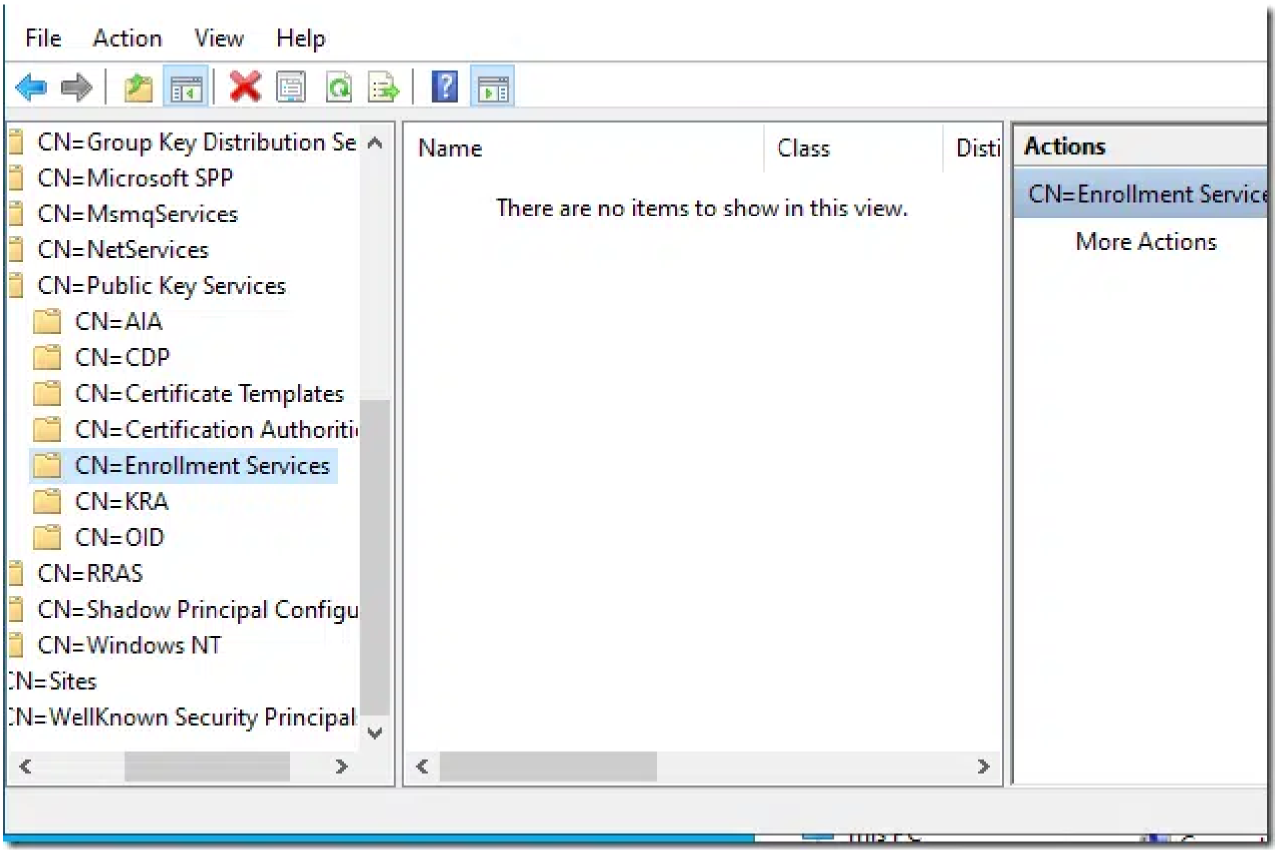
Escalating from child domain’s admins to enterprise admins in 5 minutes by abusing AD CS, a follow up

pkisolutions.com/escalating-from-child-domains-admins-to-enterprise-admins-in-5-minutes-by-abusing-ad-cs-a-follow-up

Vadims Podāns

January 24, 2024



Hello everyone, long time no see. I’m still extremely busy on my main job stuff, specifically PKI Spotlight commercial product development, so my blogging has slowed, and I’m here again!

