Groupe de travail Réseau
Request for Comments : 3887
Catégorie : En cours de normalisation

T. Hansen, AT&T Laboratories
septembre 2004
Traduction Claude Brière de L'Isle

Protocole d'interrogation de suivi de message

Statut de ce mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Les consommateurs qui achètent des systèmes de messagerie d'entreprise demandent souvent : Puis-je suivre les messages ? Le suivi de message est la capacité de découvrir le chemin qu'a pris un certain message à travers le système de messagerie et l'état actuel de ce message. Le présent document décrit le protocole d'interrogation de suivi de message qui est utilisé en conjonction avec les extensions au protocole ESMTP pour fournir une solution complète de suivi de message pour l'Internet.

Table des Matières

1. Introduction

Le document Modèles et exigences du suivi de message [RFC3888] discute des modèles que pourraient suivre les solutions de suivi de message, ainsi que les exigences d'une solution de suivi de message qui pourraient être utilisées avec l'infrastructure de message à l'échelle de l'Internet. Le présent mémoire et ses compagnons, les [RFC3885] et [RFC3886], décrit une solution complète de suivi de message qui satisfait à ces exigences. La [RFC3885] définit une extension au service SMTP qui fournit les informations nécessaires pour suivre les messages. Le présent mémoire définit un protocole qui peut être utilisé pour interroger l'état des messages qui ont été transmis sur l'Internet via SMTP. La [RFC3886] décrit le type de support d'état de suivi de message [RFC2045] qui est utilisé pour rapporter les informations d'état de suivi. En utilisant la terminologie du document modèle, cette solution utilise l'activation active et des demandes actives avec des références de demande et de chaînage.

1.1 Terminologie

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Toutes les descriptions de syntaxe utilisent l'ABNF spécifié par la [RFC2234]. Les nœuds terminaux non définis ailleurs dans le présent document sont définis dans les [RFC2234], [RFC2396], [RFC2554], [RFC2821], ou [RFC3885].

2. Fonctionnement de base

Le protocole d'interrogation de suivi de message (MTQP, *Message Tracking Query Protocol*) est similaire aux nombreux autres protocoles Internet en mode ligne, tels que la [RFC1939] et la [RFC0977]. Initialement, l'hôte serveur commence le service MTQP en écoutant sur l'accès TCP 1038.

Lorsque un client MTQP souhaite utiliser le service de suivi de message, il établit une connexion TCP avec l'hôte serveur, comme enregistré lors de la soumission du message initial ou comme retourné par une demande de suivi antérieure. Pour trouver l'hôte serveur, le client MTQP fait d'abord une recherche SRV sur l'hôte serveur en utilisant les enregistrements DNS SRV, avec un nom de service de "mtqp" et un nom de protocole de "tcp", comme dans _mtqp._tcp.smtp3.example.com. (Voir les détails au paragraphe "Règles d'usage" dans la [RFC2782].) Si il n'y a pas d'enregistrement SRV, le client MTQP fait alors une recherche d'enregistrement d'adresse pour l'hôte serveur. Lorsque la connexion est établie, le serveur MTQP envoie le mot d'accueil. Le client MTQP et le serveur MTQP échangent respectivement des commandes et des réponses jusqu'à ce que la connexion soit close ou interrompue.

2.1 Considérations sur le service de suivi dans le DNS

À cause de la façon dont sont effectuées les recherches sur l'hôte serveur, de nombreuses configurations de serveur de suivi différentes sont acceptées.

Un système de messagerie qui utilise un seul hôte serveur de messagerie et a l'hôte serveur MTQP sur le même hôte serveur va très vraisemblablement avoir un seul enregistrement MX qui pointe sur le serveur hôte, et sinon, il aura un enregistrement d'adresse. Les clients de messagerie et de MTQP auront tous deux accès directement à cet hôte.

Un système de messagerie qui utilise un seul hôte serveur de messagerie, mais veut que les interrogations de suivi soient effectuées sur une machine différente, DOIT avoir un enregistrement SRV MTQP qui pointe sur cette autre machine.

