

Politique des mots de passe dans OpenLDAP 2.3



Document Technique LINAGORA

Référence : DTL-OpenLDAP-Ppolicy

Document créé le jeudi 13 mars 2008

Groupe LINAGORA

27, rue de Berri – 75008 PARIS

<http://www.linagora.com>

Tél : 01 58 18 68 28 – Fax : 01 58 18 68 29

N° SIRET : 431 473 669 0056

SOMMAIRE

1. Normes et standards de la politique des mots de passe.....	4
1.1. <i>Brouillon de la RFC.....</i>	4
1.1.1. Avertissement.....	4
1.1.2. Contenu du document.....	4
1.2. <i>Implémentation dans OpenLDAP 2.3.....</i>	4
1.2.1. Configuration de l'overlay.....	4
1.2.2. Affichage du contrôle dans le RootDSE.....	5
1.2.3. Schéma.....	5
2. Contenu du brouillon de la RFC.....	6
2.1. <i>Conventions et champ d'application.....</i>	6
2.2. <i>Fonctionnalités de la politique des mots de passe.....</i>	6
2.2.1. Limitation des authentifications infructueuses.....	6
2.2.2. Expiration du mot de passe.....	6
2.2.3. Historique du mot de passe.....	7
2.2.4. Âge minimum du mot de passe.....	7
2.2.5. Qualité et taille minimale du mot de passe.....	7
2.2.6. Modification du mot de passe par l'utilisateur.....	7
2.2.7. Modification du mot de passe après sa réinitialisation.....	7
2.2.8. Modification sécurisée.....	7
2.3. <i>Schéma.....</i>	8
2.3.1. Classe d'objet.....	8
2.3.2. Attributs de la politique.....	8
2.3.3. Attributs opérationnels des utilisateurs.....	11
2.4. <i>Utilisation des contrôles.....</i>	13
2.5. <i>Réplication.....</i>	13

CONVENTIONS DE NOTATION

Les contenus de fichiers sont retranscrits ainsi :

```
# Ceci est un fichier  
parametre="valeur"
```

Les commandes sont indiquées sous la forme suivante et précédées d'un caractère '\$' (commande à taper avec des droits utilisateurs) ou '#' (commande à taper avec des droits root) qui n'est pas à saisir dans la ligne de commande :

```
$ commande1  
# commande2
```

Les remarques ou autres éléments particuliers sont notés dans les cadres suivants :

Style des remarques

1. NORMES ET STANDARDS DE LA POLITIQUE DES MOTS DE PASSE

1.1. Brouillon de la RFC

1.1.1. Avertissement

La dernière version du document est la version 9, datant du 17 juillet 2005. Elle est disponible sur le site suivant : <http://www.faqs.org/ftp/pub/internet-drafts/draft-behera-ldap-password-policy-09.txt>.

Comme son nom l'indique, ce document n'est pas une RFC¹ à part entière, mais un brouillon (draft en anglais), dont la durée de vie au sein de l'IETF² est de 6 mois maximum. Ainsi depuis le 18 janvier 2006; ce document a expiré et n'est plus disponible sur le site de l'IETF. Aucune nouvelle version du brouillon n'a vu le jour et la RFC n'est pas sortie. Cela signifie donc que la norme décrite dans ce document n'a aucune valeur de standard et que les préconisations de cette étude devront être remises en cause à la sortie d'une prochaine version du brouillon ou de la RFC finale.

1.1.2. Contenu du document

Étant donnée la taille du document, cette partie a été placée en annexe (cf. chapitre 2, page 6). Les chapitres suivants ne traitent que des différences entre l'implémentation dans OpenLDAP et les recommandations du brouillon de la RFC.

1.2. Implémentation dans OpenLDAP 2.3

1.2.1. Configuration de l'overlay

La politique des mots de passe est implémentée sous forme d'overlay dans OpenLDAP. Ce morceau de programme peut être greffé à n'importe quel backend (BDB, HDB, LDAP, etc.), voire se trouver dans la configuration globale et agir sur tous les backends. Il n'est pas compilé par défaut, et l'option `--enable-ppolicy` doit donc être indiquée lors de la phase de configuration de la compilation (exécution de `./configure`).

