draft-chu-ldap-kdc-schema-00

LDAP KDC Schema

Introduction

Motivations

Kerberos et LDAP sont fréquemment utilisés séparément pour une authentification distribuée. Ils peuvent également être utilisé conjointement, mais leur base de données restent généralement séparées. Ces bases séparées implique une duplication de données et sont source de charge d'administation supplémentaire. Il est donc préférable de partager la même base de données. Un certain nombre d'implémentations Kerberos supportent déjà l'utilisation de LDAP comme stockage du KDC. Cependant, chaque implémentation utilise son propre schéma.

Terminologie

Les OID définis ci-dessous sont dérivés de TBD.OID :

```
KRBSYN = TBD.OID.0
KRBATTR = TBD.OID.1
KRBOC = TBD.OID.2
```

Attributs

Les attributs et classes définies dans ce document sont (additionnellement, certains attributs définis dans LDAP Password Policy sont requis) :

krbPrincipalName Nom canonique du principal

krbPrincipalAliases Alias du nom du principal. Vu que cet attribut est un sous-type de krbPrincipalName, une recherche sur krbPrincipalName va également rechercher les alias.

krbTicketMaxLife Maintient le maximum ticket lifetime, en secondes pour un principal.

krbTicketMaxRenewal Maintient la durée maximal de renouvellement d'un ticket

krbEncSaltTypes Maintient les combinaisons chiffrement/salt permis pour ce principal

krbRealmName Nom du domaine Kerberos

krbPrincipalRealm DN de l'entrée krbRealm

krbKeySet Maintient les clés du principal, optionnellement chiffrés avec la clé maître. l'attribut est encodé en utilisant ASN.1

Le format de la valeur pour cet attribut est :

```
KrbKeySet ::= SEQUENCE {
 kvno [0] UInt32,
 mkvno [1] UInt32 OPTIONAL,
```

```
 keys [2] SEQUENCE OF KrbKey,
 ...
}

KrbKey ::= SEQUENCE {
 salt [0] KrbSalt OPTIONAL,
 key [1] EncryptionKey,
 s2kparams [2] OCTET STRING OPTIONAL,
 ...
}

KrbSalt ::= SEQUENCE {
 type [0] Int32,
 salt [1] OCTET STRING OPTIONAL
}

EncryptionKey ::= SEQUENCE {
 keytype [0] Int32,
 keyvalue [1] OCTET STRING
}
```

krbKeyVersion Stocke le numéro de version de la clé courante

krbTicketPolicy Définis les flags qu'un utilisateur peut utiliser ou demander à utiliser dans une demande de ticket

```
krb5KDCFlagsSyntax SYNTAX ::= {
 WITH SYNTAX INTEGER
  initial(0), - require as-req
  forwardable(1),- may issue forwardable
  proxiable(2), - may issue proxiable
  renewable(3), - may issue renewable
  postdate(4), - may issue postdatable
  server(5), - may be server
  client(6), - may be client
  invalid(7), - entry is invalid
  require-preauth(8), - must use preauth
  change-pw(9), - change password service
  require-hwauth(10), - must use hwauth
  ok-as-delegate(11), - as in TicketFlags
  user-to-user(12), - may use user-to-user auth
  immutable(13) - may not be deleted
 ID { 1.3.6.1.4.1.5322.10.0.1 }
}
```

krbExtraData Maintient des données arbitraires qui peuvent être utilisés par certaines implémentations. la valeur est encodée en DER ASN.1

```
ExtraData ::= SEQUENCE {
 tag [0] OCTET STRING,
 data [1] OCTET STRING
}
```

krbPrincNamingAttr Enregistre que attributs sont utilisés pour nommer les nouvelles créations de principal

krbPrincContainer Pointe vers un conteneur sous lequel les nouvelles entrées de principaux sont créés.

krbPwdPolicy Point vers une subentry de stratégie de mot de passe contenant la stratégie qui sera appliquée aux principaux Kerberos. Noter que pour les serveur avec le support complet des subentry, cet subentry définissent à quelles entrée elles s'applique. Donc cet attribut n'est pas nécessaire.

krbLDAPURI URI ldap que le KDC utilise pour localiser les principaux.

Classes d'objets

krbKDCInfo
krbPrincipal
krbRealm

Définition des attributs

```
( KRBATTR.1 NAME 'krbPrincipalName' DESC 'Canonical principal name' EQUALITY caseExactIA5Match SUBSTR
caseExactSubstringsMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )
( KRBATTR.2 NAME 'krbPrincipalAliases' SUP krbPrincipalName )
( KRBATTR.3 NAME 'krbTicketMaxLife' EQUALITY integerMatch ORDERING integerOrderingMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
( KRBATTR.4 NAME 'krbTicketMaxRenewal' EQUALITY integerMatch ORDERING integerOrderingMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
( KRBATTR.5 NAME 'krbEncSaltTypes' EQUALITY caseIgnoreMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
( KRBATTR.6 NAME 'krbRealmName' EQUALITY octetStringMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
( KRBATTR.7 NAME 'krbPrincipalRealm' DESC 'DN of krbRealm entry' SUP distinguishedName )
( KRBATTR.8 NAME 'krbKeyVersion' EQUALITY integerMatch ORDERING integerOrderingMatch SYNTAX
1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
( KRBATTR.9 NAME 'krbKeySet' EQUALITY octetStringMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
( KRBATTR.10 NAME 'krbTicketPolicy' EQUALITY integerMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
( KRBATTR.11 NAME 'krbExtraData' EQUALITY octetStringMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 )
( KRBATTR.12 NAME 'krbPrincNamingAttr' EQUALITY objectIdentifierMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.38
SINGLE-VALUE )
( KRBATTR.13 NAME 'krbPrincContainer' DESC 'DN of container entry for principals' SUP distinguishedName
SINGLE-VALUE )
( KRBATTR.14 NAME 'krbPwdPolicy' DESC 'DN of password policy subentry' SUP distinguishedName SINGLE-VALUE )
( KRBATTR.15 NAME 'krbLDAPURI' DESC 'LDAP search parameters for locating principals' SUP labeledURI )
```

