9.4.3 使用 SMTP 发送附件

在设计 SMTP 计时从未计划使其成为一种传输文件的途径，但由于使用邮件发送文件的便捷性，因此它成为了很多人的首选文件共享方式。让我们使用一个简短的实例，从数据包层面分析 SMTP 的文件传输过程。

在抓取文件 mail_sender_attachment.pcapng 中，用户使用客户端（172.16. 16.225）向同一网络内另一个用户发送邮件，本地 SMTP 服务器位于 172.16.16.221。这封邮件包含一些文本内容，以及一个图片文件附件。

使用 SMTP 发送附件与发送文本没有太多区别。它们都只是向服务器发送数据；尽管过程中经常会使用一些特殊编码，我们仍使用 DATA 命令进行数据传送。请打开抓包文件，跟踪 SMTP 传输的 TCP 流，查看这一操作过程。TCP 流如图 9-35 所示。

本例的通信过程在开始部分与之前的场景类似，包括服务识别和可用协议信息交换。当客户端准备传输邮件信息时，它会提供发件方地址和收件方地址，并发送DATA命令，通知服务器分配用于接收邮件数据的缓冲区。从这部分开始出现了少许差异。

在之前的例子中，客户端将文本直接传输至服务器，然后传输完成。在本例中，除了明文文本信息之外，客户端还需要发送图片附件的二进制数据。为了实现这个目的，客户端将内容类型标记为`multipart/mixed`，以`------------050407080301000500070000`❶ 作为文本信息和二进制数据的分界线。这告知服务器，传输的邮件内容包含多种类型的数据，每种数据类型有特定的 MIME 类型和编码方式，各种类型的数据使用指定的边界值分隔。通过这种机制，当另一个邮件客户端接收邮件时，基于分界线和每个数据块内指定的 MIME 类型、编码方式，接收端能够知道如何解析邮件数据。

在本例中，邮件数据包含两部分。第一部分是邮件文本，内容类型为 text/plain❷。在此之后，我们能看到一个分隔标记和第二部分的起始 ❸。第二部分包含图片文件，内容类型为`image/jpeg`❹。同样值得注意的是，`Content-Transfer-Encoding`值设为`base64`❺，这表示数据需要使用 base 64 解码。余下的数据包中包含编码过的图片文件 ❻。

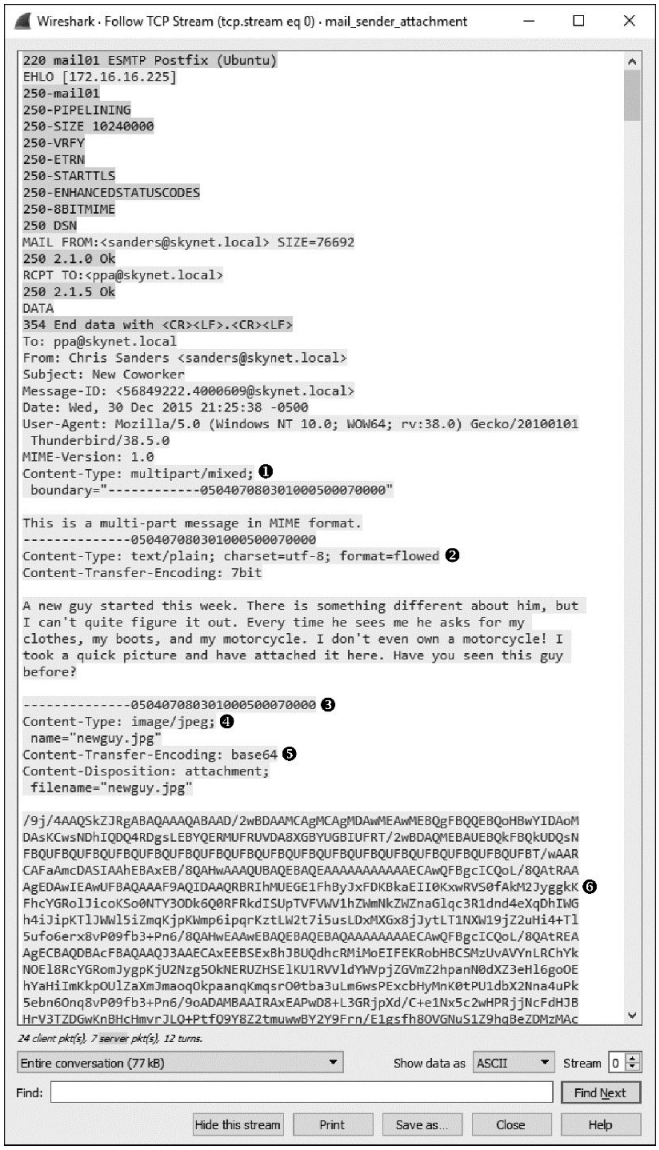

图 9-35　用户使用 SMTP 发送附件

在任何情况下，都不要将编码方式和加密方式弄混。Base 64 编码几乎能够被瞬间解码，任何截获这一通信的攻击者都能够毫不费力地获得此图片文件。如果你想要自行将图片文件从抓包文件中分离出来，在第 12 章的远程接入木马部分，有一个类似的场景——从基于 HTTP 的文件传输中分离一个图片；阅读了该章节后，回到本例的数据包分析过程，你可以尝试找出这个发件人的神秘的新同事是谁。