



Projet : Mise en place d'une authentification centralisée Active Directory pour systèmes Linux

Contexte

Le projet a été conçu et réalisé de manière autonome dans un environnement de laboratoire personnel, dans une démarche de montée en compétences et de mieux comprendre le rôles d'un Active Directory au sein d'un réseau d'entreprise. Bien que les environnements utilisateurs soient majoritairement Windows, l'intégration de Linux à Active Directory répond à des besoins réels en entreprise, notamment pour la gestion centralisée des accès aux serveurs. L'ensemble des choix techniques, configuration et diagnostics a été réalisé de manière autonome.

Objectif du projet

Mettre en place une authentification centralisée Active Directory pour des systèmes Linux, en reproduisant un cas réel d'infrastructure d'entreprise : gestion des comptes, accès SSH et droits administrateur, sans solutions tierces propriétaires.

L'objectif principal était de comprendre et maîtriser les mécanismes sous-jacents (DNS, Kerberos, SSSD, PAM), et pas uniquement d'obtenir un fonctionnement "clé en main".

Architecture mise en place

- Windows Server 2022
 - Active Directory Domain Services
 - DNS
 - Kerberos (KDC)
- Debian GNU/Linux
 - realmd
 - SSSD
 - PAM / NSS
- pfSense pour le routage
- Domaine : homelab.local

Windows Server agit comme autorité d'authentification, Linux délègue l'authentification et le contrôle d'accès à AD.



Extension de l'infrastructure : Ajout d'un client Windows 10

Contexte technique

Après avoir intégré avec succès le client Linux au domaine Active Directory, j'ai souhaité étendre l'infrastructure en ajoutant un poste client Windows pour :

- Valider l'interopérabilité entre clients Linux et Windows dans le même domaine
- Reproduire un environnement d'entreprise réaliste (parc mixte)
- Tester les fonctionnalités natives d'Active Directory sur Windows

Choix de Windows 10 LTSC 32 bits

Raison du choix : Optimisation des ressources matérielles

Le choix de Windows 10 LTSC (Long-Term Servicing Channel) en version 32 bits a été motivé par des contraintes de performances de la machine hôte :

- **Légèreté** : Windows 10 LTSC est une version allégée sans applications préinstallées (Microsoft Store, Cortana, Edge)
- **Consommation mémoire réduite** : Version 32 bits consomme moins de RAM qu'une version 64 bits
- **Stabilité de la machine hôte** : Éviter les ralentissements ou le freeze du PC physique pendant la virtualisation de plusieurs machine simultanément
- **Performance équilibrée** : Maintenir des performances comparables au client Debian pour un environnement de test homogène

Cette approche permet de conserver un laboratoire malgré les limitations matérielles, tout en reproduisant un scénario professionnel réaliste.

Intégration au domaine

Processus :

- Configuration DNS : Serveur DNS pointant vers le contrôleur de domaine (192.168.2.3)
- Jonction au domaine via Win + R → sysdm.cpl
- Redémarrage de la machine puis connexion avec un compte du domaine (simple-user)

Résultat : Intégration réussie sans incident technique. Durée d'intégration : ~15 minutes



Validation fonctionnelle

Les tests suivants ont été réalisés pour valider l'intégration :

- Authentification avec un compte du domaine réussie
- Application des stratégies de groupe (GPO) validée
- Accès aux ressources réseau partagées fonctionnel
- Résolution DNS et connectivité vers le contrôleur de domaine opérationnelle

Bénéfices de cette extension

Cette extension de l'infrastructure permet de démontrer :

- **Interopérabilité** : Gestion unifiée de clients hétérogènes (Linux et Windows) dans le même domaine
- **Scalabilité** : Capacité à étendre le laboratoire avec de nouveaux clients
- **Pragmatisme** : Adaptation aux contraintes matérielles tout en maintenant un environnement représentatif
- **Polyvalence** : Maîtrise de l'intégration AD pour différents systèmes d'exploitation

Flux réseau et modèle OSI, Intégration Linux ↔ Active Directory

Le but de cette section est de décrire les flux réseau nécessaires à l'authentification centralisée des systèmes Linux dans un domaine Active Directory, en les associant aux couches du modèle OSI et au sens des communications.

