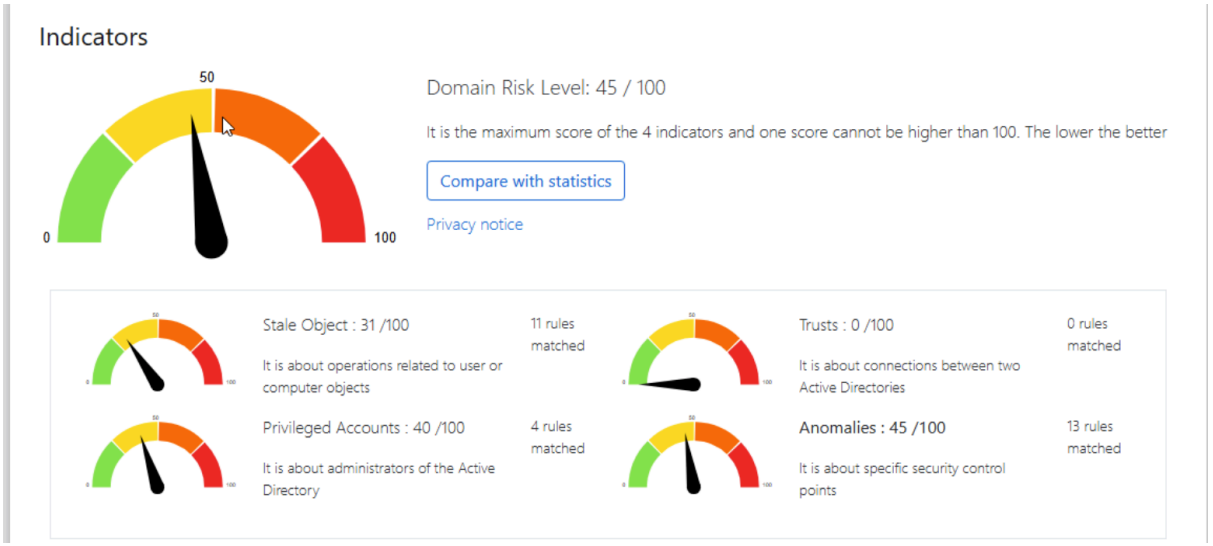
Atelier de professionnalisation : Ping Castle

Adrien BLAIZE

Audit avant corrections de failles



Quand j'ai fait mon premier audit, mon domaine présentait un score de 45/100.
Ce score indique que la sécurité était mitigée :

- Stale Object : 31/100 : Beaucoup de comptes utilisateurs ou ordinateurs obsolètes ou mal suivis étaient présents. Ils pouvaient servir de porte d'entrée à un attaquant.
- Privileged Accounts : 40/100 : Les comptes administrateurs ou à privilèges n'étaient pas assez protégés, ce qui augmentait les risques de compromission.
- Trusts : 0/100 : Il n'y avait pas de relation de confiance dangereuse entre plusieurs domaines, ce point était sécurisé.
- Anomalies : 45/100 : Plusieurs configurations étaient risquées ou incorrectes, selon les règles de sécurité recommandées.

Ce premier audit m'a montré que de nombreuses failles pouvaient permettre à un attaquant de s'infiltrer ou de se déplacer dans le domaine. Il fallait donc agir pour renforcer la sécurité globale.

1. Administrateurs non inclus dans les utilisateurs protégés

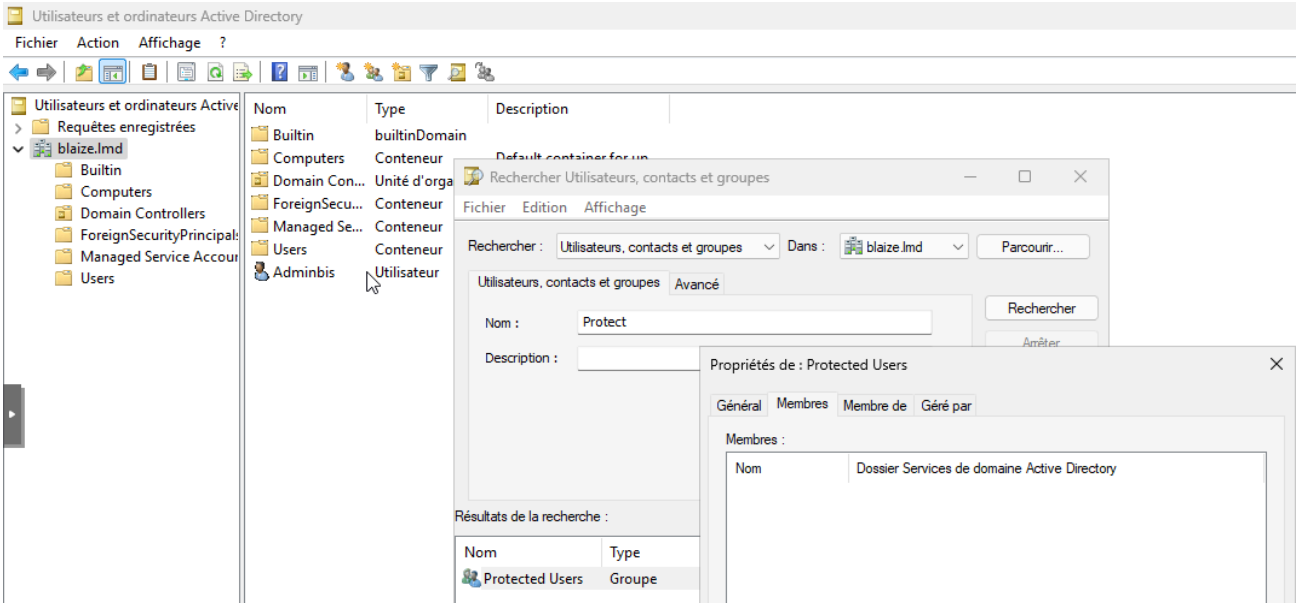
Deux comptes administrateurs du domaine ne sont pas membres du groupe « Utilisateurs protégés » (Protected Users).
Cela signifie qu'ils ne bénéficient pas des protections avancées qu'offre ce groupe dans Active Directory.

Le problème étant que les comptes administrateurs non protégés sont la cible principale des attaquants.
S'ils ne sont pas inclus dans ce groupe, ils :

→ Utilisent des protocoles d'authentification faibles.
→ Peuvent être exposés aux attaques Pass-the-Hash/Pass-the-Ticket.
→ Ont un mot de passe qui ne se renouvelle jamais ou des sessions ouvertes comportant des risques.
→ autorisent le cache de leur mot de passe lors de connexions sur des postes clients.

La conséquence est dramatique :
En cas d'attaque réussie, l'attaquant peut se déplacer sur le réseau, s'emparer d'autres identités et compromettre l'ensemble du SI.

