Groupe de travail Réseau
Request for Comments : 4468
RFC mise à jour : 3463
Catégorie : Sur la voie de la normalisation

C. Newman, Sun Microsystems

mai 2006
Traduction Claude Brière de L'Isle

Extension BURL de soumission de message

Résumé
Le profil de soumission du protocole simple de transfert de messagerie (SMTP, *Simple Messaging Transfer Protocol*) fournit un moyen standardisé pour qu'un client de messagerie électronique soumette un message complet à livrer. La présente spécification étend le profil de soumission en ajoutant une nouvelle commande BURL qui peut être utilisée pour aller cherche les données de soumission auprès d'un serveur de protocole d'accès au message Internet (IMAP, *Internet Message Access Protocol*). Cela permet à un client de messagerie d'injecter le contenu provenant d'un serveur IMAP dans l'infrastructure SMTP sans le télécharger au client et de le télécharger à nouveau en retour au serveur.

Table des matières

1. Introduction

La présente spécification définit une extension à la norme de protocole de soumission de message [RFC4409] pour permettre d'aller chercher les données auprès d'un serveur IMAP au moment de la soumission d'un message. Ceci PEUT être utilisé en conjonction avec le mécanisme CHUNKING [RFC3030] afin que ces tronçons de message puissent venir d'un serveur IMAP externe. Cela donne la capacité de transmettre un message électronique sans le télécharger d'abord au client.

2. 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 en forme Backus-Naur augmentée (ABNF) [RFC4234] incluant le cœur des règles définies dans l'Appendice B de la RFC 4234.

3. Extension de soumission BURL

Cette Section définit l'extension de soumission BURL.

3.1 Enregistrement d'extension de soumission SMTP

1. Le nom de cette extension de soumission est "BURL". Cela étend le protocole de soumission de message sur l'accès 587 et NE DOIT PAS être annoncé par un serveur SMTP régulier [RFC2821] sur l'accès 25 qui agit comme relais pour les messages entrants provenant d'autres relais SMTP.

2. La valeur du mot clé EHLO associée à l'extension est "BURL".

3. Le mot clé EHLO BURL aura zéro, un ou plusieurs arguments. Le seul argument défini pour l'instant est l'argument "imap", qui DOIT être présent afin d'utiliser les URL IMAP avec BURL. Les clients DOIVENT ignorer les autres arguments après le mot clé EHLO BURL sauf si ils sont définis par une spécification ultérieure de l'IETF sur la voie de la normalisation. Les arguments qui apparaissent après le mot clé EHLO BURL peuvent changer suite à l'utilisation de AUTH SMTP [RFC2554], de sorte qu'un serveur qui annonce BURL sans argument avant l'authentification indique que BURL est pris en charge mais qu'il est exigé que l'authentification l'utilise.

4. Cette extension ajoute le verbe SMTP BURL. Ce verbe est utilisé en remplacement de la commande DATA et n'est permis que durant une transaction de messagerie électronique après au moins un RCPT TO réussi.

3.2 Transaction BURL

Une transaction BURL simple va consister en un MAIL FROM, un ou plusieurs en-têtes RCPT TO, et une commande BURL avec l'étiquette "LAST". La commande BURL va inclure un URL IMAP pointant sur un message pleinement formé prêt à être injecté dans l'infrastructure SMTP. Si PIPELINING [RFC2920] est annoncé, le client PEUT envoyer la transaction entière en un aller-retour. Si aucune adresse RCPT TO valide n'est fournie, la commande BURL va simplement échouer, et aucune résolution de l'argument d'URL BURL ne sera effectué. Si au moins une adresse RCPT TO valide est fournie, alors l'argument d'URL BURL sera résolu avant que le serveur réponde à la commande.

