Audit des Permissions DNS dans Active Directory : Guide Complet 2026

Auditez les permissions DNS dans Active Directory : scripts PowerShell, détection ADIDNS Hijack, durcissement et conformité NIS 2. Guide complet 2026.

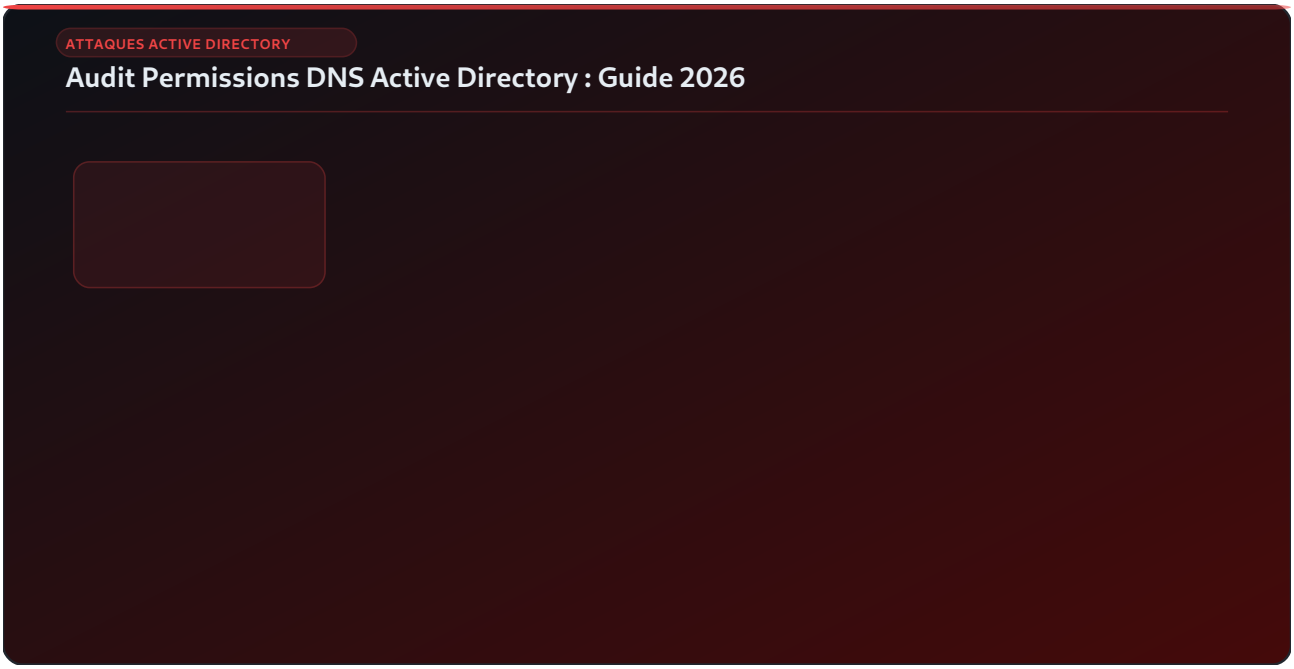
📋 EN BREF

En bref

- Les zones DNS intégrées à Active Directory héritent d'ACL trop permissives : tout utilisateur authentifié peut créer des enregistrements par défaut.
- L'attaque ADIDNS Hijack (wildcard DNS + MITM) est réalisable avec Powermad en quelques secondes sur un domaine non durci.
- Le groupe DNSAdmins permet une escalade de privilèges vers SYSTEM via le chargement de DLL arbitraires dans dns.exe.
- Les Event IDs 292, 770, 771 (canal DNSServer/Audit) et 4662/5136 (Security) couvrent la détection des modifications DNS malveillantes.
- NIS 2 (Art. 21§2.d) et le guide ANSSI DNS 2022 imposent l'audit et le durcissement des permissions DNS dans les organisations françaises.

Dans les infrastructures Active Directory des entreprises françaises, le service DNS est omniprésent mais rarement audité avec la rigueur qu'il mérite. Pourtant, à Paris comme dans toute organisation utilisant Windows Server, les zones DNS intégrées à l'annuaire constituent un vecteur d'attaque de premier plan, exploitable par n'importe quel compte utilisateur du domaine sans nécessiter de privilèges particuliers. Les RSSI confrontés aux exigences de NIS 2 et aux

recommandations de l'ANSSI doivent impérativement intégrer l'audit des permissions DNS dans leur programme de sécurité Active Directory. Ce guide complet couvre l'architecture des permissions DNS dans AD, les techniques d'attaque ADIDNS Hijack utilisées par les red teams, les scripts PowerShell d'audit et de durcissement, ainsi que la mise en place d'un monitoring continu avec Wazuh et les règles Sigma adaptées aux infrastructures françaises. L'objectif est de transformer un angle mort sécuritaire récurrent en un contrôle robuste et auditable, conformément aux standards réglementaires en vigueur en 2026. Ce guide s'adresse aux administrateurs système, architectes sécurité et équipes SOC qui administrent des environnements Active Directory en production et cherchent à éliminer une classe entière de vulnérabilités souvent négligées lors des audits de sécurité traditionnels.



Le DNS est le système nerveux central de tout environnement Active Directory. Sans une résolution DNS fonctionnelle, l'authentification Kerberos échoue, les GPO ne s'appliquent pas et les services d'annuaire deviennent inaccessibles. Cette criticité opérationnelle a conduit Microsoft à intégrer les zones DNS directement dans la base de données NTDS.dit via les **zones**

DNS intégrées à l'Active Directory (AD-integrated DNS zones), offrant réplication automatique et haute disponibilité. Mais cette intégration a un coût sécuritaire considérable.