Prologue

Disclaimer: This post contains steps and information that can lead to legal issues with your employer and lawsuits if you execute them in a production environment.

Today I’ll talk about how you can escalate from Domain Admins in child domains to Enterprise Admins in your Active Directory forest where no PKI is deployed yet. This post is based on a great article From DA to EA with ESC5 by Andy Robbins. I strongly recommend you read that article first, I enjoyed it extremely and the vast majority of credit from this post go to Andy!

In short, any writable domain controller in the forest (literally, any) can write into Configuration Naming Context AD partition which is replicated across all domain controllers in the forest. As long as you can escalate yourself to a local system account (hi, psexec!) on a DC (assuming you are one of admins on a writable DC, no matter what domain it is located in), then you can utilize default “System = Full Control” permissions and write whatever you want in these containers.

Coincidentally, Microsoft PKI stores critical and sensitive information to Configuration Naming Context. Not actual secrets, but it is a trust store for the entire Active Directory forest. Enterprise CAs are registered there as enrollment services. Enterprise CAs store a list of certificate templates they support and it contains a store of unconditionally trusted private PKIs. If you can write into Configuration Naming Context, you can install your fake root CA certificates and do really mad things. Andy’s blog post I referenced contains all the detailed information about what and how. This blog post is more like a “follow up”, because the referenced blog post ends with:

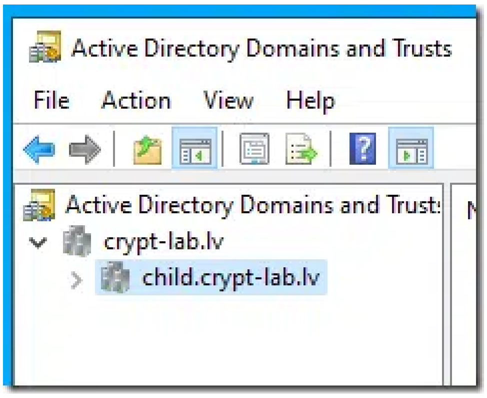
> The next question I want answered: What if ADCS isn’t already deployed? Can we bootstrap the necessary LDAP objects and issuing CA to turn DA into EA if we have, for example, full control of the “Public Key Services” container but there are no CAs?

And the answer is:
Actually, don’t watch, just read.

