Vérification des Mots de Passe AD avec Have I Been Pwned : Guide Complet 2026

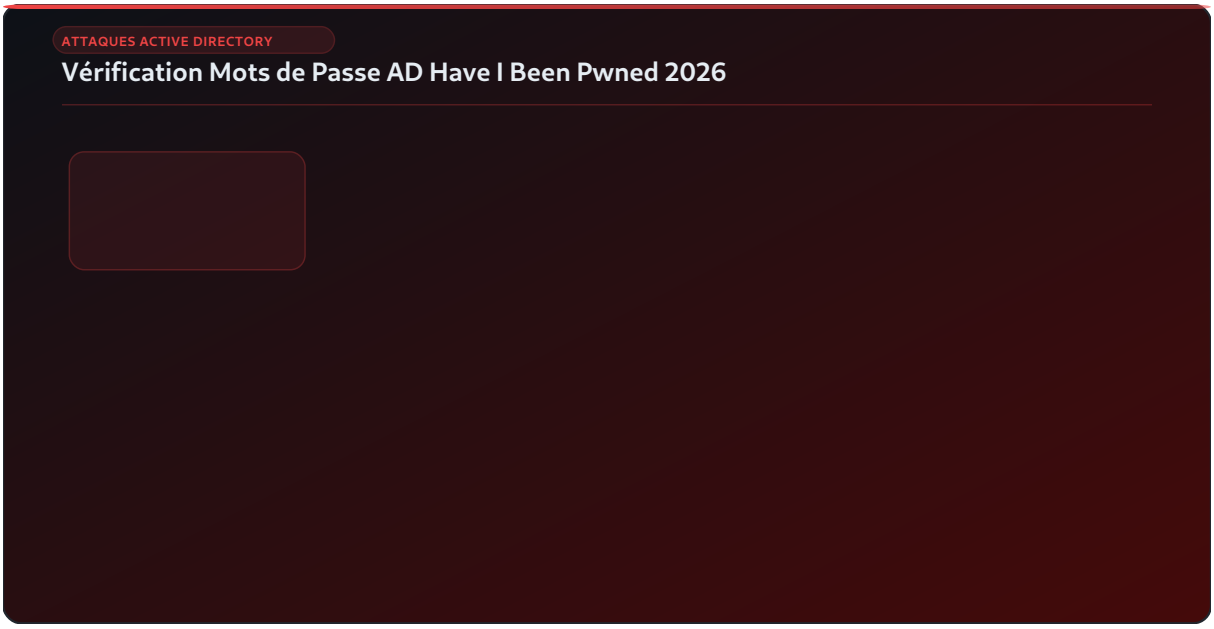
Auditez les mots de passe Active Directory compromis avec Have I Been Pwned, DSInternals et PowerShell. Guide complet RGPD et NIS 2 pour les RSSI français.

📋 EN BREF

En bref

- Les mots de passe Active Directory compromis représentent le vecteur d'attaque n°1 en France en 2025-2026, selon l'ANSSI et les données CERT-FR.
- Have I Been Pwned offre une API k-Anonymity : seuls les 5 premiers caractères du hash SHA-1 transitent sur le réseau, garantissant la confidentialité.
- DSInternals permet d'extraire les hash NTLM du domaine AD et de les comparer avec la base HIBP sans exposer aucun mot de passe en clair.
- Un audit offline complet est possible en téléchargeant la base NTLM HIBP (~12 Go), idéal pour les environnements classifiés ou OT/IT isolés.
- La conformité RGPD et NIS 2 impose de documenter les audits de mots de passe et d'informer les utilisateurs dans la politique de sécurité.

En 2025 et 2026, les violations de données liées à des identifiants compromis touchent massivement les organisations françaises. Les RSSI, DSI et équipes de sécurité font face à un défi quotidien : comment savoir si les mots de passe utilisés dans leur Active Directory ont été exposés dans des fuites publiques ? Sans cet audit, une entreprise parisienne peut opérer des mois avec des comptes Active Directory dont les identifiants circulent librement sur des forums cybercriminels et des bases de données vendues sur le dark web. L'ANSSI, dans son rapport 2025, identifie le credential stuffing et le password spraying comme les deux techniques d'intrusion initiale les plus fréquentes dans les incidents traités par le CERT-FR. La réglementation NIS 2, transposée en droit français, et le RGPD Article 32 obligent désormais les entités essentielles et importantes à mettre en place des mesures techniques robustes pour protéger les données d'authentification. Ce guide technique complet explique comment mettre en place un audit systématique des mots de passe Active Directory en utilisant Have I Been Pwned (HIBP), DSInternals, et des scripts PowerShell prêts à l'emploi, le tout dans le respect du cadre légal français encadré par la CNIL et le RGPD. Que vous soyez RSSI d'une ETI francilienne, administrateur système dans un établissement de santé soumis à NIS 2, ou consultant en sécurité offensive, ce guide vous donnera toutes les clés pour un programme d'hygiène des mots de passe Active Directory robuste et conforme.



