Mode Opératoire

Rotation du compte `krbtgt` (double rotation Golden Ticket)

Code :	MO-AD-002
Version :	1.0
Date :	15 avril 2026
Auteur :	Cédric LEGRAND
Classification :	USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	15/04/2026	Création initiale — procédure exécutée au titre du Sprint 2 (T21). Phase 1 : 14/04 13 :02 :38. Phase 2 : 15/04 18 :52 :21.

1 Objet

Ce mode opératoire décrit la **double rotation du compte `krbtgt`**, compte système Active Directory qui signe l'intégralité des tickets Kerberos (TGT) émis par les contrôleurs de domaine. La procédure vise à neutraliser toute attaque de type *Golden Ticket* : une clé `krbtgt` compromise permet à un attaquant de forger des tickets Kerberos arbitraires, valables jusqu'à ce que la clé soit invalidée par une rotation — **et cette rotation doit être exécutée deux fois** (la première ne suffit pas, voir § 4).

Fréquence recommandée : tous les six mois (ou 180 jours), ou immédiatement après toute suspicion de compromission : extraction `NTDS.dit`, attaque *DCSync*, départ d'un administrateur de domaine.

⚠ Contexte initial

Le compte `krbtgt` du domaine `bts.sio` a été créé le **06/09/2022 à 17 :38** et n'avait **jamais été tourné** avant l'exécution de cette procédure en avril 2026, soit 1 316 jours de fenêtre d'exploitation potentielle. PingCastle signalait cet état via la règle *A-Krbtgt #37*. Ce mode opératoire formalise le traitement de ce risque.

2 Champ d'application

Public concerné	Administrateurs du domaine BTS SIO (rôle *Domain Admins*)
Systèmes ciblés	Domaine `bts.sio`, contrôleurs DC1 (Srv2022, 10.0.112.2) et DC2 (Srv2022Phy, 10.0.112.3)
Outils	PowerShell avec module ActiveDirectory, `repadmin`, WinRM via `pywinrm`
Authentification	Compte Domain Admin (`BTS\Administrateur` ou équivalent)
Durée	Deux interventions de ≈ 5 minutes espacées de 10 à 48 heures
Présentiel requis	Non : la procédure s'exécute intégralement via WinRM depuis le poste d'administration, sous couvert d'une connexion VPN WireGuard à l'infrastructure

3 Prérequis

Prérequis

- **MO-AD-005 exécuté** : santé AD validée (`repadmin /replsummary` sans erreur bloquante, services `NTDS`, `kdc`, `Netlogon` UP, FSMO accessibles)
- **MO-AD-006 exécuté** : backup *System State* de DC1 récent (< 7 jours), vérifié par `wbadmin get versions`. Une restauration authoritative de l'objet `krbtgt` ne se fait que depuis ce backup
- Accès WinRM (port 5985) aux deux contrôleurs de domaine avec un compte Domain Admin
- Fenêtre d'intervention d'au moins **12 heures** entre Phase 1 et Phase 2, idéalement 24 à 48 heures
- Pas de séquence d'examens ou de TP critiques sur les deux jours concernés (risque marginal de déconnexion si la Phase 2 est déclenchée trop tôt)

Erreur RPC 110 chronique sur DC2

Dans l'infrastructure actuelle, `repadmin /syncall /AdeP` et certaines sous-commandes de `repadmin` remontent une **erreur 110 (RPC timeout)** sur le serveur `Srv2022Phy` (port 5722 bloqué, DFSR partiellement cassé depuis le 27/03/2026). Cette erreur **n'interdit pas la rotation** : la réplication AD principale passe par les ports 135, 389 et 636 qui sont ouverts, et `repadmin /replsummary` montre une latence inférieure à une minute entre les deux DCs. Le dépannage de DFSR/RPC 5722 fait l'objet d'une intervention distincte ; il ne doit pas retarder la rotation `krbtgt`.

4 Théorie : compte `krbtgt` et attaque Golden Ticket

4.1 Rôle du compte `krbtgt`

Le compte `krbtgt` est un compte de service AD automatiquement créé à la promotion du premier contrôleur de domaine. Il est :

— **désactivé** au sens ouverture de session (aucun logon interactif ou réseau possible) ;
— **porteur de la clé symétrique utilisée par tous les KDCs** du domaine pour signer et chiffrer les TGT ;
— **stocké dans `NTDS.dit`** avec un hash NTLM accessible en lecture au système et aux comptes de haut privilège.

