WILD CODE SCHOOL

Active Directory partie 2

Service d'annuaire

LDAP ?

Arborescence AD ?

Domaine, arbre, forêt ?

Catalogue global ?

Plan

1 - Protocoles réseaux associés

2 - Fonctionnalités avancées

3 - Les objets AD

4 - Bonnes pratiques d'administration

WILD CODE SCHOOL

Protocoles réseaux associés

Différents protocoles

- DNS
- SNTP
- LDAP/LDIF
- Kerberos
- X509
- NTFS
- SMB
- CIFS

DNS

- Service obligatoire pour l'utilisation d'AD
- Utilisé pour la résolution des noms ET la résolution des services
- Attention aux problèmes de DNS qui peuvent affecter l'AD

SNTP (Simple Network Time Protocol)

- Permet la synchronisation des horloges des systèmes (stockage de l'heure UTC, affichée en tenant compte du fuseau horaire)
- Impératif pour le protocole d'authentification de Windows (Kerberos)
- Une mauvaise synchronisation du temps amène des problèmes :
  - D'authentification Kerberos
  - De réplication
  - D'échecs de connexion aux ressources
  - D'échecs de communication sécurisée (certificats)
  - De journalisation et d'audit

LDAP

- Colonne vertébrale d'AD
- Standard des services d'annuaire
- Opérations LDAP de base (bind, search, add, delete, modify)

commandes LDAP

LDIF

Les fichiers LDIF (*LDAP Data Interchange Format*) sont des fichiers textes permettant d'interagir avec l'AD.

- Permet des importations, exportations et modifications d'AD
- Permet de charger AD à partir d'une BDD externe ou inversement : ex.: gestions des comptes utilisateurs à partir de la BDD RH

Kerberos

- Protocole d'authentification central par défaut depuis Windows 2000 (remplace LM/NTLM utilisé jusqu'à NT4)
- Compatible Kerberos V5
- Authentification mutuelle du client et du serveur
- Adresse du serveur Kerberos utilisé pour l'ouverture de session extraite du DNS

X509

- Windows Server propose les services de certificats compatibles X.509
- Renforcement de la sécurité (authentification, intégrité, confidentialité, non répudiation)
- Utilisable par les autres services :
  - Authentification par carte à puce
  - Chiffrement de fichiers (EFS)
  - Chiffrement des données sur le réseau (IPSEC)

NTFS

- Gestion des droits et des contrôles d'accès
- Gestion des permissions par les groupes AD

WILD CODE SCHOOL

Fonctionnalités avancées

Le niveau fonctionnel

A la création d'un domaine, le niveau fonctionnel correspond à la version de l'OS serveur depuis lequel on crée le domaine.
domaine server 2016 aura **au maximum** un niveau fonctionnel *server 2016*.
Niveau fonctionnel de la forêt = niveau fonctionnel minimum des domaines.

Pourquoi faire une montée de niveau fonctionnel ?

- Avoir les dernières fonctionnalités
- Prendre en charge les derniers OS (client et serveur)

 Impossible de faire machine arrière sur une montée de niveau.

Le niveau fonctionnel

Lors d'une évolution de version d'AD :

- Le niveau fonctionnel peut être différent de la version de l'OS.
- Le niveau fonctionnel sera au niveau de l'OS le plus ancien.

ex:

5 DC sous Windows Server 2012 R2
→ Niveau fonctionnel = Windows Server 2012 R2

3 DC sous Windows Server 2012 R2 et 2 DC sous Windows Server 2019

→ Niveau fonctionnel = Windows Server 2012 R2

Le schéma

Un schéma est la définition des **attributs** qui font partie d'un annuaire distribué et sont similaires aux **champs** et **tables** d'une BDD.

ex: Pour un utilisateur, les attributs peuvent être :

- Un identifiant : SID, GUID, name, …
- Une classe : user
- Une information système : last-logon, …

 Attributs liés au schéma : lien entre objets

La réplication

La réplication amène la **redondance de données**.
Ces données doivent être identiques sur les différents DC.
La réplication gère :