Une transaction BURL plus sophistiquée PEUT survenir quand le serveur annonce aussi CHUNKING [RFC3030]. Dans ce cas, les commandes BURL et BDAT peuvent être entrelacées jusqu'à ce qu'une d'elles termine la transaction avec l'argument "LAST". Si PIPELINING [RFC2920] est aussi annoncé, alors le client peut traiter en parallèle la transaction entière en un aller retour. Cependant, il DOIT attendre le résultat de la commande BDAT ou BURL "LAST" avant d'initier une nouvelle transaction.

La commande BURL ordonne au serveur d'aller chercher l'objet de données auquel se réfère l'URL et l'inclure dans le message. Si la recherche d'URL échoue, le serveur va faire échouer la transaction entière.

3.3 Options IMAP BURL

Quand "imap" est présent dans la liste séparée par des espaces d'arguments suivant le mot clé BURL EHLO, il indique que la commande BURL prend en charge la forme étendue URLAUTH [RFC4467] des URL IMAP [RFC2192] et que le serveur de soumission est configuré avec les accréditifs nécessaires pour résoudre les URL IMAP "urlauth=submit+" pour le domaine du serveur de soumission.

Suite à une commande SMTP AUTH réussie, le serveur de soumission PEUT indiquer une relation de confiance pré-

arrangée avec un serveur IMAP spécifique en incluant un argument de mot clé BURL EHLO de la forme "imap://imap.exemple.com". Dans ce cas, le serveur de soumission va permettre un URL IMAP régulier se référant aux messages ou parties de messages sur imap.exemple.com auquel l'utilisateur qui s'est authentifié auprès du serveur de soumission peut accéder. Noter que cette forme n'implique pas que le serveur de soumission prenne en charge les URL URLAUTH ; le serveur de soumission doit annoncer "imap" et "imap://imap.exemple.com" pour indiquer la prise en charge des deux formes étendue et non étendue d'URL.

Quand le serveur de soumission se connecte au serveur IMAP, il agit comme un client IMAP et est donc soumis à la fois aux capacités IMAP de mise en œuvre obligatoire du paragraphe 6.1.1 de la RFC 3501, et aux considérations sur la sécurité de la Section 11 de la RFC 3501. Précisément, ceci exige que le serveur de soumission mette en œuvre une configuration qui utilise STARTTLS suivi par SASL PLAIN [RFC4616] pour s'authentifier auprès du serveur IMAP.

Quand le serveur de soumission résout un URL URLAUTH IMAP, il utilise les accréditifs du serveur de soumission quand il s'authentifie auprès du serveur IMAP. L'identité et le mot de passe d'authentification utilisés pour soumettre les accréditifs DOIVENT être configurables. La chaîne "submit" est suggérée comme valeur par défaut pour l'identité d'authentification, mais pas de valeur par défaut pour le mot de passe. Normalement, l'identité d'autorisation est vide dans ce cas ; donc le serveur IMAP va déduire l'identité d'autorisation de l'identité d'authentification. Si l'URL IMAP utilise le préfixe d'identifiant d'accès "submit+", le serveur de soumission DOIT refuser la commande BURL sauf si le userid dans le jeton <access> de l'URL correspond à l'identité d'autorisation du client de soumission.

Quand le serveur de soumission résout un URL IMAP régulier, il utilise l'identité d'autorisation du client de soumission quand il s'authentifie auprès du serveur IMAP. Si le client de soumission et le client IMAP incorporé du serveur de soumission utilisent tous deux SASL PLAIN (ou un équivalent) le serveur de soumission DEVRAIT transmettre les accréditifs du client si et seulement si le serveur de soumission sait que le serveur IMAP est dans le même domaine administratif. Si le serveur de soumission prend en charge des mécanismes SASL autres que PLAIN, il DOIT mettre en œuvre une configuration dans laquelle le client IMAP incorporé du serveur de soumission utilise STARTTLS et SASL PLAIN avec l'identité d'authentification et le mot de passe du serveur de soumission (pour le serveur IMAP concerné) et l'identité d'autorisation du client de soumission.

