警示：实验报告如有雷同，雷同各方当次实验成绩均以 0 分计；在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计；实验报告文件以 PDF 格式提交。

院系	数据院	班级	1 班	学号	18342048	姓名	李佳
完成日期： 2020 年 12 月 16 日							

网络扫描实验

【实验目的】

1. 掌握网络扫描技术的原理。
2. 学会使用 Nmap 扫描工具。

【实验环境】

实验主机操作系统：windows10 IP地址：172.26.21.83
目标机操作系统：windows10 IP地址：172.18.41.240
网络环境：局域网。

【实验工具】

Nmap (Network Mapper，网络映射器) 是一款开放源代码的网络探测和安全审核的工具。其设计目标是快速地扫描大型网络，也可以扫描单个主机。Nmap 以新颖的方式使用原始 IP 报文来发现网络上的主机及其提供的服务，包括其应用程序名称和版本，这些服务运行的操作系统包括版本信息，它们使用什么类型的报文过滤器/防火墙，以及一些其它功能。虽然 Nmap 通常用于安全审核，也可以利用来做一些日常管理维护的工作，比如查看整个网络的信息，管理服务升级计划，以及监视主机和服务的运行。

【实验过程】（要有实验截图）

假设以下测试命令假设目标机 IP 是 172.16.1.101。
在实验过程中，可通过 Wireshark 捕获数据包，分析 Nmap 采用什么探测包。

1．主机发现：进行连通性监测，判断目标主机。

假设本地目标 IP 地址为 172.16.1.101，首先确定测试机与目标机物理连接是连通的。

① 关闭目标机的防火墙，分别命令行窗口用 Windows 命令

Ping 172.16.1.101

和 Nmap 命令

nmap -sP 172.16.1.101

进行测试，记录测试情况。简要说明测试差别。

- Windows 命令

```
C:\WINDOWS\system32>ping 172.18.41.240

正在 Ping 172.18.41.240 具有 32 字节的数据:
来自 172.18.41.240 的回复: 字节=32 时间=5ms TTL=60
来自 172.18.41.240 的回复: 字节=32 时间=5ms TTL=60
来自 172.18.41.240 的回复: 字节=32 时间=5ms TTL=60
来自 172.18.41.240 的回复: 字节=32 时间=6ms TTL=60

172.18.41.240 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 5ms，最长 = 6ms，平均 = 5ms
```

- Nmap 命令

```
D:\Program Files (x86)\Nmap>nmap -sP 172.18.41.240
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-16 23:44 ?D1ú±ê×?ê±??
Nmap scan report for 172.18.41.240
Host is up (0.017s latency).
Nmap done: 1 IP address (1 host up) scanned in 16.92 seconds
```

Windows ping 命令通过将 ICMP(Internet 控制消息协议)回显数据包发送到计算机并侦听回显回复数据包来验证与一台或多台远程计算机的连接。每个发送的数据包最多等待一秒，打印已传输和接收的数据包数。通过 Wireshark 捕获数据包，可以看到就是 ping 就是单纯地发送了四个 ICMP 回显数据包。

9548	658.983877	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=62/15872, ttl=128 (reply in 9549)
9549	658.989292	172.18.41.240	172.26.21.83	ICMP	74	Echo (ping) reply id=0x0100, seq=62/15872, ttl=60 (request in 9548)
9550	659.987295	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=63/16128, ttl=128 (reply in 9551)
9551	659.992571	172.18.41.240	172.26.21.83	ICMP	74	Echo (ping) reply id=0x0100, seq=63/16128, ttl=60 (request in 9550)
9552	660.991419	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=64/16384, ttl=128 (reply in 9553)
9553	660.996614	172.18.41.240	172.26.21.83	ICMP	74	Echo (ping) reply id=0x0100, seq=64/16384, ttl=60 (request in 9552)
9556	661.995618	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=65/16640, ttl=128 (reply in 9557)
9557	662.001796	172.18.41.240	172.26.21.83	ICMP	74	Echo (ping) reply id=0x0100, seq=65/16640, ttl=60 (request in 9556)