- La base d'annuaire AD
- Les GPO
- Les scripts
- Les DNS

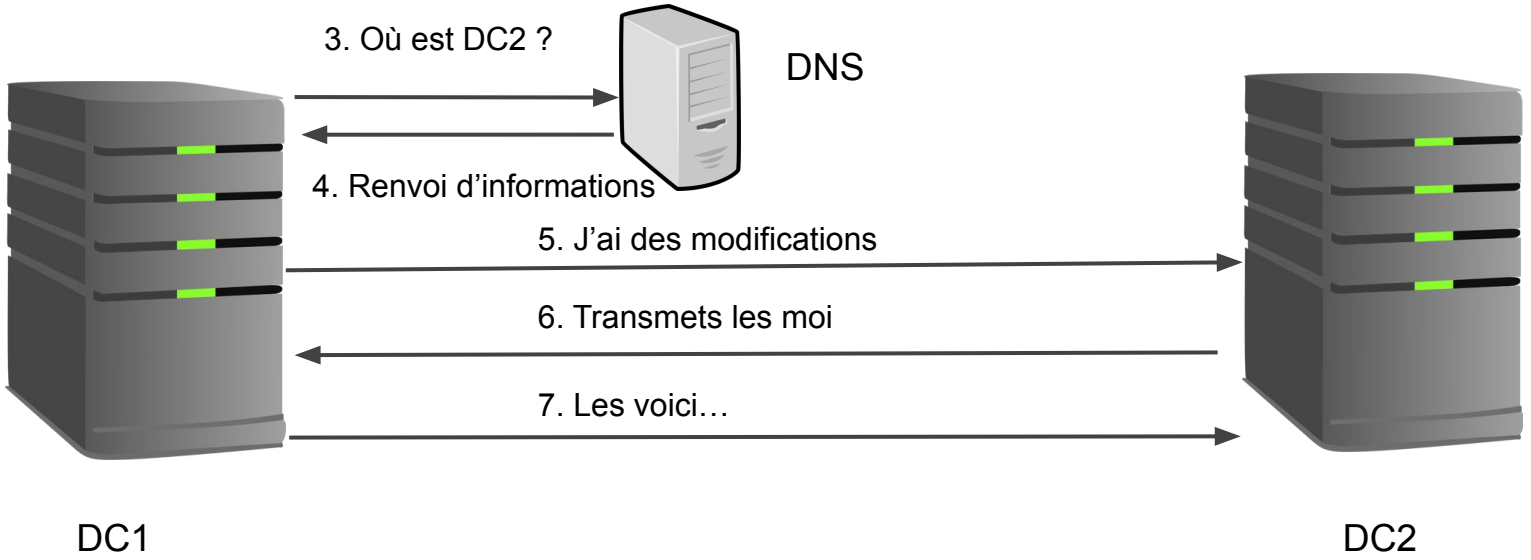
Aller plus loin

Le processus de réplication

Suite à une modification d'objet sur DC1 :

- DC1 demande au **KCC** (*Knowledge Consistency Checker*) s'il y a d'autres DC
- Le KCC indique qu'il y a un DC2
- DC1 demande au DNS où est le DC2
- Le DNS envoi les informations concernant le DC2
- DC1 indique au DC2 qu'il a des modifications
- DC2 demande à DC1 quelles sont ces modifications
- DC1 envoie les modifications à DC2
- DC2 met à jour sa BDD

