slapd-ldap

Backend ldap

Ce backend n'est pas une base, il agit comme proxy pour transférer des requêtes entrantes à un autre serveur LDAP. Les référants sont pleinement traités au lieu d'être retournés au client. Les sessions qui Bind explicitement avec ce backend créent toujours leur propre connexion privée au serveur LDAP distant. Les sessions anonymes vont partager une seule connexion anonyme. Pour les sessions liées avec d'autres mécanismes, toutes les sessions avec le même DN vont partager la même connexion. Celà permet d'améliorer l'efficacité du proxy.

La base ldap peut aussi agir comme service d'information, par ex. l'identité des clients authentifiés localement est affirmé sur le serveur distant, possiblement sous une forme modifiée. Dans ce but, le proxy bind au serveur distant avec une identité administrative, et, si requis, authorise l'identité fournie.

L'instance proxy de slapd doit contenir le schéma pour les attributs et les classes d'object utilisé dans les requêtes. slapd doit être configuré avec le support des threads, et le paramètre threads adapté.

Configuration

uri <ldapurl> Serveur LDAP à utiliser. Plusieurs URI peuvent être spécifiés (ex : uri "ldap ://host/ ldap ://backup-host/")

acl-bind bindmethod=simple|sasl [binddn=<simple DN>] [credentials=<simple password>] [saslmech=<SASLmech>]

[secprops=<properties>] [realm=<realm>] [authcId=<authenticationID>] [authzId=<authorization ID>]

[starttls=no|yes|critical] [tls_cert=<file>] [tls_key=<file>] [tls_cacert=<file>] [tls_cacertdir=<path>]

[tls_reqcert=never|allow|try|demand] [tls_ciphersuite=<ciphers>] [tls_protocol_min=<version>] [tls_crlcheck=none|peer|all] Permet de définir les paramètres de la méthode d'authentification utilisé par le proxy pour collecter les informations liées au contrôle d'accès L'identité définie par cette directive doit avoir l'accès read sur le serveur cible aux attributs utilisés sur le proxy pour la vérification des ACL.

cancel {ABANDON|ignore|exop [-discover]} Définis comment manipuler les annulations d'opérations. abandon (abandonne immédiatement l'opération), ignore (aucune action et la réponse est ignorée), exop (une opération cancel est envoyée au serveur distant), exop-discover (supporte l'opération étendue cancel)

chase-referrals {YES|no} Active/désactive le repérage de référence automatique, qui est déléguée à libldap, avec un rebind éventuel si la directive rebind-as-user est utilisée défaut : yes

conn-ttl <time> Force la suppression d'une connexion en cache et la recrée après le ttl donné, sans regarder s'il elle est active ou non.

idassert-authzFrom <authz-regexp> Sélectionne quelles identités locales sont autorisées à exploiter la fonction d'assertion d'identité

idassert-bind bindmethod=none|simple|sasl [binddn=<simpleDN>] [credentials=<simple password>]

[saslmech=<SASL mech>] [secprops=<properties>] [realm=<realm>] [authcId=<authenticationID>]

[authzId=<authorization ID>] [authz=native|proxyauthz] [mode=<mode>] [flags=<flags>] [starttls=no|yes|critical]

[tls_cert=<file>] [tls_key=<file>] [tls_cacert=<file>] [tls_cacertdir=<path>] [tls_reqcert=never|allow|try|demand]

[tls_ciphersuite=<ciphers>] [tls_protocol_min=<version>] [tls_crlcheck=none|peer|all] Permet de définir les paramètres de la méthode d'authentification utilisée en interne par le proxy pour autoriser les connexions qui sont authentifiées par d'autres bases de données. L'identité définie doit avoir l'accès auth sur le serveur cible sur les attributs utilisés et doit avoir le droit d'authoriser d'autres users (proxyAuth sur tout le DN, ex : authzTo=dn.subtree :"") et le serveur distant doit avoir authz-policy mis à to ou both.

<mode> := {legacy|anonymous|none|self} legacy implique que le proxy va effectuer un simple bind (authcDN) ou SASL (authcID) et assert l'identité du client quand il n'est pas anonyme. Anonymous et self, assert une chaîne vide ou l'identité du client, respectivement.

