CHAPTER 03

第 3 章 端口扫描技术

本章介绍如下内容。

- ❑ 端口的概念。
- ❑ 端口的分类。
- ❑ Nmap 中对端口状态的定义。
- ❑ Nmap 中的各种端口扫描技术。
- ❑ Nmap 中如何指定扫描端口的范围。
- ❑ 端口扫描技术的实现原理。
- ❑ 端口扫描的算法。

3.1 端口的概念

如果把每一台网络设备看作一间房子，那么这间房子应该有能够进去和出来的出入口，不过一般的房子只有一个出入口，这些出入口是供人们进出房子使用的。但是每个网络设备却有很多个出入口，最多可以达到 65 536（216）个，而这些出入口是供数据进出网络设备的。

设立端口的目的其实就是实现了“一机多用”。假设没有端口技术，一台主机通常只能运行一种网络服务，总是只有一个程序进行网络通信，那么只会有一个端口，甚至没有端口这个概念了。正因为有很多并且将有更多的程序要通过网络进行通信，而所有信息实际上都

要通过网卡这个接口出入，那么如何区分出入的信息是给哪个程序使用的呢？这个任务交由操作系统处理，而它所采用的机制就是划分出 65 536 个端口，程序在发送的信息中加入端口编号，而操作系统在接收到信息后会按照端口号将信息分流到当前内存中使用该端口号的程序。

3.2 端口的分类

根据端口使用情况的不同，可以简单地将端口分为如下几类：

- 公认端口（well known port）：这一类端口是最为常用的端口，因此也被称为“常用端口”。所使用的从 0 到 1024 的端口都是公认端口。通常这些端口已经明确地和某种服务的协议进行了关联，一般不应该对其进行改变。例如所熟知的 80 端口运行的总是 http 通信，而 telnet 也总是适用 23 号端口。这些端口的作用一般已经约定好，因此不会被其他程序所使用。
- 注册端口（registered port）：这部分端口号的范围是从 1025 到 49 151。它们通常也会关联到一些服务上，但是并没有明确的规定，不同的程序可以根据实际情况进行定义。
- 动态和 / 或私有端口（dynamic and/or private port）：这部分端口号的范围是从 49 152 到 65 535。一般来说，常见的服务不应该使用这部分端口，但是由于这部分端口不容易引起注意，因此有些程序尤其是一些木马或者病毒程序十分钟爱这部分端口。

另外，根据使用协议的不同，又可以将这些端口分成“ TCP 协议端口”和“ UDP 协议端口”两种不同的类型。第 2 章中我们已经提到过，传输层的功能是为通信双方的主机提供端到端的服务，这一层的协议包括 TCP 协议和 UDP 协议。针对使用以上这两种通信协议的服务所提供的端口，也可以分为“TCP 协议端口”和“UDP 协议端口”。

表 3-1 给出了一些常见的端口及其对应的服务。

表 3-1 常见的端口及其对应的服务与作用

端 口 号	名 称	作 用
1	tcpmux	TCP 端口服务多路复用
5	rje	远程作业入口
7	echo	echo 服务
9	discard	用于连接测试的空服务
11	systat	用于列举连接了的端口的系统状态
13	daytime	给请求主机发送日期和时间
17	qotd	给连接了的主机发送每日格言
18	msp	消息发送协议

（续）

端　口　号	名　　称	作　　用
19	chargen	字符生成服务；发送无止境的字符流
20	ftp-data	FTP 数据端口
21	ftp	文件传输协议（FTP）端口，有时被文件服务协议（FSP）使用
22	ssh	安全 Shell（SSH）服务
23	telnet	telnet 服务
25	smtp	简单邮件传输协议（SMTP）
⋮	⋮	⋮

3.3　Nmap 中对端口状态的定义

本章的主要目的是判断目标端口的状态。Nmap 中对于端口给出了 6 种不同的状态描述：

- ❑ open：如果目标端口的状态为 open，这表明在该端口有应用程序接收 TCP 连接或者 UDP 报文。
- ❑ closed：如果目标端口的状态为 closed，这里要注意 closed 并不意味着没有任何反应，状态为 closed 的端口是可访问的，这种端口可以接收 Nmap 探测报文并做出响应。相比较而言，没有应用程序在 open 上监听。
- ❑ filtered：产生这种结果的原因主要是存在目标网络数据包过滤，由于这些设备过滤了探测数据包，导致 Nmap 无法确定该端口是否开放。这种设备可能是路由器、防火墙甚至专门的安全软件。
- ❑ unfiltered：这种结果很少见，它表明目标端口是可以访问的，但是 Nmap 却无法判断它到底是 open 还是 closed 的。通常只有在进行 ACK 扫描时才会出现这种状态。
- ❑ open | filtered：无法确定端口是开放的还是被过滤了，开放的端口不响应就是一个例子。
- ❑ closed|filtered：无法确定端口是关闭的还是被过滤了。只有在使用 idle 扫描时才会发生这种情况。

