Groupe de travail Réseau
Request for Comments : 4954
RFC rendue obsolète :2554
RFC mise à jour : 3463
Catégorie : Sur la voie de la normalisation

R. Siemborski, éd., Google, Inc.
A. Melnikov, éd., Isode Limited
juillet 2007

Traduction Claude Brière de L'Isle

Extension de service SMTP pour l'authentification

Résumé
Le présent document définit une extension au protocole simple de transport de messagerie (SMTP, *Simple Mail Transport Protocol*) par laquelle un client SMTP peut indiquer un mécanisme d'authentification au serveur, effectuer un échange de protocole d'authentification, et facultativement, négocier une couche de sécurité pour les interactions de protocole suivantes durant cette session. Cette extension inclut un profil de l'authentification simple et couche de sécurité (SASL, *Simple Authentication et Security Layer*) pour SMTP.

Le présent document rend obsolète la RFC 2554.

Table des matières

1. Introduction

Le présent document définit une extension au protocole simple de transport de messagerie (SMTP, *Simple Mail Transport Protocol*) par laquelle un client SMTP peut indiquer un mécanisme d'authentification au serveur, effectuer un échange de protocole d'authentification, et facultativement de négocier une couche de sécurité pour les interactions de protocole suivantes durant cette session et, durant une transaction de messagerie, facultativement spécifier une boîte aux lettres associée à l'identité qui a soumis le message au système de livraison de messages.

Cette extension inclut un profil de l'authentification simple et couche de sécurité (SASL, *Simple Authentication et Security Layer*) pour SMTP.

Quand on le compare à la RFC 2554, le présent document déconseille l'utilisation du code de réponse 538, ajoute un nouveau code d'état amélioré, ajoute l'exigence de la prise en charge du profil SASLprep pour préparer les identités d'autorisation, recommande l'utilisation des types de transmission de la RFC 3848 dans le champ d'en-tête Trace reçue, et précise l'interaction avec l'extension SMTP PIPELINING [RFC2920].

2. Comment lire ce document

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].

Dans les exemples, "C:" et "S:" indiquent les lignes envoyées respectivement par le client et le serveur.

3. Extension de service Authentification

1. Le nom de cette extension de service [RFC2821] est "Authentication".

2. La valeur de mot clé EHLO associée à cette extension est "AUTH".

3. Le mot clé EHLO AUTH contient comme paramètre une liste séparée par des espaces des noms des mécanismes disponibles de la [RFC4422]. La liste des mécanismes disponibles PEUT changer après une commande STARTTLS réussie [RFC3207].

4. Un nouveau verbe [RFC2821] "AUTH" est défini.

5. Un paramètre facultatif utilisant le mot clé "AUTH" est ajouté à la commande MAIL FROM, et étend la longueur de ligne maximum de la commande MAIL FROM de 500 caractères.

6. Cette extension est appropriée pour le protocole de soumission de la [RFC4409].

4. Commande AUTH

Mécanisme AUTH [réponse initiale]

Arguments :
Mécanisme : chaîne identifiant un mécanisme d'authentification de la [RFC4422].
Réponse initiale : réponse initiale facultative du client. Si elle est présente, cette réponse DOIT être codée comme décrit à la Section 4 de la [RFC4648] ou contenir un seul caractère "=".

Restrictions : après qu'une commande AUTH a été achevée avec succès, aucune autre commande AUTH ne peut être produite dans la même session. Après l'achèvement réussit d'une commande AUTH, un serveur DOIT rejeter toute autre commande AUTH avec une réponse 503. La commande AUTH n'est pas permise durant une transaction de messagerie. Une commande AUTH produite durant une transaction de messagerie DOIT être rejetée avec une réponse 503.

Discussion : la commande AUTH initie un échange d'authentification de la [RFC4422] entre le client et le serveur. Le client identifie le mécanisme SASL à utiliser avec le premier paramètre de la commande AUTH. Si le serveur prend en charge le mécanisme d'authentification demandé, il effectue l'échange SASL pour authentifier l'utilisateur. Facultativement, il négocie aussi une couche de sécurité pour les interactions de protocole suivantes durant cette session. Si le mécanisme d'authentification demandé est invalide (par exemple, n'est pas supporté ou exige une couche de chiffrement) le serveur rejette la commande AUTH avec une réponse 504. Si le serveur prend en charge l'extension de la [RFC2034], il DEVRAIT retourner un code de réponse amélioré 5.5.4.

L'échange d'authentification SASL consiste en une série de défis du serveur et de réponses du client qui sont spécifiques du mécanisme choisi [RFC4422].

Un défi du serveur est envoyé comme une réponse 334 avec la partie texte contenant la chaîne codée de la [RFC4648] fournie par le mécanisme SASL. Ce défi NE DOIT PAS contenir de texte autre que le défi codé en BASE64.