Les zones AD-intégrées stockent leurs enregistrements comme objets LDAP dans deux partitions distinctes : `DomainDNSZones` (répliquée aux DC du domaine) et `ForestDNSZones` (répliquée à tous les DC de la forêt). Par défaut, les ACL de ces partitions accordent au groupe *Authenticated Users* le droit `CreateChild`, ce qui signifie que tout compte du domaine — y compris un compte de service compromis — peut créer des enregistrements DNS arbitraires.

Les trois familles d'attaques DNS dans AD

ADIDNS Hijack : Un attaquant ayant un pied dans le domaine utilise cet accès pour créer un enregistrement DNS pointant vers sa machine. Si une application cliente cherche à résoudre un nom qui n'existe pas encore (ou dont l'enregistrement a expiré), elle sera redirigée vers l'attaquant. Couplé à Responder ou Inveigh, cela permet la capture de hachages NTLMv2 sans interaction réseau particulière.

DNS Wildcard Abuse : La technique la plus agressive consiste à créer un enregistrement wildcard (`*`) dans la zone DNS du domaine. Tout nom d'hôte non résolu retournera l'IP de l'attaquant. Kevin Robertson, chercheur en sécurité, a documenté cette technique en 2018 et développé l'outil Powermad pour l'automatiser. L'impact est massif : tous les NTLM challenges des machines du réseau cherchant à résoudre des noms inexistants transitent par l'attaquant.

DNSAdmins Privilege Escalation : Le groupe DNSAdmins, conçu pour déléguer l'administration DNS sans droits Domain Admin, permet de configurer un plugin DLL pour le service DNS via `dnscmd /config /serverlevelplugindll`. Comme dns.exe s'exécute en tant que SYSTEM sur les contrôleurs de domaine, charger une DLL malveillante équivaut à une exécution de code avec les privilèges les plus élevés du système. Cette technique, identifiée comme CVE-2021-40469 dans sa variante serveur distant, est répertoriée sous MITRE ATT&CK T1543.003.

L'outil `dnstool.py` de la suite Impacket (disponible dans Kali Linux) automatise la création d'enregistrements ADIDNS depuis Linux, rendant ces attaques accessibles sans nécessiter un poste Windows dans le domaine cible — uniquement des identifiants valides.

Architecture des Permissions DNS dans Active Directory

Pour auditer efficacement les permissions DNS, il faut comprendre leur structure dans l'annuaire. Les zones DNS intégrées ne sont pas stockées dans la base de données DNS classique (`%SystemRoot%\System32\dns`) mais directement comme objets LDAP.

> ⚡ **Retour terrain**
>
> J'utilise BloodHound systématiquement en début de mission AD. Pour un groupe de distribution retail, la cartographie a révélé un chemin de 4 sauts depuis un compte utilisateur standard vers Domain Admin — via un GPO modifiable par le compte, appliqué à une OU contenant un admin, avec une ACL WriteOwner sur un groupe privilégié. Aucun des quatre liens n'était visible sans la vue graphique. La correction a pris 2 heures une fois le chemin identifié.

Partitions DNS et leur portée

La partition `DomainDNSZones` (`DC=DomainDnsZones,DC=corp,DC=lan`) est répliquée uniquement aux contrôleurs de domaine du domaine local. Elle contient les zones DNS du domaine, notamment la zone principale `corp.lan`. La partition `ForestDNSZones` (`DC=ForestDnsZones,DC=corp,DC=lan`) est répliquée à **tous** les DC de la forêt entière, ce qui inclut les zones `_msdcs` critiques pour Kerberos et la localisation des DC.

Les enregistrements DNS individuels sont stockés sous `CN=MicrosoftDNS,DC=DomainDnsZones,DC=corp,DC=lan` puis sous le DN de la zone concernée. Chaque enregistrement est un objet de classe `dnsNode` avec des attributs comme `dnsRecord` (format binaire propriétaire Microsoft).

ACL par défaut et leur problème

Les permissions par défaut sur l'objet `DomainDNSZones` incluent :

Permission DNS	Groupe par défaut	Risque	Recommandation
CreateChild (dnsNode)	Authenticated Users	CRITIQUE — ADIDNS Hijack possible	Retirer immédiatement
GenericAll	Domain Admins	Faible — attendu	Conserver, auditer les membres
GenericAll	Enterprise Admins	Faible — attendu	Conserver
GenericAll	DNSAdmins	ÉLEVÉ — DLL injection possible	Restreindre les membres, activer audit
GenericRead + CreateChild	Domain Computers	MOYEN — machines compromises	Retirer CreateChild si non nécessaire
WriteProperty (spécifique)	SYSTEM	Faible — requis pour le service	Conserver

La colonne "Risque CRITIQUE" pour *Authenticated Users: CreateChild* résume à elle seule pourquoi l'audit des permissions DNS AD doit figurer dans tout pentest ou audit sécurité de l'annuaire. Pour approfondir la sécurité globale de l'annuaire, consultez notre guide sur l'audit des groupes privilégiés Active Directory.

Audit des ACL DNS avec PowerShell

L'audit des permissions DNS nécessite une combinaison de cmdlets PowerShell natifs et d'accès LDAP direct. Voici le script complet d'audit qui génère un rapport HTML exploitable par un RSSI ou une équipe d'audit.

