Groupe de travail Réseau
Request for Comments : 4467
RFC mise à jour : 3501
Catégorie : Sur la voie de la normalisation

M. Crispin, University of Washington

mai 2006
Traduction Claude Brière de L'Isle

Extension URLAUTH au protocole d'accès au message Internet (IMAP)

Résumé
Le présent document décrit l'extension URLAUTH au protocole d'accès au message Internet (IMAP, *Internet Message Access Protocol*) (RFC 3501) et au schéma d'URL IMAP (IMAPURL) (RFC 2192). Cette extension donne un moyen pour qu'un client IMAP puisse utiliser les URL qui portent une autorisation d'avoir un accès limité aux données de message sur le serveur IMAP.

Un serveur IMAP qui prend en charge cette extension l'indique avec le nom de capacité de "URLAUTH".

Table des matières

1. Introduction

Dans la [RFC2192], un URL de la forme imap://fred@exemple.com/INBOX/;uid=20 exige une autorisation pour l'identifiant d'utilisateur "fred". Cependant, la [RFC2192] implique qu'il prend seulement en charge l'authentification et confond les concepts d'authentification et d'autorisation.

L'extension URLAUTH définit un mécanisme d'autorisation pour les URL IMAP pour remplacer le mécanisme

d'authentification seule de la [RFC2192]. URLAUTH porte l'autorisation dans la chaîne d'URL elle-même et réutilise une portion de la syntaxe du mécanisme d'authentification de la [RFC2192] pour porter l'identité d'autorisation (qui définit aussi l'espace de noms par défaut dans la [RFC3501]).

L'extension URLAUTH donne le moyen par lequel un utilisateur autorisé d'un serveur IMAP peut créer des URL IMAP autorisés par URLAUTH. Un URL autorisé par URLAUTH porte l'autorisation (pas l'authentification) pour les données adressées par cet URL. Cet URL peut être utilisé dans une autre session IMAP pour accéder à un contenu spécifique sur le serveur IMAP, sans fournir par ailleurs d'autorisation pour toutes autres données (comme d'autres données dans la boîte aux lettres spécifiée dans l'URL) possédées par l'utilisateur qui donne l'autorisation.

Conceptuellement, un URL autorisé par URLAUTH peut être vu comme un "ticket de gage" qui ne porte pas d'informations d'authentification et peut être honoré quel que soit celui qui le présente. Cependant, à la différence d'un ticket de gage, URLAUTH a des mécanismes facultatifs pour restreindre l'usage d'un URL autorisé par URLAUTH. En utilisant ces mécanismes, les URL autorisés par URLAUTH peuvent être utilisables par :
. un anonyme (le "ticket de gage" modèle)
. les utilisateurs authentifiés seulement
. un utilisateur authentifié spécifique seulement
. une soumission de message agissant au nom d'un utilisateur spécifique seulement.

Il y a aussi un mécanisme d'expiration.

Un URL autorisé par URLAUTH peut être utilisé dans l'argument de la commande BURL dans la composition d'un message, comme décrit dans la [RFC4468], pour des objets comme de permettre à un client (avec une mémoire ou autres ressources limitées) de soumettre une transmission de message ou de renvoyer à partir d'une boîte aux lettres IMAP sans exiger que le client aille chercher des données de message.

Le URLAUTH est généré en utilisant un nom de mécanisme d'autorisation et un jeton d'autorisation, qui est généré en utilisant une clé d'accès secrète de boîte aux lettres. Un client IMAP peut demander que le serveur génère et alloue une nouvelle clé d'accès de boîte aux lettres (révoquant donc effectivement tous les URL courants en utilisant URLAUTH avec la vieille clé d'accès de boîte aux lettres) mais ne peut pas régler la clé d'accès de boîte aux lettres à une clé de son propre choix.

1.1 Conventions utilisées dans le 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].

La syntaxe formelle utilise la notation de forme Backus-Naur augmentée (ABNF, *Augmented Backus-Naur Form*) incluant les règles cœur définies dans l'Appendice A de la [RFC4234].

Dans les exemples, "C:" et "S:" indiquent les lignes envoyées respectivement par le client et le serveur. Si une seule étiquette "C:" ou "S:" s'applique à plusieurs lignes, le saut à la ligne entre ces lignes est seulement pour la facilité de lecture et ne fait pas partie de l'échange de protocole réel.