Un système de messagerie qui utilise des serveurs de messagerie multi rattachements a le choix entre deux solutions pour fournir les services de suivi : soit tous les serveurs de messagerie font fonctionner des serveurs de suivi qui sont capables de restituer les informations sur tous les messages, soit le service de suivi doit être effectué sur une ou plusieurs machines qui sont capables de restituer les informations sur tous les messages. Dans le premier cas, aucun enregistrement DNS supplémentaire n'est nécessaire en plus des enregistrements MX déjà en place pour le système de messagerie. Dans le dernier cas, des enregistrements SRV MTQP sont nécessaires pour pointer sur la ou les machines qui font fonctionner le service de suivi. Dans les deux cas, on notera que le service de suivi DOIT être capable de traiter les interrogations pour tous les messages acceptés par ce système de messagerie.

2.2 Commandes

Les commandes dans MTQP consistent en un mot-clé insensible à la casse, éventuellement suivi par un ou plusieurs paramètres. Toutes les commandes se terminent par une paire CRLF. Les mots clés et les paramètres consistent en caractères ASCII imprimables. Les mots clés et les paramètres sont séparés par des espaces (un ou plusieurs caractères espace ou tabulation). Une ligne de commande est limitée à 998 caractères avant le CRLF.

2.3 Réponses

Dans MTQP, les réponses consistent en un indicateur d'état qui indique la réussite ou l'échec. Les commandes réussies peuvent aussi être suivies par des lignes supplémentaires de données. Toutes les lignes de réponse sont terminées par une paire CRLF et sont limitées à 998 caractères avant le CRLF. Il y a plusieurs indicateurs d'état : "+OK" indique la réussite; "+OK+" indique un succès suivi par des lignes de données supplémentaires, une réponse de succès multi lignes ; "-TEMP" indique un échec temporaire; "-ERR" indique un échec permanent ; et "-BAD" indique une erreur de protocole (comme pour une commande non reconnue).

Un indicateur d'état PEUT être suivi par une série d'informations de réponse analysables par la machine et insensibles à la casse, qui donnent plus de données sur les erreurs. Elles sont séparées de l'indicateur d'état et de chaque autre par un seul caractère barre oblique ("/", code décimal 47). À la suite de cela, il PEUT y avoir des espaces et un message de texte lisible par l'homme. Le message de texte lisible par l'homme n'est pas destiné à être présenté à l'utilisateur final, mais devrait être approprié au classement dans un journal d'événements pour être utilisé à des problèmes de débogage.

Dans une réponse de succès multi lignes, chaque ligne est terminée par une paire CRLF et limitée à 998 caractères avant le CRLF. Lorsque toutes les lignes de la réponse ont été envoyées, une ligne finale est envoyée, consistant en un seul caractère point (".", code décimal 046) et une paire CRLF. Si une ligne de la réponse multi lignes commence par un point, la ligne sera "bourrée de point" en ajoutant un second point devant le point. Lorsque il examine une réponse multi lignes, le client vérifie pour voir si la ligne commence par un point. Si c'est le cas, et si des octets autres que CRLF suivent, le premier octet de la ligne (le point) est éliminé. Si il en est ainsi, et si un CRLF suit immédiatement le point, la réponse du serveur MTQP se termine et la ligne contenant le ".CRLF" n'est pas considérée comme faisant partie de la réponse multi lignes.

Un serveur MTQP DOIT répondre à une commande non reconnue, non mise en œuvre ou syntaxiquement invalide par un indicateur d'état négatif -BAD. Un serveur DOIT répondre à une commande produite lorsque la session est dans un état incorrect avec un indicateur d'état négatif -ERR.

2.4 Considérations sur les pare-feu

Lorsque elle reçoit une interrogation de suivi pour un hôte à l'intérieur de son domaine, une passerelle de messagerie pare-feu a le choix entre retourner une réponse à l'interrogation disant que le message a été bien passé mais qu'aucune autre information n'est disponible, et effectuer elle-même une opération de chaînage, collectant les informations sur le message auprès des hôtes de messagerie derrière le pare-feu, et retournant au client MTQP les informations pour chaque bond derrière le pare-feu, ou éventuellement juste les informations du bond final, en déguisant éventuellement aussi les noms de tout hôte derrière le pare-feu. L'option à retenir est une décision administrative qui ne sera pas plus commentée par le présent document.

