Rotation KRBTGT : Script PowerShell Fiable et Guide Complet 2026

> Rotation KRBTGT : guide complet avec script PowerShell qui corrige les failles du script Microsoft. Double rotation, réplication, Golden Ticket.

Le compte KRBTGT est le compte de service le plus critique de toute infrastructure Active Directory : il signe chaque ticket Kerberos émis dans le domaine. Sa compromission donne à un attaquant la capacité de fabriquer des Golden Tickets — des tickets Kerberos valides à vie, contournant totalement l'authentification. La rotation régulière de son mot de passe est la seule contre-mesure efficace, mais le script officiel Microsoft présente des lacunes opérationnelles qui exposent les équipes sécurité à des rotations incomplètes ou silencieusement défaillantes. Ce guide présente **Invoke-KrbtgtRotation.ps1**, un script PowerShell open source conçu pour les équipes SOC et les administrateurs Active Directory exigeants, avec vérification de réplication sur tous les DCs, double rotation automatisée, logs structurés JSON et support natif `-WhatIf`.

À RETENIR

Points clés

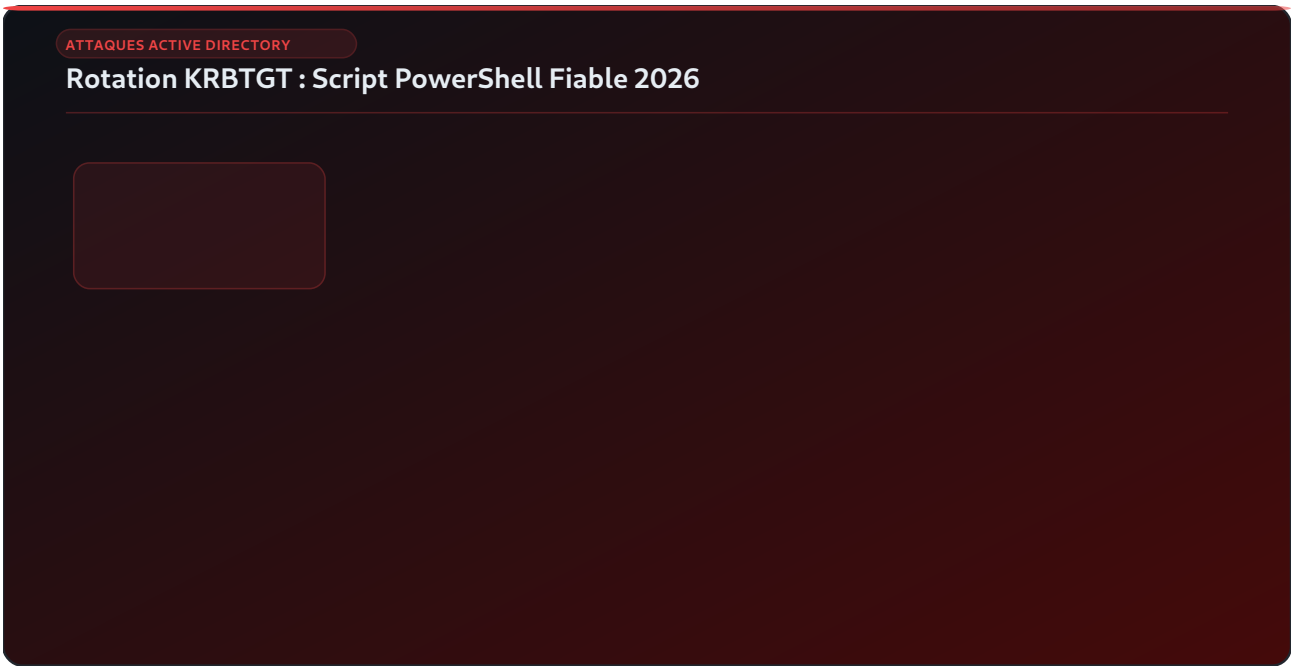
Le compte KRBTGT doit être tourné tous les 90 jours en exploitation normale, immédiatement après un incident.

Une double rotation (deux réinitialisations successives) est obligatoire après compromission pour invalider tous les Golden Tickets existants.

Le script officiel Microsoft ne vérifie pas la réplication sur tous les DCs : des contrôleurs peuvent rester avec l'ancien hash pendant des heures.

