12.3.2 远程访问特洛伊木马

到目前为止，我们已经利用先验知识查看了一些安全事件。这是学习攻击形态的好办法，但却不太符合实际。在真实世界里，守护网络安全的人不可能查看网络中的每一个数据包。他们会使用各式各样的 IDS 设备，基于预定义的攻击特征，来提醒他们留意网络流量里的异常情况，并实施进一步检查。

在下一个案例中，我们将像分析真正的网络威胁一样，从一个简单的警报开始。在这个例子中，我们的 IDS 生成了这个警报：

```
   [**] [1:132456789:2] CyberEYE RAT Session Establishment [**]
[Classification:A Network Trojan was detected] [Priority:1]
07/18-12:45:04.656854 172.16.0.111:4433 -> 172.16.0.114:6641
TCP TTL:128 TOS:0x0 ID:6526 IpLen:20 DgmLen:54 DF
***AP*** Seq:0x53BAEB5E  Ack:0x18874922  Win:0xFAF0 TcpLen:20
```

下一步我们将查看触发此次警报的特征规则：

```
   alert tcp any any -> $HOME_NET any (msg:"CyberEYE RAT Session Es
content:"|41 4E 41 42 49 4C 47 49 7C|"; classtype:trojan-activity;
sid:132456789; rev:2;)
```

这个规则是这样定义的：当它发现一个进入内网的数据包含有十六进制内容 41 4E 41 42 49 4C 47 49 7C 时，就产生警报。这个内容转换成可读 ASCII 码是 ANA BILGI。当检测到这个字符串响起警报时，可能预示着 CyberEYE 远程访问木马（Remote-access Trojan，RAT）的出现。RAT 是秘密运行在受害者计算机上的恶意应用程序，它向攻击者建立连接，使攻击者能远程管理受害者的机器。

> 注意
>
> CyberEYE 是来自土耳其的工具，用于产生 RAT 程序和管理「肉鸡」。有趣的是，这里看到的 Snort 规则有一个敏感字符串「ANA BILGI」，它其实是土耳其文字，表示「基本信息」的意思。

现在我们将在 ratinfected.pcap 文件中查看与警报有关的流量。Snort 通常只捕获触发警报的单个数据包，但幸好我们有主机间的完整通信序列。让我们搜索 Snort 规则中提到的十六进制字符串，直接跳到关键部分。

（1）选择**Edit->Find Packet**。

（2）选择**Hex Value**单选按钮。

（3） 在文本框输入**41 4E 41 42 49 4C 47 49 7C**。

（4） 单击**Find**。

如图 12-23 所示，你最先在数据包 4 的数据部分发现了以上字符串 ❶。

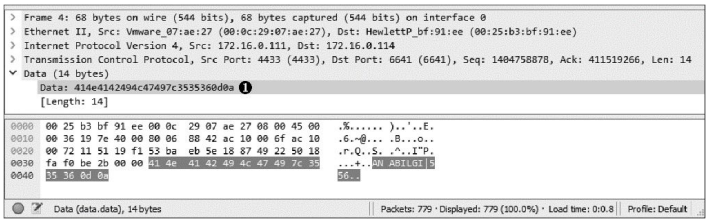

图 12-23　最先在数据包 4 中发现了 Snort 警报中的字符串内容

若多次选择**Edit->Find Next**，你会看见这个字符串也在数据包 5、10、32、156、280、405、531 和 652 出现了。虽然这个捕获文件里的所有通信都是在攻击者（172.16.0.111）和受害者（172.16.0.114）之间产生的，但看起来这个字符串出现在了不同的会话中。数据包 4 和 5 使用 4433 端口和 6641 端口通信，而其他大部分实例出现在 4433 端口和其他随机选择的临时端口之间。通过查看 Conversations 窗口的 TCP 标签，我们可以确认存在多个会话，如图 12-24 所示。

