12.3.1 极光行动

2010 年 1 月，极光行动利用了一个当时未知的 IE 浏览器漏洞。这个漏洞允许攻击者取得 Google 及其他公司的目标机器的 root 级别的远程控制权限。

用户只需要用包含该漏洞的 IE 浏览器访问一个网站，就能执行这个恶意代码。然后，攻击者就能立刻获得用户机器的管理员权限。攻击者使用了「网络钓鱼」引诱受害者。所谓「网络钓鱼」（Spear phishing）是指攻击者向受害者发送一封电子邮件，诱导受害者单击邮件中的链接，从而将其引导至一个恶意网站。通常，网络钓鱼信息看起来都像来自可信的源，因此它们经常能成功。

在极光行动的案例中，我们从用户单击钓鱼邮件中链接的那一刻开始讲起。这些数据包包含在文件 aurora.pcap 中。

这个捕获文件以受害者（192.168.100.206）和攻击者（192.168.100.202）之间的三次握手开始。初始化连接的目标是 80 端口，这使我们相信它是 HTTP 流量。第四个数据包证实了我们的假设，它是一个对/info 的 HTTP GET 请求❶，如图 12-16 所示。

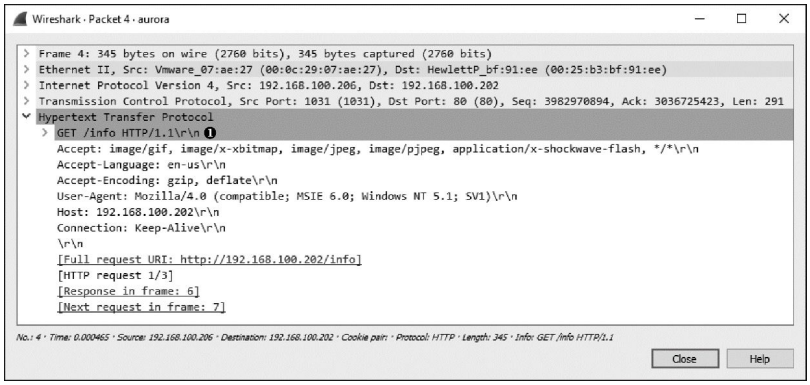

图 12-16　受害者做了一次针对/info 的 GET 请求

攻击者的机器确认收到了 GET 请求，并在数据包 6 中返回了一个 302（Moved Temporarily）码。这个状态码通常用于将浏览器重定向到另一个页面，在这个案例中正是如此。与 302 返回码 ❶ 一起来的还有一个 Location 字段，它指明重定向位置是/info? rFfWELUjLJHpP❷，如图 12-17 所示。

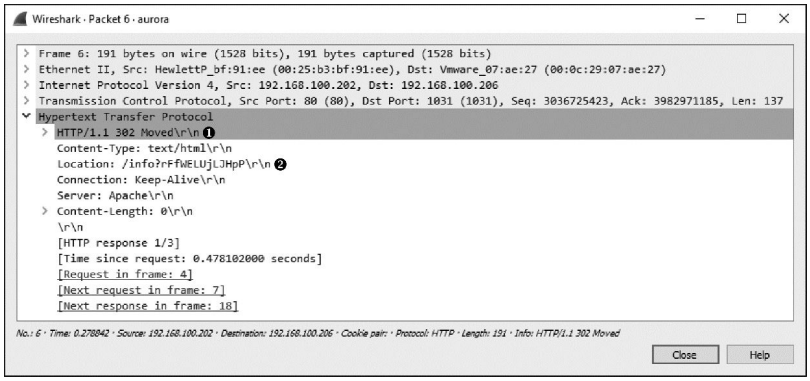

图 12-17　客户端浏览器被这个数据包重定向

收到 HTTP 302 数据包后，客户端在数据包 7 中初始化了另一个针对/info？rFfWELUjLJHpP 这个 URL 的 GET 请求，然后在数据包 8 中收到了一个 ACK。ACK 之后的几个包表示从攻击者发往受害者的数据。为了更好地查看那些数据，右击当前 TCP 流的任一个数据包，如数据包 9，选择 **Follow TCP Stream**。在这个数据流的输出中，我们看到了初始 GET 请求、302 重定向，以及第二个 GET 请求，如图 12-18 所示。