Une fois installé, l'overlay peut être paramétré dans le fichier standard de configuration, `slapd.conf` :

```
[...]
database                                bdb
[...]
overlay                                  ppolicy
ppolicy_default                          ou=standard,ou=securite,dc=linagora,dc=com
ppolicy_hash_cleartext
ppolicy_use_lockout
[...]
```

Signification des paramètres :

- **ppolicy_default** : indique l'entrée de configuration par défaut. Cette valeur sera écrasé par celle de l'attribut opérationnel `pwdPolicySubentry` du compte d'un utilisateur.
- **ppolicy_hash_cleartext** : permet de chiffrer automatiquement les mots de passe (en SSHA) lors des opérations `add` ou `modify`. Cela contredit toutefois le modèle X.500/LDAP qui impose que les mots de passe soient en clair dans l'annuaire.
- **ppolicy_use_lockout** : par défaut un client reçoit toujours le message d'erreur `invalidCredentials`

1 Request For Comments

2 Internet Engineering Task Force

lorsqu'il se connecte à un compte bloqué. Si le client utilise le contrôle de requête de la politique des mots de passe, alors il obtiendra en plus l'information que le compte est bloqué ce qui peut avantager les attaquants (ils savent que le compte existe et qu'il est bloqué). Cette option doit être désactivée sur les sites sensibles.

1.2.2. Affichage du contrôle dans le RootDSE

Par défaut les contrôles supportés par un annuaire sont affichés dans le RootDSE. Cependant, étant donné que la politique des mots de passe n'est pas standardisée, OpenLDAP a fait le choix de ne pas afficher le contrôle, bien qu'il soit utilisable lorsque l'overlay est activé. Une requête sur les contrôles du RootDSE donnera donc :

```
dn:  
supportedControl: 1.3.6.1.4.1.4203.1.9.1.1  
supportedControl: 2.16.840.1.113730.3.4.18  
supportedControl: 2.16.840.1.113730.3.4.2  
supportedControl: 1.3.6.1.4.1.4203.1.10.1  
supportedControl: 1.2.840.113556.1.4.319  
supportedControl: 1.2.826.0.1.334810.2.3  
supportedControl: 1.2.826.0.1.3344810.2.3  
supportedControl: 1.3.6.1.1.13.2  
supportedControl: 1.3.6.1.1.13.1  
supportedControl: 1.3.6.1.1.12
```

On note bien que le contrôle 1.3.6.1.4.1.42.2.27.8.5.1 n'apparaît pas.

1.2.3. Schéma

Le schéma décrit dans le brouillon de la RFC est implémenté à plusieurs endroits :

- la classe d'objet et les attributs permettant le paramétrage de la politique sont dans un fichier texte classique, nommé *ppolicy.schema*, à ajouter aux autres schémas dans *slapd.conf*,
- les attributs opérationnels utilisés dans les comptes sont inscrits en dur dans le code de l'overlay (*ppolicy.c*).

On peut toutefois noter quelques différences entre le schéma d'OpenLDAP et celui du brouillon de la RFC :

- Une nouvelle classe d'objet, nommée *pwdPolicyChecker*, contenant un unique attribut facultatif, nommé *pwdCheckModule*, qui permet de déclarer des modules de vérification de la qualité des mots de passe.
- L'attribut opérationnel *pwdPolicySubentry* peut être modifié par un utilisateur ou un administrateur, ce qui permet de déclarer une autre politique associée au compte.
- L'attribut opérationnel *pwdAccountLockedTime* peut être modifié par un utilisateur ou un administrateur, ce qui permet de déverrouiller manuellement un compte.

Remarque : dans OpenLDAP 2.4, un nouveau contrôle nommé « *relax* » permettra d'effectuer des opérations modifiant la structure de l'annuaire, comme par exemple écrire sur les attributs opérationnels. Les deux dérogations précédentes sauteront donc vraisemblablement et il faudra adapter les transactions en conséquence.

