Introducing PoshADCS

cyberstoph.org/posts/2019/11/introducing-poshadcs

November 9, 2019

TL;DR;TL;DR;

The script from this article leverages certificate templates to craft arbitrary smartcard logon certificates so you can impersonate any user. Use this to gain long-term stealthy persistence inside an Active Directory Domain or escalate your privileges to any user. Requirements: control over a user with write access on any kind of certificate template. The heavy lifting in the script is done with PowerView by Will Schroeder, so the kudos go to him.

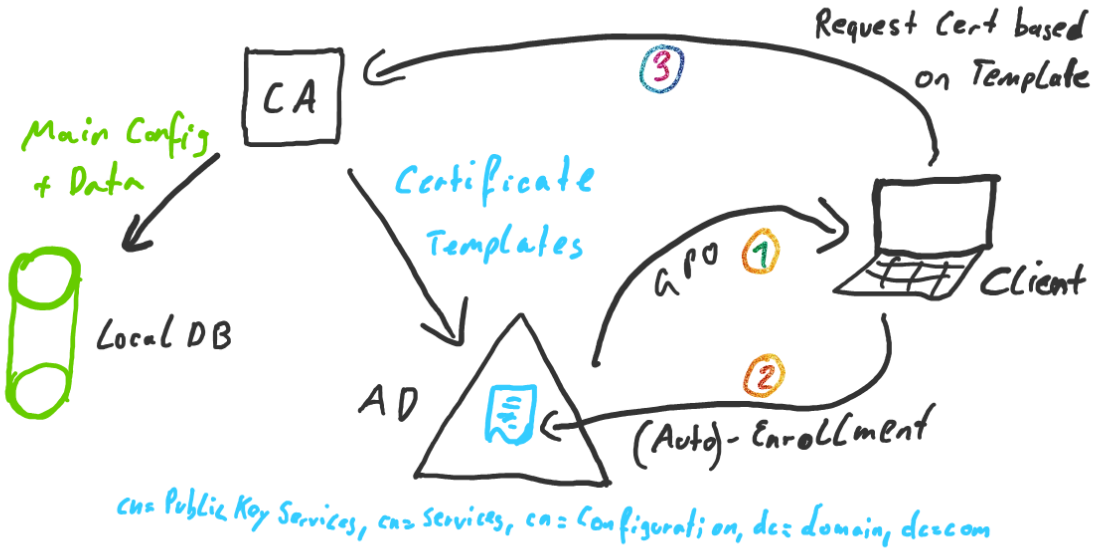
TL;DR;

Active Directory integrated Certificate Authorities (Enterprise CAs) store a part of their configuration in Active Directory. Espescially of interest are the so called “Certificate Templates”. Certificate templates are used by clients als well as by the CA to determine how to populate the fields in a certificate request as well as the resulting certificate. Usually there are a couple of published certificate templates in any organization that uses an AD integrated CA. If an attacker gains write access (Write and Enroll or WriteDACL) on any of these templates (e.g. through a service account) it is possible to “rewrite” any template so the attacker can enroll a smart card certificate for arbitrary users (e.g. domain admin) and then impersonate that user. This can be used as an ACL-based backdoor as well as an offensive attack vector.

What’s ADCS?

Active Directory Service Certificates is a server-role for Windows server that allows you to run a PKI (Public Key Infrastructure) on Windows. Upon installation, you can decide if you want to install a standalone or an enterprise CA. Simply put: a standalone CA is just a certificate authority running on Windows, whereas an enterprise CA integrates with Active Directory. You typically use the standalone CA for your root CA (because in can be offline or disconnected) and the enterprise CA for the issuing CA. So what does “enterprise” and “integrated” mean specifically?