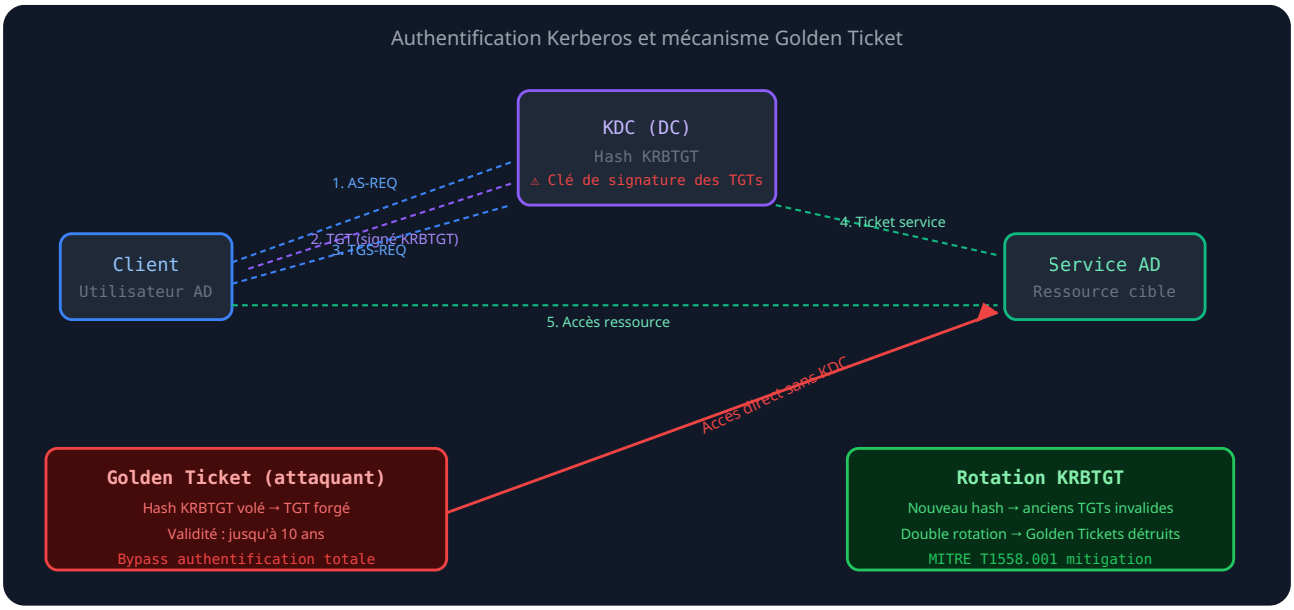
Invoke-KrbtgtRotation.ps1 v2.0 corrige ces lacunes et est téléchargeable gratuitement ci-dessous.

> MITRE ATT&CK répertorie l'abus KRBTGT sous T1558.001 — Golden Ticket.



Dans l'architecture Kerberos de Microsoft, le **Key Distribution Center** (KDC) tourne sur chaque contrôleur de domaine et repose sur deux fonctions principales : l'*Authentication Service* (AS) qui délivre les Ticket Granting Tickets (TGT), et le *Ticket Granting Service* (TGS) qui délivre les tickets de service. Ces deux composants partagent un secret commun : le hash du mot de passe du compte `KRBTGT`.

Concrètement, chaque TGT émis par un KDC est chiffré et signé avec le hash KRBTGT. Un compte utilisateur qui s'authentifie reçoit un TGT qu'il présentera ensuite au TGS pour obtenir des tickets d'accès aux ressources. La durée de vie par défaut d'un TGT est de 10 heures, renouvelable jusqu'à 7 jours.



Si un attaquant réussit à extraire le hash KRBTGT — via une attaque DCSync, un accès physique à un DC ou une technique Mimikatz — il peut forger des TGTs arbitraires pour n'importe quel utilisateur, y compris des comptes inexistants dans l'annuaire, avec une durée de validité pouvant atteindre dix ans. C'est le **Golden Ticket**.

La seule façon de dévalider un Golden Ticket existant est de changer deux fois le mot de passe KRBTGT. Une seule rotation laisse l'ancienne clé disponible en tant que clé N-1 (Kerberos conserve les deux derniers hashs pour la compatibilité ascendante). Seule la double rotation efface définitivement la clé compromise.

