# ADCS Offline Root CA : Architecture PKI Sécurisée Windows Server 2025

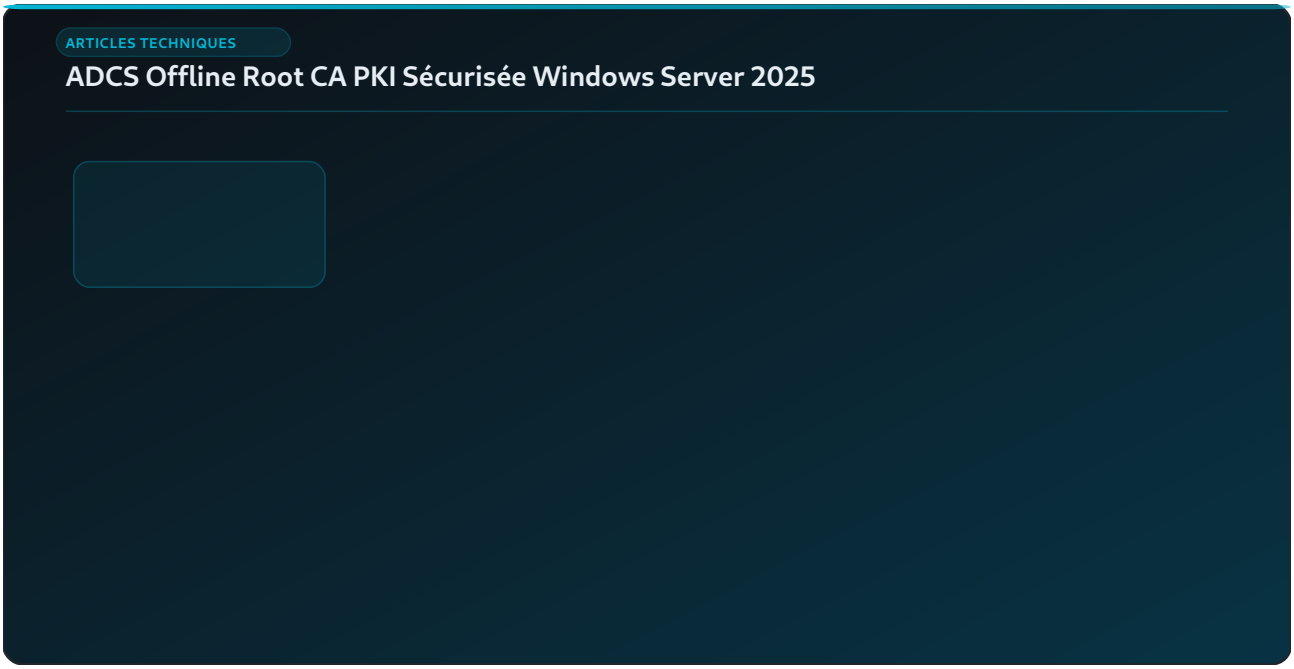
Architecturez une PKI ADCS deux niveaux sur Windows Server 2025 : Root CA offline, durcissement ESC1-ESC8, OCSP, conformité NIS 2 et RGS ANSSI.

📋 EN BREF

- Une PKI à deux niveaux (Root CA offline + Issuing CA online) est le standard de sécurité recommandé par l'ANSSI et le NIST pour toute infrastructure Windows Server 2025.
- La Root CA offline doit être démarrée uniquement pour signer des certificats de CA subordonnée, puis immédiatement éteinte — jamais connectée au réseau.
- Les vulnérabilités ESC1 à ESC8 d'ADCS permettent une élévation de privilèges jusqu'au Domain Admin en quelques minutes si les templates ne sont pas durcis.
- La révocation via OCSP combinée aux CRL delta garantit une infrastructure PKI résiliente avec des temps de contrôle inférieurs à 5 secondes.
- La conformité NIS 2 et le RGS de l'ANSSI imposent une gestion documentée des clés cryptographiques et des procédures de révocation testées.

Dans les grandes organisations françaises — banques, hôpitaux, collectivités territoriales, opérateurs de services essentiels soumis à NIS 2 — la gestion des identités numériques repose sur une infrastructure à clé publique (PKI) robuste. Active Directory Certificate Services (ADCS) sous Windows Server 2025 constitue la colonne vertébrale de cette PKI pour des milliers d'entreprises en France et en Europe. Pourtant, la majorité des RSSI et responsables informatiques négligent la sécurisation de leur Root CA, laissant des failles béantes exploitables par des attaquants. Une Root CA compromise, c'est l'ensemble du système de confiance de

l'organisation qui s'effondre : VPN, authentification par certificat, chiffrement des emails, signature de code. Ce guide complet accompagne les équipes sécurité dans la conception, l'installation et le durcissement d'une architecture PKI ADCS à deux niveaux sur Windows Server 2025, en intégrant les recommandations de l'ANSSI, les exigences NIS 2 et la protection contre les attaques ADCS documentées par SpecterOps. Les RSSI parisiens et les DSI des grandes métropoles françaises trouveront ici une référence opérationnelle alignée sur le Référentiel Général de Sécurité (RGS) et les obligations réglementaires européennes en vigueur.




L'architecture PKI à deux niveaux est le modèle de référence pour toute organisation soucieuse de sécurité. Elle sépare la racine de confiance (Root CA) des opérations quotidiennes (Issuing CA), limitant drastiquement la surface d'attaque.