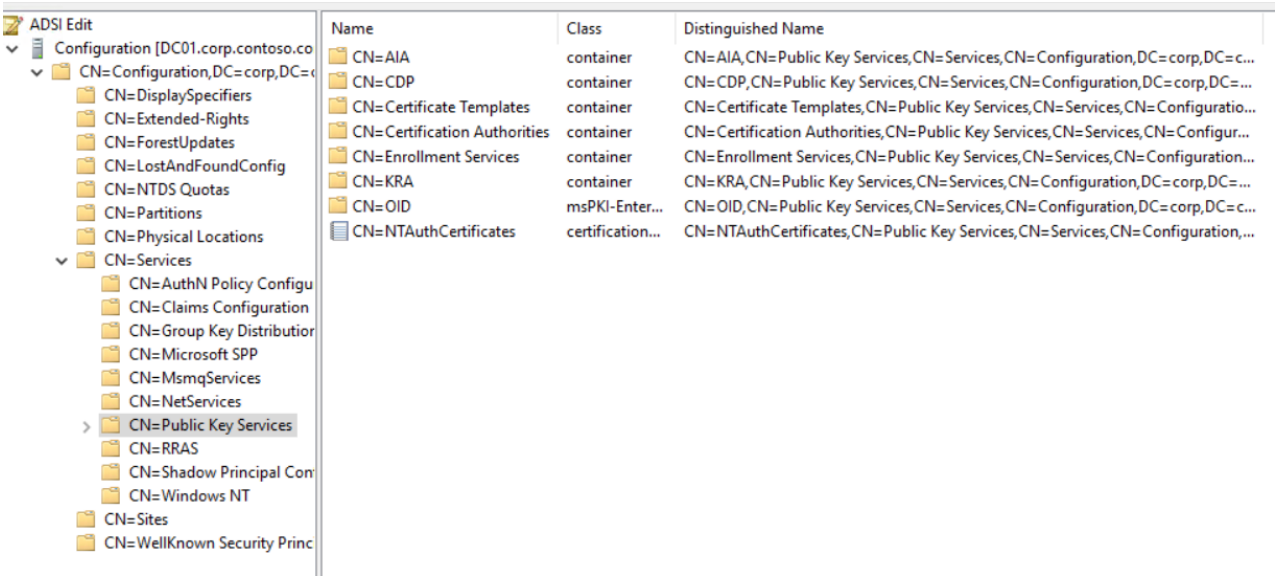
I tried to show the relevant interconnections in a picture. Though it looks like my little daughter drew it, I hope you get the point ;-)



An enterprise CA not only stores its configuration in a local database but also in the configuration partition of Active Directory under the following key:

```
CN=Public Key Services, CN=Services, CN=Configuration, DC=domain, dc=com
```

The data is split in different containers like “AIA” or “Certificate Templates”. We’ll focus on those relevant to our attack scenario for now.