2. Concepts

2.1 URLAUTH

URLAUTH est un composant, ajouté à la fin d'un URL, qui porte l'autorisation d'accéder aux données adressées par cet URL. Il contient un identifiant d'accès autorisé, un nom de mécanisme d'autorisation, et un jeton d'autorisation. Le jeton d'autorisation est généré à partir de l'URL, de l'identifiant d'accès autorisé, du nom de mécanisme d'autorisation, et d'une clé d'accès de boîte aux lettres.

2.2 Clé d'accès de boîte aux lettres

La clé d'accès de boîte aux lettres est une chaîne aléatoire avec au moins 128 bits d'entropie. Elle est générée par le logiciel

(pas par l'utilisateur humain) et DOIT être imprévisible.

Chaque utilisateur a un tableau des boîtes aux lettres et une clé d'accès de boîte aux lettres associée pour chaque boîte aux lettres. Par conséquent, la clé d'accès de boîte aux lettres est par utilisateur et par boîte aux lettres. En d'autres termes, deux utilisateurs qui partagent la même boîte aux lettres ont chacun une clé d'accès de boîte aux lettres différente pour cette boîte aux lettres, et chaque boîte aux lettres accédée par un seul utilisateur a aussi une clé d'accès de boîte aux lettres différente.

2.3 Identifiant d'accès autorisé

L'identifiant d'accès autorisé restreint l'utilisation de l'URL autorisé par URLAUTH à certains utilisateurs autorisés sur le serveur, comme décrit à la Section 3.

2.4 Mécanisme d'autorisation

Le mécanisme d'autorisation est l'algorithme par lequel le URLAUTH est généré et ensuite vérifié, en utilisant la clé d'accès de boîte aux lettres.

2.4.1 Mécanisme d'autorisation INTERNAL

La présente spécification définit le mécanisme INTERNAL, qui utilise un algorithme de génération de jeton choisi par le serveur et qui n'implique pas la divulgation de la clé d'accès de boîte aux lettres au client.

Note : L'algorithme de génération de jeton choisi par la mise en œuvre de serveur devrait être moderne et raisonnablement sûr. Au moment de la rédaction du présent document, HMAC-SHA1 [RFC2104] est recommandé. Si il devient nécessaire de changer l'algorithme de génération de jeton du mécanisme INTERNAL (par exemple, parce qu'une attaque contre l'algorithme en cours a été découverte) tous les URL autorisés par URLAUTH actuellement existants seront invalidés par le changement d'algorithme. Comme ce serait une mauvaise surprise pour les applications qui dépendent de la validité d'un URL autorisé par URLAUTH, et qu'il n'y a pas de bon moyen pour faire une mise à jour brute des URL déployés existants, il vaut mieux éviter cette situation en utilisant un algorithme sûr plutôt qu'un "assez bon". Les mises en œuvre de serveur DEVRAIENT envisager la possibilité de changer d'algorithme. Dans certains cas, il peut être souhaitable de mettre en œuvre le changement d'algorithme de façon à ce que les nouveaux jetons générés utilisent le nouvel algorithme, mais que pendant un délai limité les jetons qui utilisent soit le nouveau soit l'ancien algorithme puissent être validés. Par conséquent, le serveur DEVRAIT incorporer un moyen pour identifier l'algorithme de génération de jeton au sein du jeton.

Bien que la présente spécification soit extensible pour d'autres mécanismes, aucun n'est défini dans le présent document. En plus du nom du mécanisme lui-même, d'autres mécanismes peuvent avoir des données spécifiques du mécanisme, qui sont à interpréter conformément à la définition de ce mécanisme.

2.5 Jeton d'autorisation

Le jeton d'autorisation est une chaîne déterministe d'au moins 128 bits dont une entité qui a connaissance de la clé d'accès de boîte aux lettres secrète et du mécanisme d'autorisation d'URL va pouvoir se servir pour vérifier l'URL.

3. Extensions d'URL IMAP

La [RFC2192] est étendue en permettant l'ajout de ";EXPIRE=<datetime>" et ";URLAUTH=<access>:<mech>:<token>" aux URL IMAP qui se réfèrent à un message ou parties de message spécifiques.

Le URLAUTH se compose de ";URLAUTH=<access>:<mech>:<token>" et DOIT être à la fin de l'URL.

URLAUTH ne s'applique pas, et NE DOIT PAS être utilisé avec un URL IMAP qui se réfère à un serveur IMAP entier, à une liste de boîtes aux lettres, à une boîte aux lettres IMAP entière, ou à des résultats de recherche IMAP.