## Pourquoi isoler la Root CA ?

La Root CA est le sommet de la chaîne de confiance. Si elle est compromise, tous les certificats émis sont potentiellement falsifiables. En la maintenant hors ligne :

**Aucune surface réseau** : impossible d'atteindre la clé privée Root via le réseau

**Durée de vie longue** : 20 ans, le Root CA ne signe que rarement (renouvellement de la sub CA)

**Contrôle physique** : la machine Root CA est stockée dans un coffre-fort ou une armoire sécurisée

**Séparation des rôles** : l'opération Root CA nécessite une procédure formelle avec au moins deux personnes

## Topologie recommandée

```
[Root CA Offline]  —  Windows Server 2025 Core, standalone, jamais connecté
      |
      | (signature manuelle du certificat Issuing CA)
      |
[Issuing CA Online]  —  Windows Server 2025, membre du domaine AD
      |
      | (émission de certificats pour : users, machines, services)
      |
[Clients AD, serveurs, équipements réseau]
```

## Prérequis matériels et logiciels

**Root CA** : VM ou physique, Windows Server 2025 Core (minimise la surface d'attaque), 2 vCPU, 4 Go RAM, disque chiffré BitLocker

**Issuing CA** : Windows Server 2025 Standard/Datacenter, membre du domaine AD, 4 vCPU, 8 Go RAM

**Serveur CRL/OCSP** : IIS sur serveur membre, accessible depuis Internet si PKI expose des certificats publics

**HSM (optionnel mais recommandé)** : nCipher/Thales ou YubiHSM pour protéger la clé privée Root CA

---

## Installation de la Root CA Offline sur Windows Server 2025 Core

L'installation de la Root CA se fait sur un serveur **jamais connecté au réseau de production**. Windows Server 2025 Core est préféré pour sa surface d'attaque réduite. Voici le script PowerShell complet :

> ⚡ **Retour terrain**
>
> Sur un projet de hardening Active Directory pour un groupe de transport, j'ai découvert 23 comptes de service avec des mots de passe non renouvelés depuis plus de 7 ans — dont 3 avec des délégations non contraintes. L'inventaire des délégations Kerberos est systématiquement le premier point que j'audite désormais : c'est l'angle mort le plus fréquent et le plus exploitable.

### Étape 1 : Installation du rôle ADCS

```
# Sur le serveur Root CA (hors ligne)
# Installation du rôle ADCS
Install-WindowsFeature ADCS-Cert-Authority -IncludeManagementTools

# Vérification
Get-WindowsFeature ADCS-Cert-Authority
```

Étape 2 : Configuration de la Root CA (RSA 4096, 20 ans)

```
# Configuration de la Root CA Standalone
# RSA 4096 bits recommandé pour Root CA longue durée
# Alternative moderne: ECDSA P-384

$CAName = "AYINEDJIMI-Root-CA"
$CADistinguishedNameSuffix = "DC=ayinedjimi,DC=fr"
$ValidityPeriod = "Years"
$ValidityPeriodUnits = 20
$CRLPeriod = "Weeks"
$CRLPeriodUnits = 26  # CRL valide 6 mois (Root CA rarement en ligne)
$CRLDeltaPeriodUnits = 0  # Pas de delta CRL pour Root CA

Install-AdcsCertificationAuthority `
    -CAType StandaloneRootCA `
    -CACommonName $CAName `
    -CADistinguishedNameSuffix $CADistinguishedNameSuffix `
    -KeyLength 4096 `
    -HashAlgorithmName SHA256 `
    -CryptoProviderName "RSA#Microsoft Software Key Storage Provider" `
    -ValidityPeriod $ValidityPeriod `
    -ValidityPeriodUnits $ValidityPeriodUnits `
    -Force

Write-Host "Root CA installee. Configuration de la politique..."
```

# Étape 3 : Configuration CDP et AIA

```
# Suppression des CDP/AIA par défaut (non accessibles hors ligne)
$CA = "AYINEDJIMI-Root-CA"

# Supprimer toutes les CDP existantes
$CRL = Get-CACRLDistributionPoint
foreach ($c in $CRL) { Remove-CACRLDistributionPoint $c.URI -Force }

# Ajouter CDP HTTP (serveur CRL en ligne)
Add-CACRLDistributionPoint `
    -URI "http://pki.ayinedjimi.fr/crl/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl" `
    -AddToCertificateCDP `
    -AddToFreshestCrl

# Supprimer toutes les AIA existantes
$AIA = Get-CAAuthorityInformationAccess
foreach ($a in $AIA) { Remove-CAAuthorityInformationAccess $a.URI -Force }

# Ajouter AIA HTTP
Add-CAAuthorityInformationAccess `
    -URI "http://pki.ayinedjimi.fr/aia/<ServerDNSName>_<CaName><CertificateName>.crt" `
    -AddToCertificateAIA

# Configurer la durée de validité max des certificats émis
certutil -setreg ca\ValidityPeriodUnits 10
certutil -setreg ca\ValidityPeriod "Years"

# Configurer CRL
certutil -setreg ca\CRLPeriodUnits 26
certutil -setreg ca\CRLPeriod "Weeks"
certutil -setreg ca\CRLDeltaPeriodUnits 0

# Redémarrer le service
Restart-Service certsvc

# Publier la CRL initiale
certutil -crl

Write-Host "Root CA configuree. Copiez les fichiers suivants sur cle USB:"
Write-Host "1. Le certificat Root CA: C:\Windows\system32\CertSrv\CertEnroll"
Write-Host "2. La CRL Root CA"
Write-Host "Eteignez ce serveur. Il ne doit JAMAIS etre rallume sans procedure formelle."
```