Une réponse de client consiste en une ligne contenant une chaîne codée de la [RFC4648]. Si le client souhaite annuler l'échange d'authentification, il produit une ligne avec un seul caractère "*". Si le serveur reçoit une telle réponse, il DOIT rejeter la commande AUTH en envoyant une réponse 501.

L'argument de réponse initiale facultative à la commande AUTH est utilisé pour économiser un aller retour quand on utilise des mécanismes d'authentification qui prennent en charge une réponse initiale de client. Si l'argument de réponse initiale est omis et si le mécanisme choisi exige une réponse initiale de client, le serveur DOIT procéder comme défini au paragraphe 5.1 de la [RFC4422]. Dans SMTP, un défi de serveur qui ne contient pas de données est défini comme une réponse 334 sans partie de texte. Noter qu'il y a quand même une espace qui suit le code de réponse, donc la ligne de réponse complète est "334 ".

Noter que la commande AUTH est encore soumise aux limitations de longueur de ligne définies dans la [RFC2821]. Si l'utilisation de l'argument de réponse initiale causerait le dépassement de cette longueur par la commande AUTH, le client NE DOIT PAS utiliser le paramètre Réponse initiale (et plutôt procéder comme défini au paragraphe 5.1 de la [RFC4422]).

Si le client transmet une réponse initiale de longueur zéro, il DOIT alors transmettre la réponse comme un seul signe égal ("="). Cela indique que la réponse est présente, mais ne contient pas de données.

Si le client utilise un argument Réponse initiale à la commande AUTH avec un mécanisme SASL dans lequel le client ne commence pas l'échange d'authentification, le serveur DOIT rejeter la commande AUTH avec une réponse 501. Les serveurs qui utilisent l'extension de codes d'état améliorés de la [RFC2034] DEVRAIENT retourner un code d'état amélioré de 5.7.0 dans ce cas.

Si le serveur ne peut pas décoder de réponse de client, il DOIT rejeter la commande AUTH avec une réponse 501 (et un code d'état amélioré de 5.5.2). Si le client ne peut pas décoder en BASE64 les défis du serveur, il DOIT annuler l'authentification en utilisant la réponse "*". En particulier, les serveurs et les clients DOIVENT rejeter (et non ignorer) tout caractère non explicitement permis par l'alphabet BASE64, et DOIVENT rejeter toute séquence de caractères BASE64 qui contient le caractère de bourrage ('=') ailleurs qu'à la fin de la chaîne (par exemple, "=AAA" et "AAA=BBB" ne sont par permis).

Noter que ces chaînes BASE64 peuvent être beaucoup plus longues qu'une commande SMTP normale. Les clients et serveurs DOIVENT être capables de traiter la taille maximum codée des défis et réponses générés par leurs mécanismes d'authentification Cette exigence est indépendante de toutes les limitations de longueur de ligne que le client ou le serveur peuvent avoir dans d'autres parties de leur mise en œuvre de protocole. (Au moment de la rédaction du présent document, 12288 octets est considéré comme étant une limite de longueur de ligne suffisante pour le traitement des mécanismes d'authentification déployés.) Si, durant un échange d'authentification, le serveur reçoit une ligne plus longue que la mémoire tampon d'authentification du serveur, celui-ci fait échouer la commande AUTH avec la réponse 500. Les serveurs qui utilisent l'extension de codes d'état améliorés de la [RFC2034] DEVRAIENT retourner un code d'état amélioré de 5.5.6 dans ce cas.

L'identité d'autorisation générée par cet échange de la [RFC4422] est un "simple nom d'utilisateur" *(username)* (dans le sens défini dans la [RFC4013]) et le client et le serveur DEVRAIENT (*) tous deux utiliser le profil de la [RFC4013] de l'algorithme de la [RFC3454] pour préparer ces noms pour la transmission ou la comparaison. Si la préparation de l'identité d'autorisation échoue ou résulte en une chaîne vide (sauf si elle a été transmise comme la chaîne vide) le serveur DOIT faire échouer l'authentification.

(*) Note : une future révision de la présente spécification pourrait changer cette exigence en DOIT. Actuellement, le DEVRAIT est utilisé afin d'éviter de rompre avec la majorité des mises en œuvre existantes.

Si le serveur n'est pas capable d'authentifier le client, il DEVRAIT rejeter la commande AUTH avec une réponse 535 sauf si un code d'erreur plus spécifique est approprié. Si le client achevait l'échange avec succès, le serveur SMTP produirait une réponse 235. (Note que le protocole SMTP ne prend pas en charge la caractéristique de SASL de retourner des données supplémentaires avec un résultat réussi.) Ces codes d'état, avec les autres définis par cette extension, sont discutés à la Section 6 du présent document.

