AYI NEDJIMI Consultants

# Tiering Model Active Directory : Segmentation des : Guide

Catégorie : Attaques Active Directory  Lecture : 5 min  Publié le : 08/03/2026  Auteur : Ayi NEDJIMI

*Guide complet du Tiering Model Active Directory : architecture Tier 0/1/2, Enterprise Access Model, PAW, jump servers, authentication policy silos.*

Cet article suppose une connaissance solide d'Active Directory (GPO, OU, groupes de sécurité, délégation). Pour les techniques d'attaque référencées, consultez nos articles sur l'escalade de privilèges Windows, le post-exploitation et pivoting, et les attaques par mots de passe. Guide complet du Tiering Model Active Directory : architecture Tier 0/1/2, Enterprise Access Model, PAW, jump servers, authentication policy silos. Active Directory reste la cible privilégiée des attaquants en environnement Windows. Comprendre tiering model ad segmentation privileges est indispensable pour les équipes offensives comme défensives. Nous abordons notamment : 9. validation du tiering : tests et audit, questions frequentes et 10. conclusion : le tiering comme fondation de la sécurité ad. 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.

Tiering Model Active Directory TIER 0 - Contrôle de l'identité Compromission = perte totale du domaine Domain Controllers AD CS / PKI AD FS Entra Connect PAW Tier 0 TIER 1 - Serveurs et Applications Accès aux données métier critiques Serveurs Fichiers / BDD Exchange SCCM/MECM Hyperviseurs VMware / HyperV Jump Servers PAW Tier 1 TIER 2 - Postes de Travail et Utilisateurs Point d'entrée le plus fréquent des attaques Postes de travail Laptops Mobiles Imprimantes IoT Helpdesk Admin Tier 2 X INTERDIT X INTERDIT Tier 0 = Identité (Crown Jewels) Tier 1 = Serveurs (Data) Tier 2 = Endpoints (Entry Point)
Une compromission d'un seul poste de travail pourrait-elle mener à votre contrôleur de domaine ?

L'avantage des silos par rapport aux GPO : ils opèrent au niveau **Kerberos**. Même si un attaquant contourne les GPO (par exemple en modifiant la registry locale), le KDC refusera d'émettre un TGT pour un compte du silo si la machine n'en fait pas partie. C'est une protection beaucoup plus robuste.

## 6.3 Credential Guard et isolation des credentials

En complément du tiering, **Windows Credential Guard** utilise la virtualisation matérielle (VBS - Virtualization Based Security) pour isoler les credentials NTLM et Kerberos dans un processus protégé inaccessible depuis le kernel Windows. Même avec des droits SYSTEM, un attaquant ne peut pas extraire les hashes NTLM ou les tickets Kerberos des sessions actives :

```
# Activation de Credential Guard via GPO
# Computer Configuration > Administrative Templates > System > Device Guard
# > Turn on Virtualization Based Security : Enabled
#   - Credential Guard Configuration : Enabled with UEFI lock
#   - Secure Launch Configuration : Enabled

# Vérification du statut
Get-CimInstance -ClassName Win32_DeviceGuard -Namespace root/Microsoft/Windows/DeviceGuard
|
  Select-Object VirtualizationBasedSecurityStatus, SecurityServicesRunning

# Statut attendu :
# VirtualizationBasedSecurityStatus : 2 (Running)
# SecurityServicesRunning : {1, 2} (Credential Guard + HVCI)
```

Credential Guard ne protège pas contre toutes les attaques de credentials : les **Kerberoasting** et **AS-REP roasting** ciblent les tickets chiffrés hors du processus protégé. Mais combiné avec le tiering, il élimine le vecteur le plus courant : le vol de credentials en mémoire via Mimikatz sur une machine compromise.

**Limitation de Credential Guard**
Credential Guard ne protège pas les credentials stockés par des applications tierces (navigateurs, clients VPN, gestionnaires de mots de passe). Il protège uniquement les secrets gérés par le SSP (Security Support Provider) Windows : NTLM hashes, Kerberos TGT et tickets de service. Les credentials sauvegardés dans le Credential Manager Windows ne sont pas non plus protégés par Credential Guard.