Pourquoi Vérifier les Mots de Passe AD contre les Fuites de Données ?

La réutilisation des mots de passe est endémique. Selon une étude Verizon DBIR 2025, 74 % des violations de données impliquent des identifiants volés ou réutilisés. En France, le CERT-FR a traité 1 412 incidents de sécurité en 2025, dont une proportion croissante liée à des campagnes d'authentification par force brute ou credential stuffing ciblant des organisations françaises utilisant Active Directory.

Le credential stuffing consiste à tester automatiquement des couples identifiant/mot de passe extraits de fuites de données sur des portails d'entreprise, des VPN, des accès OWA (Outlook Web Access) ou des RDP exposés. Si un utilisateur du service comptabilité utilise le même mot de passe pour son compte LinkedIn (fuité en 2024) et son compte de domaine Active Directory, l'attaquant obtient un premier pied dans le réseau en quelques secondes.

Le password spraying est différent : l'attaquant utilise un seul mot de passe très commun (ex : `Printemps2025!`, `Paris2025@`) et le teste sur tous les comptes du domaine, contournant les politiques de verrouillage qui se déclenchent après X tentatives par compte. Ces deux techniques sont particulièrement efficaces contre des domaines AD qui n'ont pas de contrôle proactif sur la qualité des mots de passe.

Statistiques France 2025-2026

+340 % d'augmentation des incidents de type credential stuffing ciblant des organisations françaises entre 2023 et 2025 (source : CERT-FR)

12,5 milliards de couples identifiant/mot de passe disponibles dans la base HIBP au premier trimestre 2026

83 % des PME françaises n'ont aucun processus d'audit régulier des mots de passe Active Directory (Baromètre CESIN 2025)

47 jours : temps moyen de détection d'une compromission initiale via des credentials volés en France (IBM Cost of Data Breach Report 2025)

La CNIL a prononcé 89 mises en demeure en 2025 pour défaut de mesures techniques de protection des données d'authentification

Face à ces chiffres, l'audit régulier des mots de passe AD contre les bases de fuites publiques n'est plus une option : c'est une mesure de sécurité fondamentale, exigée par NIS 2 et attendue par les assureurs cyber qui conditionnent leurs couvertures à la mise en place de telles procédures.

Have I Been Pwned : Comment Fonctionne l'API k-Anonymity ?

Have I Been Pwned (HIBP), créé par Troy Hunt, est la référence mondiale des bases de données de fuites d'identifiants. Sa base Pwned Passwords contient plus de 900 millions de hash de mots de passe uniques issus de fuites réelles. La question que se posent immédiatement les RSSI est légitime : "Est-ce que j'envoie mes mots de passe à un service externe ?" La réponse est non, grâce au protocole k-Anonymity.

⚡ **Retour terrain**

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

technique en sécurité AD est souvent invisible jusqu'au premier audit sérieux.

Le Protocole k-Anonymity Expliqué

Le principe est le suivant :

Vous calculez le hash SHA-1 du mot de passe à vérifier : `SHA1("MotDePasse123") = 5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8`

Vous extrayez uniquement les **5 premiers caractères** (le préfixe) : `5BAA6`

Vous envoyez ce préfixe à l'API : `GET https://api.pwnedpasswords.com/range/5BAA6`

L'API retourne **tous les suffixes** de hash commençant par `5BAA6` (environ 400 à 900 résultats)

Vous comparez **localement** si le suffixe de votre hash complet (`1E4C9B93F3F0682250B6CF8331B7EE68FD8`) apparaît dans la liste retournée

L'API ne reçoit jamais le hash complet, encore moins le mot de passe en clair. Avec 5 caractères hexadécimaux, il existe 16^5 = 1 048 576 préfixes possibles, ce qui rend la réidentification du mot de passe original mathématiquement impraticable depuis le côté serveur.

Exemple PowerShell : Vérification d'un Mot de Passe via k-Anonymity

```
function Test-PasswordWithHIBP {
    param([string]$Password)

    # Calcul du hash SHA-1
    $sha1 = [System.Security.Cryptography.SHA1]::Create()
    $bytes = [System.Text.Encoding]::UTF8.GetBytes($Password)
    $hashBytes = $sha1.ComputeHash($bytes)
    $hashHex = [BitConverter]::ToString($hashBytes) -replace '-', ''

    $prefix = $hashHex.Substring(0, 5)
    $suffix = $hashHex.Substring(5)

    # Requête k-Anonymity
    try {
        $response = Invoke-RestMethod -Uri "https://api.pwnedpasswords.com/range/$prefix" `
            -Method Get -UserAgent "ADPasswordAudit/1.0"

        $lines = $response -split "`n"
        foreach ($line in $lines) {
            $parts = $line.Trim() -split ':'
            if ($parts[0] -eq $suffix.ToUpper()) {
                return @{ Pwned = $true; Count = [int]$parts[1] }
            }
        }
        return @{ Pwned = $false; Count = 0 }
    }
    catch {
        Write-Warning "Erreur API HIBP: $_"
        return @{ Pwned = $false; Count = -1; Error = $true }
    }
}

# Test
$result = Test-PasswordWithHIBP -Password "Azerty123!"
if ($result.Pwned) {
    Write-Host "COMPROMIS - Apparu $($result.Count) fois dans des fuites" -ForegroundColor Red
} else {
    Write-Host "Non trouvé dans la base HIBP" -ForegroundColor Green
}
```

Cette approche est non seulement sécurisée mais également **approuvée par le NIST SP 800-63B** qui recommande explicitement de vérifier les mots de passe contre les bases de fuites connues lors de la création ou du changement de mot de passe.