Étape 4 : Export du certificat Root CA

```
# Exporter le certificat Root CA (sans la clé privée!)
# La clé privée reste sur le serveur chiffré BitLocker

$RootCACert = Get-ChildItem Cert:\LocalMachine\CA |
    Where-Object {$_.Subject -like "*AYINEDJIMI-Root-CA*"}

Export-Certificate -Cert $RootCACert `
    -FilePath "E:\RootCA\AYINEDJIMI-Root-CA.cer" `
    -Type CERT

# Copier aussi la CRL
Copy-Item "C:\Windows\System32\CertSrv\CertEnroll\*.crl" "E:\RootCA"

Write-Host "Fichiers prêts sur E:\ (clé USB chiffrée)"
Write-Host "Transportez la clé USB vers le serveur Issuing CA"
```

## Configuration de la Subordinate CA (Issuing CA) Online

L'Issuing CA est membre du domaine Active Directory et gère l'émission quotidienne des certificats. Son certificat doit être signé par la Root CA offline lors d'une procédure formelle.

### Installation de l'Issuing CA

```
# Sur le serveur Issuing CA (membre du domaine)
Install-WindowsFeature ADCS-Cert-Authority, ADCS-Online-Cert, ADCS-Web-Enrollment `
    -IncludeManagementTools

# Installer comme Enterprise Subordinate CA
Install-AdcsCertificationAuthority `
    -CAType EnterpriseSubordinateCA `
    -CACommonName "AYINEDJIMI-Issuing-CA-01" `
    -CADistinguishedNameSuffix "DC=ayinedjimi,DC=fr" `
    -KeyLength 2048 `
    -HashAlgorithmName SHA256 `
    -CryptoProviderName "RSA#Microsoft Software Key Storage Provider" `
    -OutputCertRequestFile "C:\Temp\IssuingCA.req" `
    -Force

# Le service ne démarrera pas tant que le certificat n'est pas signé par la Root CA
Write-Host "Fichier de requête créé: C:\Temp\IssuingCA.req"
Write-Host "Transportez ce fichier sur clé USB vers la Root CA"
```

### Signature par la Root CA (procédure offline)

```
# Sur la Root CA (après l'avoir démarrée selon la procédure formelle)
# Soumettre la requête de l'Issuing CA
certreq -submit -config "localhost\AYINEDJIMI-Root-CA" "E:\IssuingCA.req"

# Récupérer le RequestID affiché, puis approuver
certutil -resubmit <RequestID>

# Récupérer le certificat signé
certreq -retrieve -config "localhost\AYINEDJIMI-Root-CA" <RequestID> "E:\IssuingCA.cer"

# Éteindre immédiatement la Root CA après!
Stop-Computer -Force
```

Installation du certificat Issuing CA et publication AD

```
# Sur l'Issuing CA
# Installer le certificat signé
certutil -installcert "C:\Temp\IssuingCA.cer"

# Publier le certificat Root CA dans AD (NTAuthCertificates, AIA)
certutil -dspublish -f "C:\Temp\AYINEDJIMI-Root-CA.cer" RootCA
certutil -dspublish -f "C:\Temp\AYINEDJIMI-Root-CA.cer" NTAuthCA

# Démarrer le service
Start-Service certsvc

# Configurer CDP et AIA sur l'Issuing CA
Add-CACRLDistributionPoint `
    -URI "http://pki.ayinedjimi.fr/crl/<CaName><CRLNameSuffix><DeltaCRLAllowed>.crl" `
    -AddToCertificateCDP -AddToFreshestCrl

Add-CAAuthorityInformationAccess `
    -URI "http://pki.ayinedjimi.fr/aia/<ServerDNSName>_<CaName><CertificateName>.crt" `
    -AddToCertificateAIA

Add-CAAuthorityInformationAccess `
    -URI "http://pki.ayinedjimi.fr/ocsp" `
    -AddToCertificateOcsp

Restart-Service certsvc
```

## Templates de Certificats : Durcissement et Bonnes Pratiques

Les templates de certificats ADCS sont une source majeure de vulnérabilités. Les recherches de SpecterOps (Will Schroeder et Lee Christensen) ont documenté 8 classes d'abus (ESC1 à ESC8) permettant une élévation de privilèges jusqu'au Domain Admin.

Désactivation des templates dangereux par défaut

```
# Templates à désactiver immédiatement (dangereux par défaut)
$DangerousTemplates = @(
    "WebServer",           # Permet enrollee supplies subject
    "SubCA",              # Permet de créer des CA subordonnées
    "Administrator",      # Client Auth + enrollee supplies subject
    "DomainController",   # Rarement nécessaire pour les DCs modernes
    "CEPEncryption",      # Rarement utilisé
    "EnrollmentAgent",    # ESC3: peut être utilisé pour obtenir des certs au nom d'autres
    "EnrollmentAgentOffline"
)

foreach ($template in $DangerousTemplates) {
    try {
        Get-CATemplate | Where-Object {$_.Name -eq $template} |
            Remove-CATemplate -Force
        Write-Host "Template désactivé: $template" -ForegroundColor Green
    } catch {
        Write-Host "Template non trouvé ou déjà désactivé: $template" -ForegroundColor Yellow
    }
}
```

