Groupe de travail Réseau
**Request for Comments : 4616**
RFC mise à jour : 2595
Catégorie : En cours de normalisation

K. Zeilenga, éditeur
OpenLDAP Foundation
août 2006
Traduction Claude Brière de L'Isle

# Mécanisme PLAIN d'authentification simple et couche de sécurité (SASL)

**Statut du présent mémoire**
Le présent document spécifie un protocole de normalisation Internet pour la communauté Internet, et appelle à discussion et suggestions en vue de son amélioration. Prière de se rapporter à l'édition en cours des "Internet Official Protocol Standards" (normes officielles du protocole Internet) (STD 1) pour connaître l'état de la normalisation et le statut du présent protocole. La distribution du présent mémo n'est pas soumise à restrictions.



**Résumé**
Le présent document définit un mécanisme simple de nom d'utilisateur/mot de passe en clair d'authentification simple et de couche de sécurité (SASL, *Simple Authentication and Security Layer*) appelé le mécanisme PLAIN. Le mécanisme PLAIN est destiné à être utilisé en combinaison avec les services de confidentialité des données fournis par une couche inférieure, dans des protocoles auxquels fait défaut une simple commande d'authentification de mot de passe.

## 1. Introduction

Les mots de passe en clair, multi-usage sont simples, interopèrent avec presque toutes les bases de données d'authentification des systèmes d'exploitation, et sont utiles pour une transition en douceur vers un mécanisme d'authentification fondé sur le mot de passe plus sûr. L'inconvénient est que leur utilisation est inacceptable sur des connexions réseau où la confidentialité des données n'est pas assurée.

Le présent document définit le mécanisme PLAIN d'authentification simple et couche de sécurité [SASL] à utiliser dans des protocoles sans commande de connexion en clair (par exemple, [ACAP] ou [SMTP-AUTH]). Le présent document met à jour la RFC2595, en en remplaçant la Section 6. Les changements par rapport à la RFC2595 sont détaillés à l'appendice A.

Le nom associé à ce mécanisme est "PLAIN".

Le mécanisme PLAIN de SASL ne fournit pas de couche de sécurité.

Le mécanisme PLAIN ne devrait pas être utilisé sans une protection adéquate de la sécurité des données car ce mécanisme ne comporte par lui-même pas de protection de l'intégrité ou de la confidentialité. Le mécanisme est destiné à être utilisé avec les protections de la sécurité des données fournies par le protocole de couche d'application, normalement à travers son utilisation des services de sécurité de la couche Transport ([TLS]).

Par défaut, les mises en œuvre DEVRAIENT n'avertir du mécanisme PLAIN et n'en faire usage que lorsque des services de sécurité adéquats sont en place. Les spécifications pour les protocoles de l'IETF qui indiquent que ce mécanisme est un mécanisme d'authentification applicable DOIVENT rendre obligatoire la prise en charge par les mises en œuvre d'un service fort de sécurité des données, tel que TLS.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans [Keywords].

## 2. Mécanisme PLAIN de SASL

Le mécanisme consiste en un seul message, une chaîne de caractères [UTF-8] codés en [Unicode], du client au serveur. Le client présente l'identité d'autorisation (identité pour agir en tant que), suivie par un caractère NUL (U+0000), suivi par l'identité d'authentification (identité dont le mot de passe va être utilisé), suivi par un caractère NUL (U+0000), suivi par le mot de passe en clair. Comme avec les autres mécanismes SASL, le client ne fournit pas d'identité d'autorisation lorsqu'il souhaite que le serveur déduise une identité des accréditifs et utilise cela comme identité d'autorisation.

La grammaire formelle du message client en BNF augmenté [ABNF] est données ci-après.

```
message     = [authzid] UTF8NUL authcid UTF8NUL passwd
authcid     = 1*SAFE            ; DOIT accepter jusqu'à 255 octets
authzid     = 1*SAFE            ; DOIT accepter jusqu'à 255 octets
passwd      = 1*SAFE            ; DOIT accepter jusqu'à 255 octets
UTF8NUL     = %x00              ; caractère NUL UTF-8 codé
SAFE        = UTF1 / UTF2 / UTF3 / UTF4
                                ; tout caractère UTF-8 codé en Unicode sauf NUL
UTF1        = %x01-7F           ;; excepté NUL
UTF2        = %xC2-DF UTF0
UTF3        = %xE0 %xA0-BF UTF0 / %xE1-EC 2(UTF0) / %xED %x80-9F UTF0 / %xEE-EF 2(UTF0)
UTF4        = %xF0 %x90-BF 2(UTF0) / %xF1-F3 3(UTF0) / %xF4 %x80-8F 2(UTF0)
UTF0        = %x80-BF
```

L'identité d'autorisation (authzid), l'identité d'authentification (authcid), le mot de passe (passwd), et les délimiteurs de caractère NUL DEVRONT être transférés sous forme de chaîne de caractères Unicode codés en UTF-8. Comme le caractère NUL (U+0000) est utilisé comme délimiteur, le caractère NUL (U+0000) NE DOIT PAS apparaître dans les productions de authzid, authcid, ou passwd.