Si un serveur choisit d'effectuer lui-même une opération de chaînage, il DOIT fournir une réponse dans les 2 minutes, et DEVRAIT retourner une réponse "aucune autre information disponible" si il ne peut pas fournir une réponse à la fin de cette limite.

2.5 Temporisateurs facultatifs

Un serveur MTQP PEUT avoir un temporisateur d'auto déconnexion sur inactivité. Un tel temporisateur DOIT être d'au moins 10 minutes. La réception de toute commande de la part du client durant cet intervalle devrait suffire pour remettre à zéro le temporisateur d'auto déconnexion. Un serveur MTQP PEUT limiter le nombre de commandes, de commandes non reconnues, ou le temps total de connexion, ou PEUT utiliser d'autres critères, pour empêcher les attaques de déni de service.

Un client MTQP PEUT avoir un temporisateur d'auto déconnexion sur inactivité lorsque il attend une réponse de la part du serveur. Comme un serveur MTQP peut être un pare-feu, et peut être en train de chaîner des informations provenant

d'autres serveurs, un tel temporisateur DOIT être d'au moins 2 minutes.

3. Initialisation et réponse d'option

Une fois que la connexion TCP a été ouverte par un client MTQP, le serveur MTQP produit une réponse initiale d'état qui indique qu'il est prêt. Si la réponse d'état est positive (+OK ou +OK+) le client peut procéder aux autres commandes.

La réponse initiale d'état DOIT inclure les informations de réponse "/MTQP". Les réponses négatives DOIVENT inclure un code de cause comme informations de réponse. Les codes de cause suivants sont définis ici ; des codes de cause non reconnus qui seront ajoutés à l'avenir pourront être traités comme équivalents à "unavailable" *(indisponible)*.

"/" "unavailable"
"/" "admin"

Le code de cause "/admin" DEVRAIT être utilisé quand le service est indisponible pour des raisons administratives. Le code de cause "/unavailable" DEVRAIT être utilisé quand le service est indisponible pour d'autres raisons.

Si le serveur a des options activées, elles sont énumérées comme réponse multi lignes de la réponse d'état initiale, une par ligne. Une spécification d'option consiste en un identifiant, facultativement suivi par des paramètres spécifiques de l'option. Une spécification d'option peut être continuée sur des lignes supplémentaires en commençant les lignes de continuation par une espace *(WSP)*. L'identifiant d'option est insensible à la casse. Les identifiants d'option qui commencent par les caractères "vnd." sont réservés à l'utilisation des fabricants. (Voir ci-dessous.)

Une spécification d'option est définie ici :

STARTTLS [1*WSP "required"]

Cette capacité DOIT être mentionnée si la commande facultative STARTTLS est activée sur le serveur MQTP et si un ou plusieurs certificats ont été correctement installés.

Il y a un paramètre facultatif : le mot "required" *(exigé)*. (Les paramètres pour STARTTLS sont insensibles à la casse). Si le serveur exige que TLS soit utilisé pour certains des domaines que traite le serveur, celui-ci DOIT spécifier le paramètre "required".

3.1 Exemples

Exemple n° 1 (pas d'option) :
S: +OK/MTQP Serveur MTQP prêt

Exemple n° 2 (service temporairement indisponible) :
S: -TEMP/MTQP/admin Service interrompu pour des raisons administratives, rappeler plus tard

Exemple n° 3 (service indisponible de façon permanente) :
S: -ERR/MTQP/unavailable Service interrompu

Exemple n° 4 (autre cas pour pas d'option) :
S: +OK+/MTQP Serveur MTQP prêt
S: .

Exemple n° 5 (options disponibles) :
S: +OK+/MTQP Serveur MTQP prêt
S: starttls
S: vnd.com.example.option2 avec paramètres privés à example.com
S: vnd.com.example.option3 avec une très longue liste de paramètres
S: .