Définitions des classes d'objet

(KRBOC.1 NAME 'krbKDCInfo' SUP top AUXILIARY MAY (krbTicketMaxLife $ krbTicketMaxRenewal $ krbEncSaltTypes $ krbTicketPolicy $ krbKeySet $ krbKeyVersion))

```
( KRBOC.2 NAME 'krbPrincipal' SUP krbKDCInfo AUXILIARY MUST ( krbPrincipalName ) MAY ( krbPrincipalAliases $
krbPrincipalRealm $ krbExtraData ) )
( KRBOC.3 NAME 'krbRealm' SUP krbKDCInfo AUXILIARY MUST ( krbRealmName ) MAY ( krbPrincNamingAttr $
krbPrincContainer $ krbPwdPolicy $ krbLDAPURI ) )
```

Détails de l'implémentation

Vu que le LDAP Password Policy est intimement lié à la sécurité de ce draft, l'annuaire devrait être traité en conséquence Cela signifie que pour toute authentification Kerberos déservis, une opération LDAP correspondante est également effectuée, pour permettre au mécanisme de stratégie de mot de passe d'opérer.

Le mécanisme dessiné ici assume que les accréditifs LDAP et les accréditifs Kerberos sont unifiés (ou au moins synchronisés). Dans ce cas, pour toute requête d'authentification Kerberos entrante, le KDC peut fournir une requête ldap compare en utilisant les accréditifs connus de l'utilisateur et le contrôle ldap Password Policy. Le résultat de la requête va s'occuper de tout code d'erreur si le compte est désactivé, le mot de passe a expiré, ou divers autres échecs. Si la pré-authentification est utilisée et que la requête est invalide, un compare avec des accréditifs invalides connus peuvent être utilisés pour mettre à jours l'état de la stratégie de mot de passe.

Détails du modèle

Un nombre d'éléments de données décris dans le modèle d'information sont délégués pour LDAP DSA pour la gestion. Les détails de leurs utilisation sont décrites ici.

principalNotUsedBefore

Correspond à l'attribut **pwdStartTime**. Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement.

principalNotUsedAfter

Correspond à l'attribut **pwdEndTime**. Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement.

principalIsDisabled

Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement. Sinon cet effet est contrôlé par le paramètrage de **pwdStartTime** à une valeur supérieur ou égale à **pwdEndTime**

principalNumberOfFailedAuthenticationAttempts

Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement. Sinon, cette valeur est obtenue en comptant le nombre de valeurs stockées dans **pwdFailureTime**

principalLastFailedAuthentication

Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement. Sinon, cette valeur est obtenue en récupérant les valeur stockées dans **pwdFailureTime** et en sélectionnant la valeur la plus récente.

principalLastSuccessfulAuthentication

Correspond à l'attribut **pwdLastSuccess**. Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement.

principalLastCredentialChangeTime

Correspond à l'attribut **pwdChangedTime** Si le KDC utilise les requêtes LDAP pour opérer le mécanisme de stratégie de mot de passe, il n'a pas besoin de référencer ou manipuler cet attribut directement.

principalCreateTime

Correspond à l'attribut **createTimestamp**. Le KDC n'a pas besoin de référencer ou manipuler cet attribut directement.

principalModifyTime

Correspond à l'attribut **modifyTimestamp**. Le KDC n'a pas besoin de référencer ou manipuler cet attribut directement.

Détail de KeySet

L'attribut **krbKeySet** est multi-valué mais il est attendu qu'il ait une seule valeur. Durant une opération de changement de mot de passe le KDC peut choisir de conserver une valeur précédente pour permettre aux clients actifs de continuer à opérer en utilisant la clé précédente. La durée de rétention de cet ancien mot de passe n'est pas spécifié ici. Noter également que le mécanisme ldap Password Policy a déja une gestion d'historique de mot de passe, donc krbKeySet ne devrait pas être utilisé pour l'historique des mots de passe.

Considérations de sécurité

Tout ce document est conserné par l'implémentation d'un mécanisme d'authentification distribué sécurisé. Il faut bien comprendre que les divers clé utilisée ici sont des données sensible et doivent être protégés en conséquence en utilisant les ACL et autres mécanismes. Également, toutes les communications entre le KDC et le DSA doivent être protégés quand des données sensible sont référencées.

En pratique, le KDC et le DSA sont hébergés sur un serveur hôte et communiquent via ldapi. Celà implique que l'hôte soit également sécurisé, et un canal sécurisé doit être utilisé pour la session LDAP.