在此之后，事情变得十分奇怪了。攻击者向 GET 请求响应了一些看起来非常奇怪的内容。图 12-19 显示了这些内容的第一部分。

这些内容好像是 <script> 标签内的一系列随机数字和字母 ❶。HTML 内的 <script> 标签表示使用了一种高级脚本语言。在这个标签内，你通常会看到各种脚本语句。但这些乱码表明真正的内容可能已经被特殊编码以逃避检测了。由于我们知道这是一个漏洞利用的流量，因此我们可以假设这乱七八糟的文本包含了十六进制填充以及真正用于利用漏洞服务的 shellcode。

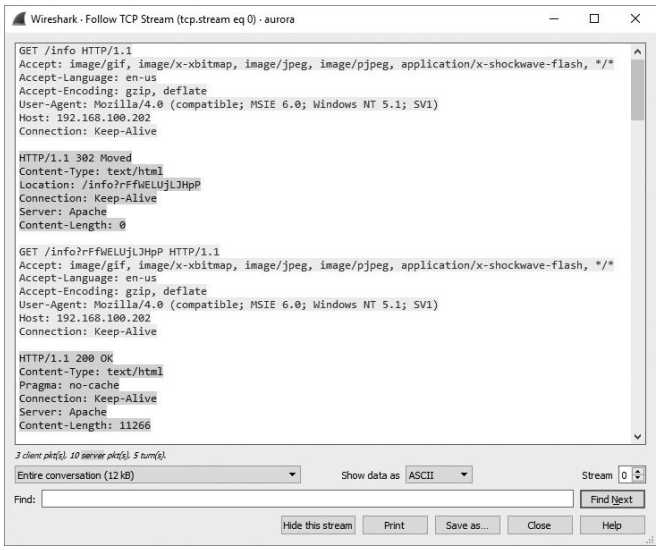

图 12-18　发送到客户端的数据流

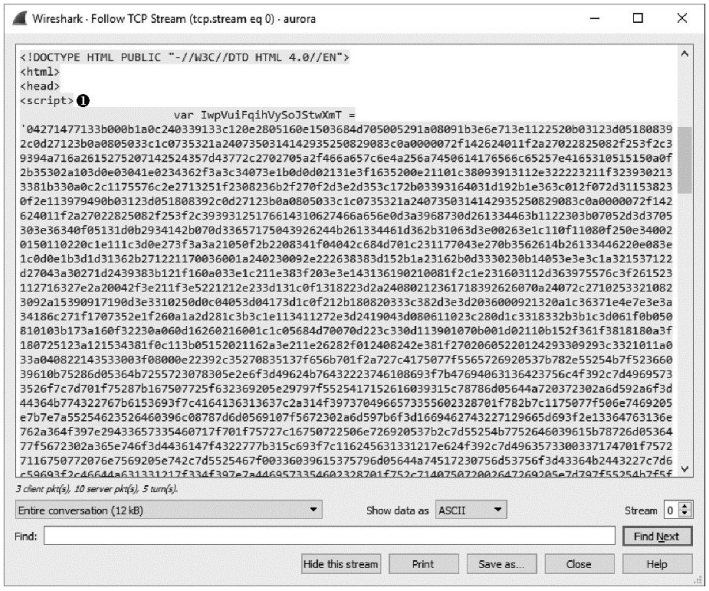

图 12-19　<script> 标签内的杂乱内容看起来像是被编码了

图 12-20 显示了攻击者发送的第二部分内容。在已编码文本之后，我们最终看到了一些可读文本。即便没有丰富的编程知识，我们也能看出这些文本像是在一些变量的基础上做一些字符串解析。这是闭合标签 </script> 之前的最后一些文本比特。

