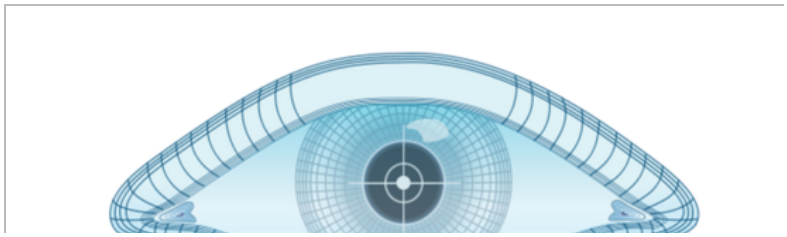
你不知道的 Nmap 之空闲扫描_酒仙桥六号部队 - MdEditor

> 你不知道的 Nmap 之空闲扫描

前言

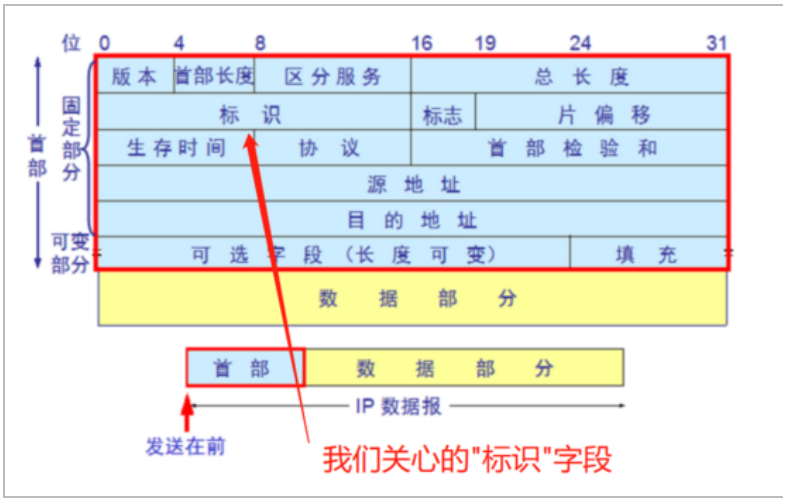
在渗透测试的过程中，我们会经常会用到 Nmap 进行信息收集。但是 Nmap 存在一个缺点就是在进行探测过程中会向目标发送大量数据包，从而产生大量流量，这样极其容易引起目标警觉，甚至追踪到渗透测试者的真实 IP 地址。那我们该如何做才能做到既隐藏了自己的真实 IP 地址同时又能实现我们信息收集的任务呢?

Nmap 中有一种比较强大的扫描方式是空闲扫描（Idle Scan），命令是 – sI（I 为 i 的大写）。这种技术是利用空闲主机欺骗目标主机 IP 并且隐藏本机真实 IP。



IP 报文中的 ID 及 TCP 握手

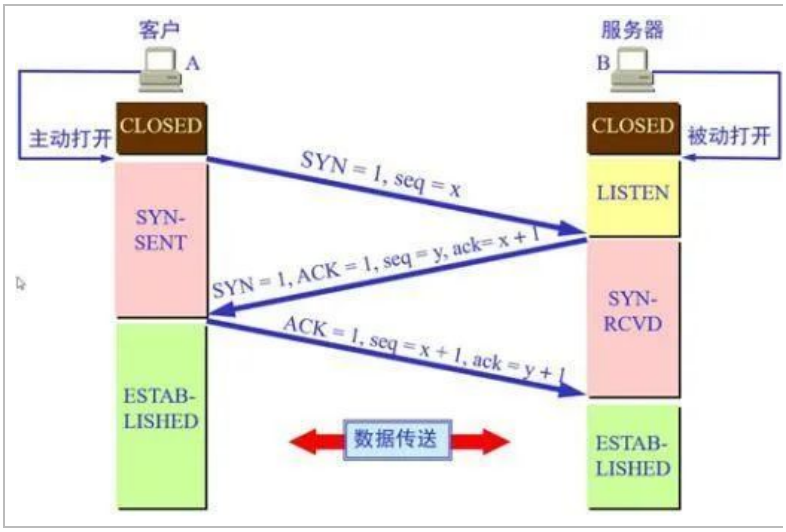
空闲扫描利用了 IP 协议报文中的 ID 和 TCP 协议通信原理。首先我们先来看下 IP 协议报文结构：



