速查 | Nmap常用命令备忘录

0x01 目标规格

```
nmap 192.168.1.1 扫描一个IP
nmap 192.168.1.1 192.168.2.1 扫描IP段
nmap 192.168.1.1-254 扫描一个范围
nmap nmap.org 扫描一个域名
nmap 192.168.1.0/24 使用CIDR表示法扫描
nmap -iL target.txt 扫描文件中的目标
nmap -iR 100 扫描100个随机主机
nmap --exclude 192.168.1.1 排除列出的主机
```

0x02 扫描手法

```
nmap 192.168.1.1 -sS TCP SYN端口扫描(有root权限默认)
nmap 192.168.1.1 -sT TCP连接端口扫描(没有root权限默认)
nmap 192.168.1.1 -sU UDP端口扫描
nmap 192.168.1.1 -sA TCP ACK端口扫描
nmap 192.168.1.1 -sW 滑动窗口扫描
nmap 192.168.1.1 -sM TCP Maimon扫描
```

0x03 扫描手法

```
nmap 192.168.1.1-3 -sL 不扫描,仅列出目标
nmap 192.168.1.1/24 -sn 禁用端口扫描
nmap 192.168.1.1-5 -Pn 跳过主机发现,直接扫描端口
nmap 192.168.1.1-5 -PS22-25,80 端口X上的TCP SYN发现,默认80
nmap 192.168.1.1-5 -PA22-25,80 端口X上的TCP ACK发现,默认80
nmap 192.168.1.1-5 -PU53 端口X上的UDP发现,默认40125
nmap 192.168.1.1/24 -PR 本地网络上的ARP发现
nmap 192.168.1.1 -n 不做DNS解析
```

0x04 端口规格

```
nmap 192.168.1.1 -p 21 扫描特定端口
nmap 192.168.1.1 -p 21-100 扫描端口范围
nmap 192.168.1.1 -p U:53,T:21-25,80 扫描多个TCP和UDP端口
nmap 192.168.1.1 -p- 扫描所有端口
```

```
nmap 192.168.1.1 -p http,https 基于服务名称的端口扫描
nmap 192.168.1.1 -F 快速扫描(100个端口)
nmap 192.168.1.1 --top-ports 2000 扫描前2000个端口
nmap 192.168.1.1 -p-65535 从端口1开始扫描
```

0x05 时间和性能

```
nmap 192.168.1.1 -T0 妄想症,非常非常慢，用于IDS逃逸
nmap 192.168.1.1 -T1 猥琐的,相当慢，用于IDS逃逸
nmap 192.168.1.1 -T2 礼貌的,降低速度以消耗更小的带宽，比默认慢十倍
nmap 192.168.1.1 -T3 正常的,默认，根据目标的反应自动调整时间模式
nmap 192.168.1.1 -T4 野蛮的,在一个很好的网络环境，请求可能会淹没目标
nmap 192.168.1.1 -T5 疯狂的,很可能会淹没目标端口或是漏掉一些开放端口
```

0x06 NSE脚本

```
nmap 192.168.1.1 -sC 使用默认的NSE脚本进行扫描
nmap 192.168.1.1 --script=banner 使用单个脚本扫描,banner示例
nmap 192.168.1.1 --script=http,banner 使用两个脚本扫描,示例http,banner
nmap 192.168.1.1 --script=http* 使用通配符扫描,http示例
nmap 192.168.1.1 --script "not intrusive" 扫描默认值,删除侵入性脚本
nmap 192.168.1.1 --script=smb-vuln* 扫描所有smb漏洞
nmap 192.168.1.1 --script=vuln 扫描常见漏洞
```

0x07 输出

```
nmap 192.168.1.1 -v 增加详细程度,-vv效果更好
nmap 192.168.1.1 -oN test.txt 标准输出写入到指定文件中
nmap 192.168.1.1 -oX test.xml 将输入写成xml的形式
nmap 192.168.1.1 -oG grep.txt 将输出写成特殊格式
nmap 192.168.1.1 -oA results 将输出所有格式,有三种 .xml/.gnmap/.nmap
nmap 192.168.1.1 --open 仅显示开放的端口
nmap 192.168.1.1 -T4 --packet-trace 显示所有发送和接收的数据包
nmap --resume test.txt 恢复扫描，配合-oG等命令使用
```

