## 12.1.2 操作系统指纹

了解目标的操作系统对攻击者有极大价值。了解到目标使用的操作系统，将确保攻击者实施具有针对性的攻击手段。同时这个信息也有助于攻击者成功进入系统后，在目标文件系统找到关键文件和目录。

「操作系统指纹术」是指在没有物理接触的情况下，用来确定机器运行的操作系统的一组技术。操作系统指纹术分为两种类型：被动式和主动式。

### 1．被动式指纹技术

被动式指纹技术通过分析目标发送的数据包的某些字段来确定目标使用的操作系统。这项技术之所以称为「被动」，是因为你只监听目标主机发送的数据包，但并不主动向目标发送任何流量。对黑客来说，这是理想的操作系统指纹技术，因为它非常隐蔽。

也就是说，我们只需要基于目标主机发送的数据包，就能确定它用的是哪一种操作系统了？嗯，这其实非常容易。由于 RFC 文件定义的协议并未规定全部技术参数的值，这完全是有可能的。虽然 TCP、UDP 和 IP 头部的各个字段都有特定含义，但并没有定义这些字段的默认值。这意味着不同操作系统实现的 TCP/IP 协议栈都必须为这些字段定义它自己的默认值。表 12-1 列出了一些与操作系统实现有关的常见字段及其默认值。

**表 12-1　常见的被动式指纹值**

| 协议头部 | 字段 | 默认值 | 操作系统 |
| --- | --- | --- | --- |
| IP | 初始 TTL | 64 | Nmap、BSD、Mac OS 10、Linux |
| | | 128 | Novell、Windows |
| | | 255 | Cisco IOS、Palm OS、Solaris |

| 协议头部 | 字段 | 默认值 | 操作系统 |
| --- | --- | --- | --- |
| IP | 不分片标志 | Set | BSD、Mac OS 10、Linux、Novell、Windows、Palm OS、Solaris |
| | | Not set | Nmap、Cisco IOS |
| TCP | 最长段大小 | 0 | Nmap |
| | | 1440 | Windows、Novell |
| | | 1460 | BSD、Mac OS 10、Linux、Solaris |
| TCP | 窗口大小 | 1024-4096 | Nmap |
| | | 65535 | BSD、Mac OS 10 |
| | | 2920-5840 | Linux |
| | | 16384 | Novell |
| | | 4128 | Cisco IOS |
| | | 24820 | Solaris |
| | | 可变 | Windows |

| 协议头部 | 字段 | 默认值 | 操作系统 |
| --- | --- | --- | --- |
| TCP | SackOK | 设置 | Linux、Windows、OpenBSD |
| | | 不设置 | Nmap、FreeBSD、Mac OS 10、Novell、Cisco IOS、Solaris |

捕获文件 passiveosfingerprinting.pcap 中的数据包是这项技术的绝佳例子。该文件含有两个数据包。它们都是目标端口为 80 的 TCP SYN 数据包，但来自不同的主机。仅仅使用这些数据包中的值，并参考表 12-1，我们就能确定每台主机使用的操作系统架构。每一个数据包的细节如图 12-7 所示。

