Cyrus SASL

Serveur SASL de Cyrus

Composants SASL

Le premier composant de la librairie SASL est la couche Glue. Il s'occupe des tâches de base :

- Charger les plugins
- Gère les propriétés de sécurité nécessaires de l'application pour l'aider dans le choix du mécanisme ou pour limiter les mécanismes disponibles.
- Lister les plugins disponibles au applications
- Choisir le meilleur mécanisme dans une liste pour une authentification particulière
- Router les paquets de données d'authentification entre l'application et le mécanisme choisi
- Fournir des informations sur la négociation SASL aux applications

Il fournis également d'autres services aux plugins et applications tels que l'encodage MIME base 64, génération de nombres aléatoires, etc. Il permet également aux mécanismes et aux applications d'accéder à types spéciaux de plugin. Auxiliary Property "auxprop" qui fournis une interface de base simple et peut retourner des propriétés sur l'utilisateur tel que le mot de passe, répertoire personnel, ou adresse mail. et Username Canonicalization, qui peut fournir des manières de canoniser un nom d'utilisateur ou effectuer d'autres tâches.

Mécanisme PLAIN et mots de passe en texte clair

auxprop Vérifie les mots de passe dans userPassword fournis par une plugin de propriété auxiliaire. Par exemple, SASL est fournis avec sasldb, qui peut être utilisé pour authentifier avec des mots de passe stockés dans /etc/sasldb2.

saslauthd Permet de vérifier des mots de passe en utilisant divers mécanismes.

courrier-IMAP authdaemond Utilisé pour contacter authdaemond pour vérifier les mots de passe.

Mécanismes à secret partagé

Cyrus SALS fournis plusieurs méthodes d'authentification basé sur un secret partagé : CRAM-MD5 et son successeur DIGEST-MD5.

Mécanismes Kerberos

Cyrus SASL fournis 2 mécanismes qui utilisent kerberos : Kerberos_v4 et GSSAPI qui permet d'utiliser kerberos v5.

Mécanismes OTP

Cyrus SASL support le mécanisme OTP, similaire à CRAM-MD5 et DIGEST-MD5. Ces OTP sont stockés dans /etc/sasldb2. SASL support également OPIE.

Auxiliary Properties

SASLv2 introduit le concept de propriétés auxiliaires. C'est la capacité de rechercher des informations liées à l'authentification ou l'authorisation depuis un répertoire durant le processus d'authentification.

Fichier de configuration par défaut

Par défaut, Cyrus SASL lit ses options depuis /usr/lib/sasl2/App.conf (où App est le nom de l'application). Les applications peuvent redéfinir comment la librairie recherche les informations de configuration.

OPTIONS

authdaemond_path (SASL library) Chemin du socket UNIX de authdaemond (défaut : /dev/null)

auto_transition (SASL library) à yes ou noplain et utilisé avec auxprop. Transferts automatiquement les users à d'autres mécanismes quand ils font une authentification plaintext réussie. à noplain, seul les secrets non plaintext sont écris. (défaut : no)

auxprop_plugin (AuxProp Plugin) Nom(s) de(s) plugin(s) auxiliaire à utiliser. (défaut : null)

canon_user_plugin (SASL library) Nom du plugin canon_user à utiliser (défaut : INTERNAL)

keytab (GSSAPI) Emplacement du fichier keytab (défaut : /etc/krb5.keytab)

ldapdb_uri (LDAPDB plugin) liste d'URI ldap

ldapdb_id (LDAPDB plugin) id d'authentification ldap

ldapdb_mech (LDAPDB plugin) mécanisme pour l'authentification

ldapdb_pw (LDAPDB plugin) password

ldapdb_rc (LDAPDB plugin) Fichier à placer dans l'environnement LDAPRC

ldapdb_starttls (LDAPDB plugin) utiliser StartTls (try, demand).