Commandes de découverte initiale

```
 # Lister toutes les zones DNS du domaine
Get-DnsServerZone -ComputerName dc1.corp.lan |
    Select-Object ZoneName, ZoneType, IsADIntegrated, ReplicationScope |
    Format-Table -AutoSize

# Examiner les ACL d'une zone spécifique via LDAP
$zoneDN = "DC=corp.lan,CN=MicrosoftDNS,DC=DomainDnsZones,DC=corp,DC=lan"
$acl = Get-Acl "AD:\$zoneDN"
$acl.Access | Select-Object IdentityReference, ActiveDirectoryRights, AccessControlType |
    Sort-Object IdentityReference | Format-Table -AutoSize

# Vérifier les permissions sur DomainDNSZones (niveau partitition)
$partitionDN = "DC=DomainDnsZones,DC=corp,DC=lan"
(Get-Acl "AD:\$partitionDN").Access |
    Where-Object { $_.AccessControlType -eq "Allow" } |
    Select-Object IdentityReference, ActiveDirectoryRights |
    Format-Table -AutoSize

# Énumération via dnscmd (natif Windows Server)
dnscmd /enumzones
dnscmd dc1.corp.lan /zoneinfo corp.lan
```

Script complet d'audit avec rapport HTML

```
 #Requires -Modules ActiveDirectory, DnsServer
<#
.SYNOPSIS
    Audit complet des permissions DNS dans Active Directory
.DESCRIPTION
    Analyse les ACL des zones DNS AD-intégrées et génère un rapport HTML
    Détecte les permissions à risque (ADIDNS Hijack, DNSAdmins abuse)
.NOTES
    Nécessite : RSAT-DNS-Server, RSAT-AD-Tools
    Auteur : Audit DNS AD - ayinedjimi-consultants.fr
    Version : 2026.1
#>

param(
    [string]$DomainController = $env:LOGONSERVER.TrimStart(''),
    [string]$Domain = $env:USERDNSDOMAIN,
    [string]$OutputPath = "C:\Audit\DNS-ACL-Report-$(Get-Date -Format 'yyyyMMdd').html"
)

# Convertir le domaine FQDN en DN LDAP
function ConvertTo-DistinguishedName {
    param([string]$DomainFQDN)
    return ($DomainFQDN.Split('.') | ForEach-Object { "DC=$_" }) -join ','
}

$domainDN = ConvertTo-DistinguishedName -DomainFQDN $Domain
$riskyPermissions = [System.Collections.Generic.List[PSObject]]::new()
$allPermissions   = [System.Collections.Generic.List[PSObject]]::new()

Write-Host "[*] Démarrage audit DNS AD pour : $Domain" -ForegroundColor Cyan

# Groupes et SIDs à risque
$riskyIdentities = @(
    "Authenticated Users",
    "Everyone",
    "Domain Computers",
    "Domain Users",
    "CREATOR OWNER"
)

$riskyRights = @(
    "CreateChild",
    "WriteProperty",
```

```
    "GenericWrite",
    "GenericAll",
    "WriteDacl",
    "WriteOwner"
)

# Partitions DNS à auditer
$dnsPartitions = @(
    "DC=DomainDnsZones,$domainDN",
    "DC=ForestDnsZones,$domainDN"
)

foreach ($partition in $dnsPartitions) {
    Write-Host "[*] Analyse partition : $partition" -ForegroundColor Yellow

    try {
        $partitionACL = Get-Acl "AD:\$partition" -ErrorAction Stop

        foreach ($ace in $partitionACL.Access) {
            $entry = [PSCustomObject]@{
                Partition          = $partition
                Identity           = $ace.IdentityReference.ToString()
                Rights             = $ace.ActiveDirectoryRights.ToString()
                AccessType         = $ace.AccessControlType.ToString()
                IsInherited        = $ace.IsInherited
                RiskLevel          = "INFO"
                Recommendation     = ""
            }

            # Évaluation du risque
            $isRiskyIdentity = $riskyIdentities | Where-Object { $ace.IdentityReference -match $_
            $hasRiskyRight   = $riskyRights | Where-Object { $ace.ActiveDirectoryRights -match $_

            if ($isRiskyIdentity -and $hasRiskyRight -and $ace.AccessControlType -eq "Allow") {
                if ($ace.ActiveDirectoryRights -match "CreateChild") {
                    $entry.RiskLevel       = "CRITIQUE"
                    $entry.Recommendation  = "Retirer CreateChild - ADIDNS Hijack possible"
                } elseif ($ace.ActiveDirectoryRights -match "GenericAll|WriteDacl|WriteOwner") {
                    $entry.RiskLevel       = "ÉLEVÉ"
                    $entry.Recommendation  = "Restreindre - contrôle total non justifié"
                } else {
                    $entry.RiskLevel       = "MOYEN"
                    $entry.Recommendation  = "Examiner la nécessité de cette permission"
                }
```

```
                $riskyPermissions.Add($entry)
            }
            $allPermissions.Add($entry)
        }
    } catch {
        Write-Warning "Impossible d'accéder à $partition : $_"
    }
}

# Audit des zones DNS individuelles
Write-Host "[*] Audit des zones DNS intégrées..." -ForegroundColor Yellow
$zones = Get-DnsServerZone -ComputerName $DomainController |
         Where-Object { $_.IsADIntegrated -eq $true }

$zoneAudit = [System.Collections.Generic.List[PSObject]]::new()

foreach ($zone in $zones) {
    $zoneDN = "DC=$($zone.ZoneName),CN=MicrosoftDNS,DC=DomainDnsZones,$domainDN"
    try {
        $zoneACL = Get-Acl "AD:\$zoneDN" -ErrorAction Stop
        $createChildCount = ($zoneACL.Access | Where-Object {
            $_.ActiveDirectoryRights -match "CreateChild" -and
            $_.AccessControlType -eq "Allow"
        }).Count

        $zoneAudit.Add([PSCustomObject]@{
            ZoneName           = $zone.ZoneName
            ReplicationScope   = $zone.ReplicationScope
            ACECount           = $zoneACL.Access.Count
            CreateChildEntries = $createChildCount
            HasWildcard        = (Get-DnsServerResourceRecord -ZoneName $zone.ZoneName `
                                    -ComputerName $DomainController -ErrorAction SilentlyContinue |
                                    Where-Object { $_.HostName -eq "*" }).Count -gt 0
        })
    } catch {
        # Zone inaccessible via LDAP (zone standard, non AD-intégrée)
    }
}

# Vérification du groupe DNSAdmins
Write-Host "[*] Vérification groupe DNSAdmins..." -ForegroundColor Yellow
$dnsAdminsMembers = Get-ADGroupMember -Identity "DNSAdmins" -Recursive -ErrorAction SilentlyConti

# Génération rapport HTML
```

```
$htmlContent = @"
<!DOCTYPE html>
<html lang="fr"><head>
<meta charset="UTF-8">
<title>Rapport Audit DNS AD - $(Get-Date -Format 'dd/MM/yyyy')</title>
<style>
  body { font-family: Segoe UI, sans-serif; margin: 20px; background: #f5f5f5; }
  h1 { color: #c0392b; }
  h2 { color: #2c3e50; border-bottom: 2px solid #3498db; }
  table { border-collapse: collapse; width: 100%; margin-bottom: 20px; background: white; }
  th { background: #2c3e50; color: white; padding: 8px 12px; text-align: left; }
  td { padding: 8px 12px; border-bottom: 1px solid #ddd; }
  tr:hover { background: #f0f0f0; }
  .CRITIQUE { color: white; background: #c0392b; padding: 2px 8px; border-radius: 3px; }
  .ÉLEVÉ { color: white; background: #e67e22; padding: 2px 8px; border-radius: 3px; }
  .MOYEN { color: black; background: #f1c40f; padding: 2px 8px; border-radius: 3px; }
  .INFO { color: white; background: #27ae60; padding: 2px 8px; border-radius: 3px; }
  .summary { background: white; padding: 15px; border-left: 4px solid #c0392b; margin-bottom: 20p
</style>
</head><body>
<h1>Rapport Audit Permissions DNS Active Directory</h1>
<div class="summary">
  <strong>Domaine :</strong> $Domain |
  <strong>Date :</strong> $(Get-Date -Format 'dd/MM/yyyy HH:mm') |
  <strong>DC analysé :</strong> $DomainController<br/>
  <strong>Permissions à risque :</strong> $($riskyPermissions.Count) |
  <strong>Membres DNSAdmins :</strong> $($dnsAdminsMembers.Count)
</div>
<h2>Permissions à Risque Détectées</h2>
"@

$htmlContent += $riskyPermissions | ConvertTo-Html -Fragment |
    ForEach-Object { $_ -replace '<td>(CRITIQUE|ÉLEVÉ|MOYEN)</td>', '<td><span class="$1">$1</spa

$htmlContent += "<h2>Membres du groupe DNSAdmins</h2>"
$htmlContent += $dnsAdminsMembers | Select-Object Name, SamAccountName, ObjectClass |
    ConvertTo-Html -Fragment

$htmlContent += "<h2>Inventaire des Zones DNS AD-Intégrées</h2>"
$htmlContent += $zoneAudit | ConvertTo-Html -Fragment
$htmlContent += "</body></html>"

New-Item -ItemType Directory -Force -Path (Split-Path $OutputPath) | Out-Null
$htmlContent | Out-File -FilePath $OutputPath -Encoding UTF8
```

```
Write-Host "[+] Rapport généré : $OutputPath" -ForegroundColor Green
Write-Host "[!] Permissions CRITIQUES trouvées : $($riskyPermissions | Where-Object RiskLevel -eq

# Affichage console des findings critiques
if ($riskyPermissions.Count -gt 0) {
    Write-Host "`n[CRITIQUE] Permissions dangereuses détectées :" -ForegroundColor Red
    $riskyPermissions | Where-Object { $_.RiskLevel -eq "CRITIQUE" } |
        Format-Table Partition, Identity, Rights, Recommendation -AutoSize
}
```