Schéma du processus de réplication

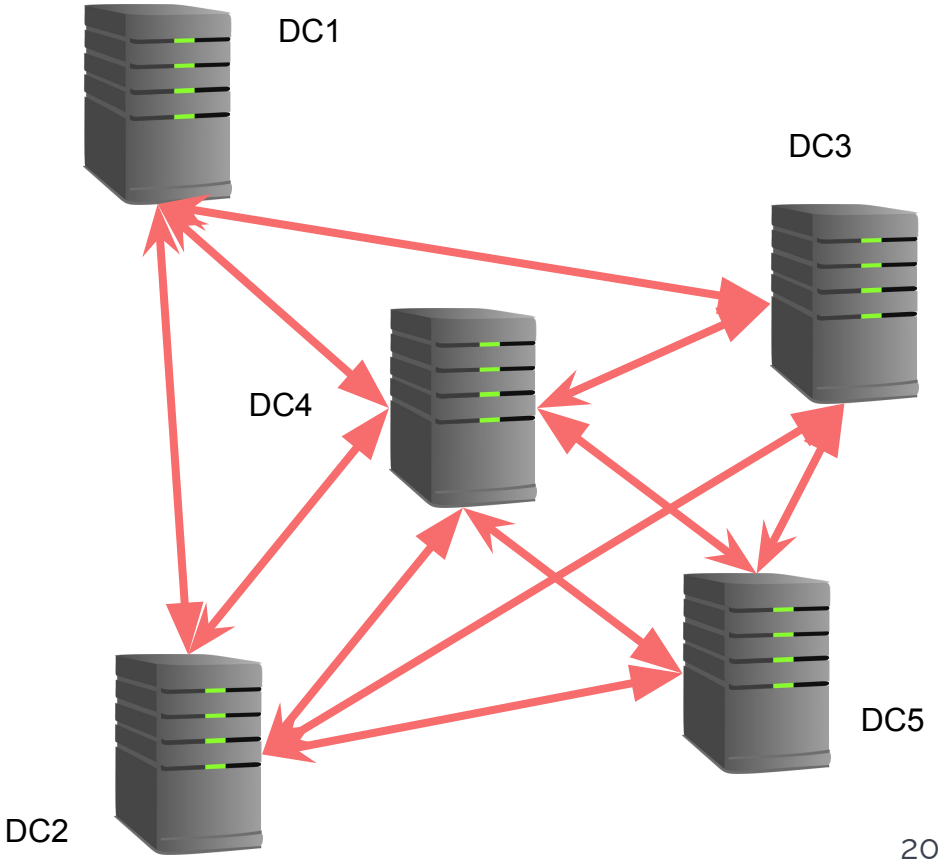


1. Modification d'objets sur DC1
2. Le KCC indique qu'il y a un DC2

8. DC2 se mets à jour

Le KCC

Le KCC est présent sur tous les DC.
Il s'occupe de générer une **topologie de réplication** entre les DC.
A chaque modification de topologie (ajout, suppression, déplacement de DC), le KCC régénère une nouvelle topologie.
Le fonctionnement est intra et inter site.



Intervalle de réplication

L'intervalle de réplication est :

- En inter-site de 180 min par défaut (15 min au minimum)
- En intra-site de 5 min par défaut, sauf actions liées à la sécurité.

Un petit intervalle réduit la latence, mais augmente la quantité de trafic réseau. Pour tenir à jour les partitions d'annuaire de domaine, une faible latence est recommandée.

Les 5 rôles FSMO

FSMO pour *Flexible Single Master Operation*.

A l'echelle de la forêt :

Maître de schéma (*Schema master*) → 1 seul par forêt (obligatoire)

- Gère les MAJ du schéma

Maître d'attribution de noms de domaine (*domain naming master*) → 1 seul par forêt

- Gère les noms de domaines, peut les supprimer, les ajouter, …

Les 5 rôles FSMO

A l'échelle du domaine :

Maître RID (*RID master*) → 1 seul par domaine
- Gère les attributions de RID/SID

Maître d'infrastructure (*infrastructure master*) → 1 seul par domaine
- Gère les relations des objets entre domaine

Émulateur PDC (PDC emulator) → 1 seul par domaine (primordial)
- Gère la synchronisation du temps, processus de verrouillage de comptes, …

FSMO - les bonnes pratiques

Ne pas avoir qu'un seul DC avec tous les rôles FSMO.
Par défaut le premier DC d'une nouvelle forêt cumule les cinq rôles.

Dès que possible avoir au moins 2 DC et séparer les rôles par zone forêt et domaine.
Idéalement : 5 DC avec un rôle installé sur chacun

Ne pas confondre !

Ne pas confondre :

- Les rôles de serveur Windows Server (DHCP, DNS, AD FS, etc.)
- Les rôles AD (AD DS, AD FS, etc.) <= qui sont des rôles serveurs !
- Les rôles FSMO (RID master, etc.) <= qui sont des rôles de DC AD DS

WILD CODE SCHOOL

Les objets AD

Définition

Chaque objet AD :

- Est une **instance** d'une classe dans le schéma
- Possède les attributs de sa classe

ex. : Un objet utilisateur existe en tant qu'instance de la classe utilisateurs et possède tous ses attributs.

