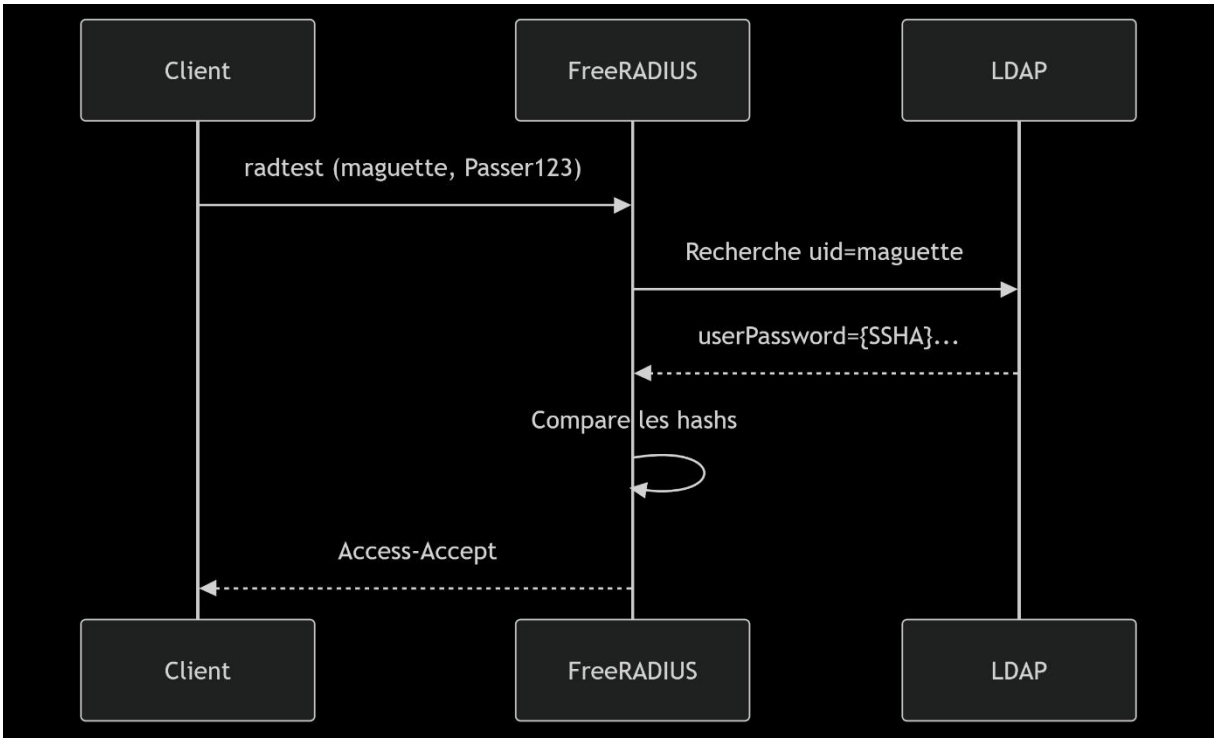
Installation et Configuration de Freeradius avec LDAP-Kerberos

Introduction

Dans le cadre de ce TP de réseau, nous avons mis en place une solution d'**authentification réseau sécurisée** basée sur **FreeRADIUS**, couplée à un annuaire **LDAP** et un service d'authentification **Kerberos**. L'objectif est d'offrir un mécanisme d'authentification centralisé et fort pour les accès réseau, notamment pour des services comme le Wi-Fi d'entreprise ou les VPN.

FreeRADIUS est un serveur RADIUS libre et puissant, souvent utilisé pour gérer les accès réseau. En l'intégrant à **LDAP**, il peut consulter une base d'utilisateurs centralisée, et grâce à **Kerberos**, il bénéficie d'une authentification forte et sécurisée via tickets.

Schéma Expliquatif :



Objectif

L'objectif de ce TP est de configurer un système d'**authentification réseau sécurisé** en utilisant **FreeRADIUS**, en le liant à un annuaire **LDAP** pour la gestion centralisée des utilisateurs, et à **Kerberos** pour une authentification forte.

Ce système permet de :

- Centraliser l'authentification des utilisateurs à partir d'un annuaire LDAP.
- Garantir la sécurité des échanges grâce à l'usage de Kerberos (GSSAPI).
- Mettre en place un contrôle d'accès fiable pour des services tels que le Wi-Fi ou les connexions VPN.