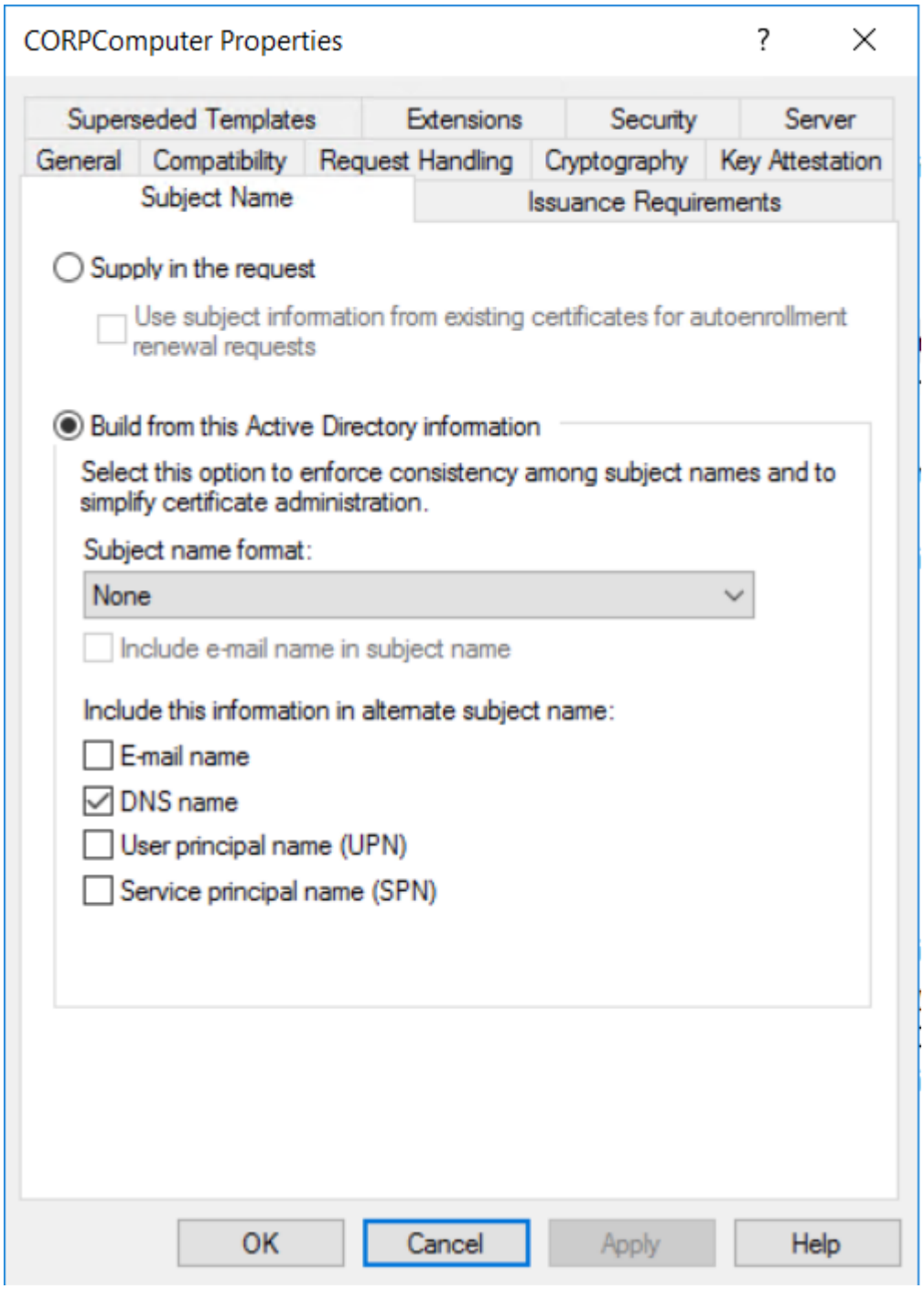


- **Certificate Templates:** stores the configuration for all certifcate templates. A certificate template basically is a blueprint for a certificate request (e.g. for an SMIME certificate). However not all certificate templates in this container are necessarily available for enrollment.

- **Enrollment Services:** Stores CA's available for certificate enrollment. Windows hosts use this container to automatically find CA's that can issue certificates to them. The respective CA object in this container has a member attribute called "certificate Templates". This attribute contains a list of all certificate templates (see above) that are available for enrollment on this CA. This is usually only a subset of all existing templates.
- **NtAuthCertificates:** Stores CA's that are permitted to issue smartcard logon certificates. If you try to log on with a smartcard certificate issued by a CA not in this list, authentication will fail. Every Enterprise CA is automatically added here.

What is a certificate template?

As mentioned earlier, a certificate template is like a blueprint to populate a certificate request. Here's an example: a certificate template for a "Computer Certificate" (e.g. for authentication using 802.1x) contains certain attributes relevant to that usage scenario. This template will typically be configured to use the requesting hosts DNS name as the Common Name in the certificate. The computer requesting the certificate will therefore populate the certificate request in accordance with the settings in the template. The CA too uses the configuration in the template for validation, so even if the client submits a wrong common name, the CA would change it to the one defined in the template before issung the certificate.