3.4　Nmap 中的各种端口扫描技术

Nmap 中提供了大量的技术来实现对端口状态的检测，由于 TCP 技术相对 UDP 技术要复杂一些，所以 TCP 的检测手段也比 UDP 要多一些。第 2 章已经详细地介绍过 TCP 的工作方式，可以利用 TCP 的三次握手机制，产生多种扫描技术。例如最为典型的是 SYN 和 Connect 两种扫描方式。

3.4.1 SYN 扫描

SYN 扫描是最为流行的一种扫描方式，同时它也是 Nmap 所采用的默认扫描方式。这种扫描方式速度极快，可以在一秒钟扫描上千个端口，SYN 扫描也不容易被网络中的安全设备所发现。

你也可以在扫描的时候输入参数 -sS 选项。其实只要你是以 root 或者 administrator 权限工作的，扫描的形式都是 SYN 的。首先 Nmap 会向目标主机的一个端口发送请求连接的 SYN 数据包，而目标计算机在接收到这个 SYN 数据包之后会以 SYN/ACK 进行应答，Nmap 在收到 SYN/ACK 后会发送 RST 包请求断开连接而不是 ACK 应答。这样，三次握手就没有完成，无法建立正常的 TCP 连接，因此，这次扫描就不会被记录到系统日志中。这种扫描技术一般不会在目标主机上留下扫描痕迹。

在对一个端口进行 TCP SYN 扫描时，结果将会是表 3-2 中 open、closed 和 filtered 三者之一。

表 3-2　对目标进行 SYN 扫描时目标主机的应答与对应端口的状态

目标主机的应答	目标端口的状态
如果目标主机给出了一个 SYN/ACK 应答	open
如果目标主机给出了一个 RST 应答	closed
如果目标主机没有给出应答	filtered
ICMP 无法抵达错误（类型 3，代码 1，2，3，4，10，13）	filtered

使用 SYN 扫描端口语法如下。

```
nmap -sS [target]
```

例如对目标 192.168.153.131 扫描，命令如下。

```
nmap -sS 192.168.153.131
```

扫描的结果如下。

```
Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-21 14:43 ?D1ú±ê×?ê±??
Nmap scan report for bogon (192.168.153.131)
Host is up (0.000030s latency).
Not shown: 996 closed ports
PORT              STATE          SERVICE
21/tcp            open           ftp
25/tcp            open           smtp
139/tcp           open           netbios-ssn
445/tcp           open           microsoft-dsMAC
Address: 00:0C:29:90:DF:C3 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 3.08 seconds
```

3.4.2　Connect 扫描

使用 Connect 扫描端口的语法如下。

```
nmap -sT [target]
```

这种扫描方式其实和 SYN 扫描很像，只是这种扫描方式完成了 TCP 的三次握手。这种扫描方式无须 root 或者 administrator 权限。

例如对目标 192.168.153.131 扫描，命令如下。

```
nmap -sT 192.168.153.131
```

扫描的结果如下。

```
Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-21 14:51
Nmap scan report for bogon (192.168.153.131)
Host is up (1.00s latency).
Not shown: 996 closed ports
PORT              STATE          SERVICE
21/tcp            open           ftp
25/tcp            open           smtp
53/tcp            open           domain
445/tcp           open           microsoft-ds
MAC Address: 00:0C:29:90:DF:C3 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 225.37 seconds
```

3.4.3　UDP 扫描

在使用 UDP 扫描（使用 -sU 选项）对一个端口进行 UDP 扫描时，结果是表 3-3 所示的 open、closed 和 filtered 三者之一。

表 3-3　对目标进行 UDP 扫描时目标的应答与对应端口的状态

目标主机的应答	目标端口的状态
从目标端口得到任意的 UDP 应答	open
如果目标主机没有给出应答	open|filtered
ICMP 端口无法抵达错误（类型 3，代码 3）	closed
ICMP 无法抵达错误（类型 3，代码 1，2，9，10，13）	filtered

