## 8.1.2 TCP 端口

所有 TCP 通信都会使用源端口和目的端口，而这些可以在每个 TCP 头中找到。端口就像是老式电话总机上的插口。一个总机操作员会监视着一个面板上的指示灯和插头，当指示灯亮起的时候，他就会连接这个呼叫者，问她想要和谁通话，然后插一根电缆将她和她的目的位置连接起来。每次呼叫都需要有一个源端口（呼叫者）和一个目的端口（接收者）。TCP 端口大概就是这样工作的。

为了能够将数据传输到远程服务器或设备的特定应用中去，TCP 数据包必须知道远程服务所监听的端口。如果你想要试着连接其他端口，那么这个通信就会失败。

这个序列中的源端口并不十分重要，所以可以随机选择。远程服务器也可以很轻易地从发送过来的原始数据包中得到这个端口（见图 8-2）。

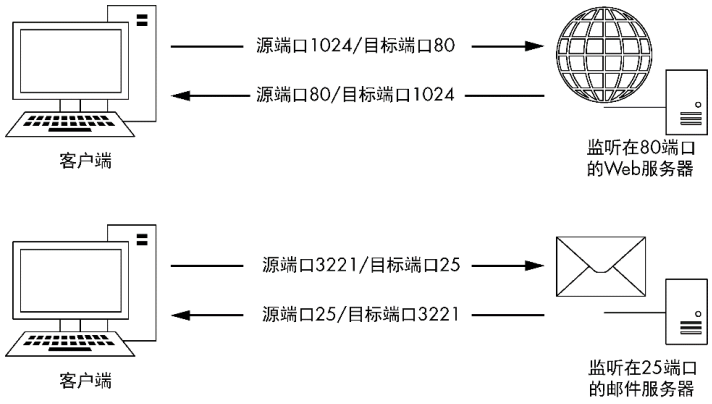


图 8-2　TCP 使用端口传输数据

在使用 TCP 进行通信的时候，我们有 65535 个端口可供使用，并通常将这些端口分成两个部分。

1～1023 是标准端口组（忽略掉被预留的 0），特定服务会用到这些通常位于标准端口分组中的标准端口。

1024～65535 是临时端口组（尽管一些操作系统对此有着不同的定义），当一个服务想在任何时间使用端口进行通信的时候，现代操作系统都会随机地选择一个源端口，让通信使用唯一源端口。这些源端口通常就位于临时端口组。

让我们打开文件 tcp_ports.pcapng，看一下一些不同的 TCP 数据包，并识别出它们所使用的端口号。在这个文件中，我们会看到一个客户端在浏

览两个网站时产生的 HTTP 通信。正如前面所提到的 HTTP 使用 TCP 进行通信，这将是一个非常典型的 TCP 流量案例。

在这个文件中的第一个数据包中（见图 8-3），一开始的两个值代表着这个数据包的源端口和目的端口。这个数据包从 172.16.16.128 发往 212.58.226.142，它的源端口是属于临时端口组的 2826（需要记住的是，源端口是由操作系统随机选取的，尽管它们可能会在随机选择的过程中选择递增策略）。目的端口是一个标准端口——80 端口。这个标准端口正是提供给使用 HTTP 的 Web 服务器使用的。

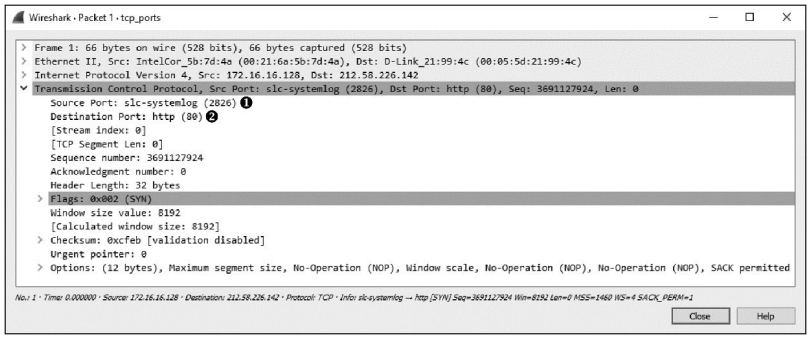


图 8-3　在 TCP 头中可以找到源端口和目标端口

你可能会注意到 Wireshark 将这些端口打上了 slc-systemlog（2826）和 http（80）的标签。Wireshark 会维护一个端口的列表，并记录它们普遍的应用。虽然列表还是以标准端口为主，但很多临时端口也关联着常用的服务。这些端口的标签可能会让人迷惑，所以一般来说最好通过关闭传输名称解析来禁用它。选择 Edit -> Preference -> Name Resolution，然后取消勾选 Enable Transport Name Resolution 就可以将其禁用了。如果你希望保留开启这个功能但改变 Wireshark 对于每一个端口的识别，则可以通过改变 Wireshark 程序目录下的 Services 文件来实现。这个文件是根据互联网数字分配机构（Internet Assigned Numbers Authority, IANA）的通用端口列表编写的（要想了解如何编辑名称解析文件，请回到第 5 章的 5.3.3 小节）。

第二个数据包是由 212.58.226.142 发往 172.16.16.128 的（见图 8-4）。除了 IP 地址之外，源端口和目的端口也同样有所改变。

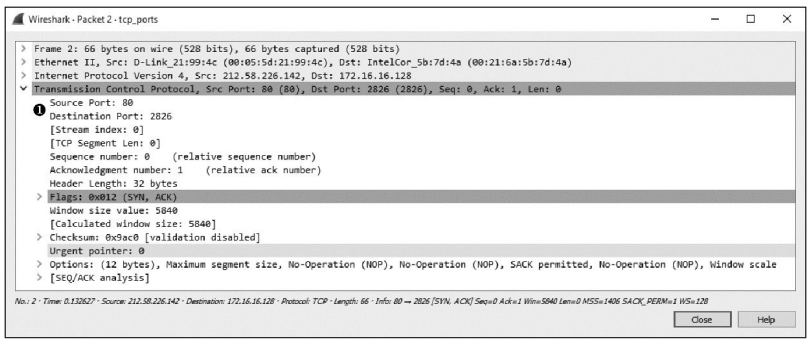

图 8-4　在反向通信中，源端口和目标端口进行了互换

所有基于 TCP 的通信都以相同的方式工作：选择一个随机的源端口与一个已知的目的端口进行通信。在发出初始数据包之后，远程设备就会与源设备使用建立起的端口进行通信。

在这个样例捕获文件中还有另外一个通信流，你可以试着找出它通信时使用的端口。

> 注意
>
> 随着这本书的深入，你将会了解更多与通用协议和端口相关联的端口，并且最终可以通过端口来识别使用它们的服务和设备。如果希望查阅详细的通用端口列表，可以在 Wireshark 的系统目录里访问「services」文件。