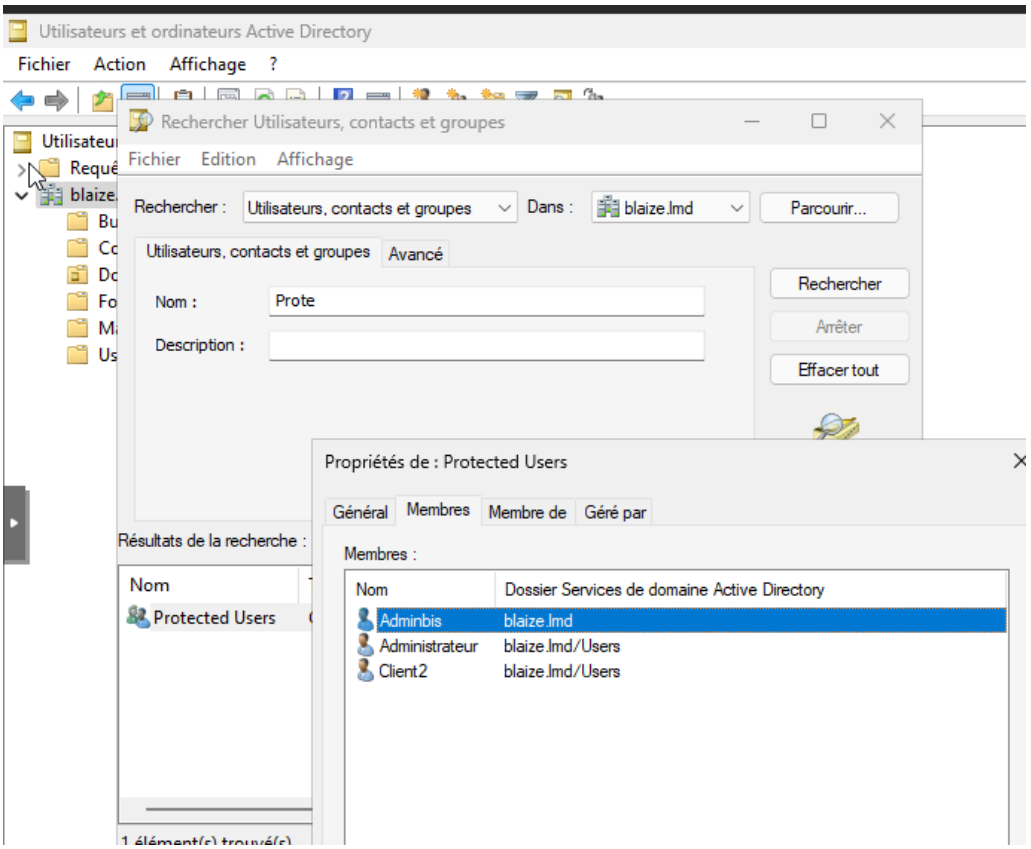
C'est le cas comme montré ci-dessous :



Présentation de la solution

Solution Technique :

Inclure tous les comptes administrateurs dans le groupe « Utilisateurs protégés » (Protected Users).



La présence du cache des mots de passe sur les postes clients des users “administrateur” comme Adminbis et Client2 ne sera donc plus émis.

Ainsi, une fois membres de ce groupe, les comptes ne pourront plus utiliser les protocoles d’authentification faibles comme NTLM ou Kerberos avec des algorithmes vulnérables. Les mots de passe ne seront plus mis en cache sur les machines clientes, ce qui évite leur vol lors d’une compromission locale. Les protections contre les attaques Pass-the-Hash et Pass-the-Ticket sont donc bien plus complexes à effectuer, et certaines fonctionnalités comme la délégation d’identifiants sont bloquées.

2. LAPS (Local Administrator Password Solution) non installé

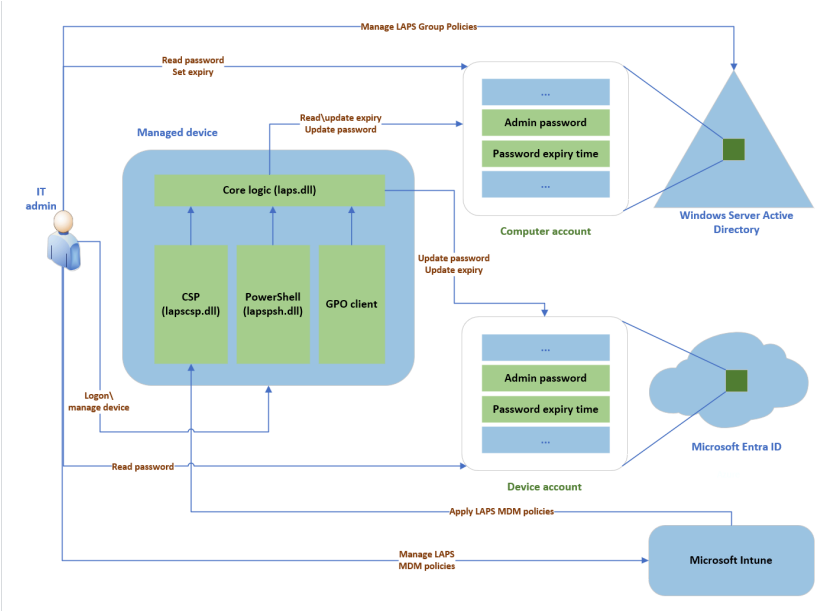
Dans un contexte de sécurité,notamment celui des mots de passe, le mot de passe administrateur local est généralement identique sur tous les postes ou changé peu fréquemment, ce qui représente une faille majeure dans la sécurité dans notre cas par exemple.

En effet, si un attaquant obtient le mot de passe du compte administrateur local d'un seul poste, il peut l'utiliser pour accéder à tous les autres ordinateurs du domaine qui partagent le même mot de passe.

- Les attaques de type "Pass-the-Hash" deviennent beaucoup plus simple : une fois un mot de passe local trouvé, l'attaquant peut escalader les privilèges ou se déplacer dans le réseau.
- Le mot de passe administrateur local peut rester inchangé pendant des mois, ce qui facilite la persistance des menaces sur les machines ciblées.

Présentation de la solution

LAPS, ou Local Administrator Password Solution, est un outil créé par Microsoft pour gérer automatiquement les mots de passe du compte administrateur local sur chaque machine du domaine Active Directory.