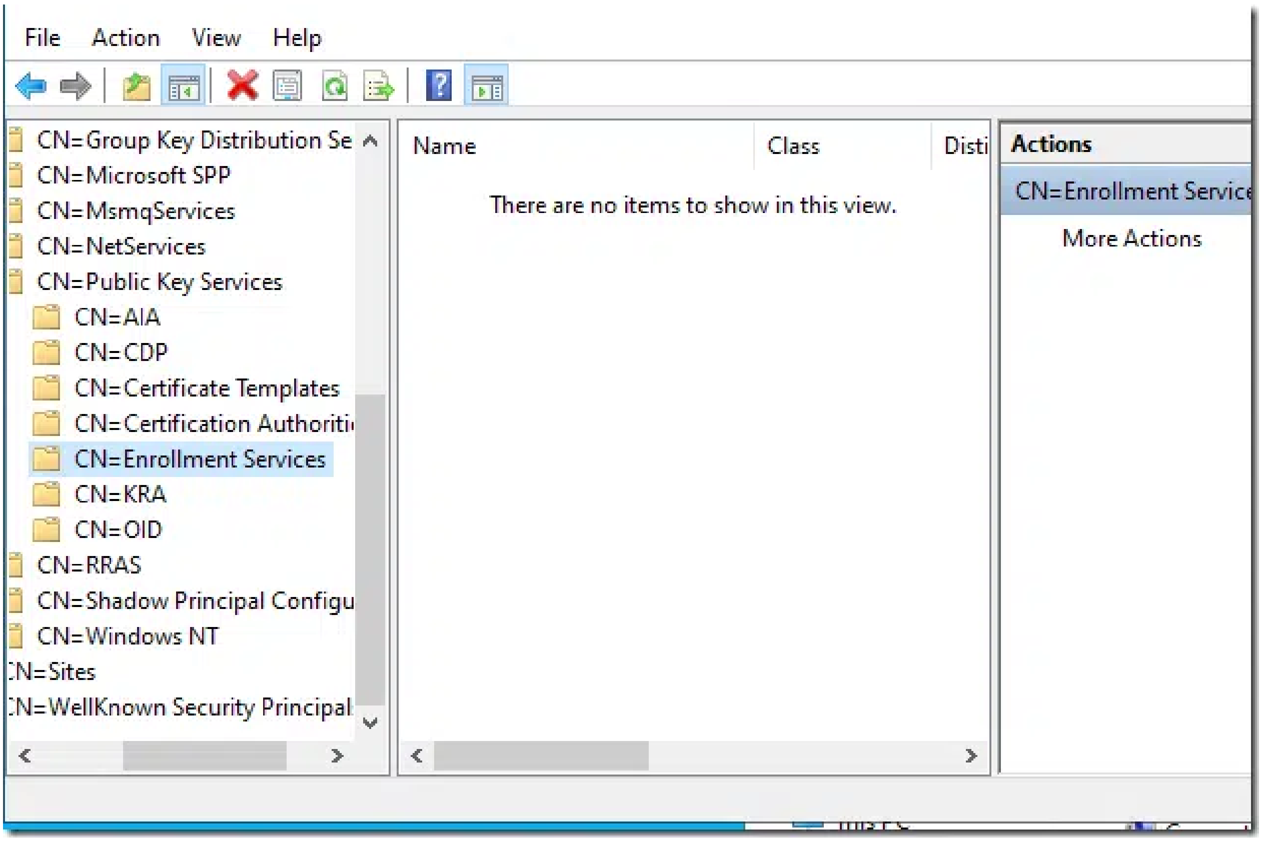
Pre-checks

Let’s continue where Andy ended: What if AD CS isn’t already deployed? If no one did this, you will deploy this yourself and be able to do really, awful, nasty things. My vanilla brand environment consists of two domains:

1. crypt-lab.lv – forest root domain
2. child.crypt-lab.lv – child domain


Active Directory Domains and Trusts
File
Action
View
Help
Active Directory Domains and Trust:
crypt-lab.lv
child.crypt-lab.lv

All steps provided in this blog post are executed on a writable DC in child domain only. There is nothing to do in forest root domain. Since it is a vanilla brand install, no PKI is deployed yet:



There are no items to show in this view.

No Enterprise CAs yet, no certificate templates, no nothing.

