mewo
CAMPUS MÉTIERS

RAPPORT

Serveur LDAP/ client

Margaux TANET
2024-2025

SOMMAIRE

I- Installation et configuration OpenLDAP 4-8

1. Importation des schémas 4
2. Création d'un mot de passe admin LDAP : SSHA 5
3. Fichier hosts 5
4. Ajout des données dans l'annuaire 6
5. Recherche dans l'annuaire 8

II- Installation et configuration des packages de LDAP Client 8-13

1. Authentification du client 8
   a. Fichier nsswitch.conf 8
   b. Fichier sssd.conf 8
   c. PAM 10
   d. Gestion de la politique des mots de passes avec PAM 11
2. Fichier ldap.conf 12
3. Oddjobd.mkhomedir 12
4. Tests de connexions d'un client pour joindre le domaine 13

III- Réaliser des sauvegardes personnalisées de l'annuaire 14

1. Création d'une sauvegarde complète 14
2. Création d'une sauvegarde incrémentielle 14

IV- Optimiser l'exploitation de l'annuaire à l'aide d'index 15

V- Scripts Perl pour interroger l'annuaire LDAP 16

VI – Audit sous Linux 18-19

a) Définition d'un audit 18
b) Installation Lynis 19

VII – Conclusion 24

Objectifs et intérêts :

Ce rapport vise à décrire la mise en place d'un serveur OpenLDAP sous Red Hat et ainsi l'intégration d'un client dans le domaine. L'objectif principal ici est de créer un annuaire centralisé permettant de gérer efficacement des utilisateurs et leurs accès le tout de manière sécurisée.

Introduction

OpenLDAP est une implémentation open-source du protocole LDAP (Lightweight Directory Access Protocol), qui permet de gérer des informations sur les utilisateurs, les groupes et d'autres ressources au sein d'un annuaire. Il est couramment utilisé pour centraliser l'authentification et la gestion des accès dans les entreprises.

RedHat est une société américaine éditant des distributions GNU/Linux. Cette solution permet d'accéder à des plateformes de systèmes d'exploitation, de stockage, de gestion, etc...

Pourquoi utiliser Redhat avec OpenLDAP ?

Redhat permet un fonctionnement fiable avec OpenLDAP et propose un support technique et des services de formation complets. Il y a des mises à jour régulières de sécurités et des correctifs. Il comprend des outils comme OpenShift, Ansible ou d'autres solutions encore qui permettent l'intégration avec OpenLDAP et d'autres applications.

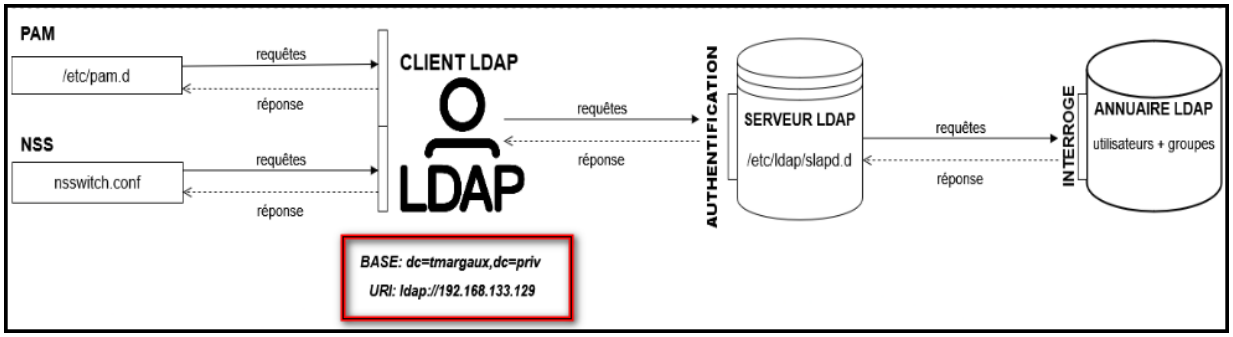


Schéma entre le serveur LDAP et le client

I- Installation et configuration OpenLDAP

Installation faite sous Linux avec Redhat

CONFIGURATION DES OPTIONS PAR DEFAUT:

BASE DN: dc=tmargaux,dc=priv

LDAP Serveur IP: 192.168.133.129

1. Importation des schémas

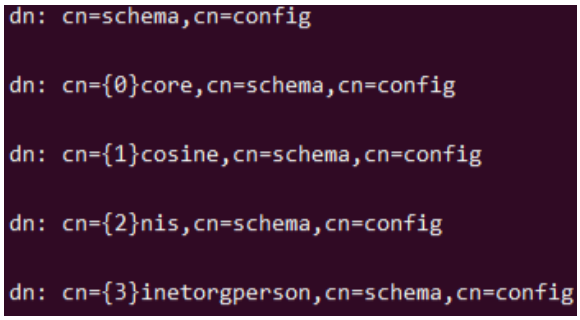
Un **schéma LDAP** est un ensemble de règles qui permettent de définir ce qui doit être enregistré en tant qu'entrée dans un annuaire LDAP. Chaque annuaire LDAP possède un schéma par défaut qui ensuite, peuvent être modifiés et personnalisés. Ce que contient un schéma sont les attributs, des classes d'objets, ...

Le serveur LDAP permet alors de mettre en place le schéma dans le but de vérifier que les modifications faites à l'annuaire soient conformes.

Les attributs sont des éléments contenus dans les entrées d'un annuaire.
Exemple : *cn, userPassword*
Cn étant le nom de l'objet.
Les schémas par défaut s'importent grâce à la commande **ldapadd**:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/core.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Une fois chargé cela doit renvoyer les noms distingués (les DN) de tous les schémas.



Dn correspond à un identifiant unique pour chaque objet (utilisateur, groupe, ...) dans LDAP. Exemple : *dn : cn=Margaux, ou=utilisateurs, dc= tmargaux, dc=priv*

Cn correspond à un attribut LDAP. Il représente une personne ou une entité dans l'annuaire LDAP. Exemple : *cn=Margaux Tanet*

2. Création d'un mot de passe admin LDAP : SSHA

LDAP stocke des mots de passe dans un format haché. Grâce à la commande *slappasswd* cela permet de générer un mot de passe administrateur sous forme haché (SSHA).

Pourquoi générer un mot de passe SSHA ?

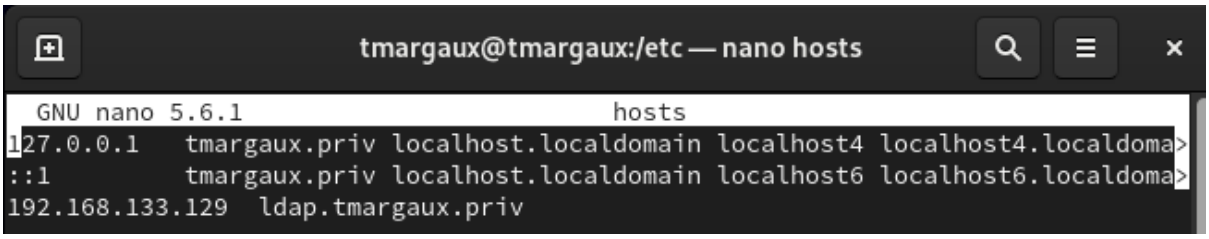
Dans un SSHA (Salted Secure Hash Algorithm), il existe un sel qui est une donnée aléatoire. Cette donnée est ajoutée au mot de passe avant que celui-ci ne soit haché. Imaginons deux utilisateurs ayant le même mot de passe, la version hachée sera différente grâce à ce sel. Sans le sel, un attaquant pourrait éventuellement retrouver un mot de passe en comparant des hachages à une base de données précalculée ou encore d'une attaque par force brute.

```
{SSHA}5d2e19393cc5ef67a5bc28e0f5a7d992ed36a446
```

Exemple d'un mot de passe haché en SSHA avec comme préfixe {SSHA} pour indiquer l'algorithme utilisé.