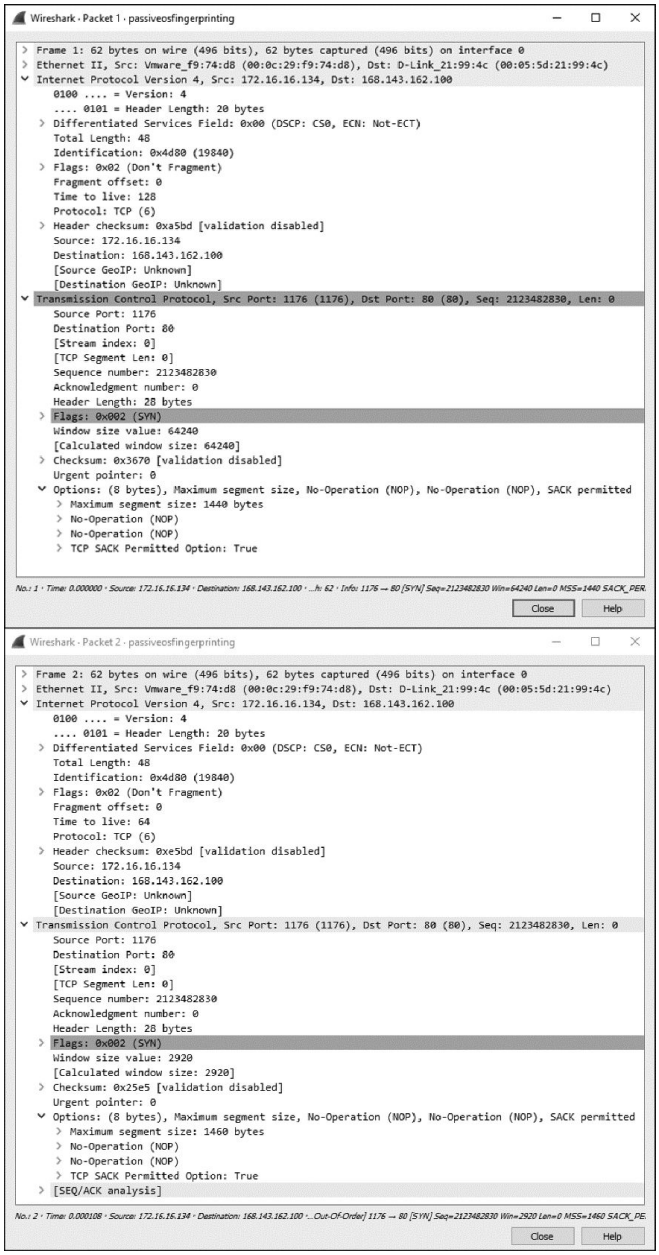


图 12-7　这些数据包可以告诉我们它们来自哪种操作系统

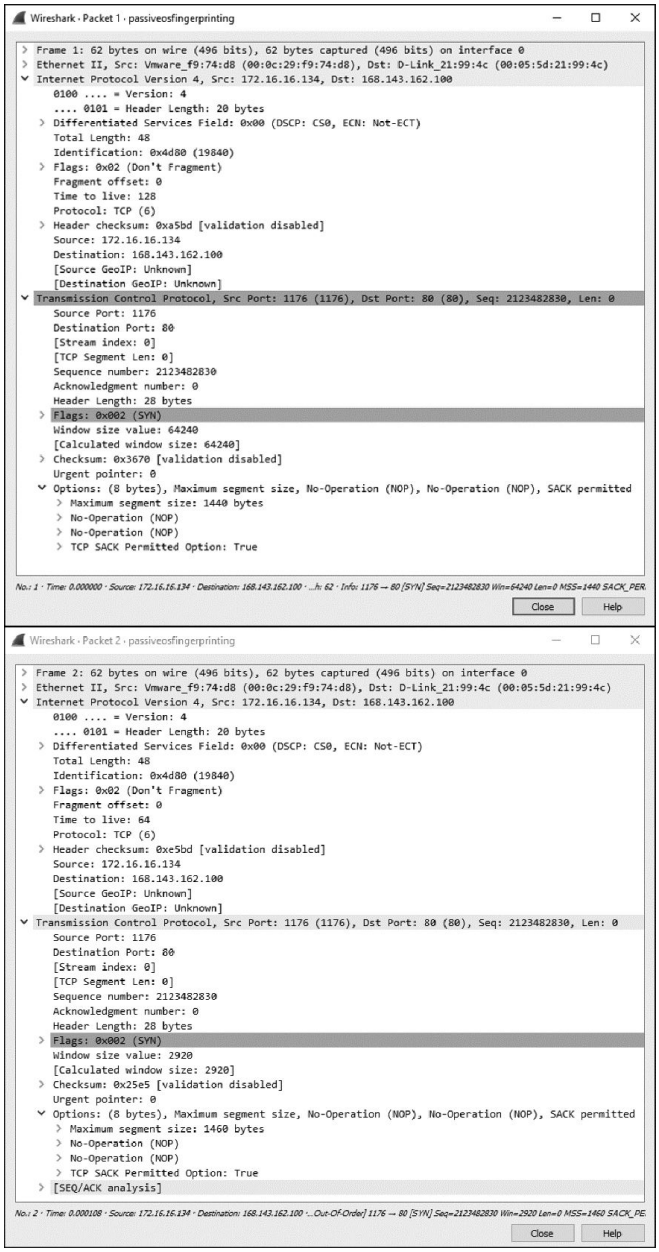


图 12-7　这些数据包可以告诉我们它们来自哪种操作系统（续）

参考表 12-1，我们将这些数据包的相关字段分类，创建了表 12-2。

**表 12-2　操作系统指纹术的数据包分类**

| 协议头部 | 字段 | 数据包 1 的值 | 数据包 2 的值 |
|---|---|---|---|
| IP | 初始 TTL | 128 | 64 |
| IP | 不分片标志 | 设置/不设置 | 设置/不设置 |

| 协议头部 | 字段 | 数据包 1 的值 | 数据包 2 的值 |
| --- | --- | --- | --- |
| TCP | 最长段大小 | 1440 Bytes | 1460 Bytes |
| TCP | 窗口大小 | 64240 Bytes | 2920 Bytes |
| TCP | SackOK | 设置/不设置 | 设置/不设置 |

基于这些值，我们可以得出结论：发送数据包 1 的设备运行 Windows 的可能性最大，而发送数据包 2 的设备运行 Linux 的可能性最大。

要记住，表 12-2 列出的被动式操作系统指纹技术的常见识别字段并不完整。很多实现上的「怪癖」可能会导致真实值与期望值的偏差。所以，你不能完全依赖被动式操作系统指纹技术得到的结果。

> 注意
>
> 有一个叫 p0f 的工具使用了操作系统指纹识别技术。该工具分析捕获数据包的相关字段，然后输出可能的操作系统。使用像 p0f 这样的工具，你不仅能了解到操作系统架构，有时甚至能了解适当的版本号或者补丁级别。

### 2．主动式指纹技术

当被动监听流量不能得出想要的结果时，可能需要一个更直接的方法。这种方法叫做「主动式指纹技术」。它是指攻击者主动向受害者发送特意构造的数据包以引起响应，然后从响应数据包中获知受害者机器操作系统的技术。当然，由于这种方法要与受害者直接通信，因此它并不是最隐蔽的，但它可以做到非常高效。

捕获文件 activeosfingerprinting.pcap 包含了使用 Nmap 扫描工具发起主动式指纹扫描的例子。文件中有一些是 Nmap 发送的探测数据包，这些探测数据包引起的响应可用于识别操作系统。Nmap 记录下对这些探测数据包的响应并创建一个指纹，与指纹数据库对比后得出结论。

> 注意
>
> Nmap 使用的主动式操作系统指纹技术十分复杂。想了解 Nmap 如何执行主动式操作系统指纹技术，请阅读 Nmap 的权威指南——*Nmap Network Scanning*。这是 Nmap 扫描器作者 Gordon「Fyodor」 Lyon 的著作。