Chaque ticket TGT émis par un KDC est signé avec la clé courante de `krbtgt`. Sa durée de vie par défaut est de dix heures (extensible à sept jours par renouvellement). Tant qu'un TGT est valide, son porteur peut demander des tickets de service (TGS) auprès du KDC sans réauthentification.

4.2 Attaque Golden Ticket

Un attaquant qui parvient à extraire le hash NTLM de `krbtgt` (par *DCSync*, dump mémoire LSASS sur un DC, ou lecture `NTDS.dit` hors ligne) peut alors forger, sur n'importe quelle machine et sans base AD, un TGT arbitraire. Les caractéristiques typiques d'un *Golden Ticket* sont :

— identité usurpée libre (y compris un compte inexistant ou *Administrateur*) ;
— appartenance aux groupes souhaités dont *Enterprise Admins* ou *Domain Admins* ;
— durée de vie poussée à dix ans (valeur arbitraire choisie par l'attaquant) ;
— validité tant que la clé `krbtgt` qui l'a signé est acceptée par les KDCs.

La seule contre-mesure définitive consiste à **changer le hash `krbtgt`**. Les tickets forgés avec l'ancien hash deviennent alors invalides dès qu'ils sont présentés à un KDC dont la clé `krbtgt` a été rotatée.

4.3 Mécanique N / N-1 et nécessité d'une double rotation

Active Directory conserve **deux versions** du hash `krbtgt` : la version courante N et la précédente N-1. Ce mécanisme évite d'invalider massivement les tickets encore en circulation lors d'un changement de clé. Concrètement :

— après **une seule rotation**, un ticket signé avec l'ancien hash (N-1) reste accepté le temps de sa durée de vie résiduelle. Un Golden Ticket forgé avant la rotation continue donc de fonctionner ;

— **la deuxième rotation**, exécutée après expiration normale des tickets actifs, décale la clé N-1 vers la clé N-2 (jetée par AD). À ce moment, tout ticket signé avec l'ancien hash est rejeté.

L'intervalle minimal entre les deux rotations est dicté par la durée de vie des TGTs (dix heures par défaut). Microsoft recommande 10 à 24 heures, et interdit formellement les deux rotations consécutives rapprochées qui « cassent » l'ensemble des tickets en vol.

5 Procédure Phase 1 : première rotation

5.1 Pré-vérifications

Se connecter à DC1 via WinRM depuis le poste d'administration :

```
python3 -c "
import winrm
s = winrm.Session('10.0.112.2',
    auth=('BTS\\\\Administrateur', OLD_PW), transport='ntlm')
print(s.run_ps(open('/tmp/krbtgt_precheck.ps1').read()).std_out.decode('cp850'))"
```

Contenu du script `krbtgt_precheck.ps1` :

```
$k = Get-ADUser krbtgt -Properties PasswordLastSet, Created
$days = ((Get-Date) - $k.PasswordLastSet).Days
Write-Host "Derniere rotation : $($k.PasswordLastSet) ($days jours)"

repadmin /replsummary
Get-Service NTDS, kdc, DNS, Netlogon | Format-Table
netdom query fsmo
```

Vérification

- ☐ `repadmin /replsummary` : 0 échec, latence < 5 min
- ☐ Services `NTDS`, `kdc`, `Netlogon` : état `Running` sur DC1 et DC2
- ☐ Cinq rôles FSMO localisés sur un DC disponible (ici DC1)
- ☐ Backup *System State* DC1 de moins de sept jours (via `wbadmin get versions`)

5.2 Exécution Phase 1

Étape 1 — Génération et application du nouveau mot de passe

```
# Bloc PowerShell injecte via WinRM sur DC1
$charset = 33..126 | ForEach-Object { [char]$_ }
$newPwd  = -join ($charset | Get-Random -Count 32)
$secure  = ConvertTo-SecureString -String $newPwd -AsPlainText -Force

Set-ADAccountPassword -Identity krbtgt -NewPassword $secure -Reset

$newPwd = $null     # ne JAMAIS logger le mot de passe
[System.GC]::Collect()
```

Le mot de passe n'est **pas stocké** et n'a pas besoin de l'être : on ne se sert jamais du mot de passe de `krbtgt` en clair, seul son hash est utilisé par les KDCs. Microsoft recommande une longueur de 24 à 32 caractères aléatoires.