## Vulnérabilités ESC1 à ESC8 — Tableau de référence

| ID | Nom | Condition vulnérable | Impact |
|---|---|---|---|
| **ESC1** | Enrollee Supplies Subject | CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT + Client Auth EKU | Cert pour n'importe quel utilisateur (Domain Admin) |
| **ESC2** | AnyPurpose EKU | EKU = Any Purpose ou pas d'EKU | Authentification client + Agent d'enrôlement |
| **ESC3** | Enrollment Agent | EKU = Certificate Request Agent | Obtenir des certs au nom d'autres utilisateurs |
| **ESC4** | Vulnerable Template ACL | Write permissions sur template pour utilisateurs non-admins | Modifier le template pour activer ESC1 |
| **ESC5** | Vulnerable PKI Object ACL | Write sur objets PKI dans AD | Modifier la configuration PKI |
| **ESC6** | EDITF_ATTRIBUTESUBJECTALTNAME2 | Flag activé sur CA + Client Auth template | SAN arbitraire sur n'importe quel template |
| **ESC7** | Vulnerable CA ACL | Manage CA ou Issue rights pour non-admins | Approuver des requêtes de certificats |
| **ESC8** | NTLM Relay vers ADCS HTTP | HTTP enrollment endpoints actifs | Relay NTLM pour obtenir cert machine/DC |

Script PowerShell d'audit des templates vulnérables

```
# Script d'audit ADCS - Détection ESC6 (perspective défensive)
# Vérification EDITF_ATTRIBUTESUBJECTALTNAME2

function Test-ESC6Vulnerability {
    param([string]$CAConfig = "IssuingCA.ayinedjimi.fr\AYINEDJIMI-Issuing-CA-01")
    $flags = certutil -config $CAConfig -getreg policy\EditFlags 2>&1
    if ($flags -match "EDITF_ATTRIBUTESUBJECTALTNAME2") {
        return [PSCustomObject]@{
            Vulnerability = "ESC6"
            Severity      = "CRITIQUE"
            Details       = "EDITF_ATTRIBUTESUBJECTALTNAME2 enabled on CA"
            Remediation   = "certutil -config $CAConfig -setreg policy\EditFlags -EDITF_ATTRIBUTE
        }
    } else {
        return [PSCustomObject]@{
            Vulnerability = "ESC6"
            Severity      = "OK"
            Details       = "Flag non activé"
        }
    }
}

# Audit ACL templates (ESC4)
function Test-ESC4TemplateACL {
    $SearchBase = "CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,D
    $Templates = Get-ADObject -SearchBase $SearchBase `
        -Filter {objectClass -eq "pKICertificateTemplate"} -Properties *

    $Results = @()
    foreach ($tmpl in $Templates) {
        $ACL = Get-ACL "AD:\$($tmpl.DistinguishedName)"
        $DangerousACE = $ACL.Access | Where-Object {
            $_.ActiveDirectoryRights -match "Write|GenericAll|GenericWrite" -and
            $_.IdentityReference -notmatch "Domain Admins|Enterprise Admins|SYSTEM|Cert Publisher
        }
        if ($DangerousACE) {
            foreach ($ace in $DangerousACE) {
                $Results += [PSCustomObject]@{
                    Template  = $tmpl.Name
                    Identity  = $ace.IdentityReference
                    Rights    = $ace.ActiveDirectoryRights
                    ESC       = "ESC4"
```

```
                    Severity  = "HIGH"
                }
            }
        }
    }
    return $Results
}

# Corriger ESC6 si détecté
$esc6 = Test-ESC6Vulnerability
if ($esc6.Severity -eq "CRITIQUE") {
    Write-Warning "ESC6 détecté! Application de la correction..."
    certutil -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2
    Restart-Service certsvc
    Write-Host "[ESC6] Corrigé avec succès" -ForegroundColor Green
}

# Activer Manager Approval sur templates sensibles (mitigation ESC1)
function Enable-ManagerApproval {
    param([string]$TemplateName)
    $templateDN = "CN=$TemplateName,CN=Certificate Templates,CN=Public Key Services,CN=Services,C
    $template = [ADSI]"LDAP://$templateDN"
    $currentFlags = $template.Properties["msPKI-Enrollment-Flag"].Value
    # Ajouter CT_FLAG_PEND_ALL_REQUESTS (0x2)
    $template.Properties["msPKI-Enrollment-Flag"].Value = $currentFlags -bor 0x2
    $template.CommitChanges()
    Write-Host "Manager Approval activé sur: $TemplateName" -ForegroundColor Green
}
```

## Révocation et CRL : Configuration Optimale

La révocation de certificats est critique pour répondre à une compromission. Une CRL inaccessible ou expirée peut bloquer l'ensemble des services qui dépendent de la PKI.

## Configuration CDP haute disponibilité

```
# Configuration IIS pour servir les CRL
# Sur le serveur web PKI (pki.ayinedjimi.fr)

Import-Module WebAdministration

# Créer le répertoire virtuel pour les CRL
$PhysicalPath = "C:\PKI\CRL"
New-Item -ItemType Directory -Path $PhysicalPath -Force

# Configurer IIS
New-WebVirtualDirectory -Site "Default Web Site" `
    -Name "crl" `
    -PhysicalPath $PhysicalPath

# Autoriser la double extension dans IIS (nécessaire pour delta CRL)
Set-WebConfigurationProperty `
    -Filter "system.webServer/security/requestFiltering" `
    -PSPath "IIS:\Sites\Default Web Site\crl" `
    -Name "allowDoubleEscaping" `
    -Value $true

# Script de copie automatique des CRL (à planifier en Scheduled Task horaire)
$CopyScript = @'
$Source = "\IssuingCA\C$\Windows\System32\CertSrv\CertEnroll"
$Dest   = "C:\PKI\CRL"
Copy-Item "$Source*.crl" $Dest -Force
Copy-Item "$Source*.crt" "C:\PKI\AIA" -Force
'@

Set-Content -Path "C:\Scripts\Copy-CRL.ps1" -Value $CopyScript

$Action  = New-ScheduledTaskAction -Execute "PowerShell.exe" `
    -Argument "-File C:\Scripts\Copy-CRL.ps1"
$Trigger = New-ScheduledTaskTrigger -RepetitionInterval (New-TimeSpan -Hours 1) `
    -Once -At (Get-Date)
Register-ScheduledTask -TaskName "Copy-PKI-Files" `
    -Action $Action -Trigger $Trigger -RunLevel Highest
```