DSInternals : Audit des Hash NTLM Active Directory

DSInternals (Directory Services Internals) est un module PowerShell open source développé par Michael Grafnetter, permettant d'interagir avec les données internes d'Active Directory, notamment d'extraire les hash NTLM des comptes pour un audit de sécurité. Il est utilisé par les équipes Red Team et les auditeurs de sécurité comme outil légitime de test d'intrusion et d'audit.

Pour le contexte des articles liés à la sécurité AD, voir notre guide sur les événements Windows à surveiller sur les contrôleurs de domaine pour comprendre comment tracer ces opérations d'audit.

Installation et Prérequis

```
 # Installation depuis PowerShell Gallery (exécuter en tant qu'administrateur)
Install-Module -Name DSInternals -Scope AllUsers -Force

# Vérification de l'installation
Import-Module DSInternals
Get-Command -Module DSInternals | Measure-Object
```

Permissions Requises

Pour utiliser `Get-ADReplAccount`, le compte d'exécution doit disposer du droit **DS-Replication-Get-Changes-All** (GUID : 1131f6ad-9c07-11d1-f79f-00c04fc2dcd2).

Ce droit est accordé par défaut aux groupes Domain Admins, Enterprise Admins et SYSTEM. Pour créer un compte de service dédié moins privilégié :

```
 # Créer un compte de service dédié à l'audit (à faire par un Domain Admin)
New-ADUser -Name "svc-password-audit" `
    -SamAccountName "svc-password-audit" `
    -AccountPassword (ConvertTo-SecureString "P@ssw0rd-Audit-2026!" -AsPlainText -Force) `
    -PasswordNeverExpires $true `
    -Enabled $true

# Accorder uniquement les droits de réplication nécessaires
$rootDSE = Get-ADRootDSE
$domainDN = $rootDSE.defaultNamingContext
$replicationRights = @(
    "1131f6ad-9c07-11d1-f79f-00c04fc2dcd2",  # DS-Replication-Get-Changes-All
    "89e95b76-444d-4c62-991a-0facbeda640c"   # DS-Replication-Get-Changes-In-Filtered-Set
)
# Ces droits doivent être configurés via l'éditeur ADSI ou dsacls
```

Extraction et Audit Complet des Hash NTLM

```
 Import-Module DSInternals
Import-Module ActiveDirectory

# Récupérer la Boot Key du contrôleur de domaine (nécessite accès local au DC)
$bootKey = Get-BootKey -Online

# Extraire tous les comptes avec leurs hash NTLM via réplication
$allAccounts = Get-ADReplAccount -All -Server "DC01.mondomaine.local" -NamingContext "DC=mondor

# Filtrer les comptes actifs uniquement
$activeAccounts = $allAccounts | Where-Object {
    $_.Enabled -eq $true -and
    $_.SamAccountType -eq "User" -and
    $_.NTHash -ne $null
}

Write-Host "Comptes actifs avec hash NTLM: $($activeAccounts.Count)"

# Analyse de qualité des mots de passe avec DSInternals
$passwordReport = $allAccounts | Test-PasswordQuality -WeakPasswordHashesSortedFile "C:\Audit\

# Afficher le rapport
$passwordReport
```

La commande `Test-PasswordQuality` génère automatiquement un rapport complet incluant : comptes avec mots de passe vides, mots de passe partagés entre plusieurs comptes (même hash NTLM), mots de passe trouvés dans la liste de compromission, et comptes avec attributs problématiques (PasswordNotRequired, PasswordNeverExpires sur des comptes sensibles).

Consultez notre article sur le groupe Protected Users Active Directory pour comprendre comment protéger les comptes privilégiés dont les hash sont les plus critiques à sécuriser.

HIBP Offline NT Hash Database : Audit Local Sans Envoi Réseau

Pour les environnements sensibles — réseaux industriels OT/IT, systèmes classifiés Défense, établissements de santé avec données hébergées en local — la méthode offline est préférable. HIBP propose le téléchargement complet de sa base de hash NTLM (format NT Hash) pour des audits entièrement déconnectés d'internet.