Étape 2 — Forcer la réplication vers DC2

```
repadmin /syncall /AdeP
Start-Sleep -Seconds 10
repadmin /replsummary
```

Le drapeau `/AdeP` force une synchronisation **pull** sur tous les DCs et attend la fin de chaque réplication avant de rendre la main. L'option `/e` étend à tous les sites.

Étape 3 — Confirmer la propagation sur les deux DCs

```
foreach ($dc in (Get-ADDomainController -Filter *)) {
    Get-ADUser krbtgt -Server $dc.HostName -Properties PasswordLastSet |
        Format-Table SamAccountName, PasswordLastSet
}
```

La valeur de `PasswordLastSet` doit être identique sur DC1 et DC2 (à la seconde près, une fois la réplication terminée).

Horodatage de référence

Lors de l'exécution réelle du Sprint 2, Phase 1 a été exécutée le **14/04/2026 à 13 :02 :38**. Conserver ce timestamp (capture d'écran ou journal de session) est indispensable pour décider du moment de la Phase 2.

6 Fenêtre d'attente de 10 à 48 heures

Entre Phase 1 et Phase 2, **ne rien modifier** sur les comptes de service ni sur les tickets. Le délai a trois objectifs :

1. permettre à tous les TGTs en circulation de se renouveler avec la clé N (celle qui vient d'être posée) ;
2. permettre à la réplication AD de propager la nouvelle clé sur l'ensemble des DCs (mesurée en secondes dans une infra saine, mais laisser une marge) ;
3. détecter précocement toute régression (services cassés, `klist` des postes clients ne renouvelant pas leurs tickets, etc.) avant de figer la rotation.

Tracker l'échéance

Un simple oneliner PowerShell lancé périodiquement répond à la question « quand puis-je lancer la Phase 2 ? » :

```
$h = ((Get-Date) - (Get-ADUser krbtgt -Properties
    PasswordLastSet).PasswordLastSet).TotalHours
"Heures depuis Phase 1 : {0:N2}" -f $h
```

7 Procédure Phase 2 : seconde rotation

7.1 Garde-fou pré-Phase 2

Avant toute chose, vérifier que l'intervalle est respecté :

```
$h = ((Get-Date) - (Get-ADUser krbtgt -Properties
    PasswordLastSet).PasswordLastSet).TotalHours
if ($h -lt 10) {
    throw "Intervalle insuffisant ($([math]::Round($h,1)) h) : ANNULATION
    Phase 2"
}
```

7.2 Exécution Phase 2

Le bloc est **rigoureusement identique** à la Phase 1 : même génération aléatoire 32 caractères, même `Set-ADAccountPassword -Identity krbtgt -Reset`, même `repadmin /syncall /AdeP`. Cette répétition est l'essence de la procédure : c'est le deuxième cycle qui jette définitivement la clé N-1.

Étape 4 — Vérification finale

```
$before = [datetime]"2026-04-14 13:02:38"    # horodatage Phase 1
$after  = (Get-ADUser krbtgt -Properties PasswordLastSet).PasswordLastSet
$delta  = ($after - $before).TotalHours

"Phase 1 : $before"
"Phase 2 : $after"
"Delta   : $([math]::Round($delta,2)) heures"
```

La valeur de `$delta` doit être supérieure à 10 heures et inférieure à la durée maximale recommandée ($\sim$ 48 h). Dans l'exécution de référence de ce MO, Δ = **29,81 heures** (Phase 2 le 15/04/2026 à 18 :52 :21).

8 Vérifications post-rotation

Vérification

Après la Phase 2, valider chaque item ci-dessous :

- ☐ `PasswordLastSet krbtgt` identique sur DC1 et DC2
- ☐ `repadmin /replsummary` : 0 échec, latence < 5 min
- ☐ `repadmin /showrepl` : dernières réplications réussies sur chaque DC
- ☐ Connexion interactive testée sur un poste du VLAN administratif (logon standard)
- ☐ Connexion interactive testée sur un poste du VLAN pédagogique (compte élève)
- ☐ Tâches planifiées critiques vérifiées (T43 *backup System State* notamment)
- ☐ Capture `klist tgt` avec nouveau `KerbTicket Encryption Type` et incrément de `kvno`

Côté poste client, forcer le renouvellement pour valider l'opération :

```
klist purge                 # vide le cache local
gpupdate /force             # force un rafraichissement GPO
klist tgt                   # doit afficher un ticket frais
```

Cache NTLM

Le *cache NTLM* des DCs et des clients conserve pendant environ une heure la correspondance `utilisateur` ↔ `mot de passe` pour les authentifications non-Kerberos. Ce cache n'a rien à voir avec `krbtgt` : il concerne les mots de passe des comptes *utilisateur*, jamais la clé `krbtgt`. Sa présence ne remet pas en cause l'efficacité de la rotation.