Les limites critiques du script officiel Microsoft

Microsoft publie depuis 2015 le script `New-KrbtgtKeys.ps1`, téléchargeable depuis le TechNet. Bien qu'il soit fonctionnel dans la majorité des cas, les équipes sécurité opérationnelles se heurtent à plusieurs limitations importantes :

⚡ Retour terrain

Dans mes missions de hardening Active Directory, le point le plus sous-estimé est la gestion du groupe 'Opérateurs de compte' et des délégations personnalisées créées au fil des années. Pour un groupe logistique de 1 500 utilisateurs, j'ai trouvé 312 ACL personnalisées dont 89 accordaient des permissions d'écriture sur des attributs sensibles (adminCount, memberOf sur groupes privilégiés) à des comptes d'utilisateurs standards. La dette technique en sécurité AD est souvent invisible jusqu'au premier audit sérieux.

Fonctionnalité	New-KrbtgtKeys.ps1 (Microsoft)	Invoke-KrbtgtRotation.ps1 v2.0
Vérification réplication avant rotation	Non	Oui — Get-ADReplicationFailure
Vérification KVNO post-rotation sur tous les DCs	Non	Oui — comparaison KVNO par DC
Support natif -WhatIf	Non	Oui — CmdletBinding(SupportsShouldProcess)
Double rotation automatisée	Manuelle seulement	Oui — paramètre -DoubleRotation
Logs structurés JSON	Non	Oui — export JSON complet
Mode audit (VerifyOnly)	Non	Oui — lecture seule, aucune modification
Génération mot de passe cryptographiquement sûr	System.Random (faible)	RNGCryptoServiceProvider, 127 chars
Vérification connectivité réseau DCs	Non	Oui — Test-Connection préalable
Rapport synthèse parseable	Sortie texte uniquement	JSON structuré + console colorée
Délai configurable avant rotation #2	Non	Oui — -DoubleRotationDelayMinutes

La limitation la plus dangereuse est l'absence de vérification post-rotation sur tous les contrôleurs de domaine. Dans un environnement avec des DCs en WAN ou des liens de réplication lents, il est courant qu'un DC reste plusieurs heures avec l'ancien hash KRBTGT. Si un attaquant dispose

encore d'un accès à ce DC, il peut continuer à émettre des tickets valides avec l'ancienne clé. Le script Microsoft n'alertera jamais sur cette situation.

Invoke-KrbtgtRotation.ps1 : architecture et fonctionnalités

Le script est structuré autour de huit fonctions spécialisées, chacune responsable d'une étape du processus. Cette architecture modulaire permet de tester chaque composant indépendamment et facilite l'intégration dans des pipelines d'automatisation (Ansible, Terraform, scripts CI/CD de hardening).

Test-ADReplicationHealth : le gardien de la cohérence

Avant toute modification, cette fonction interroge `Get-ADReplicationFailure` sur chaque DC pour détecter les partenariats de réplication défaillants. Elle retourne un booléen et journalise chaque échec avec son partenaire, le nombre d'erreurs consécutives et le dernier code d'erreur Windows. Cette vérification préalable évite les rotations partielles — le scénario le plus dangereux.

Get-KrbtgtStatus : visibilité complète avant et après

Cette fonction récupère le KVNO (Key Version Number) du compte KRBTGT depuis tous les DCs et détecte les divergences. Le KVNO est incrémenté à chaque changement de mot de passe : si deux DCs affichent des valeurs différentes, la réplication est incomplète. La fonction calcule également le nombre de jours depuis la dernière rotation et émet des alertes à 90 et 180 jours.

Invoke-KrbtgtPasswordReset : rotation avec génération sécurisée

Voici l'extrait central du script pour la génération du nouveau mot de passe :

```
function New-SecureRandomPassword {
    [OutputType([System.Security.SecureString])]
    param([int]$Length = 127)

    $charset  = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()_+-=[]{}
    $rng      = [System.Security.Cryptography.RNGCryptoServiceProvider]::new()
    $bytes    = [byte[]]::new($Length)
    $rng.GetBytes($bytes)
    $rng.Dispose()

    $password = [System.Text.StringBuilder]::new($Length)
    foreach ($b in $bytes) {
        [void]$password.Append($charset[$b % $charset.Length])
    }

    $secure = ConvertTo-SecureString -String $password.ToString() -AsPlainText -Force
    $password.Clear()  # Effacement mémoire
    return $secure
}
```

L'utilisation de `RNGCryptoServiceProvider` au lieu de `System.Random` (utilisé par le script Microsoft) garantit une entropie cryptographiquement sûre. Le mot de passe généré fait 127 caractères, bien au-delà de la limite Active Directory de 256 octets. La chaîne est immédiatement effacée de la mémoire après conversion en SecureString.