### Installation et configuration OCSP

```
# Sur le serveur Online Responder
Install-WindowsFeature ADCS-Online-Cert -IncludeManagementTools
Install-AdcsOnlineResponder

# Monitoring: vérifier la validité OCSP
function Test-OCSPResponse {
    param([string]$CertThumbprint)
    $Cert  = Get-Item "Cert:\LocalMachine\My\$CertThumbprint"
    $Chain = New-Object System.Security.Cryptography.X509Certificates.X509Chain
    $Chain.ChainPolicy.RevocationMode    = "Online"
    $Chain.ChainPolicy.RevocationFlag    = "ExcludeRoot"
    $Chain.ChainPolicy.UrlRetrievalTimeout = New-TimeSpan -Seconds 30
    $Result = $Chain.Build($Cert)
    if (-not $Result) {
        Write-Warning "OCSP check failed: $($Chain.ChainStatus.StatusInformation)"
    } else {
        Write-Host "OCSP OK - Certificat valide" -ForegroundColor Green
    }
}
```

Monitoring expiration CRL

```
# Script de monitoring CRL - à exécuter quotidiennement via Scheduled Task
function Get-CRLExpiryStatus {
    param(
        [string]$CAConfig = "IssuingCA.ayinedjimi.fr\AYINEDJIMI-Issuing-CA-01",
        [int]$WarningDays = 7,
        [int]$CriticalDays = 3
    )
    $CRLInfo    = certutil -config $CAConfig -getcrls 2>&1
    $NextUpdate = ($CRLInfo | Select-String "Next CRL Publish").ToString() -replace ".*: ", ""
    $ExpiryDate = [datetime]::ParseExact($NextUpdate.Trim(), "M/d/yyyy h:mm tt", $null)
    $DaysLeft   = ($ExpiryDate - (Get-Date)).Days

    $Status = if ($DaysLeft -gt $WarningDays) { "OK" }
              elseif ($DaysLeft -gt $CriticalDays) { "WARNING" }
              else { "CRITICAL" }

    if ($Status -ne "OK") {
        Send-MailMessage -To "rssi@ayinedjimi.fr" `
            -Subject "[$Status] CRL PKI expire dans $DaysLeft jour(s)" `
            -Body "La CRL de l'Issuing CA expire le $ExpiryDate. Action requise." `
            -SmtpServer "smtp.ayinedjimi.fr" `
            -From "monitoring@ayinedjimi.fr"
    }

    return [PSCustomObject]@{
        CA           = $CAConfig
        CRLExpiry    = $ExpiryDate
        DaysRemaining = $DaysLeft
        Status       = $Status
    }
}

Get-CRLExpiryStatus | Format-Table -AutoSize
```

## Sécurisation Physique et Logique de la Root CA

### Chiffrement BitLocker de la Root CA

```
# Sur le serveur Root CA, avant de l'éteindre définitivement
Enable-BitLocker -MountPoint "C:" `
    -EncryptionMethod XtsAes256 `
    -UsedSpaceOnly `
    -TpmProtector

# Sauvegarder la clé de récupération (coffre-fort physique, jamais en ligne)
$BLV = Get-BitLockerVolume -MountPoint "C:"
$RecoveryKey = $BLV.KeyProtector |
    Where-Object {$_.KeyProtectorType -eq "RecoveryPassword"}

Write-Host "=== CLÉ DE RÉCUPÉRATION BITLOCKER ==="
Write-Host $RecoveryKey.RecoveryPassword
Write-Host "Stockez sur deux médias chiffrés séparés, en deux lieux physiques différents."
Write-Host "Ne stockez JAMAIS sur un serveur connecté au réseau."
```

### Procédure formelle d'activation de la Root CA

La Root CA ne doit être démarrée que pour des opérations spécifiques (renouvellement de l'Issuing CA, signature d'une nouvelle sub CA). La procédure doit être documentée et requérir au minimum deux personnes autorisées :

**Déclencheur** : Demande formelle signée par le RSSI

**Préparation** : Deux administrateurs présents physiquement en salle sécurisée

**Démarrage** : Démarrage du serveur Root CA en local uniquement (jamais via iDRAC/ILO réseau)

**Opération** : Signature du certificat demandé, publication CRL

**Vérification** : Double-vérification du certificat émis (empreinte SHA-256)

**Arrêt** : Arrêt immédiat et retour en coffre-fort sous scellé

**Journalisation** : Documentation dans le registre d'opérations CA (papier + électronique signé)

Protection et sauvegarde de la clé privée Root CA

```
# Export de la clé privée Root CA — opération unique lors de l'installation initiale
$Password = ConvertTo-SecureString "MotDePasseTresComplexe!PKI2025" -AsPlainText -Force
$Cert = Get-ChildItem Cert:\LocalMachine\My |
    Where-Object {$_.Subject -like "*AYINEDJIMI-Root-CA*"}

Export-PfxCertificate -Cert $Cert `
    -FilePath "E:\RootCA\AYINEDJIMI-Root-CA-PRIVATE.pfx" `
    -Password $Password `
    -ChainOption EndEntityCertOnly

# Vérifier l'intégrité du backup
$BackupHash = Get-FileHash "E:\RootCA\AYINEDJIMI-Root-CA-PRIVATE.pfx" -Algorithm SHA256
Write-Host "Empreinte SHA-256 du backup: $($BackupHash.Hash)"
Write-Host "Documentez cette empreinte dans le registre PKI."
```

## ADCS et les Attaques : ESC1 à ESC8 — Détection et Remédiation

Les vulnérabilités ADCS ont révolutionné le paysage des attaques Active Directory depuis la publication de "Certified Pre-Owned" par SpecterOps en 2021. Des outils comme Certify (C#, SpecterOps) et Certipy (Python, Oliver Lyak) permettent d'exploiter ces failles en quelques commandes depuis n'importe quel compte utilisateur du domaine. La posture défensive nécessite une compréhension approfondie de ces techniques d'attaque.

