西南科技大学

网络攻击与防御
实验报告

实验题目：网络扫描与信息收集

学生姓名：

学生学号：

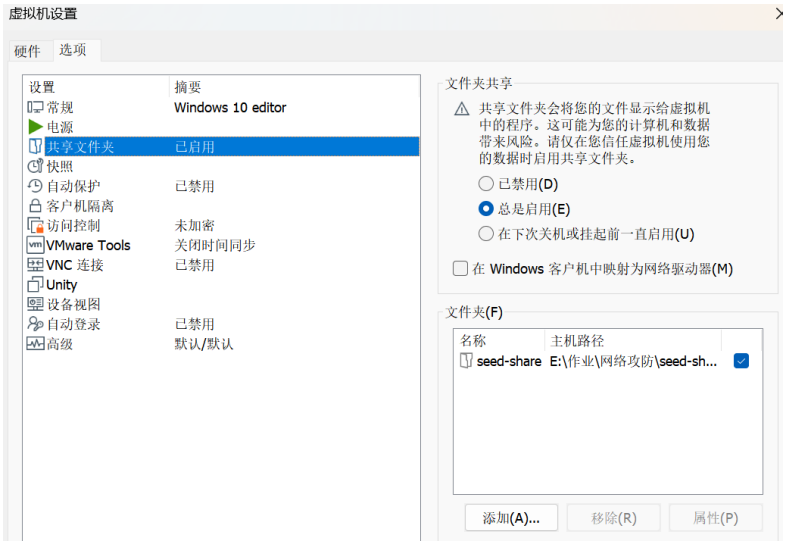
一、 实验作业题目

1. 安装环境，配置共享文件夹
2. 完成 windows 相关系统命令
3. 熟悉 net 命令，任选两个 net 命令，演示 net 命令执行结果。
4. 利用 net 命令，借助空连接列举目标主机上的用户和共享，得到用户列表，并写出攻击步骤。
5. 使用 NMAP 扫描目标主机端口、漏洞、操作系统
6. 使用 Nessus 扫描目标主机端口、漏洞、操作系统

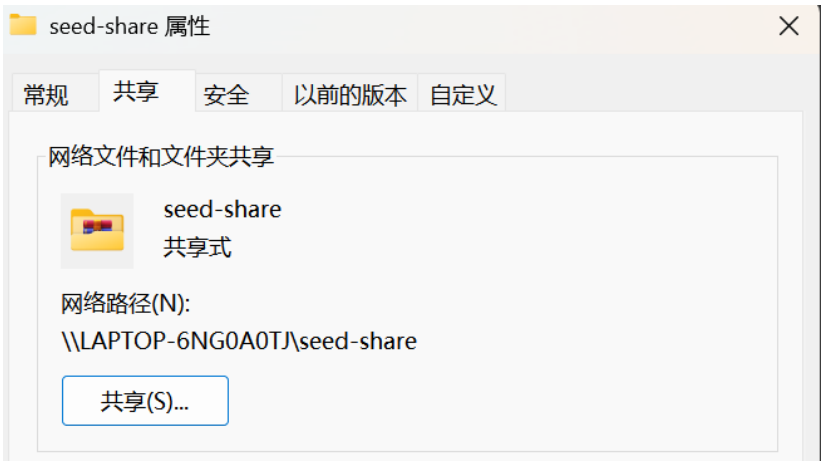
二、 实验思路

安装环境

按照实验环境要求下载文件后进行解压缩，win10 虚拟机打开共享文件夹设置，如下图：



选择要共享的文件夹，右键打开属性->共享，选择共享，然后重启即可设置成功。



完成 windows 相关系统命令

命令 ipconfig：显示和管理网络配置信息，当前 ip 为 61.139.2.130

命令 ping -a：将目标主机的 IP 地址解析为主机名，由下图可知 61.139.2.130 的主机名为 DESKTOP-ESPCPAP.localdomain

```
C:\Users\26032>ping -a 61.139.2.130

正在 Ping DESKTOP-ESPCPAP.localdomain [61.139.2.130] 具有 32 字节的数据:
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128

61.139.2.130 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 = 0ms，最长 = 0ms，平均 = 0ms
```

命令 ping -n 向 61.139.2.130 发送 10 个数据包的过程当中，返回了 10 个数据包。

```
C:\Users\26032>ping -n 10 61.139.2.130

正在 Ping 61.139.2.130 具有 32 字节的数据:
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128
来自 61.139.2.130 的回复: 字节=32 时间<1ms TTL=128

61.139.2.130 的 Ping 统计信息:
    数据包: 已发送 = 10，已接收 = 10，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms，最长 = 0ms，平均 = 0ms
```

命令 ping -l 用于指定发送的 ICMP 数据包的大小

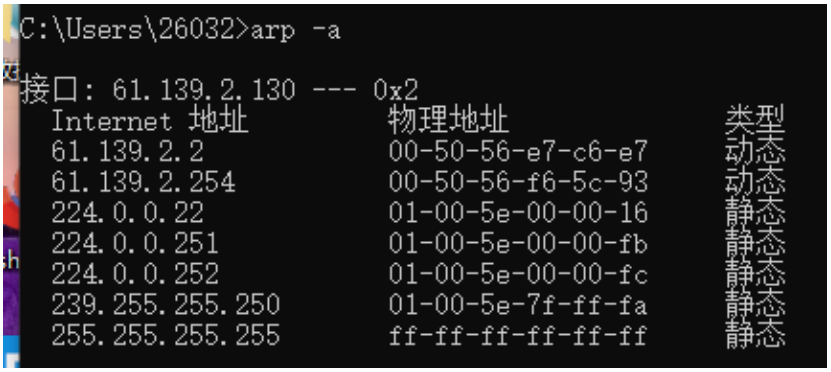
课件上说在我们自定义数据包最大只能发送 65527byte。危险性？为什么?我的虚拟机上操作最多只能 65500byte？

