Configuration LDAP ET Kerberos

Introduction

Dans le cadre de ce TP de réseau, nous avons mis en place un système d'authentification centralisée basé sur **LDAP (Lightweight Directory Access Protocol)** et **Kerberos**. L'objectif principal est de centraliser la gestion des utilisateurs et d'assurer une authentification sécurisée au sein d'un réseau.
LDAP permet de stocker et d'organiser les informations des utilisateurs dans un annuaire hiérarchique, tandis que Kerberos fournit un mécanisme d'authentification forte basé sur un système de tickets.

Pour automatiser l'installation et la configuration de ces deux services, nous avons utilisé un **script shell**, ce qui garantit une installation reproductible et plus rapide. Ce rapport présente les étapes suivies, les commandes utilisées, ainsi que les tests réalisés pour vérifier le bon fonctionnement du système.

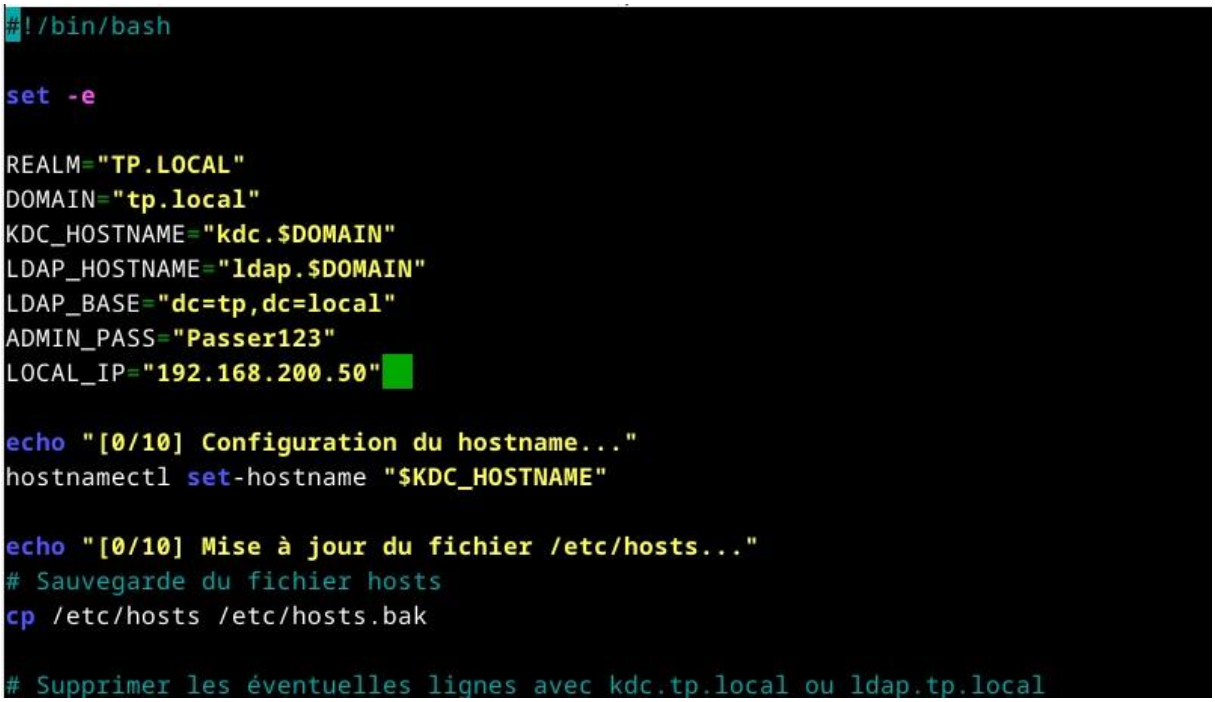
Objectifs du TP :

- Installer et configurer un serveur LDAP pour la gestion des utilisateurs.
- Mettre en place Kerberos pour l'authentification sécurisée.
- Automatiser le déploiement à l'aide d'un script.
- Vérifier le bon fonctionnement de l'ensemble