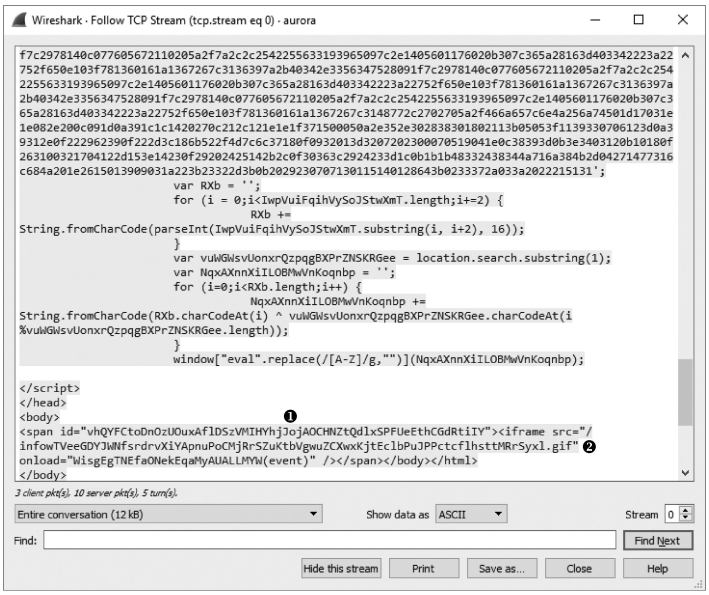

图 12-20　服务器发送的这部分内容包含了可读文本和可疑的 iframe

攻击者发给客户端的最后一段数据包含两个部分。第 1 部分是 ❶。第 2 部分包含在 标签内，是 <iframe src=「/infowTVeeGDYJWNfsrdrvXiYApnuPoCMjRrSZuKtbVgwuZCXwxKjtEclbPuJPPctcflhsttMRrSyxl.gif」 onload=「WisgEgTNEfaONekEqaMyAUALLMYW（event）」 />❷。这些

内容也有可能是恶意行为的标志，因为这些文本出奇的长、包含不可读的随机字符串和可能被混淆过的文本。

 标签内包含了一个 iframe，这是攻击者惯用的手法，用于在 HTML 页面中嵌入额外内容。<iframe> 标签创建了一个内联帧，并不被用户察觉。在这个案例中，<iframe> 标签引用了一个名字古怪的 GIF 文件。如图 12-21 所示，当受害者的浏览器发现对这个文件的引用时，它在数据包 21 中发送了一个 GET 请求 ❶，紧接着 GIF 文件就被传送过来了 ❷。这个 GIF 文件有可能被用来以某种方式，触发已经下载到受害者机器上的漏洞利用代码。

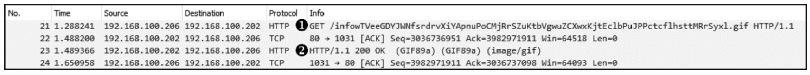

图 12-21　受害者请求并下载 iframe 中指明的 GIF 文件

当受害者向攻击者的 4321 端口初始化连接时，文件中最奇怪的一部分出现了，请看数据包 25。从 Packet Details 面板中查看第二个通信流并不能得出太多信息，因此我们再次查看 TCP 通信流以更清楚地了解传输的数据。图 12-22 显示了 Follow TCP Stream 窗口的输出。