Télécharger et installer le script

Le script est distribué gratuitement sous licence MIT. Il n'a aucune dépendance externe hormis le module RSAT Active Directory, disponible dans toutes les installations Windows Server.

Invoke-KrbtgtRotation.ps1

Script PowerShell — Version 2.0 — MIT License

Télécharger le script

Pour débloquer le script après téléchargement depuis Internet (Windows marque automatiquement les fichiers téléchargés comme potentiellement dangereux), exécutez en tant qu'administrateur :

```
Unblock-File -Path .\Invoke-KrbtgtRotation.ps1
Get-Help .\Invoke-KrbtgtRotation.ps1 -Full
```

Pré-requis et permissions nécessaires

Quelles sont les conditions minimales pour exécuter le script en production ? Trois niveaux de prérequis sont à vérifier avant le premier lancement :

Prérequis système :

PowerShell 5.1 ou PowerShell 7+ (recommandé pour les environnements Core)

Module RSAT Active Directory installé (`Add-WindowsFeature RSAT-AD-PowerShell` sur Windows Server, ou `Add-WindowsCapability -Name Rsat.ActiveDirectory* -Online` sur Windows 10/11)

Accès réseau aux ports 135, 389, 445 et 49152+ vers tous les DCs

L'outil `repadmin.exe` doit être dans le PATH (inclus avec les RSAT AD)

Prérequis Active Directory :

Appartenance au groupe **Domain Admins**, ou délégation explicite du droit *Reset Password* sur `CN=krbtgt,CN=Users,DC=...`

Niveau fonctionnel de domaine Windows Server 2008 R2 minimum (pour `msDS-KeyVersionNumber`)

Tous les DCs doivent être joignables — les DCs hors ligne seront signalés mais n'empêchent pas la rotation si `-Force` est spécifié

Prérequis de politique de sécurité :

La politique de mot de passe du domaine doit accepter des mots de passe de 127 caractères (vérifiez avec `Get-ADDefaultDomainPasswordPolicy`)

Les comptes de service et les applications utilisant l'authentification Kerberos doivent être capables d'acquérir de nouveaux tickets sans interruption prolongée

---

Procédure de rotation pas à pas

Voici la procédure complète recommandée pour une rotation planifiée en production. Respecter l'ordre de ces étapes est critique pour éviter les interruptions de service.

T-7 jours — Préparation : inventoriez tous vos contrôleurs de domaine, vérifiez leur état de réplication avec `repadmin /showrepl` et `repadmin /replsummary`, et identifiez les éventuels DC en retard. Notifiez les équipes applicatives des applications critiques utilisant Kerberos (SQL Server, Exchange, SharePoint, applications Java avec SPNEGO).

T-1 jour — Audit préalable :

```
# Exécuter depuis un DC ou une station avec RSAT
.\Invoke-KrbtgtRotation.ps1 -VerifyOnly -LogPath "C:\Logs\krbtgt-audit-avant.log"
```

Analysez le fichier de log JSON et corrigez tout problème de réplication détecté avant de procéder.

Jour J — Rotation (fenêtre de maintenance) :

```
# Simulation préalable obligatoire
.\Invoke-KrbtgtRotation.ps1 -WhatIf

# Rotation effective avec confirmation interactive
.\Invoke-KrbtgtRotation.ps1 -ReplicationWaitSeconds 60 -LogPath "C:\Logs\krbtgt-rotation-$(Get-Da
```

Le script demandera de taper `CONFIRMER` avant toute action. Après la rotation, attendez 15 minutes et vérifiez les alertes sur vos systèmes de supervision (SIEM, Splunk, Sentinel) pour détecter toute application ne renouvelant pas correctement ses tickets.