实际上最大就是 65500byte，因为 Windows 系列的系统都有一个**安全漏洞**（也许还包括其他系统）就是当向对方一次发送的数据包大于或等于 65532 时，对方就很有可能当机，所以微软公司为了解决这一安全漏洞于是限制了 ping 的数据包大小。

这个参数配合什么参数以后，会造成一个攻击性命令？

当-l 参数与-t 参数（持续不断地向目标主机发送 ICMP 数据包，直到用户手动停止）配合使用时形成类似死亡之 Ping。

命令 arp -a：显示当前系统的 ARP 缓存表内容，ARP 缓存表记录了 IP 地址和 MAC 地址的映射关系

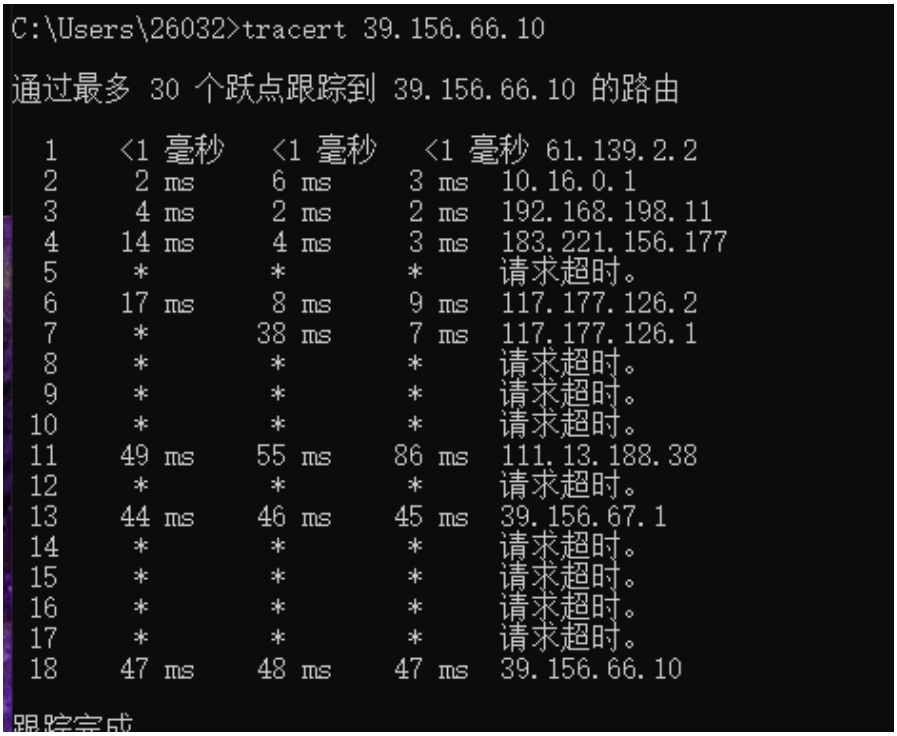


命令 netstat -ano：netstat 显示网络连接、路由表和网络接口等网络相关信息-，a 显示所有活动的 TCP 连接以及监听的 TCP 和 UDP 端口。-n 以数字形式显示地址和端口号，-o 显示每个连接或监听端口所对应的进程 ID（PID）

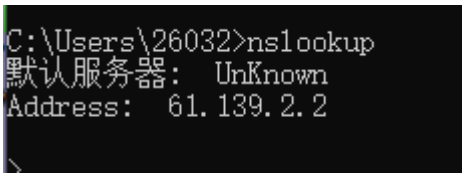
```
C:\Users\26032>netstat -ano

活动连接

  协议  本地地址          外部地址        状态           PID
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING       468
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING       3644
  TCP    61.139.2.130:49880     106.127.135.72:80      ESTABLISHED     3644
  TCP    61.139.2.130:49980     42.202.165.228:80      ESTABLISHED     3644
  TCP    61.139.2.130:50018     182.242.215.76:80      ESTABLISHED     3644
  TCP    61.139.2.130:50019     220.185.175.60:80      ESTABLISHED     3644
  TCP    61.139.2.130:50070     20.197.71.89:443       ESTABLISHED     2988
  TCP    61.139.2.130:50227     58.216.4.187:80        ESTABLISHED     3644
  TCP    61.139.2.130:50237     104.121.144.200:443    ESTABLISHED     9836
  TCP    61.139.2.130:50331     220.185.175.60:80      ESTABLISHED     3644
  TCP    61.139.2.130:50337     103.228.12.65:443      CLOSE_WAIT      9560
  TCP    61.139.2.130:50338     103.228.12.65:443      CLOSE_WAIT      9560
  TCP    61.139.2.130:50339     52.98.92.2:443         ESTABLISHED     9560
  TCP    61.139.2.130:50342     182.140.210.178:80     ESTABLISHED     4560
  TCP    61.139.2.130:50363     113.142.77.43:80       ESTABLISHED     3644
  TCP    61.139.2.130:50366     40.126.35.151:443      ESTABLISHED     9248
  TCP    61.139.2.130:50373     20.42.73.25:443        ESTABLISHED     3088
  TCP    61.139.2.130:50374     113.142.77.43:80       ESTABLISHED     3644
  TCP    61.139.2.130:50376     13.107.21.239:443      ESTABLISHED     9124
  TCP    61.139.2.130:50377     204.79.197.239:443     ESTABLISHED     9124
  TCP    61.139.2.130:50379     40.79.150.121:443      SYN_SENT        3604
  TCP    [::]:7680              [::]:0                 LISTENING       3644
  UDP    0.0.0.0:123            *:*                                    8764
```

命令 tracert：确定数据包从源主机到目标主机所经过的所有路由器的路径，ping -r 主要目的是在发送 CMP 回显请求数据包时，记录数据包经过的路由信息。

```
C:\Users\26032>ping -r 1 baidu.com

正在 Ping baidu.com [39.156.66.10] 具有 32 字节的数据:
来自 39.156.66.10 的回复: 字节=32 时间=44ms TTL=128
来自 39.156.66.10 的回复: 字节=32 时间=45ms TTL=128
来自 39.156.66.10 的回复: 字节=32 时间=54ms TTL=128
来自 39.156.66.10 的回复: 字节=32 时间=51ms TTL=128

39.156.66.10 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 = 44ms, 最长 = 54ms, 平均 = 48ms
```