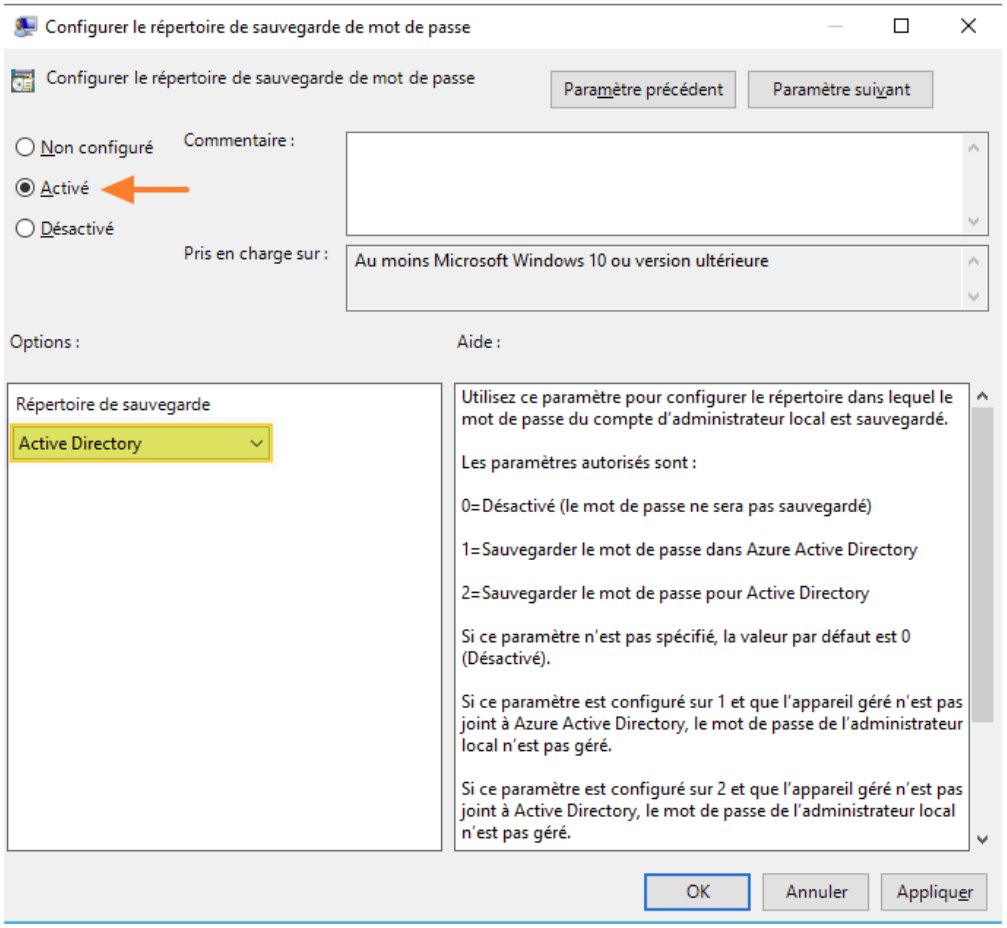
En pratique, LAPS permet à un administrateur de s'assurer que les mots de passe ne sont pas tous identiques, qu'ils changent régulièrement, et qu'ils restent accessibles uniquement aux bonnes personnes. Sur chaque ordinateur, LAPS crée un mot de passe unique, le sauvegarde dans un endroit sécurisé (Active Directory ou le cloud Microsoft), et le renouvelle automatiquement.

Quand un administrateur a besoin de se connecter en local sur un poste, il peut récupérer le mot de passe sans risquer qu'un pirate ait déjà l'accès. Si un attaquant vole un mot de passe, il ne pourra pas l'utiliser sur d'autres machines, car chaque poste a son propre mot de passe qui change souvent.

Ce système évite les gros problèmes de sécurité qui arrivent quand tous les mots de passe administrateurs sont identiques et jamais mis à jour. Grâce à LAPS, on rend le réseau beaucoup plus sûr, et on simplifie la gestion pour les admins.

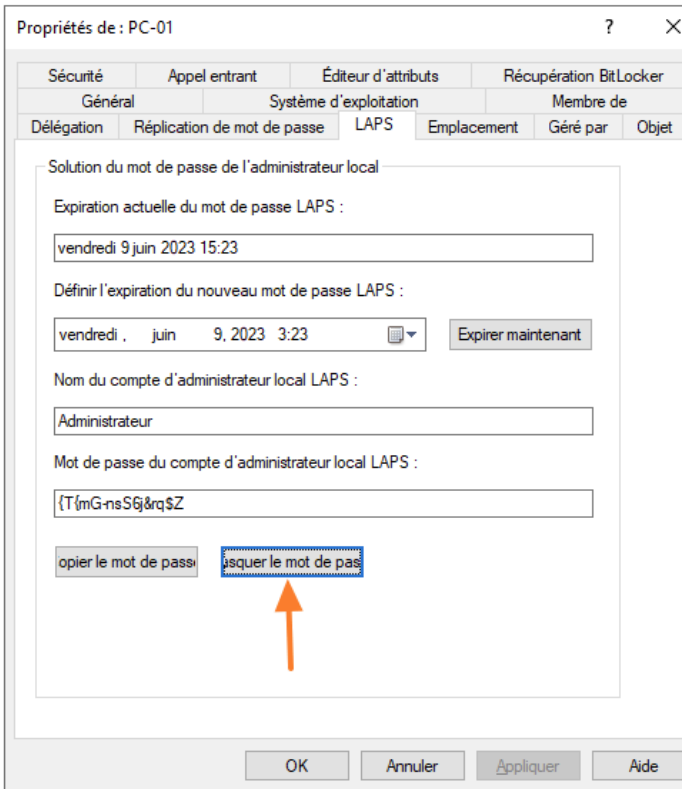
Paramétrage de la GPO LAPS

Cette première capture montre la configuration d'une stratégie de groupe (GPO) dans la console "Group Policy Management". Ici, le paramètre "Configuration du répertoire de sauvegarde du mot de passe" est activé, avec "Active Directory" sélectionné. Cette étape est indispensable car elle indique au système que les mots de passe générés pour les comptes administrateurs locaux seront stockés et protégés dans l'Active Directory, selon la politique concrète définie par l'administrateur.



Onglet “LAPS” dans les propriétés d’un objet ordinateur

La deuxième image présente les propriétés d’un poste client dans la console "Active Directory Users and Computers". Un nouvel onglet LAPS apparaît grâce à la mise en place de la solution. Ici, on retrouve la date d’expiration du mot de passe, le nom du compte administrateur géré, et surtout un bouton permettant d’expirer manuellement le mot de passe ou de le copier. Cet onglet permet aux administrateurs de consulter ou de forcer la rotation du mot de passe administrateur local, en garantissant que chaque poste dispose d’un mot de passe unique et sécurisé.



Lecture du mot de passe LAPS via PowerShell

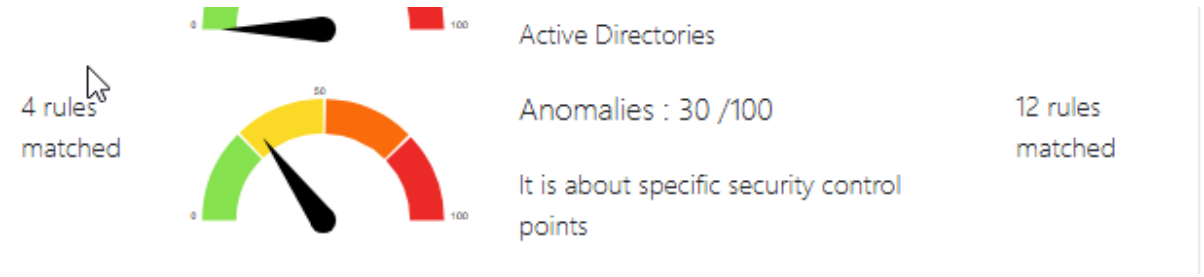
Cette dernière capture affiche l'utilisation de la commande PowerShell Get-LapsADPassword sur un poste client géré. La réponse me donne le nom de la machine, l'objet correspondant dans Active Directory, le nom du compte, le mot de passe généré, et les informations de statut et d'expiration. Cette étape devrait me permettre de récupérer en clair le mot de passe local d'un poste pour intervenir en fonction des droits que je me suis attribués, tout en gardant une traçabilité et une gestion sécurisée des accès.