1. Création du Script

```
phone1@phone:~$ sudo nano ldap-kerb
```

Nous créons un fichier nommé **ldap-kerb** pour y stocker le script d'installation.



```
#!/bin/bash

set -e

REALM="TP.LOCAL"
DOMAIN="tp.local"
KDC_HOSTNAME="kdc.$DOMAIN"
LDAP_HOSTNAME="ldap.$DOMAIN"
LDAP_BASE="dc=tp,dc=local"
ADMIN_PASS="Passer123"
LOCAL_IP="192.168.200.50"

echo "[0/10] Configuration du hostname..."
hostnamectl set-hostname "$KDC_HOSTNAME"

echo "[0/10] Mise à jour du fichier /etc/hosts..."
# Sauvegarde du fichier hosts
cp /etc/hosts /etc/hosts.bak

# Supprimer les éventuelles lignes avec kdc.tp.local ou ldap.tp.local
```

#!/bin/bash : ca s'appelle **SHEBANG ,** Indique que le script doit être exécuté avec l'interpréteur Bash.

set -e : Le script s'arrête automatiquement si une commande échoue.

REALM=… , LOCAL_IP="192.168.200.50" : Ces variables définissent les paramètres de configuration pour LDAP et Kerberos :

- **REALM** : domaine Kerberos (en majuscules).
- **DOMAIN** : nom DNS utilisé.
- **KDC_HOSTNAME et LDAP_HOSTNAME** : noms d'hôte complets.
- **LDAP_BASE** : base DN de l'annuaire LDAP.
- **ADMIN_PASS** : mot de passe administrateur (à protéger).
- **LOCAL_IP** : IP locale du serveur.

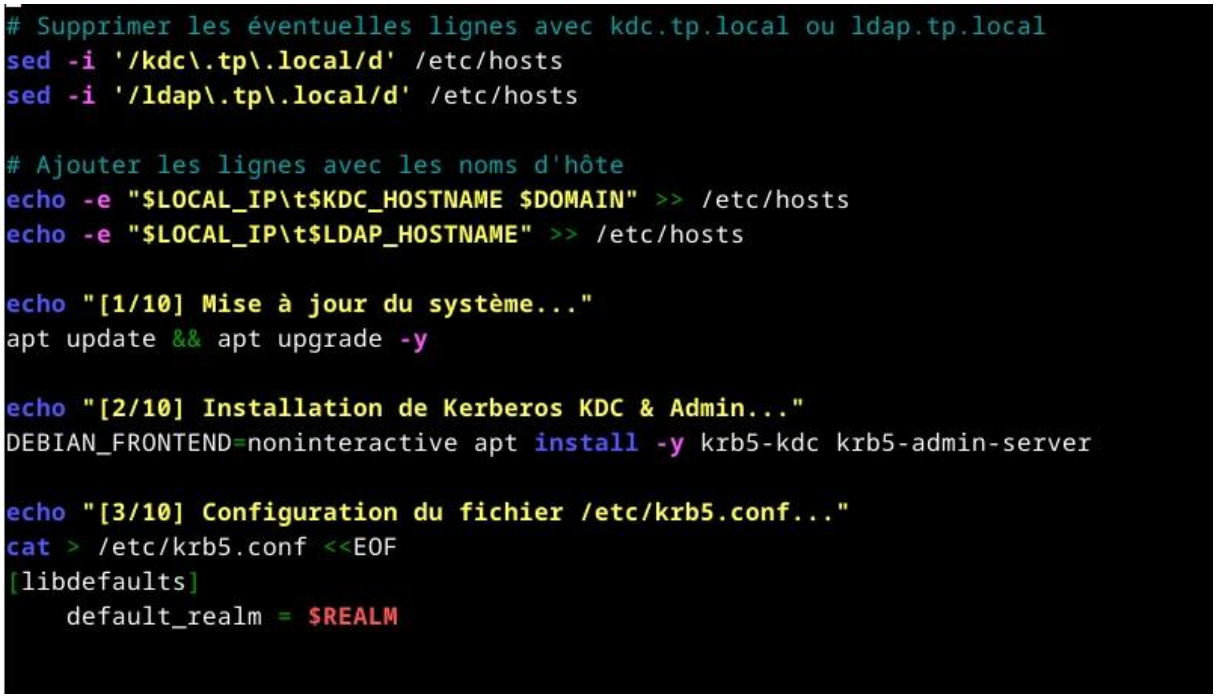
echo "[0/10] Configuration du hostname..."