La forme de la production authzid est spécifique du profil SASL du protocole de niveau application. Les productions authcid et passwd sont de forme libre. L'utilisation de caractères non visibles ou de caractères qu'un utilisateur pourrait être incapable d'entrer sur certains claviers est déconseillée.

Les serveurs DOIVENT être capables d'accepter les productions authzid, authcid, et passwd jusqu'à 255 octets inclus. On notera que le codage UTF-8 d'un caractère Unicode peut aller jusqu'à 4 octets.

À réception du message, le serveur va vérifier l'identité d'authentification (authcid) présentée (dans le message) et le mot de passe (passwd) dans la base de données d'authentification du système, et il va vérifier que les accréditifs d'authentification permettent au client d'agir en tant qu'identité d'autorisation (authzid) (présentée ou déduite). Si les deux étapes réussissent, l'usager est authentifié.

Les chaînes identité et mot de passe d'authentification présentées, ainsi que les chaînes identité et mot de passe d'authentification de base de données, sont à préparer avant d'être utilisées dans le processus de vérification. Le profil [SASLPrep] de l'algorithme [StringPrep] est l'algorithme de préparation RECOMMANDÉ. L'algorithme de préparation SASLprep est recommandé pour améliorer la probabilité que les comparaisons se comportent de la façon prévue. L'algorithme de préparation SASLprep n'est pas obligatoire de façon à permettre au serveur d'employer d'autres algorithmes de préparation (y compris ancun) en tant que de besoin. Par exemple, l'utilisation d'un algorithme de préparation différent peut être nécessaire pour que le serveur interopère avec un système externe.

Lors de la préparation des chaînes présentées en utilisant [SASLPrep], les chaînes présentées sont à traiter comme des chaînes "d'interrogation" (Section 7 de [StringPrep]) et donc des codets non alloués peuvent apparaître dans le résultat préparé. Lors de la préparation des chaînes de base de données en utilisant [SASLPrep], les chaînes de base de données sont à traiter comme des chaînes "mémorisées" (Section 7 de [StringPrep]) et donc l'apparition de codets non alloués est interdite dans leur résultat préparé.

Sans considération de l'algorithme de préparation utilisé, si le résultat d'une fonction non réversible (par exemple, de hachage) de la chaîne attendue est mémorisé, la chaîne DOIT être préparée avant de l'introduire dans cette fonction.

Sans considération de l'algorithme de préparation utilisé, si la préparation échoue ou résulte en une chaîne vide, la vérification DEVRA échouer.
Lorsque aucune identité d'autorisation n'est fournie, le serveur déduit une identité d'autorisation de la représentation préparée de la chaîne d'identité d'authentification fournie. Ceci assure que la déduction des différentes représentations de l'identité d'authentification produit la même identité d'autorisation.

Le serveur PEUT utiliser les accréditifs pour initialiser toute nouvelle base de données d'authentification, tels que l'un de ceux qui conviennent pour [CRAM-MD5] ou [DIGEST-MD5].

## 3. Pseudo-Code

La présente section fournit le pseudo-code illustrant le processus de vérification (en utilisant des mots de passe hachés et la fonction de préparation SASLprep ) exposé ci-dessus. Cette section n'est pas normative.

```
booléen Vérifier(chaîne authzid, chaîne authcid, chaîne passwd) {
        string pAuthcid = SASLprep(authcid, vrai)       ; # préparer authcid
        string pPasswd = SASLprep(passwd, vrai)         ; # préparer passwd
        si (pAuthcid == NULL || pPasswd == NULL) {
        returner faux                                   ; # échec de la préparation
        }
        si (pAuthcid == "" || pPasswd == "") {
        returner faux                                   ; # chaîne préparée vide
        }

storedHash = FetchPasswordHash(pAuthcid);
        if (storedHash == NULL || storedHash == "") {
        returner faux                                   ; # erreur ou authcid inconnu
        }

        si (!Comparer(storedHash, Hash(pPasswd))) {
        returner faux                                   ; # mot de passe incorrect
        }

        si (authzid == NULL ) {
        authzid = DeriveAuthzid(pAuthcid);
        si (authzid == NULL || authzid == "") {
        returner faux                                   ; # authzid n'a pas pu être déduit
        }
        }

          si (!Authorize(pAuthcid, authzid)) {
          returner faux       ;     # non autorisé
          }

          returner vrai;
          }
```