Les objets sont de 3 types :

- **Les ressources** (poste de travail, dossiers partagés, agendas, …)
- **Les utilisateurs** (comptes individuels et groupes)
- **Les services** (courrier électronique,…)

Les attributs

Les attributs définissent :

- Les éléments d'information qu'une classe peut contenir (et donc une instance de cette classe).
- Les caractéristiques et les informations qu'un objet peut contenir.

Chaque classe d'objet a son propre jeu d'attribut.
Ils sont définis par le **schéma AD**.

Les attributs

```
PS C:\Lab> Get-ADUser -Filter * -Properties * | Where-Object {$_.Name -eq "User1"}

AccountExpirationDate                 :
accountExpires                        : 9223372036854775807
AccountLockoutTime                    :
AccountNotDelegated                   : False
AllowReversiblePasswordEncryption     : False
AuthenticationPolicy                  : {}
AuthenticationPolicySilo              : {}
BadLogonCount                         : 0
badPasswordTime                       : 0
badPwdCount                           : 0
CannotChangePassword                  : False
CanonicalName                         : lab.lan/LabUtilisateurs/User1
Certificates                          : {}
City                                  :
CN                                    : User1
codePage                              : 0
Company                               : MyCompany
```

Exemple d'attribut : la classe

Cet attribut définit le type d'un objet (*conteneur* ou *leaf objects*).

Conteneur :

- Conteneur natif (*container*)
- Unité d'organisation (*Organizational Units* ou *OU*)
- Groupe (*group*)

Leaf objects:

- Utilisateur (*user*)
- Ordinateur (*computer*) → Client ou serveurs
- Imprimante (printer)

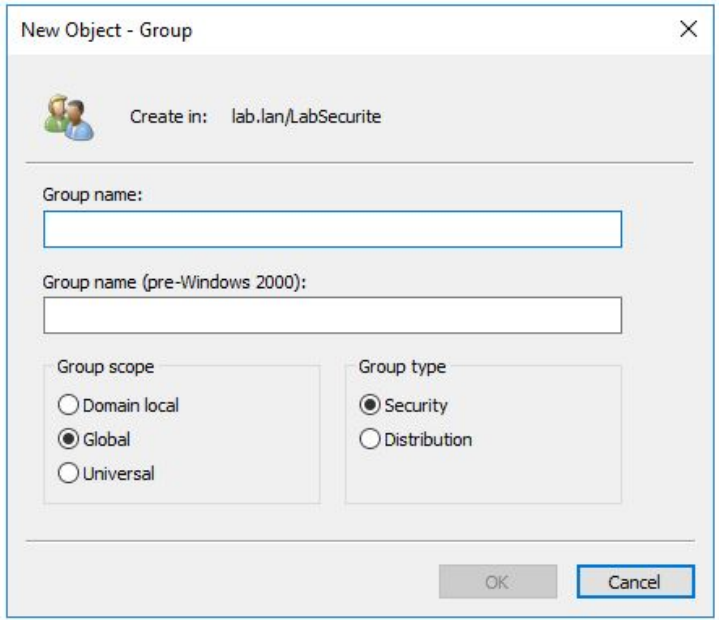
Exemple 1 de classe : les groupes

L'étendue:

- Domaine local
- Globale
  - Dans tous les domaines approuvés
  - Peut contenir des groupes du domaine
- Universelle
  - Portée maximale (ensemble de la forêt)
  - Peut contenir tous les objets de la forêt

Le type:

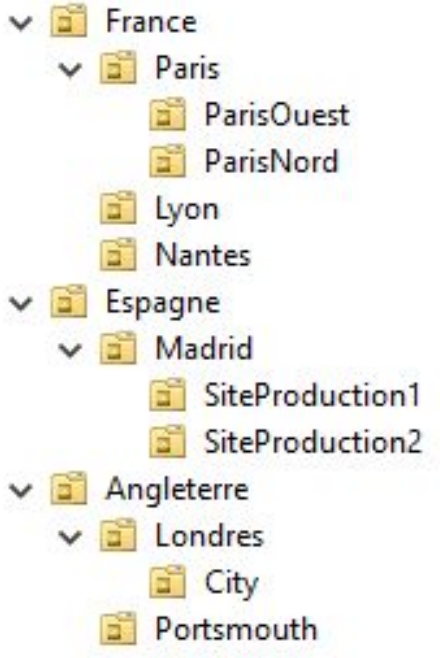
- Sécurité ⇒ Autorisation
- Distribution ⇒ Liste de distribution



Exemple 2 de classe : les OU

Les OU sont des **Unités d'Organisation**
(*Organizational Unit* ou **OU**)

Une OU est un objet de classe conteneur, qui est utilisé pour hiérarchiser l'AD.
Une OU peut contenir d'autres classes d'objets.



Exemple 3 de classe : les id uniques

Le **GUID** (*Globally Unique Identifier*) :

- Unique au sein d'une forêt
- Attribué à un objet à la création (ne change jamais)
- Longueur codé sur 128 bits
- Attribut : **ObjectGUID**
- Composé de:
  - Nombres aléatoires (122 bits)
  - Nombres fixes (6 bits)
- ex:

3F2504E0-4F89-11D3-9A0C-0305E82C3301

Le **SID** (*Security Identifier*) :

- Unique au sein d'un domaine
- Attribué à un objet à la création et peut changer (ex changement de domaine)
- Longueur maximale de 256 caractères
- Attribut: **ObjectSID**
- Composé de:
  - Un Security principal domain SID
  - Un RID
- ex:

s-1-5-21-156063872-1535639461-3779917529-1134

Un autre identifiant : le DN

DN = *Distinguished Name*

- Unique au sein d'une forêt
- Correspond au chemin LDAP dans l'annuaire AD
- La longueur dépend de l'emplacement de l'objet dans l'AD
- Attribut : **DistinguishedName**
- Composé de :
  - Le nom du domaine
  - Le(s) conteneur(s) où se trouve l'objet
  - Le nom de l'objet

ex: cn=wilder,ou=RS,ou=utilisateurs,ou=Paris,ou=France,dc=masociete,dc=fr

Les identifiants uniques d'un objet

```
PS C:\Lab> $Var1 = Get-ADUser -Filter * -Properties * | Where-Object {$_.Name -eq "User1"}

PS C:\Lab> $Var1 | Select Name,DistinguishedName,ObjectGUID,ObjectSID | Format-List

Name                : User1
DistinguishedName   : CN=User1,OU=LabUtilisateurs,DC=lab,DC=lan
ObjectGUID          : 2c188179-b4cd-4f44-80cc-98e0720b8bfc
ObjectSID           : S-1-5-21-11617303-4238263364-3208815124-1103
```

WILD CODE SCHOOL

Bonnes pratiques d'administration

Gestion des identités et des accès

Principe de moindre privilège (**JEA**) :

- Limiter les droits d'accès au strict nécessaire
- Uniquement les droits nécessaires pour exécuter des tâches précises

Utiliser des comptes d'administration séparés :

- Avoir des comptes distincts pour les tâches administratives et les tâches quotidiennes (**règles des Tiers** ou ***Tiering Model***)

Changer le mot de passe du compte administrateur local des clients :

- Utilisation de LAPS

Politique et contrôles d'accès

Renforcer les politiques de mot de passe :

- Établir des règles strictes pour la création de mots de passe robustes (Utilisation de la complexité, de la longueur et des délais d'expiration)

Séparation identité des utilisateurs et ressources (**AGDLP**) :

- Les droits d'accès ne sont pas directement sur les utilisateurs
- Les droits d'accès sont sur les groupes
- Les ressources ne connaissent que les groupes

Opérations sur le réseau

Protéger les DC :

- Les mettre dans un réseau dédié et sécurisé (vlan, DMZ)

Effectuer les MAJ :

- Application régulière des MAJ de sécurité (maintenance cyclique)
- Faire évoluer les systèmes (OS, appliance)