3. Fichier hosts

Le fichier hosts se trouve dans le répertoire /etc/hosts. Il permet de faire correspondre des adresses IP avec des noms de domaines sans passer par le serveur DNS. Très pratique pour éviter de retaper l'adresse IP dans certains fichiers de configuration ou des commandes.



Fichier hosts

- *127.0.0.1 tmargaux.priv: associe l'adrese IP du localhost à mon nom d'hote (tmargaux.priv)*
- *192.168.133.129 tmargaux localhost.localdomain: tmargaux et localhost.localdomain sont deux noms d'hôtes pour l'adresse IP 192.168.133.129.*
- *::1 tmargaux.priv: résolution de l'adresse ::1 vers tmargaux.priv en utilisant le protocole IPv6. C'st l'équivalent de l'adresse 127.0.0.1 en IPv4. Cela veut dire que quand on envoit une requête à l'adresse "::1" elle est renvoyée vers la machne locale.*

4. Ajout de données dans l'annuaire

Un annuaire est une base de données contenant différents types d'objets (nom, prénom, etc) normés par des schémas. Des applications interrogent l'annuaire et ressortent les valeurs saisies. Le protocole LDAP (Lightweight Directory Access Protocol) permet donc d'interagir avec les objets stockés dans l'annuaire.

Chaque objet est identifié par un attribut "DN" (Distinguished Name) qui permet de renvoyer son positionnement dans l'annuaire. Pour créer et stocker des utilisateurs, il faut tout d'abord créer des OU (Organizational Unit).
Sous LDAP, ajouter des données à un annuaire peut être inscrit dans un fichier "LDIF". Il permet de décrire des changements à appliquer à un annuaire LDAP. La commande **ldapadd** utilise ce fichier pour insérer les données dans l'annuaire.

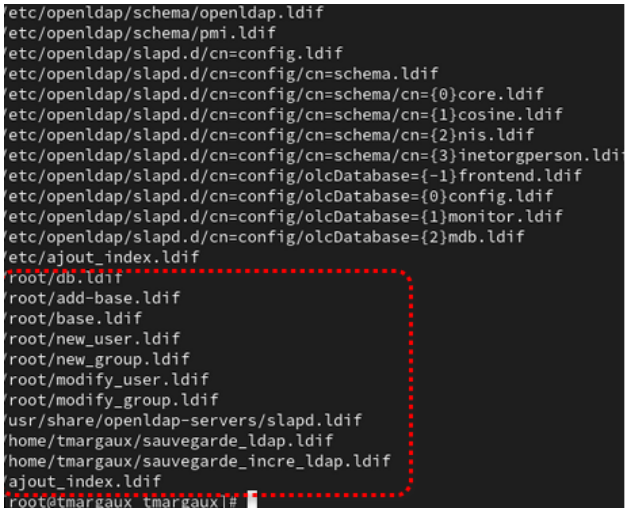
Exemple d'ajout d'un fichier contenant un utilisateur à ajouter dans le répertoire :

***Ldapadd** –x –D "cn=admin, dc=tmargux, dc=priv" -W –f new_user.ldif*

- *-x utilise authentification simple*
- *-D: spécifie le nom distingué (DN) de l'administrateur autorisé à effectuer l'ajout*
- *-W: mot de passe pour l'utilisateur DN*
- *-f + .ldif : fichier ldif contenant les données à ajouter*

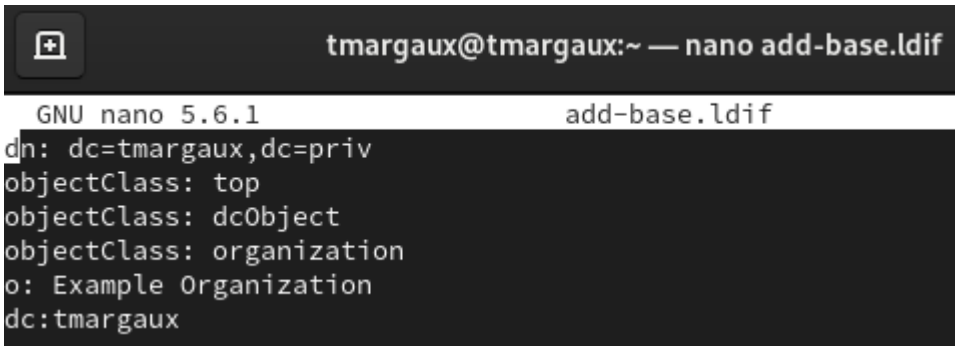
Pourquoi utiliser des fichiers LDIF ?

- Utilisation simple lors de l'export/import des données des annuaires.
- Simplifie la création, la modification et la suppression d'utilisateurs ou d'objet dans l'annuaire.
- Les outils LDAP comme **ldapmodify**, **ldapsearch**, etc sont conçus pour travailler avec des fichiers LDIF.
- Pour un ajout en masse d'utilisateurs ou d'OU, on peut simplement importer le fichier LDIF comportant les données LDAP.



Ensemble des fichiers LDIF crées

- La racine d'un annuaire (fichier LDIF) est la partie sur laquelle les OU et les utilisateurs par exemple vont être ajoutés.



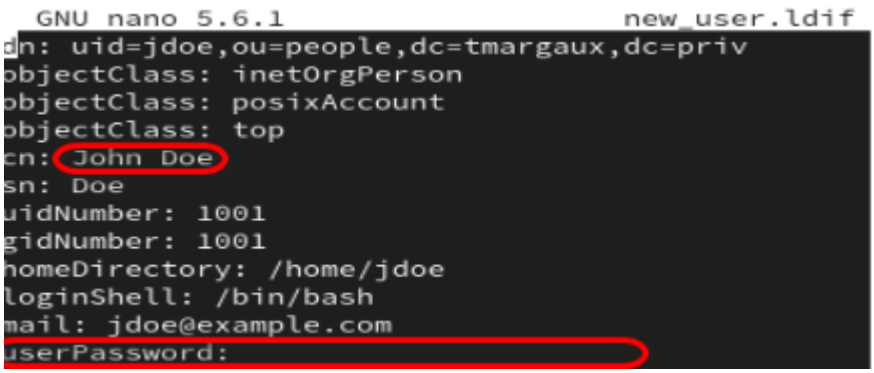
Racine de mon annuaire

La commande pour ajouter la base DN : *sudo ldapadd -x -D "cn'admin,dc-example, dc-com" -W -f add-base.ldif*

Pour vérifier les ajouts d'OU : *ldapsearch -x -b "dc=tmargaux,dc=priv" "(objectClass=organizationalUnit)"*. Voici le fichier ldif pour la création d'OU, ici "People" et "Groups" :

Fichier LDIF regroupant mes différentes OU (People et Groups)

La commande qui permet d'ajouter une entrée lors de la création d'un utilisateur dans l'annuaire : *sudo ldapadd -x -D cn=admin,dc=-margaux,dc= priv-f new-user.ldif*



Fichier LDIF correspondant à la création de mon utilisateur avec le SSHA en mot de passe (caché ici)

5. Recherche dans l'annuaire

La commande **ldapsearch** est utilisée pour interroger le répertoire et récupérer des informations demandées.

