

终端安全

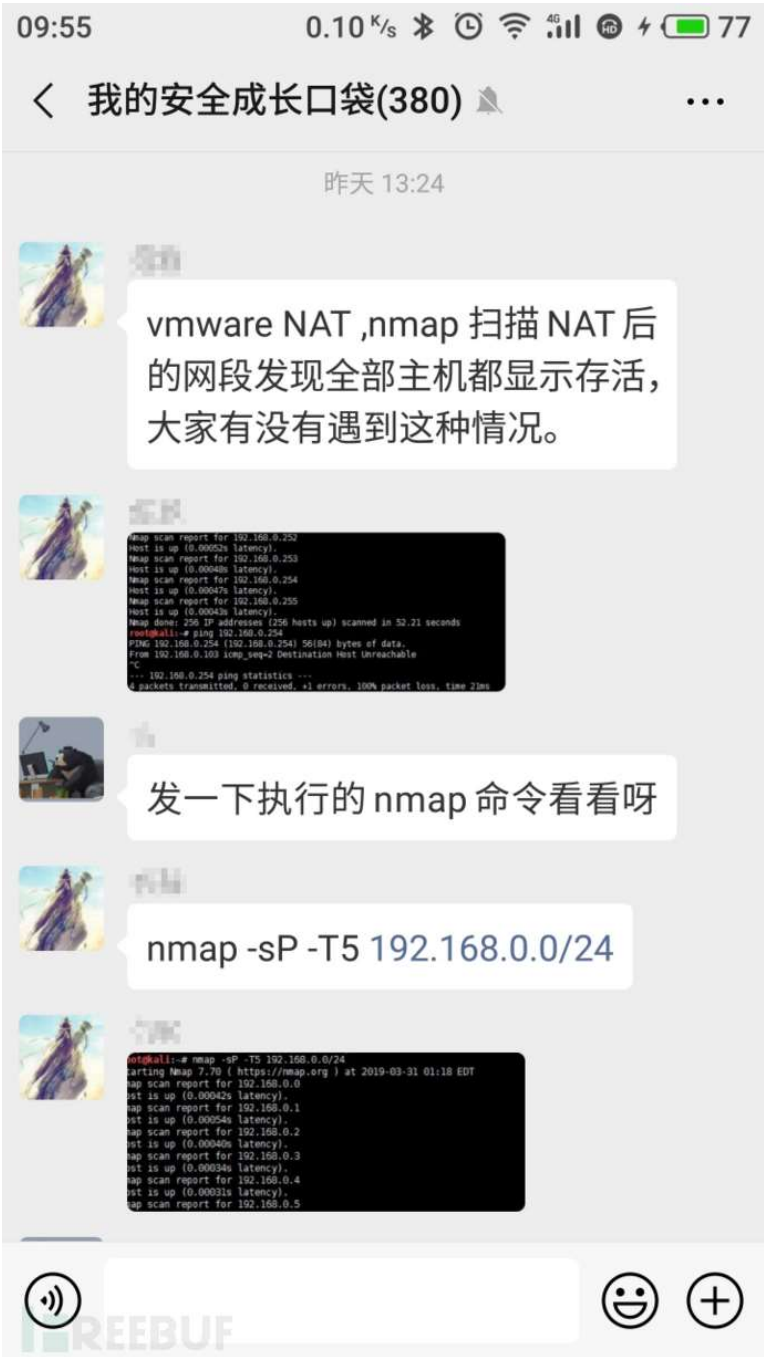
# Nmap在VMware NAT网络下探测主机存活误报的分析

Shad0wpf\_ 2019-04-08 09:00:16 222121 9

\*本文作者: Shad0wpf\_，本文属 FreeBuf 原创奖励计划，未经许可禁止转载。

## 起因

昨天，在某安全交流群，看到关于Nmap扫描的讨论，对一个段做主机存活扫描，发现主机全部存活。



扫描源主机是安装在虚拟机上的Kali，网络连接为NAT模式，使用-sP参数对192.168.0.0/24网段扫描，探测存活主机，返回结果为256个IP全部在线，而实际情况是该网段只有几个IP存活，使用Ping命令，其它不在线的IP是Ping不通的。

## 文章目录

- 起因
- 分析
- 解决
- 参考
- 后记

11 文章数 32 评论数 7 关注者

- MongoDB未授权访问漏洞分析及整改建议  
2019-09-11
- 技术分享 | 无需四次握手包破解WPA & WPA2密码  
2018-08-15
- 经验分享 | Burpsuite抓取非HTTP流量  
2018-01-04