Malheureusement, je n'ai pas réussi à le finir, cependant dans "Account" et "Password", j'aurais dû pouvoir voir le profil et le mot de passe.

```
PS C:\Users\Administrator> Get-LapsADPassword -Identity "Client1-W11" -AsPlainText


ComputerName        : CLIENT1-W11
DistinguishedName   : CN=CLIENT1-W11,OU=PC,DC=ablai,DC=home
Account             :
Password            :
PasswordUpdateTime  : 22/11/2025 22:29:31
ExpirationTimestamp : 22/12/2025 22:29:31
Source              : EncryptedPassword
DecryptionStatus    : Unauthorized
AuthorizedDecryptor : ABLAI\test


PS C:\Users\Administrator> S
```



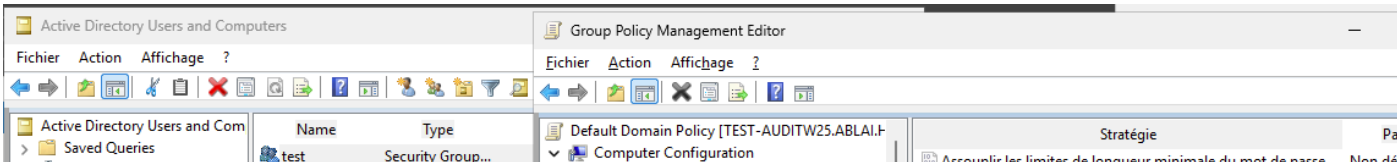
3. Configuration des règles des mots de passes pour une sécurité renforcée.

Exiger un mot de passe de moins de 8 caractères dans la politique de sécurité du domaine est considéré comme une faille car ces mots de passe sont trop courts pour être vraiment efficaces contre les attaques modernes. Un attaquant peut facilement casser un mot de passe simple par force brute ou en utilisant des listes de mots courants. Plus le mot de passe est court, plus il est facile à deviner ou à “craquer”.

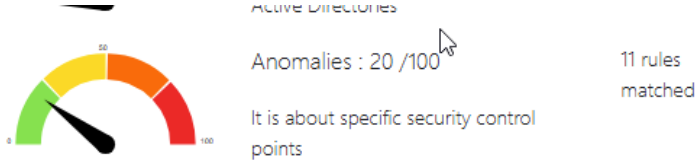
C’est pour cela que CERT-FR, l’ANSSI et la majorité des agences de sécurité nous recommandent de configurer une longueur minimale de 8 à 12 caractères pour les mots de passe, afin de garantir une meilleure protection pour les comptes utilisateurs.

Présentation de la solution

Afin de sécuriser d’avantages l’accès aux mots de passes, j’ai décidé de modifier des règles les concernant :



- Longueur minimale du mot de passe : passée à 12 caractères. Cela signifie que tous les utilisateurs doivent maintenant choisir des mots de passe d’au moins 12 lettres ou signes, ce qui rend leur compte beaucoup plus difficile à pirater.
- Audit de la longueur minimale du mot de passe : activé à 8 caractères. Cela me permet de vérifier si la règle de longueur minimale est bien respectée dans mon domaine.
- Complexité de mot de passe : activée. Les mots de passe doivent maintenant contenir des minuscules, majuscules, chiffres ou symboles, augmentant encore le niveau de sécurité.
- Durée de vie maximale du mot de passe : réglée à 60 jours. Les utilisateurs vont devoir renouveler régulièrement leur mot de passe, pour éviter qu’un vieux mot de passe soit compromettant (soit 2 mois).
- Historique du mot de passe : mémorise les 24 anciens mots de passe. Ainsi, il est interdit pour un utilisateur de reprendre les anciens mots de passe déjà utilisés.



4. Utilisation des protocoles NTLMv1 ou LM.

Les protocoles NTLMv1 et LAN Manager (LM), son successeur, permettent l'authentification des utilisateurs sur les réseaux Windows.

Dans le fonctionnement, lorsqu'un utilisateur souhaite accéder à une ressource réseau, le protocole NTLM ou LM va générer et envoyer un hash dérivé du mot de passe pour vérifier l'identité auprès du serveur.

Cela n'est pas réellement sécurisé car ces hash peuvent être interceptés par un attaquant positionné sur le réseau (attaque de type man-in-the-middle).

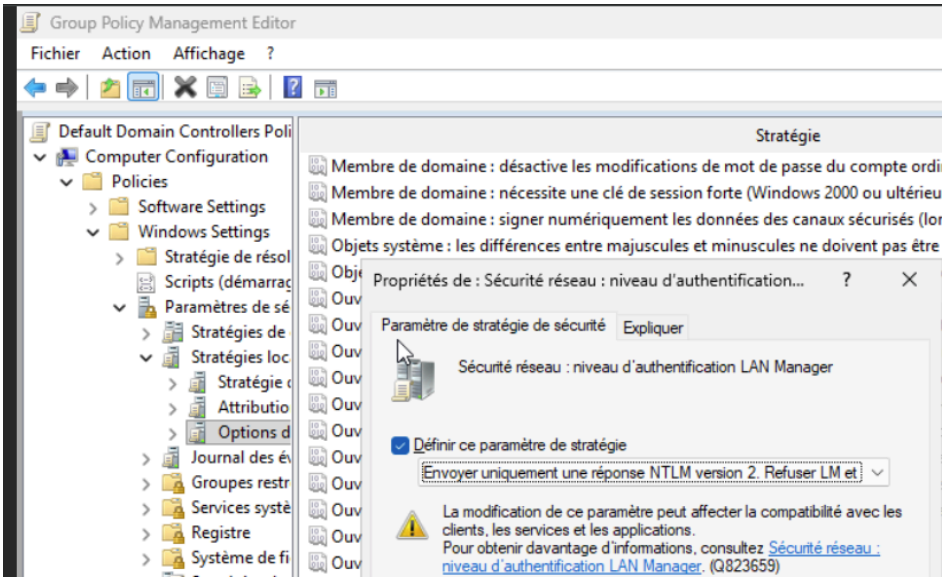
Les hackers disposent aujourd'hui d'outils permettant d'exploiter ces hash pour s'authentifier frauduleusement, relayer des connexions (attaque NTLM relay), ou même retrouver le mot de passe original via des attaques par force brute.

Le principal souci, et ce qui en fait une vraie faille selon les standards actuels en sécurité système, c'est que l'utilisation de NTLMv1 et LM rend le réseau très vulnérable à différents types d'attaques d'usurpation d'identité et d'escalade de privilèges.

Présentation de la solution

En utilisant NTLMv2, Les utilisateurs pourront prouver leur identité sans envoyer leur mot de passe directement. Le principe, c'est que lorsqu'un utilisateur veut accéder à une ressource, le serveur lui envoie un défi (appelé challenge). L'ordinateur de l'utilisateur chiffre ce défi avec son mot de passe, puis renvoie une réponse au serveur. Le serveur compare la réponse avec ce qu'il attend, et si ça correspond, l'accès est autorisé.