Téléchargement de la Base NTLM HIBP

```
 # Téléchargement via Pwned Passwords Downloader (outil officiel HIBP)
# Source: https://github.com/HaveIBeenPwned/PwnedPasswordsDownloader

# Installation de l'outil
dotnet tool install --global haveibeenpwned-downloader

# Téléchargement de la liste NTLM (format NT Hash, ~12 GB décompressé)
# L'outil télécharge en parallèle et reprend en cas d'interruption
haveibeenpwned-downloader -t 64 -n C:\Audit\hibp-ntlm-sorted.txt

# Ou via PowerShell avec aria2c pour les grands volumes
aria2c -x 16 -s 16 "https://downloads.pwnedpasswords.com/passwords/pwned-passwords-ntlm-ordered
    -d "C:\Audit" -o "hibp-ntlm.7z"
```

Le fichier téléchargé contient les hash NTLM triés par ordre de fréquence (du plus au moins fréquent), un format optimisé pour une recherche binaire efficace. Chaque ligne est au format : `NTLMHASH:COUNT`

Script PowerShell Complet : Audit Offline des Mots de Passe AD

```
 #Requires -Modules DSInternals, ActiveDirectory
#Requires -RunAsAdministrator

<#
.SYNOPSIS
    Audit offline des mots de passe Active Directory contre la base HIBP NTLM
.DESCRIPTION
    Extrait les hash NTLM via DSInternals et compare avec la base HIBP locale
.NOTES
    Nécessite : DSInternals, droits DS-Replication-Get-Changes-All
    RGPD : résultats à traiter comme données confidentielles, durée de conservation limitée
#>

param(
    [Parameter(Mandatory=$true)]
    [string]$DomainController,

    [Parameter(Mandatory=$true)]
    [string]$NamingContext,

    [Parameter(Mandatory=$true)]
    [string]$HibpNtlmFile,

    [string]$ReportPath = "C:\Audit\password-audit-$(Get-Date -Format 'yyyyMMdd-HHmm').html",

    [switch]$ForcePasswordReset,

    [string[]]$ExcludedAccounts = @("krbtgt", "Guest", "Administrator")
)

Import-Module DSInternals -ErrorAction Stop
Import-Module ActiveDirectory -ErrorAction Stop

$startTime = Get-Date
Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Démarrage de l'audit HIBP offline..." -Foreground

# Vérification du fichier HIBP
if (-not (Test-Path $HibpNtlmFile)) {
    throw "Fichier HIBP NTLM introuvable : $HibpNtlmFile"
}

$hibpFileSize = (Get-Item $HibpNtlmFile).Length / 1GB
Write-Host "[*] Base HIBP : $([math]::Round($hibpFileSize, 2)) GB" -ForegroundColor Gray
```

```
# Extraction des comptes AD
Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Extraction des hash NTLM via réplication AD..." -F

try {
    $allAccounts = Get-ADReplAccount -All -Server $DomainController -NamingContext $NamingConte
}
catch {
    Write-Error "Erreur lors de l'extraction AD : $_"
    exit 1
}

$activeUsers = $allAccounts | Where-Object {
    $_.Enabled -eq $true -and
    $_.SamAccountType -eq "User" -and
    $_.NTHash -ne $null -and
    $_.SamAccountName -notin $ExcludedAccounts
}

Write-Host "[*] Comptes utilisateurs actifs à vérifier : $($activeUsers.Count)" -ForegroundColo

# Chargement de la base HIBP en mémoire (HashSet pour O(1) lookup)
Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Chargement de la base HIBP..." -ForegroundColor Cy
$hibpHashes = [System.Collections.Generic.HashSet[string]]::new([StringComparer]::OrdinalIgnore

$reader = [System.IO.StreamReader]::new($HibpNtlmFile)
$lineCount = 0
while (($line = $reader.ReadLine()) -ne $null) {
    $hash = $line.Split(':')[0]
    [void]$hibpHashes.Add($hash)
    $lineCount++
    if ($lineCount % 10000000 -eq 0) {
        Write-Host "  Chargé : $($lineCount / 1000000)M entrées..." -ForegroundColor Gray
    }
}
$reader.Close()
Write-Host "[*] Base HIBP chargée : $lineCount hash uniques" -ForegroundColor Gray

# Comparaison des hash NTLM
Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Comparaison des hash NTLM..." -ForegroundColor Cya

$compromisedAccounts = @()
$cleanAccounts = @()
```

```
foreach ($account in $activeUsers) {
    $ntHashHex = [BitConverter]::ToString($account.NTHash) -replace '-', ''

    if ($hibpHashes.Contains($ntHashHex)) {
        $compromisedAccounts += [PSCustomObject]@{
            SamAccountName    = $account.SamAccountName
            DisplayName       = $account.DisplayName
            Email             = $account.EmailAddress
            LastLogonDate     = $account.LastLogonDate
            DistinguishedName = $account.DistinguishedName
            NTHashPrefix      = $ntHashHex.Substring(0, 8) + "..."  # Tronqué pour le rapport
        }
    } else {
        $cleanAccounts += $account.SamAccountName
    }
}

Write-Host "[!] Comptes avec mots de passe compromis : $($compromisedAccounts.Count)" -Foregrou
Write-Host "[+] Comptes avec mots de passe sains     : $($cleanAccounts.Count)" -ForegroundCol

# Forcer le changement de mot de passe si demandé
if ($ForcePasswordReset -and $compromisedAccounts.Count -gt 0) {
    Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Activation du flag ChangePasswordAtLogon..." 
    foreach ($account in $compromisedAccounts) {
        try {
            Set-ADUser -Identity $account.SamAccountName -ChangePasswordAtLogon $true
            Write-Host "  [+] Flag activé pour : $($account.SamAccountName)" -ForegroundColor Y
        }
        catch {
            Write-Warning "  Erreur pour $($account.SamAccountName) : $_"
        }
    }
}

# Génération du rapport HTML
$duration = (Get-Date) - $startTime
$reportHtml = @"
<!DOCTYPE html>
<html lang="fr">



<body>
```