T+30 minutes — Vérification post-rotation :

```
.\Invoke-KrbtgtRotation.ps1 -VerifyOnly -LogPath "C:\Logs\krbtgt-audit-apres.log"
```

Tous les DCs doivent afficher le même KVNO (ancien + 1). Si des DCs restent en retard, forcez la réplication avec `repadmin /syncall /AdeP` manuellement.

La double rotation : quand et pourquoi

La double rotation est la best practice imposée par Microsoft et l'ANSSI dans tous les scénarios post-incident impliquant une compromission potentielle de l'Active Directory. Pourquoi deux rotations et pas une seule suffisent-elles ?

Kerberos maintient les deux dernières versions du hash KRBTGT pour assurer la compatibilité : un TGT émis juste avant la rotation reste temporairement valide car le KDC accepte encore la clé N-1. Après une seule rotation, un Golden Ticket forgé avec l'ancienne clé (N-1) est encore valide. Après deux rotations, la clé compromise passe de N-1 à N-2 et est définitivement rejetée par tous les KDCs.

```
# Double rotation post-incident (recommandée après compromission)
.\Invoke-KrbtgtRotation.ps1 -DoubleRotation -Force -ReplicationWaitSeconds 60 -DoubleRotationDela
```

Le paramètre `-DoubleRotationDelayMinutes` (défaut : 10 minutes) correspond au délai minimum entre les deux rotations. Ce délai doit être supérieur à la moitié de la durée de vie maximale des tickets (*MaxTicketLifetime*, généralement 10 heures par défaut). Dans un contexte d'incident actif, 10 minutes constituent le minimum acceptable : cela laisse aux services le temps d'acquérir de nouveaux tickets avec les clés de la rotation #1 avant que la rotation #2 n'invalide à nouveau tout le parc de tickets.

Dans quel contexte déclencher une double rotation ? Après toute compromission confirmée ou suspectée d'un compte Domain Admin, après une attaque DCSync, après la découverte d'un Golden Ticket en production (logs Kerberos avec PAC anormal), ou après une réinstallation complète d'un DC compromis.

Vérifier la propagation sur tous les DCs

La propagation du nouveau hash KRBTGT à tous les contrôleurs de domaine est critique. Un DC qui n'a pas reçu la mise à jour continue à signer des tickets avec l'ancienne clé — ce qui peut maintenir un Golden Ticket valide si l'attaquant cible spécifiquement ce DC retardataire.

Le script automatise cette vérification via `Test-KrbtgtRotationSuccess`, mais voici comment effectuer une vérification manuelle complémentaire :

```
 # Vérifier le KVNO sur tous les DCs du domaine
$domain = Get-ADDomain
Get-ADDomainController -Filter * | ForEach-Object {
    $dc = $_
    try {
        $kvno = (Get-ADUser krbtgt -Properties msDS-KeyVersionNumber -Server $dc.HostName).'msDS-
        [PSCustomObject]@{
            DC   = $dc.Name
            KVNO = $kvno
            Site = $dc.Site
            OS   = $dc.OperatingSystem
        }
    } catch {
        [PSCustomObject]@{ DC = $dc.Name; KVNO = "ERREUR: $_"; Site = $dc.Site; OS = $dc.Operatin
    }
} | Format-Table -AutoSize
```

En complément, l'outil `repadmin /showrepl` permet de visualiser l'état de la réplication partition par partition. La partition `CN=Configuration` et la partition de domaine doivent toutes deux afficher une réplication réussie récente. Pour forcer une réplication urgente depuis le PDC vers tous les DCs :

```
 # Forcer la réplication sortante depuis le PDC vers tous les partenaires
repadmin /syncall /AdeP $((Get-ADDomain).DistinguishedName)

# Vérifier les erreurs de réplication actuelles
Get-ADReplicationFailure -Target (Get-ADDomainController -Filter *)
```

Planifier la rotation régulière (Task Scheduler / GPO)

La documentation Microsoft recommande une rotation KRBTGT tous les 180 jours en exploitation normale, et l'ANSSI préconise 90 jours dans son guide de sécurisation Active Directory. La pratique recommandée par les équipes Red Team est 90 jours, alignée sur les rotations de secrets d'infrastructure (certificats, comptes de service).