Quand ";EXPIRE=<datetime>" est utilisé, cela indique la dernière date et heure où l'URL est valide. Après cette date et heure, l'URL a expiré, et les mises en œuvre de serveur DOIVENT rejeter l'URL. Si ";EXPIRE=<datetime>" n'est pas utilisé, l'URL n'a pas de date d'expiration, mais peut quand même être révoqué comme expliqué ci-dessous.

Le URLAUTH prend la forme ";URLAUTH=<access>:<mech>:<token>". Il est composé de trois parties. La portion <access> fournit les identifiants d'accès autorisé, qui peuvent restreindre les opérations et les utilisateurs à qui il est permis d'utiliser cet URL. La portion <mech> donne le mécanisme d'autorisation utilisé par le serveur IMAP pour générer le jeton d'autorisation qui suit. La portion <token> donne le jeton d'autorisation.

Le préfixe d'identifiant d'accès "submit+", suivi par un identifiant d'utilisateur, indique que seul un identifiant d'utilisateur autorisé comme entité de soumission de message au nom de l'identifiant d'utilisateur spécifié a la permission d'utiliser cet URL. Le serveur IMAP ne valide pas l'identifiant d'utilisateur spécifié mais il valide que la session IMAP a une identité d'autorisation qui est autorisée comme entité de soumission de message. L'entité de soumission de message autorisée DOIT valider l'identifiant d'utilisateur avant de contacter le serveur IMAP.

Le préfixe d'identifiant d'accès "user+", suivi par un identifiant d'utilisateur *(userid)* indique que l'utilisation de cet URL est limitée aux sessions IMAP qui sont enregistrées comme identifiant d'utilisateur spécifié (c'est-à-dire qui ont une identité d'autorisation comme cet identifiant d'utilisateur).

Note : si un mécanisme SASL qui fournit à la fois les identifiants d'autorisation et d'authentification est utilisé pour s'authentifier auprès du serveur IMAP, l'identifiant d'accès "user+" DOIT correspondre à l'identifiant d'autorisation.

L'identifiant d'accès "authuser" indique que l'utilisation de cet URL est limitée aux sessions IMAP qui sont enregistrées comme un utilisateur autorisé (c'est-à-dire, qui ont l'identité d'autorisation comme utilisateur autorisé) de ce serveur IMAP. L'utilisation de cet URL est interdite aux sessions IMAP anonymes.

L'identifiant d'accès "anonymous" indique que l'utilisation de cet URL n'est pas restreinte par l'identité d'autorisation de session ; c'est-à-dire que toute session IMAP en état authentifié ou choisi (comme défini dans la [RFC3501]) incluant les sessions anonymes, peut produire un URLFETCH utilisant cet URL.

Le jeton d'autorisation est représenté comme une chaîne hexadécimale codée en ASCII, qui est utilisée pour autoriser l'URL. La longueur et le calcul du jeton d'autorisation dépendent du mécanisme utilisé ; mais dans tous les cas, le jeton d'autorisation fait au moins 128 bits (et donc au moins 32 chiffres hexadécimaux).

4. Discussion des questions d'autorisation de URLAUTH

Dans la [RFC2192], le userid avant le "@" dans l'URL a deux objets :
1) Il fournit le contexte pour des chemins de boîte aux lettres spécifiques de l'utilisateur comme "INBOX".
2) Il spécifie que la résolution de l'URL exige de s'enregistrer comme cet utilisateur et limite l'utilisation de cet URL à ce seul utilisateur.

Une limitation évidente d'utiliser le même champ pour deux objets est que l'URL ne peut être résolu que par le propriétaire de la boîte aux lettres.

URLAUTH outrepasse le second objet de l'identifiant d'utilisateur dans l'URL IMAP et permet par défaut que l'URL soit résolu par tout utilisateur permis par l'identifiant d'accès.

L'identifiant d'accès "user+<userid>" limite la résolution de cet URL à un identifiant d'utilisateur particulier, tandis que l'identifiant d'accès "submit+<userid>" est plus général et exige simplement que la session soit autorisée par un utilisateur à qui a été accordé le rôle "submit" au sein du système d'authentification. L'utilisation de l'un ou l'autre de ces identifiants d'accès rend impossible à un attaquant, qui espionne la session, d'utiliser le même URL, soit directement, soit par la soumission à une entité de soumission de message.

Les identifiant d'accès"authuser" et "anonymous" n'ont pas ce niveau de protection et devraient être utilisés avec prudence. Ces identifiants d'accès sont principalement utiles pour l'exportation publique de données à partir d'un serveur IMAP, sans exiger qu'il soit copié sur un serveur de la Toile ou un serveur FTP anonyme. Voir plus de détails dans les "Considérations sur la sécurité".