命令 nslookup：将域名解析为对应的 IP 地址，或者反向查询 IP 地址对应的域名，下图表示即当前 ip 无域名



命令 at：任务计划工具，可以查看计划任务。系统提示 at 命令已弃用。请改用 schtasks.exe，即 at 命令在较新的 Windows 系统中不再推荐使用，需用 schtasks.exe 替代。

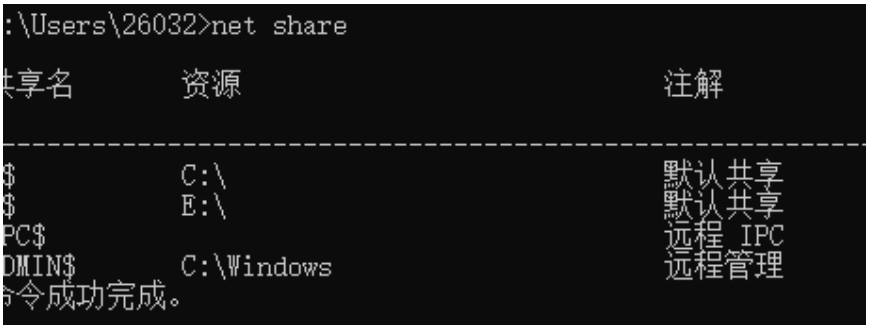
```
C:\Users\26032>at
AT 命令已弃用。请改用 schtasks.exe。

不支持该请求。

C:\Users\26032>schtasks.exe

文件夹: \
任务名                                  下次运行时间           模式
====================================== ====================== ===============
OneDrive Reporting Task-S-1-5-21-4245667 2025/2/28 19:35:21     就绪
OneDrive Standalone Update Task-S-1-5-21 2025/2/28 18:24:45     就绪

文件夹: \Microsoft
任务名                                  下次运行时间           模式
====================================== ====================== ===============
信息: 目前在你的访问级别上不存在任何可用的计划任务。

文件夹: \Microsoft\OneCore
任务名                                  下次运行时间           模式
====================================== ====================== ===============
信息: 目前在你的访问级别上不存在任何可用的计划任务。

文件夹: \Microsoft\Windows
任务名                                  下次运行时间           模式
====================================== ====================== ===============
信息: 目前在你的访问级别上不存在任何可用的计划任务。

文件夹: \Microsoft\Windows\.NET Framework
任务名                                  下次运行时间           模式
====================================== ====================== ===============
.NET Framework NGEN v4.0.30319          N/A                    就绪
.NET Framework NGEN v4.0.30319 64       N/A                    就绪
.NET Framework NGEN v4.0.30319 64 Critic N/A                   已禁用
.NET Framework NGEN v4.0.30319 Critical  N/A                   已禁用

文件夹: \Microsoft\Windows\Active Directory Rights Management Services Client
任务名                                  下次运行时间           模式
====================================== ====================== ===============
AD RMS Rights Policy Template Management N/A                   已禁用
AD RMS Rights Policy Template Management N/A                   就绪

文件夹: \Microsoft\Windows\AppID
任务名                                  下次运行时间           模式
====================================== ====================== ===============
```

熟悉 net 命令，任选两个 net 命令，演示 net 命令执行结果。

net user：查看本地用户账户列表，列出当前计算机上的所有用户账户，当前计算机上有 26032、Administrator、DefaultAccount、Guest 和 WDAGUtilityAccount 用户账号。

```
C:\Users\26032>net user

\\DESKTOP-ESPCPAP 的用户帐户

-------------------------------------------------------------------------------
26032                    Administrator            DefaultAccount
Guest                    WDAGUtilityAccount
命令成功完成。
```

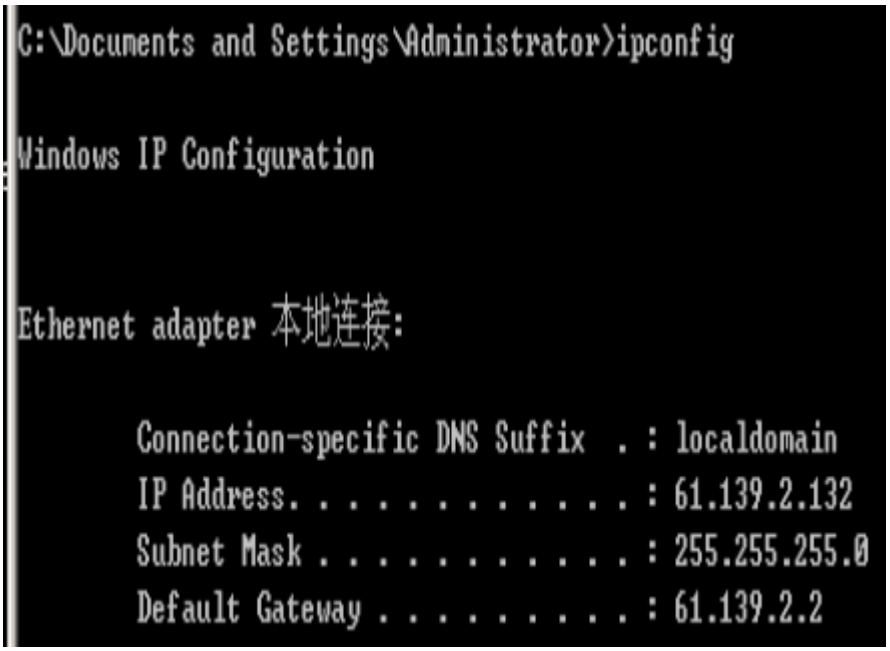
net share：查看当前计算机的共享资源列表。远程 IPC 是为了让进程间通信使用的命名管道，是默认开启的共享。ADMIN$对应资源为 C:\Windows，管理共享，用于管理员远程访问 Windows 目录。