Pour automatiser la rotation via le Task Scheduler Windows :

```
 # Créer une tâche planifiée pour la rotation trimestrielle
$action  = New-ScheduledTaskAction -Execute "PowerShell.exe" `
    -Argument "-NonInteractive -File C:\Scripts\Invoke-KrbtgtRotation.ps1 -Force -LogPath C:\Logs
$trigger = New-ScheduledTaskTrigger -Weekly -WeeksInterval 13 -DaysOfWeek Sunday -At "02:00AM"
$settings = New-ScheduledTaskSettingsSet -RunOnlyIfNetworkAvailable -WakeToRun
$principal = New-ScheduledTaskPrincipal -UserId "DOMAIN\KrbtgtRotationSvc" -LogonType Password -R

Register-ScheduledTask -TaskName "KRBTGT Quarterly Rotation" `
    -Action $action -Trigger $trigger -Settings $settings -Principal $principal
```

Quelques précautions pour l'automatisation : le compte de service utilisé doit avoir les droits minimaux nécessaires (Reset Password sur CN=krbtgt uniquement, pas Domain Admins complet). Activez la journalisation des tâches planifiées et envoyez les logs JSON vers votre SIEM pour déclencher une alerte si la rotation échoue. La rotation automatisée ne doit jamais remplacer la vérification humaine post-rotation.

Pour l'intégration dans une GPO de hardening AD, consultez notre guide Kerberoasting 2026 qui couvre les politiques Kerberos avancées (durée de vie des tickets, AES forcé, comptes protégés).

MITRE ATT&CK : T1558.001 Golden Ticket — ce que la rotation empêche

La technique T1558.001 — Steal or Forge Kerberos Tickets: Golden Ticket est classée dans la tactique *Credential Access* et *Persistence* de la matrice MITRE ATT&CK. Elle est utilisée par des groupes APT majeurs : APT29 (Cozy Bear), APT28 (Fancy Bear), et des groupes de

ransomware comme LockBit et BlackCat lors de phases de déplacement latéral et de persistance longue durée.

Comment la rotation KRBTGT contrecarre-t-elle cette technique ? En changeant le hash KRBTGT, tous les TGTs signés avec l'ancien hash deviennent invalides dès que les KDCs ont reçu la mise à jour. Un attaquant qui avait forgé un Golden Ticket avec l'ancien hash se retrouve avec un artefact inutilisable. Il ne peut pas générer de nouveau Golden Ticket sans disposer à nouveau du hash courant.

Les indicateurs de compromission (IoC) d'un Golden Ticket à surveiller dans votre SIEM :

Événement Windows **4769** (demande de ticket Kerberos) avec un compte utilisateur dont le *Ticket Encryption Type* est RC4 (0x17) sur des systèmes configurés pour AES uniquement

Événement **4768** (demande TGT) suivi immédiatement d'accès à des ressources sensibles sans historique d'authentification normal

Durée de vie de ticket anormalement élevée dans les logs Kerberos

Accès réseau depuis des comptes avec SID correspondant à des comptes désactivés ou supprimés

Pour une protection complète contre les attaques de persistance Kerberos, la rotation KRBTGT doit être combinée avec : le déploiement du *Protected Users Security Group* pour les comptes privilégiés, l'activation de *Kerberos Armoring* (FAST), la désactivation de RC4 (guide ANSSI de sécurisation Active Directory), et la mise en place du tiering Active Directory.

Questions fréquentes

La rotation KRBTGT coupe-t-elle les connexions en cours ?

Non, les sessions TCP/IP et SMB existantes ne sont pas interrompues immédiatement. Les applications qui ont déjà obtenu des tickets de service (TGS) continueront à fonctionner jusqu'à l'expiration naturelle de ces tickets (généralement 10 heures). Seules les nouvelles demandes de TGT nécessiteront la ré-authentification si le client essaie d'utiliser un TGT expiré. En pratique, la disruption est minimale pour les utilisateurs interactifs. Les applications utilisant des tickets

avec renouvellement automatique — comme Exchange ou SQL Server — peuvent nécessiter un redémarrage du service si leur ticket de service expire pendant la fenêtre de propagation.

Quelle est la différence entre KVNO et le hash KRBTGT ?