As you can see in the screenshot above, it is however also possible to allow the enrollment client to submit an arbitrary common name. This poses a certain risk because the CA has to trust the client to provide a correct CN. The CA administrator can limit the acces to a certificate template through the ACL of the template object in Active Directory. The ACL of the template not only defines who can modify the template but also who can enroll a template. Certificate enrollment can either happen automatically (Permission = Auto Enrollment) or manually (Permission = Enroll). Auto enrollment is configured via group policy and enforced throuh the group policy client during processing of the policy. If auto enrollment is enabled, the group policy client will look for and enroll all available certificate templates where the auto enrollment permission is set.

Attacking certificate templates

From a sysadmins perspective, certificate templates seem quite different. Every Enterprise CA ships with a couple of default templates and it is common practice that, if you want to use a certain template, you create a copy of one of the default templates and work with that. If you want to give a Windows client a certificate so it can participate in 802.1x, you would use a “Computer” template. If you want to issue SMIME certificates to your users, you’ll use a copy of the “User” template. Every template is named after its intended cause and this strengthens the idea, that you can only issue computer certificates from a “computer”-template. However, there is no fundamental difference between two different templates. Every template can issue every kind of certficate, if populated with the right parameters. If an attacker gains access (Write/Enroll or WriteDACL) to any template, it is possible to reconfigure that template to issue certificates for Smartcard Logon. The attacker can even enroll these certificate for any given user, since the setting that defines the CN of the certificate is controlled in the template.

Long story short, the attacker can impersonate any user by enrolling a smartcard logon certificate for that user. If the domain already uses smartcards for authentication, all requirements are already met and the attack should work out of the box. If smartcards are currently not in use in the target environment, the attack will still work as long as the following is true:

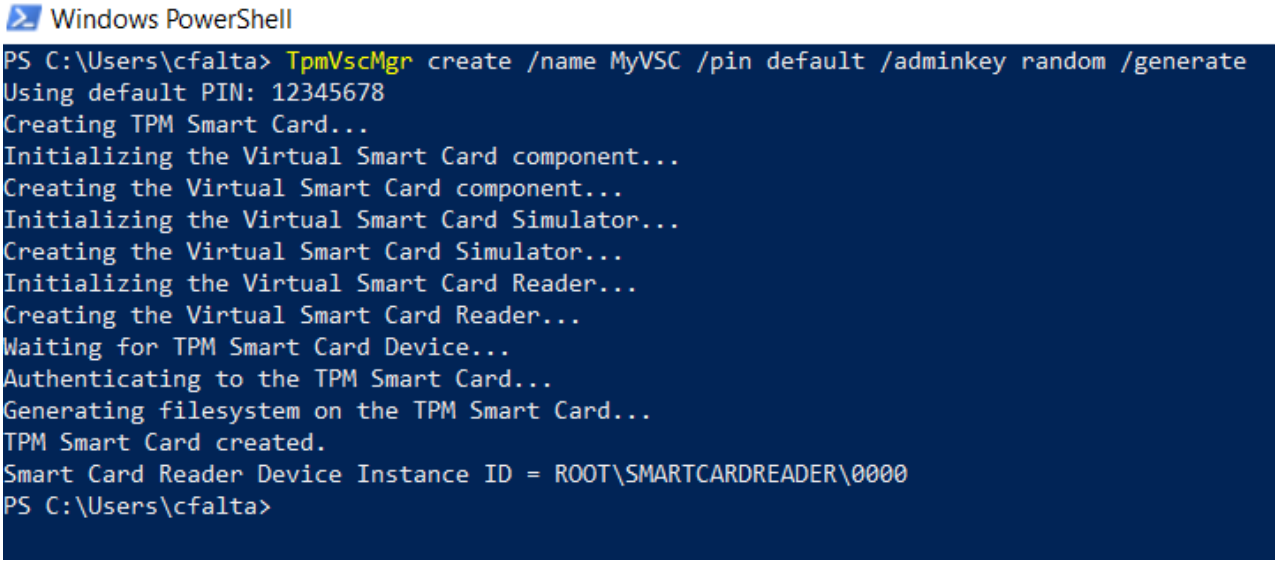
- The certificate of the Enterprise CA issuing the smartcard certificate needs to be present under “CN=NtAuthCertificates, CN=Public Key Services, CN=Services, CN=Configuration, DC=domain, dc=com”. This is done automatically during setup of the CA, so it shouldn’t be a problem.
- You obviously need a smartcard. This can be a physical smartcard, however bringing a real smartcard implies the need of physical access, which can be a challenge. Luckily, there’s a solution called “virtual smartcard”. More on that later.
- The domain controller(s) need’s a certificate issued from one of the following templates: Domain Controller, Domain Controller Authentication, Kerberos Authentication. This is probably the only crucial requirement that might not be met. However if there is an enterprise CA and auto enrollment enabled, from my experience it is very likely that the domain controllers already got the certificate automatically.