hostnamectl set-hostname "$KDC_HOSTNAME"

- Affiche un message et configure le nom d'hôte du serveur avec **hostnamectl**.

echo "[0/10] Mise à jour du fichier /etc/hosts..."
cp /etc/hosts /etc/hosts.bak

- Affiche un message et sauvegarde le fichier **/etc/hosts** avant modification.



```
# Supprimer les éventuelles lignes avec kdc.tp.local ou ldap.tp.local
sed -i '/kdc\.tp\.local/d' /etc/hosts
sed -i '/ldap\.tp\.local/d' /etc/hosts

# Ajouter les lignes avec les noms d'hôte
echo -e "$LOCAL_IP\t$KDC_HOSTNAME $DOMAIN" >> /etc/hosts
echo -e "$LOCAL_IP\t$LDAP_HOSTNAME" >> /etc/hosts

echo "[1/10] Mise à jour du système..."
apt update && apt upgrade -y

echo "[2/10] Installation de Kerberos KDC & Admin..."
DEBIAN_FRONTEND=noninteractive apt install -y krb5-kdc krb5-admin-server

echo "[3/10] Configuration du fichier /etc/krb5.conf..."
cat > /etc/krb5.conf <<EOF
[libdefaults]
    default_realm = $REALM
```

sed -i '/kdc.tp.local/d' /etc/hosts

sed -i '/ldap.tp.local/d' /etc/hosts

- **Rôle** : Supprime toutes les lignes contenant **kdc.tp.local** ou **ldap.tp.local** dans /etc/hosts.

Installation de Kerberos (KDC + Admin Server)

DEBIAN_FRONTEND=noninteractive apt install -y krb5-kdc krb5-admin-server

- **Rôle** : Installe les paquets nécessaires pour Kerberos :

1. **krb5-kdc** : **Serveur KDC** (Key Distribution Center).
2. **krb5-admin-server :** Outils d'administration.

- **Détails** :

  - **DEBIAN_FRONTEND=noninteractive** : Désactive les prompts interactifs (utile pour les scripts).

```
[realms]
    $REALM = {
        kdc = $KDC_HOSTNAME
        admin_server = $KDC_HOSTNAME
    }

[domain_realm]
    .$DOMAIN = $REALM
    $DOMAIN = $REALM
EOF

echo "[4/10] Initialisation de la base Kerberos..."
krb5_newrealm

echo "[5/10] Création de l'utilisateur Kerberos admin/admin avec mot de passe.."
echo "addprinc -pw $ADMIN_PASS admin/admin" | kadmin.local
```

Nous définissons les déférents paramètre dans le fichier **/etc/krb5.conf**

```
# Lancer l'installation interactivement
apt install -y slapd ldap-utils

echo "[7/10] Installation des modules SASL et configuration..."
apt install -y sasl2-bin libsasl2-modules-gssapi-mit

echo "[8/10] Configuration SASL pour OpenLDAP..."
mkdir -p /etc/ldap/sasl2
cat > /etc/ldap/sasl2/slapd.conf <<EOF
pwcheck_method: saslauthd
mech_list: GSSAPI
EOF

echo "[9/10] Configuration de saslauthd pour Kerberos..."
sed -i 's/^START=.*/START=yes/' /etc/default/saslauthd
sed -i 's/^MECHANISMS=.*/MECHANISMS="kerberos5"/' /etc/default/saslauthd
systemctl restart saslauthd

echo "[10/10] Configuration de ldap.conf pour SASL GSSAPI..."
```

3. Nous installons **slapd** et ces dépendances.
4. Crée le répertoire **/etc/ldap/sasl2** (si inexistant) avec **mkdir -p**.
5. Configure SASL pour :

a. Utiliser **saslauthd** comme méthode de vérification des mots de passe.
b. Restreindre les mécanismes d'authentification à GSSAPI (Kerberos).