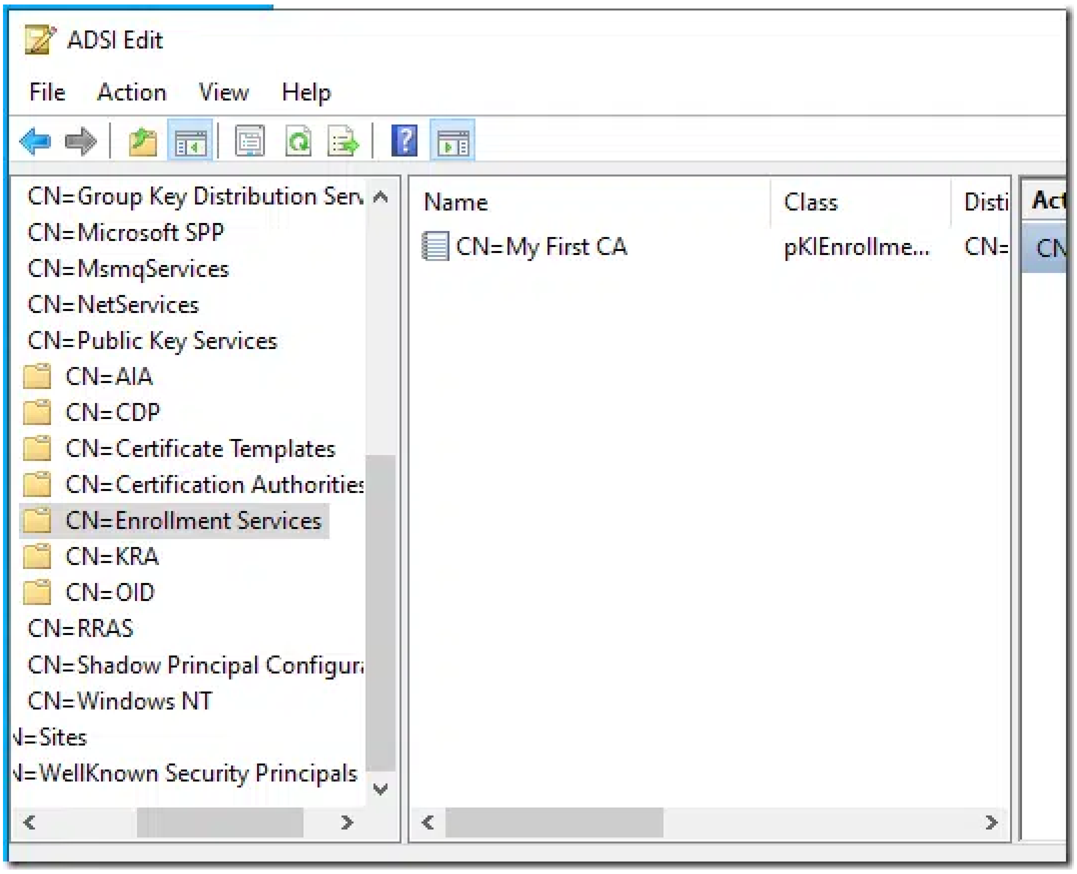
Installing Enterprise CA in child domain without Enterprise Admins permissions

Let’s add some. First, you need to download a psexec tool from Sysinternals and execute a PowerShell under local system account by calling `psexec /s /i powershell`. Then in elevated PowerShell console install AD CS components and your first Enterprise CA with default settings:

5:30 PM

```
PS C:\> Install-WindowsFeature AD-Certificate, ADCS-Cert-Authority -
IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {Active Directory Certificate Services,
Ce...


PS C:\> Install-AdcsCertificationAuthority -CAType EnterpriseRootCA -CACommonName
"My First CA"

Confirm
Are you sure you want to perform this action?
Performing the operation "Install-AdcsCertificationAuthority" on target "LAB-
CHILD-DC1".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
"Y"):

ErrorId ErrorString
------- -----------
      0


PS C:\>
```

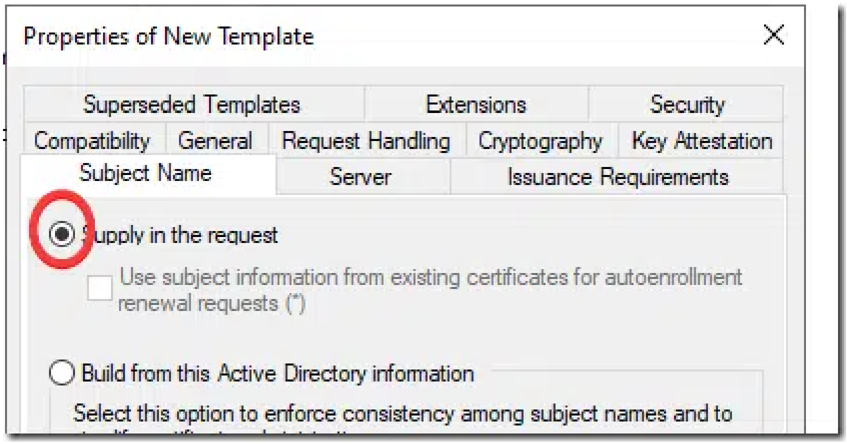
Do not close elevated PowerShell console and let’s validate if everything is up:



Name	Class	Disti
CN=My First CA	pKIEnrollme...	CN=

We see that CA is successfully registered as enrollment service. It is now trusted by **any** AD forest member, user, computer, domain controller. By default, no additional steps are required. Next step is to prepare a couple of templates. First, from elevated PowerShell prompt, type `certtmpl.msc` to open Certificate Templates MMC snap-in. We need to create a copy of a default User template and configure two tabs:

1. Subject tab set radio button to “Supply in request”:



● Supply in the request

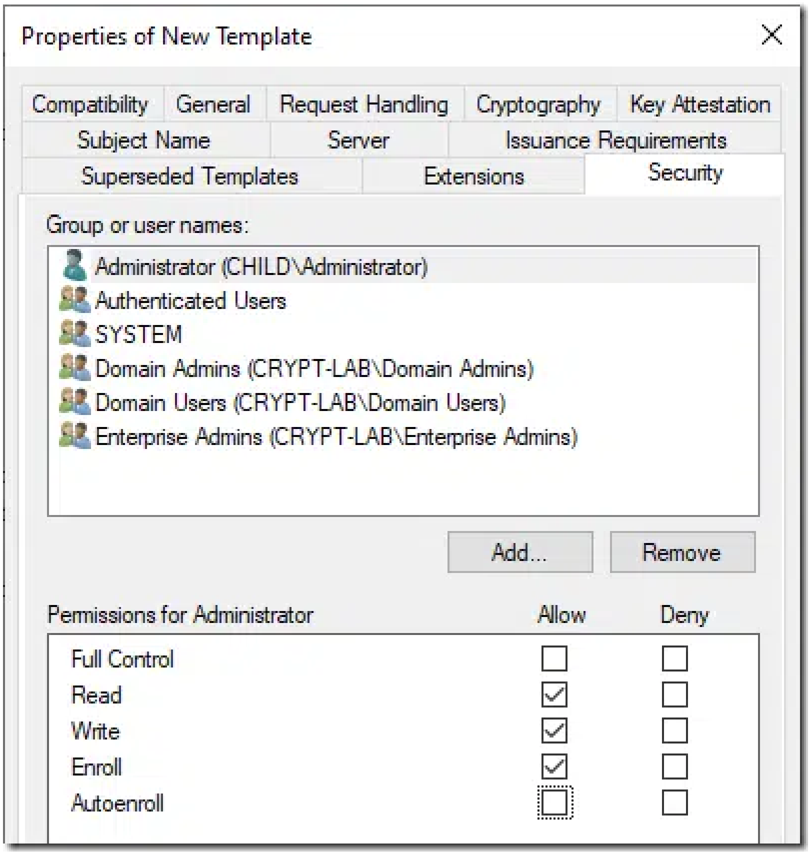
☐ Use subject information from existing certificates for autoenrollment renewal requests (*)

○ Build from this Active Directory information

Select this option to enforce consistency among subject names and to

2. Security tab

grant yourself or your group Read, Write and Enroll permissions:



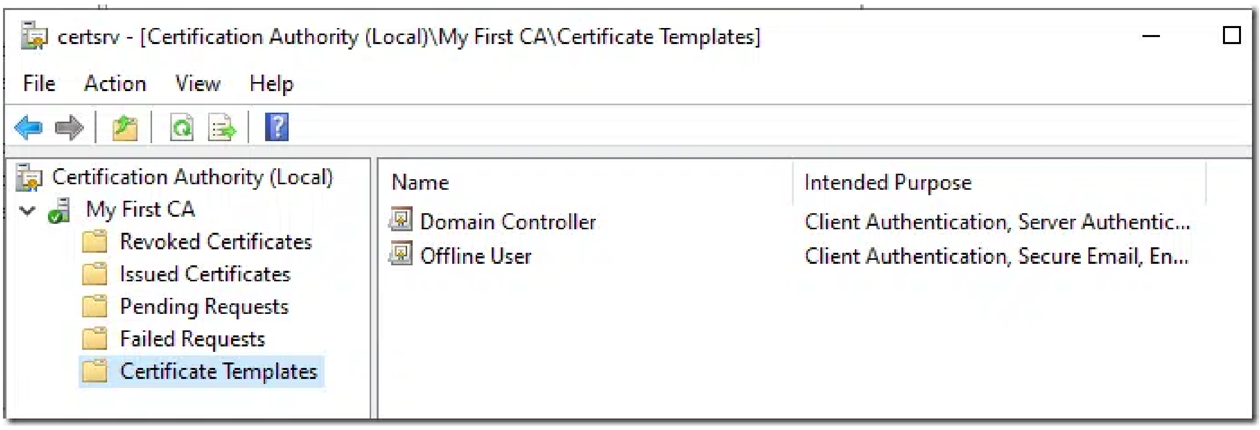
Group or user names:

Administrator (CHILD\Administrator)
Authenticated Users
SYSTEM
Domain Admins (CRYPT-LAB\Domain Admins)
Domain Users (CRYPT-LAB\Domain Users)
Enterprise Admins (CRYPT-LAB\Enterprise Admins)

Permissions for Administrator	Allow	Deny
Full Control	☐	☐
Read	☑	☐
Write	☑	☐
Enroll	☑	☐
Autoenroll	☐	☐

You can provide your own name to the template and save. Create additional templates as needed (I’m not going to use them in this blog post, it is up to you).

Now, let's add this template to CA server. To do so, in normal command prompt or PowerShell, type: `certsrv.msc`. In the opened console, expand CA node and select Certificate Templates node. Delete everything except Domain Controller template. Then, right-click on Certificate Templates node, New –> Certificate Template To Issue. In the dialog, pick the user template you created a couple of steps before. It should look like this:

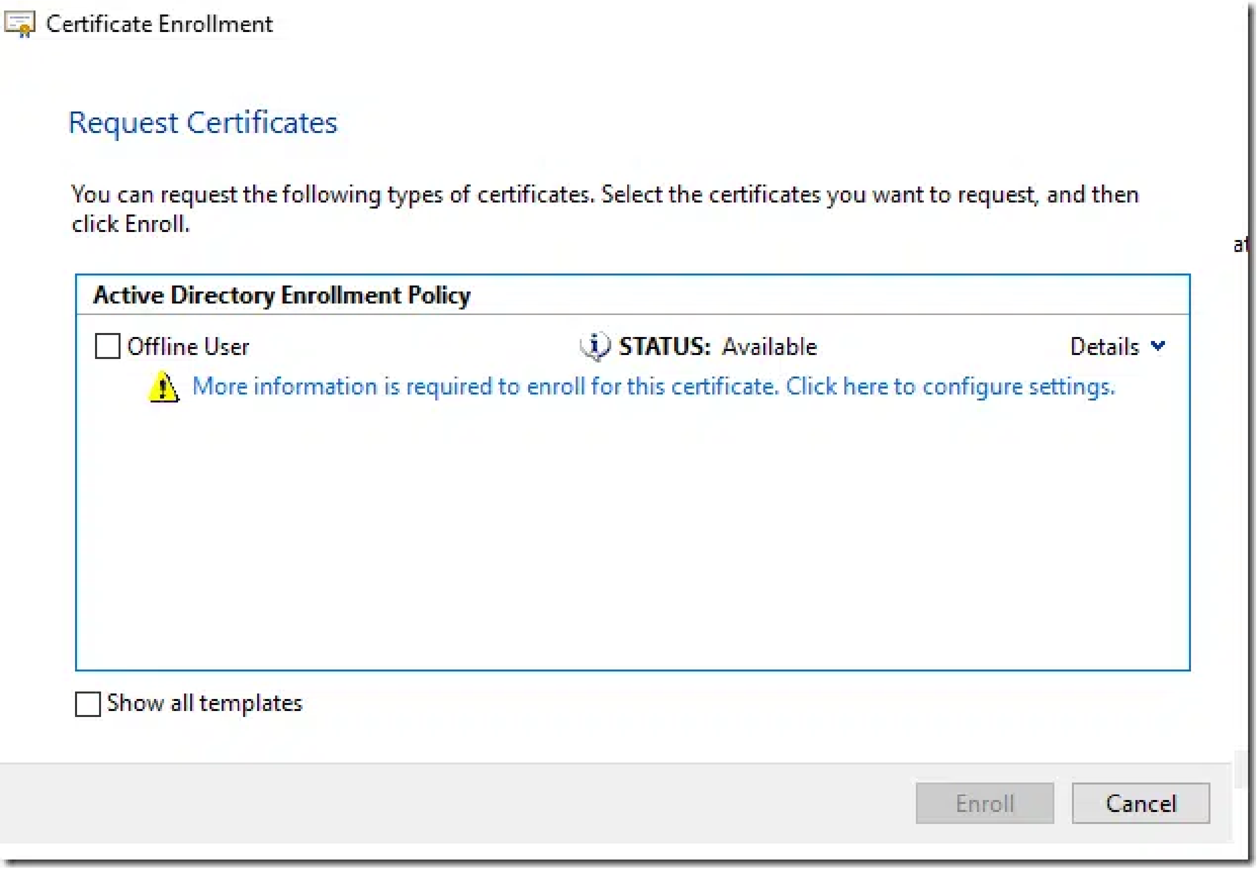


Name	Intended Purpose
Domain Controller	Client Authentication, Server Authentic...
Offline User	Client Authentication, Secure Email, En...