Si une couche de sécurité est négociée durant l'échange SASL, elle prend effet pour le client sur l'octet qui suit immédiatement le CRLF qui conclut la dernière réponse générée par le client. Pour le serveur, elle prend effet immédiatement après le CRLF de sa réponse de succès.

Quand une couche de sécurité prend effet, le protocole SMTP est remis à l'état initial (l'état de SMTP après qu'un serveur produit un message d'accueil 220 Service prêt). Le serveur DOIT éliminer toutes les informations obtenues du client, comme l'argument EHLO, qui n'ont pas été obtenues de la négociation SASL elle-même. De même, le client DOIT éliminer toute information obtenue du serveur, comme la liste des extensions de service SMTP, qui n'a pas été obtenue de la négociation SASL elle-même. (Noter qu'un client PEUT comparer les mécanismes SASL annoncés avant et après l'authentification afin de détecter une attaque active de dégradation de négociation).

Le client DEVRAIT envoyer une commande EHLO comme première commande après une négociation SASL réussie qui résulte en l'activation d'une couche de sécurité.

Quand une entité (qu'elle soit l'extrémité client ou serveur) envoie des données, et que les deux couches de sécurité TLS et SASL sont activées, le codage TLS DOIT être appliqué après le codage SASL, sans considération de l'ordre dans lequel les couches ont été négociées.

Le nom de service spécifié par ce profil de protocole de SASL est "smtp". Ce nom de service est aussi à utiliser pour le protocole de la [RFC4409].

Si une commande AUTH échoue, le client PEUT poursuivre sans authentification. Autrement, le client PEUT essayer un autre mécanisme d'authentification ou présenter des accréditifs différents en produisant une autre commande AUTH.

Note : une mise en œuvre de serveur DOIT utiliser une configuration dans laquelle elle NE permet PAS de mécanisme de mot de passe en clair, sauf si la commande STARTTLS [RFC3207] a été négociée ou si quelque autre mécanisme qui protège la session contre la falsification de mot de passe a été fournie. Les sites de serveur NE DEVRAIENT PAS utiliser de configuration qui permette un mécanisme de mot de passe en clair sans un mécanisme de protection contre la divulgation du mot de passe.

Pour assurer l'interopérabilité, les mises en œuvre de client et de serveur de cette extension DOIVENT appliquer le mécanisme SASL PLAIN fonctionnant sur TLS [RFC4346], [RFC3207]. Voir aussi à la Section 15 des exigences supplémentaires sur les mises en œuvre de PLAIN sur TLS.

Noter que de nombreuses mises en œuvre existantes de client et de serveur utilisent le mécanisme SASL CRAM-MD5 de la [RFC2195]. Afin d'assurerl'interopérabilité avec le logiciel déployé, les nouvelles mises en œuvre PEUVENT l'appliquer ; cependant, les mises en œuvre devraient être conscientes que ce mécanisme SASL ne fournit aucune authentification du serveur. Noter qu'au moment de la rédaction du présent document; le groupe de travail SASL travaille sur plusieurs mécanismes de remplacement de SASL qui assurent l'authentification du serveur et d'autres caractéristiques.

Quand la commande AUTH est utilisée avec l'extension PIPELINING de la [RFC2920], elle DOIT être la dernière commande dans un groupe de commandes traitées en parallèle. La seule exception à cette règle est quand la commande AUTH contient une réponse initiale pour un mécanisme SASL qui permet au client d'envoyer d'abord les données, le mécanisme SASL est connu pour s'achever en un aller-retour, et une couche de sécurité n'est pas négociée par le client. Deux exemples de ces mécanismes SASL sont PLAIN [RFC4616] et EXTERNAL [RFC4422].

4.1 Exemples

Voici un exemple de client qui tente AUTH en utilisant le mécanisme SASL de la [RFC4616] sous une couche TLS, et en utilisant la réponse initiale du client :

S: 220-smtp.exemple.com ESMTP Server
C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250-AUTH GSSAPI DIGEST-MD5
S: 250-ENHANCEDSTATUSCODES
S: 250 STARTTLS
C: STARTTLS
S: 220 Prêt à commencer TLS
... la négociation TLS se poursuit, avec d'autres commandes protégées par la couche TLS ...