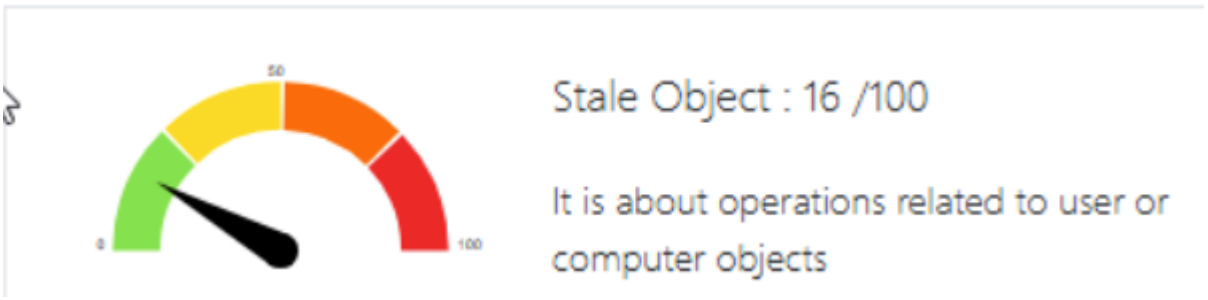
NTLMv2 est bien plus sécurisé parce qu'il utilise un chiffrement plus complexe, et qu'il rajoute des informations comme la date et le nom du serveur dans le calcul.
Les attaques sont beaucoup plus difficiles à mettre en place par rapport aux anciennes versions comme NTLMv1 ou LM. Grâce à ça, on réduit fortement les risques de piratage des mots de passe et d'usurpation d'identité sur le réseau.



Pour renforcer la sécurité du domaine, je suis allé dans la console "Group Policy Management Editor" sur le contrôleur de domaine. J'ai modifié la stratégie de groupe par défaut des contrôleurs de domaine ("Default Domain Controllers Policy"), afin d'imposer un niveau d'authentification plus sécurisé pour les échanges sur le réseau.

Dans la branche "Configuration ordinateur" > "Stratégies" > "Paramètres Windows" > "Paramètres de sécurité" > "Stratégies locales" > "Options de sécurité", j'ai trouvé le paramètre "Sécurité réseau : niveau d'authentification LAN Manager". Ensuite, j'ai défini cette stratégie sur "Envoyer uniquement une réponse NTLM version 2. Refuser LM et NTLM".

Il faut évidemment appliquer la mise à jour de la stratégie.



5. activation du paramètre “non déléguable” pour limiter l’usurpation d’identité

La faille détectée concerne le fait que certains comptes administrateurs de mon domaine n’ont pas le drapeau « Ce compte est sensible et ne peut pas être délégué » activé.

Sans ce paramètre, un compte admin peut être ciblé par la délégation Kerberos, ce qui signifie qu’un service ou une machine pourrait emprunter son identité via une attaque d’usurpation.
Cela peut permettre à un attaquant qui contrôle un service pouvant utiliser la délégation de s’attribuer les droits d’un administrateur, ce qui ouvre la porte à des actions dangereuses comme créer des comptes, modifier des droits ou accéder à des données confidentielles.

Présentation de la solution

Sur mon contrôleur de domaine, j’utilise la commande suivante pour afficher tous les membres du groupe Domain local (de toute manière je n’ai qu’un “Admin” existant) :

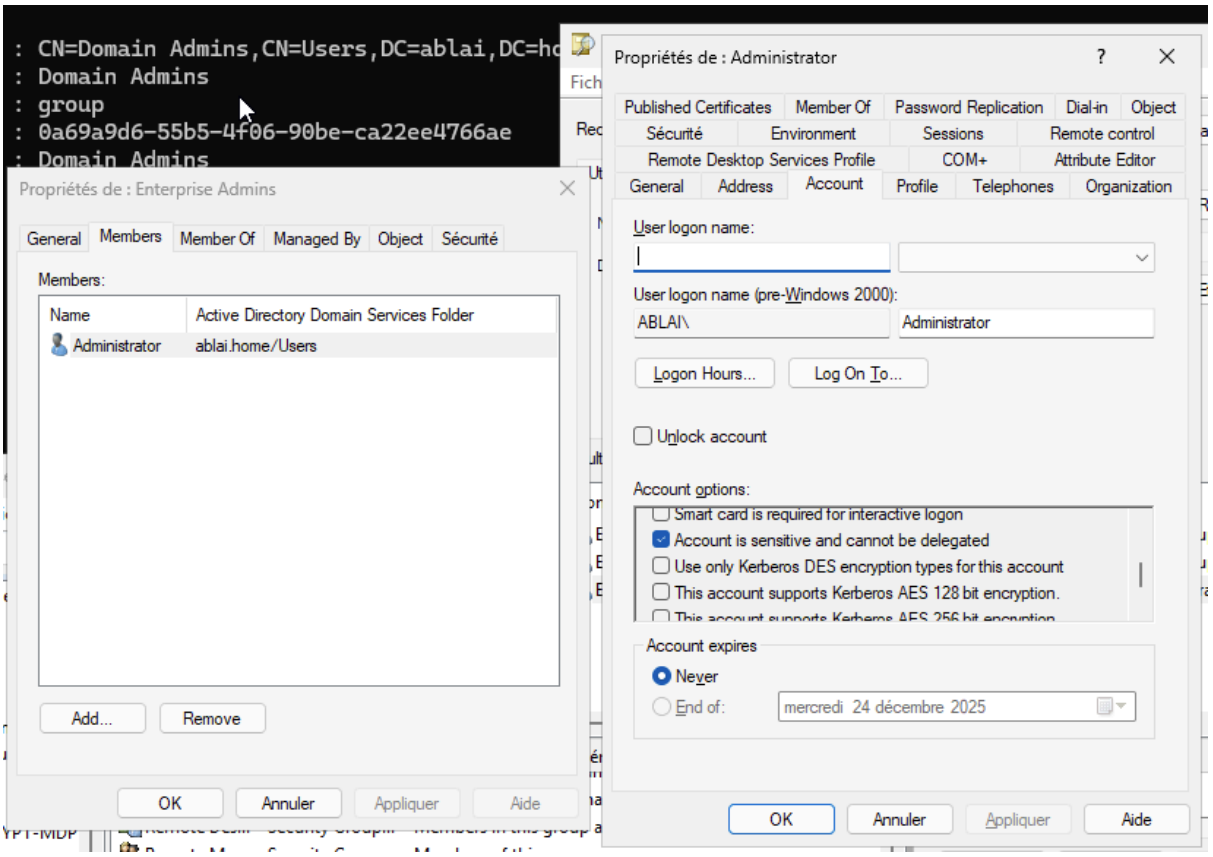
```
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Administrators"


distinguishedName : CN=Domain Admins,CN=Users,DC=ablai,DC=home
name              : Domain Admins
objectClass       : group
objectGUID        : 0a69a9d6-55b5-4f06-90be-ca22ee4766ae
SamAccountName    : Domain Admins
SID               : S-1-5-21-1057521362-775650006-2636437948-512

distinguishedName : CN=Enterprise Admins,CN=Users,DC=ablai,DC=home
name              : Enterprise Admins
objectClass       : group
objectGUID        : 9d61e080-357c-466e-956d-07788290ed9e
SamAccountName    : Enterprise Admins
SID               : S-1-5-21-1057521362-775650006-2636437948-519

distinguishedName : CN=Administrator,CN=Users,DC=ablai,DC=home
name              : Administrator
objectClass       : user
objectGUID        : faf0c882-fcd5-4139-bff5-29860e39fbf7
SamAccountName    : Administrator
SID               : S-1-5-21-1057521362-775650006-2636437948-500
```

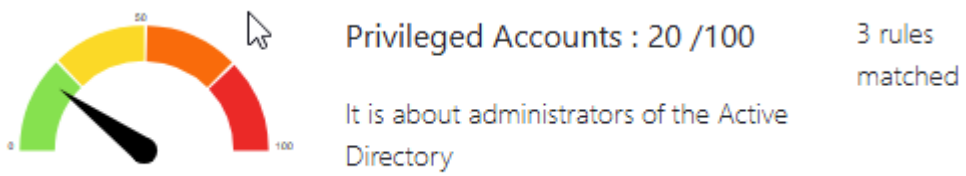
Ainsi, je n'ai plus qu'à désactiver l'option de délégation comme c'est un compte sensible.