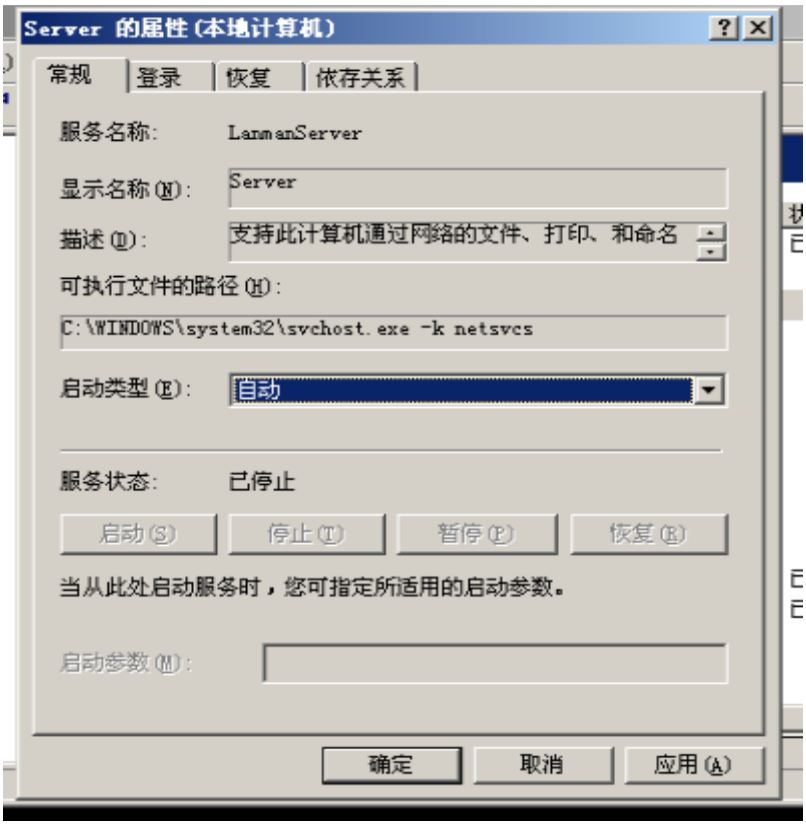
利用 net 命令，借助空连接列举目标主机上的用户和共享，得到用户列表，并写出攻击步骤。

我们通过文章 Windows 空连接（主机 IP、用户名、密码）_允许空密码链接 net use-CSDN 博客，了解空连接是不使用用户名和密码的 IPC 连接，默认发生在 win NT/2000/XP/7 环境下，所以我们选择实验环境为 win xp 作为目标主机进行空连接。

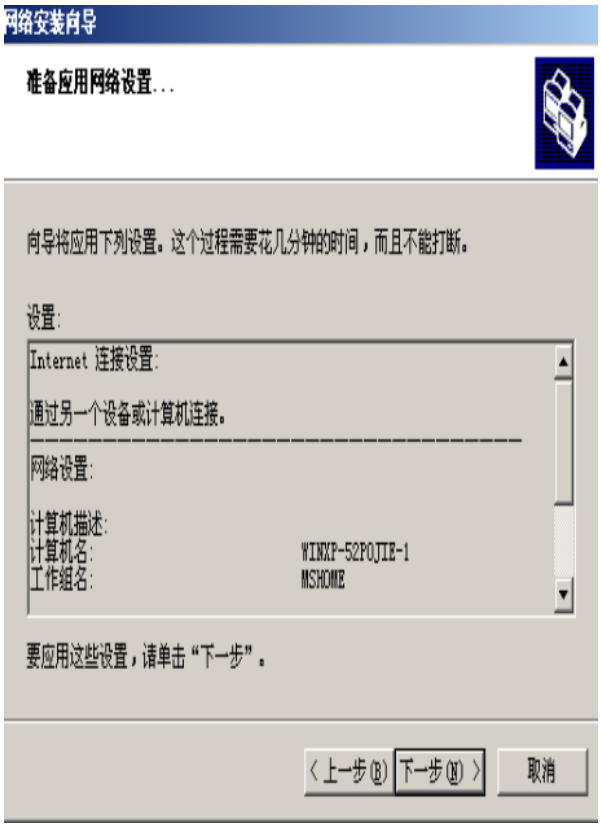
Win xp 靶机 ip：61.139.2.132，通过 ipconfig 查看。

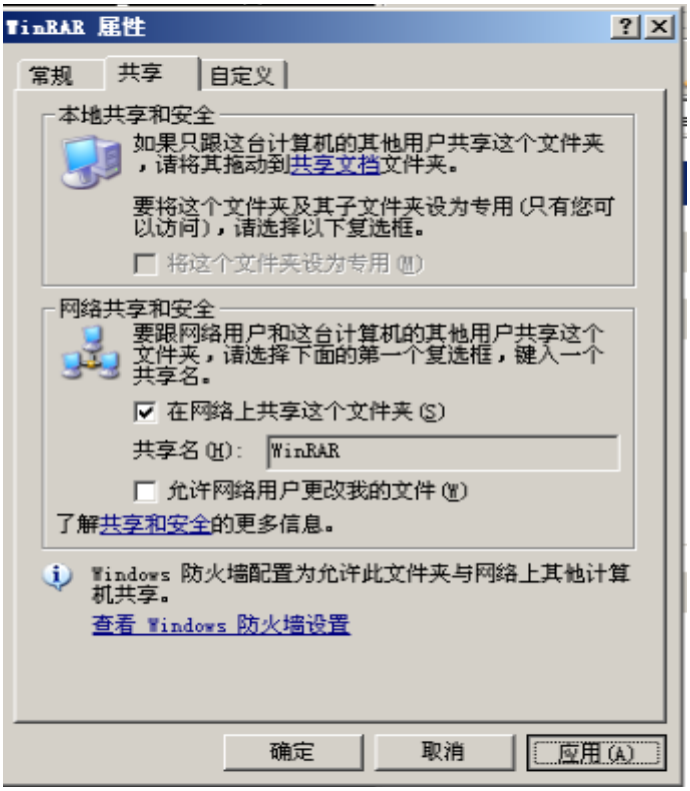


打开 ipc$共享，还需要打开 LanmanSever 服务（即 Sever 服务，在 services.msc 中打开），它提供了 RPC 支持、文件、打印以及命名管道共享， ipc$依赖于此服务，没有它主机将无法响应发起方的连接请求，不过没有它仍可发起 ipc$ 连接。