5. Génération des URL autorisés par URLAUTH

Un URL autorisé par URLAUTH est généré à partir d'un URL initial comme suit :

Un URL initial est construit, se terminant par ";URLAUTH=<access>" mais sans les composants ":<mech>:<token>. Un mécanisme d'autorisation est choisi et utilisé pour calculer le jeton d'autorisation, avec l'URL initial comme données et un secret connu du serveur IMAP comme clé. L'URL autorisé par URLAUTH est généré en prenant l'URL initial et en ajoutant ":", le nom du mécanisme d'autorisation d'URL, ":", et la représentation hexadécimale codée en ASCII du jeton d'autorisation.

Note : l'hexadécimal codé en ASCII est utilisé à la place du BASE64 parce que une représentation BASE64 peut avoir des caractères de bourrage "=", ce qui pourrait poser des problèmes dans un URL.

Dans le mécanisme INTERNAL, la clé d'accès de boîte aux lettres pour cette boîte aux lettres est le secret connu du serveur IMAP, et un algorithme choisi par le serveur est utilisé comme décrit au paragraphe 2.4.1.

6. Validation des URL autorisés par URLAUTH

Un URL autorisé par URLAUTH est validé comme suit :

L'URL est partagé au ":" qui sépare "<access>" de "<mech>:<token>" dans la portion ";URLAUTH=<access>:<mech>:<token>" de l'URL. La portion "<mech>:<token>" est d'abord analysée et sauvegardée comme mécanisme d'autorisation et jeton d'autorisation. L'URL est tronqué, en éliminant le ":" décrit ci-dessus, pour créer un "URL croupion" (l'URL moins le ":" et la portion "<mech>:<token>"). L'URL croupion est alors analysé pour identifier la boîte aux lettres.

Si la boîte aux lettres ne peut pas être identifiée, un jeton d'autorisation est calculé sur l'URL croupion, en utilisant des clés aléatoires "plausibles" (choisies par le serveur) en tant que de besoin, avant de retourner un échec de validation. Cela empêche des attaques de temporisation visant à identifier les noms de boîtes aux lettres.

Si la boîte aux lettres peut être identifiée, le jeton d'autorisation est calculé sur l'URL croupion et un secret connu du serveur IMAP en utilisant le mécanisme d'autorisation d'URL donné. La validation réussit si et seulement si le jeton d'autorisation calculé pour ce mécanisme correspond au jeton d'autorisation fourni dans ";URLAUTH=<access>:<mech>:<token>".

La suppression de la portion ":<mech>:<token>" de l'URL DOIT être la seule opération appliquée à l'URL autorisé par URLAUTH pour obtenir l'URL croupion. En particulier, un décodage d'échappement en pourcentage d'URL et un changement de casse (incluant la partie domaine de l'URL) NE DOIT PAS se produire.

Dans le mécanisme INTERNAL, la clé d'accès de boîte aux lettres pour cette boîte aux lettres est utilisée comme secret connu du serveur IMAP, et le même algorithme choisi par le serveur qui est utilisé pour générer les URL est utilisé pour calculer le jeton d'autorisation pour la vérification.

7. Commandes supplémentaires

Ces commandes sont des extensions au protocole de base de la [RFC3501].

Les têtes de section de ces commandes sont destinées à correspondre à ce qu'elles auraient été si elles étaient situées dans le document de base du protocole.

BASE.6.3.RESETKEY. Commande RESETKEY
Arguments : nom facultatif de boîte aux lettres, noms de mécanismes facultatifs
Réponses : aucune autre que dans le résultat.
Résultat : OK - RESETKEY achevé, URLMECH contenant de nouvelles données
NO - erreur de RESETKEY : ne peut pas changer la clé de cette boîte aux lettres
BAD - commande inconnue ou arguments invalides

La commande RESETKEY a deux formes.
La première forme accepte un nom de boîte aux lettres comme argument et génère une nouvelle clé d'accès de boîte aux lettres pour la boîte aux lettres concernée dans le tableau de clés d'accès de boîte aux lettres de l'utilisateur, remplaçant toute clé d'accès de boîte aux lettres antérieure (et révoquant tous les URL autorisés avec un URLAUTH utilisant cette clé) dans ce tableau. Par défaut, la clé d'accès de boîte aux lettres est générée pour le mécanisme INTERNAL ; d'autres mécanismes peuvent être spécifiés avec l'argument de mécanisme facultatif.
La seconde forme, sans argument, supprime toutes les clés d'accès de boîte aux lettres dans le tableau des clés d'accès de boîte aux lettres de l'utilisateur, révoquant tous les URL actuellement autorisés en utilisant URLAUTH par l'utilisateur.

