9.3.1 使用 HTTP 浏览

HTTP 常被用来浏览 Web 服务器上使用浏览器访问的网页。捕获文件 http_google.pcap 就给出了这样一个使用 TCP 作为传输层协议的 HTTP 传输的例子。通信以客户端 172.16.16.128 和 Google 的 Web 服务器 74.125.95.104 的三次握手开始。

在建立了连接之后，第一个被标为 HTTP 的数据包从客户端发往服务器，如图 9-22 所示。

HTTP 数据包通过 TCP 被传输到服务器的 80 端口，也就是 HTTP 通信的标准端口（8080 端口也常被使用）。

HTTP 数据包会被确定为 8 种不同请求方法中的一种（根据 HTTP 规范版本 1.1 的定义）。这些请求方法指明了数据包发送者想要对接收者采取的动作。如图 9-22 所示，这个数据包的方法是 GET，它请求/作为通用资源标识符（Uniform Resource Indicator），并且请求版本是 HTTP/1.1。这些信息告诉我们这个客户端请求使用 HTTP 的 1.1 版本，下载 Web 服务器的根目录（/）。

接下来，主机向 Web 服务器发送关于自己的信息。这些信息包含了正在使用的用户代理（浏览器）、浏览器接受的语言（Accept Languages）和 Cookie 信息（位于捕获的底部）。为保证兼容性，服务器可以利用这些信息，决定返回给客户端的数据。

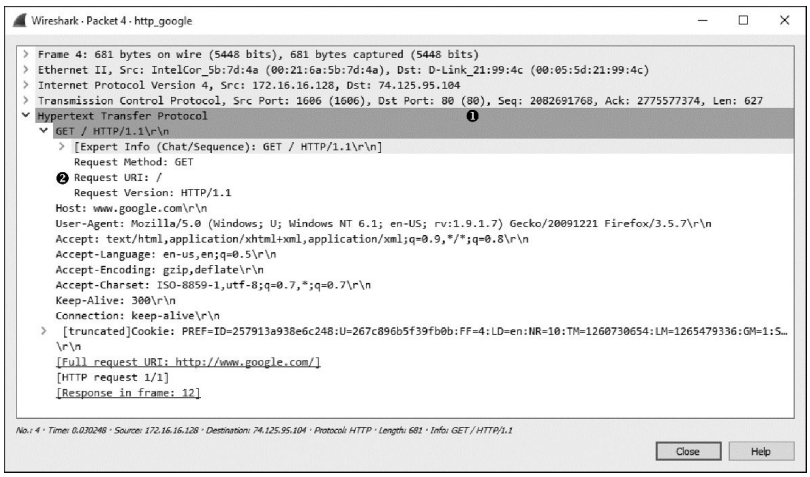
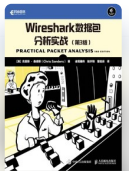


图 9-22 初始 HTTP GET 请求数据包

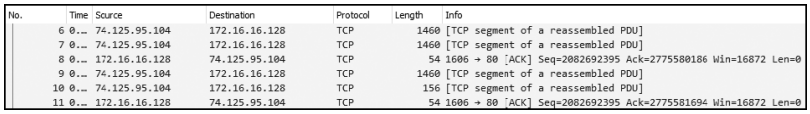


Wireshark 数据包分析实战（第 3 版）
作者：[美]克里斯·桑德斯（Chris Sander…

59%

扫码下载知

服务器在数据包 6 和 7 中发送数据，数据包 8 是来自客户端的确认，数据包 9 和 10 是另外两个数据包，数据包 11 是另外一个确认，如图 9-23 所示。虽然 HTTP 仍然负责这些传输，但所有这些数据包在 Wireshark 中都被显示为 TCP 分片而不是 HTTP 数据包。



No.	Time	Source	Destination	Protocol	Length	Info
6	0.…	74.125.95.104	172.16.16.128	TCP	1460	[TCP segment of a reassembled PDU]
7	0.…	74.125.95.104	172.16.16.128	TCP	1460	[TCP segment of a reassembled PDU]
8	0.…	172.16.16.128	74.125.95.104	TCP	54	1606 → 80 [ACK] Seq=2082692395 Ack=2775580186 Win=16872 Len=0
9	0.…	74.125.95.104	172.16.16.128	TCP	1460	[TCP segment of a reassembled PDU]
10	0.…	74.125.95.104	172.16.16.128	TCP	156	[TCP segment of a reassembled PDU]
11	0.…	172.16.16.128	74.125.95.104	TCP	54	1606 → 80 [ACK] Seq=2082692395 Ack=2775581694 Win=16872 Len=0

图 9-23　客户端浏览器和 Web 服务器之间在使用 TCP 传输数据

在数据传输结束后，数据的重组装流就已经被发送完了，如图 9-24 所示。

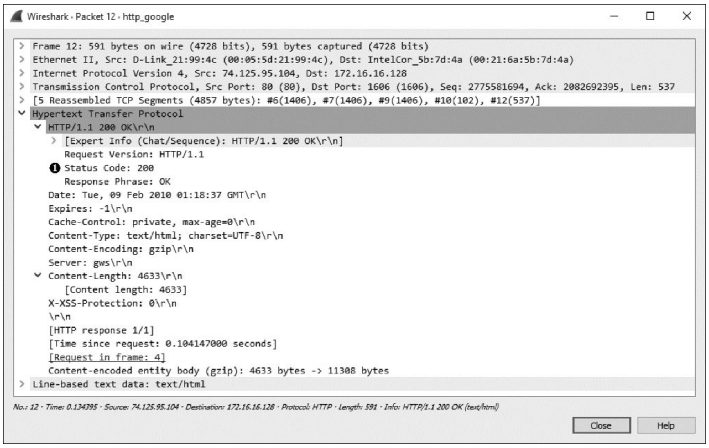


图 9-24　最后有着相应码 200 的 HTTP 数据包

> 注意
>
> 在许多情况下，当浏览包列表时，你无法看到可读的 HTML 数据，因为这些数据被 gzip 压缩以提高带宽效率，这是由 Web 服务器的 HTTP 响应中的内容偏码字段表示的。只有查看完整的流时，数据才能被解码并易于读取。

HTTP 使用了一些预定义的相应码，来表示请求方法的结果。在这个例子中，我们看到一个带有 200 响应码的数据包，表示一次成功的请求方法。这个数据包同样包含一个时间戳，以及一些关于 Web 服务器内容编码和配置参数的额外信息。当客户端接收到这个数据包后，这次处理便完成了。