Virtual smartcards to the rescue

Since bringing a physical smartcard to a host you might have only remote access to can pose a challenge, there is a solution called virtual smartcard. Virtual smartcards where implemented in Windows 8 and allow you to use a TPM chip to create a virtual smartcard device. Since most modern business clients ship with a TPM chip, this shouldn’ be a problem. In fact, virtual smartcards are much more usable for the attack than real smartcards because they work out of the box on Windows clients and servers without the need of any special drivers and they work even over RDP. So you can use the virtual smartcard on a compromised client to log in to a server without TPM just as you would

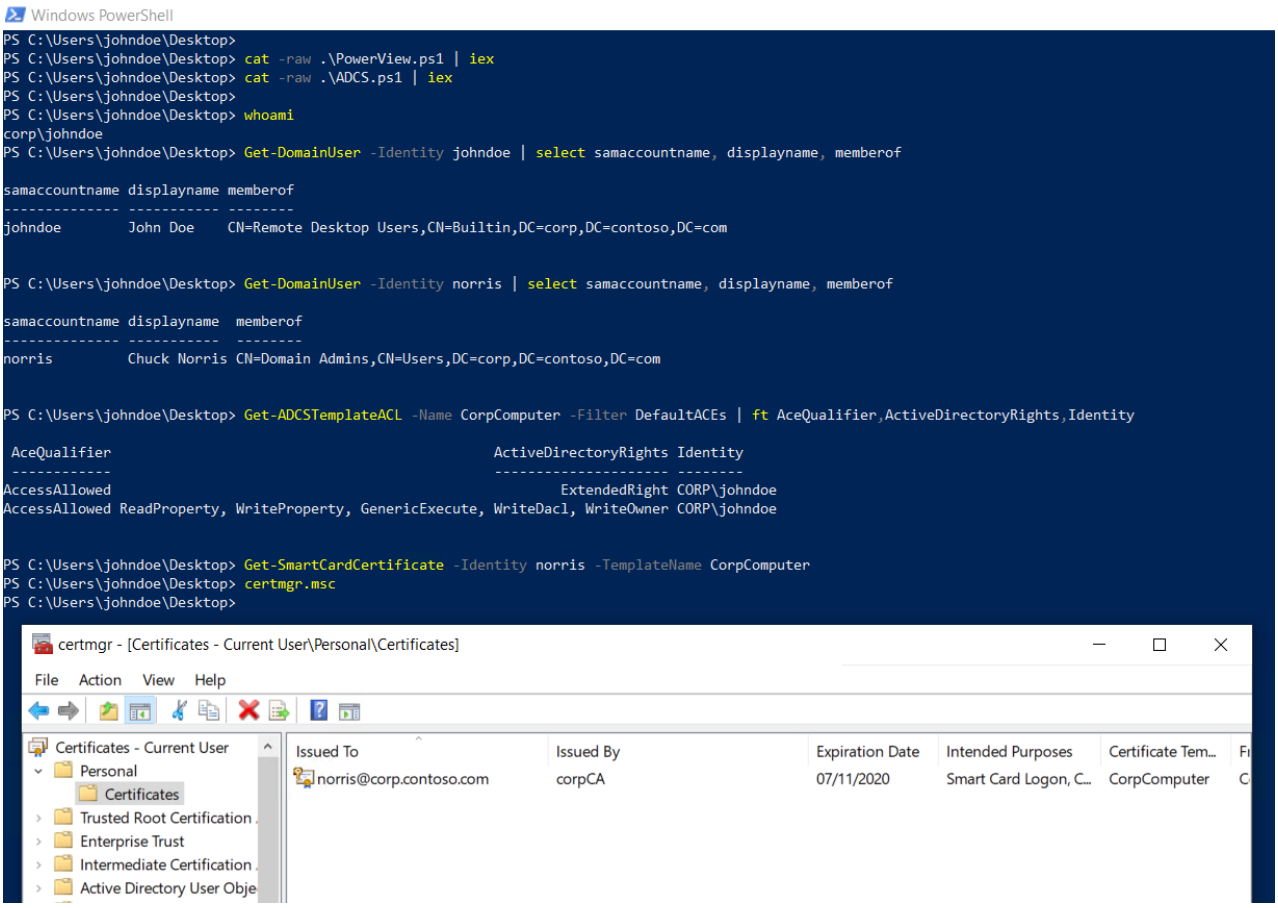
with username/password. Creating a virtual smartcard is simple as Windows provides a management tool called tpmvscmgr.exe Just run the command below to generate a smartcard with the default pin (12345678).

```
tpmvscmgr.exe /create /name VSC01 /pin default /adminkey random /generate
```