0x08 服务和版本检测

```
尝试确定端口上运行的服务的版本
nmap 192.168.1.1 -sV

强度级别0到9，数字越大，正确性越强，默认值为7
nmap 192.168.1.1 -sV --version-intensity 8
```

```
轻量级版本扫描，使扫描进程加快，但它识别服务的正确率降低
nmap 192.168.1.1 -sV --version-light

version-all相当于version-intensity的最高级别9，保证对每个端口尝试每个探测报文
nmap 192.168.1.1 -sV --version-all

启用操作系统检测，版本检测，脚本扫描和跟踪路由...
nmap 192.168.1.1 -A

使用TCP/IP进行远程OS指纹识别
nmap 192.168.1.1 -O

当Nmap无法确定所检测的操作系统时，会尽可能地提供最相近的匹配
nmap 192.168.1.1 -O --osscan-guess
```

0x09 防火墙/IDS规避和欺骗

```
报文分段，请求的扫描(包括ping扫描)使用微小的碎片IP数据包，包过滤器检测更难
nmap 192.168.1.1 -f

利用数据包分片技术，某些防火墙为了加快处理速度而不会进行重组处理，这样从而逃脱防火墙或闯入检测系统的检测，注意:mtu的值必须是8的倍数(如8,16,24,32等)
nmap 192.168.1.1 --mtu 32

使用-D选项就可以达到IP欺骗的目的，可以指定多个IP或者使用RND随机生成几个IP地址
nmap -D 10.1.1.1,20.2.2.2 192.168.1.1
nmap -D RND:11 192.168.1.1

源地址欺骗，从Microso扫描Facebook
nmap -S www.microso.com www.facebook.com

指定源主机端口，来手动设定用来扫描的端口,常用的如20、53、67端口
nmap -g 53 192.168.1.1

通过HTTP/SOCKS4代理中继连接
nmap --proxies http://191.1.1.1:1080,http://192.2.2.2:1080 192.168.1.1

添加垃圾数据，通过在发送的数据包末尾添加随机的垃圾数据，以达到混淆视听的作效果，200是垃圾数据长度
nmap --data-length 200 192.168.1.1
```

```
伪装MAC地址，可以手动指定MAC地址的值。或者为了简单起见，可以填写数字0，这将生成一个随机的MAC地址
nmap --spoof-mac 0 192.168.1.1

伪造检验值，这将使用伪造的TCP/UDP/SCTP校验和发送数据
nmap --badsum 192.168.1.1
```

示例IDS规避命令

```
nmap -f -T0 -n -Pn --data-length 200 -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1
```

0x010 端口状态

open(开放的)：

应用程序正在该端口接收TCP 连接或者UDP报文。发现这一点常常是端口扫描 的主要目标。安全意识强的人们知道每个开放的端口 都是攻击的入口。攻击者或者入侵测试者想要发现开放的端口。而管理员则试图关闭它们或者用防火墙保护它们以免妨碍了合法用户。非安全扫描可能对开放的端口也感兴趣，因为它们显示了网络上那些服务可供使用。

closed(关闭的)：

关闭的端口对于Nmap也是可访问的(它接受Nmap的探测报文并作出响应)， 但没有应用程序在其上监听。它们可以显示该IP地址上(主机发现，或者ping扫描)的主机正在运行up 也对部分操作系统探测有所帮助。因为关闭的关口是可访问的，也许过会儿值得再扫描一下，可能一些又开放了。系统管理员可能会考虑用防火墙封锁这样的端口。那样他们就会被显示为被过滤的状态，下面讨论。

filtered(被过滤的)：

由于包过滤阻止探测报文到达端口， Nmap无法确定该端口是否开放。过滤可能来自专业的防火墙设备，路由器规则 或者主机上的软件防火墙。这样的端口让攻击者感觉很挫折，因为它们几乎不提供 任何信息。有时候它们响应ICMP错误消息如类型3代码13 (无法到达目标: 通信被管理员禁止)，但更普遍的是过滤器只是丢弃探测帧， 不做任何响应。这迫使Nmap重试若干次以访万一探测包是由于网络阻塞丢弃的。这使得扫描速度明显变慢。

unfiltered(未被过滤的)：

未被过滤状态意味着端口可访问，但Nmap不能确定它是开放还是关闭。只有用于映射防火墙规则集的ACK扫描才会把端口分类到这种状态。用其它类型的扫描如窗口扫描，SYN扫

描，或者FIN扫描来扫描未被过滤的端口可以帮助确定 端口是否开放。

open | filtered(开放或者被过滤的)：

当无法确定端口是开放还是被过滤的，Nmap就把该端口划分成 这种状态。开放的端口不响应就是一个例子。没有响应也可能意味着报文过滤器丢弃 了探测报文或者它引发的任何响应。因此Nmap无法确定该端口是开放的还是被过滤的。UDP，IP协议， FIN，Null，和Xmas扫描可能把端口归入此类。

closed | filtered(关闭或者被过滤的)：

该状态用于Nmap不能确定端口是关闭的还是被过滤的。它只可能出现在IPID Idle扫描中。