Script de durcissement complet ADCS

```powershell
# Script de durcissement ADCS - Version défensive complète
# Basé sur les recommandations ANSSI et SpecterOps "Certified Pre-Owned"

param(
    [string]$CAServer = "IssuingCA.ayinedjimi.fr",
    [string]$CAName   = "AYINEDJIMI-Issuing-CA-01",
    [switch]$WhatIf   = $false
)

$CAConfig = "$CAServer\$CAName"

Write-Host "=== DURCISSEMENT ADCS ===" -ForegroundColor Cyan
Write-Host "CA cible: $CAConfig" -ForegroundColor Cyan

# 1. ESC6: Désactiver EDITF_ATTRIBUTESUBJECTALTNAME2
Write-Host "`n[ESC6] Vérification..." -ForegroundColor Yellow
$flags = certutil -config $CAConfig -getreg policy\EditFlags 2>&1
if ($flags -match "EDITF_ATTRIBUTESUBJECTALTNAME2") {
    if (-not $WhatIf) {
        certutil -config $CAConfig -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2
        Write-Host "[ESC6] CORRIGÉ" -ForegroundColor Green
    }
} else {
    Write-Host "[ESC6] OK - Flag non activé" -ForegroundColor Green
}

# 2. ESC8: Activer Extended Protection for Authentication (EPA) sur /certsrv/
Write-Host "`n[ESC8] Activation EPA sur /certsrv/..." -ForegroundColor Yellow
if (-not $WhatIf) {
    Import-Module WebAdministration
    Set-WebConfigurationProperty `
        -Filter "system.webServer/security/authentication/windowsAuthentication" `
        -PSPath "IIS:\Sites\Default Web Site\certsrv" `
        -Name "extendedProtection.tokenChecking" `
        -Value "Require"
    Write-Host "[ESC8] EPA activé" -ForegroundColor Green
}

# 3. Audit ACL templates (ESC4)
Write-Host "`n[ESC4] Audit ACL templates..." -ForegroundColor Yellow
$SearchBase = "CN=Certificate Templates,CN=Public Key Services,CN=Services,CN=Configuration,DC=ay
$Templates  = Get-ADObject -SearchBase $SearchBase `
```

```
    -Filter {objectClass -eq "pKICertificateTemplate"} -Properties DistinguishedName, Name

foreach ($tmpl in $Templates) {
    $ACL = Get-ACL "AD:\$($tmpl.DistinguishedName)"
    $BadACE = $ACL.Access | Where-Object {
        $_.ActiveDirectoryRights -match "Write|GenericAll" -and
        $_.IdentityReference -notmatch "Domain Admins|Enterprise Admins|SYSTEM|Cert Publishers"
    }
    if ($BadACE) {
        Write-Warning "[ESC4] Template $($tmpl.Name) a des ACL dangereuses:"
        $BadACE | ForEach-Object { Write-Warning "  $($_.IdentityReference): $($_.ActiveDirectory
    }
}

# 4. Rapport
Write-Host "`n=== FIN DU DURCISSEMENT ===" -ForegroundColor Cyan
Write-Host "Timestamp: $(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')" -ForegroundColor White
Write-Host "Redémarrage du service CA requis si des corrections ont été appliquées." -ForegroundC
if (-not $WhatIf) { Restart-Service certsvc }
```

## Monitoring ADCS : Event IDs et Alertes

Le monitoring des événements ADCS est essentiel pour détecter les abus de l'infrastructure PKI. Windows génère des événements spécifiques pour chaque opération de la CA que tout RSSI doit connaître et surveiller.

## Event IDs critiques ADCS

| Event ID | Description | Priorité | Canal |
|---|---|---|---|
| **4886** | Certificat demandé | Info | Security |
| **4887** | Certificat émis | Info / HIGH si EKU sensible | Security |
| **4888** | Certificat refusé | Medium | Security |
| **4898** | Template de certificat chargé | Info | Security |
| **4890** | Paramètres gestionnaire CA modifiés | HIGH | Security |
| **4896** | Entrée base de données CA supprimée | HIGH | Security |

## Activation de l'audit ADCS

```
# Activer l'audit complet sur la CA
certutil -config "IssuingCA.ayinedjimi.fr\AYINEDJIMI-Issuing-CA-01" `
    -setreg CA\AuditFilter 127

# AuditFilter 127 = bitmask tous événements:
# 1=Start/Stop, 2=Backup/Restore, 4=Cert Issued, 8=Cert Revoked,
# 16=CRL Published, 32=Config Changed, 64=Key Archived

Restart-Service certsvc

# Activer l'audit Certification Services dans la stratégie d'audit
auditpol /set /subcategory:"Certification Services" /success:enable /failure:enable

Write-Host "Audit ADCS activé. Vérification:"
auditpol /get /subcategory:"Certification Services"
```