Vue d'ensemble des flux

L'authentification repose sur une chaîne de dépendances strictes :

1. Résolution DNS du domaine Active Directory
2. Authentification via Kerberos
3. Résolution des identités et groupes via LDAP
4. Accès aux services (SSH) selon les autorisations

Tout dysfonctionnement sur un flux en amont empêche les étapes suivantes.



Service	Rôle	Ports	Transport	OSI (principales couches)	Sens du flux
DNS	Résolution des noms et services AD (A, PTR, SRV)	53	UDP / TCP	L4, L7	Client Linux → DC
Kerberos	Authentification et délivrance de tickets	88	UDP / TCP	L4, L7	Client Linux → DC
LDAP	Résolution des comptes et groupes	389	TCP	L4, L7	Client Linux → DC
SSH	Accès distant au système Linux	22	TCP	L4, L7	Client utilisateur → Linux
SMB	Authentification Windows	445	TCP	L4,L7	Client Windows → DC
Kerberos	Accès au ressources	88	UDP / TCP	L4,L7	Client Windows → DC

Détail par service : DNS

- Indispensable au fonctionnement d'Active Directory
- Utilisé pour :
 - localiser les contrôleurs de domaine (enregistrements SRV),
 - résoudre le FQDN du domaine et des services Kerberos.
- Une configuration DNS incorrecte empêche Kerberos de fonctionner.

Flux : Client Linux → Serveur DNS AD

Kerberos

- Assure l'authentification centralisée
- Repose sur :
 - une résolution DNS correcte,
 - une synchronisation horaire stricte (NTP).
- Toute dérive temporelle ou incohérence DNS entraîne des erreurs d'authentification.

Flux : Client Linux → KDC (DC Active Directory)



LDAP / LDAPS

- Utilisé par SSSD pour :
 - récupérer les identités utilisateurs,
 - résoudre les groupes Active Directory,
 - appliquer les règles d'autorisation.
- LDAPS permet le chiffrement des échanges LDAP si activé.

Flux : Client Linux → Active Directory (LDAP)

SSH

- Permet l'accès distant au système Linux
- L'autorisation SSH dépend :
 - de l'authentification Kerberos,
 - du filtrage par groupes AD via PAM et SSSD.
- SSH n'est accessible qu'aux utilisateurs explicitement autorisés.

Flux : Client utilisateur → Serveur Linux

Synthèse

- Les flux sont **majoritairement sortants depuis le client Linux vers le contrôleur de domaine**
- DNS et Kerberos sont des **pré-requis critiques**
- L'architecture respecte une séparation claire :
 - authentification (Kerberos),
 - résolution des identités (LDAP / SSSD),
 - autorisation (PAM),
 - accès au service (SSH).



Principe de fonctionnement

- Authentification Linux via Kerberos
- Résolution des identités et groupes via SSSD
- Contrôle d'accès assuré par PAM
- Accès explicitement filtré par groupes Active Directory
 - linux-users : accès SSH
 - linux-admins : accès SSH + sudo
- Aucun accès implicite via Domain Users

Problèmes rencontrés (principaux)

- DNS : résolution FQDN incomplète, enregistrements AD manquants
 - correction des enregistrements A / PTR / SRV côté AD
- Kerberos : erreurs lors de kinit malgré des identifiants valides
 - incohérences DNS / realm, comptes AD désactivés, configuration krb5.conf
- SSSD / PAM : authentification réussie mais accès SSH refusé
 - filtrage par groupes AD mal appliqué, cache SSSD non purgé
- Home directories : répertoires utilisateurs absents
 - activation de la création automatique via PAM

Ces incidents ont permis de différencier clairement authentification, autorisation et résolution de noms.

Résultat final

- Intégration complète et fonctionnelle entre Active Directory et Linux
- Accès SSH contrôlé par groupes AD
- Gestion des droits sudo via Active Directory
- Création automatique des répertoires utilisateurs
- Infrastructure stable et reproductible

La solution a été validée par des tests fonctionnels incluant des comptes autorisés et non autorisés, ainsi que des vérifications des accès SSH et des droits sudo.



Compétences mises en avant

- Administration Active Directory
- DNS (A, PTR, SRV)
- Kerberos
- Intégration Linux / AD (SSSD, PAM)
- Gestion des accès et des privilèges
- Diagnostic et résolution d'incidents systèmes

Topologie Réseau