La troisième phase déploie les **Privileged Access Workstations** et configure les **Authentication Policy Silos** pour les comptes Tier 0. C'est la phase qui verrouille définitivement le modèle :

```
# Phase 3 - Déploiement des PAW et configuration des silos

# 1. Vérification des prérequis pour les Authentication Policy Silos
$DFL = (Get-ADDomain).DomainMode
if ($DFL -lt "Windows2012R2Domain") {
    Write-Warning "DFL $DFL insuffisant ! Les silos nécessitent DFL 2012 R2 minimum."
    Write-Warning "Procédure : Mettre à jour tous les DC, puis élever le DFL."
    exit 1
}

# 2. Vérifier que Kerberos armoring (FAST) est activé
# GPO : Computer Configuration > Policies > Administrative Templates
#   > System > KDC > KDC support for claims, compound authentication and Kerberos armoring
# Doit être "Always provide claims" sur les DC

# 3. Créer les silos pour chaque tier
$Tiers = @(
    @{Name="Tier0-Silo"; TGT=240; Desc="Silo Tier 0 - DC, ADFS, PKI, PAW-T0"}
    @{Name="Tier1-Silo"; TGT=480; Desc="Silo Tier 1 - Serveurs membres, PAW-T1"}
)

foreach ($Tier in $Tiers) {
    # Politique d'authentification
    New-ADAuthenticationPolicy -Name "$($Tier.Name)-Policy" `
      -Description $Tier.Desc `
      -UserTGTLifetimeMins $Tier.TGT `
      -Enforce `
      -ProtectedFromAccidentalDeletion $true

    # Silo
    New-ADAuthenticationPolicySilo -Name $Tier.Name `
      -Description $Tier.Desc `
      -UserAuthenticationPolicy "$($Tier.Name)-Policy" `
      -ComputerAuthenticationPolicy "$($Tier.Name)-Policy" `
      -ServiceAuthenticationPolicy "$($Tier.Name)-Policy" `
      -Enforce `
      -ProtectedFromAccidentalDeletion $true
}

# 4. Assigner les comptes et machines aux silos
# Tier 0
Get-ADGroupMember -Identity "Tier0-Admins" -Recursive | ForEach-Object {
    Set-ADUser -Identity $_.SamAccountName -AuthenticationPolicySilo "Tier0-Silo"
}
Get-ADDomainController -Filter * | ForEach-Object {
    Set-ADComputer -Identity $_.Name -AuthenticationPolicySilo "Tier0-Silo"
}

# 5. Configurer les conditions d'accès FAST
Set-ADAuthenticationPolicy -Identity "Tier0-Silo-Policy" `
  -UserAllowedToAuthenticateFrom `
  "O:SYG:SYD:(XA;OICI;CR;;;WD;(@USER.ad://ext/AuthenticationSilo == `"Tier0-Silo`"))"

Write-Host "[+] Phase 3 terminée. Tester avec :" -ForegroundColor Green
Write-Host "    klist  # depuis une PAW Tier 0 avec un compte T0" -ForegroundColor Cyan
Write-Host "    # Le TGT doit montrer le silo d'authentification" -ForegroundColor Cyan
```

