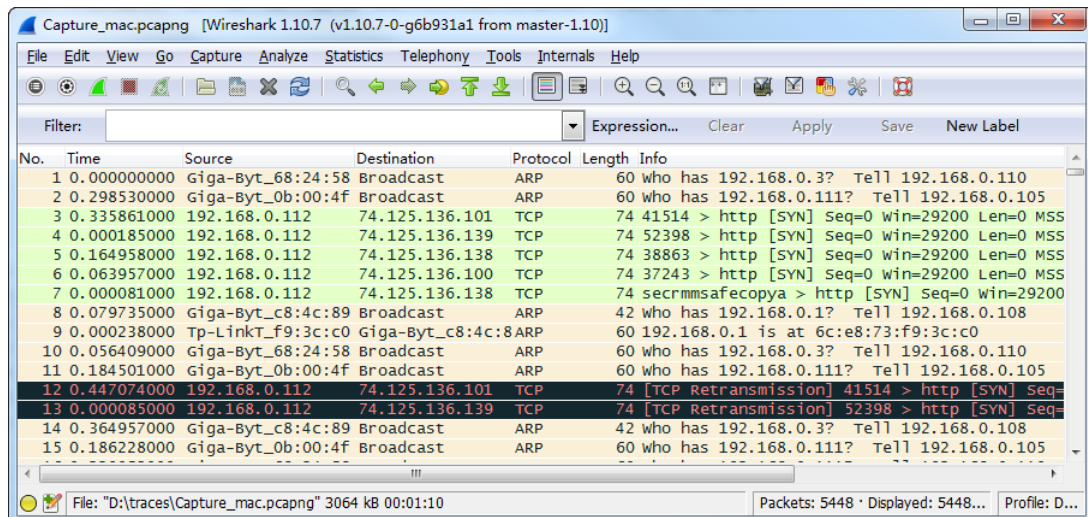


图 3.38 捕获的数据

(10) 在该界面通过滚动鼠标，查看捕获的所有数据。在该捕获文件中，将不会出现 MAC 地址为 00-0C-29-56-BD-21 主机的数据。

3.9 捕获端口应用程序数据

在 Wireshark 中想要使用捕获过滤器捕获应用程序的数据时，需要使用端口过滤器。本节将介绍捕获端口应用程序数据。

3.9.1 捕获所有端口号的数据

在网络中，大部分的应用程序都有相应的端口号，如 DNS、HTTP、FTP。下面列出了一个最常用应用程序捕获过滤器的列表。如下所示：

- ☐ port 53: 捕获到/来自端口号为 53 的 UDP/TCP 数据（典型的 DNS 数据）。
- ☐ not port 53: 捕获除到/来自端口号为 53 的所有 UDP/TCP 数据。
- ☐ port 80: 捕获到/来自端口号为 80 的 UDP/TCP 数据（典型的 HTTP 数据）。
- ☐ udp port 67: 捕获到/来自端口号为 67 的 UDP 数据（典型的 DHCP 数据）。
- ☐ tcp port 21: 捕获到/来自端口号为 21 的 TCP 数据（典型的 FTP 命令行）。
- ☐ portrange 1-80: 捕获到/来自 1-80 端口号的 UDP/TCP 数据。
- ☐ tcp portrange 1-80: 捕获到/来自 1-80 端口号的 TCP 数据。