2. CONTENU DU BROUILLON DE LA RFC

2.1. Conventions et champ d'application

Le brouillon définit deux comptes privilégiés :

- L'administrateur des mots de passe : il a le droit de modifier les mots de passe des utilisateurs.
- L'administrateur de la politique des mots de passe : il a le droit de modifier les paramètres de la politique des mots de passe.

Le terme **utilisateur** définit une personne physique ou une application cliente de l'annuaire possédant un compte avec un mot de passe. Il est possible de déclarer un ou plusieurs utilisateurs qui ne seront pas soumis à la politique des mots de passe.

La politique des mots de passe peut s'appliquer sur n'importe quel attribut contenant un mot de passe, celui-ci doit néanmoins ne posséder qu'une seule valeur. Il est possible par exemple pour Samba de déclarer une politique sur les attributs *sambaLMPassword* et *sambaNTPassword*.

***Attention** : l'attribut standard **userPassword** est multivalué dans le schéma par défaut, c'est donc à l'annuaire implémentant la politique des mots de passe de s'assurer que la valeur est bien unique.*

La politique s'applique également aux types d'authentifications suivants :

- Authentification simple avec mot de passe en clair (opération *bind*).
- Authentification SASL basée sur un mot de passe (par exemple : *CRAM-MD5*, *DIGEST-MD5*).

Il est possible de définir plusieurs politiques différentes, mais un utilisateur donné ne sera soumis qu'à une seule de ces politiques.

Toutes les dates sont au format AAAAMMJJhhmmssZ, soit 20061011125443Z pour le 11 octobre 2006 à 12h54m43s, heure de Greenwich.

2.2. Fonctionnalités de la politique des mots de passe

2.2.1. Limitation des authentifications infructueuses

L'objectif est d'empêcher un attaquant de deviner un mot de passe par l'envoi de mots de passe générés automatiquement, à partir d'un dictionnaire par exemple. Cette politique contient cinq éléments :

- Définition d'une limite de tentatives d'authentification infructueuses.
- Un compteur référençant les tentatives infructueuses.
- Un intervalle de temps dans lequel ces authentifications infructueuses doivent avoir lieu.
- Une action à entreprendre lors de l'atteinte de cette limite (verrouillage du compte).
- Un temps de verrouillage du compte.

2.2.2. Expiration du mot de passe

Si un mot de passe est changé régulièrement, il aura moins de chance d'être cassé par un attaquant. Il est donc possible de donner une durée de vie à un mot de passe, ce qui forcera les utilisateurs à le changer. Le principal problème est d'avertir l'utilisateur de cette date d'expiration, pour cela deux moyens :

- Un avertissement est envoyé à l'utilisateur dans un laps de temps précédent l'expiration, et le compte est bloqué à l'expiration du mot de passe.
- Un nombre d'authentification « de grâce » est défini, qui permet à l'utilisateur de se connecter même si le mot de passe est expiré. Un avertissement est alors envoyé et un compteur recense le nombre

d'authentifications en grâce. Une fois toutes ces authentifications consommées, le compte est bloqué.

La date d'expiration est calculée en associant la date de dernier changement de mot de passe avec la durée de vie d'un mot de passe.

2.2.3. Historique du mot de passe

L'historique permet d'éviter qu'un utilisateur utilise toujours les mêmes mots de passe. Pour cela, un nombre maximum de mots de passe dans l'historique est donné, et les mots de passe sont stockés à l'intérieur quand ils sont changés. Si le nombre de mots de passe stockés est plus grand que la taille limite, les mots de passes les plus anciens sont supprimés de l'historique. Un utilisateur ne peut changer son mot de passe par un mot de passe déjà présent dans l'historique.

2.2.4. Âge minimum du mot de passe

L'âge minimum empêche un utilisateur de changer consécutivement plusieurs fois son mot de passe (pour faire le tour de son historique). Ainsi un âge minimal du mot de passe de 24h n'autorisera un utilisateur à ne changer son mot de passe qu'une fois par jour.