J'ai fait la résolution de la faille qui concerne la délégation des comptes administrateurs dans mon Active Directory. Pour ça, j'ai recherché tous les comptes qui font partie des groupes à privilèges, comme "Domain Admins" ou "Enterprise Admins", afin de repérer les comptes sensibles (Administrator).

J'ai ouvert la console "Utilisateurs et ordinateurs Active Directory", puis j'ai sélectionné mon compte administrateur. Dans les propriétés du compte, onglet "Compte", j'ai coché la case "Ce compte est sensible et ne peut pas être délégué". Grâce à cette manipulation, j'ai empêché la délégation Kerberos sur ces comptes : même si un service essaie d'utiliser leurs droits par délégation, ça ne marchera pas.

Ce paramètre rend mes admins non déléguables, ce qui protège contre les attaques où un serveur ou service pourrait essayer de s'approprier les droits d'un administrateur par usurpation. Avec cette modification, je limite grandement les risques d'escalade et de compromission totale du domaine

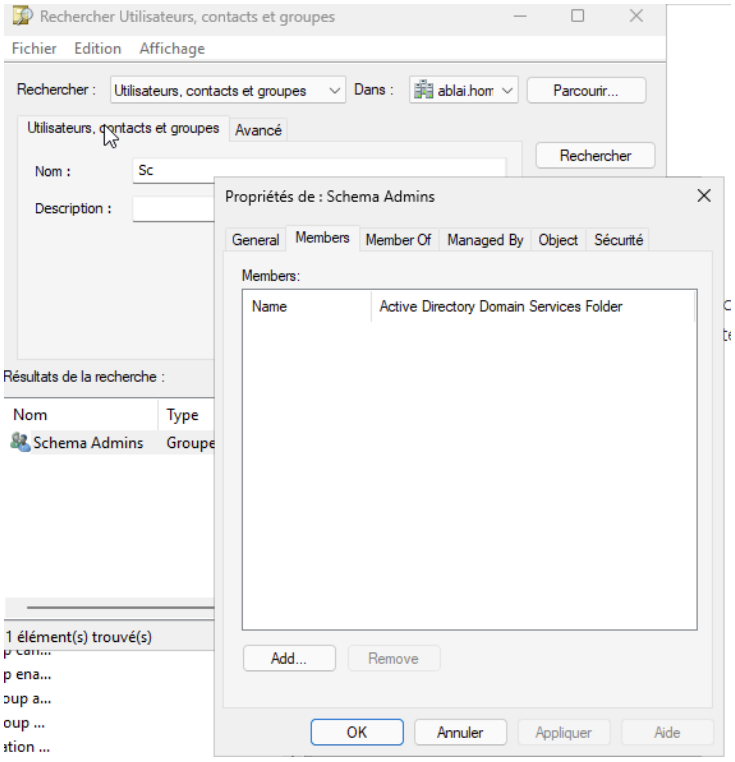


6. Sécuriser le groupe “Schema Admins” pour éviter des modifications inattendues du schéma AD

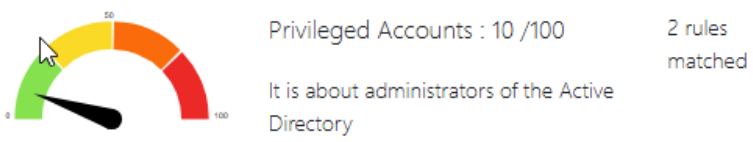
Le groupe “Schema Admins” permet de modifier le schéma de l’Active Directory.
Dans ce groupe, des accès excessivement sensible : une modification du schéma (ajout de nouveaux objets, nouveaux attributs…) impacte tout le domaine et est irréversible. Si un utilisateur non autorisé reste membre de ce groupe, il pourrait modifier le schéma sans surveillance, casser l’annuaire, voire obliger un rebuild complet du domaine. Laisser des comptes dans ce groupe en permanence augmente considérablement le risque d’incident ou de sabotage, ce qui est donc mon cas.

Présentation de la solution

Afin d’éviter que cela se produise, je vais donc enlever mon compte “Administrator” afin que le groupe soit en état de sûreté puisque aucun compte (notamment Admin) soit dedans.



Pour l’enlever, je me suis rendu dans “AD user and computer”, fait une recherche sur mon domaine le groupe “Schema Admins” puis j’ai enlevé mon comte “Administrator” de ce groupe.



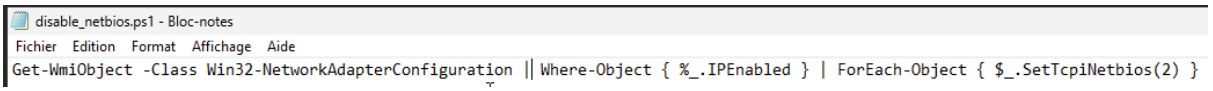
7. Désactivation de NetBIOS via GPO pour renforcer la sécurité du réseau Active Directory"

Je considère NetBIOS comme une faille de sécurité parce que c'est un ancien protocole qui n'est plus adapté aux standards actuels. Quand NetBIOS est activé sur les postes de mon réseau, je sais qu'un attaquant pourrait intercepter ou détourner des requêtes réseau pour récupérer des identifiants ou des informations sensibles. Il pourrait se faire passer pour une ressource, piéger des ordinateurs et obtenir les mots de passe ou les noms d'utilisateur de mes collègues.

Je vois aussi que NetBIOS facilite des attaques connues, comme le spoofing ou le relay de NTLM. Ce type de menace met en danger tout le domaine si je ne sécurise pas correctement mes PC. Comme NetBIOS ne chiffre pas les échanges et ne vérifie pas l'intégrité, je dois le désactiver : c'est la meilleure façon de protéger mon environnement contre ces méthodes d'attaque qui ciblent notamment des réseaux d'entreprises.