Le KVNO (Key Version Number) est un entier stocké dans l'attribut `msDS-KeyVersionNumber` du compte KRBTGT et incrémenté à chaque changement de mot de passe. Il permet au KDC d'identifier quelle version de la clé a été utilisée pour signer un ticket donné. Le hash est la valeur cryptographique réelle dérivée du mot de passe. Pour invalider des Golden Tickets, c'est le hash qui doit changer — le KVNO est simplement l'indicateur de version qui permet de vérifier que le changement s'est bien propagé sur tous les DCs.

Peut-on automatiser la rotation sans fenêtre de maintenance ?

Techniquement oui, mais ce n'est pas recommandé en production sans préparation. Une rotation non annoncée peut créer des incidents sur des applications sensibles à l'authentification Kerberos, notamment les applications Java utilisant JAAS/JGSS, les services Linux intégrés à AD via SSSD (voir notre article sur l'intégration Linux dans Active Directory), et les serveurs SQL Server configurés avec la délégation Kerberos contrainte. La rotation avec `-Force` en dehors d'une fenêtre de maintenance est réservée aux réponses à incident.

Comment vérifier qu'un Golden Ticket est utilisé sur mon réseau ?

Activez l'audit Kerberos avancé (stratégie *Audit Kerberos Authentication Service* et *Audit Kerberos Service Ticket Operations* sur les DCs), puis recherchez dans les événements 4768/4769 les tickets avec une durée de vie supérieure à `MaxTicketLifetime` + `MaxClockSkew`, ou les tickets utilisés après expiration. Pour une détection avancée, le framework BloodHound et Microsoft Defender for Identity disposent de détections spécifiques Golden Ticket.

Intégration dans un plan de réponse à incident

La rotation KRBTGT n'est qu'une étape dans la chaîne de réponse à une compromission Active Directory. Les équipes IR (Incident Response) suivent généralement un séquencement précis pour éviter de « télégraphier » leurs actions à l'attaquant tout en restaurant rapidement l'intégrité du domaine.

La séquence recommandée après détection d'une compromission Active Directory étendue est la suivante :

Phase 1 — Confinement (0-2 heures) : Isoler le ou les DCs compromis du réseau si possible, révoquer les sessions actives des comptes administrateurs compromis, désactiver les comptes suspects. Ne pas encore changer KRBTGT — l'attaquant utilisant un Golden Ticket ne sera pas alerté par les logs d'authentification et vous avez besoin de temps pour évaluer l'étendue.

Phase 2 — Collecte forensique (2-6 heures) : Collecter les journaux d'événements Windows de tous les DCs (événements 4624, 4625, 4648, 4672, 4768, 4769, 4776), exporter les attributs de réplication AD (`repadmin /showrepl /csv`), capturer la liste des comptes membres de Domain Admins et Enterprise Admins, analyser les tickets Kerberos actifs avec `klist` et Mimikatz en environnement forensique.

Phase 3 — Éradication (6-12 heures) : Réinitialiser TOUS les comptes administrateurs compromis *avant* la rotation KRBTGT (sinon un attaquant peut utiliser les anciens credentials pour se ré-authentifier), puis effectuer la double rotation KRBTGT avec `Invoke-KrbtgtRotation.ps1 -DoubleRotation -Force`. Réinitialiser également le compte DSRM sur chaque DC si la compromission a pu l'exposer.

Phase 4 — Vérification (12-24 heures) : Surveiller les événements 4769 pour les tentatives d'utilisation d'anciens tickets, valider la propagation KVNO sur tous les DCs avec `-VerifyOnly`, vérifier les logs applicatifs pour détecter les services impactés par la rotation.

Les techniques MITRE ATT&CK associées à cette chaîne d'attaque couvrent également T1207 (Rogue Domain Controller), T1003.006 (NTDS extraction), et T1550.003 (Pass-the-Ticket) — toutes nécessitent des contre-mesures complémentaires à la rotation KRBTGT.

Hardening complémentaire autour du compte KRBTGT

La rotation KRBTGT est nécessaire mais pas suffisante. Un programme de hardening Kerberos complet comprend plusieurs couches de protection qui réduisent à la fois la surface d'attaque et la durée de vie d'une compromission.