要注意 UDP 扫描的速度是相当慢的。使用 Connect 扫描端口的语法如下。

```
nmap -sU [target]
```

例如对目标 192.168.153.131 的端口进行 UDP 扫描，命令如下。

```
nmap -sU 192.168.153.131
```

扫描的结果如下。

```
Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-21 15:00
Nmap scan report for bogon (192.168.153.131)
Host is up (0.000080s latency).
```

```
Not shown: 987 closed ports
PORT              STATE                SERVICE
42/udp            open|filtered        nameserver
53/udp            open                 domain
68/udp            open|filtered        dhcpc
135/udp           open                 msrpc
137/udp           open                 netbios-ns
138/udp           open|filtered        netbios-dgm
445/udp           open|filtered        microsoft-ds
500/udp           open|filtered        isakmp
1028/udp          open|filtered        ms-lsa
1031/udp          open|filtered        iad2
1034/udp          open                 activesync-notify
1035/udp          open                 mxxrlogin
3456/udp          open|filtered        IISrpc-or-vat
MAC Address: 00:0C:29:90:DF:C3 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 3.74 seconds
```

在扫描过程中可能会产生一些状态为 filtered 的端口，这些端口的真实状态可能是 open，也有可能是 closed。要从这些状态为 filtered 的端口中找到那些其实是 open 的端口，需要进一步的测试。与 TCP 不同，UDP 程序的服务通常不会对 Nmap 所发送的空数据包做出回应，UDP 程序需要使用它们自己的格式，例如，一个 SNMP 请求数据包的格式就与 DHCP 或者 DNS 的请求包的格式完全不同。如果保证能向所有常见的 UDP 服务发送合适的数据包，Nmap 需要一个很大的数据库来存储这些格式，Nmap 将这些数据包的格式存储在 Nmap-service-probes 中。可以使用 -sV 或者 -A（这两个参数会在后面内容中提到），Nmap 将会向每个 open|filtered 的端口发送 UDP probe，如果目标端口对任何一个 probe 有了反应，状态都会被修改为 open。

3.4.4 TCP FIN 扫描

TCP FIN 扫描方法向目标端口发送一个 FIN 数据包。按照 RFC 793 的规定（http://www.ietf.org/rfc/rfc0793.txt），对于所有关闭的端口，目标系统应该返回 RST 标志。

使用 TCP FIN 扫描端口的语法如下。

```
nmap -sF [target]
```

例如对目标 192.168.153.131 进行扫描，命令如下。

```
nmap -sF 192.168.153.131
```

3.4.5 NULL 扫描

TCP NULL 扫描方法是向目标端口发送一个不包含任何标志的数据包。按照 RFC 793 的规定，对于所有关闭的端口，目标系统应该返回 RST 标志。

使用 TCP NULL 扫描端口的语法如下。

```
nmap -sN [target]
```

例如对目标 192.168.153.131 进行扫描，命令如下。

```
nmap  -sN 192.168.153.131
```

3.4.6　Xmas Tree 扫描

TCP Xmas Tree 扫描方法是向目标端口发送一个含有 FIN、URG 和 PUSH 标志的数据包。按照 RFC 793 的规定，对于所有关闭的端口，目标系统应该返回 RST 标志。

使用 TCP Xmas Tree 扫描端口的语法如下。

```
nmap -sX [target]
```

例如对目标 192.168.153.131 进行扫描，命令如下。

```
nmap -sX 192.168.153.131
```

3.4.7　idle 扫描

这种扫描方式在思路上十分巧妙，在整个扫描过程中，扫描者无须向目标主机发送任何数据包，怎么样，是不是光听着就觉得十分神奇？不过，不向目标主机发送数据包并不是指不发送数据包，在这种扫描方式中，需要一个“第三方”，它扮演了一个被利用的无辜角色，因为这个第三方对我们的扫描也是一无所知，所以我们的扫描是隐蔽的。

下面介绍这种扫描的原理。

步骤 1：检测第三方的 IP ID 值并记录下来。

步骤 2：在本机上伪造一个源地址为第三方主机的数据包，并将数据包发送给目标主机端口，根据目标端口状态的不同，目标主机可能会导致第三方主机的 IP ID 值增加。

步骤 3：再回来检查第三方主机的 IP ID 值。比较这两次的值。

这时，第三方主机的 IP ID 值应该是增加了 1 到 2，如果只是增加了 1，那么说明第三方主机在这期间并没有向外发送数据包，这种情况就认为目标主机的端口是关闭的。如果增加了 2，就表明第三方主机在这期间向外部发送了数据包，这样就说明目标主机的端口是开放的。