【实例 3-8】捕获端口为 80 的所有数据包。具体操作步骤如下所示：

- (1) 启动 Wireshark 工具。
- (2) 在捕获选项窗口中设置捕获 80 端口数据的过滤器，并保存该文件，如图 3.39 所示。

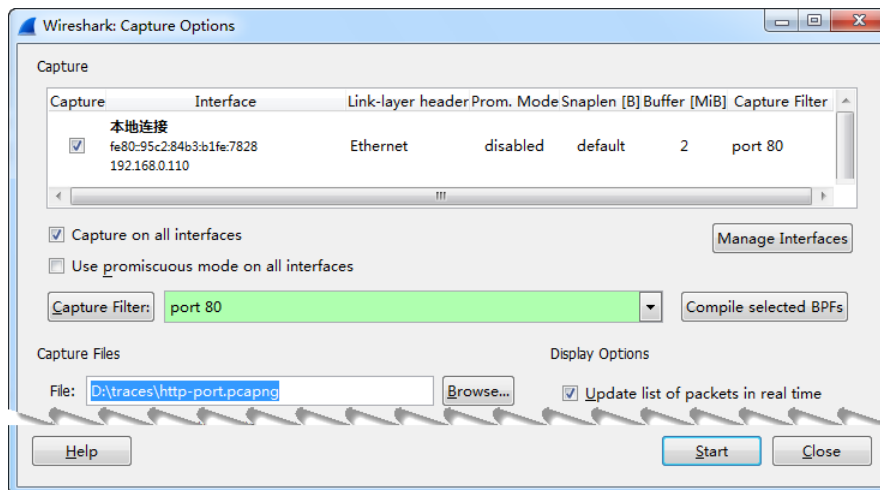


图 3.39 设置端口过滤器

(3) 从该界面可以看到设置的捕获过滤器和文件保存位置。设置完后单击 **Start** 按钮，将显示如图 3.40 所示的界面。

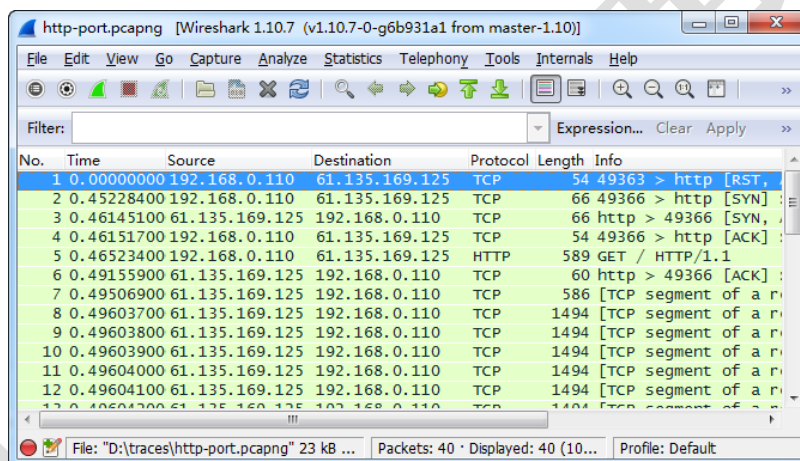


图 3.40 捕获 80 端口的数据

(4) 从该捕获文件的 **Protocol** 列可以看到所有的协议都为 **TCP** 和 **HTTP**。这两种协议的数据包，都是来自 80 端口的。

3.9.2 结合基于端口的捕获过滤器

当用户想要捕获到/来自各种非连续端口号的数据，可以通过组合各种逻辑运算符来实现。如下所示：

- ❑ port 20 or port 21: 捕获到/来自 20 或 21 端口号的所有 UDP/TCP 数据。
- ❑ host 10.3.1.1 and port 80: 捕获到/来自端口号为 80，并且是到达/来自 10.3.1.1 主机的 UDP/TCP 数据。
- ❑ host 10.3.1.1 and not port 80: 捕获到/来自 10.3.1.1 主机，并且是非 80 端口的 UDP/TCP 数据。
- ❑ udp src port 68 and udp dst port 67: 捕获来自端口为 68，目标端口号为 67 的所有 UDP 数据（典型的 DHCP 客户端到 DHCP 服务的数据）。

- ❑ udp src port 67 and udp dst port 68: 捕获来自端口号为 67，目标端口号为 68 的所有 UDP 数据（典型的 DHCP 服务器到 DHCP 客户端的数据）。

提示：尽可能不要使用捕捉过滤器。当捕获大量的数据时，可以通过使用显示过滤器过滤特定的数据。

【实例 3-9】捕获 192.168.0.110 主机上非 80 端口的数据。具体操作步骤如下所示：

- (1) 启动 Wireshark 工具。
- (2) 在捕获选项窗口中设置捕获主机 192.168.0.110 上非 80 端口数据的过滤器，并保存该文件，如图 3.41 所示。