9 Dépannage

Symptôme	Diagnostic et correction
`repadmin /syncall /AdeP` renvoie `Access Denied` sur `CN=NTDS Settings`	L'erreur n'est pas liée à la rotation mais à des ACL RPC particulières. Tant que `repadmin /replsummary` montre 0 échec, la réplication passe par les canaux normaux. Voir MO-AD-005 pour diagnostic approfondi.
Utilisateurs déconnectés massivement après Phase 2	Phase 2 lancée trop tôt (moins de 10 h après Phase 1). Les TGTs

10 Automatisation : script `T21_rotation_krbtgt.ps1`

Un script PowerShell paramétré est déjà disponible dans le dépôt :

```
audit/remediation/sprint2/T21_rotation_krbtgt.ps1
```

Son usage :

```
# Etat courant, sans modification
.\T21_rotation_krbtgt.ps1 -Phase Check

# Premiere rotation (demande confirmation 'oui')
.\T21_rotation_krbtgt.ps1 -Phase Phase1

# Seconde rotation (garde-fou < 10h)
.\T21_rotation_krbtgt.ps1 -Phase Phase2
```

Une planification semestrielle automatique peut être posée via `Register-ScheduledTask`, en s'assurant **impérativement** que les deux phases soient espacées de plus de 12 heures :

```
$trig1 = New-ScheduledTaskTrigger -Once -At "2026-10-14T13:00:00"
$act1  = New-ScheduledTaskAction -Execute "powershell.exe" `
    -Argument "-File C:\Scripts\T21_rotation_krbtgt.ps1 -Phase Phase1"
Register-ScheduledTask -TaskName "krbtgt-phase1" -Trigger $trig1 -Action $act1
    `
    -RunLevel Highest -User "BTS\Administrateur"

$trig2 = New-ScheduledTaskTrigger -Once -At "2026-10-15T13:00:00"
$act2  = New-ScheduledTaskAction -Execute "powershell.exe" `
    -Argument "-File C:\Scripts\T21_rotation_krbtgt.ps1 -Phase Phase2"
Register-ScheduledTask -TaskName "krbtgt-phase2" -Trigger $trig2 -Action $act2
    `
    -RunLevel Highest -User "BTS\Administrateur"
```

⚠ Automatisation complete deconseillee

Planifier Phase 2 sans vérification humaine revient à accepter que toute erreur opérationnelle (panne DC2, perte de connectivité entre DCs, incident de réplication) se propage directement à une rotation consécutive. Une exécution `-Phase Phase2` manuelle reste préférable, avec le garde-fou intégré qui vérifie l'intervalle.

11 Rollback

Il n'existe **pas de rollback simple** pour une rotation `krbtgt`. Une fois le nouveau mot de passe posé, l'ancien hash n'est plus reconstructible (la génération était aléatoire et le mot de passe clair jamais conservé). Les solutions en cas d'incident majeur :

— **Attendre** l'expiration naturelle des tickets actifs ($\leq$ 10 heures) si des sessions utilisateurs sont cassées ;

— **Restauration authoritative** de l'objet `krbtgt` depuis le backup *System State* (MO-AD-006) : opération lourde, nécessitant le démarrage d'un DC en *Directory Services Restore Mode* (DSRM). Réserver aux situations où la rotation a littéralement cassé l'authentification sur tout le domaine.

Dans la pratique, le pire scénario observable à la rotation est la déconnexion temporaire des sessions, résolue par un simple `klist purge` et une nouvelle ouverture de session.

12 Voir aussi

— **MO-AD-005** — Vérification de la santé AD (prérequis)

— **MO-AD-006** — Sauvegarde *System State* des DCs (prérequis)

— **MO-AD-007** — Audit périodique des comptes et groupes AD (contrôle que `krbtgt` est bien tourné régulièrement)

— **MO-AD-008** — Rotation périodique des mots de passe critiques (complémentaire : comptes administrateur classiques)

— PingCastle, règle *A-Krbtgt* #37 (*krbtgt password last set*)

— MITRE ATT&CK `T1558.001` (*Forge Kerberos Tickets : Golden Ticket*)

— Microsoft Docs « *Reset the krbtgt account password* » (article `KB2549833`)