在这个过程中，因为需要伪造一个源地址为第三方的数据包，所以必须要求扫描工具具有伪造数据包的能力，Nmap 已经有了 decoy scanning（-D）的功能，可以帮助使用者隐藏自己的身份。要知道任何发往目标主机的数据包都有可能被目标主机的日志记录下来，而通过 idle 扫描方式，目标主机日志中记录下来的是第三方的地址。

idle 扫描的另一个优势在于可以绕开网络中的一些安全机制，如路由器中的访问控制列表技术（ACL），某些单位的内部网络限制只允许指定 IP 地址对其内部进行访问。有些时候，一些工作人员为了方便，经常会在防火墙或者路由器上设置例外，允许他们从家中对单位的网络进行访问。如果没有做好安全保障，这其实是一种相当危险的做法。

idle 扫描的缺点也同样明显，通常一个 SYN 扫描所需要花费的时间只是几秒，而 idle 扫描的扫描时间要远远多于这个时间，另一点就是很多时候，宽带提供商并不会允许你向外发送伪造的数据包。还有最为重要的一点是，idle 扫描要求你必须能找到一个正在工作的第三方主机。

这其实是一个并不简单的问题，首先这台第三方主机产生 IP ID 的方法必须是整体增加的，而不是根据每个通信自行开始的，最好是空闲的，因为大量的无关流量将会导致结果极为混乱。当然主机与第三方主机的延迟小也是理想的情况。

那么如何查看一台主机是否适合作为一台第三方主机呢？方法一是，在对一台主机进行扫描的时候，执行一个端口扫描以及操作系统检测，启动详细模式（-v），操作系统就会检测 IP ID 增长方法，如果返回值为“IP ID Sequence Generation: Incremental”，这是一个好消息。但还需要确定一下，因为很多系统其实为每一个通信开启了一个 IP ID，另外，如果这台主机和外界进行大量的通信，这种方法也是不适用的。

方法二是运行 ipidseq NSE 脚本。关于脚本的问题，后面会详细描述。

好了，当找到一台第三方主机后，一切就都简单了。例如使用 sI 参数指定 kiosk.adobe.com 作为第三方主机，然后对 www.riaa.com 进行扫描。扫描结果如下。

```
Nmap -Pn -p- -sI kiosk.adobe.com www.riaa.com
Nmap done: 1 IP address (1 host up) scanned in 2594.47 seconds
Starting Nmap ( http://Nmap.org )
Idlescan using zombie kiosk.adobe.com (192.150.13.111:80); Class: Incremental
Nmap scan report for 208.225.90.120
(The 65522 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http
111/tcp    open        sunrpc
135/tcp    open        loc-srv
443/tcp    open        https
1027/tcp   open        IIS
1030/tcp   open        iad1
2306/tcp   open        unknown
5631/tcp   open        pcanywheredata
7937/tcp   open        unknown
7938/tcp   open        unknown
36890/tcp  open        unknown
Nmap done: 1 IP address (1 host up) scanned in 2594.47 seconds
```

3.5 指定扫描的端口

在扫描过程中也可以指定扫描的端口。表 3-4 给出了端口的指定方法。

表 3-4　指定扫描端口的方法

指 定 端 口	指定端口的选项
扫描常见的 100 端口	-F [target]
指定某一个端口	-p [port]
使用名字来指定扫描端口	-p [name]
使用协议指定扫描端口	-p U：[UDP ports], T：[TCP ports
扫描所有端口	-p *
扫描常用端口	--top-ports [number]

1. 扫描常见的 100 个端口

命令语法：-F [target]

对目标 192.168.153.131 的 100 个常见端口进行扫描，命令如下。

```
nmap -F 192.168.153.131
```

2. 指定某一个端口

命令语法：nmap -p [port] [target]

对目标 192.168.153.131 的 80 端口进行扫描，命令如下。

```
nmap -p 80 192.168.153.131
```

3. 使用名字来指定扫描端口

命令语法：nmap -p [port name(s)] [target]

对目标 192.168.153.131 的 smtp、http 端口进行扫描，命令如下。

```
nmap -p smtp,http 192.168.153.131
```

4. 使用协议指定扫描端口

命令语法：nmap -p U:[UDP ports],T:[TCP ports] [target]

对目标 192.168.153.131 的 53 端口进行 UDP 扫描，25 端口进行 TCP 扫描，命令如下。

```
nmap  -sU -sT -p U:53,T:25 192.168.153.131
```

5. 扫描所有端口

命令语法：nmap -p“*”[target]

对目标 192.168.153.131 的所有端口进行扫描，命令如下。

```
nmap -p * 192.168.153.131
```

这个扫描方法消耗很大资源，要慎用。

6. 扫描常用端口

命令语法：nmap --top-ports [number] [target]

对目标 192.168.153.131 常见的 10 个端口进行扫描，命令如下。

```
nmap --top-ports 192.168.153.131
```

小结

本章介绍了端口的相关概念，并给出了 Nmap 中对端口进行扫描的各种技术。另外通过本章的学习，读者也学到了如何在扫描过程中自行决定扫描的端口。第 4 章将介绍 Nmap 的高级部分——服务扫描和操作系统扫描。