## Règles Wazuh pour ADCS

```
<!-- /var/ossec/etc/rules/adcs_rules.xml -->
<group name="windows,adcs,pki">

  <rule id="100200" level="10">
    <if_group>windows</if_group>
    <id>4887</id>
    <field name="win.eventdata.templateName">Administrator|SubCA|WebServer|EnrollmentAgent</field
    <description>ADCS: Certificat émis depuis un template a risque élevé</description>
    <mitre><id>T1649</id></mitre>
  </rule>

  <rule id="100201" level="14">
    <if_group>windows</if_group>
    <id>4887</id>
    <field name="win.eventdata.requesterName">\.*(admin|DA-|EA-).*</field>
    <description>ADCS: Certificat Client Auth émis pour compte privilégié</description>
    <mitre><id>T1649</id></mitre>
  </rule>

  <rule id="100202" level="12">
    <if_group>windows</if_group>
    <id>4890</id>
    <description>ADCS: Configuration CA modifiée — investigation immédiate requise</description>
    <mitre><id>T1484</id></mitre>
  </rule>

  <rule id="100203" level="10">
    <if_group>windows</if_group>
    <id>4896</id>
    <description>ADCS: Suppression d'entrée base CA — possible tentative de couvrir des traces</d
    <mitre><id>T1070</id></mitre>
  </rule>

</group>
```

KQL Microsoft Sentinel — Émission de certificats suspectes

```
// Détection: certificats Client Auth émis pour comptes privilégiés (ADCS ESC abuse)
SecurityEvent
| where EventID == 4887
| where TimeGenerated > ago(24h)
| extend
    Template         = tostring(EventData.TemplateName),
    Requester        = tostring(EventData.RequesterName),
    SubjectAltName   = tostring(EventData.SubjectAltName),
    CertSerialNumber = tostring(EventData.CertificateSerialNumber)
| where Template in ("Administrator", "DomainController", "SmartcardLogon")
    or Requester has_any ("admin", "DA-", "EA-", "krbtgt")
    or SubjectAltName has_any ("administrator@", "krbtgt@", "DC$@")
| project TimeGenerated, Computer, Template, Requester, SubjectAltName, CertSerialNumber
| order by TimeGenerated desc

// Détection: pic anormal d'émission de certificats (automatisation malveillante possible)
SecurityEvent
| where EventID == 4887
| where TimeGenerated > ago(1h)
| summarize CertCount = count() by bin(TimeGenerated, 5m), Computer
| where CertCount > 20
| extend Alert = strcat("Volume anormal: ", CertCount, " certs en 5 min")
| project TimeGenerated, Computer, CertCount, Alert
| order by CertCount desc
```

## Conformité NIS 2 et eIDAS 2 : PKI et Identité Numérique en France

Pour les organisations françaises soumises à NIS 2 — opérateurs de services essentiels (OES) et opérateurs d'importance vitale (OIV) — la PKI n'est plus un sujet optionnel mais une composante réglementaire obligatoire. La directive NIS 2, transposée en droit français via la loi du 17 avril 2025, impose une gestion rigoureuse des identités numériques et de la cryptographie.

## Exigences NIS 2 applicables à la PKI

**Article 21 NIS 2** : Mesures de gestion des risques incluant le chiffrement et la cryptographie comme mesures techniques obligatoires

**Politique de sécurité cryptographique** : Documentation des algorithmes utilisés, durées de vie des clés, procédures de renouvellement et de révocation

**Gestion des accès** : Principe du moindre privilège appliqué aux administrateurs CA et aux opérateurs PKI

**Journalisation** : Conservation des logs CA pendant au minimum 12 mois (art. 23 NIS 2)

**Tests de continuité** : Procédures de révocation et de reprise après incident testées au moins annuellement

**Notification d'incident** : En cas de compromission PKI, notification à l'ANSSI sous 24h (incident significatif)

## ANSSI RGS et Référentiel Cryptographique

Le Référentiel Général de Sécurité (RGS) de l'ANSSI impose des exigences spécifiques pour les PKI utilisées dans les échanges avec les administrations françaises :

**Algorithmes approuvés** : RSA minimum 2048 bits, ECDSA P-256/P-384, SHA-256/SHA-384 minimum

**Durée de vie Root CA** : Maximum 20 ans (RGS v2.0)

**Durée de vie Sub CA** : Maximum 10 ans

**Durée de vie certificats utilisateurs** : Maximum 3 ans

**Révocation OCSP** : Obligatoire pour toute PKI RGS qualifiée niveau standard ou élevé

**HSM** : Obligatoire pour les PKI qualifiées RGS niveau élevé (FIPS 140-2 Level 3 ou CC EAL 4+)

### eIDAS 2.0 et les certificats qualifiés européens

eIDAS 2.0 (règlement UE 2024/1183) introduit les European Digital Identity Wallets (EUDI Wallet). Pour les organisations françaises développant des services compatibles eIDAS 2 :

Les certificats qualifiés (QWAC, QSealC) doivent être émis par des Prestataires de Services de Confiance Qualifiés (PSCQ) référencés dans la trust list européenne — ADCS interne ne peut pas émettre ces certificats

En revanche, la PKI interne ADCS peut coexister avec des certificats eIDAS pour les services internes vs externes

L'interopérabilité exige l'alignement sur les profils de certificats définis dans ETSI EN 319 412

Pour les RSSI parisiens et des grandes métropoles françaises gérant des PKI pour des OES, la mise en conformité NIS 2 est l'opportunité de moderniser l'infrastructure PKI tout en renforçant la posture de sécurité globale. Les liens entre PKI, silos d'authentification Windows Server 2025 et Credential Guard forment un ensemble cohérent de protections complémentaires dans une architecture Zero Trust.