标识：唯一的标识主机发送的每一分数据报。通常每发送一个报文，它的值 + 1。当 IP 报文长度超过传输网络的 MTU（最大传输单元）时必须分片，这个标识字段的值被复制到所有数据分片的标识字段中，使得这些分片在达到最终目的地时可以依照标识字段的内容重新组成原先的数据。

就是说当我们发送的 IP 报文未超过 MTU 时，通常每个报文的标识会 + 1，当然这个是该端口所有 IP 报文公用的。当我们向 10 个目标及端口发送 IP 包时，每个报文的标识会依次递增 + 1。这也是为什么我们需要空闲的主机的原因，可以根据 IP 报文中的标识来推测扫描结果。

TCP 协议正常三次握手：



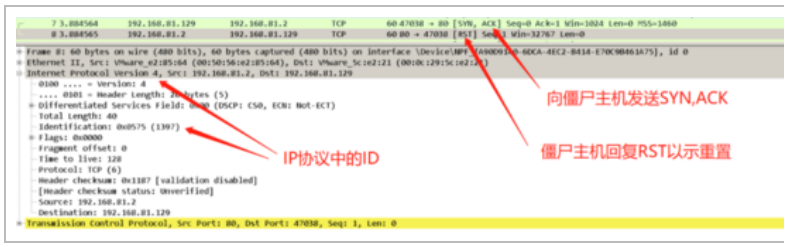
熟悉 TCP 协议的同学知道 TCP 在建立链接的时候会有三次握手行为。除了正常的控制位，还有一些用于其他情况的标识位如：RST

RST：重置连接标志，用于重置由于主机崩溃或其他原因而出现错误的连接。或者用于拒绝非法的报文段和拒绝连接请求。

当打开的 TCP 端口接收到非法报文会回复 RST 以示对面重置该连接。空闲扫描也正是利用了这点达到目的。

空闲扫描原理

空闲扫描利用 TCP 的通信原理：当直接发送 SYN,ACK 包时目标会因为握手流程不合法，所以会回复 RST 包以重置。但此时回复的包中会带有目标 IP 包中的 ID。

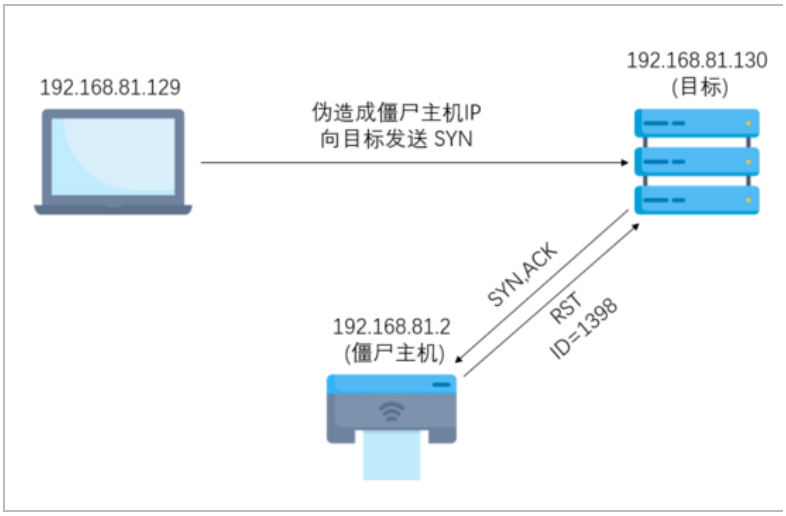


第一步：

向僵尸主机开放的 TCP 端口（如 80 端口的 HTTP 服务）发送 SYN,ACK 包，僵尸主机会回复 RST。僵尸主机回复的报文中的 IP 协议中 ID 为 1397。

第二步：

伪造僵尸主机的 IP（192.168.81.2）向目标的端口发送 SYN 报文，如果该端口开放会按照 TCP 协议握手流程向僵尸主机回复 SYN,ACK 报文。但是僵尸主机收到的第一个报文为 SYN,ACK 流程不合法，会回复 RST 并且 ID 会 +1 。



