

12.1.1 SYN 扫描

首先对系统作 TCP SYN 扫描，又称为隐秘扫描或半开扫描。SYN 扫描是一种常见的扫描类型，有以下几个原因。

- 快速可靠。
- 在所有平台上都很准确，与 TCP 协议栈的实现无关。
- 比其他扫描技术更安静，不容易被发现。

TCP SYN 扫描依赖于三步握手过程，可以确定目标主机的哪些端口是开/的。攻击者发送 TCP SYN 数据包到受害者的一定范围的端口上，就像要在这些端口上建立用于正常通信的连接似的。如图 12-1 所示，一旦受害者收到这个数据包，就可能会做出某些响应。

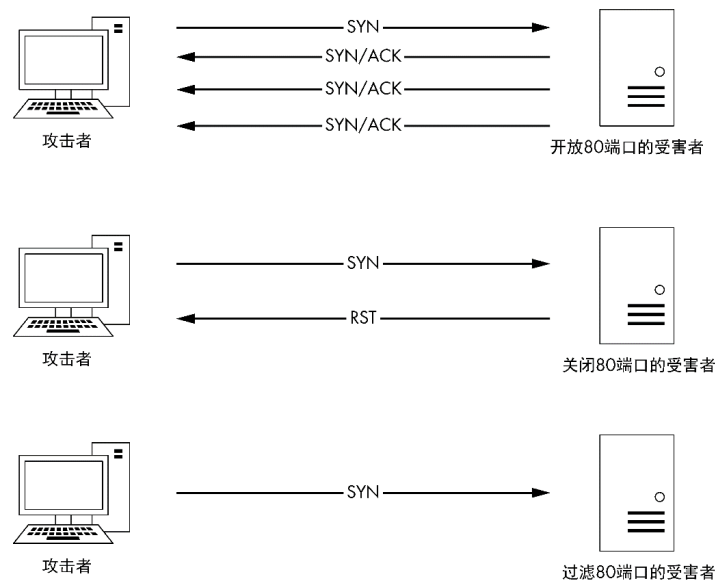


图 12-1 一次 TCP SYN 扫描的结果

如果受害者机器上某个服务正在监听的端口收到了 SYN 数据包，那么它将向攻击者回复一个 TCP SYN/ACK 数据包，也就是 TCP 握手的第二部分。这样攻击者就能知道这个端口是开放的，并且有一个服务在上面监听。正常情况下会发送一个 TCP ACK 包以完成连接握手，但此刻攻击者并不想这样，因为他还不想与主机通信。所以，攻击者并不打算完成 TCP 握手。

如果没有服务在被扫描的端口上监听，那么攻击者就收不到 SYN/ACK。按照受害者操作系统的不同配置，攻击者可能会收到响应的 RST 数据包，表示端口关闭了，或者，攻击者看不到任何响应。这意味着端口被某个中间设备过滤了，或许是防火墙，或许是主机本身。另一方面，也有可能是因

为响应数据包在传输过程中丢失了。这个结果通常表明端口是关闭的，但说服力并不强。

捕获文件 `synscan.pcap` 提供了用 Nmap 工具进行 SYN 扫描的绝佳例子。Nmap 是 Fyodor 创立的一款稳定的网络扫描程序。它可以执行你能想到的任何一种扫描方式。

我们捕获的样本大概包含 2000 个数据包，说明这种扫描有一定的规模。确定这个扫描范围大小的最好办法之一就是查看 Conversations 窗口，如图 12-2 所示。在这里，你会看到攻击者（172.16.0.8）和受害者（63.13.134.52）之间只有一个 IPv4 会话 ❶。你也会看到，那里有 1994 个 TCP 会话 ❷——通信基本上是每一个端口对应一个新会话。