Sécuriser les communications LDAP :

- Protéger les transmissions avec LDAPS (LDAP sur SSL/TLS)

Surveillance

Effectuer des audits réguliers et surveiller les logs :

- Audit interne ou externe
- Examiner les logs pour détecter les activités anormales

Mettre en place une stratégie de sauvegarde et de récupération :

- Préparer des plans de sauvegarde et de restauration pour les urgences
- Utiliser la règles 3, 2, 1

Microsoft Tiering Model

- Modèle de sécurité qui sépare les ressources et les administrateurs pour limiter les risques de propagation d'attaques dans l'environnement AD.
- On sépare les composants de l'infrastructure en fonction de leur niveau d'importance
- On rend ces différentes couches hermétiques les unes des autres

Séparation en niveaux de sécurité ou **tiers**.

Microsoft Tiering Model

Tier 0 :

DC, administrateurs d'entreprise et autres actifs avec contrôle direct sur l'ensemble de l'environnement AD (serveurs AD, PKI, ADFS …).

Couche la plus importante.

Un admin T0 peut gérer uniquement des composants de la couche T0. Il ne peut RDP que sur des serveurs intégrés à ce niveau. L'accès ne doit PAS être utilisé pour se connecter à des serveurs d'une couche inférieure.

Microsoft Tiering Model

Tier 1 :
Serveurs et applications, administrateurs qui gèrent les services serveur et les applications d'entreprise (SCCM, WSUS, SCOM, etc.).
Un admin T1 gère les serveurs applicatifs et middlewares de l'entreprise. Il ne doit pas être utilisé pour se connecter à une couche supérieure ou inférieure.

Microsoft Tiering Model

Tier 2 :

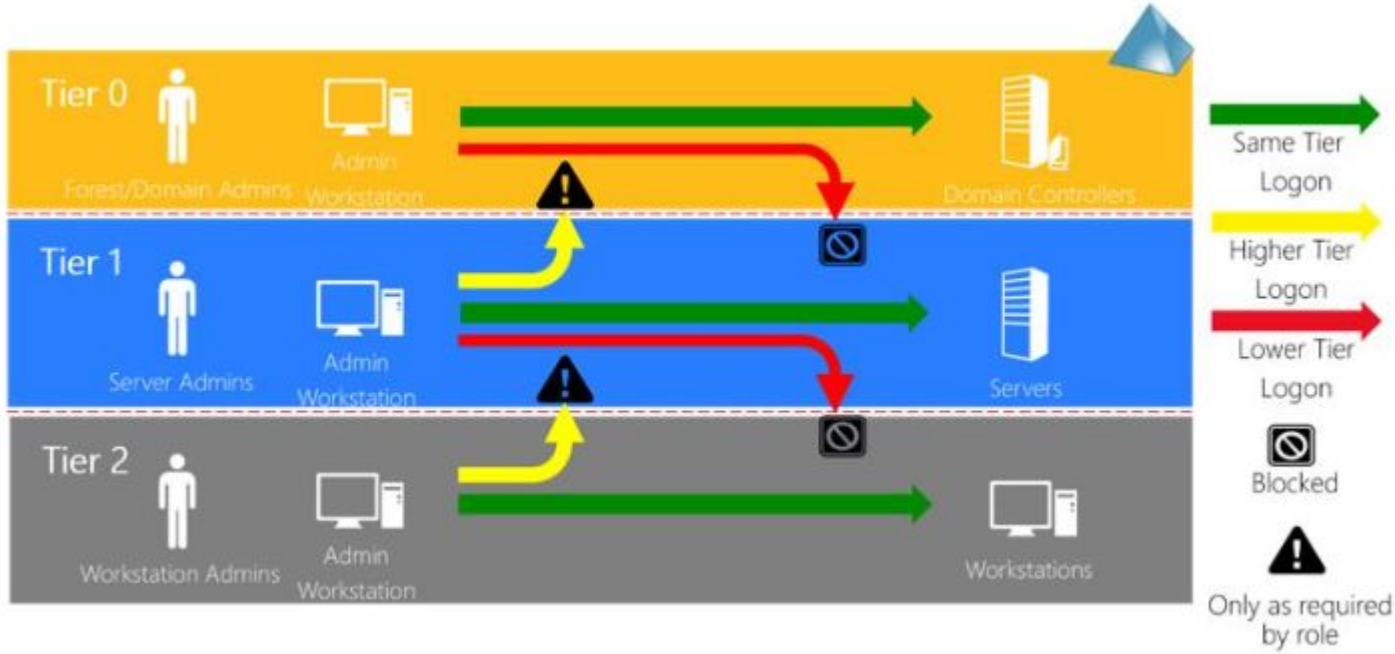
Postes de travail des utilisateurs finaux, y compris ceux des administrateurs.