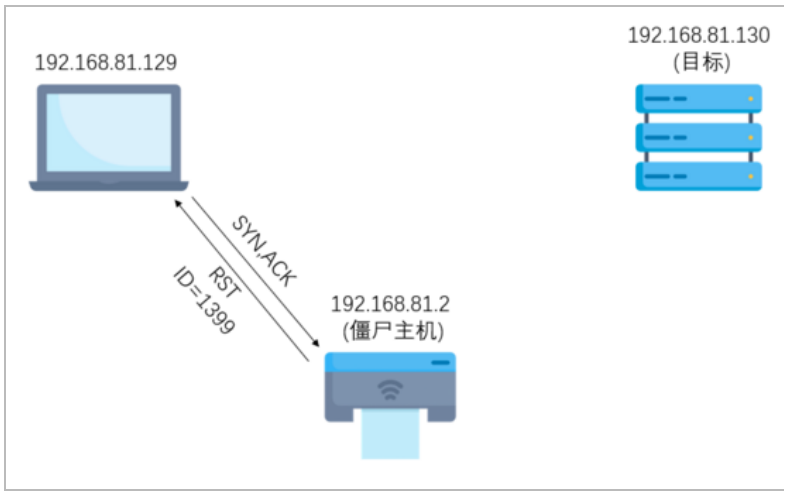
第三步：

这时候我们重复第一步的流程，向僵尸主机开放的 TCP 端口（如 80 端口的 HTTP 服务）发送 SYN ACK 包，僵

端口（如 80 端口的 HTTP 服务）发送 SYN,ACK 包，僵尸主机会回复 RST。我们可以根据僵尸主机回复的报文中的 IP 协议中 ID 来判断目标主机跟僵尸主机是否产生了通信：

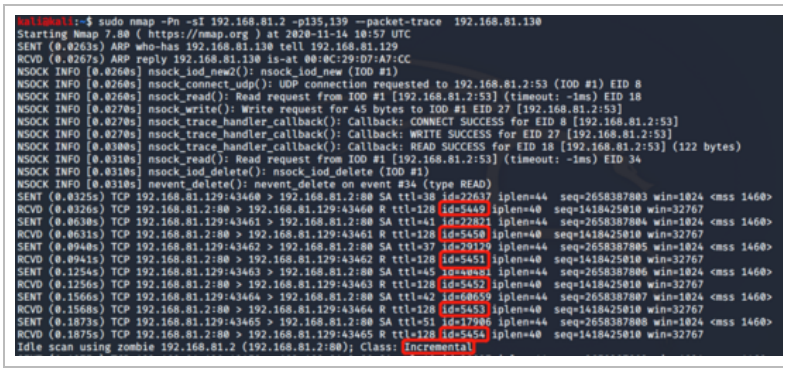
如果 ID=1399（跟 1397 比 + 2），目标端口跟僵尸主机产生过通信，故目标端口开放。

如果 ID=1398（跟 1397 比 + 1），目标端口跟僵尸主机未产生过通信，故目标端口未开放。

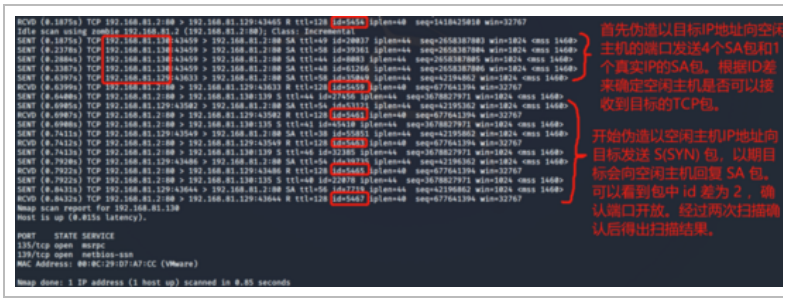


Nmap 空闲扫描算法实现

虽然我们对原理进行了阐述，但是 Nmap 在实际中的实现要复杂一些。这时可以利用包追踪的方式来理解 Nmap 的实现。Nmap 的 ---packet-trace 选项可以显示出包追踪的详情：



可以看到 Nmap 首先对空闲僵尸主机 192.168.81.2 尝试发送了 6 个 SA（SYN，ACK）的 TCP 包，空闲僵尸主机回复了 6 个 R（RST）的 TCP 包。6 个回复的 RST 包中的 id 为 5449–5454，Nmap 确认其类型为递增，开始进行下一步。