Date d'audit : $(Get-Date -Format 'dd/MM/yyyy HH:mm')

Contrôleur de domaine : $DomainController

Durée de l'audit : $([math]::Round($duration.TotalMinutes, 1)) minutes

Comptes vérifiés : $($activeUsers.Count)

Comptes compromis : $($compromisedAccounts.Count)

Taux de compromission : $([math]::Round($compromisedAccounts.Count / $activeUsers.Count * 10

Comptes avec Mots de Passe Compromis

$(foreach ($acc in $compromisedAccounts) {
""
})

Compte	Nom affiché	Email	Dernière connexion
$($acc.SamAccountName)	$($acc.DisplayName)	$($acc.Email)	$($acc.LastLogonDate)

</body>

```
</html>
"@

$reportHtml | Out-File -FilePath $ReportPath -Encoding UTF8
Write-Host "[$(Get-Date -Format 'HH:mm:ss')] Rapport généré : $ReportPath" -ForegroundColor Cya
```

Utilisation du Script

```
 # Audit standard avec rapport HTML
.\Invoke-HibpAudit.ps1 `
    -DomainController "DC01.mondomaine.local" `
    -NamingContext "DC=mondomaine,DC=local" `
    -HibpNtlmFile "C:\Audit\hibp-ntlm-sorted.txt" `
    -ReportPath "C:\Audit
apport-$(Get-Date -Format 'yyyyMMdd').html"

# Audit avec forçage du changement de mot de passe
.\Invoke-HibpAudit.ps1 `
    -DomainController "DC01.mondomaine.local" `
    -NamingContext "DC=mondomaine,DC=local" `
    -HibpNtlmFile "C:\Audit\hibp-ntlm-sorted.txt" `
    -ForcePasswordReset `
    -ExcludedAccounts @("krbtgt", "Guest", "svc-backup", "svc-monitoring")
```

Déploiement via GPO : Audit Planifié des Comptes Compromis

Pour automatiser l'audit sur une base régulière (hebdomadaire ou mensuelle), la GPO offre un déploiement centralisé sans avoir à se connecter manuellement sur chaque contrôleur de domaine. L'automatisation permet également d'envoyer des alertes email aux utilisateurs concernés et à leur responsable.

Configuration du Task Scheduler via GPO

```
 # Script de déploiement GPO pour planifier l'audit mensuel
# À exécuter dans un script de démarrage d'ordinateur (Computer Configuration)

$taskAction = New-ScheduledTaskAction `
    -Execute "PowerShell.exe" `
    -Argument "-NonInteractive -ExecutionPolicy Bypass -File C:\Scripts\Invoke-HibpAudit.ps1 -D

$taskTrigger = New-ScheduledTaskTrigger `
    -Weekly `
    -DaysOfWeek Monday `
    -At "02:00"

$taskSettings = New-ScheduledTaskSettingsSet `
    -RunOnlyIfNetworkAvailable `
    -StartWhenAvailable `
    -ExecutionTimeLimit (New-TimeSpan -Hours 4)

$taskPrincipal = New-ScheduledTaskPrincipal `
    -UserId "MONDOMAINE\svc-password-audit" `
    -LogonType Password `
    -RunLevel Highest

Register-ScheduledTask `
    -TaskName "HIBP-PasswordAudit-Weekly" `
    -TaskPath "\Securite\Audits" `
    -Action $taskAction `
    -Trigger $taskTrigger `
    -Settings $taskSettings `
    -Principal $taskPrincipal `
    -Description "Audit hebdomadaire des mots de passe AD vs base HIBP NTLM"
```