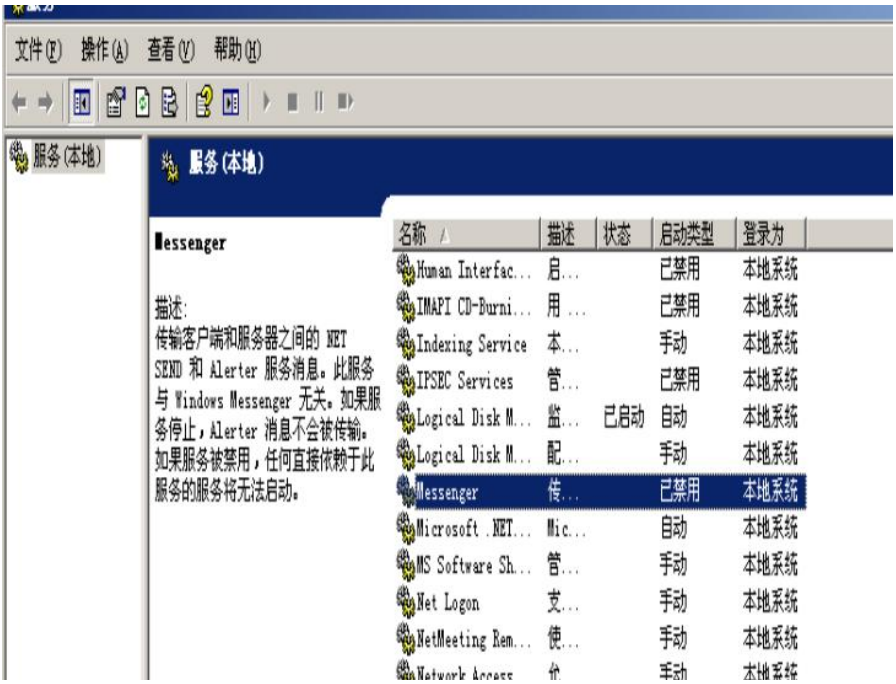


然后设置共享文件夹（选择要共享的文件夹进行网络安装向导后设置共享）





启用 Messenger 服务



打开命令提示符，输入 services.msc 并回车，打开服务管理窗口。在服务列表中查找 “Messenger” 服务，默认状态通常是禁用的。右键点击该服务，选择 “属性”，将 “启动类型” 改为 “手动” 或 “自动”，然后点击 “确定”。再次右键点击 “Messenger” 服务，选择 “启动”。

设置完后利用另一台机子进行空连接：

先得到目标主机的 NetBIOS 用户名列表： nbtstat -A 61.139.2.132

进行空连接：net use \\61.139.2.132\IPC$ "" /user:""

虽然空会话是非信任会话，没有多少权限，但我们可以通过空连接获得用户列表、进行口令猜解。

查看共享资源：在靶机上使用 net view 查看共享资源，net view \\61.139.2.132

查看用户：在靶机上使用 net user 查看用户，net user \\61.139.2.132

知道了用户名，可以尝试进行空密码连接

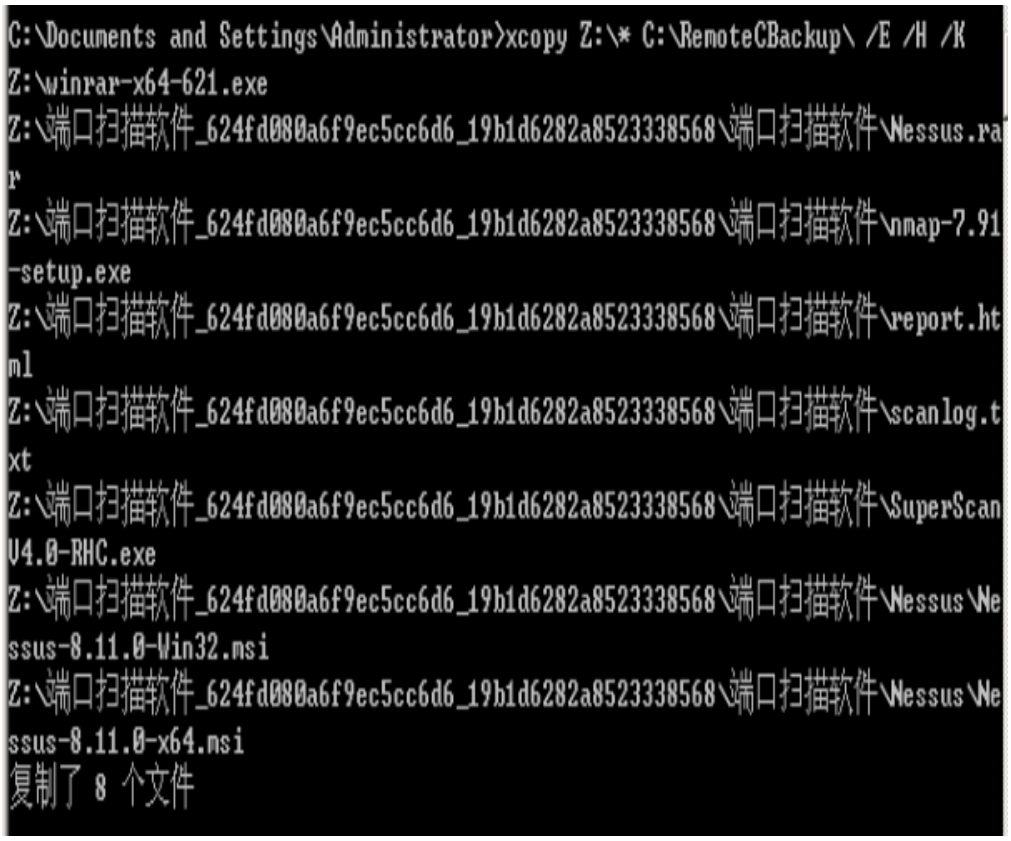
先断开连接：net use * /delete

发现 Administrator 能够连接上，建立信任连接。

建立远程连接：使用 net use 命令将目标主机的 C 盘共享挂载到本地的 Z 盘，net use Z: \\61.139.2.132\C$ "" /user:"Administrator"