浏览更多

广告 腾讯安全生态大会

Tencent 腾讯 | 腾讯安全

腾讯全球  
数字生态大会

当云成为安全主战场，看腾讯怎么做？  
9月11日9:30 | C55安全领袖峰会-产业专场

点击查看

终端安全

👍

💬 9

🔗

Nmap scan report for 192.168.0.254  
Host is up (0.00047s latency).  
Nmap scan report for 192.168.0.255  
Host is up (0.00043s latency).  
Nmap done: 256 IP addresses (256 hosts up) scanned in 52.21 seconds  
root@kali:~# ping 192.168.0.254  
PING 192.168.0.254 (192.168.0.254) 56(84) bytes of data.  
From 192.168.0.103 icmp\_seq=2 Destination Host Unreachable  
^C  
--- 192.168.0.254 ping statistics ---  
4 packets transmitted, 0 received, 100% packet loss, time 21ms

分析

难道是Nmap出错了？带着这个疑问，我查阅了Nmap的官方文档，Nmap使用-sn参数指定只做主机存活发现（以前的一些版本中使用-sP，新版本中-sn和-sP都可以使用，为同一参数），不做端口扫描。Nmap默认的主机存活发现使用了以下四种方式：

ICMP echo request  
TCP SYN to port 443  
TCP ACK to port 80  
ICMP timestamp request

只要其中一个主机存活的特征存在，Nmap判断为主机存活。

Linux下默认Ping命令使用的ICMP echo请求，现在Ping不通，那第一种请求是没有主机存活特征的，问题可能是在后面的三种请求。

抓个包分析吧。

Wireshark packet capture showing a TCP RST response from 192.168.0.103 to 192.168.0.10. The packet details show: Source Port: 80, Destination Port: 40437, Sequence number: 1, Window size value: 32767. The packet is identified as a RST (Reset) flag.

扫描源IP为10.10.10.128，筛选了其中一个不存在的IP 192.168.0.10。ICMP echo请求ICMP timestamp请求没有响应，但奇怪的是，到80端口的ACK请求，有一个RST标志的响应，窗口大小是32767。

正常情况下，数据包发送到一个不存在的IP，不会收到任何响应。

TCP连接端口未监听、请求超时、异常关闭等几种情况会发送RST标志的响应。IP存在，但端口关闭，收到ACK标志的包时，会返回一个RST标志的包，内容为Seq=1，Win=0，Len=0。

这里返回了一个RST标志的包，而且窗口大小不为0，Nmap认为该端口开放（参考Nmap的TCP Windows Scan），也就是这个包导致了Nmap对主机存活的误判断。这个包是哪来的呢？网络结构很简单，除了VMware的NAT外，没有其它网络设备，说明这个包是NAT返回的。

解决

现在已经知道了引起Nmap误判断的原因是VMware的NAT，那用什么方法来避免呢？**方法一：**只使用ICMP echo请求探测（-PE）探测主机时，添加-PE参数，nmap只发送一个ICMP echo请求。

nmap -sn -PE -n 192.168.0.0/24

现在可以扫描出准确的主机存活信息了。

文章目录

起因

分析

解决

参考

后记

📝

🔍

📱

🔔

请 登录 / 注册 后在FreeBuf发布内容哦

👍 0    💬 9    + 收入专辑    ...