C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250 AUTH GSSAPI DIGEST-MD5 PLAIN
C: AUTH PLAIN dGVzdAB0ZXN0ADEyMzQ=
S: 235 2.7.0 Authentification réussie

Voici un autre client qui tente AUTH PLAIN sous une couche TLS, cette fois sans la réponse initiale. Les parties de la négociation avant l'établissement de la couche TLS ont été omises :

... la négociation TLS se poursuit, avec d'autres commandes protégées par la couche TLS ...
C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250 AUTH GSSAPI DIGEST-MD5 PLAIN
C: AUTH PLAIN
(note : il y a une seule espace qui suit le 334 sur la ligne suivante)
S: 334
C: dGVzdAB0ZXN0ADEyMzQ=
S: 235 2.7.0 Authentification réussie

Voici un exemple qui utilise CRAM-MD5 [RFC2195], un mécanisme dans lequel le client ne commence pas l'échange d'authentification, et inclut un défi du serveur :

S: 220-smtp.exemple.com ESMTP Server
C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250-AUTH DIGEST-MD5 CRAM-MD5
S: 250-ENHANCEDSTATUSCODES
S: 250 STARTTLS
C: AUTH CRAM-MD5
S: 334 PDQxOTI5NDIzNDEuMTI4Mjg0NzJAc291cmNlZm91ci5hbmRyZXcuY211LmVkdT4=
C: cmpzMyBlYzNhNTlmZWQzOTVhYmExZWM2MzY3YzRmNGI0MWFjMA==
S: 235 2.7.0 Authentification réussie

Voici un exemple d'un client qui tente une AUTH EXTERNAL sous TLS, en utilisant l'identifiant d'autorisation déduit (et donc une réponse initiale de client de longueur zéro).

S: 220-smtp.exemple.com ESMTP Server
C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250-AUTH GSSAPI DIGEST-MD5
S: 250-ENHANCEDSTATUSCODES
S: 250 STARTTLS
C: STARTTLS
S: 220 Prêt à commencer TLS
... la négociation TLS se poursuit, avec d'autres commandes protégées par la couche TLS ...
C: EHLO client.exemple.com
S: 250-smtp.exemple.com Hello client.exemple.com
S: 250 AUTH EXTERNAL GSSAPI DIGEST-MD5 PLAIN
C: AUTH EXTERNAL =
S: 235 2.7.0 Authentification réussie

5. Paramètre AUTH à la commande MAIL FROM

AUTH=mailbox

Arguments : une <boîte aux lettres> *(mailbox)* (voir le paragraphe 4.1.2 de la [RFC2821]) associée à l'identité qui soumet le message au système de livraison, ou la séquence des deux caractères "<>" indiquant que cette identité est inconnue ou insuffisamment authentifiée. Pour se conformer aux restrictions imposées aux paramètres ESMTP, la <boîte aux lettres> est codée dans un xtext. La syntaxe d'un xtext est décrite à la Section 4 de la [RFC3461].

Note : pour les besoins de cette discussion, "identité authentifiée" se réfère à l'identité (si il y en a une) déduite de l'identité d'autorisation de la commande AUTH précédente, tandis que les termes "identité autorisée" et "<boîte aux lettres> fournie" se réfèrent à l'identité de l'envoyeur associée à un message particulier. Noter qu'une identité authentifiée peut être capable d'identifier des messages comme étant envoyés par tout nombre d'identités autorisées au sein d'une session. Par exemple, ce peut être le cas quand un serveur SMTP (une identité authentifiée) traite sa file d'attente (de nombreux messages avec des identités autorisées distinctes).

Discussion : le paramètre facultatif AUTH de la commande MAIL FROM permet à des agents qui coopèrent dans un environnement de confiance de communiquer l'identité d'autorisation associée à des messages individuels.

Si le serveur fait confiance à l'identité authentifiée du client pour affirmer que le message a été à l'origine soumis par la <boîte aux lettres> fournie, alors le serveur DEVRAIT fournir la même <boîte aux lettres> dans un paramètre AUTH quand il relaye le message à tout autre serveur qui prend en charge l'extension AUTH.

Pour cette raison, les serveurs qui annoncent la prise en charge de cette extension DOIVENT accepter le paramètre AUTH à la commande MAIL FROM même quand le client ne s'est pas authentifié lui-même auprès du serveur.

Un paramètre MAIL FROM de AUTH=<> indique que le soumettant original du message n'est pas connu. Le serveur NE DOIT PAS traiter le message comme ayant été originellement soumis par l'identité authentifiée qui résultait de la commande AUTH.

Si le paramètre AUTH de la commande MAIL FROM n'est pas fourni, si le client s'est authentifié, et si le serveur croit que le message est une soumission originale, le serveur PEUT générer une <boîte aux lettres> à partir de l'identité authentifiée de l'utilisateur pour l'utiliser dans un paramètre AUTH quand il relaye le message à tout serveur qui prend en charge l'extension AUTH. La <boîte aux lettres> générée est spécifique de la mise en œuvre, mais elle DOIT se conformer à la syntaxe de la [RFC2821]. Si la mise en œuvre ne peut pas générer une <boîte aux lettres> valide, elle DOIT transmettre AUTH=<> quand elle relaye ce message.