## 7.4 Phase 4 : Monitoring et audit continu (Semaines 17+)

Le tiering sans monitoring est un château de cartes. La phase 4 met en place la **détection des violations du modèle** -- un administrateur qui contourne les restrictions, un nouveau chemin d'attaque créé par une modification d'ACL, ou une dérive de la configuration des silos :

```
# Phase 4 - Monitoring des violations de tiering

# Event IDs critiques pour le monitoring du tiering
$TieringEvents = @{
    # Violations de connexion
    4625 = "Échec de connexion (tentative de connexion cross-tier)"
    4768 = "TGT demandé (vérifier le silo et la machine source)"
    4769 = "Ticket de service demandé (vérifier la machine cible)"
    4771 = "Pré-authentification Kerberos échouée"

    # Violations Authentication Policy Silo
    4820 = "Un TGT Kerberos a été refusé car le compte n'est pas dans le silo"
    4821 = "Un ticket de service Kerberos a été refusé (violation de silo)"

    # Modifications de la configuration du tiering
    5136 = "Modification d'un objet AD (ACL, membership, silo)"
    4728 = "Membre ajouté à un groupe global (Tier0/1/2-Admins)"
    4729 = "Membre retiré d'un groupe global"
}

# Requête PowerShell pour détecter les connexions cross-tier
# Connexions Tier 0 sur des machines non-Tier 0
$Tier0Users = (Get-ADGroupMember -Identity "Tier0-Admins" -Recursive).SamAccountName
$DCs = (Get-ADDomainController -Filter *).Name

$Events = Get-WinEvent -FilterHashtable @{
    LogName   = 'Security'
    Id        = 4624  # Logon réussi
    StartTime = (Get-Date).AddDays(-1)
} | Where-Object {
    $xml = [xml]$_.ToXml()
    $TargetUser = $xml.Event.EventData.Data | Where-Object {$_.Name -eq 'TargetUserName'}
| Select-Object -ExpandProperty '#text'
    $Workstation = $xml.Event.EventData.Data | Where-Object {$_.Name -eq
'WorkstationName'} | Select-Object -ExpandProperty '#text'

    ($TargetUser -in $Tier0Users) -and ($Workstation -notin $DCs) -and ($Workstation
-notlike "PAW-T0*")
}

if ($Events) {
    Write-Warning "ALERTE : $($Events.Count) connexions Tier 0 détectées sur des machines
non-autorisées !"
    $Events | ForEach-Object {
        $xml = [xml]$_.ToXml()
        [PSCustomObject]@{
            Time = $_.TimeCreated
            User = ($xml.Event.EventData.Data | Where-Object {$_.Name -eq
'TargetUserName'}).'#text'
            Machine = ($xml.Event.EventData.Data | Where-Object {$_.Name -eq
'WorkstationName'}).'#text'
            LogonType = ($xml.Event.EventData.Data | Where-Object {$_.Name -eq
'LogonType'}).'#text'
        }
    } | Format-Table -AutoSize
}
```

Le tiering génère inévitablement des demandes d'exceptions : "J'ai besoin de me connecter en RDP au DC depuis mon poste pour dépanner rapidement." Ces exceptions, si elles deviennent permanentes, créent des **chemins d'attaque** que BloodHound identifie immédiatement. Chaque exception doit être :

- **Temporaire** : avec une date d'expiration automatique (max 24h)
- **Documentée** : ticket de changement avec justification et approbation
- **Auditée** : logs centralisés de toutes les connexions effectuées pendant l'exception
- **Revue régulièrement** : revue trimestrielle de toutes les exceptions actives

## 8.4 Erreur n°4 : ignorer les chemins d'attaque indirects

Le tiering protège contre les attaques directes (connexion cross-tier), mais les attaquants exploitent des **chemins indirects** : délégation Kerberos non contrainte, ACL permissives sur les OU, GPO modifiables par des comptes Tier 1 mais liées aux DC, partages réseau accessibles depuis tous les tiers contenant des scripts avec des credentials hardcodés. L'analyse régulière avec BloodHound est indispensable pour détecter ces chemins.

**Piège critique : la délégation Kerberos non contrainte**
Un serveur Tier 1 configuré en **délégation Kerberos non contrainte (unconstrained delegation)** peut capturer le TGT de n'importe quel utilisateur qui s'y authentifie, y compris les comptes Tier 0 via des connexions de service. Un attaquant qui compromet ce serveur peut ensuite rejouer le TGT du compte Tier 0 pour accéder aux contrôleurs de domaine. La remédiation consiste à migrer vers la **délégation contrainte basée sur les ressources (RBCD)** et à n'autoriser la délégation non contrainte que sur les contrôleurs de domaine.