6. Active **saslauthd** au démarrage (**START=yes**).
7. Définit le mécanisme d'authentification sur **kerberos5.**

8. **Redémarrage : systemctl restart saslauthd** pour appliquer les changements.

```
echo "[10/10] Configuration de ldap.conf pour SASL GSSAPI..."
cat > /etc/ldap/ldap.conf <<EOF
BASE    $LDAP_BASE
URI     ldap://$LDAP_HOSTNAME
SASL_MECH GSSAPI
EOF

echo "✔ Installation terminée."
echo "➡ Le mot de passe admin Kerberos est : $ADMIN_PASS"
echo "➡ Exécutez maintenant :"
echo "  kinit admin/admin@$REALM"
echo "  ldapwhoami -Y GSSAPI"
```

- **BASE** : DN de base pour les requêtes LDAP (stocké dans **$LDAP_BASE**).
- **URI** : URL du serveur LDAP (utilise la variable **$LDAP_HOSTNAME**).
- **SASL_MECH** : Spécifie l'utilisation de GSSAPI (Kerberos) pour SASL.

2. Exécution du Script

```
phone1@phone:~$ sudo chmod +x ldap-kerb
phone1@phone:~$ sudo ./ldap-kerb
[0/10] Configuration du hostname...
```

- **sudo chmod +x ldap-kerb** Rend le script exécutable
- **sudo ./ldap-kerb** Lance le script avec les droits root

```
phone1@phone:~$ kinit admin/admin@TP.LOCAL
Password for admin/admin@TP.LOCAL:
phone1@phone:~$ systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access>
     Loaded: loaded (/etc/init.d/slapd; generated)
    Drop-In: /usr/lib/systemd/system/slapd.service.d
             └─slapd-remain-after-exit.conf
     Active: active (running) since Sat 2025-06-21 17:54:50 EDT; 1min 50s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 4963 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCC>
      Tasks: 3 (limit: 1029)
     Memory: 5.4M
        CPU: 46ms
     CGroup: /system.slice/slapd.service
             └─4970 /usr/sbin/slapd -h "ldap:/// ldapi:///" -g openldap -u open>
```

- Authentifie l'utilisateur **admin/admin** auprès du KDC Kerberos pour obtenir un TGT (Ticket Granting Ticket).
- On vérifie le bon fonctionnement de **kerberos** avec **systemctl status.**

3. Activation de LDAPS :

```
phone1@kdc:~$ sudo mkdir -p /etc/ldap/ssl
cd /etc/ldap/ssl
phone1@kdc:/etc/ldap/ssl$ sudo openssl req -x509 -newkey rsa:4096 -keyout key.pe
m -out cert.pem -days 3650 -nodes
```

- Crée un répertoire dédié **(/etc/ldap/ssl**) pour stocker les clés/certificats.
- Génère une paire de clés SSL/TLS :
    - **Algorithmes** : RSA 4096 bits (sécurisé).
    - **Options** :
        - **-x509** : Génère un certificat auto-signé.
        - **-nodes** : Ne pas chiffrer la clé privée (évite les prompts de mot de passe au démarrage de LDAP).
        - **-days 365** : Durée de validité (1 an).

```
-----
Country Name (2 letter code) [AU]:SN
State or Province Name (full name) [Some-State]:Dakar
Locality Name (eg, city) []:Dakar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SMARTTECH
Organizational Unit Name (eg, section) []:SN
Common Name (e.g. server FQDN or YOUR name) []:ldap.tp.local
Email Address []:SMARTTECH@gmail.com
```

- **Paramètres du certificat** (saisis interactivement) :

- **Pays (SN)** : Sénégal.
- **Ville/Organisation** : Dakar, SMARTTECH.
- **CN (Common Name)** : **ldap.tp.local** → Doit correspondre au FQDN du serveur LDAP**.**
- **Email** : SMARTTECH@gmail.com.

```
phone1@kdc:/etc/ldap/ssl$ sudo chown -R openldap:openldap /etc/ldap/ssl
phone1@kdc:/etc/ldap/ssl$ sudo chmod 640 /etc/ldap/ssl/key.pem
phone1@kdc:/etc/ldap/ssl$ sudo chmod 644 /etc/ldap/ssl/cert.pem
phone1@kdc:/etc/ldap/ssl$ sudo nano tls.ldif
```