Template Email d'Alerte Utilisateur

```
function Send-CompromisedPasswordAlert {
    param(
        [string]$UserEmail,
        [string]$DisplayName,
        [string]$SMTPServer = "smtp.mondomaine.local"
    )

    $subject = "[URGENT - Sécurité] Votre mot de passe nécessite un changement immédiat"

    $body = @"
Bonjour $DisplayName,

Notre équipe de sécurité a détecté que votre mot de passe de domaine
apparaît dans des bases de données de mots de passe compromis publiquement connues.

CELA NE SIGNIFIE PAS que votre compte a été piraté. Cela signifie que le mot de passe
que vous utilisez a déjà été exposé lors d'une fuite de données (sur un autre service,
une ancienne plateforme, etc.) et pourrait être utilisé par des attaquants.

ACTION REQUISE :
→ Connectez-vous à votre poste de travail
→ Appuyez sur Ctrl+Alt+Suppr → "Modifier un mot de passe"
→ Choisissez un mot de passe unique, long (12 caractères minimum),
  que vous n'utilisez nulle part ailleurs
→ Utilisez un gestionnaire de mots de passe (KeePass, Bitwarden) pour
  mémoriser vos mots de passe

ATTENTION : Si vous ne changez pas votre mot de passe dans les 48 heures,
votre session sera automatiquement bloquée à la prochaine connexion.

En cas de question, contactez le support informatique : support@mondomaine.fr

Cordialement,
L'équipe Sécurité Informatique

---
Cet email a été généré automatiquement dans le cadre de notre programme de
sécurisation des accès conformément à notre politique de sécurité et au RGPD.
"@

    Send-MailMessage `
        -To $UserEmail `
        -From "securite-noreply@mondomaine.fr" `
```

```
        -Subject $subject `
        -Body $body `
        -SmtpServer $SMTPServer `
        -Encoding UTF8
}
```

La mise en place de cette automatisation, combinée avec notre guide sur les Authentication Silos Windows Server 2025, permet d'isoler les comptes à haut risque pendant la période de remédiation.

Outils Alternatifs : PurpleKnight, Spectre, L0phtCrack

Au-delà de DSInternals, plusieurs outils du marché proposent des fonctionnalités d'audit de mots de passe AD, chacun avec ses avantages et ses limites.

Comparatif des Outils

Outil	Prix	Vérification HIBP	Interface	Rapport	Recommandé pour
DSInternals	Gratuit (open source)	Oui (offline/ online)	PowerShell	Personnalisable	Experts, Red Team, RSSI
PurpleKnight	Gratuit (Semperis)	Non natif	GUI Windows	HTML/PDF	Audit global AD, direction
Spectre	Commercial	Oui (intégré)	Web dashboard	Complet + alertes	Grandes entreprises, MSP
L0phtCrack 7	Commercial	Non	GUI	Oui	Tests de robustesse (rainbow tables)
Lithnet Password Protection	Gratuit (open source)	Oui (HIBP offline)	PowerShell + GPO	Event Logs	Intégration AD native, prévention

PurpleKnight de Semperis se distingue par son approche orientée tableau de bord pour la direction. Il évalue plus de 140 indicateurs de sécurité AD (délégations Kerberos non contraintes, SPN sur comptes privilégiés, AdminSDHolder, etc.) mais ne vérifie pas nativement contre HIBP — il faut combiner avec DSInternals.

Lithnet Password Protection est particulièrement intéressant car il s'intègre directement dans la politique de mots de passe AD (Password Filter DLL) et bloque en temps réel les mots de passe compromis lors de leur création ou modification, en comparant avec la base HIBP locale.

Intégration dans le Cycle IAM : Azure AD Password Protection

Microsoft Entra ID Password Protection (anciennement Azure AD Password Protection) permet d'étendre la protection des mots de passe à l'environnement hybride on-premises. Cette solution s'intègre directement dans les contrôleurs de domaine AD pour bloquer les mots de passe faibles, qu'ils soient issus de la liste Microsoft ou de votre liste personnalisée.

Pour sécuriser davantage les accès privilégiés, consultez notre article sur les Privileged Access Workstations (PAW) 2026 qui complète cette approche.

Architecture de Déploiement

Le déploiement se fait en deux composants :

AzureADPasswordProtectionProxy : installé sur un serveur membre (pas forcément un DC), télécharge les politiques depuis Microsoft Entra ID

AzureADPasswordProtectionDCAgent : installé sur chaque DC, implémente le filtre de mots de passe et journalise les événements

Installation et Configuration

```
 # Sur le serveur proxy (membre du domaine, accès internet)
Install-Module -Name AzureADPasswordProtection -Force

# Inscription du proxy auprès de Microsoft Entra ID
Import-Module AzureADPasswordProtection
Register-AzureADPasswordProtectionProxy -AccountUpn "admin@mondomaine.onmicrosoft.com"

# Inscription de la forêt AD
Register-AzureADPasswordProtectionForest -AccountUpn "admin@mondomaine.onmicrosoft.com"

# Sur chaque contrôleur de domaine (via GPO ou manuellement)
# Installer AzureADPasswordProtectionDCAgent.msi
# Le service démarre automatiquement après reboot du DC
```

Configuration d'une Liste de Mots de Passe Interdits Personnalisée

Via le portail Microsoft Entra ID (entra.microsoft.com) :

Sécurité → Méthodes d'authentification → Protection par mot de passe

Activer "Appliquer la liste personnalisée"

Ajouter vos termes métier spécifiques (nom de l'entreprise, ville, secteur)