3.4 Exemples

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

Deux soumissions réussies (avec et sans traitement en parallèle) suivent :

<SSL/TLS couche de chiffrement négociée>
C: EHLO potter.exemple.com
S: 250-owlry.exemple.com
S: 250-8BITMIME
S: 250-BURL imap
S: 250-AUTH PLAIN
S: 250-DSN
S: 250 ENHANCEDSTATUSCODES
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
S: 235 2.7.0 PLAIN authentification réussie.
C: MAIL FROM:<harry@gryffindor.exemple.com>
S: 250 2.5.0 Adresse OK.
C: RCPT TO:<ron@gryffindor.exemple.com>
S: 250 2.1.5 ron@gryffindor.exemple.com OK.
C: BURL imap://harry@gryffindor.exemple.com/outbox
;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
:internal:91354a473744909de610943775f92038 LAST
S: 250 2.5.0 OK.

<SSL/TLS couche de chiffrement négociée>
C: EHLO potter.exemple.com
S: 250-owlry.exemple.com
S: 250-8BITMIME

S: 250-PIPELINING
S: 250-BURL imap
S: 250-AUTH PLAIN
S: 250-DSN
S: 250 ENHANCEDSTATUSCODES
C: AUTH PLAIN aGFycnkAaGFycnkAYWNjaW8=
C: MAIL FROM:<harry@gryffindor.exemple.com>
C: RCPT TO:<ron@gryffindor.exemple.com>
C: BURL imap://harry@gryffindor.exemple.com/outbox
;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
:internal:91354a473744909de610943775f92038 LAST
S: 235 2.7.0 PLAIN authentification réussie.
S: 250 2.5.0 Adresse OK.
S: 250 2.1.5 ron@gryffindor.exemple.com OK.
S: 250 2.5.0 OK.

Noter que le PIPELINING de la commande AUTH n'est permis que si le mécanisme choisi peut être achevé en un seul aller-retour, qu'une réponse initiale du client est fournie, et qu'aucune couche de sécurité SASL n'est négociée. C'est possible pour PLAIN et EXTERNAL, mais pas pour la plupart des autres mécanismes SASL.

Quelques exemples de cas d'échec :

C: MAIL FROM:<harry@gryffindor.exemple.com>
C: RCPT TO:<malfoy@slitherin.exemple.com>
C: BURL imap://harry@gryffindor.exemple.com/outbox
;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
:internal:91354a473744909de610943775f92038 LAST
S: 250 2.5.0 Adresse OK.
S: 550 5.7.1 Relais non admis : malfoy@slitherin.exemple.com
S: 554 5.5.0 Aucun receveur n'a été spécifié.

C: MAIL FROM:<harry@gryffindor.exemple.com>
C: RCPT TO:<ron@gryffindor.exemple.com>
C: BURL imap://harry@gryffindor.exemple.com/outbox
;uidvalidity=1078863300/;uid=25;urlauth=submit+harry
:internal:71354a473744909de610943775f92038 LAST
S: 250 2.5.0 Adresse OK.
S: 250 2.1.5 ron@gryffindor.exemple.com OK.
S: 554 5.7.0 Échec d'autorisation de l'URL IMAP

3.5 Syntaxe formelle

La spécification de syntaxe suivante hérite de l'ABNF [RFC4234] et des identifiants de ressource universels [RFC3986].

burl-param = "imap" / ("imap://" authority) ; paramètre du mot clé BURL EHLO

burl-cmd = "BURL" SP URI-absolu [SP marqueur de fin] CRLF

marqueur de fin = "LAST"

4. 8-Bit et binaire

Un serveur de soumission qui annonce BURL DOIT aussi annoncer 8BITMIME [RFC1652] et effectuer la conversion décrite dans cette spécification sur le message complet résultant si des données en 8 bits sont reçues avec la commande BURL et passées à un serveur à 7 bits. Si l'argument d'URL pour BURL se réfère à des données binaires, le serveur de soumission PEUT alors refuser la commande ou convertir comme décrit dans SMTP binaire [RFC3030].