Présentation de la solution

Désactivation par script PowerShell

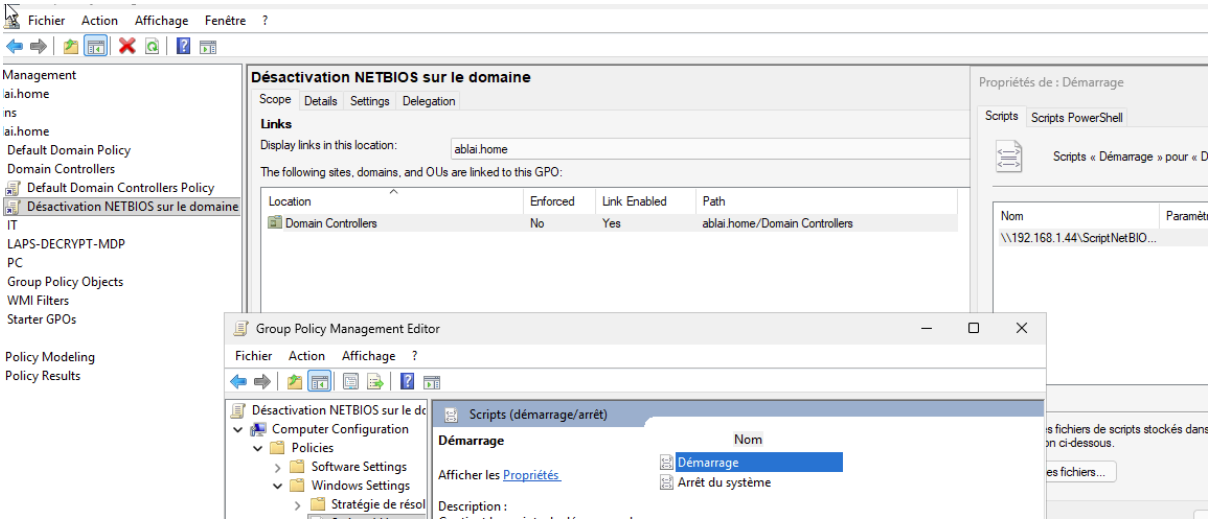


Le script que j'ai préparé sert à désactiver NetBIOS sur toutes les interfaces réseau de mon ordinateur.
Je l'écris dans un fichier texte que je sauvegarde en .ps1 (fichier PowerShell).

Voici le détail :

- Quand je lance le script, il commence par récupérer toutes les cartes réseau qui sont actives (celles qui ont une adresse IP).
- Pour chaque carte réseau trouvée, il applique la commande SetTcpipNetbios(2).
- Le chiffre 2 veut dire : "Désactiver NetBIOS sur TCP/IP", c'est-à-dire couper le service sur cette interface.
- Le script fait le tour de chaque interface, même si j'en ai plusieurs (ex : Wi-Fi, Ethernet).

Déploiement du script avec une GPO



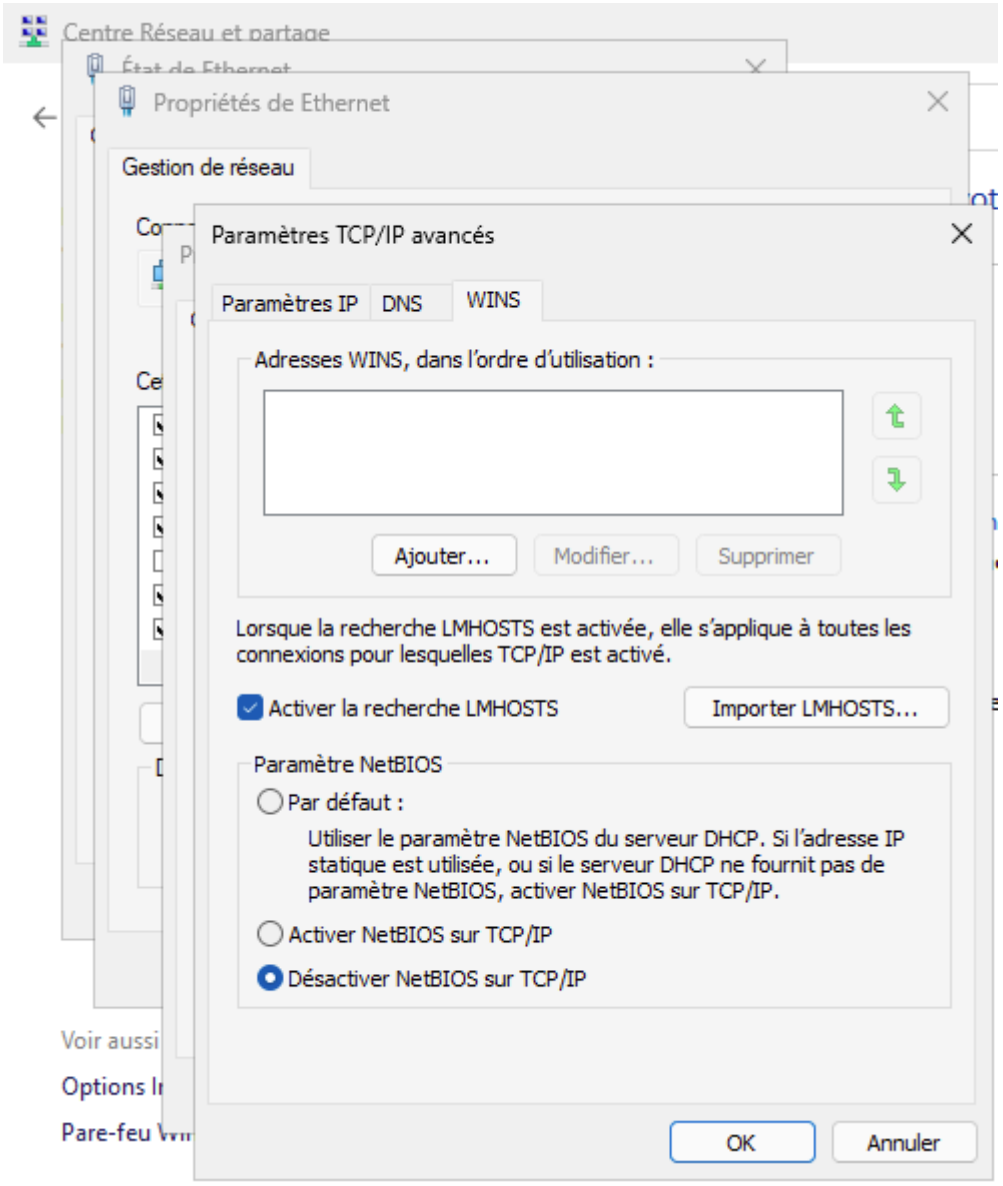
Quand je veux appliquer la solution à tout le domaine, j'utilise la GPMC pour créer et lier une GPO. Je configure la section "Scripts de démarrage" dans "Configuration ordinateur", puis j'ajoute mon script PowerShell par son chemin réseau dans la GPO.
En appliquant la GPO, je m'assure que chaque poste du domaine exécute automatiquement le script au démarrage et désactiver NetBIOS sans action manuelle de ma part.

Ainsi, je l'ai déployé depuis un dossier partagé avec les users et pc du domaine depuis un fichier texte contenant le script.

Désactivation du NetBIOS dans les propriétés réseau

Cette capture montre comment on désactive NetBIOS sur TCP/IP via l'interface graphique Windows.

- Il faut aller dans "Centre Réseau et partage" → "Propriétés de la carte Ethernet" → "Propriétés IPv4" → "Avancé" → Onglet "WINS" → puis cocher "Désactiver NetBIOS sur TCP/IP".
- Solution technique : Cette méthode manuelle est adaptée pour quelques postes ou des machines hors domaine. Elle garantit que NetBIOS est aligné côté client, mais n'est pas automatisable à grande échelle.