Couche la plus « à risque » :

- Erreurs et intrusions (phishing, ransomware, etc.)
- Périphériques mobiles (smartphone, tablette, …).

Un admin T2 gère les postes de travail des utilisateurs. Ce compte ne doit pas être utilisé pour accéder aux couches supérieures.

Microsoft Tiering Model



JIT & JEA

JIT (*Just-In-Time*) :
Permet aux administrateurs d'obtenir les privilèges nécessaires pour une tâche spécifique pendant une période limitée.
A l'expiration, les droits élevés sont révoqués automatiquement.

JEA (*Just Enough Administration*) :
Limite les privilèges des administrateurs aux seuls droits nécessaires pour effectuer une tâche spécifique, réduisant ainsi les risques associés à l'utilisation de comptes à privilèges élevés.

JIT & JEA

En général, dans les entreprises, les comptes à privilèges élevés gardent ces privilèges tout au long de leur vie.
⇒ En cas de compromission de compte ces privilèges constituent un trou de sécurité

JIT & JEA

Solution :
Avoir les privilèges nécessaire à un instant donné pour une période donnée.

- Nécessitée d'une approbation par une ou plusieurs personnes (légitimité de la demande ?)
- Détails de l'intervention à donner

AGDLP

Signification :

- A -> Account => Les utilisateurs
- G -> Global groups => Les groupes métiers
- DL -> Domain Local Groups => Les groupes de droits
- P -> Permissions => Les permissions sur les ressources

AGDLP (suite)

AGDLP est une méthode d'organisation des permissions geré dans AD DS.
Elle évite la **TRÈS** mauvaise pratique de donner des droits d'accès directement à un utilisateur.

Principe :

- Un utilisateur appartient à un groupe métier
- Ce groupe métier appartient à un ou plusieurs groupes de droits
- Ces groupes de droit ont des permissions sur des ressources

AGDLP (suite)

Exemple 1 :

- Bob et Alice sont 2 comptables nouvellement embauchés
- Ils doivent avoir accès aux dossiers suivants :
  - "Compta" en lecture et écriture
  - "Paye" en lecture

=> On ne donne pas directement accès aux 2 dossiers aux 2 utilisateurs !

AGDLP (suite)

- Bob et Alice sont tous les 2 mis dans le groupe **Grp_Compta**
- Le groupe **Grp_Compta** est membre des groupes :
  - **Grp_Compta_RW** => Accès NTFS RW sur le dossier partagé **Compta**
  - **Grp_Paye_R** => Accès NTFS R sur le dossier partagé **Paye**

=> Bob et Alice ont accès aux 2 dossiers avec les bons droits.

AGDLP (suite)

Exemple 2 :

- Alice doit avoir accès à un dossier "Finance" en lecture, mais pas bob !

=> On ne donne pas directement accès à ce dossier à Alice !

AGDLP (suite)

- Alice est ajoutée au groupe (transverse) **Grp_Compta_Finance**
- Le groupe **Grp_Compta_Finance** est membre du groupe :
  - **Grp_Compta_Finance_R** => Accès NTFS R sur le dossier partagé **Finance**

=> Alice a accès au dossier avec les bons droits.

Conclusion

- Détails sur les services systèmes et réseaux
- Niveau fonctionnel, schéma, réplication
- Rôles FSMO
- Objets AD, classe d'attribut
- Bonnes pratiques d'administration
- Sécurité avancée (Tiers Model, JIT&JEA, AGDLP)