---Problematique (a faire)

1. Création du script :

```
┌─[practice@parrot]─[~]
└──╼ $sudo nano freeradius.sh
```

Nous créons le fichier **freeradius.sh** avec **nano**.

- **Étape 0 : Déclarations des Variables**

```
#!/bin/bash
set -e

# ----------------------------
# VARIABLES À ADAPTER
# ----------------------------
LDAP_HOSTNAME="ldap.tp.local"                    # Nom ou IP résoluble du>
LDAP_IP="192.168.200.50"                         # Pour /etc/hosts si pas>
LDAP_BIND_DN="uid=freeradius_admin,ou=freeradius_users,dc=tp,dc=local"
LDAP_BIND_PW="Passer123"                         # Mot de passe du bind DN
BASE_DN="dc=tp,dc=local"                         # Suffixe LDAP
CLIENT_NAME="borne-wifi"
CLIENT_IP="192.168.200.10"
CLIENT_SECRET="secretpartage"
CERT_FILE="/etc/freeradius/3.0/certs/ldap-ca.pem"      # Chemin pour stocker le>
LDAP_CONF="/etc/freeradius/3.0/mods-available/ldap"
CLIENTS_CONF="/etc/freeradius/3.0/clients.conf"
SITE_DEFAULT="/etc/freeradius/3.0/sites-enabled/default"
SITE_INNERTUNNEL="/etc/freeradius/3.0/sites-enabled/inner-tunnel"
```

Cette section définit les **paramètres de configuration** utilisés tout au long du script. Cela rend le script **plus lisible** et **facilement modifiable**.
Variables Déclarées :

→ Adresse du serveur LDAP (nom DNS et IP).

→ Identifiants utilisés par FreeRADIUS pour se connecter à l'annuaire LDAP.

→ Base de recherche dans l'annuaire LDAP.

→ Emplacement où sera stocké le certificat du serveur LDAP.

→ Chemins vers les **fichiers de configuration de FreeRADIUS** à modifier.

◇ Étape 1 : Installation des paquets nécessaires

```
# ----------------------------
# 1. Installation des paquets nécessaires
# ----------------------------
echo "[*] Installation de FreeRADIUS et utilitaires LDAP..."
apt update -y
apt install -y freeradius freeradius-ldap ldap-utils openssl curl
```

Cette étape installe les composants nécessaires :

- **FreeRADIUS** : le serveur RADIUS.
- **freeradius-ldap** : le module pour interagir avec LDAP.
- **ldap-utils** : outils de test LDAP (ldapsearch, etc.).
- **openssl, curl** : pour sécuriser les connexions et récupérer le certificat.

◇ Étape 2 : Récupération du certificat LDAP via STARTTLS

```
# ----------------------------
# 2. Récupération du certificat via STARTTLS
# ----------------------------
echo "[*] Récupération du certificat LDAP via STARTTLS sur $LDAP_HOSTNAME:389..."
mkdir -p "$(dirname "$CERT_FILE")"
if echo | openssl s_client -connect "${LDAP_HOSTNAME}:389" -starttls ldap -showcerts 2>/dev/null | \
    awk '/BEGIN CERTIFICATE/,/END CERTIFICATE/' > "$CERT_FILE"; then
    if [[ -s "$CERT_FILE" ]]; then
        chmod 644 "$CERT_FILE"
        echo "✔ Certificat extrait dans $CERT_FILE"
        echo "   Empreinte SHA256 :"
        openssl x509 -in "$CERT_FILE" -noout -fingerprint -sha256
    else
        echo "✖ Aucun certificat récupéré (fichier vide). Vérifiez la prise en charge STARTTLS sur le serveur."
        exit 1
    fi
else
    echo "✖ Échec de récupération du certificat via STARTTLS. Vérifiez la connectivité et la configuration du serveur LDAP."
    exit 1
fi
```

Cette commande utilise **openssl** pour établir une connexion sécurisée **STARTTLS** sur le port LDAP (389) et récupérer le certificat du serveur.
Le certificat est ensuite stocké dans un fichier (**ldap-ca.pem**) et sa **valeur d'empreinte SHA256** est affichée pour vérification.
Cela permet à **FreeRADIUS** de faire confiance au serveur LDAP (authentification TLS).