镜像远程 C 盘：xcopy Z:* C:\RemoteCBackup\ /E /H /K。

Z:*：表示远程 C 盘（挂载到 Z 盘）的所有内容。



C:\RemoteCBackup\：本地用于存放备份内容的目录（需提前创建）。/E：复制目录和子目录，包括空目录。/H：复制隐藏和系统文件。/K：复制文件时保留文件的属性。

复制本地 C 盘的 test.txt 到目标主机的 C 盘目录下：copy C:\test.txt Z:\。不知道为什么复制不了，直接网络映射挂载一下然后再 copy

使用 net send 命令发送消息：net send 61.139.2.132 "这是一条测试消息"

断开连接：net use * /delete

使用 NMAP 扫描目标主机端口、漏洞、操作系统

靶机 ip 为 61.139.2.129

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:28:60:a7
          inet addr:61.139.2.129  Bcast:61.139.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe28:60a7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:40 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4305 (4.2 KB)  TX bytes:7220 (7.0 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21437 (20.9 KB)  TX bytes:21437 (20.9 KB)
```

扫描端口：

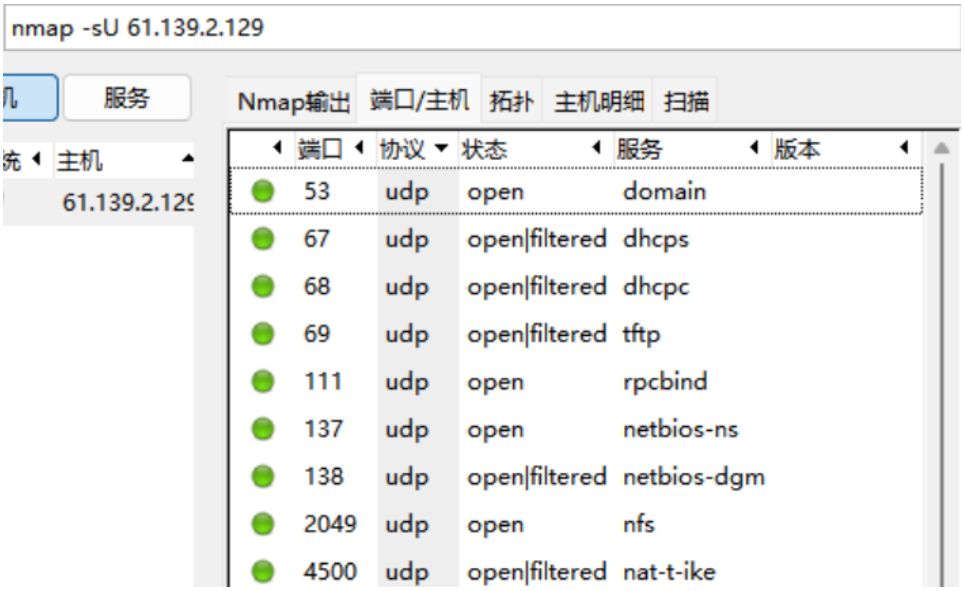
-P 扫描开放端口

```
nmap -Pn 61.139.2.129
Starting Nmap 7.91 ( https://nmap.org ) at
2025-03-02 18:14 ?D1ú±ê×?ê±??
Host discovery disabled (-Pn). All addresses will
be marked 'up' and scan times will be slower.
Nmap scan report for 61.139.2.129
Host is up (0.00075s latency).
Not shown: 978 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:28:60:A7 (VMware)

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

-sT 为全连接扫描开放的 TCP 端口，扫描得到 21、22、23、25、80、110、111、139、445 等端口开放的，对应的的服务也列出来了

-sS 使用 SYN 半开放扫描开放的 TCP 端口，不会构成完整的 TCP 三次握手，与上面全连接扫描对比可以知道，SYN 扫描没有全扫描那么精确，比如该例子就漏掉了开放端口 110

-sU 为扫描开放的 UDP 端口



扫描漏洞：

nmap -sV --script vulners --script-args mincvss=5.0 61.139.2.129，调用 vulners 脚本查询漏洞数据库，匹配目标服务可能存在的 CVSS 评分大于等于 5.0 的漏洞