Le serveur de soumission PEUT refuser d'accepter une commande BURL ou une combinaison de commandes BURL et

BDAT qui résulte en données non codées sur 8 bits dans le message ou dans les en-têtes MIME [RFC2045]. Autrement, le serveur PEUT accepter de telles données et convertir en le codage d'en-tête MIME de la [RFC2047].

5. Mises à jour à la RFC 3463

Les serveurs SMTP ou de soumission qui annoncent ENHANCEDSTATUSCODES [RFC2034] utilisent les codes d'état améliorés définis dans la [RFC3463]. L'extension BURL introduit de nouveaux cas d'erreur que ne considérait pas cette RFC. Les codes d'état améliorés supplémentaires suivants sont définis par la présente spécification :

X.6.6 Contenu de message non disponible
Le contenu du message n'a pas pu être récupéré sur le système distant. Cela peut être utile comme notification permanente ou temporaires persistante.

X.7.14 Relation de confiance exigée
Le serveur de soumission exige que soit configurée une relation de confiance avec un serveur tiers afin d'accéder au contenu du message.

6. Codes de réponse

Cette Section comporte des exemples de codes de réponse à la commande BURL. Un autre texte peut être utilisé avec les mêmes codes de réponse. Cette liste n'est pas exhaustive, et les clients BURL DOIVENT tolérer tout code de réponse SMTP valide. La plupart de ces exemples comportent le code d'état amélioré approprié [RFC3463].

554 5.5.0 Aucun receveur n'a été spécifié
Ce code de réponse survient lorsque BURL est utilisé (par exemple, avec PIPELINING) et que tous les RCPT TO ont échoué.

503 5.5.0 Un RCPT TO valide est exigé avant BURL
Ce code de réponse est une solution de remplacement du précédent lorsque BURL est utilisé (par exemple, avec PIPELINING) et que tous les RCPT TO ont échoué.

554 5.6.3 Conversion exigée mais non prise en charge
Ce code de réponse survient lorsque l'URL pointe sur des données binaires et que la mise en œuvre ne prend pas en charge la conversion en base64. Cela peut aussi être utilisé si l'URL pointe sur des données de message avec un contenu en 8 bits dans les en-têtes et que le serveur ne sait pas convertir un tel contenu.

554 5.3.4 Message trop gros pour le système
Le message (suite à la résolution de l'URL) est plus grand que la limite de taille par message pour ce serveur.

554 5.7.14 La résolution de l'URL exige une relation de confiance
Le serveur de soumission n'a pas une relation de confiance avec le serveur IMAP spécifié dans l'argument d'URL pour BURL.

552 5.2.2 Boîte aux lettres pleine
Le receveur est local, le serveur de soumission prend en charge la livraison directe, et le receveur a excédé son quota et toute période de grâce pour les tentatives de livraison.

554 5.6.6 Échec de la résolution d'URL IMAP
La commande IMAP URLFETCH a retourné une erreur ou pas de données.

250 2.5.0 Attente de commandes BURL ou BDAT supplémentaires
Une commande BURL sans le modificateur "LAST" a été envoyée. L'URL pour cette commande BURL a été résolu avec succès, mais le contenu ne va pas nécessairement être affecté à une mémorisation persistante jusqu'à ce que le reste du contenu du message soit collecté. Par exemple, un serveur Unix peut avoir écrit le contenu dans une mémoire tampon de fichiers en file d'attente, mais peut n'avoir pas encore effectué une opération fsync(). Si le serveur perd son alimentation, le contenu peut encore être perdu.

451 4.4.1 Serveur IMAP indisponible
La connexion au serveur IMAP pour résoudre l'URL est défaillante.