◇ Étape 3 : Configuration du module LDAP

```
# ----------------------------
# 3. Configuration du module LDAP (mods-available/ldap)
# ----------------------------
echo "[*] Écriture de la configuration LDAP dans $LDAP_CONF..."
cat > "$LDAP_CONF" <<EOF
ldap {
    server = "$LDAP_HOSTNAME"
    port = 389
    identity = "$LDAP_BIND_DN"
    password = "$LDAP_BIND_PW"
    base_dn = "$BASE_DN"

    user {
        base_dn = "$BASE_DN"
        filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})"
        scope = "sub"
        access_positive = yes
    }

    authenticate {
        auth_type = bind
    }
```

```
    tls {
        start_tls = yes
        ca_file = "$CERT_FILE"
        require_cert = "demand"
    }

    timeout = 5
    timelimit = 3
    net_timeout = 3
}
EOF
echo "✔ Module LDAP écrit dans $LDAP_CONF (STARTTLS, require_cert=\"demand\")."

# ----------------------------
```

On écrit la configuration complète du module **LDAP de FreeRADIUS**.
Les paramètres définissent :

- Le **serveur LDAP**, son port, le DN de liaison (**bind DN**), le mot de passe, et le **base_dn**.
- Le **filtre de recherche utilisateur**.
- Le **mode d'authentification** : **bind**.
- L'utilisation de **STARTTLS**, avec obligation d'un certificat (**require_cert = "demand"**).

◇ Étape 4 : Activation du module LDAP

```
# ----------------------------
# 4. Activation du module LDAP
# ----------------------------
echo "[*] Activation du module LDAP..."
ln -sf ../mods-available/ldap /etc/freeradius/3.0/mods-enabled/ldap
echo "✔ Lien symbolique créé pour ldap"
```

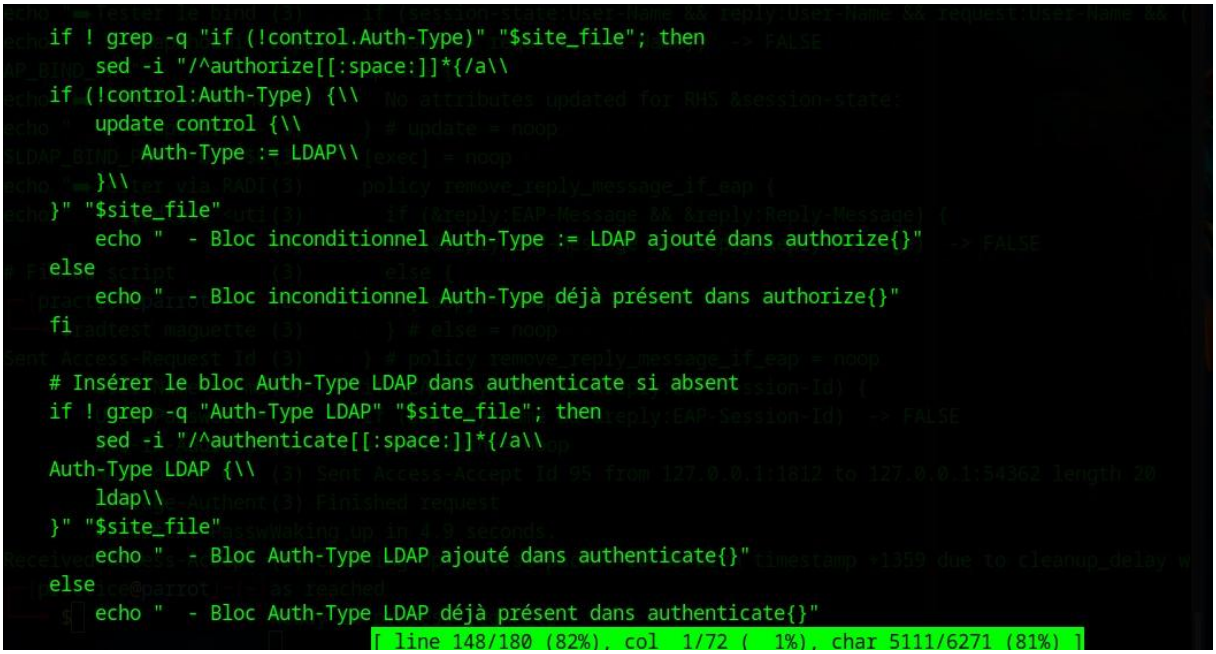
On **active le module LDAP** en créant un lien symbolique vers la configuration préparée dans **mods-available**.