8. Vérifier l’application par défaut pour l’exécution des fichiers script

Par défaut, l’extension .ps1 (PowerShell) s’ouvre dans le Bloc-notes, ce qui sécurise, mais d’autres extensions comme .js, .vbs, .wsf s’ouvrent encore dans leur moteur d’exécution.
Si un script malveillant tombe entre les mains d’un utilisateur (e-mail de phishing, archive corrompue), il peut s’exécuter directement, permettant une infection ou un accès non autorisé au système.
Le risque principal est que des scripts puissants soient lancés par accident ou à l’insu de l’utilisateur, facilitant les attaques (ex : ransomware, vol de données).

Présentation de la solution

Lorsque tous les fichiers script s’ouvrent dans le Bloc-notes, ils ne s’exécutent plus directement.
Cela protège contre les ouvertures involontaires et pièges de phishing.
La navigation web n’est pas perturbée et la menace côté messagerie ou téléchargements est supprimée pour ces formats.
Il reste utile de compléter par une règle "Attack Surface Reduction" de Windows pour bloquer l’exécution de JavaScript et VBScript malveillants au niveau du système, surtout depuis Windows 10.

Il faut rediriger l’ouverture des fichiers script (.js, .vbs, .wsf, etc.) pour qu’ils s’ouvrent par défaut dans un éditeur comme le Bloc-notes et non leur moteur d’exécution.
Pour cela :

- Clic droit sur le fichier → "Ouvrir avec" → "Choisir une autre application" → Sélectionne "Bloc-notes" et coche "Toujours utiliser cette application".
- Il est possible de le faire en masse avec une GPO ou une modification du registre pour tous les postes du domaine (ce que je vais faire).
- Vérifie chaque extension : .js, .jse, .vbs, .vbe, .vb, .wsh, .wsf.

Script de l’ouverture des fichiers script

```
$exts = @(".js",".jse",".vbs",".vbe",".vb",".wsh",".wsf")
foreach ($ext in $exts) {
  New-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\$ext\UserChoi
ce" -Name "Progid" -Value "Applications\notepad.exe" -Force
}
```

Quand j’exécute ce script, il commence par définir une liste d’extensions de fichiers à risque : .js, .jse, .vbs, .vbe, .vb, .wsh, .wsf.

Pour chaque extension dans la liste, le script va modifier une clé du registre Windows, située ici : HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\$ext\UserChoice

Il utilise la commande New-ItemProperty pour créer ou écraser la propriété appelée "Progid".
La valeur que je mets, c’est "Applications\notepad.exe" : ça indique à Windows que, pour chaque fichier ayant l’extension ciblée, il faut l’ouvrir avec le Bloc-notes par défaut.

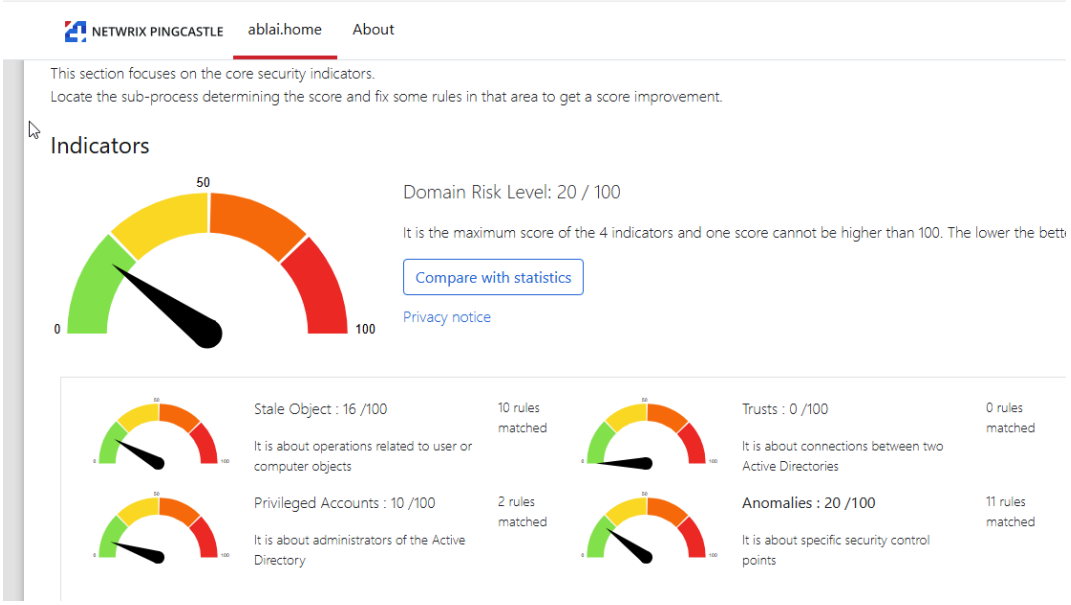
Le paramètre -Force me permet d’écraser la valeur existante s’il y en a déjà une.

En résumé, ce script :

- Parcourt toutes les extensions de fichiers script que je veux protéger.
- Modifie la configuration de Windows pour qu’ils s’ouvrent avec le Bloc-notes, et non leur moteur d’exécution d’origine.
- Cette modification se fait au niveau de chaque utilisateur (HKCU pour HKEY_CURRENT_USER), donc chaque personne du domaine bénéficie de la protection dès la prochaine ouverture de session.

Ainsi, après l’application de cette stratégie, tout script souhaitant s’exécuter ne pourra le faire seulement sur le Bloc note.

Audit après corrections de failles



Stale Object : 10/100
J'ai analysé et supprimé les comptes utilisateurs et ordinateurs inactifs ou non utilisés.
Je me suis assuré que chaque objet présent dans l'AD sert vraiment à la production ou à l'administration, ce qui limite les risques de compromission via un vieux compte oublié.

Privileged Accounts : 10/100
J'ai inclus tous les administrateurs dans le groupe "Utilisateurs protégés", ce qui bloque l'usage des protocoles faibles et empêche le cache du mot de passe sur les machines clientes.
J'ai également renforcé les mots de passe et supprimé les droits de délégation sur tous les comptes sensibles, pour empêcher les attaques de type pass-the-hash et l'usurpation d'identité.

Trusts : 0/100
Aucune relation de confiance entre domaines n'a été détectée, donc ce score est parfait.
Je n'ai rien eu à corriger ici car mon architecture ne comporte qu'un domaine sans jonction à l'extérieur.

Anomalies : 20/100
J'ai mis en place LAPS pour que chaque machine ait un mot de passe administrateur local unique, qui se change automatiquement et régulièrement.
J'ai configuré des règles de sécurité strictes pour les mots de passe (longueur minimale, complexité, historique, durée de vie).
J'ai désactivé les protocoles LM et NTLMv1 sur le domaine afin d'imposer NTLMv2 pour toute l'authentification réseau.
J'ai supprimé tous les membres inutiles du groupe "Schema Admins" pour éviter des modifications imprévues du schéma AD.
J'ai aussi déployé un script via une GPO pour désactiver NetBIOS sur tous les postes du domaine, ce qui élimine une faille réseau importante.

J'ai modifié les associations de fichiers script (.js, .vbs, .wsf, etc.)
Pour éviter l'exécution accidentelle de fichiers scripts malveillants, j'ai préparé un script PowerShell ou une préférence de registre via GPO.