250 2.5.0 OK.
L'URL a bien été résolu, et les données du message complet ont été affectées à une mémorisation persistante.

250 2.6.4 Conversion d'en-tête MIME effectuée avec des pertes
L'URL pointait sur des données de message qui incluaient de la messagerie ou des en-têtes MIME avec des données en 8 bits. Ces données ont été converties en codage d'en-tête MIME [RFC2047], mais le serveur de soumission peut n'avoir pas deviné correctement le jeu de caractères non étiqueté.

7. Considérations relatives à l'IANA

L'extension SMTP "BURL" comme décrite à la Section 3 a été enregistrée. Cet enregistrement a été marqué comme étant utilisé pour la soumission de message [RFC4409] seulement dans le registre.

8. Considérations sur la sécurité

Les serveurs modernes de soumission SMTP incluent souvent des mécanismes de sécurité et de défense contre le déni de service fondés sur le contenu comme le filtrage de virus, des limites de taille, des signatures générées par le serveur, le filtrage des messages non désirés, etc. Les mises en œuvre de BURL devraient aller chercher le contenu de l'URL avant l'application de tels mécanismes fondés sur le contenu afin de préserver leur fonction.

Les clients qui génèrent de la messagerie de masse non sollicitée ou des messages avec des virus pourraient utiliser ce mécanisme pour compenser une liaison lente entre le client et le serveur de soumission. En particulier, ce mécanisme rendrait faisable pour un téléphone cellulaire programmable ou autre appareil sur une liaison lente de devenir une source significative de messages non sollicités de masse et/ou de virus. Cela rend plus important que les fabricants de serveurs de soumission qui mettent en œuvre BURL aient une surveillance et/ou des défenses contre de telles attaques de déni de service incluant l'authentification obligatoire, un enregistrement qui associe des identifiants univoques de client aux transactions de messagerie électronique, des limites sur la réutilisation du même URL IMAP, des limites de débit, des limites de compte de receveur, et des filtres de contenu.

Le transfert de la forme URLAUTH [RFC4467] des URL IMAP en clair peut exposer le jeton d'autorisation à des espions sur le réseau. Les mises en œuvre qui prennent en charge de tels URL peuvent traiter ce problème en utilisant un mécanisme fort de protection de la confidentialité. Par exemple, les extensions SMTP STARTTLS [RFC3207] et IMAP STARTTLS [RFC3501], en combinaison avec un réglage de configuration qui exige leur utilisation avec de tels URL IMAP, réglerait ce problème.

L'utilisation d'une relation de confiance pré arrangée entre un serveur de soumission et un serveur IMAP spécifique introduit des problèmes de sécurité. Une compromission du serveur de soumission ne devrait pas compromettre automatiquement tous les comptes sur le serveur IMAP, de sorte que les relations de confiance impliquant des accréditifs de super utilisateur mandataire sont fortement déconseillées. Un système qui exige que le serveur de soumission s'authentifie auprès du serveur IMAP avec une soumission d'accréditifs et qui ensuite exige qu'un URL URLAUTH aille chercher un contenu règle ce problème. Un modèle de tiers de confiance pour les accréditifs de mandataire (comme celui fourni par Kerberos 5 [RFC4120]) serait aussi suffisant.

Quand un client utilise SMTP STARTTLS pour envoyer une commande BURL qui fait référence à des informations non publiques, l'utilisateur s'attend à ce que le contenu entier du message soit traité de manière confidentielle. Pour répondre à cette attente, le serveur de soumission de message DEVRAIT utiliser STARTTLS ou un mécanisme fournissant une confidentialité des données équivalente quand il va chercher le contenu référencé par cet URL.