Si le serveur n'a pas une confiance suffisante dans l'identité authentifiée du client, ou si le client n'est pas authentifié, alors le serveur DOIT se comporter comme si le paramètre AUTH=<> était fourni. Le serveur PEUT, cependant, écrire la valeur de tout paramètre AUTH fourni dans un fichier de journal.

Si un paramètre AUTH=<> a été fourni, explicitement ou par suite de l'exigence du paragraphe précédent, alors le serveur DOIT fournir le paramètre AUTH=<> quand il relaye le message à tout serveur qu'il a authentifié en utilisant l'extension AUTH.

Un serveur PEUT traiter l'expansion d'une liste de diffusion comme une nouvelle soumission, réglant le paramètre AUTH à la liste d'adresses de diffusion ou à l'adresse d'administration de liste de diffusion quand il relaye le message aux abonnés à la liste.

Noter qu'une mise en œuvre qui est programmée à traiter tous les clients comme étant insuffisamment de confiance est conforme à la présente spécification. Dans ce cas, la mise en œuvre ne fait rien de plus qu'analyser et éliminer des paramètres AUTH syntaxiquement valides dans la commande MAIL FROM, et fournit des paramètres AUTH=<> à tout les serveurs auxquels elle s'authentifie.

5.1 Exemples

Un exemple où l'identité originale de l'envoyeur est connue et de confiance :

C: MAIL FROM:<e=mc2@exemple.com> AUTH=e+3Dmc2@exemple.com
S: 250 OK

Un exemple où l'identité de l'envoyeur n'est pas de confiance ou est autrement supprimée par le client :

C: MAIL FROM:<john+@exemple.org> AUTH=<>
S: 250 OK

6. Codes d'état

Les codes d'erreur suivants peuvent être utilisés pour indiquer diverses conditions de réussite ou d'échec. Les serveurs qui retournent des codes d'état améliorés de la [RFC2034] DEVRAIENT utiliser les codes améliorés suggérés ici.

235 2.7.0 Authentification réussie
Cette réponse à la commande AUTH indique que l'authentification a réussi.

432 4.7.12 Une transition de mot de passe est nécessaire
Cette réponse à la commande AUTH indique que l'utilisateur doit passer au mécanisme d'authentification choisi. Ceci est normalement fait en s'authentifiant une fois en utilisant le mécanisme d'authentification de la [RFC4616]. Le mécanisme choisi DEVRAIT alors fonctionner pour les authentifications dans les sessions suivantes.

454 4.7.0 Échec temporaire d'authentification
Cette réponse à la commande AUTH indique que l'authentification a échoué à cause d'une défaillance temporaire du serveur. Le client NE DEVRAIT PAS inviter l'utilisateur à faire un autre mot de passe dans ce cas, et devrait plutôt notifier à l'utilisateur la défaillance du serveur.

534 5.7.9 Mécanisme d'authentification trop faible
Cette réponse à la commande AUTH indique que le mécanisme d'authentification choisi est plus faible que ce que la politique du serveur permet pour cet utilisateur. Le client DEVRAIT réessayer avec un nouveau mécanisme d'authentification.

535 5.7.8 Accréditifs d'authentification invalides
Cette réponse à la commande AUTH indique que l'authentification a échoué à cause d'accréditifs d'authentification invalides ou insuffisants. Dans ce cas, le client DEVRAIT demander à l'utilisateur de fournir de nouveaux accréditifs (comme de présenter une boîte de dialogue de mots de passe).

500 5.5.6 La ligne d'échange d'authentification est trop longue
Cette réponse à la commande AUTH indique que l'authentification a échoué parce que le client envoie une réponse BASE64 [RFC4648] qui est plus longue que la taille maximum de mémoire tampon disponible pour le mécanisme SASL actuellement choisi.

530 5.7.0 Authentification requise
Cette réponse DEVRAIT être retournée par toute commande autre que AUTH, EHLO, HELO, NOOP, RSET, ou QUIT quand la politique du serveur exige l'authentification afin d'effectuer l'action demandée et que l'authentification n'est pas actuellement en vigueur.

538 5.7.11 Chiffrement exigé pour le mécanisme d'authentification demandé
Cette réponse à la commande AUTH indique que le mécanisme d'authentification choisi peut seulement être utilisé quand la connexion SMTP sous-jacente est chiffrée. Noter que ce code de réponse n'est documenté ici que pour des raisons historiques. Les mises en œuvre modernes NE DEVRAIENT PAS annoncer des mécanismes qui ne sont pas permis à cause du manque de chiffrement, sauf si une couche de chiffrement de force suffisante est actuellement employée.