Why we left Domain Controller template? Well, it is necessary to automatically enable smart-card logons in entire forest. The beauty of this template is that AD domain controllers are hardcoded internally to automatically pick the certificate from this template once it is available. Other templates, like Domain Controller Authentication or Kerberos Authentication require autoenrollment policy configuration, which you may not execute easily. With Domain Controller template, everything is done automatically, no actions are required, all domain controllers in entire forest will enroll DC certificate. And, as I mentioned, this certificate is required to enable smart card logon globally.

The final step in our journey is to enroll a certificate for one of Enterprise Admins in your forest. To do so, in normal command prompt or PowerShell console type: `certmgr.msc`.

In the opened MMC, right-click on Personal, select All Tasks –> Request New Certificate. Follow the wizard until the list of templates appear:



Certificate Enrollment

Request Certificates

You can request the following types of certificates. Select the certificates you want to request, and then click Enroll.

Active Directory Enrollment Policy

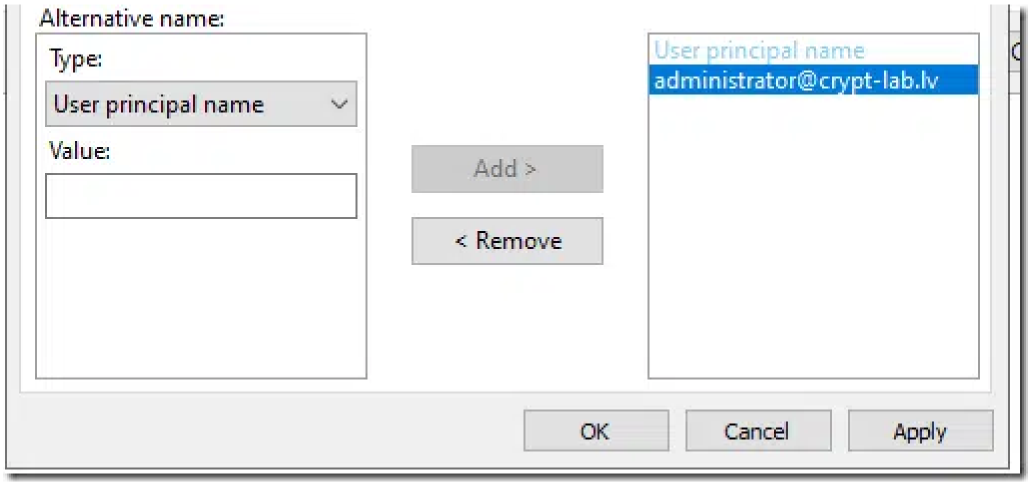
□ Offline User STATUS: Available Details

More information is required to enroll for this certificate. Click here to configure settings.