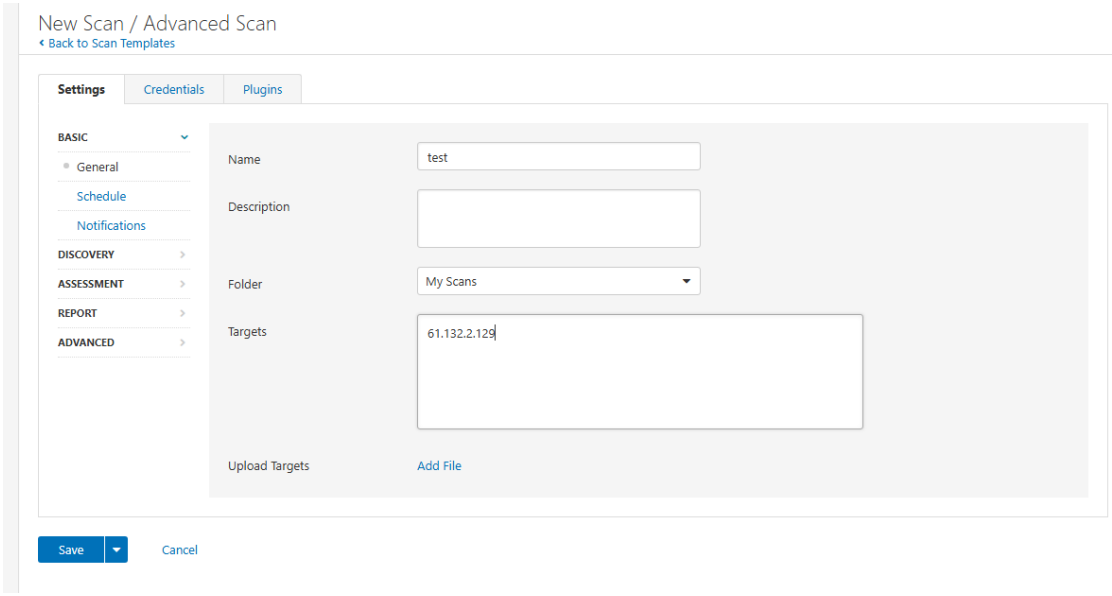
```
Starting Nmap 7.91 ( https://nmap.org ) at 2025-03-02 19:01 ?D1ú±ê×?ê±??
Nmap scan report for 61.139.2.129
Host is up (0.00s latency).
Not shown: 978 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
| vulners:
|   vsftpd 2.3.4:
|       PACKETSTORM:162145      10.0    https://vulners.com/packetstorm/PACKETSTORM:162145      *EXPLOIT*
|       EDB-ID:49757    9.8     https://vulners.com/exploitdb/EDB-ID:49757      *EXPLOIT*
|       CVE-2011-2523   9.8     https://vulners.com/cve/CVE-2011-2523
|_      1337DAY-ID-36095        9.8     https://vulners.com/zdt/1337DAY-ID-36095        *EXPLOIT*
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:4.7p1:
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A    10.0    https://vulners.com/githubexploit/2C119FFA-ECE0-5E14-A4A4-354A2C38071A
*EXPLOIT*
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       CVE-2016-1908   9.8     https://vulners.com/cve/CVE-2016-1908
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23
*EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623
*EXPLOIT*
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC
*EXPLOIT*
|       887EB570-27D3-11EE-ADBA-C80AA9043978    9.8     https://vulners.com/freebsd/887EB570-27D3-11EE-ADBA-C80AA9043978
|       5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8     https://vulners.com/githubexploit/5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A
*EXPLOIT*
|       33D623F7-98E0-5F75-80FA-81AA666D1340    9.8     https://vulners.com/githubexploit/33D623F7-98E0-5F75-80FA-81AA666D1340
*EXPLOIT*
|       0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4B9E2D1B587
*EXPLOIT*
|       95499236-C9FE-56A6-9D7D-E943A24B633A    8.9     https://vulners.com/githubexploit/95499236-C9FE-56A6-9D7D-E943A24B633A
*EXPLOIT*
|       CVE-2015-5600   8.5     https://vulners.com/cve/CVE-2015-5600
|       5B74A5BC-348F-11E5-BA05-C80AA9043978    8.5     https://vulners.com/freebsd/5B74A5BC-348F-11E5-BA05-C80AA9043978
|       PACKETSTORM:179290      8.1     https://vulners.com/packetstorm/PACKETSTORM:179290      *EXPLOIT*
|       FB2E9ED1-43D7-585C-A197-0D6628B20134    8.1     https://vulners.com/githubexploit/FB2E9ED1-43D7-585C-A197-0D6628B20134
*EXPLOIT*
|       FA3992CE-9C4C-5350-8134-177126E0BD3F    8.1     https://vulners.com/githubexploit/FA3992CE-9C4C-5350-8134-177126E0BD3F
*EXPLOIT*
|       F8981437-1287-5B69-93F1-657DFB1DCE59    8.1     https://vulners.com/githubexploit/F8981437-1287-5B69-93F1-657DFB1DCE59
*EXPLOIT*
```

扫描操作系统：

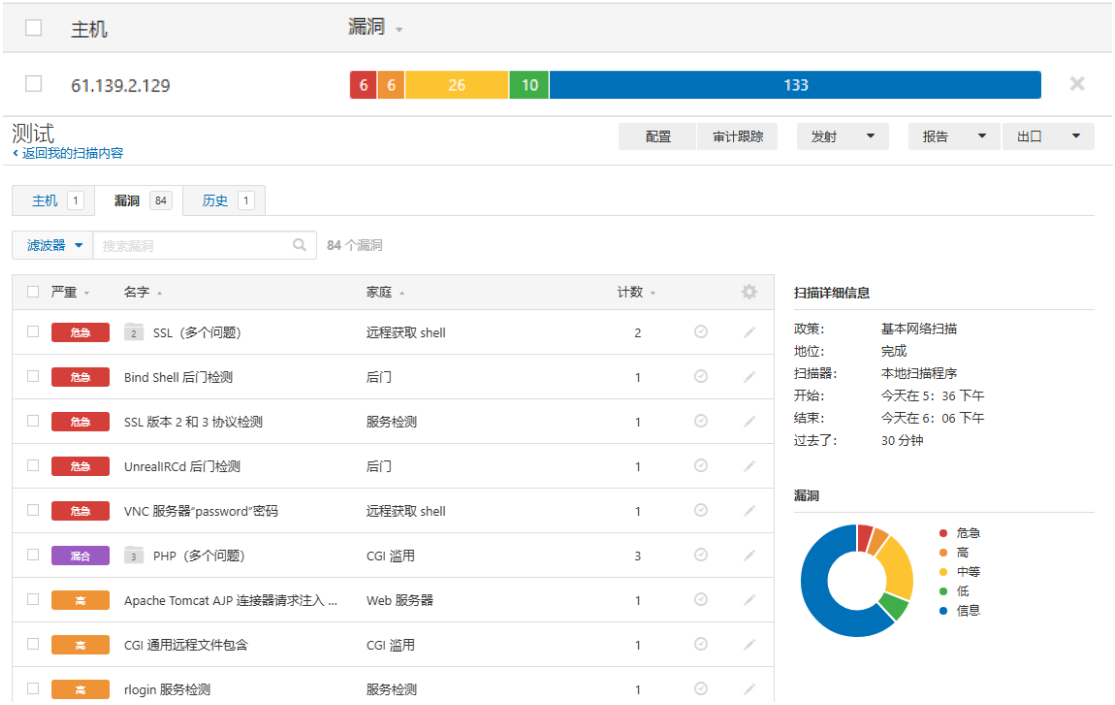
nmap -O 61.139.2.129，推测是 Linux 2.6.X 内核，具体版本范围在 Linux 2.6.9 - 2.6.33 之间，OS CPE（通用平台枚举）为 cpe:/o:linux:linux_kernel:2.6

```
8180/tcp open  unknown
MAC Address: 00:0C:29:28:60:A7 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 3.51 seconds
```