Le second paramètre de la fonction SASLprep, lorsqu'il est vrai, indique que les codets non alloués sont permis dans l'entrée. Lorsque la fonction SASLprep est invoquée pour préparer le mot de passe avant de calculer le hachage mémorisé, le second paramètre sera faux.

Le second paramètre fourni à la fonction Authorize n'est pas préparé par ce code. Le profil SASL de niveau application devrait être consulté pour déterminer quelle préparation est nécessaire, s'il en est.

Noter que les fonctions DeriveAuthzid et Authorize (qu'elles soient mises en œuvre comme une fonction ou deux, qu'elles soient conçues d'une façon par laquelle ces fonctions, ou que le mécanisme de mise en œuvre, peuvent être réutilisées ailleurs) exigent la connaissance et la compréhension du mécanisme et de la spécification et/ou de la mise en œuvre des détails du protocole de niveau application à appliquer.

Noter que le résultat de la fonction Authorize dépend clairement des détails du modèle d'autorisation et de politique local. Les deux fonctions peuvent aussi bien dépendre d'autres facteurs.

## 4. Exemples

Cette section donne des exemples d'échanges d'authentification PLAIN.

Les exemples sont destinés à aider le lecteur à comprendre le texte précédent. Ces exemples ne sont pas normatifs.

"C:" et "S:" indiquent les lignes envoyées respectivement par le client et le serveur. "<NUL>" représente un seul caractère NUL (U+0000). Le protocole d'accès aux configurations d'application ([ACAP]) est utilisé dans les exemples.

Le premier exemple montre comment le mécanisme PLAIN pourrait être utilisé pour l'authentification de l'utilisateur.

```
S: * ACAP (SASL "CRAM-MD5") (STARTTLS)
C: a001 STARTTLS
S: a001 OK "Commencer maintenant la négociation TLS" <Négociation TLS, les autres commandes sont sous la couche
        TLS>
S: * ACAP (SASL "CRAM-MD5" "PLAIN")
C: a002 AUTHENTICATE "PLAIN"
S: + ""
C: {21}
C: <NUL>tim<NUL>tanstaaftanstaaf
S: a002 OK "Authentifié"
```

Le second exemple montre comment le mécanisme PLAIN pourrait être utilisé pour tenter d'assumer l'identité d'un autre usager. Dans cet exemple, le serveur rejette la demande. Cet exemple utilise aussi la capacité de réponse initiale facultative du protocole pour éliminer un aller-retour.

```
S: * ACAP (SASL "CRAM-MD5") (STARTTLS)
C: a001 STARTTLS
S: a001 OK "Commencer maintenant la négociation TLS" <Négociation TLS, les autres commandes sont sous la couche
        TLS>
S: * ACAP (SASL "CRAM-MD5" "PLAIN")
C: a002 AUTHENTICATE "PLAIN" {20+}
C: Ursel<NUL>Kurt<NUL>xipj3plmq
S: a002 NO "Non autorisé pour l'identité d'autorisation demandée"
```

## 5. Considérations pour la sécurité

Comme le mécanisme PLAIN ne fournit pas par lui-même de protection de l'intégrité ou de la confidentialité, il ne devrait pas être utilisé sans protection adéquate de la sécurité des données externes, telle que les services TLS fournis par de nombreux protocoles de couche application. Par défaut, les mises en œuvre NE DEVRAIENT PAS avertir et NE DEVRAIENT PAS utiliser le mécanisme PLAIN si des services adéquats de sécurité des données ne sont pas en place.

Lorsque le mécanisme PLAIN est utilisé, le serveur obtient la capacité de personnaliser l'usager de tous les services avec le même mot de passe sans considération du chiffrement fourni par TLS ou d'autres mécanismes de protection de la confidentialité. Bien que beaucoup d'autres mécanismes d'authentification aient des faiblesses similaires, de plus forts mécanismes SASL traitent cette question. Les clients sont invités à avoir un mode de fonctionnement dans lequel tous les mécanismes qui pourraient révéler le mot de passe de l'usager au serveur sont désactivés.

Les considérations générales de sécurité [SASL] s'appliquent à ce mécanisme.