Attaques ADIDNS — Comment les Détecter et les Prévenir

Comprendre les techniques offensives est indispensable pour configurer une détection efficace. Cette section adopte une perspective défensive : les techniques sont présentées pour permettre aux équipes SOC et aux administrateurs de comprendre ce qu'ils doivent détecter et bloquer.

ADIDNS Hijack avec Powermad

Powermad est un module PowerShell développé par Kevin Robertson qui exploite directement l'API LDAP pour créer des enregistrements DNS sans nécessiter de droits DNS administratifs. Voici comment un attaquant opèrerait — et ce que vous devez surveiller dans vos logs :

```
 # Ce que fait un attaquant (à détecter dans vos logs) :
# Import-Module Powermad
# Invoke-DNSUpdate -DNSName "serveur-facturation" -DNSData "192.168.1.99"
# Invoke-DNSUpdate -DNSName "*" -DNSData "192.168.1.99"  # wildcard = MITM massif

# Ce que vous verrez dans les Event Logs (canal Security) :
# Event 4662 : Object operation (DomainDNSZones)
# Event 5136 : Directory service object was modified
# Objet : CN=serveur-facturation,DC=corp.lan,...

# Vérifier l'existence d'un wildcard DNS (signe d'attaque en cours) :
Resolve-DnsName -Name "hote-qui-nexiste-pas.corp.lan" -Server dc1.corp.lan
# Si cette commande retourne une IP valide → wildcard DNS actif → incident en cours
```

L'outil `dnstool.py` d'Impacket permet la même opération depuis Linux :

```
 # Attaque depuis Linux (perspective défensive - ce qui génère les alertes)
# python3 dnstool.py -u 'corp\jdupont' -p 'Password123' -r 'evilhost' #   -d '192.168.1.99' --act

# Ce que vous voyez côté DNS (à monitorer) :
dnscmd dc1.corp.lan /enumrecords corp.lan @ /additional | grep -E "^\*|wildcard"
```

Inveigh : capture NTLM via ADIDNS Wildcard

Une fois le wildcard DNS créé, l'attaquant lance Inveigh pour capturer les authentifications NTLMv2. Depuis votre SOC, les indicateurs de compromission sont : un volume anormal de tentatives d'authentification NTLM vers une IP inconnue, des Event ID 4776 (validation NTLM) en échec massif, et des alertes Zeek/Suricata sur le trafic SMB vers des hôtes non reconnus. La mise en œuvre de LDAP Signing et Channel Binding réduit considérablement la surface d'attaque NTLM relay associée à ces techniques.