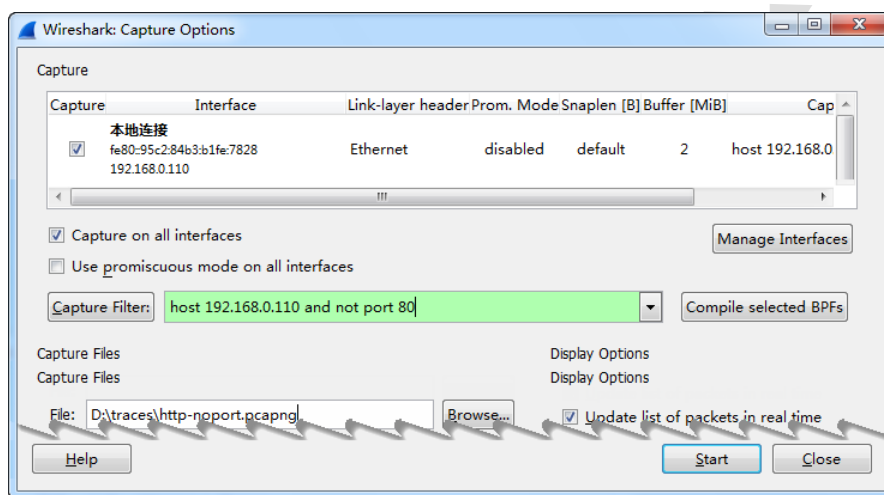


图 3.41 设置的过滤器

- (3) 在捕获过滤器区域设置捕获过滤器后，单击 Start 按钮，将显示如图 3.42 所示的界面。

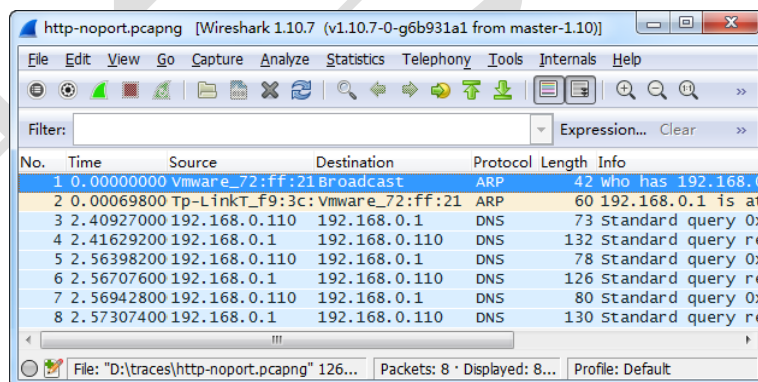


图 3.42 捕获的数据

- (4) 此时，在该捕获文件中的 Protocol 列，将不会看到有 TCP 和 HTTP 的数据。因为 TCP 和 HTTP 协议的数据包，端口号是 80。

3.10 捕获特定 ICMP 数据


互联网控制消息协议（ICMP）是一种协议。当一个网络中出现性能或安全问题时，将

会看到该协议。在这种情况下，用户必须使用一个偏移量来表示在一个 ICMP 中字段的位置。偏移量为 0 表示是 ICMP 字段类型；偏移量为 1 表示 ICMP 的位置代码字段。

下面将列出几个 ICMP 捕获过滤器的结构。

- ❑ icmp: 捕获所有 ICMP 数据包。
- ❑ icmp[0]=8: 捕获所有 ICMP 字段类型为 8 (Echo Request) 的数据包。
- ❑ icmp[0]=17: 捕获所有 ICMP 字段类型为 17 (Address Mask Request) 的数据包。
- ❑ icmp[0]=8 or icmp[0]=0: 捕获所有 ICMP 字段类型为 8 (Echo Request) 或 ICMP 字段类型为 0 (Echo Reply) 的包。
- ❑ icmp[0]=3 and not icmp[1]=4: 捕获所有 ICMP 字段类型为 3 (Destination Unreachable)，除了 ICMP 字段类型为 3/代码为 4 (Fragmentation Needed and Don't Fragment was Set) 的数据包。