Les considérations de sécurité pour Unicode, [UTF-8], et [StringPrep] s'appliquent aussi.

## 6. Considérations relatives à l'IANA

L'entrée du registre Mécanisme SASL [IANA-SASL] pour le mécanisme PLAIN a été mise à jour par l'IANA pour refléter que le présent document fournit maintenant sa spécification technique.

À : iana@iana.org
Sujet : Mise à jour de l'enregistrement du mécanisme SASL PLAIN
Nom du mécanisme SASL : PLAIN
Considérations pour la sécurité : Voir la RFC 4616.
Spécification publiée (facultative, recommandée) : RFC 4616
Adresse et mèl de la personne à contacter pour plus d'informations :
Kurt Zeilenga <kurt@openldap.org>

IETF SASL WG <ietf-sasl@imc.org>
Usage de destination : COMMUN
Auteur/Contrôleur des changements : IESG <iesg@ietf.org>
Note : Met à jour l'entrée existante pour PLAIN

## 7. Remerciements

Le présent document est une révision de la RFC 2595 par Chris Newman. Les portions de grammaire définies à la Section 2 ont été empruntées à [UTF-8] par François Yergeau.

Le présent document a été produit par le groupe de travail Authentification simple et couche de sécurité (SASL) de l'IETF.

## 8. Références normatives

[ABNF] D. Crocker, éd. et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", RFC 4234, octobre 2005.

[Keywords] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, RFC 2119, mars 1997.

[SASL] A. Melnikov, éd., et K. Zeilenga, éd., "Authentification simple et couche de sécurité (SASL)", RFC 4422, juin 2006.

[SASLPrep] K. Zeilenga, "SASLprep : Profil Stringprep pour les noms d'utilisateur et les mots de passe", RFC 4013, février 2005.

[StringPrep] P. Hoffman et M. Blanchet, "Préparation des chaînes internationalisées ("stringprep")", RFC 3454, décembre 2002.

[Unicode] The Unicode Consortium, "Norme Unicode, version 3.2.0" définie par "The Unicode Standard, Version 3.0" (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5), amendée par "Unicode Standard Annexe n° 27 : Unicode 3.1" (http://www.unicode.org/reports/tr27/) et par "Unicode Standard Annexe n° 28 : Unicode 3.2" (http://www.unicode.org/reports/tr28/).

[UTF-8] F. Yergeau, "UTF-8, un format de transformation de ISO 10646", STD 63, RFC 3629, novembre 2003.

[TLS] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", RFC 4346, avril 2006.

## 9. Références informatives

[ACAP] C. Newman et J. Myers, "ACAP – Protocole d'accès aux configurations d'application", RFC 2244, novembre 1997.

[CRAM-MD5] L. Nerenberg, éd., "The CRAM-MD5 SASL Mechanism", Travail en cours, juin 2006.

[DIGEST-MD5] A. Melnikov, éd., "Utilisation de l'authentification de résumé comme mécanisme SASL", Travail en cours, juin 2006.

[IANA-SASL] IANA, "Mécanismes d'authentification simple et de couche de sécurité (SASL)", <http://www.iana.org/assignments/sasl-mechanisms>.

[SMTP-AUTH] J. Myers, "Extension du service SMTP pour l'authentification", RFC 2554, mars 1999.

## Appendice A. Changements depuis la RFC 2595

Cet appendice n'est pas normatif. Le présent document remplace la Section 6 de la RFC 2595. La spécification précise comment le serveur va comparer les chaînes de caractères fournies par le client avec les chaînes de caractères mémorisées.

La grammaire ABNF a été mise à jour. En particulier, la grammaire permet maintenant les caractères LINE FEED (U+000A) et CARRIAGE RETURN (U+000D) dans les productions authzid, authcid, passwd. Cependant, la possibilité d'utiliser ces caractères de contrôle dépend des règles de préparation de chaîne applicables à la production. Pour les productions passwd et authcid, les caractères de contrôle sont interdits. Pour authzid, on doit consulter le profil SASL de niveau application. Ce changement permet à PLAIN de porter toutes les chaînes d'identité d'autorisation possibles permises dans SASL.

Le pseudo-code a été ajouté.

La section exemple a été développée pour illustrer plus de caractéristiques du mécanisme PLAIN.

### Adresse de l'éditeur

Kurt D. Zeilenga
OpenLDAP Foundation
mél : Kurt@OpenLDAP.org

## Déclaration complète de copyright



### Propriété intellectuelle



### Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'activité de soutien administratif (IASA) de l'IETF.