Détection des événements DNS suspects

Le canal d'audit `Microsoft-Windows-DNSServer/Audit` doit être activé explicitement. Il n'est pas activé par défaut sur Windows Server :

```
 # Activer le canal d'audit DNS sur tous les DC (via GPO ou commande directe)
wevtutil set-log "Microsoft-Windows-DNSServer/Audit" /enabled:true /quiet:true

# Vérifier l'activation
wevtutil get-log "Microsoft-Windows-DNSServer/Audit" | Select-String "enabled"

# Rechercher les créations d'enregistrements DNS (Event 292)
Get-WinEvent -LogName "Microsoft-Windows-DNSServer/Audit" -MaxEvents 1000 |
    Where-Object { $_.Id -eq 292 } |
    Select-Object TimeCreated,
        @{N="RecordName";  E={$_.Properties[0].Value}},
        @{N="RecordType";  E={$_.Properties[1].Value}},
        @{N="Zone";        E={$_.Properties[2].Value}},
        @{N="TTL";         E={$_.Properties[3].Value}} |
    Sort-Object TimeCreated -Descending

# Alerter sur création wildcard (signe d'attaque ADIDNS)
Get-WinEvent -LogName "Microsoft-Windows-DNSServer/Audit" |
    Where-Object { $_.Id -eq 292 -and $_.Message -match '\*' } |
    ForEach-Object { Write-Warning "ALERTE : Wildcard DNS créé à $($_.TimeCreated)" }
```

Pour une surveillance des contrôleurs de domaine plus complète, consultez notre guide sur les Windows Events à surveiller sur un contrôleur de domaine.

Durcissement des Permissions DNS Active Directory

Le durcissement DNS AD comporte deux axes majeurs : retirer les permissions excessives des objets LDAP DNS, et sécuriser le groupe DNSAdmins.

Script PowerShell de durcissement complet

```
 #Requires -Modules ActiveDirectory
#Requires -RunAsAdministrator
<#
.SYNOPSIS
    Durcissement des permissions DNS Active Directory
.DESCRIPTION
    - Retire CreateChild d'Authenticated Users sur DomainDNSZones
    - Supprime les wildcards DNS existants
    - Audit le groupe DNSAdmins
    - Configure les SACL d'audit sur les zones DNS
.NOTES
    Tester en environnement de LAB avant production
    Sauvegarder les ACL actuelles avant modification
#>

param(
    [switch]$WhatIf,
    [string]$Domain = $env:USERDNSDOMAIN
)

function ConvertTo-DN { param([string]$fqdn)
    return ($fqdn.Split('.') | ForEach-Object { "DC=$_" }) -join ',' }

$domainDN   = ConvertTo-DN $Domain
$partitions = @(
    "DC=DomainDnsZones,$domainDN",
    "DC=ForestDnsZones,$domainDN"
)

# ── ÉTAPE 1 : Sauvegarde des ACL actuelles ──────────────────────────
$backupPath = "C:\Audit\DNS-ACL-Backup-$(Get-Date -Format 'yyyyMMdd-HHmm').xml"
New-Item -ItemType Directory -Force -Path C:\Audit | Out-Null

$aclBackup = @{}
foreach ($partition in $partitions) {
    $aclBackup[$partition] = (Get-Acl "AD:\$partition").Sddl
}
$aclBackup | Export-Clixml -Path $backupPath
Write-Host "[+] ACL sauvegardées dans $backupPath" -ForegroundColor Green

# ── ÉTAPE 2 : Retirer CreateChild d'Authenticated Users ─────────────
$authenticatedUsersSID = [System.Security.Principal.SecurityIdentifier]::new(
    [System.Security.Principal.WellKnownSidType]::AuthenticatedUserSid, $null)
```

```
foreach ($partition in $partitions) {
    Write-Host "[*] Traitement : $partition" -ForegroundColor Yellow

    $adPath = "AD:\$partition"
    $acl    = Get-Acl $adPath

    $acesToRemove = $acl.Access | Where-Object {
        $_.IdentityReference.Translate([System.Security.Principal.SecurityIdentifier]) `
            -eq $authenticatedUsersSID -and
        $_.ActiveDirectoryRights -band [System.DirectoryServices.ActiveDirectoryRights]::CreateCh
        $_.AccessControlType -eq [System.Security.AccessControl.AccessControlType]::Allow
    }

    if ($acesToRemove) {
        foreach ($ace in $acesToRemove) {
            if (-not $WhatIf) {
                $acl.RemoveAccessRule($ace) | Out-Null
                Write-Host "  [-] Suppression : $($ace.IdentityReference) - $($ace.ActiveDirector
                    -ForegroundColor Red
            } else {
                Write-Host "  [WhatIf] Supprimerait : $($ace.IdentityReference) - $($ace.ActiveDi
            }
        }
        if (-not $WhatIf) {
            Set-Acl -Path $adPath -AclObject $acl
            Write-Host "  [+] ACL mise à jour sur $partition" -ForegroundColor Green
        }
    } else {
        Write-Host "  [OK] Aucune ACL CreateChild trouvée pour Authenticated Users" -ForegroundCo
    }
}

# ── ÉTAPE 3 : Supprimer les wildcards DNS existants ──────────────────────
Write-Host "[*] Recherche de wildcards DNS..." -ForegroundColor Yellow
$zones = Get-DnsServerZone | Where-Object { $_.IsADIntegrated -eq $true -and $_.ZoneType -eq "Pri

foreach ($zone in $zones) {
    $wildcards = Get-DnsServerResourceRecord -ZoneName $zone.ZoneName -ErrorAction SilentlyContin
                 Where-Object { $_.HostName -eq "*" }

    foreach ($wc in $wildcards) {
        Write-Warning "WILDCARD trouvé dans zone $($zone.ZoneName) → $($wc.RecordData)"
        if (-not $WhatIf) {
```

```
            Remove-DnsServerResourceRecord -ZoneName $zone.ZoneName -RRType $wc.RecordType `
                -Name "*" -Force
            Write-Host "  [-] Wildcard supprimé" -ForegroundColor Red
        }
    }
}

# ── ÉTAPE 4 : Audit du groupe DNSAdmins ──────────────────────────────
Write-Host "[*] Membres actuels du groupe DNSAdmins :" -ForegroundColor Yellow
Get-ADGroupMember -Identity "DNSAdmins" -Recursive |
    Select-Object Name, SamAccountName, ObjectClass, DistinguishedName |
    Format-Table -AutoSize

Write-Host "[!] Vérifiez que chaque membre a un besoin opérationnel justifié." -ForegroundColor Y
Write-Host "[!] Envisagez de désactiver le chargement de plugins DNS :" -ForegroundColor Yellow
Write-Host "    dnscmd /config /ServerLevelPluginDll '' (vider la valeur)" -ForegroundColor Gray

# ── ÉTAPE 5 : Configurer SACL d'audit sur DomainDNSZones ─────────────
Write-Host "[*] Configuration des SACL d'audit (audit d'accès réussi et échoué)..." -ForegroundCo
# Cette opération requiert auditpol pour activer l'audit DS Access
auditpol /set /subcategory:"Directory Service Access" /success:enable /failure:enable
auditpol /set /subcategory:"Directory Service Changes" /success:enable /failure:enable
Write-Host "[+] Audit Directory Service activé" -ForegroundColor Green

Write-Host "`n[+] Durcissement terminé. Redémarrez le service DNS si nécessaire." -ForegroundColo
Write-Host "[+] Vérifiez que vos applications ne dépendaient pas de CreateChild pour Authenticate
```