Proof of concept

I wrote a proof of concept script that implements the attack described above. It takes the samaccountname of a domain user to impersonate and the name of a certificate template you have access to. The script will rewrite the template to allow for smartcard enrollment, get the certificate and then reset the template to its original configuration :-)


Windows PowerShell
PS C:\Users\johndoe\Desktop>
PS C:\Users\johndoe\Desktop> cat -raw .\PowerView.ps1 | iex
PS C:\Users\johndoe\Desktop> cat -raw .\ADCS.ps1 | iex
PS C:\Users\johndoe\Desktop>
PS C:\Users\johndoe\Desktop> whoami
corp\johndoe
PS C:\Users\johndoe\Desktop> Get-DomainUser -Identity johndoe | select samaccountname, displayname, memberof
samaccountname displayname memberof
johndoe John Doe CN=Remote Desktop Users,CN=Builtin,DC=corp,DC=contoso,DC=com
PS C:\Users\johndoe\Desktop> Get-DomainUser -Identity norris | select samaccountname, displayname, memberof
samaccountname displayname memberof
norris Chuck Norris CN=Domain Admins,CN=Users,DC=corp,DC=contoso,DC=com
PS C:\Users\johndoe\Desktop> Get-ADCSTemplateACL -Name CorpComputer -Filter DefaultACEs | ft AceQualifier,ActiveDirectoryRights,Identity
AceQualifier ActiveDirectoryRights Identity
AccessAllowed ExtendedRight CORP\johndoe
AccessAllowed ReadProperty, WriteProperty, GenericExecute, WriteDacl, WriteOwner CORP\johndoe
PS C:\Users\johndoe\Desktop> Get-SmartCardCertificate -Identity norris -TemplateName CorpComputer
PS C:\Users\johndoe\Desktop> certmgr.msc
PS C:\Users\johndoe\Desktop>
certmgr - [Certificates - Current User\Personal\Certificates]
File Action View Help
Certificates - Current User
Personal
Certificates
Trusted Root Certification
Enterprise Trust
Intermediate Certification
Active Directory User Obje
Issued To Issued By Expiration Date Intended Purposes Certificate Tem...
norris@corp.contoso.com corpCA 07/11/2020 Smart Card Logon, C... CorpComputer