

[Énumération] AD

Introduction

L'énumération d'un environnement Active Directory est primordial pour trouver ses faibles.



Énumération distante

Scan réseau

Si vous avez un accès sur un sous-réseau et que vous cherchez les contrôleurs de domaine disponibles ou les serveurs Windows en général, vous allez pouvoir les identifier rapidement grâce à **CrackMapExec** :

```
cme smb <NET_IP/MASK>
```

Vous allez récupérer les versions de Windows ainsi que les noms des domaines associés à ces serveurs.

Scan d'un contrôleur de domaine

Vous pouvez récupérer quelques informations utiles avec **CrackMapExec** :

```
cme smb <DC_IP|DC_FQDN>
```

Énumération des utilisateurs du domaine

Il est parfois possible de récupérer la liste des utilisateurs du domaine notamment avec **enum4linux** :

```
enum4linux -U <DC_IP>
```

Ou avec **kerbrute** :

```
kerbrute userenum --dc <DC_IP> -d <DOMAIN_FQDN> <USERLIST>
```

Énumération de la base de registre

```
reg.py <DOMAIN>/<USER>:<PASSWORD>@<IP> query -keyName "<REGISTRY_FOLDER>"
```

Voici un exemple pour lister les logiciels présent sur le poste :

```
reg.py <DOMAIN>/<USER>:<PASSWORD>@<IP> query -keyName "HKLM\\SOFTWARE"
```

Énumération des services

```
services.py <DOMAIN>/<USER>:<PASSWORD>@<IP> list
```

Récupérer les SID des utilisateurs

```
lookupsid.py <DOMAIN>/<USER>:<PASSWORD>@<IP>
```

Récupérer les comptes de service

```
GetUserSPNs.py <DOMAIN>/<USER>:<PASSWORD>
```

On peut récupérer les TGS avec l'option **-request** à placer à la fin de la commande.

Récupérer des jetons TGT

Permet de récupérer les jetons TGT des comptes utilisateurs qui ont la pré-authentification Kerberos désactivée :

```
GetNPUsers.py <DOMAIN>/<USER>:<PASSWORD> -request
```

Récupérer des hashes de mots de passe

Permet d'interroger le serveur pour récupérer les hashes de mots de passe qui pourraient se trouver dans le **ntds.dit**, ou autre :

```
secretsdump.py <DOMAIN>/<USER>:<PASSWORD>@<IP>
```

Énumération locale

Ce petit mémo regroupe toutes les commandes pour énumérer des informations sur un domaine si vous avez un premier accès sur un poste du domaine.

Trouver le nom du domaine

```
systeminfo | findstr Domain
```

Lister les utilisateurs du domaine

```
Get-ADUser -Filter *
```

Vous pouvez aussi ajouter un filtre pour lister tous les éléments d'un OU spécifique (ici **Users**) :

```
Get-ADUser -Filter * -SearchBase "CN=Users,DC=<DC_DN>,DC=<DC_TLD>"
```

On peut aussi récupérer les utilisateurs à privilèges élevés :

```
Get-ADUser -Filter * | select SamAccountName
```

Lister les logiciels anti-virus

```
wmic /namespace:\\root\\securitycenter2 path antivirusproduct
```

Afficher l'état de Windows Defender

```
Get-Service WinDefend
```

Et pour afficher l'état des différents services de Windows Defender :

```
Get-MpComputerStatus
```

Afficher l'état du pare-feu

```
Get-NetFirewallProfile | Format-Table Name, Enabled
```

Afficher les règles du pare-feu

```
Get-NetFirewallRule | select DisplayName, Enabled, Description
```

Désactiver le pare-feu

```
Set-NetFirewallProfile -Profile Domain, Public, Private -Enabled False
```

Nécessite les droits administrateurs !

Tester l'état d'un port

Dans un premier shell, se mettre en écoute :

```
Test-NetConnection -ComputerName <IP> -Port <PORT>
```

Puis initiez une connexion :

```
(New-Object System.Net.Sockets.TcpClient("<IP>", "<PORT>")).Connected
```

Afficher les sources de logs

```
Get-EventLog -List
```

Afficher le processus Sysmon

S'il existe il sera affiché :

```
Get-Process | Where-Object { $_.ProcessName -eq "Sysmon" }
```

Lister les applications installées

```
wmic product get name,version
```

Afficher les dossiers et fichiers cachés

```
Get-ChildItem -Hidden -Path <DIR_PATH>
```

Afficher les services en cours d'exécution

```
net start
```

Afficher les informations d'un service

```
wmic service where "name like '<SERVICE>'" get Name,PathName
```

Afficher les informations sur un processus

```
Get-Process -Name <PROCESS>
```

Afficher les connexions en lien avec un processus

```
netstat -noa |findstr "LISTENING" |findstr "<PID>"
```

Afficher la politique de mot de passe du domaine

```
net accounts /domain
```

Afficher les informations d'un utilisateur du domaine

```
Get-ADUser -Identity <USER> -Server <DC_FQDN> -Properties *
```

Afficher les informations d'un groupe du domaine

```
Get-ADGroup -Identity <GROUP> -Server <FQDN_DC>
```

Afficher les membres d'un groupe du domaine

```
Get-ADGroupMember -Identity <GROUP> -Server <FQDN_DC>
```

Afficher des informations génériques sur le domaine

```
Get-ADDomain -Server <FQDN_DC>
```

Afficher les relations de confiance

```
nltest /domain_trusts
```

Bloodhound

Sharphound

Cet outil du framework va se charger de l'énumération du domaine.

Voici le lien du repos :

- <https://github.com/BloodHoundAD/SharpHound>

Il doit être lancé depuis un compte utilisateur du domaine.

Cependant, il est détecté par les antivirus et EDR, c'est pourquoi il est recommandé d'utiliser notre propre machine Windows et lancer Sharphound dessus avec les droits de l'utilisateur grâce à **runas**.

Cet outil prend soit la forme d'un exécutable, soit d'un script Powershell et peut être lancé de la manière suivante :

```
SharpHound.exe --CollectionMethods All --Domain <FQDN_DC> --ExcludeDCs
```

Normalement, une archive zip a été générée.

Nous pourrions la récupérer sur notre machine pour l'utiliser avec Bloodhound

Bloodhound GUI

Lancez la console neo4j (nécessaire au fonctionnement de l'outil) :

```
neo4j console
```

Selon les distributions, vous devrez mettre ou non le mot **start** à la fin de la commande.

Puis dans un autre terminal, lancez bloodhound :

```
bloodhound --no-sandbox
```

Les identifiants par défaut sont **neo4j:neo4j**.

Revision #18

Created 12 February 2024 12:22:35 by Elieroc

Updated 3 May 2024 15:33:53 by Elieroc