SpaceFox

Zeste de savoir

Travailler avec LDAP en 2022

21 novembre 2022

Table des matières

Introduction

Salut les agrumes,

J'ai eu besoin récemment de travailler avec LDAP, et ce que j'y ai trouvé m'a pas mal surpris. Voici donc un billet pour préciser quelques points avec cette technologie, si ça peut vous éviter de perdre autant de temps que j'en ai perdu, ça sera toujours ça de pris.

1. Qu'est-ce que LDAP ?

Aujourd'hui, c'est une norme pour les systèmes d'annuaires, on peut le voir comme une base de données arborescente. Cf Wikipédia ou ldap.com pour plus de détails.

2. Un serveur LDAP ?

Historiquement, le serveur de référence est OpenLDAP. C'est beaucoup moins vrai aujourd'hui, et il existe des alternatives: plus de détails ici. Par contre, OpenLDAP continue à être la référence d'implémentation, pour le meilleur et pour le pire. En particulier, OpenLDAP est atrocement complexe à utiliser[1], et à tendance à tout casser à chaque mise à jour (qui ne suivent *absolument pas* SemVer).

Enfin, tout ça c'est si vous avez la possibilité de choisir le serveur, ce qui est rarement le cas en pratique.

1. Dédicace à la disparition du fichier de configuration compréhensible, remplacé par des commandes LDAP à lancer sur la base de données LDAP de configuration… résultat, le moyen le plus simple d'initialiser une configuration OpenLDAP moderne est… d'utiliser l'ancien fichier de config, et de lui accoler la moulinette (officielle, heureusement) de compatibilité.

D'autre part, on peut utiliser le protocole LDAP pour se connecter à Azure Active Directory , le serveur d'annuaire de Microsoft, et ça sera souvent le cas en entreprise. C'est bien le même protocole qu'avec les autres implémentations, mais avec plein de microsofteries au milieu.

3. La documentation sur LDAP

La documentation d'OpenLDAP semble très complète , mais est difficilement utilisable: c'est plus un aide-mémoire à destination des personnes qui savent faire plutôt qu'une vraie documentation.

Le livre sur LDAP sur zytrax.com (non-libre mais en accès libre) m'a été d'une bien plus grande aide.

4. L'aide sur LDAP sur Internet

C'est le point le plus important de ce billet:

> Personne, sur Internet, ne sait se servir de LDAP.

Je n'ai trouvé virtuellement aucune aide utile sur LDAP sur Internet. On est dans un culte du cargo massif et généralisé, chacun y va de son petit bout de code miracle à copier/coller. Dans le meilleur des cas, on a une solution qui était techniquement exacte, mais il y a plus de 10 ans – et complètement obsolète aujourd'hui.

Je *suppose* que c'est parce qu'en réalité personne ou presque ne développe quoi que ce soit avec LDAP (contrairement à SQL par exemple, qui est massivement utilisé par les développeurs eux-mêmes). Les utilisateurs de LDAP semblent être surtout des administrateurs système qui savent *administrer* un LDAP pour une utilisation simple, mais qui n'ont pas besoin de compétences avancées ni d'une bonne compréhension du truc.

5. Communiquer avec LDAP

En ligne de commande

Ça va dépendre de votre implémentation, mais avec OpenLDAP, il y a toute une batterie de commandes différentes selon ce que vous voulez faire: `ldapsearch`, `ldapadd`, `ldapmodify`… leur syntaxe est assez absconse, et les différentes options ont tendance à se cumuler pour faire des effets supplémentaires quand utilisées ensemble. Les amateurs de jeux de carte à collectionner ou de jeux de decks se sentiront chez eux, les autres… moins.

Par interface graphique

La *seule* interface graphique encore développée est Apache Directory Studio , basée sur Eclipse. Elle est d'une ergonomie catastrophique: elle ne respecte à peu près aucun des standards habituels de l'ergonomie, les informations sont éparpillées partout (dédicace à la gestion des logs) et utilise énormément de caches qui font qu'on travaille sur des données complètement périmées (vérifier la présence d'un bouton de rafraichissement sur l'interface pour avoir les données à jour).

Mais c'est la seule: le plugin pour les IDE JetBrains ou phpLDAPadmin ne sont plus développés depuis des années et ne fonctionnent plus. Et encore, quand je dis que Apache Directory Studio est «développé» c'est à prendre avec des pincettes: la dernière *release* officielle date de 2010, depuis on tourne sur des «milestones» d'une future version 2.0 (qui sont en réalité la seule version utilisable).

Via les API

À voir ce que vous avez comme API disponible dans votre langage.

Dans le cas de la JVM, c'est la jungle: plein de monde s'est décidé à faire son implémentation dans son coin, beaucoup de ces implémentations ne sont plus maintenues et surtout sont inutilisables. Quand les implémentations ne sont pas partielles, elles sont scandaleusement lentes. Par exemple, sur un projet, on a gagné un facteur 100 (cent) sur les performances juste en changeant *la bibliothèque de connexion au LDAP* – pas le serveur LDAP, hein: juste la bibliothèque de connexion.

