Groupe de travail Réseau
Request for Comments : 3112
Catégorie : Information

K. Zeilenga, OpenLDAP Foundation
mai 2001
Traduction Claude Brière de L'Isle

Schéma d'authentification LDAP par mot de passe

Statut de ce mémoire
Le présent mémoire apporte des informations pour la communauté de l'Internet. Le présent mémoire ne spécifie aucune sorte de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document décrit un schéma pour la prise en charge de l'authentification par utilisateur/mot de passe dans un répertoire du protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*) incluant le type d'attribut authPassword. Ce type d'attribut contient des valeurs déduites du ou des mots de passe de l'utilisateur (généralement en utilisant un hachage unidirectionnel de force cryptographique). authPassword est destiné à être utilisé à la place de userPassword.

Table des matières

1. Fondements et utilisation prévue

Le type d'attribut userPassword *(mot de passe d'utilisateur)* [RFC2256] est destiné à être utilisé pour prendre en charge l'opération LDAP bind "simple" [RFC2251]. Cependant, les valeurs de userPassword doivent être des mots de passe de texte en clair. Il est souvent désirable de mémoriser des valeurs dérivées du ou des mots de passe d'utilisateur plutôt que les mots de passe réels.

Le type d'attribut authPassword est destiné à être utilisé pour mémoriser les informations utilisées pour mettre en œuvre l'authentification fondée sur un simple mot de passe. Le type d'attribut peut être utilisé par les serveurs LDAP pour mettre en œuvre la méthode d'authentification "simple" de l'opération Bind de LDAP.

Le type d'attribut supporte plusieurs schémas de mémorisation. Une règle de correspondance est fournie pour être utilisée avec des filtres de recherche extensibles afin de permettre aux clients d'affirmer qu'un mot de passe de texte en clair "correspond" à une des valeurs de l'attribut.

Les schémas de mémorisation utilisent souvent un hachage unidirectionnel de force cryptographique. Bien que l'utilisation d'un hachage unidirectionnel réduise le potentiel que des valeurs exposées permettent un accès non autorisé au répertoire

(sauf si l'algorithme/mise en œuvre de hachage est fautif) le hachage des mots de passe est destiné à être comme une couche de protection supplémentaire. Il est RECOMMANDÉ que des valeurs hachées soient protégées comme si elles étaient des mots de passe en clair.

Cet attribut peut être utilisé en conjonction avec des mécanismes de génération de mot de passe du côté du serveur (comme l'opération étendue de modification du mot de passe LDAP de la [RFC3062]).

L'accès à cet attribut peut être gouverné par des commandes administratives telles que celles qui mettent en œuvre les politiques de changement de mot de passe.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Définitions de schéma

Les définitions de schéma suivantes sont décrites en termes de définition de syntaxe d'attribut LDAPv3 [RFC2252] avec la syntaxe spécifique qui utilise le BNF augmenté de la [RFC2234].

2.1 Syntaxe de authPassword

(1.3.6.1.4.1.4203.1.1.2 DESC "syntaxe d'authentification par mot de passe")

Les valeurs de cette syntaxe sont codées conformément à :

```
authPasswordValue = w scheme s authInfo s authValue w
scheme = %x30-39 / %x41-5A / %x2D-2F / %x5F                ; 0 à 9, A à Z, "-", ".", "/", ou "_"
authInfo = schemeSpecificValue
authValue = schemeSpecificValue
    schemeSpecificValue = *( %x21-23 / %x25-7E )           ; ASCII imprimable moins "$" et " "
s = w SEP w
w = *SP
SEP = %x24                                                 ; "$"
SP = %x20                                                  ; " " (espace)
```

où scheme décrit le mécanisme et authInfo et authValue sont un schéma spécifique. Le champ authInfo est souvent un sel codé en base64. Le champ authValue est souvent une valeur codée en base64 déduite d'un ou de mots de passe d'un utilisateur. Les valeurs de cet attribut sont sensibles à la casse.

Le transfert de valeurs de cette syntaxe est fortement déconseillé lorsque le service de transport sous-jacent ne peut pas garantir la confidentialité et peut résulter en la divulgation des valeurs à des parties non autorisées.