Désactiver le chargement de plugins DNSAdmins

```
 # Vérifier si un plugin DLL est configuré (signe d'abus ou de configuration légitime)
$dnsKey = "HKLM:\SYSTEM\CurrentControlSet\Services\DNS\Parameters"
$plugin = Get-ItemProperty -Path $dnsKey -Name "ServerLevelPluginDll" -ErrorAction SilentlyContin

if ($plugin) {
    Write-Warning "Plugin DNS configuré : $($plugin.ServerLevelPluginDll)"
    Write-Warning "Vérifier si cette DLL est légitime !"
} else {
    Write-Host "[OK] Aucun plugin DNS configuré" -ForegroundColor Green
}

# Supprimer un plugin malveillant (si identifié)
# dnscmd /config /serverlevelplugindll ""
# Remove-ItemProperty -Path $dnsKey -Name "ServerLevelPluginDll"
# Restart-Service DNS
```

La restriction des Authentication Silos Windows Server 2025 pour les comptes DNS privilégiés constitue une couche de défense complémentaire efficace contre l'abus du groupe DNSAdmins.

Monitoring Continu des Permissions DNS

Un audit ponctuel ne suffit pas : les permissions DNS peuvent être modifiées à tout moment par une attaque, une erreur d'administration ou un script malveillant. Un monitoring continu est indispensable.

Baseline ACL et détection des dérives

```
 # Exporter la baseline ACL DNS (à exécuter après durcissement)
$baseline = @{}
$partitions = @("DC=DomainDnsZones,DC=corp,DC=lan", "DC=ForestDnsZones,DC=corp,DC=lan")
foreach ($p in $partitions) {
    $baseline[$p] = (Get-Acl "AD:\$p").Access |
        Select-Object IdentityReference, ActiveDirectoryRights, AccessControlType
}
$baseline | Export-Clixml "C:\Audit\DNS-ACL-Baseline.xml"

# Script de comparaison quotidien (planifier en tâche planifiée)
$baseline = Import-Clixml "C:\Audit\DNS-ACL-Baseline.xml"
foreach ($partition in $baseline.Keys) {
    $currentACL = (Get-Acl "AD:\$partition").Access |
        Select-Object IdentityReference, ActiveDirectoryRights, AccessControlType

    $diff = Compare-Object -ReferenceObject $baseline[$partition] `
                           -DifferenceObject $currentACL `
                           -Property IdentityReference, ActiveDirectoryRights

    if ($diff) {
        Write-Warning "DÉRIVE ACL détectée sur $partition !"
        $diff | Format-Table
        # Envoyer alerte par email ou SIEM ici
    }
}
```

Règle Sigma : détection création ADIDNS Wildcard

```
 title: ADIDNS Wildcard Record Creation
id: a7c3e2b1-4d5f-4a8e-b9c2-3d1e5f7a9b4c
status: production
description: |
    Détecte la création d'un enregistrement DNS wildcard (*) dans Active Directory,
    technique utilisée pour les attaques ADIDNS Hijack et MITM.
references:
    - https://blog.netspi.com/exploiting-adidns/
    - https://github.com/Kevin-Robertson/Powermad
author: ayinedjimi-consultants.fr
date: 2026/06/01
tags:
    - attack.credential_access
    - attack.t1557.001
    - attack.lateral_movement
logsource:
    product: windows
    service: dns-server
    definition: Nécessite activation canal Microsoft-Windows-DNSServer/Audit
detection:
    selection:
        EventID: 292
        RecordName|contains: '*'
    filter_legitimate:
        # Exclure DC légitimes si nécessaire
        SourceAddress|cidr:
            - '10.0.0.0/8'   # adapter selon votre plan d'adressage DC
        DomainName|contains: '_msdcs'
    condition: selection and not filter_legitimate
falsepositives:
    - Création légitime de wildcard DNS par l'équipe réseau (documenter)
    - Migration DNS avec wildcard temporaire
level: high
fields:
    - RecordName
    - RecordType
    - ZoneName
    - SourceAddress
    - ClientID
```