- **Sécurité** :
  - **chown** : Donne la propriété des fichiers au user/group **openldap**.
  - **chmod** 640 (clé privée) : Lecture pour le propriétaire (**openldap**), lecture pour le groupe, aucun droit aux autres.
  - **chmod 644** (certificat) : Lecture pour tous (nécessaire pour les clients LDAPS).

nano tls.ldif : Ce fichier **LDIF** va contenir la configuration pour activer **TLS/SSL** dans **OpenLDAP**

```
dn: cn=config
changetype: modify
replace: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/ssl/cert.pem
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/ssl/cert.pem
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/ssl/key.pem
```

Configuration TLS via LDIF (tls.ldif)

Objectif :
Configure les chemins des fichiers TLS pour **OpenLDAP** :

- **Certificat CA** : Utilise **cert.pem** comme CA (auto-signé ici).
- **Certificat serveur** et **clé privée** : Pointent vers les fichiers générés précédemment.

```
phone1@kdc:/etc/ldap/ssl$ ldapmodify -x -D "cn=admin,cn=config" -W -H ldap://loc
alhost -f tls.ldif
Enter LDAP Password:
modifying entry "cn=config"
```

Application de la configuration TLS

- **Commandes clés** :
  - **-D "cn=admin,cn=config"** : Authentification en tant qu'admin LDAP.
  - **-W** : Demande le mot de passe interactivement.
  - **-H ldap://localhost** : Cible le serveur LDAP local.
- **Sortie** :
  modifying entry "cn=config" → Succès.

```
 GNU nano 7.2                    /etc/default/slapd

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldap:///  ldapi:/// ldaps:///"
```

Activation de LDAPS dans slapd

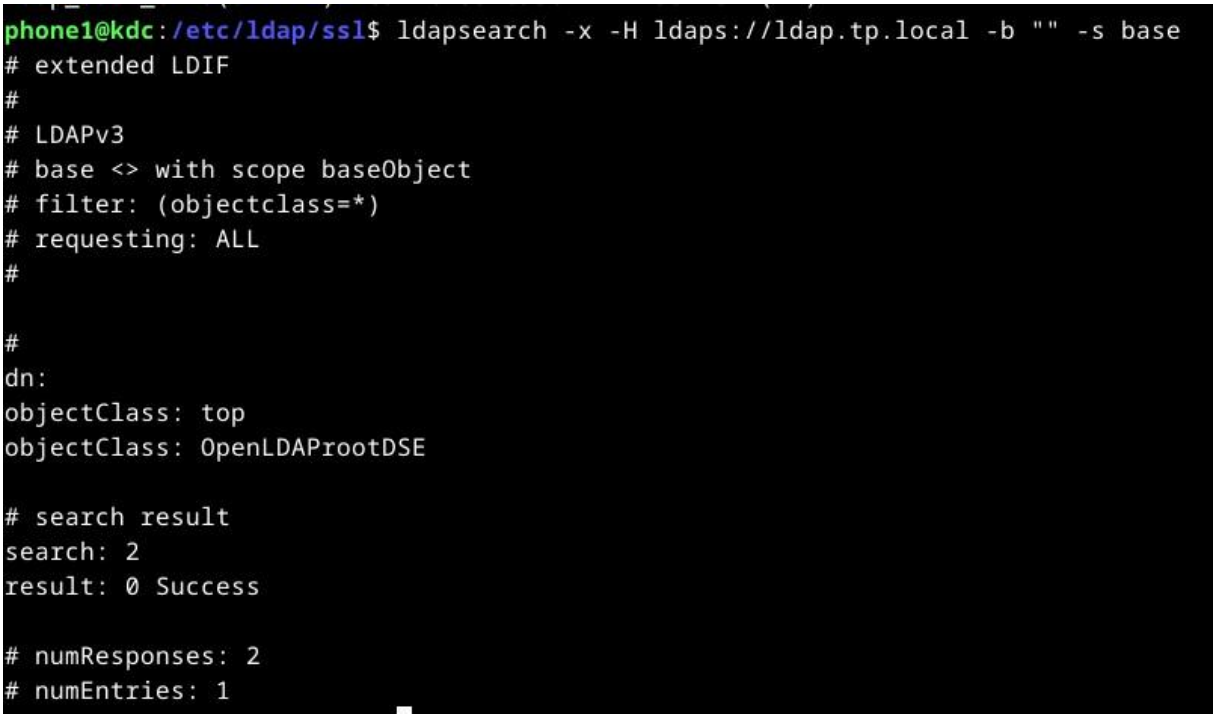
- **Effet** :
  - Active les écouteurs sur :
    - **ldap:///** (port 389, non chiffré).
    - **ldaps:///** (port 636, chiffré via TLS).
    - **ldapi:///** (socket Unix).

```
phone1@kdc:/etc/ldap/ssl$ sudo systemctl restart slapd
```