Exemple d'une recherche pour trouver tous les utilisateurs ayant un attribut cn contenant le mot "jean" dans le domaine tmargaux.priv:

Ldapsearch –x –b "dc=tmargaux, dc=priv" "(cn=Jean)"*

La commande retournera tous les attributs (cn) "jean" correspondant au filtre.

II- Installation et configuration des packages de LDAP client

1. Authentification du client

a) Le fichier nsswitch.conf

Le programme gérant les bases de comptes utilisables par un système Linux se nomme **NSS** (*Name Switching Service)* et va s'appuyer sur un service "**SSSD**" qui pointe sur le serveur LDAP et présente les comptes qu'il stocke. Le fichier **nsswitch.conf** indique comment le système doit récupérer les informations des utilisateurs, des groupes, etc. Il permet de dire quels services est utilisé pour chaque type d'informations : LDAP, SSSD. Il se situe dans le répertoire */etc/nsswitch.conf*.

```
passwd:     files sss systemd
group:      files sss systemd
netgroup:   sss files
automount:  sss files
services:   sss files
```

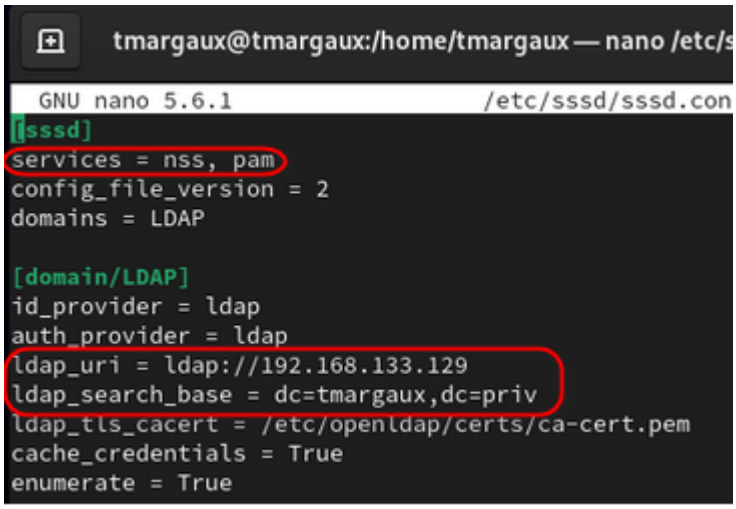
Configuration du fichier NSS > nsswitch.conf

Actions réalisées : *Le programme NSS interroge le service SSSD (sss) pour compléter la base locale (les files).*

b) Fichier sssd.conf

Configuration de LDAP client pour s'authentifier grâce au LDAP serveur

Pour s'authentifier avec le client, nous allons utiliser ce service "SSSD". Il permet d'accéder à un fournisseur d'authentification distants sur le système client. Le fichier se trouve dans le répertoire */etc/sssd/sssd.conf*. Il détermine les domaines d'authentification (LDAP, Kerberos), les serveurs à contacter, ainsi que les paramètres de mise en cache des informations d'utilisateur. Sur le client, il faut configurer le service SSSD avec les informations de notre serveur LDAP.

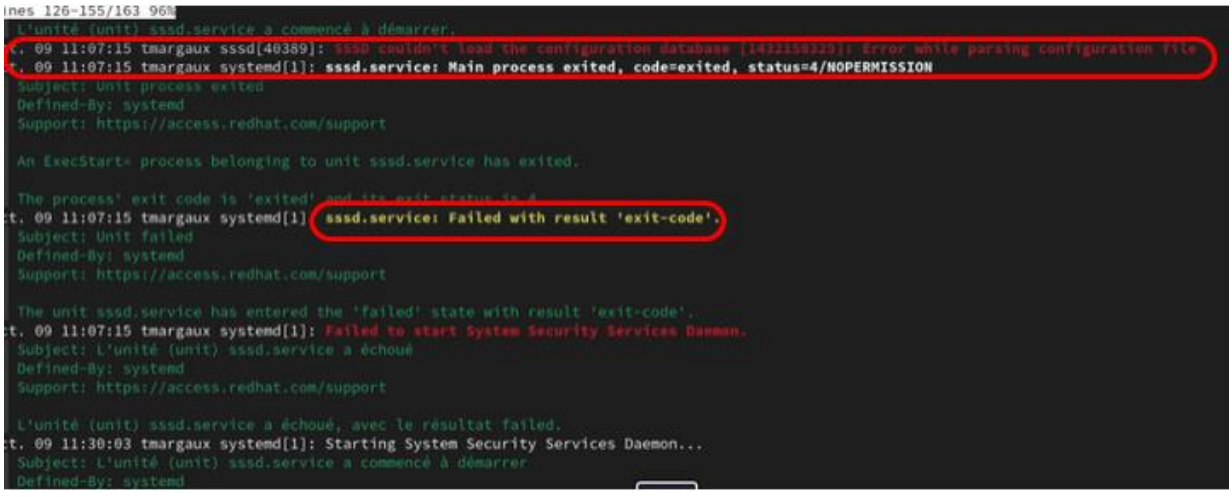


Fichier de configuration SSSD

La configuration définit le serveur LDAP à l'adresse ldap://192.168.133.129 et indique que la base de recherche est dc=tmargaux,dc=priv. Il utilise les module nss et pam. “Services” indique que SSSD sera manipulé pour la gestion des comptes (nss) et l’authentification (pam).

- Configuration SSSD sur le client pour faire la connexion du client à LDAP et se connecter ainsi au client

J’ai rencontré un problème lors de sa configuration : Je suis allée voir dans les logs : impossible de démarrer le service SSSD et un problème dans le fichier de configuration sssd.conf:



J’ai donc procédé à une purge du package SSSD pour réinitialiser et faire une désinstallation propre du logiciel/ fichiers avec ces commandes :

- sudo yum remove sssd sssd-tools
- sudo rm -f /etc/sssd/sssd.conf
- sudo rm -rf /etc/sssd/

J'ai vu qu'il me restait des fichiers non supprimés avec le nom contenant sssd : rpm -qa | grep sssd. J'ai purgé tous ces fichiers.
J'ai réinstallé le package proprement avec ces commandes :

```
sudo yum install authselect nss-pam-ldapd
sudo authselect select sssd with-mkhomedir --force
sudo nano /etc/sssd/sssd.conf
--
[sssd] services = nss, pam config_file_version = 2 domains = LDAP [domain/LDAP]
id_provider = ldap auth_provider = ldap ldap_uri = ldap://192.168.133.129
ldap_search_base = dc=tmargaux,dc=priv ldap_tls_cacert = /etc/openldap/certs/ca-
cert.pem # Si vous utilisez TLS/SSL cache_credentials = True enumerate = True
--
sudo chmod 600 /etc/sssd/sssd.conf
sudo systemctl enable sssd sudo systemctl start sssd
getent passwd jdoe
```

c) PAM (Pluggable Authentication Modules)