Le présent document décrit un certain nombre de schémas, ainsi que les exigences pour la dénomination de schéma, à la Section 3.

2.2 authPasswordExactMatch

```
( 1.3.6.1.4.1.4203.1.2.2 NOM 'authPasswordExactMatch'
    DESC "règle de correspondance exacte d'authentification par mot de passe"
    SYNTAXE 1.3.6.1.4.1.4203.1.1.2 )
```

Cette règle de correspondance permet à un client d'affirmer qu'une valeur affirmée de authPasswordSyntax correspond aux valeurs de authPasswordSyntax. Elle est destinée à être utilisée comme règle de correspondance EQUALITY des attributs dont la SYNTAXE est authPasswordSyntax.

L'assertion est "VRAI" si il y a une valeur d'attribut qui a les mêmes composants scheme, authInfo, et authValue que la valeur affirmée ; "FAUX" si aucune valeur d'attribut n'a les mêmes composants que la valeur affirmée ; et "Indéfinie" autrement.

2.3 authPasswordMatch

(1.3.6.1.4.1.4203.1.2.3 NOM 'authPasswordMatch'
DESC "règle de correspondance d'authentification par mot de passe"
SYNTAXE 1.3.6.1.4.1.1466.115.121.1.40{128})

Cette règle de correspondance permet à un client d'affirmer qu'un mot de passe correspond aux valeurs de authPasswordSyntax en utilisant un composant de filtre extensibleMatch. Chaque valeur est confrontée selon ce schéma. L'assertion est "VRAI" si une ou plusieurs valeurs d'attribut correspondent à la valeur affirmée, "FAUX" si aucune valeur ne correspond, et "Indéfini" autrement.

Les serveurs qui prennent en charge l'utilisation des règles de correspondance DEVRAIENT publier les valeurs appropriées de matchingRuleUse selon le paragraphe 4.4 de la [RFC2252].

Le transfert de valeurs de l'assertion authPasswordMatch est fortement déconseillé lorsque le service de transport sous-jacent ne peut pas garantir la confidentialité et peut résulter en la divulgation des valeurs à des parties non autorisées.

2.4 supportedAuthPasswordSchemes

(1.3.6.1.4.1.4203.1.3.3 NOM 'supportedAuthPasswordSchemes'
DESC "schémas de mémorisation de mot de passe pris en charge"
EQUALITY caseExactIA5Match
SYNTAXE 1.3.6.1.4.1.1466.115.121.1.26{32}
USAGE dSAOperation)

Les valeurs de cet attribut sont les noms des schémas d'authentification par mot de passe que le serveur prend en charge. La syntaxe d'un nom de schéma est décrite au paragraphe 2.1. Cet attribut peut seulement être présent dans l'entrée spécifique d'un agent de système de répertoire (*DSE, DSA Specific Entry*) racine. Si le serveur ne prend en charge aucun schéma de mot de passe, cet attribut ne sera pas présent.

2.5 authPassword

(1.3.6.1.4.1.4203.1.3.4 NOM 'authPassword'
DESC "informations d'authentification par mot de passe"
EQUALITY 1.3.6.1.4.1.4203.1.2.2
SYNTAXE 1.3.6.1.4.1.4203.1.1.2)

Les valeurs de cet attribut sont représentatives du ou des mots de passe de l'utilisateur et se conforment à la authPasswordSyntax décrite au paragraphe 2.1. Les valeurs de cet attribut peuvent être utilisées aux fins d'authentification.

Le transfert des valeurs de authPassword est fortement déconseillé lorsque le service de transport sous-jacent ne peut pas garantir la confidentialité et peut résulter en la divulgation des valeurs à des parties non autorisées.

2.6 authPasswordObject

(1.3.6.1.4.1.4203.1.4.7 NOM 'authPasswordObject'
DESC "mélange de mots de passe d'authentification dans une classe"
PEUT 'authPassword'
AUXILIARY)

Les entrées de cette classe d'objets peuvent contenir des types d'attribut authPassword.