□ Show all templates

Select checkbox and press on a “More information” link:

In the Subject tab, navigate to Alternative Name box and from drop-down list select “User Principal Name” name type. In the text box below, type target account’s UPN. It must be in a form `accountname@domain.fqdn` and press Add button, so the name is moved to the right:



Alternative name:

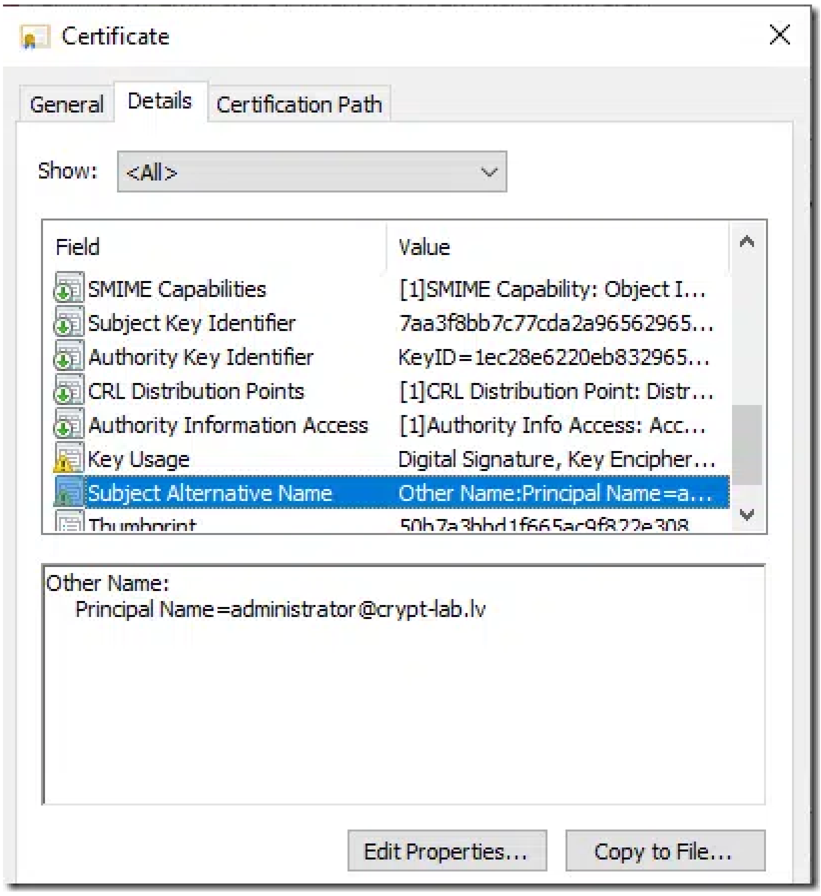
Type: User principal name

Value:

User principal name

administrator@crypt-lab.lv

Press Ok, then Enroll and voila:



Field	Value
SMIME Capabilities	[1]SMIME Capability: Object I...
Subject Key Identifier	7aa3f8bb7c77cda2a96562965...
Authority Key Identifier	KeyID=1ec28e6220eb832965...
CRL Distribution Points	[1]CRL Distribution Point: Distr...
Authority Information Access	[1]Authority Info Access: Acc...
Key Usage	Digital Signature, Key Encipher...
Subject Alternative Name	Other Name:Principal Name=a...
Thumbprint	50b7a3bbd1f665ac9f822e308

```
Other Name:
     Principal Name=administrator@crypt-lab.lv
```

You have a client authentication certificate for one of the enterprise admins.

Note: Be aware that the built-in Administrator account in every domain has no UPN by default. This means that you cannot spoof accounts without a UPN. This applies to built-in accounts only. Any newly created account will have a UPN and is subject for spoofing.

Epilogue

Although this article is somewhat entertaining, I need to ask closing question: How can you protect your AD from this sort of attack? Many attacks provided by the SpecterOps Certified Preowned report are somewhat imaginary and require reasonable efforts to prepare the ground for attack. But this one is quite serious, enabled by default and hard to protect from. One way I can imagine is to remove SYSTEM permissions from Public Key Services container ACLs and explicitly grant write permissions only to DCs from forest root. Wouldn’t this cause unfortunate side effects with replication? And permission inheritance is somewhat complex, it is easy to make a mistake and ruin your AD. Hope someone else will write mitigation steps.

And thanks, if you reached the end of the blog post!