https://www.freebuf.com/news/199711.html

2/6

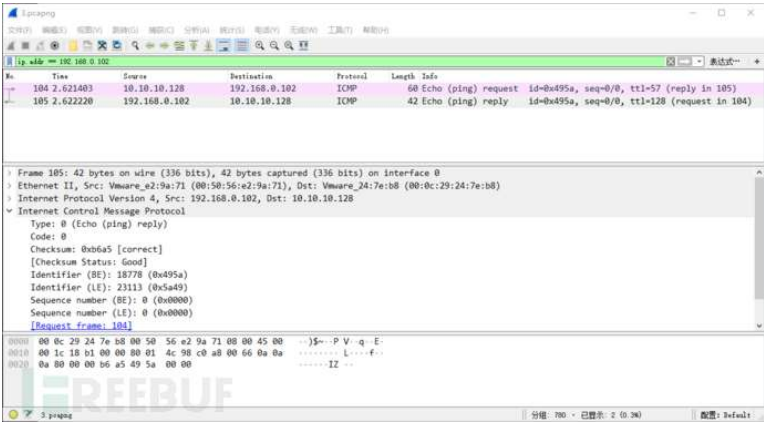
终端安全

9

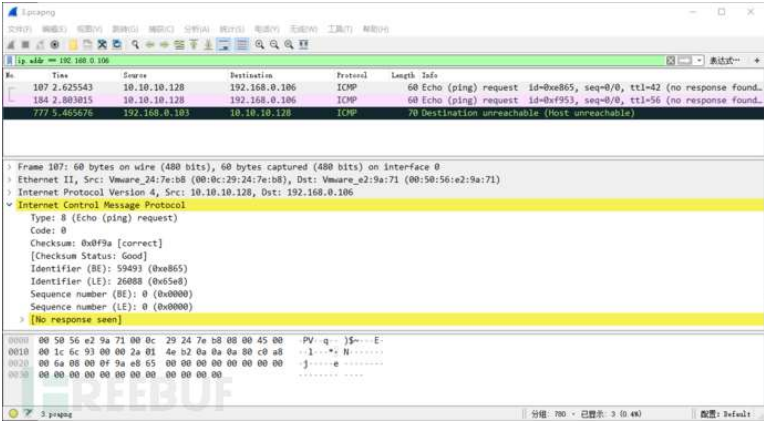
```
root@kali:~# nmap -sn -PE -n 192.168.0.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-31 10:50 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00049s latency).
Nmap scan report for 192.168.0.100
Host is up (0.090s latency).
Nmap scan report for 192.168.0.101
Host is up (0.089s latency).
Nmap scan report for 192.168.0.102
Host is up (0.0012s latency).
Nmap scan report for 192.168.0.103
Host is up (0.00028s latency).
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.94 seconds
```

-n参数禁用域名解析，避免浪费大量实在在域名解析等待上。

过滤其中一个存活的IP192.168.0.102，Nmap发送一次ICMP echo请求，收到102的ICMP响应后，判断主机存活，不在发送其它数据包。



不存活的IP192.168.0.106，发现两次ICMP echo请求，仍未收到106的ICMP响应，则判断为主机不存活。



不过这种方式存在一个问题，Nmap最多发送两次ICMP echo请求，主机禁Ping或者丢包时造成漏报。

方法二：VMware虚拟机网络连接方式改为桥接

既然问题是VMware的NAT引起的，那就索性不用NAT了，改成桥接，这也是VMware官方的建议。

文章目录

起因

分析

解决

参考

后记

终端安全

设备

内存

处理器

硬盘 (SCSI)

CD/DVD (IDE)

网络适配器

USB 控制器

显示器

摘要

8 GB

4

100 GB

正在使用文件 E:\Software\OS\...

自定义 (VMnet2)

存在

自动检测

设备状态

☒ 已连接(C)

☒ 启动时连接(O)

网络连接

☒ 桥接模式(B): 直接连接物理网络

☒ 复制物理网络连接状态(P)

☐ NAT 模式(N): 用于共享主机的 IP 地址

☐ 仅主机模式(H): 与主机共享的专用网络

☐ 自定义(U): 特定虚拟网络

VMnet2

☐ LAN 区段(L):

LAN 区段(S)...

高级(V)...

文章目录

起因

分析

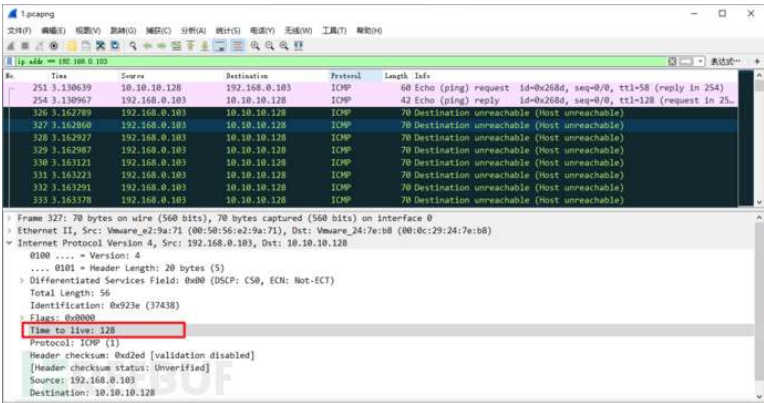
解决

参考

后记

其实，VMware网络连接使用桥接模式，好处不只是解决了Nmap对主机存活的误判断，也避免了无法tracert和操作系统判断不准确的问题，可以直接创建反弹shell监听端口，网络性能上也优于NAT。

影响操作系统判断的原因，这里提一下，经过VMware NAT返回的数据包，TTL永远是128，即使对方操作系统是Linux。



下图是两种网络连接模式，使用-O参数探测操作系统是的结果对比。



所以，作为渗透测试的虚拟机，还是直接用桥接网卡吧。

参考

- Nmap 主机发现扫描: <https://nmap.org/book/man-host-discovery.html>
- Nmap TCP Window Scan (-sW): <https://nmap.org/book/scan-methods-window-scan.html>
- 不同操作系统的默认TTL: <https://subinsb.com/default-device-ttl-values/>

后记

至于VMware NAT为什么会存在这些问题，由于个人的时间精力有限、对TCP/IP/NAT等各种协议的理解有限，未做深入研究。