PAM gère l'authentification des utilisateurs comme les sessions SSH. Il peut interagir avec le module SSSD et se configure dans le répertoire */etc/pam.d* en ajoutant ces deux lignes :

```
auth    sufficient    pam_sss.so
auth    required      pam_unix.so
```

pam_sss.so : utilise SSSD pour l'authentification
Pam_unix.so: module par défaut pour authentification locale
Pour permettre l'authentification d'utilisateurs LDAP il faut vérifier ou modifier les lignes suivantes dans le répertoire */etc/pam.d pour le fichier system-auth* :

```
  GNU nano 5.6.1                                  system-auth
# Generated by authselect on Thu Sep 26 09:06:31 2024
# Do not modify this file manually.

auth        required                                    pam_env.so
auth        required                                    pam_faildelay.so delay=2000000
auth        sufficient                                  pam_fprintd.so
auth        [default=1 ignore=ignore success=ok]        pam_usertype.so isregular
auth        [default=1 ignore=ignore success=ok]        pam_localuser.so
auth        sufficient                                  pam_unix.so nullok
auth        [default=1 ignore=ignore success=ok]        pam_usertype.so isregular
auth        sufficient                                  pam_sss.so forward_pass
auth        required                                    pam_deny.so
auth        required                                    pam_ldap.so

account     required                                    pam_unix.so
account     sufficient                                  pam_localuser.so
account     sufficient                                  pam_usertype.so issystem
account     [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required                                    pam_permit.so
account     requireed                                   pam_ldap.so

password    requisite                                   pam_pwquality.so local_users_only
password    sufficient                                  pam_unix.so sha512 shadow nullok use_authtok
password    [success=1 default=ignore]                  pam_localuser.so
password    sufficient                                  pam_sss.so use_authtok
password    required                                    pam_deny.so
password    required                                    pam_ldap.so

session     optional                                    pam_keyinit.so revoke
session     required                                    pam_limits.so
-session    optional                                    pam_systemd.so
session     [success=1 default=ignore]                  pam_succeed_if.so service in crond quiet use_uid
session     required                                    pam_unix.so
session     optional                                    pam_sss.so
session     optional                                    pam_ldap.so
```

Fichier system-auth

Voici la signification de ces trois lignes :

	Utilité	*required*	*Effet*
auth	*Permet l'authentification des utilisateurs*	*Authentification obligatoire*	*Si le serveur LDAP est inaccessible ou les informations sont incorrectes, l'accès est refusé.*
account	*Gérer l'autorisation et vérification de comptes*	*Vérification de compte via LDAP obligatoire*	*Si LDAP ne reconnait pas l'utilisateur ou que le compte n'est pas valide, l'accès est refusé.*
Password	*Gestion des mots de passes et de changements*	*Gestion du mot de passe obligatoire*	*Les utilisateurs doivent passer par LDAP pour changer le mot de passe. Si c'est impossible, PAM n'exécute pas les autres modules.*

d) Gestion de la politique des mots de passes avec PAM

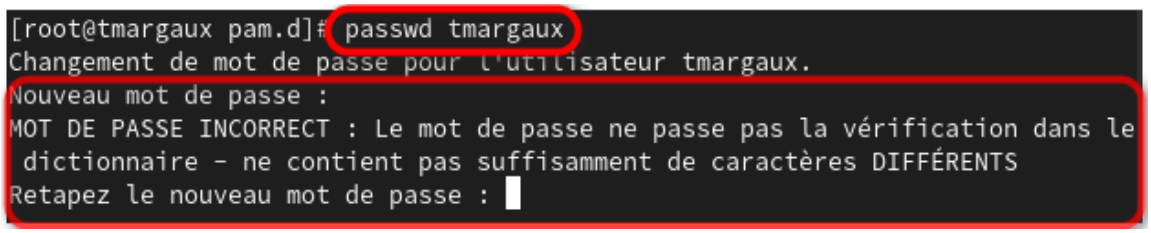
Il est important d'avoir une bonne gestion et une bonne configuration de la politique des mots de passe sur son système. J'utilise le module "**pwquality**" qui permet de gérer tous les paramètres relatifs à la structure du mot de passe.

S'il n'est pas déjà installé, installer le module "**libpam-pwquality**.

Se rendre ensuite dans le répertoire etc/pam.d où se trouvent les fichiers de configuration de PAM. Le fichier **system-auth** gère les règles d'authentifications et tout ce qui est politiques, gestion, session. On ajoute la ligne suivante : *password requisite pam_pwquality.so retry=3 minlen=12*.

Après la modification cela permet d'autoriser 3 essais pour saisie de mot de passe (retry), de détermine la longueur minimale du mot de passe à 12 caractères (minlen). Pour tester si la politique fonctionne, il faut créer/utiliser un utilisateur et définir un mot de passe. PAM analysera le mot de passe saisi.

Je teste avec mon nom : passwrd tmargaux. Je rentre un mot de passe qui ne possède pas 12 caractères pour voir si la règle mise en place est effective.



Résultat lors du changement de mot avec moins de 12 caractères

La règle mise en place fonctionne bien car quand je saisi un nouveau mot de passe et qu'il ne rentre pas dans la politique de mot de passe imposée cela me refuse le nouveau mot de passe.

2. Fichier ldap.conf

Le fichier ldap.conf est un fichier de configuration utilisé par les outils LDAP comme ldapsearch, ldapadd, ... Il contient les paramètres pour que le système se connecte et interagisse avec un serveur LDAP. Toutes les informations inscrites dans ce fichier allègeront les lignes de commande puisqu'il ne sera plus nécessaire de préciser les options positionnées ici. Cela permet d'établir la connexion au serveur ldap et de le faire tourner. Il se trouve dans le répertoire : */etc/openldap/ldap.conf*

Une partie du fichier ldap.conf avec mentionner ici, notre nom de domaine ainsi que notre serveur LDAP avec lequel les applications se connecteront.

3. Oddjob-mkhomedir

Le fichier est lié en quelque sorte à PAM. En effet, il est utilisé pour la création de répertoire personnel. Si le répertoire personnel de l'utilisateur n'est pas créé, oddjob s'en charge.

Pour se faire il faut installer le système authselect:

```
Authselect select ssd with-mkhomedir -force
Le profil "sssd" a été sélectionné.
Les cartes nsswitch sont écrasées par le profil:
- passwd
- groupe
- groupe net
- auto-montée
- services

Assurez-vous que le service SSSD est configuré et activé. Voir la documentation SSSD pour plus d'informa

- avec-mkhomedir est sélectionné, assurez-vous que le module pam'oddjob'mhomedir
  est présent et le service oddjobd est activé
  - systemctl permet impairjobd.service
  - systemctl start impairjobd.service
```

4. Tester la connexion d'un utilisateur en SSH

- Essai de connexion avec l'utilisateur créer sur mon serveur sur mon client :

Commande : su *utilisateur*

```
[root@tmargaux ~]# id jdoe
uid=1001(jdoe) gid=1001(developers) groupes=1001(developers)
[root@tmargaux ~]#
```

Affichage de l'IUD et du GID de l'utilisateur souhaité avec son groupe (ici jdoe/developers)

J'ai installé ssh : yum install zsh

Saisir la commande : su *nom_utilisateur (ici jdoe)*

J'ai pu me connecter en SSH sur mon client :

```
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~).  This function can help you with a few settings that should
make your use of the shell easier.

You can:

(q)  Quit and do nothing.  The function will be run again next time.

(0)  Exit, creating the file ~/.zshrc containing just a comment.
     That will prevent this function being run again.

(1)  Continue to the main menu.

--- Type one of the keys in parentheses ---
```

J'ai procédé à un autre test, le client arrive à joindre le domaine :

```
[root@tmargaux ~]# ping tmargaux.priv
PING tmargaux.priv(tmargaux.priv (::1)) 56 octets de données
64 octets de tmargaux.priv (::1) : icmp_seq=1 ttl=64 temps=0.500 ms
64 octets de tmargaux.priv (::1) : icmp_seq=2 ttl=64 temps=0.354 ms
64 octets de tmargaux.priv (::1) : icmp_seq=3 ttl=64 temps=0.313 ms
64 octets de tmargaux.priv (::1) : icmp_seq=4 ttl=64 temps=1.44 ms
^C
--- statistiques ping tmargaux.priv ---
4 paquets transmis, 4 reçus, 0% packet loss, time 3049ms
rtt min/avg/max/mdev = 0.313/0.652/1.443/0.461 ms
```

Nous avons configuré le client LDAP, mis en place SSSD qui permet de communiquer avec le serveur LDAP. Configurer un client LDAP pour s'authentifier auprès d'un serveur LDAP.