## 9. Validation du tiering : tests et audit

Un modèle de tiering ne vaut que s'il est **testé et validé régulièrement**. Voici les tests essentiels à réaliser pour vérifier l'intégrité du cloisonnement :

```
# Script de validation du tiering - à exécuter mensuellement

function Test-TieringCompliance {
    [CmdletBinding()]
    param()

    $Results = @()
    $BaseDN = (Get-ADDomain).DistinguishedName

    # Test 1 : Vérifier qu'aucun compte Tier 0 n'a de SPN (Kerberoasting)
    Write-Host "[*] Test 1 : Comptes Tier 0 avec SPN..." -ForegroundColor Yellow
    $T0WithSPN = Get-ADGroupMember -Identity "Tier0-Admins" -Recursive |
        Get-ADUser -Properties ServicePrincipalName |
        Where-Object { $_.ServicePrincipalName.Count -gt 0 }

    if ($T0WithSPN) {
        $Results += [PSCustomObject]@{
            Test = "T0-SPN"; Status = "FAIL"; Count = $T0WithSPN.Count
            Details = "Comptes Tier 0 avec SPN : $($T0WithSPN.SamAccountName -join ', ')"
        }
    } else {
        $Results += [PSCustomObject]@{Test="T0-SPN"; Status="PASS"; Count=0; Details="OK"}
    }

    # Test 2 : Vérifier qu'aucun compte Tier 0 n'a de session ouverte hors DC/PAW
    Write-Host "[*] Test 2 : Sessions Tier 0 actives..." -ForegroundColor Yellow
    $DCs = (Get-ADDomainController -Filter *).HostName
    # Utiliser PSLoggedOn ou quser sur chaque serveur pour détecter les sessions cross-
tier

    # Test 3 : Vérifier les ACL sur les OU Tier 0
    Write-Host "[*] Test 3 : ACL sur les OU Tier 0..." -ForegroundColor Yellow
    $T0OU = "OU=Domain Controllers,$BaseDN"
    $ACL = Get-Acl "AD:\$T0OU"
    $DangerousACE = $ACL.Access | Where-Object {
        $_.IdentityReference -notmatch "S-1-5-32-544|Domain Admins|Enterprise Admins|
SYSTEM" -and
        $_.ActiveDirectoryRights -match "WriteProperty|WriteDacl|WriteOwner|GenericAll|
GenericWrite"
    }

    if ($DangerousACE) {
        $Results += [PSCustomObject]@{
            Test = "T0-ACL"; Status = "FAIL"; Count = $DangerousACE.Count
            Details = "ACE dangereuses : $($DangerousACE.IdentityReference -join ', ')"
        }
    } else {
        $Results += [PSCustomObject]@{Test="T0-ACL"; Status="PASS"; Count=0; Details="OK"}
    }

    # Test 4 : Vérifier les silos d'authentification
    Write-Host "[*] Test 4 : Configuration des silos..." -ForegroundColor Yellow
    $Silos = Get-ADAuthenticationPolicySilo -Filter *
    $UnenforcesSilos = $Silos | Where-Object { -not $_.Enforce }

    if ($UnenforcesSilos) {
        $Results += [PSCustomObject]@{
            Test = "SILOS"; Status = "WARN"; Count = $UnenforcesSilos.Count
            Details = "Silos non enforced : $($UnenforcesSilos.Name -join ', ')"
        }
    } else {
        $Results += [PSCustomObject]@{Test="SILOS"; Status="PASS"; Count=0; Details="OK"}
```

```
    }

    # Test 5 : Vérifier l'absence de délégation non contrainte hors DC
    Write-Host "[*] Test 5 : Délégation Kerberos non contrainte..." -ForegroundColor
Yellow
    $UnconstrainedDeleg = Get-ADComputer -Filter {TrustedForDelegation -eq $true}
-Properties TrustedForDelegation |
        Where-Object { $_.DistinguishedName -notmatch "Domain Controllers" }

    if ($UnconstrainedDeleg) {
        $Results += [PSCustomObject]@{
            Test = "DELEG"; Status = "FAIL"; Count = $UnconstrainedDeleg.Count
            Details = "Machines avec délégation non contrainte hors DC : $
($UnconstrainedDeleg.Name -join ', ')"
        }
    } else {
        $Results += [PSCustomObject]@{Test="DELEG"; Status="PASS"; Count=0; Details="OK"}
    }

    # Affichage des résultats
    Write-Host "`n========== RÉSULTATS AUDIT TIERING ==========" -ForegroundColor Cyan
    $Results | Format-Table -AutoSize

    $FailCount = ($Results | Where-Object {$_.Status -eq "FAIL"}).Count
    if ($FailCount -gt 0) {
        Write-Host "[!] $FailCount test(s) en échec -- actions correctives nécessaires"
-ForegroundColor Red
    } else {
        Write-Host "[+] Tous les tests passent -- tiering conforme" -ForegroundColor Green
    }

    return $Results
}

# Exécution
Test-TieringCompliance
```

