12.2.2 会话劫持

现在你了解了如何恶意使用 ARP 缓存投毒。我现在将演示一种利用 ARP 缓存投毒的技术：会话劫持。在会话劫持中，攻击者窃取一个 HTTP 会话 cookie，然后伪装为另一个用户；我们将很快对 HTTP cookie 进行学习。为了达到这个目的，攻击者使用 ARP 缓存投毒截获一次目标通信，并找到相关的会话 cookie 信息。随后，攻击者能够使用窃取的 cookie，以对应用户的身份访问目标 Web 应用。

这个场景的通信数据在 sessionhijacking.pcapng 中。抓包文件包含目标（172.16.16.164）与 Web 应用（172.16.16.181）之间的传输数据包。用户在不知情的情况下成为攻击受害者，他们的通信过程被攻击者（172.16.16.154）主动监听。这些数据包在 Web 服务器上被抓取，这与会话劫持发生时，防御者的角度一致。

> 注意
>
> 本例中访问的 Web 应用是 Damn Vulnerable Web Application（DVWA）。此应用预留了很多可被多种攻击方式利用的漏洞，经常被作为教学工具使用。

抓取的通信过程基本由两个会话组成。第一个会话在目标用户和 Web 服务器之间进行，使用过滤器`ip.addr == 172.16.16.164 && ip.addr == 172.16.16.181`分离此会话。这是一次正常的网页浏览通信，并没有特别之处。出于特殊的目的，我们主要关注请求中的 cookie 值。例如，在数据包 14 的 GET 请求中，你会在数据包详情窗口中发现 cookie，如图 12-12 所示。此处，cookie 使用 PHPSESSID 值`ncobrqrb7fj2a2sinddtk567q4`❶ 标识会话 ID。

图 12-12　查看目标用户的会话 cookie

网站使用 cookie 维护每个用户的会话信息。当一个新用户访问网站时，用户将使用一个会话 ID 用于身份识别（本例中为 PHPSESSID）。在用户验证过程中，很多应用在用户使用会话 ID 完成验证后，在数据库中创建相应的记录，将会话 ID 作为已验证的会话凭证。任何使用这一 ID 的用户都能够使用此次验证记录接入应用。当然，开发者愿意相信，只有一个用户能够使用某一个特定的 ID，因为生成方式保证 ID 是独一无二的。然而，这种处理会话 ID 的方式是不安全的，因为恶意用户能够窃取其他用户的 ID，然后伪造身份。目前有一些方法可以用于防止会话劫持的发生，但是很多网站，包括 DVWA，仍然能够被会话劫持。

受害者没有意识到他们的通信正在被监听，或是发现他们的会话 cookie 被攻击者截取，如图 12-12 所示。现在，攻击者只需要使用窃取的 cookie 值即可与 Web 服务器通信。可以通过某些代理服务器完成这项工作，不过，使用浏览器插件会更简单，如 Chrome 的 Cookie 管理器。利用这个插件，攻击者可以将 PHPSESSID 设置为从以上通信获取的值，如图 12-13 所示。

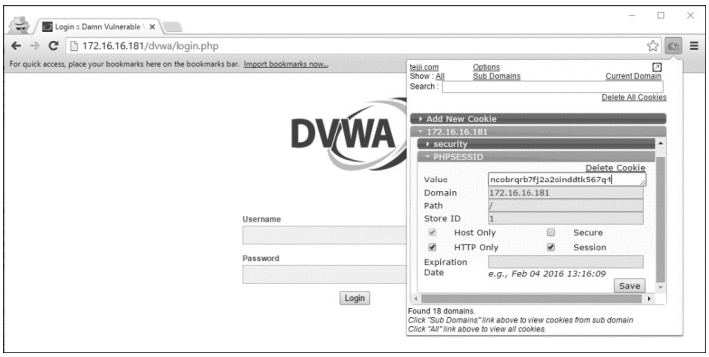

图 12-13　使用 Cookie 管理器插件盗用受害者身份

如果你清除了之前设置的筛选条件，并浏览数据包，就会看到攻击者的 IP 地址与 Web 服务器的通信。你可以将筛选条件设为`ip.addr == 172.16.16.154 && ip.addr ==172.16.16.181`来限定查看范围。

在进一步研究之前，让我们添加一个栏目，用于在数据包列表面板中显示 cookie 值。如果在 ARP 缓存投毒的过程中添加了一些栏目，请先将它们删除。之后，按照在 ARP 缓存投毒章节提到的操作步骤新增栏目，栏目的字段名为`http.cookie_pair`。添加完成后，将此栏目放在目的地址栏之后。界面看起来如图 12-14 所示。

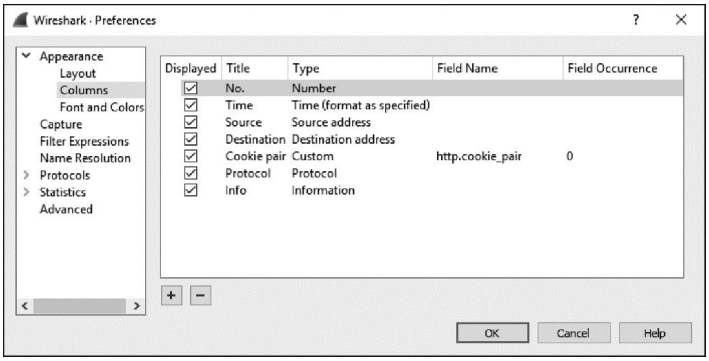

图 12-14　设置用于研究会话劫持的栏目

新的栏目设置完成后，修改筛选条件，仅显示 HTTP 请求，此处 TCP 通信没有用处。新的筛选条件为 `(ip.addr==172.16.16.154 && ip.addr== 172.16.16.181) && (http.request.method || http.response.code)`。过滤后的数据包如图 12-15 所示。

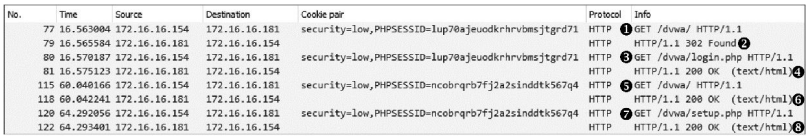

图 12-15　攻击者伪装为受害者

现在，我们查看攻击者和服务器之间的通信。在前 4 个数据包中，攻击者请求/dvwa/目录 ❶，接收到的响应状态码为 302；Web 服务器响应 302 表示请求重定向至其他 URL。此时，攻击者被重定向至登录页面/dvwa/login.php❷。攻击者的计算机请求登录页面 ❸，并返回为请求成功 ❹。两次请求均使用会话 ID lup70ajeuodkrhrvbmsjtgrd71。

随后，再次请求/dvwa/目录，我们注意到现在会话 ID❺ 发生了变化。会话 ID 现在是 ncobrqrb7fj2a2sinddtk567q4，与之前受害者使用的相同。这表明，攻击者操纵会话，使用了窃取的 ID。此时我们并没有被重定向至登录页面，请求返回了 HTTP 200 状态码，页面内容与登录后的受害者看到的一致 ❻。攻击者使用了受害者的 ID dvwa/setup.php❼，页面内容同样返回成功 ❽。攻击者和验证成功的受害者一样访问 DVWA 网站。这一过程中我们并不知道受害者的用户名或密码。

这只是攻击者将数据包分析变为攻击工具的一个例子。通常我们认为，当攻击者能够看到与通信过程相关的数据包时，将为恶意活动提供发生的可能性。这是安全专家提倡通过加密保护数据传输的原因之一。