rfc3112

Schéma de mot de passe d'authentification

L'attribut **userPassword** est conçu pour utiliser l'opération simple bind password. Cependant les valeurs de **userPassword** doivent être des mots de passe en texte clair. L'attribut **authPassword** est conçus pour stocker des informations utilisées pour implémenter une authentification par mot de passe simple. L'attribut supporte plusieurs schéma de stockage. Une règle de correspondance est fournie pour l'utilisation avec des filtres de recherche qui permettent aux clients d'affirmer qu'un mot de passe matche une des valeurs de l'attribut.

Définition du schéma

```
( 1.3.6.1.4.1.4203.1.1.2 DESC 'authentication password syntax' )
```

Les valeurs de cette syntaxe sont encodées en accord avec :

```
authPasswordValue = w scheme s authInfo s authValue w
scheme = %x30-39 / %x41-5A / %x2D-2F / %x5F ; 0-9, A-Z, "-", ".", "/", or "_"
authInfo = schemeSpecificValue
authValue = schemeSpecificValue
schemeSpecificValue = *( %x21-23 / %x25-7E ) ; printable ASCII less "$" and " "
s = w SEP w
w = *SP
SEP = %x24 ; "$"
SP = %x20 ; " " (space)
```

où **scheme** décrit le mécanisme et **authInfo** et **authValue** sont spécifique au schéma. **authInfo** est souvent un salt encodé en base64. **authValue** est souvent un dérivé de mot de passe encodé en base 64.

authPasswordExactMatch Règle de correspondance qui permet à un client d'affirmer une valeur authPasswordSyntax (règle d'égalité)

authPasswordMatch Règle de correspondance qui permet à un client d'affirmer qu'un mot depasse match un authPasswordSyntax en utilisant un filtre extensibleMatch.

supportedAuthPasswordSchemes Les valeurs de cet attribut sont des noms de schéma de mot de passe supportés par le serveur

authPassword Les valeurs de cet attribut représentent les mots de passe de l'utilisateur

authPasswordObject Les entrées de cet classe d'objet peuvent contenir un attribut authPassword

authPasswordExactMatch

```
( 1.3.6.1.4.1.4203.1.2.2
 NAME 'authPasswordExactMatch'
 DESC 'authentication password exact matching rule'
 SYNTAX 1.3.6.1.4.1.4203.1.1.2 )
```

authPasswordMatch

```
( 1.3.6.1.4.1.4203.1.2.3
 NAME 'authPasswordMatch'
 DESC 'authentication password matching rule'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.40{128} )
```

supportedAuthPasswordSchemes

```
( 1.3.6.1.4.1.4203.1.3.3
 NAME 'supportedAuthPasswordSchemes'
 DESC 'supported password storage schemes'
 EQUALITY caseExactIA5Match
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32}
 USAGE dSAOperation )
```

authPassword

```
( 1.3.6.1.4.1.4203.1.3.4 NAME 'authPassword'
 DESC 'password authentication information'
 EQUALITY 1.3.6.1.4.1.4203.1.2.2
 SYNTAX 1.3.6.1.4.1.4203.1.1.2 )
```

authPasswordObject

```
( 1.3.6.1.4.1.4203.1.4.7 NAME 'authPasswordObject'
 DESC 'authentication password mix in class'
 MAY 'authPassword'
 AUXILIARY )
```

schéma MD5

authValue est le MD5 digest encodé base64 de la concaténation du mot de passe et du salt. L'encodage base 64 est fournis dans **authInfo**. Le salt doit faire 64 bits minimum.

schéma SHA1

authValue est le digest SHA1 encodé base64 de la concaténation du mot de passe et du salt. L'encodage base 64 est fournis dans **authInfo** Le salt doit faire 64 bits minimum.

Problèmes d'implèmentation

Les serveurs peuvent restreindre le schéma utilisé en conjonction avec un processus d'authentification particulier. Les serveurs peuvent utiliser d'autres mécanismes de stockage, tel que **userPassword** ou un stockage externe, en conjonction avec **authPassword**.

Les serveurs qui supportent le simple bind doivent supporter SHA1 et devraient supporter MD5.

Les serveurs ne devraient pas publier ou initier des opérations sur les valeurs de **authPassword** ni permettre des opérations qui exposent **authPassword** ou les assertions **authPasswordMatch** à moins qu'une protection de confidentialité soit en place.

Les serveur ne devraient pas assumer qu'un **AuthPasswordMatch** réussi, soit par compare ou par search, est suffisant pour avoir accès à l'annuaire. L'opération bind doit être utilisé pour authentifier l'annuaire.