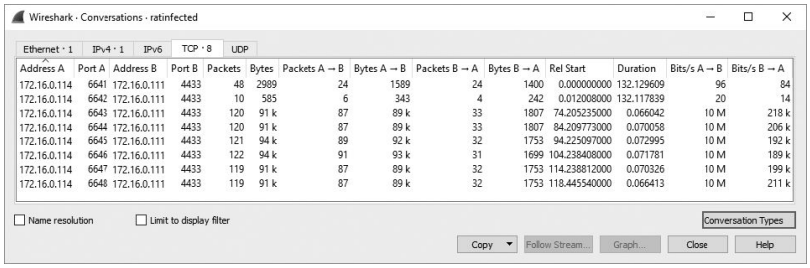

图 12-24　攻击者和受害者之间存在 3 个独立的会话

我们可以给捕获文件里的不同会话刷上不同颜色，将他们从视觉上分开。

（1） 在 Packet List 面板上面的过滤器栏里，输入过滤器 `(tcp.flags.syn == 1) && (tcp.flags.ack == 0)`。然后单击**Apply**。这将筛选出流量中每一个会话的初始 SYN 数据包。

（2） 右击第一个数据包，选择**Colorize Conversation**。

（3） 选择**TCP**，然后选择一种颜色。

（4） 为剩下的 SYN 数据包重复这个过程，分别选择不同的颜色。

（5） 完成之后，选择**Clear**移除过滤器。

为每个会话着色后，我们可以看一看他们之间有什么关联，这将帮助我们更好地跟踪两台主机之间的通信过程。第一个会话（ports 6641/4433）是两端通信的开端，从这开始不会错了。右击会话里的任一个数据包，选择**Follow TCP Stream**可以看到被传输的数据，如图 12-25 所示。

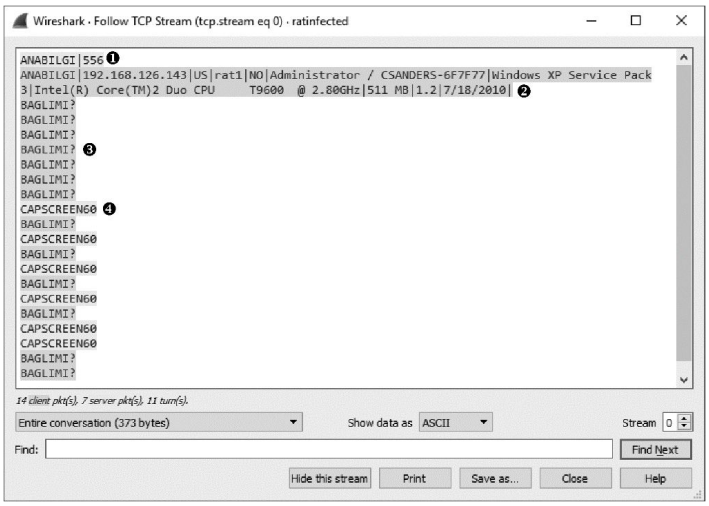

图 12-25　第一个会话产生了有趣的结果

很快，我们看到攻击者给受害者发送文本字符串「ANABILGI|556」❶。结果，受害者响应了一些基本系统信息，包括计算机名称（CSANDERS-6F7F77）和使用的操作系统（Windows XP Service Pack 3）❷，并开始给攻击者发送回一些重复的字符串「BAGLIMI?」❸。攻击者返回的消息只有字符串「CAPSCREEN60」❹，出现了 6 次。

攻击者返回的字符串「CAPSCREEN60」很有意思，看一看它要带我们去哪里。我们再次使用 search 对话框在数据包中搜索这个文本字符串，指明**String**选项。

我们搜索到数据包 27 首次出现了这个字符串。这个信息的有趣之处在于，客户端一收到攻击者发送的这个字符串，就确认接收这个数据包，并在数据包 29 发起了一个新会话。

现在，如果我们跟随这个新会话（见图 12-26）的 TCP 流输出，就能看到熟悉的字符串「ANABILGI|12」，跟在后面的是字符串「SH|556」，最后是字符串「CAPSCREEN|C:\WINDOWS\jpgevhook.dat|84972」❶。注意到字符串「CAPSCREEN」之后指明的文件路径后面跟着不可读的文本。这里最吸引人的是不可读文本有一个前导字符串「JFIF」❷，Google 搜索一下知道，JPG 文件的开头部分通常就包含它。