Toute session IMAP en cours enregistrée sous le nom de l'utilisateur qui a la boîte aux lettres choisie va recevoir une réponse OK non étiquetée avec le code de réponse d'état URLMECH (voir la section BASE.7.1.URLMECH pour les détails sur le code de réponse d'état URLMECH).

Exemple :
C: a31 RESETKEY
S: a31 OK Toutes les clés sont supprimées
C: a32 RESETKEY INBOX
S: a32 OK [URLMECH INTERNAL] mechs
C: a33 RESETKEY INBOX XSAMPLE
S: a33 OK [URLMECH INTERNAL XSAMPLE=P34OKhO7VEkCbsiYY8rGEg==] fait

BASE.6.3.GENURLAUTH. Commande GENURLAUTH
Argument : une ou plusieurs paires URL/mécanisme
Réponse : réponse non étiquetée : GENURLAUTH
Résultat : OK - GENURLAUTH achevé
NO - erreur de GENURLAUTH : ne peut générer un URLAUTH
BAD - commande inconnue ou arguments invalides

La commande GENURLAUTH demande que le serveur génère un URL autorisé par URLAUTH pour chacun des URL donnés qui utilisent le mécanisme d'autorisation d'URL donné.

Le serveur DOIT valider chaque URL fourni comme suit :
(1) Le composant de boîte aux lettres de l'URL DOIT se référer à une boîte aux lettres existante.
(2) Le composant de serveur de l'URL DOIT contenir un identifiant d'utilisateur *(userid)* valide qui identifie le propriétaire du tableau de clés d'accès de boîte aux lettres qui sera utilisé pour générer l'URL autorisé par URLAUTH. Par conséquent, la règle iserver de la [RFC2192] est modifiée de telle sorte que iuserauth est obligatoire.
Note : le composant de serveur de l'URL est généralement l'identifiant d'utilisateur qui s'enregistre et le serveur. Sinon, l'identifiant d'utilisateur et le serveur enregistrés DOIVENT avoir l'accès de type propriétaire au tableau de clés d'accès de boîte aux lettres possédé par l'identifiant d'utilisateur et serveur indiqués par le composant de serveur de l'URL.
(3) Il y a un identifiant d'accès valide qui, dans le cas de "submit+" et "user+", va contenir un identifiant d'utilisateur valide. Cet identifiant d'utilisateur n'est pas nécessairement le même que l'identifiant de propriétaire décrit en (2).
(4) Le serveur PEUT aussi vérifier que les composants iuid et/ou isection (si ils sont présents) sont valides.

Si une des vérifications ci-dessus échoue, le serveur DOIT retourner une réponse BAD étiquetée avec l'exception suivante. Si un userid invalide est fourni comme propriétaire de clé d'accès de boîte aux lettres et/ou au titre de l'identifiant d'accès, le serveur PEUT produire une réponse OK étiquetée avec une clé de boîte aux lettres générée qui échoue toujours à la validation quand elle est utilisée avec une commande URLFETCH. Cette exception empêche un attaquant de valider les identifiants d'utilisateur.

Si il n'y a actuellement pas de clé d'accès de boîte aux lettres pour la boîte aux lettres concernée dans le tableau de clés d'accès de boîte aux lettres du propriétaire, il en est généré une automatiquement. C'est-à-dire qu'il n'est pas nécessaire d'utiliser RESETKEY avant la première utilisation de GENURLAUTH.

Si la commande réussit, un code de réponse GENURLAUTH est retourné qui fait la liste des URL demandés comme URL autorisés par URLAUTH.

Exemples :
C: a775 GENURLAUTH "imap://joe@exemple.com/INBOX/;uid=20/ ; section=1.2" INTERNAL
S: a775 BAD identifiant d'accès manquant dans l'URL fourni

C: a776 GENURLAUTH "imap://exemple.com/Shared/;uid=20/ ; section=1.2;urlauth=submit+fred" INTERNAL
S: a776 BAD propriétaire du nom d'utilisateur manquant dans l'URL fourni
C: a777 GENURLAUTH "imap://joe@exemple.com/INBOX/;uid=20/ ; section=1.2;urlauth=submit+fred" INTERNAL
S: * GENURLAUTH "imap://joe@example.com/INBOX/;uid=20/ ; section=1.2
;urlauth=submit+fred:internal:91354a473744909de610943775f92038"
S: a777 OK GENURLAUTH terminé