2.2.5. Qualité et taille minimale du mot de passe

L'objectif est de forcer un utilisateur à créer un mot de passe non trivial. Pour cela, les méthodes suivantes peuvent être utilisées :

- Vérification auprès d'un dictionnaire, pour empêcher les termes génériques.
- Obligation d'avoir des chiffres dans le mot de passe.
- Interdire les anagrammes du prénom ou du nom de l'utilisateur.
- Forcer la taille minimale du mot de passe.

Cependant l'implémentation de cette politique pose les problèmes suivants :

- Si le mot de passe est chiffré par le client avant son envoi à l'annuaire, les vérifications ne peuvent se faire.
- Il n'y a pas de définition exacte de la signification de « vérification de la qualité ». Cela peut poser problème dans des environnements hétérogènes, car un mot de passe considéré de bonne qualité sur un système pourra être refusé sur un autre.

2.2.6. Modification du mot de passe par l'utilisateur

Dans certains cas il est souhaitable d'interdire aux utilisateurs de modifier leur mot de passe. Cette fonctionnalité fait partie de la politique des mots de passe.

2.2.7. Modification du mot de passe après sa réinitialisation

Il est possible d'obliger un utilisateur à changer son mot de passe après les événements suivants :

- Première création du mot de passe.
- Réinitialisation du mot de passe par un administrateur.

Cela évite qu'un mot de passe par défaut ait une durée de vie équivalente à un mot de passe classique.

2.2.8. Modification sécurisée

Il arrive qu'un utilisateur établisse une connexion à l'annuaire et la laisse ouverte, avec ses droits. Il est alors possible pour un attaquant d'utiliser la connexion et de changer le mot de passe à l'insu de l'utilisateur. Pour prévenir ce risque, il peut être demandé d'envoyer l'ancien mot de passe en même temps que le nouveau dans l'opération de modification.

2.3. Schéma

2.3.1. Classe d'objet

La classe d'objet suivante représente une entrée de configuration de la politique des mots de passe. Elle n'est pas appliquée à un compte utilisateur.

```
( 1.3.6.1.4.1.42.2.27.8.2.1
NAME 'pwdPolicy'
SUP top
AUXILIARY
MUST ( pwdAttribute )
MAY ( pwdMinAge $ pwdMaxAge $ pwdInHistory $ pwdCheckQuality $
pwdMinLength $ pwdExpireWarning $ pwdGraceAuthNLimit $ pwdLockout
$ pwdLockoutDuration $ pwdMaxFailure $ pwdFailureCountInterval $
pwdMustChange $ pwdAllowUserChange $ pwdSafeModify ) )
```

2.3.2. Attributs de la politique

pwdAttribute : nom de l'attribut auquel est appliquée la politique des mots de passe. Par exemple : *userPassword*.

```
( 1.3.6.1.4.1.42.2.27.8.1.1
NAME 'pwdAttribute'
EQUALITY objectIdentifierMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.38 )
```

pwdMinAge : durée de vie minimale d'un mot de passe, en secondes. L'absence de l'attribut équivaut à la valeur 0.

```
( 1.3.6.1.4.1.42.2.27.8.1.2
NAME 'pwdMinAge'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
```

pwdMaxAge : durée de vie maximale d'un mot de passe. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie que le mot de passe n'expire jamais. Si différente de 0, la valeur doit être supérieure à celle de *pwdMinAge*.

```
( 1.3.6.1.4.1.42.2.27.8.1.3
NAME 'pwdMaxAge'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
```

pwdInHistory : nombre de mots de passe dans l'historique. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie que les mots de passe ne sont pas stockés dans l'historique.

```
( 1.3.6.1.4.1.42.2.27.8.1.4
NAME 'pwdInHistory'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
```

```
SINGLE-VALUE )
```

pwdCheckQuality : indique si la qualité du mot de passe (et donc sa taille minimale) doit être vérifiée. Les valeurs suivantes sont possibles :

