# Windows下使用nmap端口扫描

端口状态一般有以下几种：

|  |  |
| --- | --- |
| 状态 | 详细的参数说明 |
| Open | 端口是开启的，数据已经到达主机，并且在这个端口上有程序在接收数据 |
| Closed | 端口关闭，数据到达了主机，但是没有程序接收 |
| Filtered | 数据没有到达主机，返回空；表明数据被防火墙或者IDS过滤掉 |
| Unfiltered | 数据已经到达主机，没有被过滤掉，但是无法识别当前端口的状态 |
| Open|filtered | 端口没有返回值 |
| Closed|filtered | 发生在IP ID idle扫描中 |

### Nmap 常用参数

|  |  |
| --- | --- |
| -F | 扫描最常用的100个端口 |
| -v | 显示详细的扫描到的信息 |
| -n | 不进行域名解析，可以加快扫描速度 |
| -g port | 指定源端口，使用本机上特定的端口进行扫描 |
| -spoofmac addr | 伪造物理地址，使用随机物理地址来扫描 |
| -N | 执行反向DNS解析 |
| -O | 扫描操作系统类型和版本 |
| -scanflags | 自定义扫描包 |
| -A | 激进型扫描，启用很多扫描选项，它会做一个多类型扫描，但很容易被发现；这个命令启用了路由跟踪，会显示跟踪到的路由 |
|  |  |
| -log-errors | 输出错误日志 |
| -packet-trace | 获取从当前主机到目标主机的所有节点 |
| -p port | 指定监听某个端口的状态，也可以是某个范围内的端口，eg:port为10-1000 |
| -D ip1,ip2,ME | 同时使用多个诱饵IP进行扫描，把自己的ip藏在诱饵之中 |
| -S ip | 源地址欺骗：通过将自己的ip伪装成他人的ip进行扫描；让防火墙误认为是其他ip在使用； |
| -e | 选择发送和接受数据的网口 |
| -T | 时间优化，通过优化时间来提高通过防火墙和IDS的 |
| -f | 将可疑的探测包进行分片（比如将tcp包拆分成多个ip包），某些简单的防火墙为了加快处理速度，可能不会进行重组检查，以此来避开检查 |
| 全部参数 | <https://www.2cto.com/article/201203/125686.html> |

### 常用扫描类型

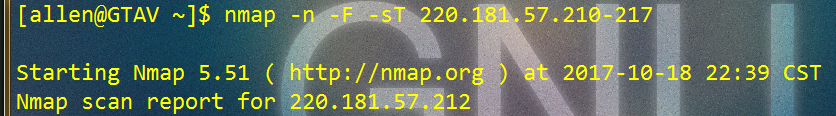
|  |  |
| --- | --- |
| -sl | 空闲扫描https://www.2cto.com/article/201505/396631.html |
| -sT | 使用TCP扫描，默认选项；但最容易被防火墙拦截 |
| -PE | 使用ICMP协议进行扫描 |
| -sS | 使用SYN扫描，这是最隐秘的一种扫描方式，半开放扫描 |
| -sA | 使用ACK扫描，检查端口是否开放，探测防火墙 |
| -sP | 使用ping扫描，快速发现网络 |
| -sR | PRC扫描，定位PRC，记录成功扫描到的机器 |
| -sU | UDP扫描，确定特定的UDP端口是否开放 |
| -sX | XMAS扫描，扫描特定配置的防火墙 |
| -sL | 列出扫描对象，列出要扫描的ip，使用-n选项确保不向网络中发送数据包 |
| -sO | 使用ip协议扫描，确定使用ip协议的主机 |
| -sM | FIN/ACK,隐蔽扫描，适用于Unix系统，查找RST数据包 |
| -sI | 闲置扫描，僵尸主机扫描，非常隐蔽 |

### 常用输出格式

|  |  |
| --- | --- |
| -oN | 输出为常规格式 |
| -oX | 输出为XML文件 |
| -oA | 输出为可检索的，常规的，xml文件 |
| -oG | 可检索格式 |
|  |  |

### 执行过程

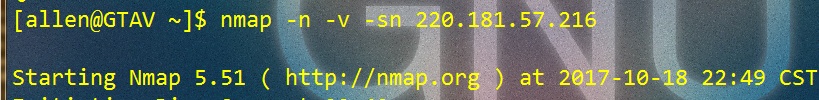
扫描网段内的所有机器，查看是否开启机器



扫描后的结果为：



只扫描机器，查看是否在线，不扫描端口



### 要点

要注意使用多种不同的扫描进行信息收集，单一的扫描可能会丢失重要信息；最好的情况是，使用多种扫描，尽可能多的收集目标主机信息，对结果进行判断，有助于下一步的工作；

### 扫描类型介绍

|  |  |
| --- | --- |
| 隐蔽扫描 | |
| 概念 | 网络扫描包括发送特殊的数据包，然后查看返回的结果进行判断；我们有时候需要对抗ids的异常行为，比如流量的异常检测；所以控制发送包的时间和速度就非常重要，这里有几个参数用来控制发送包的时间和速度 |
| 参数 | -T（0-5）： 控制扫描进度，0表示最温和，5表示最激进，5只能用在局域网里  -max-hostgroup 1: 控制主机的扫描数量为每次一个  max-parallelism 10: 一次仅仅允许10个探测请求；  scan-delay: 两次探测之间要停顿 |
|  |  |

|  |  |
| --- | --- |
| TCP扫描 | |
| 概念 | 使用tcp的三次握手来进行扫描 |
| 特点 | 扫描速度快，准确性高，对操作者没有权限上的要求，但是容易被防火墙和IDS发现 |
| 原理 | Tcp三次握手   1. 客户端发送syn； 2. 服务器端返回syn+ack，表明端口开放 3. 客户端返回ack，表明建立连接 4. 客户端主动断开连接 |

示意图：

服务器

SYN

客户端

SYN+ACK

ACK

|  |  |
| --- | --- |
| SYN扫描 | |
| 概念 | 使用syn进行扫描 |
| 特点 | 这是最隐秘的一种扫描方式，因为客户端和服务器端连接的过程中，没有形成三次握手，所以不会被防火墙检测到或者留下日志记录，但是需要root权限；Windows下没有root的概念，这里指Linux |
| 原理 |  |

示意图

SYN

服务器

客户端

SYN+ACK

RST

|  |  |
| --- | --- |
| NULL扫描 | |
| 概念 | 这是一种反向扫描法，通过发送一个没有标志位的数据包给服务器，等待服务器的响应 |
| 特点 | 这种扫描方法更加隐蔽，但准确度也是最低的，这种方式只用来判断是否为Windows系统，因为Windows系统不遵循RFC793标准，不论端口是否开启，都返回RST包 |
| 原理 |  |

示意图

NULL(没有标志位)

服务器

客户端

RST

注意：不建议使用这种扫描，因为这种扫描准确率很低，而且很容易被过滤；如果要判断操作系统，只需要加上-O 参数即可

|  |  |
| --- | --- |
| Ack扫描 | |
| 概念 | 通过发送一个ack包给服务器，不论主机的端口是否开启，都会返回一个RST包，通过包中的ttl值判断端口是否开启 |
| 特点 |  |
| 原理 | Rst包中的ttl值如果大于64，表明端口关闭 |

示意图

ACK

服务器

客户端

RST(TTL<=64)

注意：也可以自定义扫描包，下图演示这种用法