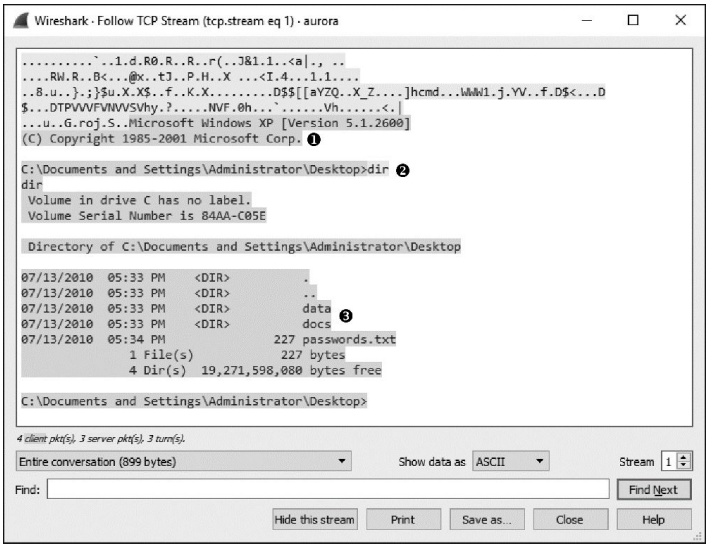

图 12-22　攻击者通过这个连接与命令窗口交互

在这里，我们看到了应该立即引发警报的东西：一个 Windows 的命令行解释器 ❶。这个 shell 是由受害者发给服务器的，这表明攻击者成功利用了漏洞：一旦漏洞利用程序启动，客户端就给攻击者发送回一个命令行解释器。在这个捕获中，我们甚至能看到攻击者与受害者的交互：输入 dir 命令 ❷ 以列出受害者机器的目录内容 ❸。

攻击者进入这个命令行解释器后，对受害者机器就有了无限制的管理权限，他几乎能做任何想做的事情。受害者只不过是轻点了一下鼠标，几秒钟之内就把计算机的完全控制权限交给了攻击者。

像这样的漏洞利用程序在线路上传输信息时通常都编码成不可识别的字符，以避免被网络 IDS（入侵检测系统）发现。就其本身而言，没有对这个漏洞利用程序的事先了解，也没有该程序的代码样本，在没有进一步分析之前很难说清受害者系统上到底发生了什么。幸好，我们可以从数据包捕获中识别出恶意代码的一些蛛丝马迹。这包括 <script> 标签里的一些混淆文本、奇怪的 iframe，以及明文表示的命令行解释器。

我们在这里总结一下极光漏洞利用程序是如何工作的。

- 受害者收到一封来自攻击者的邮件，看起来合理，实际上有针对性，单击里面的一个链接，向攻击者的恶意网站发送一个 GET 请求。
- 攻击者的 Web 服务器向受害者发出 302 重定向，受害者的浏览器向重定向 URL 自动发起一个 GET 请求。
- 攻击者的 Web 服务器向客户端发送一个含有混淆的 JavaScript 代码（包括一个漏洞利用程序）的 Web 页面，以及一个含有恶意 GIF 图像链接的 iframe。
- 受害者向恶意图像发起一个 GET 请求，将它从服务器上下载下来。
- 利用 IE 浏览器的漏洞，使用恶意 GIF 文件解混淆之前发送的 JavaScript 代码，并在受害者机器上执行。
- 一旦漏洞被成功利用，就执行隐藏在混淆代码中的 payload，从受害者向攻击者的 4321 端口打开一个新会话。
- 该 payload 会产生一个命令行解释器并返回给攻击者，以便攻击者与目标进行交互。

从防御的观点看，我们可以使用这个捕获文件为 IDS 创建一个特征，也许能有助于检测这种攻击。举个例子，我们可以过滤捕获文件的非混淆部分，比如 <script> 标签里混淆文本末尾的明文代码。另外一个思路是为所有包含 302 重定向到特定 URL 的 HTTP 流量写一个特征。为能在生产环境中使用这个特征，还需要一些额外的调整，但这已经是个很好的开端。

> 注意
>
> 对尝试防御网络未知威胁的人而言，基于恶意流量样本创建流量特征是非常关键的一步。这里描述的捕获是提升编写特征技能的好方法。