rfc2589

Extensions pour les services d'annuaire dynamiques

LDAP support les accès aux services d'annuaire statique, permettant d'effectuer des opérations relativement rapide. Les services d'annuaire dynamiques sont différents puisqu'ils stockent des informations qui ont une durée de vie. un cas typique est un client ou une personne qui est soit online, dans ce cas il a une entrée dans l'annuaire, soit offline et dans ce cas l'entrée disparait. Bien que les opérations du protocole et les attributs sont utilisés de la même manière que pour les services d'annuaires statiques, les clients qui stockent ces informations dans l'annuaire doivent périodiquement rafraîchir ces informations. Les entrées qui n'ont pas été rafraîchies après une période données sont supprimés par le serveur. Un mécanisme de contrôle de flux du serveur est aussi décrit et permet au serveur d'informer les clients de la fréquence de rafraîchissement.

Prérequis

Les extension de protocole doivent permettre d'accéder aux informations dynamiques dans un annuaire de manière standard.

Entrées dynamiques et classes d'objet

Une entrée dynamique est un objet dans l'annuaire qui a une durée de vie associée avec lui. Cette durée de vie est définie lors de la création de l'objet et quand il expire, l'objet est supprimé. En invoquant l'opération étendue refresh, le TTL est réinitialisé.

Requête de rafraichissement

Cette opération envoyée par un client indique au serveur de conserver l'entrée dynamique et de réinitialiser sa durée de vie. Un client peut demander cette opération en transmettant un PDU contenant un ExtendedRequest :

```
ExtendedRequest ::= [APPLICATION 23] SEQUENCE {
 requestName [0] LDAPOID,
 requestValue [1] OCTET STRING OPTIONAL
 }
```

requestName doit être définis à : **"1.3.6.1.4.1.1466.101.119.1"**

requestValue va contenir le DER du type ASN.1 suivant :

```
SEQUENCE {
 entryName [0] LDAPDN,
 requestTtl [1] INTEGER
 }
```

entryName est le nom UTF8 de l'entrée dynamique. Cette entrée doit déjà exister.

requestTtl est le temps en secondes (1 à 31557600) du nouveau TTL à définir.

Réponse au rafraichissement

Si un serveur implémente cette extension, le serveur doit répondre un PDU contenant ExtendedResponse :

```
ExtendedResponse ::= [APPLICATION 24] SEQUENCE {
 COMPONENTS OF LDAPResult,
 responseName [10] LDAPOID OPTIONAL,
 response [11] OCTET STRING OPTIONAL
 }
```

responseName contient la même chaîne que dans la demande. Le champ response va contenir un DER de la sequence ASN.1 suivante :

```
SEQUENCE {
 responseTtl [1] INTEGER
 }
```

Le champ **responseTtl** est le temps en secondes que le serveur a choisi comme champ ttl pour cette entrée. Il ne doit pas être plus petit que le choix du client, mais il peut être plus grand. Cependant, pour éviter tout abus, les serveurs sont autorisés à raccourcir le TTL d'un client à un minimumde 86400 secondes.

Schéma

dynamicObject Cette classe définis une entrée dynamique

entryTtl Maintient le TTL de l'entrée

dynamicSubtrees dans le rootDSE, maintient les sous-arborescences où sont supportée les entrées dynamiques.

```
( 1.3.6.1.4.1.1466.101.119.2 NAME 'dynamicObject'
DESC 'This class, if present in an entry, indicates that this entry has a limited lifetime and may disappear
automatically when its time-to-live has reached 0. There are no mandatory attributes of this class, however
if the client has not supplied a value for the entryTtl attribute, the server will provide one.'
SUP top AUXILIARY )

( 1.3.6.1.4.1.1466.101.119.3 NAME 'entryTtl'
DESC 'This operational attribute is maintained by the server and appears to be present in every dynamic
entry. The attribute is not present when the entry does not contain the dynamicObject object class. The value
of this attribute is the time in seconds that the entry will continue to exist before disappearing from the
directory. In the absence of intervening refresh operations, the values returned by reading the attribute in
two successive searches are guaranteed to be nonincreasing. The smallest permissible value is 0, indicating
that the entry may disappear without warning. The attribute is marked NO-USER-MODIFICATION since it may only
be changed using the refresh operation.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE
NO-USER-MODIFICATION USAGE dSAOperation )

( 1.3.6.1.4.1.1466.101.119.4 NAME 'dynamicSubtrees'
DESC 'This operational attribute is maintained by the server and is present in the Root DSE, if the server
supports the dynamic extensions described in this memo. The attribute contains a list of all the subtrees in
this directory for which the server supports the dynamic extensions.'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 NO-USER-MODIFICATION
USAGE dSAOperation )
```

Prérequis client

Les clients peuvent vérifier si le serveur supporte les extensions dynamiques en vérifiant le champs **supportedExtension** dans le RootDSE. Les client doivent vérifier le **dynamicSubtrees** pour vérifier si les extensions dynamiques sont supportée sur une arborescence spécifique.

Les client ne doivent pas s'attendre à ce qu'une entrée soit présente après que le TTL soit dépassé. Toutefois, les client ne doivent pas assumer que l'objet sera supprimé immédiatement après la durée de vie atteinte.

le CRP (Client Refresh Period) est définis par le serveur, basé sur le entryTtl. Une fois une opération AddRequest effectuée, le client devrait immédiatement envoyer une opération étendue refresh pour récupérer le CRP dans la responseTtl.

Les client ne doivent pas demander des refresh sur des entrées qui n'existent pas et devraient toujours être prêt à manipuler des cas d'entrées expirées. Les clients devraient également être préparés à des opération de refresh qui échouent (ex : un proxy down).

Prérequis serveur

Les serveur sont responsables de la suppression des objets expirés, mais il n'est pas requis qu'ils le fasse immédiatement.