Redémarrage de slapd

4. TEST DE FONCTIONNEMENT :

a. LDAPS :



Cette capture montre le résultat d'une commande **ldapsearch** de base, qui interroge le **DSA (Directory System Agent)** de **OpenLDAP** pour obtenir des informations sur l'entrée racine (**RootDSE**).

1. Détails de la requête

- **Protocole** : **LDAPv3**
- **Base de recherche** : <> (racine, appelée **RootDSE**)
- **Scope** : **baseObject** (seule l'entrée exacte spécifiée est retournée)
- **Filtre** : (**objectClass=*)** (tous les objets)
- **Attributs demandés : ALL** (tous les attributs disponibles)

2. Réponse du serveur

- **Contenu de l'entrée RootDSE** :

  dn:

  objectClass: top

  objectClass: OpenLDAProotDSE

- **dn:** (vide) → Indique qu'il s'agit de l'entrée racine.
- **objectClass: OpenLDAProotDSE** → Contient des métadonnées techniques sur le serveur LDAP (non affichées ici, mais typiquement : versions, extensions supportées, naming contexts, etc.).
- **Statut** : **result: 0 Success**

3. Interprétation

- **Ce test confirme que** :
  - Le serveur LDAP est accessible et répond aux requêtes.
  - La connexion LDAP de base.

b. LDAP + KERBEROS

1. Authentification Kerberos (Admin)

```
phone1@kdc:~$ kinit admin/admin@TP.LOCAL
Password for admin/admin@TP.LOCAL:
```

- **Explication** :

- kinit : Obtient un **TGT (Ticket Granting Ticket)** pour l'admin Kerberos.
- admin/admin@TP.LOCAL : Principal administrateur .

2. Création du SPN pour LDAP

```
phone1@kdc:~$ sudo kadmin.local
Authenticating as principal root/admin@TP.LOCAL with password.
kadmin.local:  addprinc -randkey ldap/kdc.tp.local@TP.LOCAL
No policy specified for ldap/kdc.tp.local@TP.LOCAL; defaulting to no policy
Principal "ldap/kdc.tp.local@TP.LOCAL" created.
```

- **Explication** :
  - **kadmin.local** : Outil d'administration local de Kerberos (nécessite les droits root).
  - **addprinc -randkey** : Crée un **principal de service** (SPN) pour LDAP sans mot de passe, en générant une clé aléatoire.
  - **ldap/kdc.tp.local@TP.LOCAL** :
    - **ldap/** : Préfixe standard pour les services LDAP.
    - **kdc.tp.local** : **Correspond au FQDN du serveur LDAP** .
    - **TP.LOCAL** : Le royaume Kerberos configuré.
- **Pourquoi ?** :
  Kerberos a besoin d'un SPN pour authentifier le service LDAP. Sans cela, l'erreur "**Server not found in Kerberos database**" apparaît.**Génération du Keytab**