Exemples de termes à ajouter pour une organisation française typique :

```
 monentreprise
Paris
Lyon
Marseille
france
janvier
fevrier
# ... tous les mois de l'année
saison2025
Q12025
```

Event IDs Azure AD Password Protection

Event ID	Source	Description
10014	AzureADPasswordProtectionDCAgent	Mot de passe rejeté en mode Enforced (trop faible)
10015	AzureADPasswordProtectionDCAgent	Mot de passe rejeté car dans la liste personnalisée
10016	AzureADPasswordProtectionDCAgent	Mot de passe rejeté car dans la liste globale Microsoft
30006	AzureADPasswordProtectionDCAgent	Nouvelle politique téléchargée depuis le proxy
30010	AzureADPasswordProtectionProxy	Proxy enregistré avec succès

Ces événements doivent être intégrés dans votre SIEM pour générer des alertes lors de pics de rejets (possible tentative de password spraying détectée à la source).

```
 # Surveiller les rejets ADPP en temps réel
Get-WinEvent -LogName "Microsoft-AzureADPasswordProtection-DCAgent/Admin" |
    Where-Object { $_.Id -in @(10014, 10015, 10016) } |
    Select-Object TimeCreated, Id, Message |
    Sort-Object TimeCreated -Descending |
    Select-Object -First 50
```

Pour les contrôleurs de domaine, notre guide sur Credential Guard Windows Server 2025 explique comment protéger les secrets LSA au niveau du noyau, complément indispensable à la politique de mots de passe.

Conformité RGPD et NIS 2 : Aspects Légaux en France

L'audit des mots de passe Active Directory en France s'inscrit dans un cadre légal précis. Bien exécuté, il est non seulement légal mais *requis* par la réglementation. Mal documenté, il peut exposer l'organisation à des sanctions de la CNIL.

RGPD Article 32 : Mesures Techniques de Sécurité

L'Article 32 du RGPD impose aux responsables de traitement de mettre en place des "mesures techniques et organisationnelles appropriées afin de garantir un niveau de sécurité adapté au risque." La CNIL, dans ses recommandations sur les mots de passe (délibération du 17 juin 2022), précise que :

Les mots de passe doivent être stockés sous forme de hash salé (bcrypt, Argon2) — en AD, les hash NTLM sont un héritage legacy à isoler

La vérification régulière contre les bases de fuites connues est une "bonne pratique" explicitement recommandée

Les résultats d'audit contenant des informations sur des comptes nominatifs sont des données personnelles et doivent être protégés en conséquence

NIS 2 Article 21 : Mesures de Gestion des Risques

La directive NIS 2, transposée par ordonnance en France en octobre 2024, impose aux entités essentielles (EE) et entités importantes (EI) de mettre en place des "mesures de gestion des risques en matière de cybersécurité", incluant explicitement :

La gestion des accès et l'authentification forte (Art. 21.2.i)

La sécurité des ressources humaines, politiques de contrôle d'accès (Art. 21.2.c)

L'utilisation de la cryptographie (Art. 21.2.h) — applicable à la gestion des secrets AD

L'ANSSI recommande dans son guide "Sécuriser Active Directory" (ANSSI-PA-070) de réaliser au minimum un audit annuel de la qualité des mots de passe du domaine, et de le documenter dans le plan de traitement des risques.

Les Hash NTLM Sont-ils des Données Personnelles ?

C'est une question complexe en droit français. En principe, un hash NTLM sans le mot de passe en clair correspondant ne permet pas d'identifier directement une personne (il identifie un compte). Cependant :

Le hash NTLM est associé au SamAccountName, qui est lui-même une donnée personnelle (identifiant nominatif)

La CNIL considère que les données qui permettent indirectement l'identification d'une personne restent des données personnelles

En pratique, traitez toujours les résultats d'audit (listes de comptes compromis, hash) comme des données confidentielles

Documentation et Traçabilité des Audits

Pour être conforme, chaque audit doit être documenté avec :

Base légale : intérêt légitime (Art. 6.1.f RGPD) — sécurisation du système d'information

Finalité : détection et remédiation des comptes avec mots de passe compromis

Durée de conservation : durée limitée à la remédiation (recommandation : 30 jours maximum)

Accès : limité aux administrateurs habilités (RSSI, équipe sécurité)

Information des utilisateurs : mention dans la politique de sécurité, le règlement intérieur ou la charte informatique

Journalisation : tous les accès aux données d'audit doivent être tracés (EventID 4662 en AD)

```
 # Journaliser le début et la fin de chaque audit dans l'observateur d'événements
$log = New-Object System.Diagnostics.EventLog("Application")
$log.Source = "HibpPasswordAudit"

# Début d'audit
$log.WriteEntry(
    "Audit HIBP démarré par $env:USERNAME sur $env:COMPUTERNAME - " +
    "Conformité RGPD Art.32 / NIS2 Art.21 - Domaine: $DomainController",
    [System.Diagnostics.EventLogEntryType]::Information,
    1001
)
```

Questions Fréquentes sur la Vérification des Mots de Passe Active Directory

Est-il légal d'auditer les mots de passe des utilisateurs Active Directory en France ?