◇ Étape 5 : Mise à jour des fichiers sites-enabled pour intégrer LDAP

```
# ----------------------------
# 5. Mise à jour des sites-enabled
# ----------------------------
echo "[*] Mise à jour de sites-enabled pour inclure LDAP..."

insert_unlang_blocks() {
    local site_file="$1"
    if [[ ! -f "$site_file" ]]; then
        echo "ℹ Fichier $site_file non trouvé, on skip."
        return
    fi
    echo "[*] Traitement de $site_file..."

    # Insérer le module ldap dans authorize s'il n'existe pas
    if ! grep -q "^[[:space:]]*ldap" "$site_file"; then
        sed -i "/^authorize[[:space:]]*{/a\    ldap" "$site_file"
        echo "  - 'ldap' ajouté dans authorize{}"
    else
        echo "  - 'ldap' déjà présent dans authorize{}"
    fi
```



```
if ! grep -q "if (!control.Auth-Type)" "$site_file"; then
    sed -i "/^authorize[[:space:]]*{/a\\
if (!control:Auth-Type) {\\
    update control {\\
        Auth-Type := LDAP\\
    }\\
}" "$site_file"
    echo "  - Bloc inconditionnel Auth-Type := LDAP ajouté dans authorize{}"
else
    echo "  - Bloc inconditionnel Auth-Type déjà présent dans authorize{}"
fi

# Insérer le bloc Auth-Type LDAP dans authenticate si absent
if ! grep -q "Auth-Type LDAP" "$site_file"; then
    sed -i "/^authenticate[[:space:]]*{/a\\
Auth-Type LDAP {\\
    ldap\\
}" "$site_file"
    echo "  - Bloc Auth-Type LDAP ajouté dans authenticate{}"
else
    echo "  - Bloc Auth-Type LDAP déjà présent dans authenticate{}"
```


```
insert_unlang_blocks "$SITE_DEFAULT"
insert_unlang_blocks "$SITE_INNERTUNNEL"
```

Dans cette étape, le script modifie automatiquement les fichiers de configuration **default** et **inner-tunnel** de FreeRADIUS afin d'activer l'utilisation du module LDAP.

Actions réalisées :

1. **Ajout du module ldap dans le bloc authorize {}**

   → Permet à **FreeRADIUS** d'interroger **LDAP** pour vérifier si l'utilisateur existe.

2. **Ajout d'un bloc Auth-Type := LDAP si aucun type d'authentification n'est encore défini**

   → Force **FreeRADIUS** à utiliser **LDAP** comme méthode d'authentification par défaut.

3. **Ajout du bloc Auth-Type LDAP dans authenticate {}**

   → Spécifie que si **Auth-Type** est défini à **LDAP**, alors **FreeRADIUS** doit appeler le module **LDAP** pour effectuer l'authentification.

4. **Application aux deux fichiers** :

- **/etc/freeradius/3.0/sites-enabled/default**
- **/etc/freeradius/3.0/sites-enabled/inner-tunnel**

◇ Étape 8 : Redémarrage et activation du service FreeRADIUS

```
# ----------------------------
# 6. Redémarrage et activation du service
# ----------------------------
echo "[*] Redémarrage du service FreeRADIUS..."
if systemctl restart freeradius; then
    echo "✔ FreeRADIUS redémarré"
    systemctl enable freeradius
else
    echo "✖ Échec du redémarrage. Lancez 'freeradius -X' pour debug."
    exit 1
fi
```

On **redémarre le service FreeRADIUS** pour prendre en compte les modifications, et on l'active pour qu'il démarre automatiquement au démarrage du système.

2. Exécution du Script :

```
┌─[practice@parrot]─[~]
└──╼ $sudo chmod +x freeradius.sh
```

On rend le fichier Executable.

```
┌─[practice@parrot]─[~]
└──╼ $sudo ./freeradius.sh
```

Puis Nous l'executons.

3. Création de l'organisation freeradius_users :

Au lieu de donner à **FreeRADIUS** l'accès direct à l'administrateur principal de LDAP, nous allons créer une **unité d'organisation dédiée**, appelée **freeradius_users**, dans laquelle seront regroupés **tous les utilisateurs authentifiés via FreeRADIUS**.
Un **compte administrateur spécifique**, nommé **freeradius_admin**, sera également créé au sein de cette organisation. Ce compte disposera des **droits nécessaires pour interroger l'annuaire LDAP** (bind et recherche), mais avec **des privilèges limités**, inférieurs à ceux de l'administrateur principal de LDAP.