Nmap 会伪造以目标 IP 地址（192.168.81.130）向空闲主机端口发送了 4 个 SA（SYN，ACK） 包和 1 个以真实 IP（192.168.81.129）向空闲主机端口发送的的 SA（SYN，ACK）包。发送真实 IP 的 SA 包用来接收空闲主机发回来的 RST 包，用 RST 包的 id 来确定之前发送

的 4 个伪造以目标 IP 的包空闲主机是否接收并产生交互。我们可以看到 RST 包中 id 为 5459 而扫描之前 id 为 5454，相差 5，正好是 4 个伪造包

- 1 个真实包。所以 Nmap 认为目标和空闲主机之间是可以通信交互的。

最后就是利用原理进行扫描：Nmap 开始伪造以空闲主机 IP 地址（192.168.81.2）向目标发送 SYN 包，以期待目标（192.168.81.130）接收到以空闲主机 IP

（192.168.81.2）的 SYN 包后，按照 TCP 握手协议来向空闲主机（192.168.81.2）发送第二次握手的 SYN，ACK 包。空闲主机直接接收到 SYN，ACK, 判定握手不合法会回复 RST 包，并且包中 id+1。Nmap 以真实 IP 向空闲主机发送 SYN，ACK 包，空闲主机回复 RST 包，包中 id 再一次 + 1。从图中而可以看到扫描的端口的包中 id 从 5459–5461，5463–5461 等均相差为 2 ，则可认为目标端口开放。

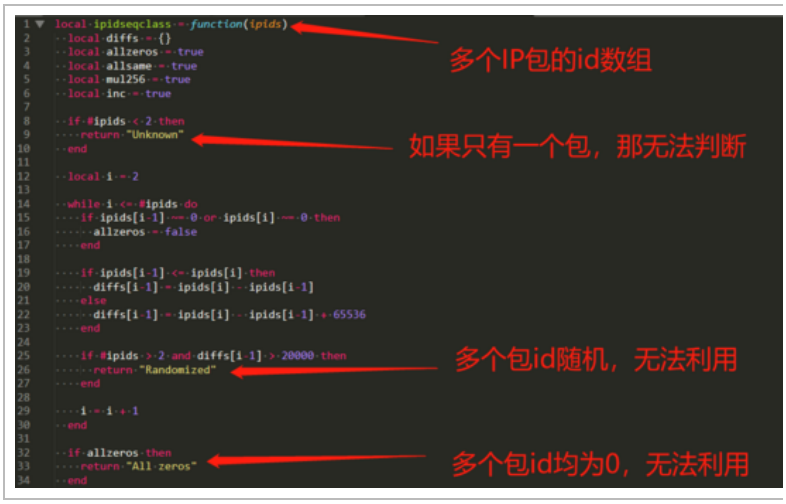
ipidseq 脚本

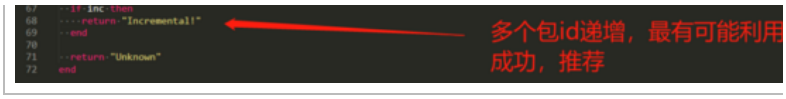
Nmap 提供基于该框架下的 NSE（Nmap ScriptEngine）脚本来进行扫描时的自定义扩展。NSE 能够完成网络发现、复杂版本探测、脆弱性探测、简单漏洞利用等功能。