Oui, sous conditions. L'audit des hash NTLM des comptes AD est légalement encadré par le RGPD (Art. 32) et la NIS 2 (Art. 21). Le responsable de traitement doit documenter la procédure dans son registre de traitement, informer les utilisateurs dans la politique de sécurité et le règlement intérieur, et veiller à ce que seuls les administrateurs habilités accèdent aux résultats. Les hash NTLM ne constituent pas des données personnelles exploitables directement, mais le processus doit respecter le principe de minimisation des données. La CNIL recommande de ne conserver les rapports d'audit que le temps nécessaire à la remédiation.

Les mots de passe en clair sont-ils envoyés à Have I Been Pwned lors de la vérification ?

Non, jamais. Le protocole k-Anonymity garantit que seuls les 5 premiers caractères du hash SHA-1 du mot de passe sont transmis à l'API HIBP. L'API retourne alors tous les suffixes de hash correspondants (en moyenne 500 résultats). La comparaison est effectuée localement sur votre infrastructure. Le mot de passe en clair, le hash complet SHA-1 et même le hash NTLM ne quittent jamais votre réseau. Pour l'audit Active Directory avec DSInternals, la vérification peut être réalisée en mode totalement offline en téléchargeant la base HIBP NT Hash Database (~12 Go) et en comparant localement sans aucune connexion externe.

Quelles permissions sont nécessaires pour utiliser DSInternals sur un domaine Active Directory ?

DSInternals requiert des privilèges élevés pour accéder aux hash NTLM du domaine. Pour `Get-ADReplAccount`, vous avez besoin du privilège "Replicating Directory Changes All" (ou être Domain Admin). Pour `Get-BootKey` et `Test-PasswordQuality`, une session sur un contrôleur de domaine avec les droits d'administration locaux est nécessaire. Une bonne pratique consiste à créer un compte de service dédié avec uniquement le droit "DS-Replication-Get-Changes-All", membre du groupe "Domain Admins" uniquement pendant l'audit, puis révoqué immédiatement après. Toutes ces opérations doivent être tracées dans les Event Logs (EventID 4662).

Quelle est la différence entre l'audit en ligne (API HIBP) et l'audit offline avec la base NTLM ?

L'audit en ligne via l'API HIBP k-Anonymity est plus simple à mettre en place et toujours à jour (la base HIBP est mise à jour continuellement), mais nécessite une connexion internet sortante vers api.pwnedpasswords.com. L'audit offline avec la base NTLM téléchargée (~12 Go compressée) est totalement isolé du réseau, idéal pour les environnements classifiés ou les réseaux industriels OT/IT. La base offline peut dater de quelques semaines selon la date du dernier téléchargement. Pour les environnements critiques, la méthode offline est recommandée car aucune donnée ne sort du réseau, même sous forme de préfixe de hash.

Comment forcer automatiquement le changement de mot de passe pour les comptes compromis détectés ?

Après l'audit, vous pouvez forcer le changement de mot de passe avec la commande PowerShell : `Set-ADUser -Identity 'nomutilisateur' -ChangePasswordAtLogon $true`. Pour une remédiation automatisée à grande échelle, intégrez cette commande dans le script d'audit avec une liste d'exclusions pour les comptes de service critiques. Envoyez une notification par email à l'utilisateur et à son responsable avant d'activer le flag. Pour les comptes de service, créez un ticket dans votre ITSM et gérez la rotation via votre solution PAM (CyberArk, BeyondTrust, etc.). Ne jamais forcer la réinitialisation d'un compte de service en production sans validation préalable car cela peut interrompre des applications critiques.

À RETENIR

À retenir

L'API k-Anonymity de Have I Been Pwned garantit qu'aucun mot de passe ni hash complet ne quitte jamais votre réseau : seuls 5 caractères hexadécimaux du hash SHA-1 sont transmis.

DSInternals (`Get-ADReplAccount` + `Test-PasswordQuality`) permet d'auditer l'intégralité des hash NTLM du domaine AD ; combiné avec la base HIBP offline (~12 Go), l'audit est totalement air-gapped.

Automatisez l'audit via une tâche planifiée GPO avec notification email aux utilisateurs concernés et forçage du changement de mot de passe à la prochaine connexion (`-ChangePasswordAtLogon $true`).

Azure AD Password Protection (Microsoft Entra) bloque en temps réel les mots de passe faibles ou issus de listes bannies ; les EventIDs 10014-10016 permettent de détecter les tentatives de password spraying.

En France, l'audit de mots de passe AD est encadré par le RGPD Art. 32 et la NIS 2 Art. 21 : documentez chaque audit dans le registre de traitement, limitez l'accès aux résultats et conservez les rapports 30 jours maximum.

Sources

ANSSI CERT-FR — Publications et alertes

MITRE ATT&CK — Framework des techniques d'attaque

CISA — Advisories de cybersécurité