III-Réaliser des sauvegardes personnalisées de l'annuaire :

ldap://192.168.133.129
Dc=tmargaux,dc=priv
Nom de sauvegarde : sauvegarde_ldap.ldif

1. Création d'une sauvegarde complète

On effectue cette commande pour la recherche dans l'annuaire LDAP et on enregistre les résultats dans le fichier sauvegarde_ldap.ldif:
ldapsearch -x -H ldap://192.168.133.129 -b "dc=tmargaux,dc=priv" -D "cn=admin,dc=tmargaux,dc=priv" -W > sauvegarde_ldap.ldif

Pour vérifier ce qu'enregistre la sauvegarde : less *nom_de_la_sauvegarde.ldif*

```
tmargaux@tmargaux:~ — less sauvegarde_ldap.ldif
# extended LDIF
#
# LDAPv3
# base <dc=tmargaux,dc=priv> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# tmargaux.priv
dn: dc=tmargaux,dc=priv
objectClass: top
objectClass: dcObject
objectClass: organization
o: Example Organization
dc: tmargaux

# People, tmargaux.priv
dn: ou=People,dc=tmargaux,dc=priv
objectClass: organizationalUnit
ou: People

# Groups, tmargaux.priv
dn: ou=Groups,dc=tmargaux,dc=priv
objectClass: organizationalUnit
ou: Groups

# jdoe, People, tmargaux.priv
dn: uid=jdoe,ou=People,dc=tmargaux,dc=priv
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: John Doe
```

La sauvegarde enregistre dans un fichier. ldif les OU, les utilisateurs, le domaine ainsi que les entrées d'annuaire

2. Création d'une sauvegarde incrémentielle

On effectue cette commande pour la recherche dans l'annuaire LDAP et on enregistre les résultats dans le fichier backup_config_2024-10-16.ldif:

ldapsearch -x -H ldap://votre_serveur_ldap -b "dc=tmargaux,dc=priv" -s sub "(&(modifyTimestamp>=$LAST_BACKUP_DATE))" -D "cn=admin,dc=tmargaux,dc=priv" -W > backup_config_2024-10-16.ldif

- Mise en place d'une sauvegarde automatique via un script :

Contenu du script :

```
# Variables
LDAP_SERVER="ldap://votre_serveur_ldap"
BASE_DN="dc=exemple,dc=com"
ADMIN_DN="cn=admin,dc=exemple,dc=com"
BACKUP_DIR="/chemin/vers/sauvegarde"
LAST_BACKUP_DATE_FILE="$BACKUP_DIR/last_backup.txt"

# Lire la dernière date de sauvegarde
if [ -f "$LAST_BACKUP_DATE_FILE" ]; then
    LAST_BACKUP_DATE=$(cat "$LAST_BACKUP_DATE_FILE")
else
    echo "Aucune date de sauvegarde précédente trouvée."
    exit 1
fi

# Effectuer la sauvegarde incrémentielle
ldapsearch -x -H "$LDAP_SERVER" -b "$BASE_DN" -s sub "(&(modifyTimestamp>=$LAST_BACKUP_DA
TE))" -D "$ADMIN_DN" -W > "$BACKUP_DIR/sauvegarde_incre_ldap_$(date +%Y%m%d%H%M%S).ldif"

# Mettre à jour la date de sauvegarde
date -u +"%Y%m%d%H%M%S" > "$LAST_BACKUP_DATE_FILE"

echo "Sauvegarde incrémentielle effectuée avec succès."
```

Script pour la sauvegarde automatique

Pour mettre en place le script, on utilise un fichier "crontab". Il permet d'automatiser des tâches répétitives et de les programmer.

Pour une sauvegarde toutes les trois heures => 0 */3 * * * /chemin/vers/sauvegarde/sauvegarde_incre_ldap.sh

- */3 : Indique que le script s'exécutera toutes les 3 heures.
- * * * : Indique que cela s'applique à tous les jours, mois et jours de la semaine

En se rendant dans le répertoire /var/sauv créer ultérieurement, la sauvegarde a bien été faite car le fichier contenant la sauvegarde est présent ("backup_config_2024-10-16. ldif") :

```
[root@tmargaux ~]# cd /var/sauv
[root@tmargaux sauv]# ls
backup_config_2024-10-16.ldif
```

IV-Optimiser l'exploitation de l'annuaire à l'aide d'index

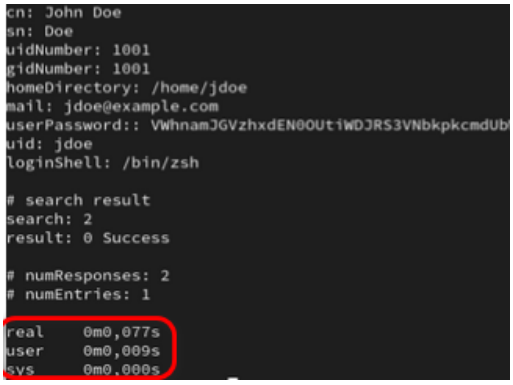
Un index LDAP permet d'améliorer la recherche d'entrées dans un annuaire. L'annuaire peut accéder directement à l'emplacement des données plutôt que de parcourir chaque entrée une par une. Les réponses sont alors plus rapides. On modifie cela dans le fichier slapd.conf dans le répertoire */etc/openldap/slapd.conf*:

```
  GNU nano 5.6.1                          slapd.conf
# Indexer les attributs uid, cn et mail pour des recherches rapides
index uid,cn,mail eq
```

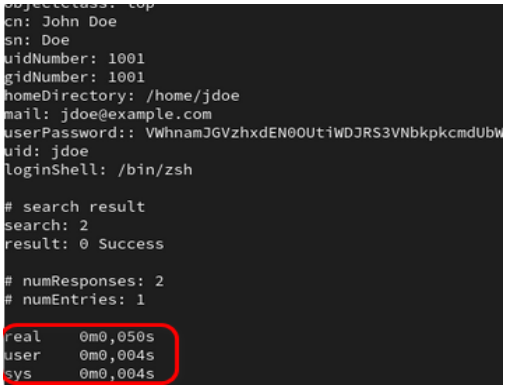
Fichier slapd.conf pour déterminer les attributs LDAP sur lesquels les index seront créés

Iud: identifant utilisateur (unique)
Cn: nom commun (eq : nom complet pour une personne)
Eq : type d'indexation, ici pour des recherches d'égalités/exactes

Temps de réponses pour les deux cas, avec et sans indexation :



Temps de réponse des requêtes Sans indexation 0.077s



Temps de réponse des requêtes avec indexation 0.050s

Le temps de réponse est plus court avec la création de l'index.

V- Ecrire des scripts Perl simples pour l'interrogation de l'annuaire LDAP.