Un utilisateur légitime d'un serveur de soumission peut essayer de compromettre d'autres compte sur le serveur en fournissant un URL IMAP URLAUTH qui pointe sur un serveur sous son contrôle qui est conçu pour saper la sécurité du serveur de soumission. Pour cette raison, le code de client IMAP que le serveur de soumission utilise doit être robuste par rapport à des tailles d'entrées arbitraires (incluant de grands littéraux IMAP) et des délais arbitraires de la part du serveur IMAP. Exiger une relation de confiance pré arrangée entre un serveur de soumission et le serveur IMAP règle aussi ce problème.

Un utilisateur autorisé du serveur de soumission pourrait établir un serveur IMAP frauduleux et passer un URL pour ce serveur au serveur de soumission. Le serveur de soumission pourrait alors contacter le serveur IMAP frauduleux pour s'authentifier avec des accréditifs de soumission et aller chercher du contenu. Il y a plusieurs moyens pour contrer cette attaque potentielle. Un serveur de soumission qui utilise seulement des accréditifs de soumission avec un ensemble fixé de serveurs IMAP de confiance ne sera pas vulnérable à l'exposition de ces accréditifs. Un serveur de soumission peut traiter le serveur IMAP comme n'étant pas de confiance et inclure des défenses contre la submersion de mémoire tampon, les ralentissements de déni de service, et autres attaques potentielles. Finalement, comme l'authentification est exigée pour utiliser BURL, il est possible de garder un chemin d'audit sécurisé et de l'utiliser pour détecter et punir l'offenseur.

9. Références

9.1 Références normatives

[RFC1652] J. Klensin et autres, "Extensions de service SMTP pour transport MIME sur 8 bits", juillet 1994. *(Remplacée par* RFC6152*) (D.S.)*

[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.)*

[RFC2554] J. Myers, "Extension de service SMTP pour l'authentification", mars 1999. (*Obsolète, voir* RFC4954) *(P.S.)*

[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*)*

[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*)*

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005. *(P.S. ; MàJ par* RFC8820*)*

[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*)*

[RFC4467] M. Crispin, "Protocole d'accès au message Internet (IMAP) - Extension URLAUTH", mai 2006. (*P.S. ; MàJ par* RFC5092*)*

9.2 Références pour information

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

[RFC2045] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. *(D. S., MàJ par* 2184, 2231, 5335.*)*

[RFC2047] K. Moore, "MIME (Extensions de messagerie Internet multi-objets) Partie trois : extensions d'en-tête de message pour texte non ASCII", novembre 1996. (*MàJ par* RFC2184, RFC2231*) (D.S.)*

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

[RFC3030] G. Vaudreuil, "Extensions de service SMTP pour la transmission de grands messages MIME binaires",

décembre 2000. *(P.S.)*

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

[RFC4120] C. Neuman et autres, "Service Kerberos d'authentification de réseau (V5)", juillet 2005. (*MàJ par* RFC4537, 5021, 6649, 7751, 8062, 8129, 8429)

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

Appendice A. Remerciements

Le présent document a été produit par le groupe de travail lemonade. Des remerciements sont dus à tous les participants à ce groupe de travail pour leurs apports. Mark Crispin a été l'nstrument de la conception de ce mécanisme. Merci à Randall Gellens, Alexey Melnikov, Sam Hartman, Ned Freed, Dave Cridland, Peter Coates, et Mark Crispin pour leurs commentaires sur le document. Merci à l'éditeur des RFC qui a corrigé les fautes de grammaire de l'auteur. Merci à Ted Hardie, Randall Gellens, Mark Crispin, Pete Resnick, et Greg Vaudreuil pour les dicussions extrêmement intéressantes sur la comparaison de cette proposition et des solutions de remplacement. Merci aux présidents du groupe de travail lemonade Eric Burger et Glenn Parsons pour avoir conclu le débat au bon moment et s'être assurés que ce document serait mené à bon terme.

Adresse de l'auteur

Chris Newman
Sun Microsystems
3401 Centrelake Dr., Suite 410
Ontario, CA 91761
US

mél : chris.newman@sun.com

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.