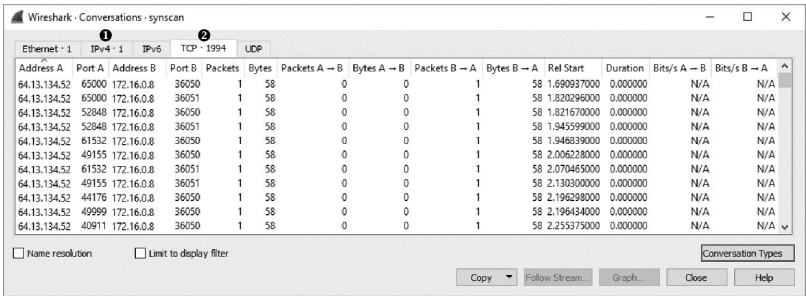


图 12-2 Conversations 窗口显示了正在进行的 TCP 通信

扫描是在极短时间内完成的，因此在捕获文件上滚动鼠标并不是寻找 SYN 数据包响应的好办法。在接收到响应之前，已经发送了更多的 SYN 数据包。幸好，我们可以创建过滤器，来帮助我们寻找正确的流量。

1. 在 SYN 扫描中使用过滤器

举一个筛选的例子。让我们看一看第 1 个数据包——发送到受害者 443 端口（HTTPS）的 SYN 数据包。为了查看是否有对这个数据包的响应，我们可以创建一个过滤器，以显示所有源端口或目标端口为 443 的流量。下面是如何快速设置的方法。

- (1) 在捕获文件中选择第一个数据包。
- (2) 在 Packet Details 面板中展开 TCP 头部。
- (3) 右键单击 Destination Port 字段，选择 Prepare as Filter，单击 Selected。
- (4) 这将在 filter 对话框放置一个过滤器，针对所有目标端口为 443 的数据包。现在，由于我们也需要源端口为 443 的数据包，所以点击屏幕顶端的 filter 栏，并删除过滤器的 dst 部分。

结果过滤器给出了两个数据包，都是攻击者发给受害者的 TCP SYN 数据包，如图 12-3 所示。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.0.8	64.13.134.52	TCP	36050 → 443 [SYN] Seq=3713172248 Win=3672 Len=0 MSS=1460
32	0.000005	172.16.0.8	64.13.134.52	TCP	36051 → 443 [SYN] Seq=3713237785 Win=2048 Len=0 MSS=1460

图 12-3 两次尝试用 SYN 数据包建立连接

两个数据包都没有得到响应，有可能是因为响应数据包被受害者主机或中间设备过滤了，或者端口是关闭的。但最终来说，对 443 端口的扫描结果是不确定的。

我们可以用同样的技术来分析其他数据包，看一看有没有不同的结果。首先，单击过滤器旁边的 Clear 按钮，清空之前创建的过滤器。然后选择列表中的第 9 个数据包。这是目标端口为 53 的 SYN 数据包，通常与 DNS 有关。使用前面提到的方法，创建一个基于目标端口的过滤器，并删除 dst 部分，这样它就应用到所有与 TCP 53 端口有关的流量了。当使用这个过滤器时，你会看见 5 个数据包，如图 12-4 所示。

No.	Time	Source	Destination	Protocol	Info
9	0.000052	172.16.0.8	64.13.134.52	TCP	36050 → 53 [SYN] Seq=3713172248 Win=3672 Len=0 MSS=1460
11	0.001852	64.13.134.52	172.16.0.8	TCP	53 → 36050 [SYN, ACK] Seq=1117405124 Ack=3713172249 Win=5840 Len=0 MSS=1380
529	0.057126	64.13.134.52	172.16.0.8	TCP	[TCP Retransmission] 53 → 36050 [SYN, ACK] Seq=1117405124 Ack=3713172249 Win=5840 Len=0 MSS=1380
2006	5.930109	64.13.134.52	172.16.0.8	TCP	[TCP Retransmission] 53 → 36050 [SYN, ACK] Seq=1117405124 Ack=3713172249 Win=5840 Len=0 MSS=1380
2009	10.029023	64.13.134.52	172.16.0.8	TCP	[TCP Retransmission] 53 → 36050 [SYN, ACK] Seq=1117405124 Ack=3713172249 Win=5840 Len=0 MSS=1380