Nmap 命令用于找出主机是否是存在在网络中。-sP 选项意味没有端口扫描也称为 Ping 扫描（主机发现）。通过 Wireshark 捕获数据包，可以看到探测过程如下：

1) 发送普通 ICMP 请求包【类型字段为 8，代码字段为 0】;
2) 向 443 端口发送 TCP SYN 包；
3) 向 80 端口发送 TCP ACK 包 ；
4) 发送时间戳 ICMP 请求包【类型字段为 13，代码字段为 0】；等方式判断主机状态

8920	542.716150	172.26.21.83	172.18.41.240	ICMP	42	Echo (ping) request id=0xa179, seq=0/0, ttl=49 (reply in 8922)
8921	542.727632	172.26.21.83	172.18.41.240	TCP	58	38026 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8922	542.728445	172.18.41.240	172.26.21.83	ICMP	60	Echo (ping) reply id=0xa179, seq=0/0, ttl=60 (request in 8920)
8923	542.735536	172.26.21.83	172.18.41.240	TCP	54	38026 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
8924	542.737372	172.18.41.240	172.26.21.83	TCP	60	443 → 38026 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
8925	542.739215	172.26.21.83	172.18.41.240	ICMP	54	Timestamp request id=0x40a0, seq=0/0, ttl=58
8926	542.741311	172.18.41.240	172.26.21.83	TCP	60	80 → 38026 [RST] Seq=1 Win=0 Len=0
8927	542.744452	172.18.41.240	172.26.21.83	ICMP	60	Timestamp reply id=0x40a0, seq=0/0, ttl=60
8929	543.737277	172.18.41.240	172.26.21.83	TCP	60	[TCP Retransmission] 443 → 38026 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0…
8930	545.737538	172.18.41.240	172.26.21.83	TCP	60	[TCP Retransmission] 443 → 38026 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0…
8938	549.737816	172.18.41.240	172.26.21.83	TCP	60	[TCP Retransmission] 443 → 38026 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0…
8974	557.738351	172.18.41.240	172.26.21.83	TCP	60	[TCP Retransmission] 443 → 38026 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0…

有主机存活时整个过程描述如下：

1) ICMP 回应请求
2) TCP SYN 到端口 443
3) ICMP 回应回复

4) TCP ACK 到端口 80
5) TCP SYN，ACK 到端口 443
6) TCP RST 到端口 80
7) ICMP 时间戳请求
8) ICMP 时间戳答复

② 开启目标机的防火墙，重复①，结果有什么不同？请说明原因。

- Windows 命令

```
C:\WINDOWS\system32>ping 172.18.41.240

正在 Ping 172.18.41.240 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

172.18.41.240 的 Ping 统计信息:
    数据包: 已发送 = 4，已接收 = 0，丢失 = 4 (100% 丢失)，
```

- Nmap 命令

```
D:\Program Files (x86)\Nmap>nmap -sP 172.18.41.240
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-16 23:39 ?D1ú±ê×?ê±??
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.42 seconds
```

Ping 测试结果为请求超时，nmap 测试结果为无主机存活。

20723	2048.318427	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=66/16896, ttl=128 (no response found!)
20739	2053.151077	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=67/17152, ttl=128 (no response found!)
20775	2058.150877	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=68/17408, ttl=128 (no response found!)
20793	2063.151672	172.26.21.83	172.18.41.240	ICMP	74	Echo (ping) request id=0x0100, seq=69/17664, ttl=128 (no response found!)

ip.addr eq 172.18.41.240

No.	Time	Source	Destination	Protocol	Length	Info
254	32.159494	172.26.21.83	172.18.41.240	ICMP	42	Echo (ping) request id=0x306e, seq=0/0, ttl=56 (no response found!)
255	32.165818	172.26.21.83	172.18.41.240	TCP	58	38605 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
256	32.170261	172.26.21.83	172.18.41.240	TCP	54	38605 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
257	32.172110	172.26.21.83	172.18.41.240	ICMP	54	Timestamp request id=0x8f9a, seq=0/0, ttl=45
258	34.161434	172.26.21.83	172.18.41.240	ICMP	54	Timestamp request id=0x6f07, seq=0/0, ttl=45
259	34.165366	172.26.21.83	172.18.41.240	TCP	54	38606 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
260	34.171398	172.26.21.83	172.18.41.240	TCP	58	38606 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
261	34.175455	172.26.21.83	172.18.41.240	ICMP	42	Echo (ping) request id=0x52ab, seq=0/0, ttl=54 (no response found!)