3. Génération du KEYTAB

```
kadmin.local:  ktadd -k /etc/ldap/ldap.keytab ldap/kdc.tp.local@TP.LOCAL
Entry for principal ldap/kdc.tp.local@TP.LOCAL with kvno 2, encryption type aes2
56-cts-hmac-sha1-96 added to keytab WRFILE:/etc/ldap/ldap.keytab.
Entry for principal ldap/kdc.tp.local@TP.LOCAL with kvno 2, encryption type aes1
28-cts-hmac-sha1-96 added to keytab WRFILE:/etc/ldap/ldap.keytab.
```

- **Explication** :
  - **ktadd** : Exporte le principal vers un **fichier keytab** (stocke les clés secrètes).
  - **/etc/ldap/ldap.keytab** : Emplacement standard pour le keytab LDAP.
  - **Sortie** : Confirme l'ajout des clés avec 2 algorithmes de chiffrement (AES-256 et AES-128).
- Le **keytab** permet à LDAP de s'authentifier auprès de Kerberos **sans saisie de mot de passe**. Critique pour les services systèmes.

```
phone1@kdc:~$ sudo chown openldap:openldap /etc/ldap/ldap.keytab
phone1@kdc:~$ sudo chmod 640 /etc/ldap/ldap.keytab
```

- **chown** : Donne au service **OpenLDAP** (**openldap**) la propriété du fichier.
- **chmod** 640 : Restreint l'accès au propriétaire (lecture/écriture) et au groupe (lecture seule).

4. Configuration de la Variable Kerberos dans slapd

```
phone1@kdc:~$ sudo nano /etc/default/slapd
```

```
# Additional options to pass to slapd
SLAPD_OPTIONS=""
export KRB5_KTNAME=/etc/ldap/ldap.keytab
```

Indique à OpenLDAP l'emplacement du keytab contenant les identifiants Kerberos du service LDAP.

5. Redémarrage de LDAP

```
phone1@kdc:~$ sudo systemctl restart slapd
```

6. Vérification des Principaux Kerberos (kadmin.local)

```
phone1@kdc:~$ sudo kadmin.local -q "listprincs"
Authenticating as principal root/admin@TP.LOCAL with password.
K/M@TP.LOCAL
admin/admin@TP.LOCAL
kadmin/admin@TP.LOCAL
kadmin/changepw@TP.LOCAL
krbtgt/TP.LOCAL@TP.LOCAL
ldap/kdc.tp.local@TP.LOCAL
phone1@kdc:~$ sudo klist -k /etc/ldap/ldap.keytab
Keytab name: FILE:/etc/ldap/ldap.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   2 ldap/kdc.tp.local@TP.LOCAL
   2 ldap/kdc.tp.local@TP.LOCAL
```

sudo kadmin.local -q "listprincs" :

- **Principaux Listés** :
  - **ldap/kdc.tp.local@TP.LOCAL** → **Service Principal** pour LDAP (existe bien).
  - **admin/admin@TP.LOCAL** → Principal administrateur.
  - **krbtgt/TP.LOCAL@TP.LOCAL** → Ticket Granting Ticket (TGT) du royaume.
- **Validation** :
  Le SPN nécessaire (**ldap/kdc.tp.local**) est correctement créé dans la base Kerberos.

sudo klist -k /etc/ldap/ldap.keytab :

- Le **keytab** contient bien les clés pour le SPN **ldap/kdc.tp.local**.
- **KVNO 2** : Version des clés (Key Version Number).
- Deux entrées = Deux algorithmes de chiffrement supportés (AES-256 et AES-128).

7. Authentification LDAP via Kerberos (GSSAPI)

```
phone1@kdc:~$ ldapwhoami -Y GSSAPI -H ldap://ldap.tp.local
SASL/GSSAPI authentication started
SASL username: admin/admin@TP.LOCAL
SASL SSF: 256
SASL data security layer installed.
dn:uid=admin/admin,cn=gssapi,cn=auth
```

Cette capture montre une **authentification LDAP réussie via Kerberos (GSSAPI).**

- **Options** :
  - -Y GSSAPI : Utilise le mécanisme d'authentification Kerberos (GSSAPI).
  - -H ldap://ldap.tp.local : Cible le serveur LDAP sur son FQDN.