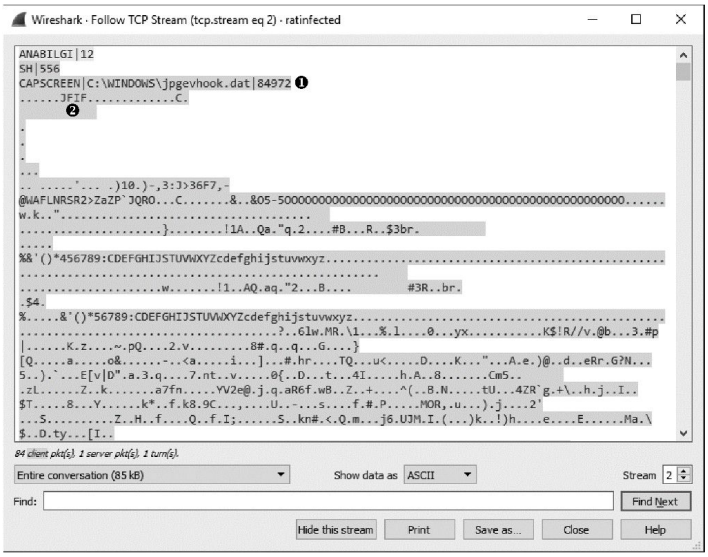

图 12-26　似乎攻击者发起了对一个 JPG 文件的请求

到这里，我们可以负责任地说，攻击者发起新会话是为了传输这个 JPG 图像。但更重要的是，我们开始从流量中推断出一个命令结构。看起来，攻击者发送「CAPSCREEN」命令就可以发起 JPG 图像的传输。实际上，不管什么时候发送「CAPSCREEN」命令，结果都是一样的。为了验证这个结论，可以查看每个会话的流，或使用下面介绍的 Wireshark 的 IO 绘图功能。

（1）选择**Statistics->IO Graphs**。

（2）在 5 个过滤器栏中分别插入`tcp.stream eq 2`、`tcp.stream eq 3`、`tcp.stream eq 4`、`tcp.stream eq 5`和`tcp.stream eq 6`。

（3）单击**Graph 1**、**Graph 2**、**Graph 3**、**Graph 4**和**Graph 5**按钮分别启用各个过滤器的数据点。

（4）将 y 轴的单位改成**Bytes/Tick**。

图 12-27 显示了结果图像。

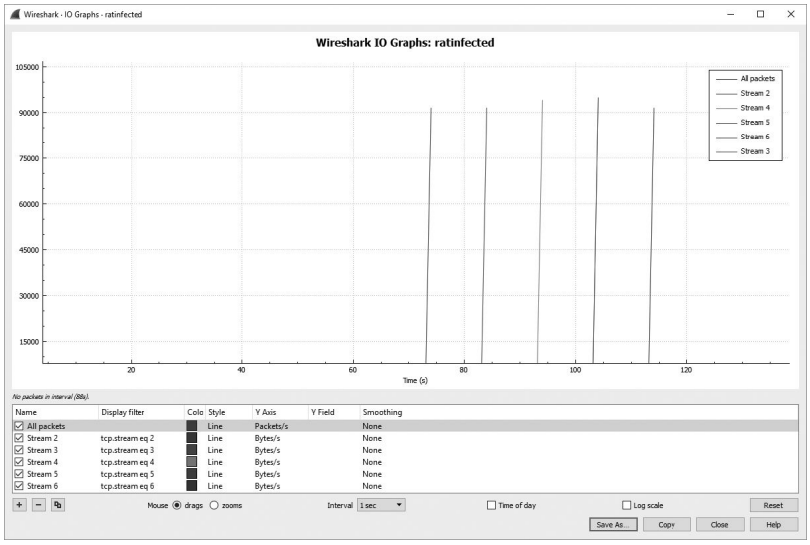

图 12-27　图像显示同样的活动重复了多次

从图 12-27 来看，似乎每个会话都包含同样大小的数据，并出现了同样多的时间。现在我们可以总结，这个活动重复了好几次。

对于传输的 JPG 图片内容，你可能已经有了些想法，所以让我们看一看是否能查看这些 JPG 文件。执行以下步骤可以从 Wireshark 中提取 JPG 数据。

（1）首先，对特定数据包的 TCP 流进行重组，如之前图 12-22 所示的文本。

（2）然后通信被分离出来，我们只能看到受害者发送给攻击者的流数据。选择下拉菜单旁边的箭头，那里写着`Entire Conversation (85033 bytes)`。确保选择合适的流量方向，就是`172.16.0.114:6643 --> 172.16.0.111:4433 (85020 bytes)`。