Si vous utilisez la JVM, je conseille UnboundID LDAP SDK , qui est encore maintenue, est complète et a des bonnes performances.

6. Des points internes à LDAP

Du vocabulaire

«DN» est l'identifiant d'une entrée LDAP.

On appele «bind» l'action de s'authentifier. Le login est donc un «bind DN».

Les objets LDAP ont des classes (une ou plusieurs classes par objet), qui elles-mêmes ont des attributs (obligatoires ou non). C'est la présence de ces attributs qui déclenche certains comportements, comme la possibilité de s'authentifier avec un objet (qui a donc un attribut de mot de passe).

Tout attribut non déclaré dans une des classes de l'objet est interdit, sauf si l'objet a une classe qui le permet et dont j'ai oublié le nom. Certains attributs sont obligatoires (c'est la classe qui décide quels attributs sont obligatoires).

Des formats de fichier

Le format LDIF (l'équivalent du SQL pour LDAP) est très pinailleur sur des choses comme la longueur de ligne, la présence de sauts de lignes à des endroits précis, ou la présence d'un tiret seul sur une ligne (seul, c'est *vraiment* seul, sans espaces ni rien).

```

```

Les deux premiers changements (séparés par un `-` sur une ligne et pas de ligne blanche) concernent la même entrée `uid=myaccount,ou=accounts,dc=example,dc=com`; le `-` suivi d'une ligne blanche indique qu'on va gérer une *autre* entrée[2].

Le format de définition des schémas (classes et attributs) est particulièrement abscons, notamment à cause de la présence d'OID, des identifiants qui se présentent comme une suite de nombres. Voyez plutôt:

```

```

L'entrée `SYNTAX` définit le type de données, `1.3.6.1.4.1.1466.115.121.1.15` désigne une chaine de caractères Unicode, et `1.3.6.1.4.1.1466.115.121.1.40` une suite arbitraire d'octets. C'est pourtant clair, non ?

Le pire, c'est que ces fichiers ne sont même pas utilisés directement: ils sont d'abord convertis en LDIF – qui sont tellement complexes qu'il est vain de vouloir les écrire à la main.

Un détail amusant avec les OID: on peut *acheter* (je ne sais pas comment) son propre préfixe, pour pouvoir créer des classes et des attributs publics sans risque de confusion avec ceux d'autres entreprises.

La conservation d'état (protocole *stateful*)

(Ce point dépend peut-être de votre bibliothèque, ici c'est testé avec UnboundID).

Le protocole LDAP est *stateful*, c'est-à-dire qu'il conserve un état d'une requête sur l'autre, et que le résultat des requêtes va dépendre de cet état préalable. Ça a un côté très pratique pour rendre le protocole peu verbeux en interne (il n'y a pas besoin de répéter des informations déjà connues du serveur) mais pose aussi d'énormes problèmes de cohérence et de reproductibilité, au point qu'aujourd'hui on essaie au maximum d'utiliser des communications sans état (*stateless*).

L'impact le plus gênant, c'est le couplage entre l'état de la connexion au LDAP et l'opération d'authentification `bind`. Cette opération va permettre de savoir si un couple (DN, *credential*)[3] a le droit de se connecter au LDAP. Le problème, c'est que cette opération va *aussi* modifier l'utilisateur connecté au LDAP en remplaçant les informations existantes par celle du nouveau `bind`.

2. Ça devient amusant quand on découvre que la nouvelle version majeure de sa bibliothèque – la seule à recevoir encore des correctifs de sécurité – *ne gère pas* de fichier LDIF contenant des `-`… ce qui impose immédiatement et irrévocablement de changer de bibliothèque.
3. Respectivement nommés «identifiant» et «mot de passe» hors de l'univers LDAP.

Ainsi, votre connexion partagée avec des droits d'administration peut se retrouver remplacée par une connexion avec un utilisateur normal, ou par une connexion non fonctionnelle si le `bind` a échoué, ou n'importe quelle combinaison catastrophique en terme de fonctionnement voire de sécurité.

Spécial OpenLDAP

On ne peut pas changer le DN racine d'une base OpenLDAP une fois cette base créée.

Conclusion

En résumé, si vous devez développer quelque chose avec LDAP, tant que vous vous contenterez de faire des opérations simples (comme vérifier qu'un utilisateur peut se connecter) avec une bibliothèque d'API performantes, ça devrait bien se passer.

Dès que vous essayerez quelque chose de plus complexe, préparez-vous à lire des tonnes de documentation pour comprendre ce qu'il se passe, à faire beaucoup d'essais avec des outils à l'ergonomie inexistante ; et surtout : n'espérez pas trouver de l'aide sur Internet.

Logo : Logo de ldap.com .