Le présent document ajoute plusieurs nouveaux codes d'état améliorés à la liste définie dans la [RFC3463] :

Les trois codes d'état amélioré suivants ont été définis ci-dessus :
5.7.8 Accréditifs d'authentification invalides
5.7.9 Mécanisme d'authentification trop faible
5.7.11 Chiffrement exigé pour le mécanisme d'authentification demandé

X.5.6 Ligne d'échange d'authentification trop longue
Ce code d'état amélioré DEVRAIT être retourné quand le serveur refuse la commande AUTH parce que le client envoie une réponse BASE64 qui est plus longue que la taille maximum de mémoire tampon disponible pour le mécanisme SASL actuellement choisi. Ceci est utile aussi bien pour les erreurs permanentes que pour les erreurs transitoires persistentes.

7. Exigences supplémentaires pour les serveurs

Comme décrit au paragraphe 4.4 de la [RFC2821], un serveur SMTP qui reçoit un message à livrer après un traitement supplémentaire DOIT insérer le champ d'en-tête "Received:" au début du contenu du message. Le présent document pose des exigences supplémentaires sur le contenu d'un champ d'en-tête "Received:" généré. Après une authentification réussie, un serveur DEVRAIT utiliser le mot clé "ESMTPA" ou "ESMTPSA" [RFC3848] (quand approprié) dans la clause "with" du champ d'en-tête Received.

8. Syntaxe formelle

La spécification de syntaxe qui suit utilise la notation de format Backus-Naur augmenté spécifiée dans la [RFC4234]. Les non terminaux référencés mais non définis ci-dessous sont comme défini dans les [RFC4234] ou [RFC4422]. Le non terminal <boîte aux lettres> est défini dans la [RFC2821].

Sauf mention contraire, tous les caractères alphabétiques sont insensibles à la casse. L'utilisation de caractères majuscules ou minuscules pour définir des chaînes de jetons est seulement pour la lisibilité. Les mises en œuvre DOIVENT accepter ces chaînes de façon insensible à la casse.

```
hexchar = "+" HEXDIG HEXDIG

xchar = %x21-2A / %x2C-3C / %x3E-7E          ;; US-ASCII sauf pour "+", "=", SP, et CTL

xtext  = *(xchar / hexchar)                  ;; le non US-ASCII est seulement permis comme hexchar

auth-command  = "AUTH" SP sasl-mech [SP réponse initiale]
                *(CRLF [base64]) [CRLF réponse d'annulation]
                CRLF                         ;; <sasl-mech> est défini dans la [RFC4422]

auth-param = "AUTH=" xtext
;; Paramètre pour la commande MAIL FROM. Ce non terminal se conforme à la syntaxe définie par esmtp-param
   [RFC2821]. La forme décodée de xtext DOIT être soit une <boîte aux lettres> soit les deux caractères "<>"

base64 = base64-terminal / ( 1*(4base64-char) [base64-terminal] )

base64-char = ALPHA / CHIFFRE / "+" / "/"      ;; Sensible à la casse.

base64-terminal = (2base64-char "==") / (3base64-char "=")

continue-req = "334" SP [base64] CRLF
;; Réponse intermédiaire à la commande AUTH. Ce non terminal se conforme à la syntaxe définie par Reply-line
   [RFC2821].

réponse initiale= base64 / "="

réponse d'annulation = "*"
```

9. Considérations sur la sécurité

Les questions de sécurité sont discutées tout au long du présent mémoire.

Si un client utilise cette extension pour obtenir un tunnel chiffré à travers un réseau non sûr avec un serveur coopératif, il doit être configuré à ne jamais envoyer de messagerie à ce serveur quand la connexion n'est pas mutuellement authentifiée et chiffrée. Autrement, un attaquant pourrait voler les messages du client en capturant la connexion [RFC2821] et soit prétendre que le serveur ne prend pas en charge l'extension d'authentification, soit causer l'échec de toutes les commandes AUTH.

Avant que commence la négociation de la [RFC4422], toutes les interactions de protocole sont effectuées en clair et peuvent être modifiées par un attaquant actif. Pour cette raison, les clients et serveurs DOIVENT éliminer toutes les informations obtenues avant le début de la négociation SASL lors de l'établissement d'une couche de sécurité.

Ce mécanisme ne protège pas l'accès TCP, de sorte qu'un attaquant actif peut rediriger une tentative de connexion de relais (c'est-à-dire, une connexion entre deux agents de transfert de messagerie (MTA, *Mail Transfer Agent*)) à l'accès de soumission [RFC4409]. Le paramètre AUTH=<> empêche une telle attaque de causer un relais de message et, en l'absence d'autre authentification d'enveloppe, de capturer l'authentification du client de relais.