Règle Wazuh : détection modification DNS suspecte

```
 <!-- Ajout dans /var/ossec/etc/rules/local_rules.xml -->

<!-- Groupe de base pour DNS Server Audit -->
<group name="windows,dns,adidns,">

  <!-- Event 292 : DNS record créé -->
  <rule id="100292" level="7">
    <if_sid>60000</if_sid>
    <field name="win.system.eventID">^292$</field>
    <description>DNS record créé dans Active Directory (ADIDNS)</description>
    <group>dns_change,</group>
  </rule>

  <!-- Event 292 avec wildcard : critique -->
  <rule id="100293" level="14">
    <if_sid>100292</if_sid>
    <field name="win.eventdata.recordName">\*</field>
    <description>CRITIQUE: Wildcard DNS (*) créé - Possible ADIDNS Hijack</description>
    <group>dns_attack,adidns_wildcard,</group>
    <options>alert_by_email</options>
  </rule>

  <!-- Event 770 : zone DNS modifiée -->
  <rule id="100770" level="6">
    <if_sid>60000</if_sid>
    <field name="win.system.eventID">^770$</field>
    <description>Zone DNS Active Directory modifiée</description>
    <group>dns_change,</group>
  </rule>

  <!-- Event 771 : enregistrement DNS supprimé -->
  <rule id="100771" level="7">
    <if_sid>60000</if_sid>
    <field name="win.system.eventID">^771$</field>
    <description>Enregistrement DNS supprimé dans Active Directory</description>
    <group>dns_change,</group>
  </rule>

  <!-- Event 5136 : modification objet AD (DomainDNSZones) -->
  <rule id="105136" level="9">
    <if_sid>60000</if_sid>
    <field name="win.system.eventID">^5136$</field>
    <field name="win.eventdata.objectDN">DomainDnsZones</field>
```

```
    <description>Modification d'objet dans la partition DomainDNSZones</description>
    <group>dns_change,ad_change,</group>
  </rule>

  <!-- Ajout membre DNSAdmins : Event 4728 -->
  <rule id="104728" level="12">
    <if_sid>60000</if_sid>
    <field name="win.system.eventID">^4728$</field>
    <field name="win.eventdata.targetUserName">DNSAdmins</field>
    <description>ALERTE: Membre ajouté au groupe DNSAdmins - Vérifier légitimité</description>
    <group>dns_attack,privilege_escalation,</group>
    <options>alert_by_email</options>
  </rule>

</group>
```

DNS Zone Transfer (AXFR/IXFR) : Contrôle et Restriction

Le transfert de zone DNS (AXFR/IXFR) est un mécanisme de réplication nécessaire entre DNS primaire et secondaires, mais il représente aussi une fuite d'information massive si non restreint : un attaquant peut obtenir l'intégralité de la base DNS du domaine, incluant tous les hôtes, serveurs d'application, et infrastructure interne.

```
 # Vérifier la configuration actuelle des transferts de zone
Get-DnsServerZoneTransfer -ZoneName "corp.lan" -ComputerName dc1.corp.lan

# Configurer les transferts sécurisés : seulement vers les NS secondaires explicites
Set-DnsServerZoneTransfer -ZoneName "corp.lan" -ComputerName dc1.corp.lan `
    -SecureSecondaries TransferToSecureServers `
    -SecondaryServers "10.0.1.2"  # IP du DNS secondaire légitime

# Via dnscmd (équivalent ligne de commande)
dnscmd dc1.corp.lan /config corp.lan /SecureSecondaries 2
dnscmd dc1.corp.lan /config corp.lan /AllowNSRecordsAutoCreation 0

# Désactiver complètement si aucun DNS secondaire n'est utilisé
Set-DnsServerZoneTransfer -ZoneName "corp.lan" -ComputerName dc1.corp.lan `
    -SecureSecondaries NoTransfer
```

```
 # Test de transfert de zone depuis l'extérieur (à effectuer en pentest)
# Si cette commande réussit, les transferts de zone sont ouverts → vulnérabilité
dig AXFR @dc1.corp.lan corp.lan
dig IXFR=0 @dc1.corp.lan corp.lan

# nslookup équivalent
nslookup -type=axfr corp.lan dc1.corp.lan

# Résultat attendu après durcissement :
# ; Transfer failed. (REFUSED)
```

Pour compléter la sécurisation du DNS, la mise en œuvre de DNSSEC sur Windows Server 2025 ajoute l'authentification cryptographique des enregistrements DNS et empêche le DNS spoofing même si un enregistrement illégitime était créé.

Conformité NIS 2 et ANSSI : Sécurité DNS comme Obligation Réglementaire

Pour les organisations françaises, la sécurité DNS dans Active Directory n'est plus uniquement une bonne pratique — c'est une exigence réglementaire croissante en 2026.

NIS 2 et la sécurité DNS

La directive NIS 2 (2022/2555/UE), transposée en droit français par la loi du 7 février 2024, impose à son article 21§2.d des mesures de sécurité des réseaux et des systèmes d'information proportionnées aux risques. Pour une entité essentielle ou importante, une compromission du DNS interne peut entraîner une défaillance généralisée des services numériques — exactement le type d'incident que NIS 2 cherche à prévenir.