【实例 3-10】创建、保存并应用一个 DNS 捕获过滤器。具体操作步骤如下所示：

(1) 在工具栏中单击  按钮，打开捕获选项界面，如图 3.43 所示。

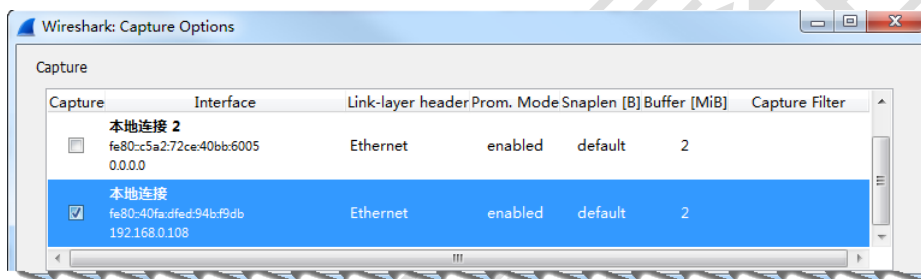


图 3.43 捕获选项

(2) 在该界面的捕获区域，选择捕获数据的接口（本地连接）的复选框。在这个捕获区域双击选择接口行的任何一处，启动编辑接口设置窗口，如图 3.44 所示。

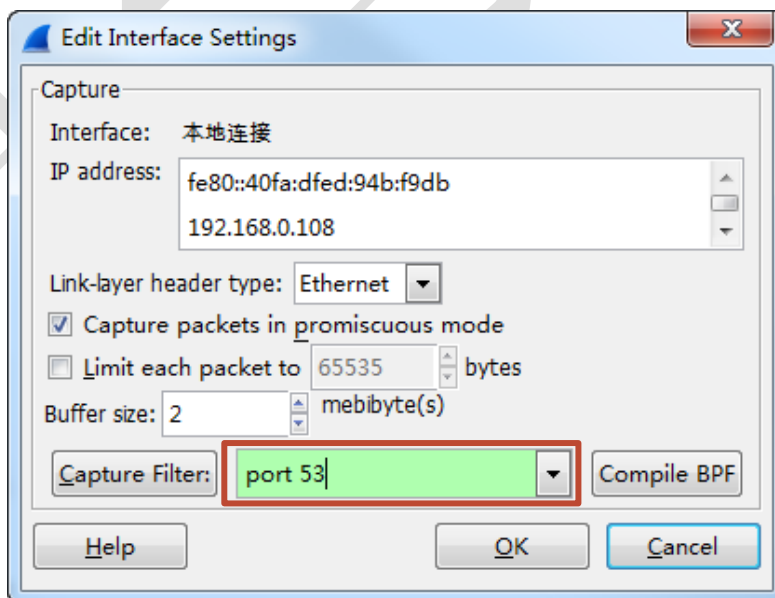


图 3.44 接口设置界面

(3) 在该界面输入 Capture Filter 对应的文本框中输入 port 53，如图 3.44 所示。此时通过单击 Capture Filter 按钮添加该捕获过滤器，如图 3.45 所示。