BASE.6.3.URLFETCH. Commande URLFETCH
Argument : un ou plusieurs URL
Réponse : réponse non étiquetée : URLFETCH
Résultat : OK - urlfetch terminé
NO - échec de urlfetch à cause d'une erreur interne du serveur
BAD - commande inconnue ou arguments invalides

La commande URLFETCH demande que le serveur retourne les données de texte associées aux URL IMAP spécifiés, comme décrit dans la [RFC2192] et étendues par ce document. Les données sont retournées pour tous les URL validés, sans considérer si la session serait par ailleurs capable ou non d'accéder à la boîte aux lettres contenant ces données via SELECT ou EXAMINE.

Note : cette commande n'exige pas que l'URL se réfère à la boîte aux lettres choisie ; ni n'exige qu'une boîte aux lettres soit choisie. Elle n'interfère non plus avec aucune boîte aux lettres choisie.

La commande URLFETCH s'exécute effectivement avec l'accès de l'identifiant d'utilisateur dans le composant de serveur de l'URL (qui est généralement l'identifiant d'utilisateur qui a produit le GENURLAUTH). Par lui-même, le URLAUTH N'accorde PAS l'accès aux données ; une fois validé, il accorde tout accès aux données qui est détenu par l'identifiant d'utilisateur dans le composant de serveur de l'URL. Cet accès peut avoir changé depuis que le GENURLAUTH a été fait.

La commande URLFETCH DOIT retourner une réponse URLFETCH non étiquetée et une réponse OK étiquetée à toute commande URLFETCH syntaxiquement valide. Une réponse NO indique une défaillance interne du serveur qui peut se résoudre par un essai ultérieur.

Note : La possibilité d'une réponse NO est pour s'accommoder des mises en œuvre qui devraient autrement produire un BYE non étiqueté avec une erreur fatale due à une incapacité de répondre à une demande valide. Idéalement, un serveur NE DEVRAIT PAS produire une réponse NO.

Le serveur DOIT retourner NIL pour tout URL IMAP qui fait référence à un serveur IMAP entier, une liste de boîtes à lettres, une boîte à lettres IMAP entière, ou des résultats de recherches IMAP.

Exemple :
Note : pour être clair, cet exemple utilise la commande LOGIN, qui NE DEVRAIT PAS être utilisée sur un chemin de communication non chiffré.

Cet exemple est d'un serveur de soumission, qui obtient un segment de message pour un message qu'il a déjà validé soumis par "fred".

S: * OK [CAPABILITY IMAP4REV1 URLAUTH] exemple.com serveur IMAP
C: a001 LOGIN submitserver secret
S: a001 OK submitserver connecté
C: a002 URLFETCH "imap://joe@exemple.com/INBOX/;uid=20/
;section=1.2;urlauth=submit+fred:internal91354a473744909de610943775f92038"
S: * URLFETCH "imap://joe@exemple.com/INBOX/;uid=20/;section=1.2
;urlauth=submit+fred:internal
:91354a473744909de610943775f92038" {28}
S: Si vis pacem, para bellum.
S:
S: a002 OK URLFETCH terminé

8. Réponses supplémentaires

Ces réponses sont des extensions au protocole de base de la [RFC3501].

Les têtes de section de ces réponses sont destinées à correspondre à la situation qu'elles auraient eu dans le document du protocole de base, si elles en avaient fait partie.

BASE.7.1.URLMECH. Code de réponse d'état URLMECH
Le code de réponse d'état URLMECH est suivi par une liste de noms de mécanismes d'autorisation d'URL. Les noms de mécanismes autres que INTERNAL peuvent être ajoutés avec un "=" et une forme codée en BASE64 des données spécifiques du mécanisme.

Ce code de réponse d'état est retourné dans une réponse OK non étiquetée en réponse à une commande RESETKEY, SELECT, ou EXAMINE. Dans le cas d'une réponse RESETKEY, ce code de réponse d'état peut être envoyé dans la réponse OK étiquetée au lieu d'exiger une réponse OK non étiquetée séparée.

Exemple :
C: a33 RESETKEY INBOX XSAMPLE
S: a33 OK [URLMECH INTERNAL XSAMPLE=P34OKhO7VEkCbsiYY8rGEg==] fait

Dans cet exemple, le serveur prend en charge le mécanisme INTERNAL et un mécanisme expérimental appelé XSAMPLE, qui contient aussi des données spécifiques du mécanisme (le nom "XSAMPLE" est seulement une illustration).