Désactiver RC4 pour les comptes sensibles : Le chiffrement RC4 (ARCFOUR-HMAC-MD5, type 23) est beaucoup plus facile à attaquer que AES-256. Pour forcer AES sur les comptes privilegiés :

```
 # Forcer AES 256 uniquement sur le compte KRBTGT
Set-ADUser -Identity krbtgt -KerberosEncryptionType AES256

# Vérifier les types de chiffrement supportés
(Get-ADUser krbtgt -Properties msDS-SupportedEncryptionTypes).'msDS-SupportedEncryptionTypes'
# 18 = AES128 + AES256 (correct)
# 31 = tout activé dont RC4 (dangereux)
```

Protected Users Security Group : Ajoutez tous les comptes administrateurs à ce groupe spécial (disponible depuis Windows Server 2012 R2). Les membres ne peuvent pas s'authentifier avec NTLM, ne peuvent pas utiliser DES ou RC4 pour Kerberos, et leurs TGTs ne peuvent pas être délégués. Cela rend les attaques Pass-the-Hash et la plupart des formes de délégation Kerberos abusive inopérantes.

```
 # Ajouter un compte au groupe Protected Users
Add-ADGroupMember -Identity "Protected Users" -Members "AdminJDupont"

# Lister tous les membres du groupe
Get-ADGroupMember -Identity "Protected Users" | Select-Object Name, SamAccountName, ObjectClass
```

Kerberos Armoring (FAST) : Le *Flexible Authentication Secure Tunneling* protège les échanges Kerberos contre les attaques d'interception et rend l'AS-REP Roasting impossible pour les comptes qui ne nécessitent pas de pré-authentification. Activez-le via GPO : *Computer Configuration* > *Policies* > *Administrative Templates* > *System* > *Kerberos* > *Support Compound Authentication*.

Durée de vie des tickets : Réduire `MaxTicketLifetime` à 4 heures (défaut : 10 heures) et `MaxRenewAge` à 7 jours limite la fenêtre d'exploitation d'un Golden Ticket compromis entre deux

rotations. Cette modification s'applique via la GPO *Default Domain Policy* dans *Computer Configuration > Windows Settings > Security Settings > Account Policies > Kerberos Policy*.

Pour aller plus loin dans l'audit de l'état Kerberos de votre domaine, notre service d'audit Active Directory automatisé couvre l'ensemble de ces paramètres avec des recommandations priorisées adaptées à votre environnement.

Logs structurés JSON : exploiter la sortie du script

L'un des apports majeurs d'Invoke-KrbtgtRotation.ps1 par rapport au script Microsoft est l'export JSON structuré, conçu pour être ingéré directement par un SIEM ou une solution de log management. Voici la structure d'un rapport type :

```
{
  "script_version": "2.0",
  "execution_start": "2026-05-25T02:00:00Z",
  "execution_end": "2026-05-25T02:08:42Z",
  "duration_seconds": 522,
  "rotations_done": 1,
  "mode": "SingleRotation",
  "events": [
    {
      "timestamp": "2026-05-25T02:00:01.123Z",
      "level": "SUCCESS",
      "message": "Module ActiveDirectory chargé",
      "data": {}
    },
    {
      "timestamp": "2026-05-25T02:00:02.456Z",
      "level": "INFO",
      "message": "Domaine détecté",
      "data": {
        "Name": "corp.contoso.com",
        "DN": "DC=corp,DC=contoso,DC=com",
        "PDC": "DC01.corp.contoso.com",
        "Level": "Windows2016Domain"
      }
    },
    {
      "timestamp": "2026-05-25T02:05:30.789Z",
      "level": "SUCCESS",
      "message": "Mot de passe KRBTGT réinitialisé (rotation #1)",
      "data": {
        "Server": "DC01.corp.contoso.com",
        "Timestamp": "2026-05-25T02:05:30Z"
      }
    }
  ]
}
```


Pour ingérer ce log dans Splunk :

```
index=windows source="C:\\Logs\\krbtgt-*.log" sourcetype=json
| spath output=level path=events{}.level
| spath output=message path=events{}.message
| search level=ERROR OR level=WARNING
| table _time, level, message
```

Pour une alerte Sentinel basée sur ces logs, créez une règle analytique qui déclenche une alerte si `rotations_done = 0` dans les logs de la tâche planifiée trimestrielle — signe que la rotation automatisée a échoué silencieusement.