在我们寻找空闲僵尸主机的时候可以使用官方脚本 ipidseq 来帮助进行寻找。

地址：
https://svn.nmap.org/nmap/scripts/ipidseq.nse

我们来看下其中的基本实现和判断，主要判断实现在 ipidseqClass 中：





整体看来如果结果是 Incremental!（递增），是最优选择。如果没有则 Brokenincremental!（损坏递增）勉强可堪一用，但扫描结果不太保证。

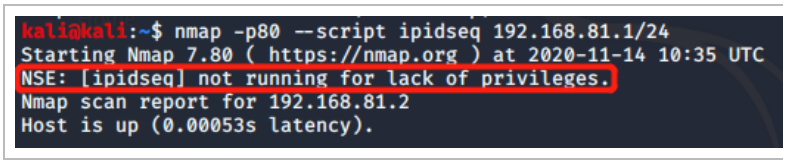
Nmap 空闲扫描实战中的注意事项

首先找到一个空闲的僵尸主机，我们可以使用上一节提到的 ipidseq 脚本来进行探测寻找。

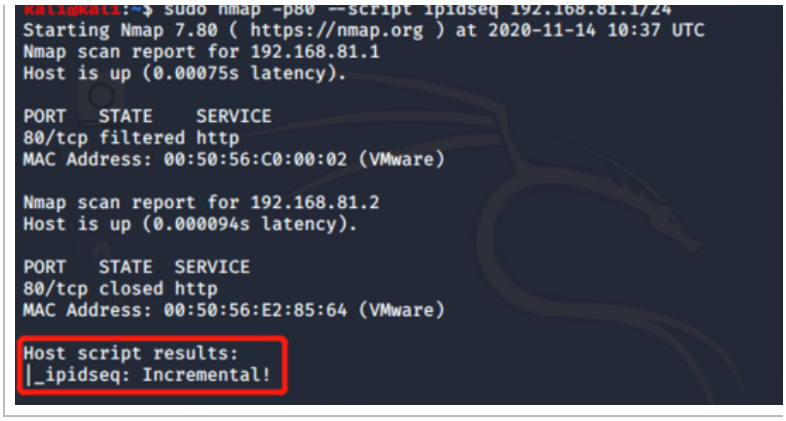
探测网段中空闲主机的命令为：

nmap --script ipidseq 192.168.81.1/24

直接使用会报没有权限的问题：



我们需要使用 sudo ：

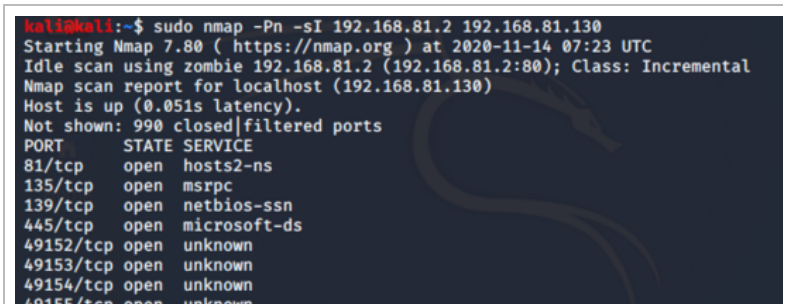


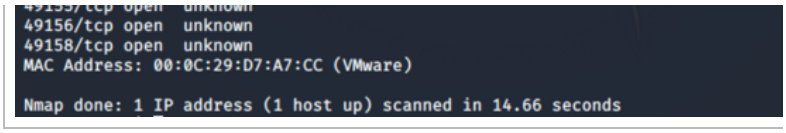
可以看到我们运气很不错，探测 192.168.81.2 的 80 端口在 ipidseq 脚本结果为 Incremental!（递增），这代表我们可以尝试使用该台主机的 80 端口作为空闲僵尸主机。

我们直接使用 192.168.81.2 作为空闲僵尸主机对目标 192.168.81.130 进行空闲扫描

一般的使用命令为：

nmap –Pn –sI 192.168.81.2 92.168.81.130





等一下之后即可看到利用成功，扫描结果也显示出来了。

在实战中如果该空闲主机不可用，则可能会报以下类型的错误：

这时候就需要更换空闲主机。

总结

虽然 Nmap 提供欺骗扫描技术（–D）来帮助用户保护自己的身份，但是这种扫描（不像空闲扫描）仍需要攻击者使用自己真实的 IP 发送很多的数据包以便获取扫描结果。空闲扫描的优点之一是即使入侵检测系统若发出警报，则会报告空闲僵尸主机已开始对他们扫描。因此可以用该种扫描技术给其他主机栽赃。当然默认情况下空闲扫描虽然可以伪造 IP 地址进行发包，但是 MAC 地址依然是真实主机的，所以在检测和防御时可以以此为依据机型判断。

随着攻防对抗的升级无论何种扫描形式最终都会被捕获察觉。当我们在研究原理和实现之后，再不断地进行优化，保持不断地自我更新才能在这日益月薪攻方的浪潮中立于前方。一起加油吧~

全文完