PERL est un langage de programmation souvent utilisé pour le traitement de texte, l'administration système, développement web, et les scripts d'automatisation.
J'ai créé un script PERL pour interroger mon annuaire LDAP. Voici le script PERL :

```
 GNU nano 5.6.1                          ldap_perl.pl
#!/usr/bin/perl
use strict;
use warnings;
use Net::LDAP;

# Configuration de la connexion
my $ldap_server = 'ldap://192.168.133.129';  # Remplacez par l'URL de votre serveur LDAP
my $base_dn     = 'dc=tmargaux,dc=priv';  # Remplacez par votre DN de base
my $filter      = '(uid=*)';              # Filtre pour récupérer tous les utilisateurs

# Connexion à LDAP
my $ldap = Net::LDAP->new($ldap_server) or die "Erreur de connexion: $@";

# Recherche
my $mesg = $ldap->search(
    base   => $base_dn,
    filter => $filter,
);

# Vérification des erreurs de recherche
if ($mesg->is_error) {
    die "Erreur lors de la recherche: " . $mesg->error;
}

# Affichage des résultats
foreach my $entry ($mesg->entries) {
    print "DN: " . $entry->dn . "\n";  # Affiche le DN de chaque entrée
}

# Déconnexion
$ldap->unbind;
```

Objectif : Le script permet d'automatiser les requêtes vers l'annuaire LDAP. Il permet de fournir par un moyen simple des données d'OpenLDAP comme les utilisateurs ou les groupes. Le script recherche des entrées (composé d'un DN et d'attributs) dans l'annuaire avec des filtres. Il inclut également une vérification pour gérer les erreurs de connexion et de recherche.

Avant la configuration, installer le module CPAN : *cpan NET::LDAP*
CPAN est un module puissant permettant qui interagit avec les serveurs LDAP dans les scripts PERL. Il facilite la gestion de données et les rend plus efficaces.

```
use strict;
use warnings;
use Net::LDAP;
```

- Use strict et use warnings sont des directives qui aident à détecter les erreurs dans le code.
- use Net::LDAP ; : Charge le module nécessaire pour interagir avec un serveur LDAP.

➢ Configuration pour la connexion :

```
my $ldap_server = 'ldap://192.168.133.129';  # l'URL du serveur LDAP
my $base_dn     = 'dc=tmargaux,dc=priv';  # DN de base
my $filter      = '(uid=*)';              # Filtre pour récupérer tous les utilisateurs
```

➢ Connexion à LDAP et message d'erreur :

Création d'une nouvelle connexion au serveur LDAP et si la connexion échoue le script renvoie un message d'erreur et s'arrête.

```
my $ldap = Net::LDAP->new($ldap_server) or die "Erreur de connexion: $@";
```

➢ Mise en place d'un filtre

Cette étape permet d'exécuter une recherche dans l'annuaire en utilisant le DN de base et le filtre spécifié.

```
my $mesg = $ldap->search(
    base   => $base_dn,
    filter => $filter,
);
```

➢ Message d'erreur pour la recherche

Si la recherche a échoué, le script renvoi un message d'erreur.

```
if ($mesg->is_error) {
    die "Erreur lors de la recherche: " . $mesg->error;
}
```

➢ Affichage des résultats et déconnexion

```
foreach my $entry ($mesg->entries) {
    print "DN: " . $entry->dn . "\n";  # Affiche le DN de chaque entrée
}

# Déconnexion
$ldap->unbind;
```

On enregistre ce script dans un nouveau fichier puis on lui donne les droits et on l'exécute : ./ldap_perl.pl.

En exécutant le script, nous obtenons pour chaque entrée trouvée, le DN :

```
[root@tmargaux Documents]# nano ldap_perl.pl
[root@tmargaux Documents]# ./ldap_perl.pl
DN: uid=jdoe,ou=People,dc=tmargaux,dc=priv
```

DN: uid:jdoe, ou=People, dc=tmargaux, dc=priv

VI-Audit sous Linux

a) Définition d'un audit

Un audit informatique permet d'analyser et d'évaluer les risques et de proposer ainsi des points d'améliorations.

Pourquoi réaliser un audit informatique ?

Le plus souvent les audits sont réalisés à la demande d'une entreprise pour qu'à la fin l'entreprise prenne de meilleures décisions stratégiques. Mieux évaluer les performances de son système d'information, la rendre plus performantes, mise en conformité (RGPD).

J'ai utilisé comme outil d'audit de sécurité "Lynis". Il teste les défenses de sécurité, effectue des analyses, des paquets de logiciels vulnérables et des problèmes de configuration.

b) Installation Lynis

L'installation est assez rapide et s'effectue sous la forme de Git. Git est un logiciel de gestion de versions. Il permet de faire le suivi et de garder les anciennes versions sans rien n'écraser. Des dépôts sont en open-sources sur des plateformes.

Pour se faire il faut installer le package *Git: yum install git*
Puis faire une copie d'un dépôt : *git clone https://github.com/CISOfy/lynis*
Et enfin cette commande qui permet de changer de répertoire dans lequel on se trouve pour celui contenant lynis. Elle exécute ensuite le script Lynis pour effectuer l'audit sur le système : *cd lynis && ./lynis audit system*

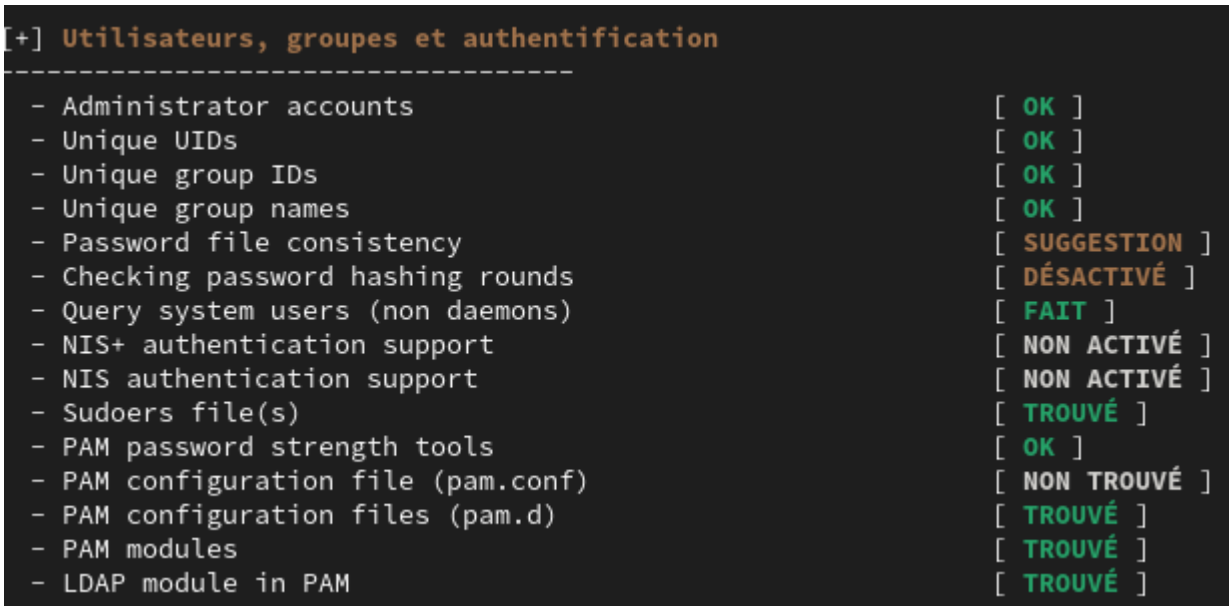
Après l'exécution de cette commande, une analyse de sécurité détaillée du serveur LDAP apparaît :

```
################################################################################
  Lynis comes with ABSOLUTELY NO WARRANTY. This is free software, and you are
  welcome to redistribute it under the terms of the GNU General Public License.
  See the LICENSE file for details about using this software.

  2007-2024, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)
################################################################################


[+] Initializing program
------------------------------------

  ############################################################
  #                                                          #
  #   NON-PRIVILEGED SCAN MODE                               #
  #                                                          #
  ############################################################

  NOTES:
  --------------
  * Some tests will be skipped (as they require root permissions)
  * Some tests might fail silently or give different results

  - Detecting OS...                                          [ DONE ]
  - Checking profiles...                                     [ DONE ]
  - Detecting language and localization                      [ fr ]

  ---------------------------------------------------
  Program version:           3.1.3
  Operating system:          Linux
  Operating system name:     RHEL
  Operating system version:  9.4
  Kernel version:            5.14.0
  Hardware platform:         x86_64
  Hostname:                  tmargaux
  ---------------------------------------------------
  Profiles:                  /home/tmargaux/lynis/default.prf
  Log file:                  /home/tmargaux/lynis.log
  Report file:               /home/tmargaux/lynis-report.dat
  Report version:            1.0
  Plugin directory:          ./plugins
  ---------------------------------------------------
  Auditor:                   [Not Specified]
  Language:                  fr
  Test category:             all
  Test group:                all
```