a. Authentification SASL/GSSAPI :

- **SASL username: admin/admin@TP.LOCAL**
  → L'utilisateur authentifié est admin/admin dans le royaume TP.LOCAL.
  → Preuve que :
    - Le ticket Kerberos est valide (obtenu via kinit).
    - Le serveur LDAP reconnaît ce principal.

b. Sécurité (SSF) :

- **SASL SSF: 256**
  → *Security Strength Factor* = 256 bits (chiffrement AES-256 activé).
  → Garantit la confidentialité/intégrité des échanges.

c. Identité LDAP Retournée :

- **dn:uid=admin/admin,cn=gssapi,cn=auth**
  → Format standard pour les authentifications SASL dans LDAP :
    - cn=auth : Contexte d'authentification.
    - cn=gssapi : Mécanisme utilisé (GSSAPI/Kerberos).
    - uid=admin/admin : Correspond au principal Kerberos.

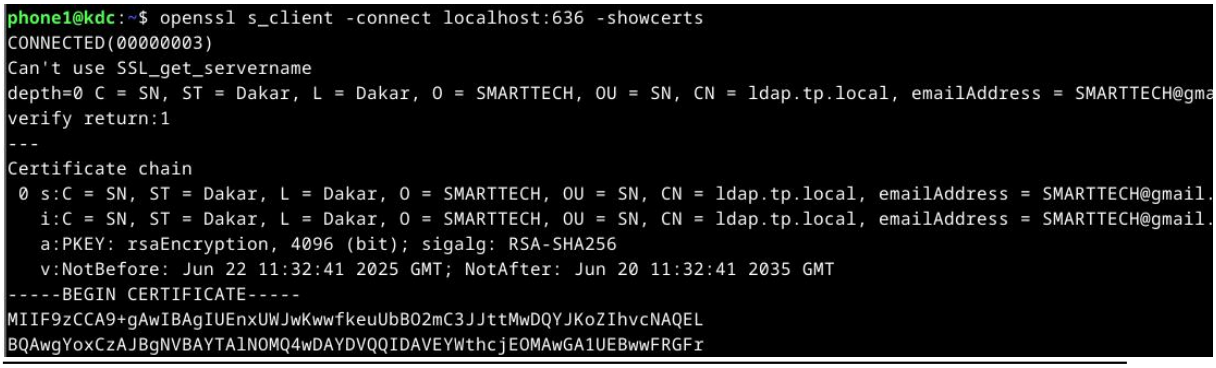
Cette sortie confirme que :
☑ **LDAP et Kerberos sont intégrés avec succès**.
☑ **L'authentification GSSAPI fonctionne** (avec chiffrement AES-256).
☑ **Les SPN et keytab sont correctement configurés**.

```
phone1@kdc:~$ ldapsearch -Y GSSAPI -H ldap://ldap.tp.local -b "dc=tp,dc=local" "(objectClass=*)"
SASL/GSSAPI authentication started
SASL username: admin/admin@TP.LOCAL
SASL SSF: 256
SASL data security layer installed.
# extended LDIF
#
# LDAPv3
# base <dc=tp,dc=local> with scope subtree
# filter: (objectClass=*)
# requesting: ALL
#

# tp.local
dn: dc=tp,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: tp.local
dc: tp
```

Cette capture montre le résultat d'une commande **ldapsearch** authentifiée via Kerberos (GSSAPI).

c. Verication TLS



Cette capture montre la vérification du certificat TLS utilisé par le serveur LDAP sur le port 636.

Conclusion du TP :

Ce TP nous a permis de mettre en œuvre une solution complète et sécurisée d'authentification centralisée en combinant les services **OpenLDAP**, **Kerberos** et **LDAPS/TLS**. À travers différentes étapes, nous avons configuré un annuaire LDAP structuré, mis en place une authentification forte avec Kerberos via GSSAPI, et sécurisé les communications à l'aide de TLS.

Les différents tests réalisés ont confirmé le bon fonctionnement de chaque composant :

- accès à l'annuaire LDAP,
- authentification Kerberos réussie avec tickets valides,
- chiffrement actif des échanges.