- **0** : la qualité n'est pas vérifiée (ce qui est aussi le cas si l'attribut n'existe pas).
- **1** : la qualité est vérifiée, mais si l'annuaire n'a pas les moyens de le faire (mot de passe chiffré), il accepte le mot de passe.
- **2** : la qualité est vérifiée, mais si l'annuaire n'a pas les moyens de la faire, il rejette le mot de passe.

```
( 1.3.6.1.4.1.42.2.27.8.1.5  
NAME 'pwdCheckQuality'  
EQUALITY integerMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
SINGLE-VALUE )
```

pwdMinLength : taille minimale du mot de passe. L'absence de l'attribut équivaut à la valeur 0.

```
( 1.3.6.1.4.1.42.2.27.8.1.6  
NAME 'pwdMinLength'  
EQUALITY integerMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
SINGLE-VALUE )
```

pwdExpireWarning : nombre de secondes précédant l'expiration d'un mot de passe pendant lesquelles une alerte sera envoyée à un utilisateur faisant une authentification. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie qu'aucune alerte ne sera envoyée. Si différente de 0, la valeur doit être inférieure à celle de *pwdMaxAge*.

```
( 1.3.6.1.4.1.42.2.27.8.1.7  
NAME 'pwdExpireWarning'  
EQUALITY integerMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
SINGLE-VALUE )
```

pwdGraceAuthNLimit : nombre de fois où un mot de passe expiré peut être utilisé (c'est le temps de grâce). L'absence de l'attribut équivaut à la valeur 0, ce qui signifie qu'aucune connexion n'est possible après expiration du mot de passe.

```
( 1.3.6.1.4.1.42.2.27.8.1.8  
NAME 'pwdGraceAuthNLimit'  
EQUALITY integerMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27  
SINGLE-VALUE )
```

pwdLockout : si la valeur est « TRUE », le compte sera bloqué si la limite des tentatives d'authentification infructueuses est dépassée. L'absence de l'attribut équivaut à la valeur « FALSE », ce qui signifie que le compte ne sera pas bloqué.

```
( 1.3.6.1.4.1.42.2.27.8.1.9  
NAME 'pwdLockout'  
EQUALITY booleanMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE )
```

pwdLockoutDuration : nombre de secondes pendant lequel le compte est bloqué. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie que le compte ne sera pas débloqué sans intervention d'un administrateur.

```
( 1.3.6.1.4.1.42.2.27.8.1.10
NAME 'pwdLockoutDuration'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
```

pwdMaxFailure : nombre maximal de tentatives d'authentification infructueuses. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie que les tentatives ne seront pas comptées, et ce qui annule la valeur de l'attribut *pwdLockout*.

```
( 1.3.6.1.4.1.42.2.27.8.1.11
NAME 'pwdMaxFailure'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
```

pwdFailureCountInterval : nombre de secondes au bout desquelles le compteur d'authentifications infructueuses est vidé, même si aucune authentification valide n'a eu lieu. L'absence de l'attribut équivaut à la valeur 0, ce qui signifie que le compteur ne sera vidé que si une authentification valide a lieu.

```
( 1.3.6.1.4.1.42.2.27.8.1.12
NAME 'pwdFailureCountInterval'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
SINGLE-VALUE )
```

pwdMustChange : si la valeur est « TRUE », un utilisateur devra changer son mot de passe dès que celui-ci est initialisé ou réinitialisé par un administrateur. L'absence de l'attribut équivaut à la valeur « FALSE ».

```
( 1.3.6.1.4.1.42.2.27.8.1.13
NAME 'pwdMustChange'
EQUALITY booleanMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE )
```

pwdAllowUserChange : détermine si un utilisateur peut changer son mot de passe. L'absence de l'attribut équivaut à la valeur « TRUE ».

```
( 1.3.6.1.4.1.42.2.27.8.1.14
NAME 'pwdAllowUserChange'
EQUALITY booleanMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE )
```

pwdSafeModify : indique si l'ancien mot de passe doit être envoyé avec le nouveau lors d'un changement de mot de passe. L'absence de l'attribut équivaut à la valeur « FALSE ».

```
( 1.3.6.1.4.1.42.2.27.8.1.15
NAME 'pwdSafeModify'
EQUALITY booleanMatch
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7  
SINGLE-VALUE )
```

2.3.3. Attributs opérationnels des utilisateurs

Ces attributs permettent de maintenir l'état de la politique des mots de passe pour chaque utilisateur. Ce sont des attributs opérationnels stockés dans l'entrée de l'utilisateur qui ne sont pas modifiables par un utilisateur ou un administrateur : seul l'annuaire intervient sur ces données.