Voici la liste des services installés sur le serveur LDAP. La valeur entre parenthèse représente la valeur d'exposition du service, une évaluation de la sécurité. Plus elle est élevée, plus le service est considéré comme plus sécurisé. Ici presque tous les services ont une valeur tournant approximativement autour d'une valeur de sécurité de 9. Entre crochet en bout de ligne, "risque" signifie que la probabilité du service ait des vulnérabilités exploitables. "Exposé" signifie le niveau d'accessibilité du service sur le réseau.

```
[+] Démarrage et services
------------------------------------
  - Service Manager                                          [ systemd ]
    - Boot loader                                            [ NONE FOUND ]
  - Check running services (systemctl)                       [ FAIT ]
        Result: found 37 running services
  - Check enabled services at boot (systemctl)               [ FAIT ]
        Result: found 49 enabled services
  - Check startup files (permissions)                        [ OK ]
  - Running 'systemd-analyze security'
      Unit name (exposure value) and predicate
      ----------------------------------
    - ModemManager.service (value=6.3)                       [ MOYEN ]
    - NetworkManager.service (value=7.8)                     [ EXPOSÉ ]
    - accounts-daemon.service (value=9.6)                    [ RISQUÉ ]
    - alsa-state.service (value=9.6)                         [ RISQUÉ ]
    - atd.service (value=9.6)                                [ RISQUÉ ]
    - auditd.service (value=8.9)                             [ EXPOSÉ ]
    - avahi-daemon.service (value=9.6)                       [ RISQUÉ ]
    - chronyd.service (value=3.9)                            [ PROTÉGÉ ]
    - colord.service (value=8.8)                             [ EXPOSÉ ]
    - crond.service (value=9.6)                              [ RISQUÉ ]
    - cups.service (value=9.6)                               [ RISQUÉ ]
    - dbus-broker.service (value=8.7)                        [ EXPOSÉ ]
    - dm-event.service (value=9.5)                           [ RISQUÉ ]
    - emergency.service (value=9.5)                          [ RISQUÉ ]
    - firewalld.service (value=9.6)                          [ RISQUÉ ]
    - fwupd.service (value=7.7)                              [ EXPOSÉ ]
    - gdm.service (value=9.8)                                [ RISQUÉ ]
    - getty@tty1.service (value=9.6)                         [ RISQUÉ ]
    - irqbalance.service (value=8.9)                         [ EXPOSÉ ]
    - iscsid.service (value=9.5)                             [ RISQUÉ ]
    - iscsiuio.service (value=9.5)                           [ RISQUÉ ]
    - libstoragemgmt.service (value=9.6)                     [ RISQUÉ ]
    - low-memory-monitor.service (value=6.3)                 [ MOYEN ]
    - lvm2-lvmpolld.service (value=9.5)                      [ RISQUÉ ]
    - mcelog.service (value=9.6)                             [ RISQUÉ ]
    - mdmonitor.service (value=9.6)                          [ RISQUÉ ]
    - mlocate-updatedb.service (value=8.1)                   [ EXPOSÉ ]
    - multipathd.service (value=9.5)                         [ RISQUÉ ]
    - plymouth-start.service (value=9.5)                     [ RISQUÉ ]
    - polkit.service (value=9.6)                             [ RISQUÉ ]
    - power-profiles-daemon.service (value=7.6)              [ EXPOSÉ ]
    - rc-local.service (value=9.6)                           [ RISQUÉ ]
    - rescue.service (value=9.5)                             [ RISQUÉ ]
```

Ce fichier contient des vérifications sur les comptes utilisateurs, les configurations liées à l'authentification et aussi à la gestion des utilisateurs.



Administrator accounts traite les comptes d'administrateurs sur le système et vérifie qu'ils sont configurés correctement.

Le "**OK**" a côté signifie que le système a passé la vérification sans problèmes.

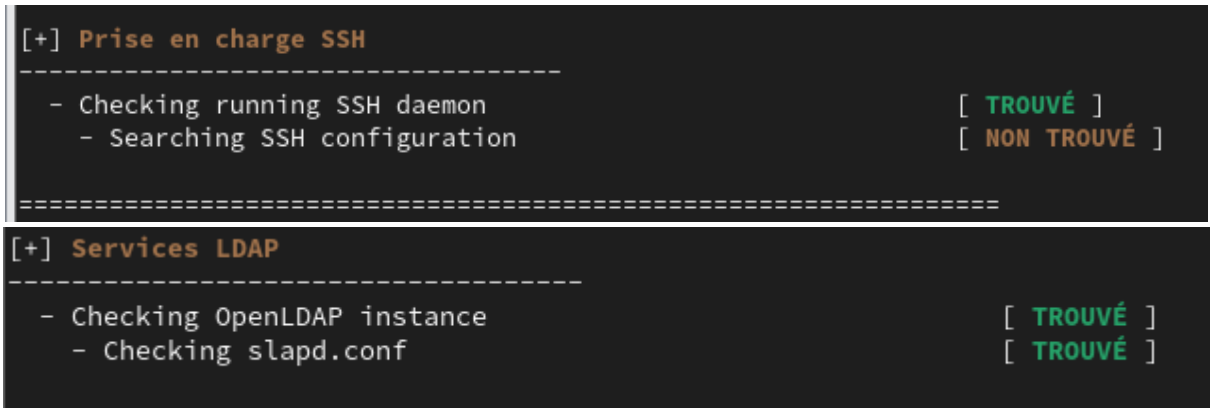
"**Trouvé**" signifie que l'outil a exécuté un élément ou un résultat lors de l'analyse. Cela veut dire: un service est en cours d'exécution, un fichier de configuration est présent ou encore une vulnérabilité a été détectée.

Unique UIDS vérifie que chaque utilisateur a un identifiant unique (UID), de même pour **group IDs** pour les groupes.

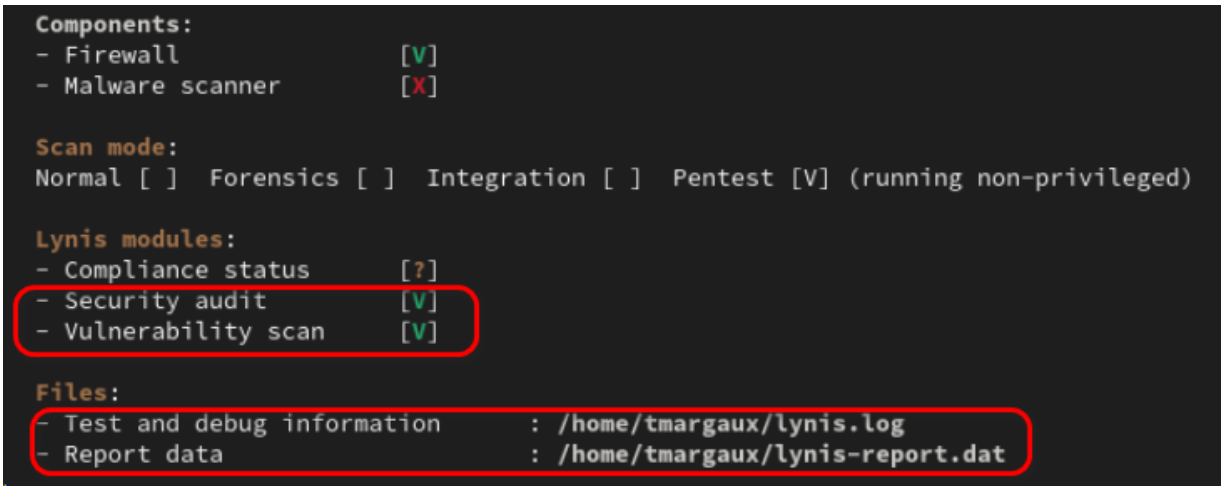
Vérifications concernant PAM:

- **PAM password strength tools**: le fichier de configuration principal de PAM correctement configuré ("OK")
- **Pam.d:** fichier de configuration spécifiques de PAM pour chaque service
- **Pam modules :** modules PAM correctement installés ainsi que leurs configurations
- **LDAP module in PAM :** vérifie si le module LDAP est utilisé dans PAM, il permet l'authentification sur l'annuaire.

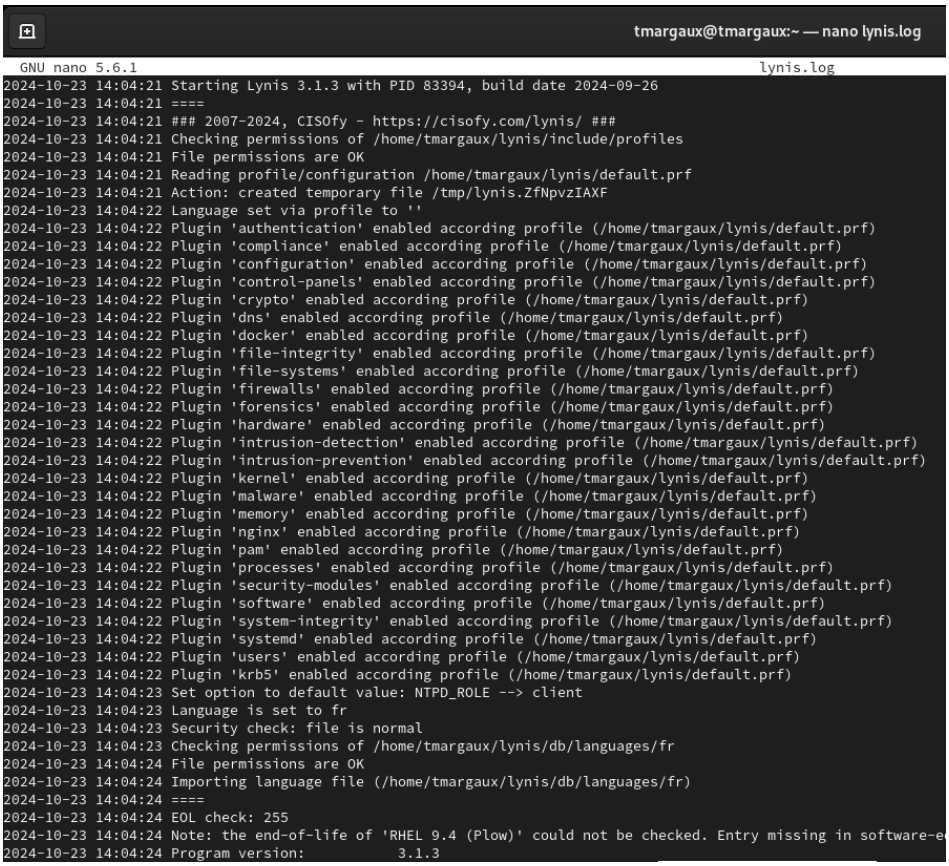
Ici, le “daemon” SSH est en cours d’exécution. C’est le même cas pour OpenLDAP et slapd.d:



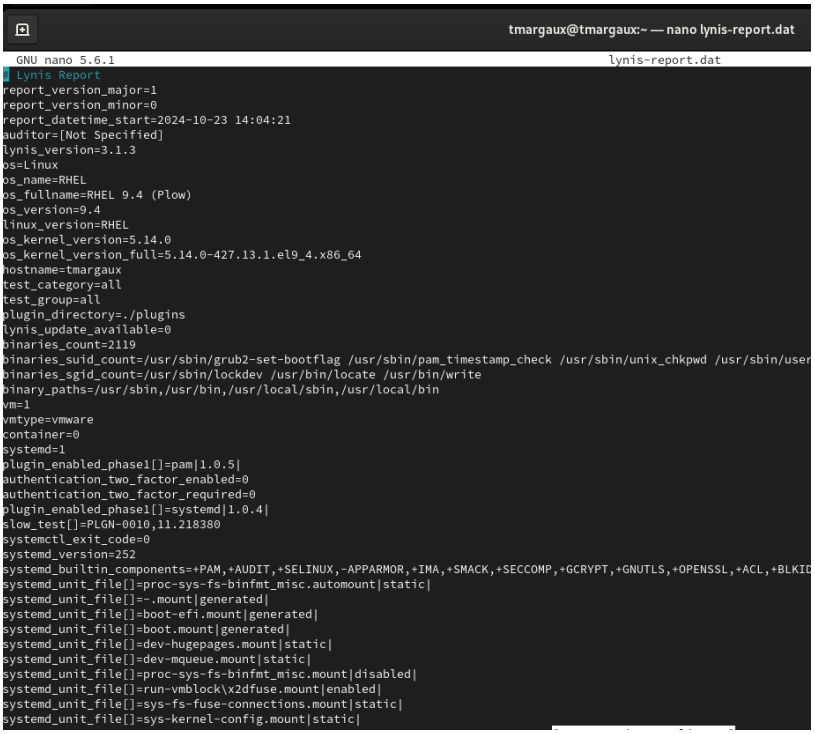
Ici il est précisé que l’audit a bien été réalisé et effectué le test. Il y a le répertoire dans lequel les rapports ont été extrait.



Et voici les extraits de mes fichiers log du rapport :



Fichier lynis.log pour la journalisation sur l'analyse de sécurité ainsi que chaque contrôle effectué



Fichier lynis-report.dat qui contient le résumé structuré des résultats de l'audit

CONCLUSION :

À la fin de ce projet, j'ai su mettre en place un annuaire OpenLDAP avec redhat en passant par l'installation avec la mise en place de schémas, d'attribut, d'OU, de client, mot de passe, ajout de données ; la configuration du serveur et de son client. Cela permet la centralisation pour gérer les utilisateurs mais aussi l'authentification et l'accès de manière sécurisée.

Pour l'authentification et afin d'étendre l'accès à l'annuaire, la configuration des fichiers nsswitch.conf, sssd.conf, PAM et ldap.conf ont été configurés. Pour la gestion de mot de passe, cela a été renforcée par PAM ainsi que la création de répertoire via oddjobd.mkhomedir. Grâce aux différentes manières de sauvegardes, cela garantit une récupération de données de l'annuaire en cas de sinistre.

Une mise en place d'index a été opté pour faciliter l'exploitation de l'annuaire en réduisant le temps de réponses et ainsi améliorer l'efficacité de l'annuaire. De plus, grâce au script PERL qui permet d'interroger l'annuaire, cela rend les interactions de l'annuaire plus rapides et efficaces.

Enfin, grâce à l'audit de sécurité Lynis, j'ai réalisé une évaluation complète du système afin d'identifier les points de vulnérabilités.