Pour approfondir ce sujet, consultez notre outil open-source kerberos-toolkit qui facilite l'analyse et le test des mécanismes Kerberos.

## Questions frequentes

### Comment mettre en place Tiering Model Active Directory dans un environnement de production ?

La mise en place de Tiering Model Active Directory en production necessite une planification rigoureuse, incluant l'evaluation des prerequis techniques, la definition d'une architecture cible, des tests de validation approfondis et un plan de deploiement progressif avec des points de controle a chaque etape.

### Pourquoi Tiering Model Active Directory est-il essentiel pour la securite des systemes d'information ?

Tiering Model Active Directory constitue un element fondamental de la securite des systemes d'information car il permet de reduire significativement la surface d'attaque, d'ameliorer la detection des menaces et de renforcer la posture globale de securite de l'organisation face aux cybermenaces actuelles.

### Comment détecter rapidement une attaque de type Tiering Model Active Directory : Segmentation ?

Surveillez les événements Windows 4662, 4624 type 3 et 4672 via votre SIEM. Corrélez-les avec des connexions inhabituelles vers les contrôleurs de domaine en dehors des heures de travail.

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

> Points clés à retenir
>
> - 9. Validation du tiering : tests et audit
> - Questions frequentes
> - 10. Conclusion : le tiering comme fondation de la sécurité AD

## 10. Conclusion : le tiering comme fondation de la sécurité AD

Le **Tiering Model** reste, en 2025, le pilier fondamental de toute stratégie de sécurisation d'Active Directory. Malgré l'évolution vers l'Enterprise Access Model de Microsoft et l'intégration cloud avec Entra ID, les principes fondamentaux demeurent inchangés : **séparer les niveaux de privilèges, isoler les identités critiques et interdire toute contamination entre les tiers**.

Les organisations qui implémentent correctement le tiering observent une réduction drastique de leur surface d'attaque AD. Les chemins d'attaque identifiés par BloodHound diminuent de 80 à 95 % après une implémentation complète, et les incidents de type compromission totale du domaine (Golden Ticket, DCSync) deviennent quasiment impossibles sans exploitation de vulnérabilités zero-day.

Les clés du succès sont :

- **Approche progressive** : déployer par phases avec des quick wins immédiats
- **Comptes dédiés par tier** : pas de compromis sur la séparation des identités
- **PAW pour le Tier 0** : l'investissement le plus rentable en sécurité AD
- **Authentication Policy Silos** : protection au niveau Kerberos, plus robuste que les GPO seules
- **Monitoring continu** : détecter et corriger les violations du modèle en temps réel
- **Soutien de la direction** : le tiering impacte les habitudes de travail et nécessite un mandat clair

Enfin, le tiering ne doit pas être vu comme un projet ponctuel mais comme un **processus continu**. Chaque nouvelle application, chaque nouveau compte de service, chaque modification d'infrastructure doit être évaluée à travers le prisme du tiering. Combiné avec les protections techniques comme LAPS, le hardening via GPO et la protection des secrets NTDS.dit, le tiering constitue la fondation sur laquelle toute la sécurité AD repose.

**En résumé :** Le tiering n'est pas optionnel. Toute organisation utilisant Active Directory sans modèle de tiering offre aux attaquants un chemin direct du poste de travail compromis au contrôleur de domaine -- et cela prend en moyenne 48 heures dans un environnement non segmenté.

---

**Ayi NEDJIMI Consultants** — Expert cybersécurité offensive & intelligence artificielle

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