Un client de soumission de message peut exiger que l'utilisateur s'authentifie chaque fois qu'un mécanisme convenable de la [RFC4422] est annoncé. Donc, il peut n'être pas souhaitable qu'un serveur de soumission de la [RFC4409] annonce un mécanisme SASL quand l'utilisation de ce mécanisme n'apporte aucun avantage aux clients par rapport à une soumission anonyme.

Les serveurs PEUVENT mettre en œuvre une politique par laquelle la connexion est éliminée après un certain nombre d'échecs de tentative d'authentification. Si ils font ainsi, ils NE DEVRAIENT PAS éliminer la connexion avant l'échec de trois tentatives d'authentification.

Si une mise en œuvre prend en charge les mécanismes de SASL qui sont vulnérables à des attaques d'espionnage passif (tels que ceux de la [RFC4616] elle DOIT alors prendre en charge au moins une configuration où ces mécanismes SASL ne sont pas annoncés ou utilisés sans la présence d'une couche de sécurité externe comme celle de la [RFC4346].

Cette extension n'est pas destinée à remplacer ou être utilisée à la place de la signature de bout en bout de message et des systèmes de chiffrement comme ceux de la [RFC3851] ou de la [RFC2015]. Cette extension vise un problème différent des systèmes de bout en bout de bout en bout ; les différences clés sont :

1. Elle n'est généralement utile que dans une enclave de confiance.

2. EIle protège l'enveloppe entière d'un message, pas seulement le corps du message.

3. Elle authentifie la soumission du message, pas l'auteur du contenu du message.

4. Quand l'authentification mutuelle est utilisée avec une couche de sécurité, elle peut donner à l'envoyeur une assurance que le message a été bien livré sur le prochain bond.

Des considérations de sécurité supplémentaires sont mentionnées dans la [RFC4422]. Les considérations de sécurité supplémentaires spécifiques d'un mécanisme SASL particulier sont décrites dans la spécification pertinentes. Les considérations de sécurité supplémentaires pour PLAIN sur TLS sont mentionnées à la Section 15 de ce document.

10. Considérations relatives à l'IANA

L'IANA a mis à jour l'entrée pour le nom de protocole SASL "smtp" afin qu'elle pointe sur le présent document.

L'IANA a mis à jour l'enregistrement de l'extension de service Authentification SMTP comme définie à la Section 3 de ce document. Ce registre est actuellement situé à <http://www.iana.org/assignments/mail-parameters>.

11. Références normatives

[RFC2034] N. Freed, "Extension de service SMTP pour le retour de codes d'erreur améliorés", octobre 1996. *(P.S.)*

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

[RFC2821] J. Klensin, éditeur, "Protocole simple de transfert de messagerie", STD 10, avril 2001. *(Obsolète, voir* RFC5321)

[RFC3207] P. Hoffman, "Extension de service SMTP pour un SMTP sécurisé sur TLS", février 2002. *(P.S., MàJ par*

RFC7817)

[RFC3280] R. Housley, W. Polk, W. Ford et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002. *(Obsolète, voir* RFC5280)

[RFC3454] P. Hoffman et M. Blanchet, "Préparation de chaînes internationalisées ("stringprep")", décembre 2002. *(P.S.)*

[RFC3461] K. Moore, "Extension de service du protocole simple de transfert de messagerie (SMTP) pour les notifications d'état de livraison (DSN)", janvier 2003. (*MàJ par* RFC3798, RFC3885, RFC5337, RFC6533, RFC8098) *(D.S.)*

[RFC3463] G. Vaudreuil, "Codes d'état améliorés du système de messagerie", janvier 2003. (*MàJ par* RFC3886, RFC4468, RFC4865, RFC4954, RFC5248) *(D.S.)*

[RFC3848] C. Newman, "Enregistrement des types de transmission ESMTP et LMTP", juillet 2004. *(P.S.)*

[RFC4013] K. Zeilenga, "SASLprep : Profil Stringprep pour les noms d'utilisateur et mots de passe", février 2005.

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

[RFC4409] R. Gellens, J. Klensin, "Soumission du message de messagerie électronique", avril 2006. (*Remplacé par la* RFC6409 STD072*)*

[RFC4422] A. Melnikov et K. Zeilenga, éd, "Authentification simple et couche de sécurité (SASL)", juin 2006. *(P.S.)*

[RFC4616] K. Zeilenga, éd., "Mécanisme PLAIN de l'authentification simple et couche de sécurité (SASL)", août 2006. *(P.S.)*

[RFC4648] S. Josefsson, "Codages de données Base16, Base32 et Base64", octobre 2006. (*Remplace* RFC3548) *(P.S.)*

12. Références pour information

[RFC2015] M. Elkins, "Sécurité de MIME avec Pretty Good Privacy (PGP)", octobre 1996. (*MàJ par* RFC3156) *(P.S.)*

[RFC2195] J. Klensin et autres, "Extension IMAP/POP AUTHorize pour mise au défi/réponse simple", septembre 1997. *(P.S.)*

[RFC2920] N. Freed, "Extension de service SMTP pour le traitement de commandes en parallèle", septembre 2000. (STD0060)

[RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. *(Obsolète, voir* RFC5751)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

13. Remerciements

Les éditeurs tiennent à remercier de leurs contributions John Myers et les autres contributeurs à la RFC 2554, sur laquelle le présent document s'appuie fortement.