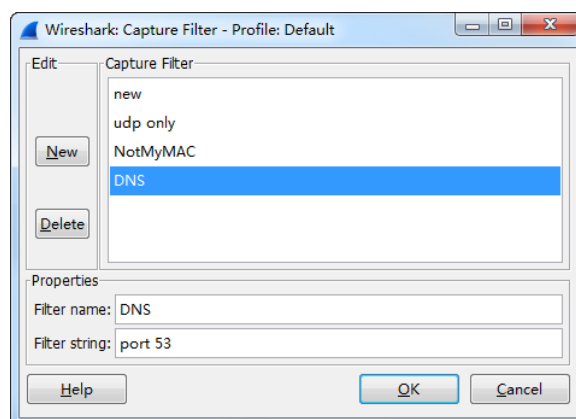


图 3.45 创建捕获过滤器

(4) 从该界面可以看到，添加的过滤器名称为 DNS。然后单击 OK 按钮，将显示如图 3.46 所示的界面。

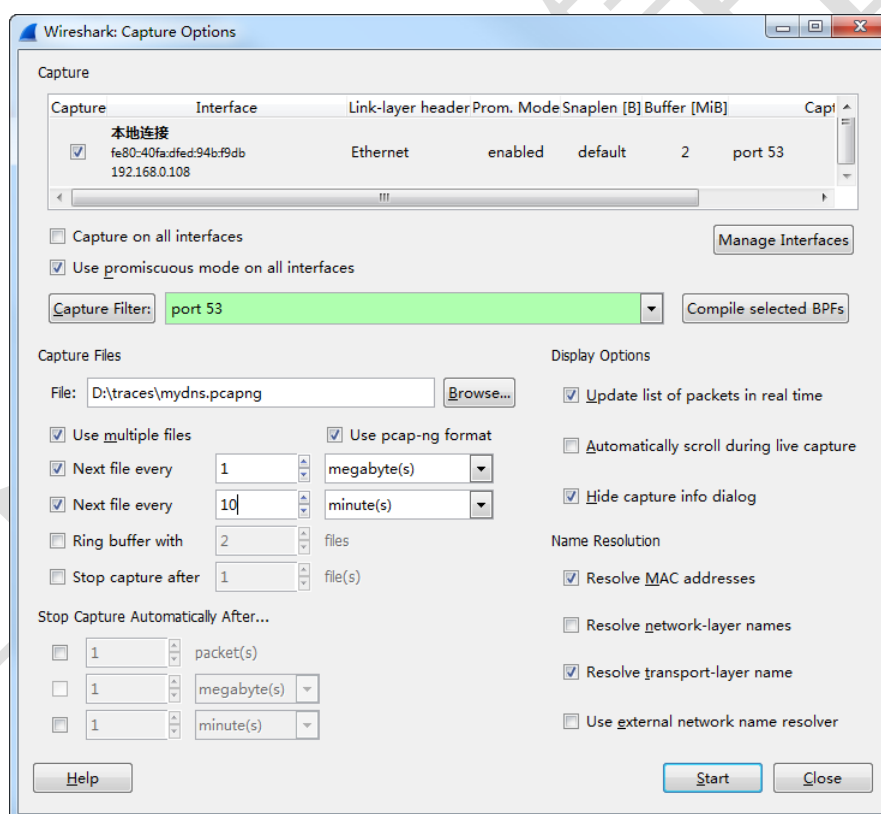

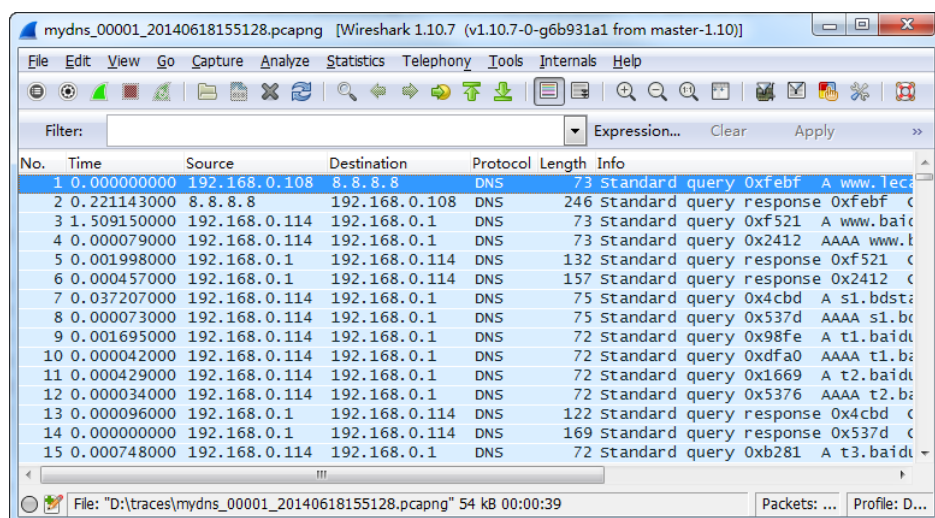


图 3.46 捕获选项

(5) 从该界面可以看到创建的捕获过滤器。在该界面指定捕获文件的位置，单击 Browse 按钮，选择并保存捕获文件。本例中设置的捕获文件为 mydns.pcapng。然后设置使用多个文件，并定义下一个生成的文件没 10 秒一个 1MB 的文件，如图 3.38 所示。单击 Start 按钮，开始捕获数据。

(6) 此时通过访问互联网上不同的网站，查看数据。最好访问最近没有访问过的网站，以确保 DNS 信息不是从缓存中加载的。

(7) 返回到 Wireshark 界面，单击 （停止捕获）按钮。显示界面，如图 3.47 所示。

A screenshot of the Wireshark network protocol analyzer interface. The title bar shows the file name 'mydns_00001_20140618155128.pcapng' and the version 'Wireshark 1.10.7 (v1.10.7-0-g6b931a1 from master-1.10)'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. Below the menu is a toolbar with various icons. A filter bar is present with the text 'Filter: Expression... Clear Apply >>'. The main packet list table is displayed with columns: No., Time, Source, Destination, Protocol, Length, and Info. It contains 15 rows of captured DNS traffic. The status bar at the bottom indicates the file path 'D:\traces\mydns_00001_20140618155128.pcapng', size '54 kB', and duration '00:00:39'.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|---------------|---------------|----------|--------|----------------------------------|
| 1 | 0.000000000 | 192.168.0.108 | 8.8.8.8 | DNS | 73 | Standard query 0xfebf A www.lect |