Les exigences applicables au DNS incluent :

Art. 21§2.a : Politiques d'analyse des risques — le DNS AD intégré doit figurer dans la cartographie des risques

Art. 21§2.d : Sécurité de la chaîne d'approvisionnement — les permissions DNS font partie du périmètre

Art. 21§2.e : Sécurité dans l'acquisition et le développement — les paramètres DNS par défaut "trop permissifs" doivent être durcis

Art. 21§2.j : Authentification multifacteur — applicable aux accès DNSAdmins

Recommandations ANSSI

L'ANSSI a publié en 2022 un guide dédié à la sécurisation des infrastructures DNS, qui recommande explicitement :

La restriction des permissions d'écriture dans les zones DNS intégrées AD

L'interdiction des transferts de zone vers des serveurs non autorisés

L'activation de DNSSEC pour les zones critiques

La journalisation complète des modifications DNS et leur centralisation dans un SIEM

La revue semestrielle des membres du groupe DNSAdmins

Le contexte RGPD renforce ces obligations : une fuite de la cartographie DNS interne peut révéler l'architecture technique d'une organisation et faciliter des attaques ciblées, avec les risques de sanction que cela implique pour les DPO. Les entreprises parisiennes et franciliennes soumises à NIS 2, notamment dans les secteurs bancaire, santé et énergie, ont intérêt à documenter précisément leurs contrôles DNS pour les audits de l'ANSSI.

Questions Fréquentes sur l'Audit des Permissions DNS

Qu'est-ce que l'ADIDNS Hijack et pourquoi est-il dangereux ?

L'ADIDNS Hijack (Active Directory Integrated DNS Hijacking) est une attaque qui exploite les permissions trop permissives sur les zones DNS intégrées à l'AD. Par défaut, tout utilisateur authentifié peut créer des enregistrements DNS dans DomainDNSZones, ce qui permet à un attaquant interne de rediriger le trafic réseau vers sa machine pour capturer des identifiants NTLM ou effectuer des attaques man-in-the-middle. La combinaison avec un enregistrement wildcard (*) amplifie considérablement l'impact.

Comment vérifier rapidement si mon domaine est vulnérable aux attaques ADIDNS ?

Exécutez cette commande PowerShell sur un DC : `(Get-Acl 'AD:\DC=DomainDnsZones,DC=corp,DC=lan').Access | Where-Object { $_.IdentityReference -match 'Authenticated Users' } | Select IdentityReference, ActiveDirectoryRights`. Si le résultat affiche 'CreateChild' pour 'Authenticated Users' ou 'Domain Computers', votre domaine est vulnérable. Retirez immédiatement cette permission après avoir validé qu'aucune application métier n'en dépend.

Le groupe DNSAdmins représente-t-il un risque de sécurité critique ?

Oui. Le groupe DNSAdmins permet de charger des DLL arbitraires dans le processus dns.exe (SYSTEM) via la commande `dnscmd /config /serverlevelplugindll`. Cette technique, documentée depuis 2017 et répertoriée dans ATT&CK T1543.003, transforme l'appartenance au groupe DNSAdmins en escalade de privilèges vers SYSTEM sur le DC. Limitez ce groupe aux seuls administrateurs DNS qui en ont un besoin opérationnel réel, activez l'audit sur ce groupe, et désactivez le chargement de plugins si vous n'utilisez pas cette fonctionnalité.

Quels Event IDs surveiller pour détecter des modifications DNS suspectes ?

Les Event IDs critiques pour la détection DNS dans Active Directory sont : 292 (DNS record created – canal Microsoft-Windows-DNSServer/Audit), 770 (DNS zone updated), 771 (DNS record deleted), 4662 (opération effectuée sur un objet AD – DomainDNSZones), 5136 (modification d'un objet d'annuaire) et 4728/4732 (ajout d'un membre au groupe DNSAdmins). Le canal DNSServer/Audit doit être activé manuellement via wevtutil ou GPO.

NIS 2 impose-t-elle des exigences spécifiques sur la sécurité DNS ?

La directive NIS 2 (transposée en droit français en 2024) impose à l'article 21§2.d la sécurisation de la chaîne d'approvisionnement et des accès. Le DNS étant l'infrastructure critique permettant la résolution de tous les services, une compromission DNS peut entraîner une violation généralisée. L'ANSSI, dans son guide 'Sécurisation des infrastructures DNS' (2022), recommande explicitement l'audit des ACL DNS, la restriction des transferts de zone et la mise en œuvre de DNSSEC. Les entités essentielles et importantes doivent pouvoir démontrer ces contrôles lors des audits de conformité.

À RETENIR

À retenir

Permission CreateChild sur Authenticated Users = vulnérabilité CRITIQUE : retirez-la immédiatement de DomainDNSZones et ForestDNSZones après avoir audité les dépendances applicatives.

DNSAdmins = chemin d'escalade vers SYSTEM : limitez ce groupe, auditez chaque ajout de membre (Event 4728), et vérifiez l'absence de plugin DLL configuré (clé de registre ServerLevelPluginDll).

Activez le canal Microsoft-Windows-DNSServer/Audit : les Event IDs 292 (création), 770 (mise à jour zone), 771 (suppression) ne sont collectés que si ce canal est explicitement activé via wevtutil.

Un wildcard DNS (*) dans une zone = incident en cours : vérifiez immédiatement avec `Resolve-DnsName` un hôte inexistant — si vous obtenez une IP, une attaque ADIDNS est probablement active.

NIS 2 et ANSSI imposent ces contrôles : documentez votre audit DNS, la baseline ACL et les résultats de durcissement pour démontrer la conformité lors des audits réglementaires.

Références et documentation

Microsoft Learn — Sécurité Windows Server

ANSSI — Guide de sécurisation Active Directory

MITRE ATT&CK — Techniques Active Directory