Les éditeurs tiennent aussi à remercier Ken Murchison, Mark Crispin, Chris Newman, David Wilson, Dave Cridland, Frank Ellermann, Ned Freed, John Klensin, Tony Finch, Abhijit Menon-Sen, Philip Guenther, Sam Hartman, Russ Housley, Cullen Jennings, et Lisa Dusseault pour le temps qu'ils ont consacré à la relecture du présent document et/ou pour les commentaires reçus.

14. Exigences supplémentaires pour l'utilisation de SASL PLAIN sur TLS

Cette Section est normative pour les mises en œuvre de SMTP qui prennent en charge SASL PLAIN sur TLS [RFC4346].

Si un client SMTP veut utiliser SASL PLAIN sur TLS pour s'authentifier auprès du serveur SMTP, le client vérifie que le certificat du serveur est en accord avec les règles de la [RFC3280]. Si le serveur n'a pas fourni de certificat, ou si la vérification du certificat échoue, le client NE DOIT PAS tenter de s'authentifier en utilisant le mécanisme SASL PLAIN.

Après une négociation réussie de TLS [RFC4346], le client DOIT vérifier sa compréhension du nom d'hôte du serveur par rapport à l'identité du serveur telle que présentée dans le message Certificat de serveur, afin d'empêcher des attaques par interposition. Si il n'y a pas de correspondance, le client NE DOIT PAS tenter de s'authentifier en utilisant le mécanisme PLAIN de SASL. La confrontation est effectuée selon les règles suivantes :

- Le client DOIT utiliser le nom d'hôte du serveur qu'il a utilisé pour ouvrir la connexion comme valeur à comparer au nom de serveur exprimé dans le certificat du serveur. Le client NE DOIT PAS utiliser de forme de nom d'hôte du serveur déduite d'une source distante non sûre (par exemple, une recherche non sûre dans le DNS). La canonisation de CNAME n'est pas faite.
- Si une extension subjectAltName de type dNSName est présente dans le certificat, elle DEVRAIT être utilisée comme source de l'identité du serveur.
- La confrontation est insensible à la casse.
- Un caractère générique "*" PEUT être utilisé comme composant le plus à gauche du nom dans le certificat. Par exemple, *.exemple.com correspondrait à a.exemple.com, foo.exemple.com, etc., mais ne correspondrait pas à exemple.com.
- Si le certificat contient plusieurs noms (par exemple, plus d'un champ dNSName) alors une correspondance avec un de ces champs est considérée comme acceptable.

15. Changements par rapport à la RFC 2554

1. Il est précisé que les serveurs DOIVENT prendre en charge l'utilisation du paramètre AUTH=mailbox pour MAIL FROM, même quand le client n'est pas authentifié.
2. Précisions sur les exigences d'envoi initial du client, et exemples supplémentaires.
3. Mise à jour des références aux nouvelles versions de diverses spécifications.
4. Exigence que SASL PLAIN (sur TLS) soit de mise en œuvre obligatoire.
5. Précision que la liste des mécanismes peut changer.
6. L'utilisation du code de réponse 538 est déconseillée.
7. Ajout de l'utilisation du profil SASLprep pour la préparation des identités d'autorisation.
8. Nettoyage substantiel des codes de réponse et indication des codes de réponse améliorés suggérés. On indique aussi quels codes de réponse devraient résulter en ce que le client invite l'utilisateur à fournir de nouveaux accréditifs.
9. Mise à jour de la section ABNF pour utiliser la RFC 4234.
10. Précision de l'interaction avec l'extension SMTP PIPELINING.
11. Ajout d'une référence à la RFC 3848.
12. Ajout d'un nouveau code d'état amélioré pour le cas d'une "ligne d'authentification trop longue".
13. Autres précisions rédactionnelles générales.

Adresse des éditeurs

Robert Siemborski
Google, Inc.
1600 Ampitheatre Parkway
Mountain View, CA 94043, USA
téléphone : +1 650 623 6925
mél : robsiemb@google.com

Alexey Melnikov
Isode Limited
5 Castle Business Village, 36 Station Road,
Hampton, Middlesex, TW12 2BX, UK
mél : Alexey.Melnikov@isode.com

Déclaration complète de droits de reproduction

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