Comme la politique peut porter sur plusieurs attributs, il est possible de différencier les attributs opérationnels grâce à une option définissant un sous-type. Cette option est de la forme `pwd-<attributMotDePasse>` et se place suite à l'attribut opérationnel, comme par exemple :

```
pwdChangedTime;pwd-userPassword : 20061010125443Z
```

Ainsi on trouve comme valeur de `pwdChangedTime` pour l'attribut `userPassword` la date 20061010125443Z.

pwdChangedTime : date de dernier changement de mot de passe. Il est utilisé pour calculer la date d'expiration du mot de passe, s'il n'existe pas, le mot de passe n'expire jamais.

```
( 1.3.6.1.4.1.42.2.27.8.1.16  
NAME 'pwdChangedTime'  
DESC 'The time the password was last changed'  
EQUALITY generalizedTimeMatch  
ORDERING generalizedTimeOrderingMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24  
SINGLE-VALUE  
NO-USER-MODIFICATION  
USAGE directoryOperation )
```

pwdAccountLockedTime : date à laquelle le compte a été bloqué. Cet attribut n'existe pas si le compte est actif. Une valeur de 0000101000000Z signifie que le compte a été bloqué pour une durée indéfinie et qu'il ne pourra être activé de nouveau que par l'intervention d'un administrateur.

```
( 1.3.6.1.4.1.42.2.27.8.1.17  
NAME 'pwdAccountLockedTime'  
DESC 'The time an user account was locked'  
EQUALITY generalizedTimeMatch  
ORDERING generalizedTimeOrderingMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24  
SINGLE-VALUE  
NO-USER-MODIFICATION  
USAGE directoryOperation )
```

pwdFailureTime : dates des dernières authentifications infructueuses. Cet attribut est multivalué, les valeurs représentant les tentatives les plus récentes. Le nombre de valeurs maximum est fixé par le paramètre `pwdMaxFailure`.

```
( 1.3.6.1.4.1.42.2.27.8.1.19  
NAME 'pwdFailureTime'  
DESC 'The timestamps of the last consecutive authentication failures'  
EQUALITY generalizedTimeMatch  
ORDERING generalizedTimeOrderingMatch  
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
```

```
NO-USER-MODIFICATION
USAGE directoryOperation )
```

pwdHistory : historique des mots de passe utilisés. Cet attribut est multivalué et chacune des valeurs est de la forme suivante :

```
time#syntaxOID#length#data
```

Avec :

- **time** : date d'enregistrement du mot de passe dans l'historique.
- **syntaxOID** : OID représentant la syntaxe utilisée pour stocker le mot de passe (par exemple : 1.3.6.1.4.1.1466.115.121.1.40 si c'est une chaîne de caractères).
- **length** : taille en nombre d'octets du mot de passe.
- **data** : le mot de passe, dans le format fixé par la valeur de *syntaxOID*.

```
( 1.3.6.1.4.1.42.2.27.8.1.20
NAME 'pwdHistory'
DESC 'The history of user s passwords'
EQUALITY octetStringMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.40
NO-USER-MODIFICATION
USAGE directoryOperation )
```

pwdGraceUseTime : dates des authentifications « en grâce » après expiration du mot de passe. Cet attribut est multivalué, et le nombre maximum de valeurs est fixé par le paramètre *pwdGraceAuthNLimit*.

```
( 1.3.6.1.4.1.42.2.27.8.1.21
NAME 'pwdGraceUseTime'
DESC 'The timestamps of the grace authentication after the password has
expired'
EQUALITY generalizedTimeMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
NO-USER-MODIFICATION
USAGE directoryOperation )
```

pwdReset : drapeau (booléen « TRUE » ou « FALSE ») indiquant si le mot de passe a été mis à jour par un administrateur ou bien si l'utilisateur doit le changer à la prochaine connexion.

```
( 1.3.6.1.4.1.42.2.27.8.1.22
NAME 'pwdReset'
DESC 'The indication that the password has been reset'
EQUALITY booleanMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
SINGLE-VALUE
USAGE directoryOperation )
```

pwdPolicySubentry : DN de l'entrée de configuration de la politique active pour ce compte.

```
( 1.3.6.1.4.1.42.2.27.8.1.23
NAME 'pwdPolicySubentry'
DESC 'The pwdPolicy subentry in effect for this object'
EQUALITY distinguishedNameMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12
SINGLE-VALUE
```

```
NO-USER-MODIFICATION
USAGE directoryOperation )
```