3. Schémas

Cette section décrit les schémas "MD5" et "SHA1". D'autres schémas peuvent être définis par d'autres documents. Les schémas qui ne sont pas décrits dans une RFC DEVRAIENT être désignés en commençant par un "X-" pour indiquer qu'ils

sont un schéma privé ou spécifique de mise en œuvre, ou peuvent être désignés en utilisant la représentation en décimal séparé par des points de la [RFC2252] d'un OID alloué au schéma.

3.1 Schéma MD5

Le nom du schéma MD5 [RFC1321] est "MD5".

La authValue est le codage en base64 d'un résumé MD5 de l'enchaînement du mot de passe d'utilisateur et du sel. Le codage en base64 du sel est fourni dans le champ authInfo. Le sel DOIT être long d'au moins 64 bits. Les mises en œuvre de ce schéma DOIVENT supporter des sels jusqu'à 128 bits de long.

Exemple :
Soit un utilisateur "joe" dont le mot de passe est "mary" et un sel de "salt", le champ authInfo va être le codage en base64 du "sel" et le champ authValue va être le codage en base64 du résumé MD5 de "marysalt".

Une confrontation entre un mot de passe affirmé et une valeur d'attribut de ce schéma DEVRA être vraie si et seulement si le résumé MD5 de l'enchaînement de la valeur affirmée et du sel est égale au résumé MD5 contenu dans AuthValue. La confrontation DEVRA être indéfinie si le serveur est incapable de réaliser l'essai d'égalité pour une raison quelconque. Autrement le résultat de la confrontation DEVRA être faux.

Les valeurs de ce schéma DEVRAIENT seulement être utilisées pour mettre en œuvre une simple authentification d'utilisateur/mot de passe.

3.2 Schéma SHA1

Le nom du schéma SHA1 [SHA1] est "SHA1".

La authValue est le codage en base64 d'un résumé SHA1 de l'enchaînement du mot de passe d'utilisateur et du sel. Le codage en base64 du sel est fourni dans le champ authInfo. Le sel DOIT être long d'au moins 64 bits. Les mises en œuvre de ce schéma DOIVENT supporter des sels jusqu'à 128 bits de long.

Exemple :
Soit un utilisateur "joe" dont le mot de passe est "mary" et un sel de "salt", le champ authInfo va être le codage en base64 du "sel" et le champ authValue va être le codage en base64 du résumé SHA1 de "marysalt".

Une confrontation entre un mot de passe affirmé et une valeur d'attribut de ce schéma DEVRA être vraie si et seulement si le résumé SHA1 de l'enchaînement de la valeur affirmée et du sel est égale au résumé SHA1 contenu dans AuthValue. La confrontation DEVRA être indéfinie si le serveur est incapable de réaliser l'essai d'égalité pour une raison quelconque. Autrement le résultat de la confrontation DEVRA être faux.

Les valeurs de ce schéma DEVRAIENT seulement être utilisées pour mettre en œuvre une simple authentification d'utilisateur/mot de passe.

4. Questions de mise en œuvre

Pour toutes les mises en œuvre de la présente spécification :

- Les serveurs PEUVENT restreindre quels schémas sont utilisés en conjonction avec un processus d'authentification particulier mais DEVRAIENT utiliser toutes les valeurs des schémas choisis. Si le mot de passe affirmé correspond à une des valeurs mémorisées, le mot de passe affirmé DEVRAIT être considéré comme valide. Les serveurs PEUVENT utiliser d'autres mécanismes de mémorisation d'authentification, comme un userPassword ou une mémorisation externe de mot de passe, en conjonction avec authPassword pour prendre en charge le processus d'authentification.

- Les serveurs qui acceptent un simple bind DOIVENT prendre en charge le schéma SHA1 et DEVRAIENT prendre en charge le schéma MD5.

- Les serveurs NE DEVRAIENT PAS publier les valeurs de authPassword ni permettre des opérations qui exposent les valeurs de authPassword ou des assertions de AuthPasswordMatch si la protection de la confidentialité n'est pas activée.