| 2 | 0.221143000 | 8.8.8.8 | 192.168.0.108 | DNS | 246 | Standard query response 0xfebf C |
| 3 | 1.509150000 | 192.168.0.114 | 192.168.0.1 | DNS | 73 | Standard query 0xf521 A www.baic |
| 4 | 0.000079000 | 192.168.0.114 | 192.168.0.1 | DNS | 73 | Standard query 0x2412 AAAA www.b |
| 5 | 0.001998000 | 192.168.0.1 | 192.168.0.114 | DNS | 132 | Standard query response 0xf521 C |
| 6 | 0.000457000 | 192.168.0.1 | 192.168.0.114 | DNS | 157 | Standard query response 0x2412 C |
| 7 | 0.037207000 | 192.168.0.114 | 192.168.0.1 | DNS | 75 | Standard query 0x4cbd A s1.bdsta |
| 8 | 0.000073000 | 192.168.0.114 | 192.168.0.1 | DNS | 75 | Standard query 0x537d AAAA s1.bd |
| 9 | 0.001695000 | 192.168.0.114 | 192.168.0.1 | DNS | 72 | Standard query 0x98fe A t1.baid |
| 10 | 0.000042000 | 192.168.0.114 | 192.168.0.1 | DNS | 72 | Standard query 0xdfa0 AAAA t1.ba |
| 11 | 0.000429000 | 192.168.0.114 | 192.168.0.1 | DNS | 72 | Standard query 0x1669 A t2.baid |
| 12 | 0.000034000 | 192.168.0.114 | 192.168.0.1 | DNS | 72 | Standard query 0x5376 AAAA t2.ba |
| 13 | 0.000096000 | 192.168.0.1 | 192.168.0.114 | DNS | 122 | Standard query response 0x4cbd C |
| 14 | 0.000000000 | 192.168.0.1 | 192.168.0.114 | DNS | 169 | Standard query response 0x537d C |
| 15 | 0.000748000 | 192.168.0.114 | 192.168.0.1 | DNS | 72 | Standard query 0xb281 A t3.baid |

图 3.47 捕获的数据

(8) 从该界面可以看到，所有的数据都是 DNS。此时可以通过滚动鼠标，查看捕获文件过程中访问过的网站。