（3）选择**Save As**按钮保存数据，确保扩展名是.jpg。

你现在试一下，一定会发现它无法打开。因为我们还需要再做一步。不像在第 8 章从 FTP 流量中提取完整文件那样，这里的流量在真实数据之外还增加了一些额外内容。在这个例子中，TCP 流的前两行实际上是木马命令序列的一部分，而不属于 JPG 数据（见图 12-28）。当我们保存数据流时，这些额外的数据也被保存了下来。结果，文件浏览器查找 JPG 文件头时遇到了预料之外的信息，导致它不能打开图片。

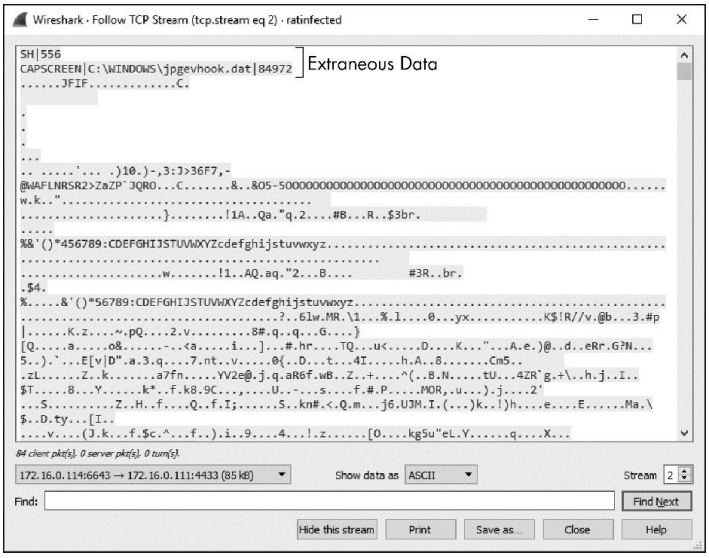

图 12-28　特洛伊木马增加的额外数据导致文件无法正确打开

用十六进制编辑器修复这个问题很简单。这个过程叫「文件修复」（File carving）。在图 12-29 中，我已经用 WinHex 选定 JPG 文件前面的一些字节。你可以使用任何十六进制编辑器删除这些字节并保存图片文件。

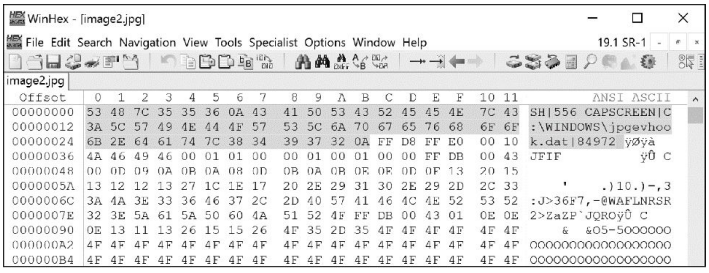

图 12-29　移除 JPG 文件中的附加字节

移除不需要的数据后，文件应该能够打开了。很显然，木马将受害者的桌面进行截屏，并发送回给攻击者（见图 12-30）。

图 12-30　发送的 JPG 文件是受害者计算机的屏幕截图

这些通信序列完成之后，通信就随着 TCP 拆除包而结束了。

这个场景展示的思考过程，就是一位入侵分析师分析 IDS 警报流量时，应遵循的典型步骤。

（1） 查看警报及触发它的特征。

（2） 在恰当的上下文确定特征确实在流量中。

（3） 查看流量，找出攻击者对「肉鸡」动了什么手脚。

（4） 在「肉鸡」泄露更多敏感信息之前，控制局面。