BASE.7.4.GENURLAUTH. Réponse GENURLAUTH
Contenu : un ou plusieurs URL.
La réponse GENURLAUTH retourne le ou les URL autorisés par URLAUTH demandés par la commande GENURLAUTH.

Exemple :
C: a777 GENURLAUTH "imap://joe@exemple.com/INBOX/;uid=20/;section=1.2;urlauth=submit+fred" INTERNAL
S: * GENURLAUTH "imap://joe@exemple.com/INBOX/;uid=20/;section=1.2
;urlauth=submit+fred:internal:91354a473744909de610943775f92038"
S: a777 OK GENURLAUTH terminé

BASE.7.4.URLFETCH. Réponse URLFETCH
Contenu : une ou plusieurs paires URL/nstring
La réponse URLFETCH retourne les données de texte du message associées à un ou plusieurs URL IMAP, comme décrit dans la [RFC2192] et étendu par le présent document. Cette réponse est le résultat d'une commande URLFETCH .

La chaîne de données retournée est NIL si l'URL est invalide pour une raison quelconque (incluant un échec de validation). Si l'URL est valide, mais si la restitution de la partie corps par IMAP a retourné NIL (cela ne devrait pas se produire) la chaîne de données retournée devrait être la chaîne vide ("") et pas NIL.

Note : cette commande n'exige pas que l'URL se réfère à la boîte aux lettres sélectionnée ; ni qu'une boîte au lettres soit sélectionnée. Elle n'interfère en aucune façon avec une boîte aux lettre choisie quelconque.

Exemple :
C: a002 URLFETCH "imap://joe@exemple.com/INBOX/;uid=20/
;section=1.2;urlauth=submit+fred:internal:91354a473744909de610943775f92038"
S: * URLFETCH "imap://joe@exemple.com/INBOX/;uid=20/;section=1.2
;urlauth=submit+fred:internal:91354a473744909de610943775f92038" {28}
S: Si vis pacem, para bellum.
S:
S: a002 OK URLFETCH terminé

9. Syntaxe formelle

La spécification de syntaxe suivante utilise la notation en forme Backus-Naur augmentée (ABNF) comme spécifié dans la [RFC4234].

Les modifications suivantes sont apportées à la syntaxe formelle de la [RFC3501] :

```
resetkey = "RESETKEY" [SP boîte aux lettres *(SP mécanisme)]
capability  =/ "URLAUTH"
command-auth =/ resetkey / genurlauth / urlfetch
resp-text-code  =/ "URLMECH" SP "INTERNAL" *(SP mécanisme ["=" base64])
genurlauth = "GENURLAUTH" 1*(SP url-rump SP mécanisme)
genurlauth-data = "*" SP "GENURLAUTH" 1*(SP url-full)
url-full = astring                                  ; contient authimapurlfull comme défini ci-dessous
url-rump = astring                                  ; contient authimapurlrump comme défini ci-dessous
urlfetch = "URLFETCH" 1*(SP url-full)
urlfetch-data = "*" SP "URLFETCH" 1*(SP url-full SP nstring)
```

Les extensions suivantes sont faites à la syntaxe formelle de la [RFC2192] :

```
authimapurl = "imap://" enc-user [iauth] "@" hostport "/" imessagepart
                                    ; remplace les règles "imapurl" et "iserver" pour les URL autorisés par URLAUTH
authimapurlfull = authimapurl iurlauth
authimapurlrump = authimapurl iurlauth-rump
enc-urlauth = 32*HEXDIG
enc-user = 1*achar                                  ; le même que "enc_user" dans la RFC 2192
iurlauth = iurlauth-rump ":" mécanisme ":" enc-urlauth
iurlauth-rump = [expire] ";URLAUTH=" accès
accès = ("submit+" enc-user) / ("user+" enc-user) / "authuser" / "anonyme"
expire = ";EXPIRE=" date-heure                      ; date-heure définie dans la [RFC3339]
mécanisme = "INTERNAL" / 1*(ALPHA / DIGIT / "-" / ".")
                                   ; insensible à la casse ; les nouveaux mécanismes DOIVENT être enregistrés par l'IANA.
```

10. Considérations sur la sécurité

Les considérations de sécurité sont discutées tout au long de ce mémoire.

La clé d'accès de boîte aux lettres DEVRAIT avoir au moins 128 bits d'entropie (se reporter à la [RFC4086] pour les détails) et DOIT être imprévisible.