## Questions fréquentes sur ADCS et PKI sécurisée

### Quelle est la différence entre une Root CA offline et une Root CA en ligne dans ADCS ?

Une Root CA offline est physiquement déconnectée du réseau et démarrée uniquement lors d'opérations rares (renouvellement de CA subordonnée, tous les 10 ans environ). Une Root CA en ligne reste connectée au réseau et est donc exposée aux attaques réseau, aux ransomwares et aux compromissions à distance. L'ANSSI et le NIST recommandent systématiquement une Root CA offline pour toute PKI de production, car la compromission de la Root CA anéantit l'ensemble de la chaîne de confiance PKI — tous les certificats émis deviendraient falsifiables. La Root CA offline peut être stockée sur un serveur physique dans un coffre-fort, avec le disque dur chiffré par BitLocker et protégé par TPM.

### Comment corriger les vulnérabilités ESC1 à ESC8 sur mon infrastructure ADCS ?

La correction des vulnérabilités ESC nécessite plusieurs actions prioritaires : premièrement, désactiver le flag EDITF_ATTRIBUTESUBJECTALTNAME2 via certutil -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2 (ESC6) ; deuxièmement, supprimer le flag CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT des templates avec Client Auth EKU (ESC1) ; troisièmement, activer l'Extended Protection for Authentication sur IIS pour les endpoints d'enrôlement (ESC8) ; quatrièmement, auditer et corriger les ACL des templates pour supprimer les droits Write des groupes non administrateurs (ESC4). Utilisez PSPKIAudit ou Certipy en mode audit pour identifier toutes les vulnérabilités de votre environnement avant de procéder aux corrections. Planifiez ces audits trimestriellement.

### Quelle est la durée de vie recommandée pour les certificats ADCS selon l'ANSSI ?

L'ANSSI recommande via le RGS v2.0 les durées suivantes : Root CA 20 ans maximum (clé RSA 4096 bits ou ECDSA P-384 obligatoire), CA subordonnée 10 ans maximum, certificats utilisateurs et machines 3 ans maximum, certificats TLS serveur 1 an recommandé (aligné sur les pratiques des navigateurs qui ont réduit la validité maximale à 398 jours en 2020). Pour les certificats NIS 2 et les PKI utilisées avec des administrations françaises, respectez impérativement le RGS v2.0 disponible sur le site de l'ANSSI. Évitez absolument les certificats de 5 ou 10 ans pour les utilisateurs finaux — un compte compromis avec un certificat de 10 ans est une porte ouverte durant toute cette période.

### Comment configurer OCSP pour réduire la charge CRL dans une grande infrastructure PKI ?

L'Online Certificate Status Protocol (OCSP) permet de vérifier le statut d'un certificat individuel sans télécharger la CRL complète (qui peut peser plusieurs mégaoctets dans les grandes organisations). Pour le déployer : installez le rôle Online Responder (ADCS-Online-Cert) sur un serveur IIS dédié ou partagé, créez un template de certificat OCSP Response Signing (durée 7

jours, auto-renouvellement activé), configurez une Revocation Configuration pointant vers votre Issuing CA, ajoutez l'URL OCSP dans l'AIA de vos templates avec Add-CAAuthorityInformationAccess -AddToCertificateOcsp. La combinaison CRL (validité 7 jours) + delta CRL (validité 24h) + OCSP offre la meilleure résilience. Vérifiez le statut OCSP avec certutil -verify -urlfetch certificate.cer.

### La PKI ADCS interne est-elle compatible avec les exigences NIS 2 et eIDAS 2 ?

Une PKI ADCS interne bien configurée répond aux exigences NIS 2 concernant la gestion des identités et la cryptographie (article 21). Elle ne peut cependant pas émettre de certificats qualifiés au sens eIDAS 2 (QWAC pour l'authentification web, QSealC pour les cachets électroniques qualifiés) — seuls les Prestataires de Services de Confiance Qualifiés (PSCQ) référencés dans la trust list européenne peuvent le faire. Pour les OES et OIV français, l'ADCS interne doit être alignée sur le RGS ANSSI, avec audit annuel documenté, procédures formelles d'opération Root CA, et journalisation conservée 12 mois minimum. En cas de doute sur le périmètre de conformité NIS 2 applicable à votre PKI, consultez notre guide d'audit des groupes privilégiés AD.

**À RETENIR**

Points clés à retenir

La Root CA offline est non négociable : elle ne doit jamais être connectée au réseau de production et doit être physiquement sécurisée avec BitLocker et idéalement un HSM certifié FIPS 140-2.

Auditez immédiatement vos templates ADCS pour les vulnérabilités ESC1-ESC8 avec PSPKIAudit — plus de 70 % des infrastructures AD testées sont vulnérables à au moins ESC1 ou ESC6.

Activez l'audit complet ADCS (AuditFilter 127) et corréllez les Event IDs 4886/4887/4890 dans votre SIEM pour détecter les abus de certificats en temps réel.

> La conformité NIS 2 impose une politique de certification documentée, des procédures de révocation testées annuellement, et une conservation des logs minimum 12 mois.
>
> Planifiez des audits PSPKIAudit trimestriels pour surveiller toute dérive de configuration PKI, notamment après chaque création ou modification de template de certificat.

Pour approfondir la sécurisation de votre environnement Active Directory, consultez nos guides sur les événements Windows à surveiller sur un contrôleur de domaine et sur l'audit des groupes privilégiés Active Directory. Ces ressources constituent avec le présent guide un ensemble cohérent pour une sécurité AD de haut niveau.

## Sources


ANSSI CERT-FR — Publications et alertes

MITRE ATT&CK — Framework des techniques d'attaque

CISA — Advisories de cybersécurité