- Les clients NE DEVRAIENT PAS initier des opérations qui fournissent ou demandent des valeurs de authPassword ou font des assertions de authPasswordMatch si la protection de la confidentialité n'est pas activée.

- Les clients NE DEVRAIENT PAS supposer qu'une AuthPasswordMatch réussie, que ce soit par comparaison ou par recherche, est suffisante pour obtenir l'accès au répertoire. L'opération bind DOIT être utilisée pour s'authentifier au répertoire.

5. Considérations de sécurité

Le présent document décrit comment les informations d'authentification peuvent être mémorisées dans un répertoire. Les informations d'authentification DOIVENT être adéquatement protégées car la divulgation involontaire va permettre à des attaquants d'obtenir un accès immédiat au répertoire, comme décrit dans la [RFC2829].

Comme des fautes peuvent être découvertes dans l'algorithme de hachage ou dans une mise en œuvre particulière de l'algorithme, ou que des valeurs pourraient être soumises à diverses attaques si elles sont exposées, les valeurs de AuthPassword DEVRAIENT être protégées comme si elles étaient des mots de passe en clair. Quand des valeurs sont transférées, les protections de confidentialité, comme IPSEC ou TLS, DEVRAIENT être en place.

Les clients DEVRAIENT utiliser des mécanismes d'authentification forts [RFC2829].

La règle de correspondance AuthPasswordMatch permet aux applications de vérifier la validité du mot de passe d'un utilisateur et donc, peut être utilisée pour monter une attaque. Les serveurs DEVRAIENT prendre des mesures appropriées pour protéger le répertoire contre de telles attaques.

Certains schémas de mot de passe peuvent exiger des opération de CPU intensives. Les serveurs DEVRAIENT prendre des mesures appropriées pour se protéger contre les attaques de déni de service.

AuthPassword ne restreint pas une identité d'authentification à un seul mot de passe. Un attaquant qui obtient l'accès en écriture à cet attribut peut mémoriser des valeurs supplémentaires sans désactiver le ou les vrais mots de passe de l'utilisateur. L'utilisation de clients et serveurs connaissant la politique de sécurité est RECOMMANDÉE.

Le niveau de protection offert contre les diverses attaques diffère d'un schéma à l'autre. Il est RECOMMANDÉ que les serveurs prennent en charge le choix de schéma comme élément de configuration. Cela permet qu'un schéma soit facilement désactivé si une faute de sécurité significative est découverte.

6. Remerciements

Le présent document fait des emprunts à un certain nombre de documents de l'IETF et se fonde sur des contributions au groupe de travail LDAPext de l'IETF.

7. Bibliographie

[RFC1321] R. Rivest, "Algorithme de résumé de message MD5", avril 1992. *(Information)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir* RFC5234)

[RFC2251] M. Wahl, T. Howes et S. Kille, "Protocole léger d'accès à un répertoire (v3)", décembre 1997.

[RFC2252] M. Wahl, A. Coulbeck, T. Howes, S. Kille, "Protocole léger d'accès à un répertoire (v3) : Définitions de syntaxe d'attribut", décembre 1997. (*Obsolète, voir* RFC4510, RFC4517, RFC4523, RFC4512) (*MàJ par*

RFC3377) *(P.S.)*

[RFC2256] M. Wahl, "Résumé du schéma d'utilisateur X.500(96) à utiliser avec LDAPv3", décembre 1997. (*Obsolète, voir* RFC4517, RFC4519, RFC4523, RFC4512, RFC4510) *(P.S.)*

[RFC2307] L. Howard, "Approche de l'utilisation de LDAP comme service d'informations réseau", mars 1998. *(Expérimentale)*

[RFC2829] M. Wahl et autres, "Méthodes d'authentification pour LDAP", mai 2000. (*Obsolète, voir* RFC4513, RFC4510) *(P.S.)*

[RFC3062] K. Zeilenga, "Opération étendue de modification de mot de passe LDAP", février 2001.

[SHA1] NIST, FIPS PUB 180-1, "Secure Hash Standard", avril 1995.

8. Adresse de l'auteur

Kurt D. Zeilenga
OpenLDAP Foundation

mél : Kurt@OpenLDAP.org

9. Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.