Cette approche permet de **renforcer la sécurité**, en appliquant le principe du moindre privilège, et de **mieux structurer** l'annuaire LDAP en séparant clairement les utilisateurs RADIUS des autres comptes.

1. Création d'une Unité Organisationnelle (OU)

```
phone1@kdc:~$ cat <<EOF | ldapadd -x -D "cn=admin,dc=tp,dc=local" -w "Passer123"

dn: ou=freeradius_users,dc=tp,dc=local
objectClass: organizationalUnit
ou: freeradius_users
EOF
adding new entry "ou=freeradius_users,dc=tp,dc=local"
```

2. Ajout d'un Utilisateur FreeRADIUS

```
phone1@kdc:~$ cat <<EOF | ldapadd -x -D "cn=admin,dc=tp,dc=local" -w "Passer123"

dn: uid=freeradius_admin,ou=freeradius_users,dc=tp,dc=local
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
cn: FreeRADIUS Admin
sn: Admin
uid: freeradius_admin
userPassword: Passer123
EOF
adding new entry "uid=freeradius_admin,ou=freeradius_users,dc=tp,dc=local"
```

Cette commande ajoute l'utilisateur **freeradius_admin** dans l'OU **freeradius_users**.

```
phone1@kdc:~$ ldapmodify -x -D "cn=admin,cn=config" -w "Passer123" -H ldap://loc
alhost <<EOF
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcAccess
olcAccess: to dn.subtree="dc=tp,dc=local"
  by dn.exact="uid=freeradius_admin,ou=freeradius_users,dc=tp,dc=local" read
  by * break
EOF
modifying entry "olcDatabase={1}mdb,cn=config"
```

Cette commande modifie la configuration d'un serveur LDAP (OpenLDAP) pour accorder des droits de lecture a **freeradius_admin**.

```
phone1@kdc:~$ sudo slappasswd
[sudo] password for phone1:
New password:
Re-enter new password:
{SSHA}dijMJs2uRBJFY5rmxRW17zAnyOqaz9/f
```

Cette commande utilise l'outil **slappasswd** pour générer un **mot de passe chiffré** compatible avec **OpenLDAP**. Cela va nous permettre de chiffré notre mot de pass.

```
phone1@kdc:~$ sudo nano maguette.ldif
```

Le fichier **maguette.ldif** contient la définition d'un nouvel utilisateur :

```
dn: uid=maguette,ou=freeradius_users,dc=tp,dc=local
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
cn: Maguette
sn: Leye
uid: maguette
uidNumber: 10020
gidNumber: 100
homeDirectory: /home/maguette
loginShell: /bin/bash
userPassword: {SSHA}vPT3FmBE5SdihIlCAysVYhHYsS/m0BXH
```

- L'utilisateur est placé dans l'OU **freeradius_users**.
- Le mot de passe est stocké de manière sécurisée (hash SSHA).
- Les classes **posixAccount/shadowAccount** permettent une intégration avec Linux.

```
phone1@kdc:~$ ldapadd -x -D "cn=admin,dc=tp,dc=local"         -w Passer123
   -H ldap://ldap.tp.local         -f maguette.ldif
adding new entry "uid=maguette,ou=freeradius_users,dc=tp,dc=local"
```

Avec cette commande nous ajoutons l'utilisateur maguette.

Et on fait de meme pour les autres utilisateurs freeradius

4. Test et logs :

1. Démarrage de FreeRADIUS en Mode Debug

```
[practice@parrot]-[~]
$sudo freeradius -X
```

- **-x** : Active le mode debug (logs détaillés).
- **But** : Vérifier que le serveur :
  - Se connecte correctement à LDAP.
  - Écoute sur les ports RADIUS standard (`1812` pour l'authentification, `1813` pour la comptabilité).

```
rlm_ldap (ldap): Opening additional connection (4), 1 of 6 pending slots used
rlm_ldap (ldap): Connecting to ldap://ldap.tp.local:389
rlm_ldap (ldap): Waiting for bind result...
rlm_ldap (ldap): Bind successful
```

On voit qu'il a pu contacter notre serveur **ldap.**

```
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 41990
Listening on proxy address :: port 32812
Ready to process requests
```

→ FreeRADIUS est opérationnel et attend des requêtes.