ldapdb_canon_attr (LDAPDB plugin) attribut contenant le nom canonique de l'utilisateur

log_level (SASL library) Niveau de debug (défaut : 1 (SASL_LOG_ERR))

mech_list (SASL library) liste de mécanisme à autoriser (défaut : tous)

ntlm_server (SASL library) Liste des serveurs

ntlm_v2 (SASL library) Envoyer des réponses ntlmv2 au serveur opiekeys (OPIE) Emplacement du fichier opiekeys (défaut : /etc/opiekeys)

otp_mda (OPIE) Algorithme pour l'OTP (md4, md5, sha1) (défaut : md5)

plugin_list (SASL library) Emplacement de la liste des plugins

pwcheck_method (SASL library) liste des mécanismes utilisés pour vérifier les mots de passe, utilisé par sasl_checkpass (auxprop, saslauthd, pwcheck, authdaemond, alwaystrue). (défaut : auxprop)

reauth_timeout (DIGST-MD5) Temps en minutes de cache de l'authentification pour une réauthentification rapide. défaut : 0

saslauthd_path (SASL library) Chemin vers le répertoire saslauthd, incluant le pipe /mux

sasldb_path (sasldb plugin) Chemin du fichier sasldb (défaut : /etc/sasldb2)

sql_engine (SQL plugin) Nom du moteur SQL à utiliser (mysql, pgsql, sqlite, sqlite3). défaut : mysql

sql_hostnames (SQL plugin) liste de serveurs :ports SQL

sql_user (SQL plugin) Utilisateur SQL pour l'authentification

sql_passwd (SQL plugin) Mode de passe pour l'authentification

sql_database (SQL plugin) Nom de la base de données qui contient les propriétés auxiliaires

sql_select (SQL plugin) Déclaration SELECT à utiliser pour récupérer les propriétés

sql_insert (SQL plugin) Déclaration INSERT pour créer des propriétés

sql_update (SQL plugin) Déclaration UPDATE pour mettre à jours des propriétés

sql_usessl (SQL plugin) à yes, créé une connexion sécurisée

srp_mda (SRP) Algorithme à utiliser pour les calculs SRP (md5, sha1, rmd160).

srvtab (kerberos 4) Emplacement du fichier srvtab (défaut : /etc/srvtab)

Notes sur les options auxprop sql

%U username

%p Nom de la propriété à modifier/ajouter

%r Royaume auquel l'utilisateur appartient

%v Valeur de la propriété

Exemples

sql_select : SELECT %p FROM user_table WHERE username = '%u' and realm = '%r'
Va envoyer la requête suivante pour l'utilisateur bovik et le royaume madoka.surf.org.uk :
SELECT userPassword FROM user_table WHERE username = 'bovik' and realm = 'madoka.surf.org.uk';
sql_insert : INSERT INTO user_table (username, realm, %p) VALUES ('%u', '%r', '%v')
Va générer les requêtes suivante pour l'utilisateur bovik dans le royaume madoka.surf.org.uk et le userPassword "wert" :
INSERT INTO user_table (username, realm, userPassword) VALUES ('bovik', 'madoka.surf.org.uk', 'wert');

Notes sur les options LDAPDB

s'active en créant /usr/lib/sasl2/slapd.conf avec :
auxprop_plugin : slapd

Exemples

ldapdb_uri : ldap ://ldap.example.com
ldapdb_id : root
ldapdb_pw : secret
ldapdb_mech : DIGEST-MD5
ldapdb_canon_attr : uid
root authcid doit avoir des privilèges de proxy authorization pour tous les comptes authorisés à s'authentifier.
ldapdb_uri : ldapi ://
ldapdb_mech : EXTERNAL
Cette configuration assume que le serveur LDAP est sur le même serveurs qui utilise SASL. C'est plus rapide et sécure et n'a pas besoin de stocker de user/password. slapd.conf doit mapper ces usernames en DN ldap :
sasl-regexp uidNumber=(.*)\\+gidNumber=(.*),cn=peercred,cn=external,cn=auth ldap ://dc=example,dc=com ? ?sub ?(&(uidNumber=$1)(gidNumber=$2))
sasl-regexp uid=(.*),cn=external,cn=auth ldap ://dc=example,dc=com ? ?sub ?(uid=$1)