③ 测试结果不连通，但实际上是物理连通的，什么原因？

因为开启了防火墙，防火墙拦截并丢弃了请求报文，因此接收不到应答。实际上同一局域网内的不同主机物理上是连通的。

2．对目标主机进行 TCP 端口扫描

① 使用常规扫描方式

Nmap -sT 172.16.1.101

请将扫描检测结果截图写入实验报告，包括所有的端口及开放情况。

```
C:\WINDOWS\system32>Nmap -sT 172.18.41.240
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 00:03 ?D1ú±ê×?ê±??
Nmap scan report for 172.18.41.240
Host is up (0.029s latency).
All 1000 scanned ports on 172.18.41.240 are filtered

Nmap done: 1 IP address (1 host up) scanned in 72.91 seconds
```

② 使用 SYN 半扫描方式

Nmap -sS 172.16.1.101

请将扫描检测结果截图写入实验报告，包括所有的端口及开放情况。

```
C:\WINDOWS\system32>Nmap -sS 172.18.41.240
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-17 00:07 ?D1ú±ê×?ê±??
Nmap scan report for 172.18.41.240
Host is up (0.016s latency).
Not shown: 985 closed ports
PORT     STATE    SERVICE
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
443/tcp  open     https
445/tcp  filtered microsoft-ds
593/tcp  filtered http-rpc-epmap
902/tcp  open     iss-realsecure
912/tcp  open     apex-mesh
1433/tcp open     ms-sql-s
2383/tcp open     ms-olap4
3389/tcp open     ms-wbt-server
4444/tcp filtered krb524
5357/tcp open     wsdapi
5800/tcp filtered vnc-http
5900/tcp filtered vnc
6667/tcp filtered irc

Nmap done: 1 IP address (1 host up) scanned in 25.46 seconds
```

③ 比较上述两次扫描结果差异、扫描所花费的时间。并进行解释。

1) -sS ：半开放扫描（非 3 次握手的 tcp 扫描）

使用频率最高的扫描选项，又称为半开放扫描，它不打开一个完全的 TCP 连接，执行得很快，效率高（一个完整的 tcp 连接需要 3 次握手，而-sS 选项不需要 3 次握手）。

优点：Nmap 发送 SYN 包到远程主机，但是它不会产生任何会话，目标主机几乎不会把连接记入系统日志。（防止对方判断为扫描攻击），扫描速度快，效率高，在工作中使用频率最高

缺点：它需要 root/administrator 权限执行

2) sT：3 次握手方式 tcp 的扫描

默认的扫描模式，不同于 Tcp SYN 扫描，Tcp connect()扫描需要完成三次握手，并且要求调用系统的 connect()。

优点：不用 root 权限。普通用户也可以使用。

缺点：这种扫描很容易被检测到，在目标主机的日志中会记录大批的连接请求以及错误信息，由于它要完成 3 次握手，效率低，速度慢。

如上所述，可以看到 sT 花费的时间几乎是 sS 的三倍。

再查看结果，发现 sT 扫描到的端口都是 filtered 状态，该状态表示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。等于没有获得有效的端口信息。

而 sS 方法扫描到了全部 1000 个端口信息。

【实验体会】

通过这次实验熟悉了扫描工具 Nmap 的使用，主要包括主机发现命令和两种方式的 TCP 端口扫描。同时巩固了使用 Wireshark 捕获和分析数据包的知识。

明白了防火墙对于计算机网络安全的重要性，当关闭了防火墙时，计算机完全暴露在各种攻击之下，这是非常危险的！