2. Test d'Authentification avec **radtest :**

```
[practice@parrot]-[~]
$radtest maguette Passer123  localhost 0 testing123
Sent Access-Request Id 145 from 0.0.0.0:35966 to 127.0.0.1:1812 length 78
        User-Name = "maguette"
        User-Password = "Passer123"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "Passer123"
Received Access-Accept Id 145 from 127.0.0.1:1812 to 127.0.0.1:35966 length 20
```

- **Paramètres** :
  - **Utilisateur** : maguette (défini dans LDAP).
  - **Mot de passe** : Passer123 (hash SSHA dans LDAP).
  - **Secret RADIUS** : testing123 (clé partagée configurée dans /etc/freeradius/3.0/clients.conf).

```
┌─[practice@parrot]─[~]
└──╼ $radtest aziz  Passer123  localhost 0 testing123
Sent Access-Request Id 75 from 0.0.0.0:58207 to 127.0.0.1:1812 length 74
        User-Name = "aziz"
        User-Password = "Passer123"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "Passer123"
Received Access-Accept Id 75 from 127.0.0.1:1812 to 127.0.0.1:58207 length 20
```

De même pour aziz

<u>Dans les Logs</u>

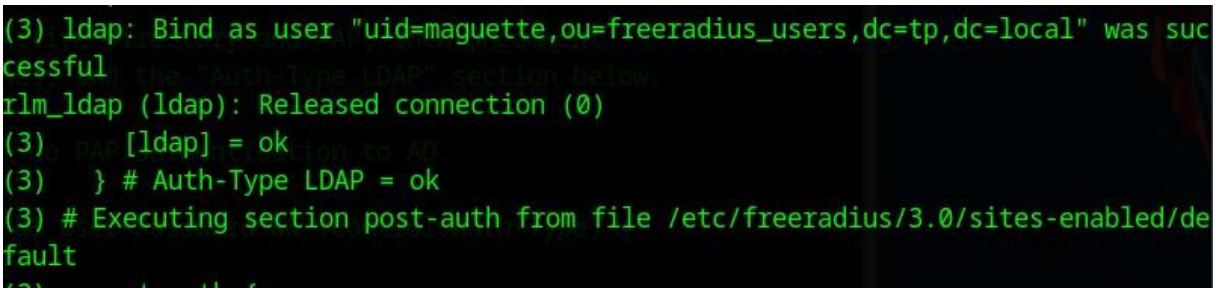
```
Ready to process requests
(1) Received Access-Request Id 75 from 127.0.0.1:58207 to 127.0.0.1:1812 length 74
(1)   User-Name = "aziz"
(1)   User-Password = "Passer123"
(1)   NAS-IP-Address = 127.0.1.1
(1)   NAS-Port = 0
(1)   Message-Authenticator = 0xc516ec0e9464f6139a00a01c2fe9cd39
(1) # Executing section authorize from file /etc/freeradius/3.0/sites-enabled/default
```

- **Étapes** :
    - FreeRADIUS reçoit une demande d'accès pour l'utilisateur **aziz**.
    - Le mot de passe fourni est **"Passer123"** (en clair dans la requête, mais chiffré via le *Message-Authenticator*).
    - Adresse NAS : 127.0.1.1 (client local en test).

```
(1) ldap: Bind as user "uid=Aziz,ou=freeradius_users,dc=tp,dc=local" was successful
rlm_ldap (ldap): Released connection (0)
```

- **Processus** :

1. FreeRADIUS interroge LDAP pour trouver l'entrée de l'utilisateur **aziz**.
2. Il se connecte avec le DN exact : uid=Aziz,ou=freeradius_users,dc=tp,dc=local.
3. **Bind successful** → La connexion LDAP est établie.



De même Pour maguette.

Conclusion :

Ce TP nous a permis de mettre en place une authentification réseau centralisée en utilisant FreeRADIUS, LDAP et Kerberos. Nous avons installé et configuré FreeRADIUS pour qu'il interroge un annuaire LDAP sécurisé. Une organisation spécifique a été créée dans LDAP pour les utilisateurs RADIUS, avec un compte administrateur dédié aux droits limités.

La sécurité a été renforcée grâce à l'utilisation de mots de passe hashés (SSHA), du protocole STARTTLS, et de l'authentification forte avec Kerberos via GSSAPI. Les tests réalisés avec radtest ont confirmé que l'authentification fonctionne correctement pour les utilisateurs LDAP.