La mise en œuvre de serveur du mécanisme INTERNAL DEVRAIT envisager la possibilité du besoin de changer l'algorithme de génération de jeton, et DEVRAIT incorporer un moyen d'identifier l'algorithme de génération de jeton au sein du jeton.

Le code de réponse d'état URLMECH peut exposer des données sensibles dans les données spécifiques de mécanisme pour les mécanismes autres que INTERNAL. Une mise en œuvre de serveur DOIT mettre en œuvre une configuration qui ne va pas retourner un code de réponse d'état URLMECH si n'est pas fourni un mécanisme qui protège la session de l'espionnage, comme la couche de sécurité TLS ou SASL qui assure la protection de la confidentialité.

Le calcul d'un jeton d'autorisation avec une clé "plausible" si la boîte aux lettres ne peut pas être identifiée est nécessaire pour éviter des attaques dans lesquelles le serveur est sondé pour voir si une certaine boîte aux lettres existe sur le serveur en mesurant la quantité de temps que prend le rejet d'un mauvais nom connu par rapport à un autre nom.

Pour protéger contre une attaque de déni de service fondée sur la capacité de calcul, un serveur PEUT imposer des délais progressivement plus longs sur des demandes portant sur plusieurs URL qui échouent à la validation.

La décision d'utiliser l'identifiant d'accès "authuser" devrait être prise avec prudence. Un identifiant d'accès "authuser" peut être utilisé par tout utilisateur autorisé par le serveur IMAP ; donc, l'utilisation de cet identifiant d'accès devrait être limité aux contenus qui peuvent être divulgués à tout utilisateur autorisé du serveur IMAP.

La décision d'utiliser l'identifiant d'accès "anonyme" devrait être prise avec une extrême prudence. Un identifiant d'accès "anonyme" peut être utilisé par n'importe qui ; donc, l'utilisation de cet identifiant d'accès devrait être limitée aux contenus qui peuvent être divulgués à tous. De nombreux serveurs IMAP ne permettent pas l'accès anonyme ; dans ce cas, l'identifiant "anonyme" est équivalent à "authuser", mais on NE DOIT PAS s'y fier.

Bien que la présente spécification n'interdise pas la capacité théorique de générer un URL avec un composant de serveur autre que l'identifiant d'utilisateur et serveur enregistré, cette capacité ne devrait être fournie que lorsque l'identifiant d'utilisateur/serveur enregistré a été autorisé comme équivalent au composant de serveur userid/serveur, ou a par ailleurs accès au tableau de clés d'accès de boîte aux lettres de cet userid/serveur.

11. Considérations relatives à l'IANA

Le présent document constitue l'enregistrement de la capacité URLAUTH dans le registre imap4-capabilities.

Les mécanismes d'autorisation URLAUTH sont enregistrés par la publication d'une RFC sur la voie de la normalisation ou d'une RFC expérimentale approuvée par l'IESG. Le registre est actuellement situé à : http://www.iana.org/assignments/urlauth-authorization-mechanism-registry

Ce registre est insensible à la casse.

Le présent document constitue l'enregistrement du mécanisme d'autorisation URLAUTH INTERNAL.

Registre des mécanismes d'autorisation IMAP URLAUTH :

Nom du mécanisme	**Référence**
INTERNAL	[RFC4467]

12. Références normatives

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

[RFC2192] C. Newman, "Schéma d'URL IMAP", septembre 1997. (*Obsolète, voir* RFC5092) *(P.S.)*

[RFC3339] G. Klyne, C. Newman, "La date et l'heure sur l'Internet : horodatages", juillet 2002. *(P.S.)*

[RFC3501] M. Crispin, "Protocole d'accès au message Internet - version 4rev1", mars 2003. *(P.S. ; MàJ par* RFC4466, 4469, 4551, 5032, 5182, 7817, 8314, 8437, 8474)

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

[RFC4468] C. Newman, "Extension BURL de soumission de message", mai 2006. (*MàJ* RFC3463) (*MàJ par* RFC5248) *(P.S.)*

13. Références pour information

[RFC2104] H. Krawczyk, M. Bellare et R. Canetti, "HMAC : Hachage de clés pour l'authentification de message", février 1997.

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005. (*Remplace* RFC1750) (BCP0106)

Adresse de l'auteur

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Avenue NE
Seattle, WA 98105-4527

téléphone : (206) 543-5762
mél : MRC@CAC.Washington.EDU

Déclaration complète de droits de reproduction

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