2.4. Utilisation des contrôles

Les contrôles sont des paramètres supplémentaires pouvant être passés aux opérations de base définies dans le standard LDAP. On distingue :

- le contrôle de requête, qui est émis par le client ;
- le contrôle de réponse, qui est émis par le serveur.

Un contrôle est composé de :

- un type représenté par un OID. Cet OID est généralement publié sur le RootDSE par annoncer le support du contrôle par l'annuaire,
- une criticité, qui est représentée par un booléen (« TRUE » ou « FALSE »),
- une valeur, qui peut ne pas exister.

Dans le cas de la politique des mots de passe, le type du contrôle est 1.3.6.1.4.1.42.2.27.8.5.1 et la criticité est laissée à la discrétion du client ou de l'annuaire.

Pour le contrôle de requête, il n'y a pas de valeur, son rôle est d'avertir l'annuaire que le client est compatible avec la politique et qu'il saura analyser le contrôle de réponse. Pour le contrôle de réponse, la valeur est de la forme :

```
PasswordPolicyResponseValue ::= SEQUENCE {
    warning [0] CHOICE {
        timeBeforeExpiration [0] INTEGER (0 .. maxInt),
        graceAuthNsRemaining [1] INTEGER (0 .. maxInt) } OPTIONAL,
    error [1] ENUMERATED {
        passwordExpired (0),
        accountLocked (1),
        changeAfterReset (2),
        passwordModNotAllowed (3),
        mustSupplyOldPassword (4),
        insufficientPasswordQuality (5),
        passwordTooShort (6),
        passwordTooYoung (7),
        passwordInHistory (8) } OPTIONAL }
```

Ainsi le contrôle de réponse permet à un client évolué d'analyser les erreurs possibles dues à l'application de la politique des mots de passe. La vocation de ce contrôle est d'être utilisé dans les transactions d'administration pour permettre d'exposer ces informations, car il est rare qu'un simple utilisateur soit capable de recevoir et de décoder ces valeurs. L'absence de contrôle n'empêche pas la politique de s'appliquer : toute connexion à l'annuaire est soumise aux mêmes contraintes, toutefois sans contrôle, les messages d'erreurs seront moins explicites.

2.5. Réplication

L'application de la politique des mots de passe peut se compliquer lors de son utilisation sur plusieurs annuaires répliqués entre eux :

- Les entrées représentant les paramètres de la politique des mots de passe doivent être répliquées sur tous les annuaires afin d'assurer l'homogénéité de la configuration. La modification de la configuration doit avoir lieu sur l'annuaire maître et être ensuite propagée. Cela est assuré par défaut par la réplication car ces données sont des valeurs d'attributs standard.

- Les attributs opérationnels des comptes doivent être également répliqués SAUF ceux qui sont modifiés même par les annuaires esclaves, c'est-à-dire *pwdAccountLockedTime*, *pwdFailureTime* et *pwdGraceUseTime*. Ces trois attributs seront maintenus de manière indépendante sur chaque annuaire, ce qui fait que le nombre effectif d'authentifications infructueuses sera égal à la limite des authentifications infructueuses multipliée par le nombre d'annuaires. Ainsi pour une limite de 10 sur 3 annuaires, la limite effective sera de 30. Ainsi les seuls attributs opérationnels répliqués sont *pwdChangedTime*, *pwdHistory*, *pwdReset* et *pwdPolicySubentry*.

***Remarque** : dans le cas d'une architecture multimaîtres, ce sont les annuaires qui doivent s'assurer de la cohérence des valeurs de ces attributs dans les différents annuaires.*