使用 Nessus 扫描目标主机端口、漏洞、操作系统



扫描靶机



三、 实验分析及总结（写自己在实践过程中遇到的问题及解决方法；本次实验体会、收获）

问题：Nessus 安装一直显示初始化中

解决办法：使用命令行进行安装即可。参考文章 Nessus Scanners

returning with blank results after recent Plugins update

问题：nessus 没有 scan 按钮

解决办法：升级专业版，Win10 安装 nessus8.10 系列_nessus8.10.1 下载安装-CSDN 博客

实验思考：一台接入 Internet 的主机出现无法访问的情况，怎样诊断原因？

物理连接问题：网线是否插好、是否连接 WIFI、网卡是否正常

IP 配置错误：DHCP 自动获取 IP，检查是否正确，如果 IP 地址是 169.254 开头的，说明 DHCP 获取失败，可能是 DHCP 服务器问题。如果是手动配置的检查输入是否有误。使用 ipconfig(windows)/ifconfig(linux)进行检查。

本地网络问题：使用 ping 命令测试网关，网关不通则可能是路由器故障或者网络适配器问题，如果正常，则说明问题可能在外部网络。

DNS 问题：nslookup 解析域名，无法解析则是 DNS 服务器有问题或者本地 DNS 配置有误，尝试更换公共 DNS。

测试外网：ping 公网 ip，若通，则可能是防火墙或安全软件阻止，检查防火墙配置，确保 80、443 等相关端口开放。若不通，使用 traceroute(linux)/tracert(windows)来跟踪数据包路径，看看在哪一跳中断了。这样可以确定问题发生在哪个节点，比如是 ISP 的问

题，还是中间某个路由节点的问题。

检查应用层：是否配置代理服务器，尝试关闭。

实验思考：假如已经通过缓冲区溢出攻击获得一台主机的 shell，有哪些方法可以将木马程序运行起来？

缓冲区溢出攻击是指攻击者输入超出缓冲区边界的恶意数据，导致数据溢出到相邻内存区域，破坏程序的堆栈，使程序转而执行恶意指令。

可以使用 chmod 运行木马，通过+x 参数给木马执行权限。

也可以使用 nohub 运行木马，加入&放入后台运行。

实验思考：如何能够做到扫描过程中不被对方的防火墙发现？

降低扫描的速度和频率，使扫描行为更像正常的网络访问，避免因扫描速度过快产生的异常流量而被防火墙检测到。

使用 FIN 扫描、SYN 扫描等扫描方式，这些方式相较于传统的全连接扫描更隐蔽，因为它不完成完整的 TCP 三次握手过程。

利用代理服务器作为中间节点来进行扫描，将真实的扫描源 IP 地址隐藏在代理服务器之后，增加被发现的难度。

构造扫描数据包的特征与正常网络流量的数据包特征相似，将扫描数据包的源 IP 地址、端口号等信息设置为常见的、合法的地址和端口，模仿正常的网络连接行为。在扫描过程中，对数据包的发送时间、数据内容等添加一定的随机化因素，避免出现规律性的扫描行为

模式。

实验思考：如何欺骗对方的扫描活动，使之得不到正确的结果？

在网络中设置蜜罐，模拟各种可能被攻击的服务和系统，如 FTP、SSH 等服务，故意暴露一些漏洞和弱点，吸引攻击者进行扫描和攻击，同时记录攻击者的行为和特征。当对方进行扫描时，会将蜜罐误认为是真正的目标系统，从而获取到大量虚假的信息。

通过配置网络设备和软件，向外部呈现一个与实际网络拓扑不同的虚假拓扑结构，让扫描者获取到错误的网络架构信息。

在网络中发送大量与扫描行为相似的干扰数据包，使扫描工具难以分辨真实的目标和干扰信息。

研究常见扫描工具的工作原理和漏洞，针对其弱点进行欺骗。某些扫描工具在处理特定类型的网络协议或数据包格式时存在缺陷，可通过构造特殊的数据包使扫描工具产生错误的结果或崩溃。

体会及收获：

在本次实验深入探究了网络故障排查、系统命令运用及安全攻防场景。面对 Nessus 安装初始化及无扫描按钮的难题，我通过命令行安装与升级专业版的方式成功解决，这极大提升了我的问题解决能力。

实验过程让我充分认识到网络安全的重要性与复杂性。从诊断主机无法访问的多元方法，到 net 命令的灵活运用，再到理解缓冲区溢

出攻击及木马运行方式，每一步都加深了我对网络环境的把控。我意识到，网络安全不仅关乎技术知识，更需严谨的操作规范。在合法合规的前提下，掌握这些技术能够助力网络管理与安全防护。此次实验不仅丰富了我的技术储备，更培养了我在面对复杂网络问题时的分析与解决思维，为今后深入学习网络安全领域奠定了坚实基础 。