图 12-4 表明端口是开放的 5 个数据包

第 1 个是我们在捕获之初选择的 SYN 数据包。第 2 个则是来自受害者的响应。这是一个 TCP SYN/ACK 数据包——实施三次握手时期望的响应。在正常情况下，下一个数据包应该是发送初始 SYN 的主机发送的 ACK。然而，在这个例子中，攻击者并不想建立连接，因而没有发送响应。受害者重传了 3 次 SYN/ACK 包才放弃。由于尝试与主机的 53 端口通信时收到了 SYN/ACK 响应，因此我们可以确定有一个服务在监听该端口。

让我们在数据包 13 上再次重复此过程。这是一个目标端口为 113 的 SYN 数据包，通常与 Ident 协议有关，此协议常用于 IRC 的身份识别和验证服务。如果你在这个数据包上使用同一类型的过滤器，就会发现 4 个数据包，如图 12-5 所示。

No.	Time	Source	Destination	Protocol	Info
13	0.000070	172.16.0.8	64.13.134.52	TCP	36050 → 113 [SYN] Seq=3713172248 Win=4096 Len=0 MSS=1460
14	0.001491	64.13.134.52	172.16.0.8	TCP	113 → 36050 [RST, ACK] Seq=2462244745 Ack=3713172249 Win=0 Len=0
530	0.006942	172.16.0.8	64.13.134.52	TCP	36061 → 113 [SYN] Seq=3696394776 Win=2048 Len=0 MSS=1460
571	0.008827	64.13.134.52	172.16.0.8	TCP	113 → 36061 [RST, ACK] Seq=1027049353 Ack=3696394777 Win=0 Len=0

图 12-5 SYN 之后紧随一个 RST，表明端口是关闭的

第 1 个数据包是初始 SYN，紧接着是来自受害者的 RST。这是受害者目标端口不接受连接的迹象，表明很可能没有服务运行在上面。

2. 识别开放和关闭的端口

理解了 SYN 扫描能引起的不同响应类型后，自然而然会想到去找一个方法——如何快速识别哪些端口是开放的还是关闭的。答案再次落到了 Conversations 窗口内。在这个窗口中，你可以通过数据包数量排序 TCP 会话，单击 Packets 列直到箭头向下就可以让最高值靠前，如图 12-6 所示。

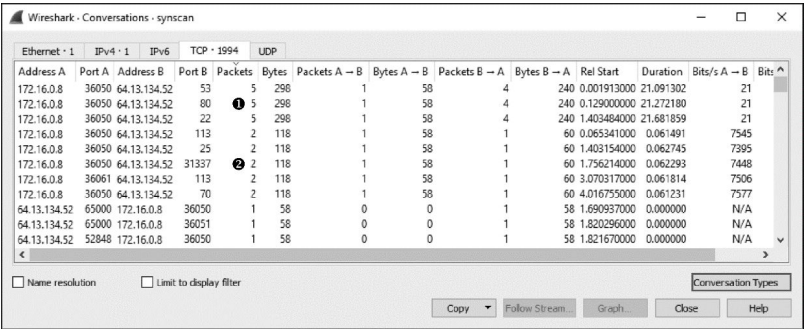


图 12-6 用 Conversations 窗口寻找开放端口

3 个被扫描的端口在各自会话中包含 5 个数据包 ❶。我们知道 53、80 和 22 端口是开放的，因为这 5 个数据包表示初始 SYN、来自受害者的 SYN/ACK 及其 3 次重传。

有 5 个端口的通信只包含 2 个数据包 ❷。第 1 个是初始 SYN，第 2 个是来自受害者的 RST。这表明 113、25、31337、113 和 70 端口是关闭的。

Conversations 窗口剩下的项只包含 1 个数据包，意味着受害者主机并没有响应初始 SYN 包。剩下的这些端口很可能是关闭的，但我们不能确定。