<flag> := {override|[non-]prescriptive|proxy-authz-[non-]critical} override, l'assertion de l'identité prend place même quand la base est autorisée pour l'identité du client. Quand prescriptive est utilisé, les opérations échouent avec inappropriateAuthentication pour les identités dont l'assertion n'est pas permise par le motif idassert-authzFrom. non-prescriptive, les opérations sont effectuées anonymement pour les identités dont l'assertion n'est pas permise par le pattern idassert-authzFrom. proxy-authz-non-critical, le contrôle proxyAuthz contrôle n'est pas marqué comme critique, en violation de la RFC 4370. il est recommandé d'utiliser proxy-authz-critical

idassert-passthru <authz-regexp> Si définis, sélectionne quelle identité local bypasse l'assertion d'identité. ces identités doivent être connues par l'hôte distant. la chaîne authz-regexp suit les règles définies pour authzFrom

idle-timeout <time> Force une connexion en cache à être supprimée puis recrée après une timeout

keepalive <idle> :<probes> :<interval> Définis les valeurs de idle, probes et interval utilisés pour certifier si un socket est actif. idle est le nombre de secondes avant qu'une connexion non-active reçoive un keepalive. probes et le nombre maximum de keepalive, et interval est le nombre de seconde entre chaque keepalive.

network-timeout <time> Définis la valeur timeout réseau après que poll(2)/select(2) suivant un retour connect(2) en cas d'inactivité, en seconde

norefs <NO|yes> à yes, ne retourne pas de réponse de référence de recherche. (défaut : NO, sauf si LDAPv2)

noundeffilter <NO|yes> à yes, retourne success au lieu de rechercher si un filtre est indéfinis ou contient des portions indéfinies. Par défaut, la recherche est propagée après avoir remplacé les portions indéfinies avec (!(objectClass=*)), qui correspond à un résultat vide.

onerr {CONTINUE|stop} Permet de sélectionner le mode en cas d'erreur retourné par le serveur distant durant une recherche. à Continue, retourne un success. à stop, l'erreur est retournée au client.

protocol-version {0,2,3} Indique quel protocole utiliser pour contacter le serveur distant (0, le protocole utilisé est le même que celui utilisé par le client)

proxy-whoami NO|yes Active le proxy de l'opération WhoAmI, back-ldap va remplacer la routine whoami de slapd avec le sien.

quarantine <interval>,<num>[;<interval>,<num>[...]] Met en quarantaine les URI qui ont retourné LDAP_UNAVAILABLE. retente seulement à interval seconde depuis la dernière tentative, pour exactement num fois ; puis utilise le prochain pattern. si num vaut +, retente indéfiniment.

rebind-as-user {NO|yes} à yes, les crédential du client sont mémorisés pour rebind, en tentant de re-établir une connexion perdue, ou lors de la poursuite d'aiguillage si chase-referrals est à yes.

session-tracking-request {NO|yes} Ajoute un contrôle de traçage de session pour toutes les requêtes. l'IP/hostname du client et l'identité associée à chaque requête sont envoyés au serveur distant pour information. Est incompatible avec protocol-version 2.

single-conn {NO|yes} Annule toute connexion en cache si le client se rebind

t-f-support {NO|yes|discover} Active le support des filtres absolus des serveurs distant (RFC4526).

timeout [<op>=]<val> [...] Permet de définir des timeouts par opération. <op> : := bind, add, delete, modrdn, modify, compare, search

tls {[try-]start|[try-]propagate|ldaps} [tls_cert=<file>] [tls_key=<file>] [tls_cacert=<file>] [tls_cacertdir=<path>]

[tls_reqcert=never|allow|try|demand] [tls_ciphersuite=<ciphers>] [tls_crlcheck=none|peer|all] Spécifie l'utilisation de TLS pour l'initialisation des connexions.

use-temporary-conn NO|yes Créé une connexion temporaire en compétition avec d'autres threads pour une connexion partagée. Sinon, attent jusqu'à ce qu'une connexion partagée soit disponible.