4. Commande TRACK

Syntaxe :

track-command = "TRACK" 1*WSP unique-envid 1*WSP mtrk-secret CRLF
mtrk-secret = base64

Unique-envid est défini dans la [RFC3885]. Mtrk-secret est le secret A décrit dans la [RFC3885], codé en base64.

Lorsque le client produit la commande TRACK, et que l'utilisateur est validé, le serveur MTQP restitue les informations de suivi sur un message électronique. Pour valider l'usager, la valeur de mtrk-secret est hachée en utilisant SHA1, comme décrit dans la [RFC3174]. La valeur du hachage est alors comparée à la valeur passée avec le message lorsque il a été envoyé à l'origine. Si les valeurs du hachage correspondent, l'usager est validé.

Une réponse réussie DOIT être multi lignes, consistant en une partie de corps de la [RFC2045]. La partie de corps MIME DOIT être du type multipart/related, avec des sous parties de message/tracking-status, comme défini dans la [RFC3886]. La réponse contient les informations de suivi sur le message électronique qui a utilisé l'identifiant de suivi mentionné. Une réponse négative à la commande TRACK peut comporter les codes de cause suivants :

"/" "tls-required"
"/" "admin"
"/" "unavailable"
"/" "noinfo"
"/" "insecure"

Le code de cause "/tls-required" DEVRAIT être utilisé quand le serveur a décidé d'exiger TLS. Le code de cause "/admin" DEVRAIT être utilisé lorsque le serveur est devenu indisponible, pour des raisons administratives, depuis que la connexion a été initialisée. Le code de cause "/unavailable" DEVRAIT être utilisé lorsque le serveur est devenu indisponible, pour d'autres raisons, depuis que la connexion a été initialisée. Le code de cause "/insecure" est décrit plus loin.

Si un message n'a pas été vu par le serveur MTQP, le serveur DOIT choisir entre deux solutions : il PEUT retourner une réponse positive avec un champ Action de "opaque" dans les informations de suivi, ou il PEUT retourner une réponse négative avec un code de cause de "noinfo".

4.1 Exemples

Dans chacun des exemples ci-dessous, le unique-envid est "<12345-20010101@example.com>", le secret A est "abcdefgh", et le hachage SHA1 B est (en hexadécimal) "734ba8b31975d0dbae4d6e249f4e8da270796c94". Le message est venu de example.com et le serveur MTQP est example2.com.

Exemple n° 6 : Message délivré :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: delivered
S: Status: 2.5.0
S:
S: --%%%%--
S: .

Exemple n° 7 : Message transféré :

C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: transferred
S: Remote-MTA: dns; example3.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S: Status:2.4.0
S:
S: --%%%%--
S: .

Exemple n° 8 : Message retardé et un en-tête bourré avec un point :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S: ..Dot-Stuffed-Header: à titre d'exemple
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: delayed
S: Status: 4.4.1 (Pas de réponse de l'hôte)
S: Remote-MTA: dns; example3.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S: Will-Retry-Until: Thu, 4 Jan 2001 15:15:15 -0500
S:
S: --%%%%--
S: .

Exemple n° 9 : Deux utilisateurs, un relayé, un en échec :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: relayed
S: Status: 2.1.9
S: Remote-MTA: dns; example3.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500

S:
S: Original-Recipient: rfc822; user2@example1.com
S: Final-Recipient: rfc822; user2@example1.com
S: Action: failed
S: Status 5.2.2 (Boîte aux lettres saturèe)
S: Remote-MTA: dns; example3.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S:
S: --%%%%--
S: .

Exemple n° 10 : Pare-feu :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: relayed
S: Status: 2.1.9
S: Remote-MTA: dns; smtp.example3.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S:

S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; smtp.example3.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user2@example1.com
S: Final-Recipient: rfc822; user4@example3.com
S: Action: delivered
S: Status: 2.5.0
S:
S: --%%%%--
S: .

Exemple n° 11 : Pare-feu, combinaison de blocs par receveur :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: relayed
S: Status: 2.1.9
S: Remote-MTA: dns; smtp.example3.com

S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S:
S: Original-Recipient: rfc822; user2@example1.com
S: Final-Recipient: rfc822; user4@example3.com
S: Action: delivered
S: Status:2.5.0
S:
S: --%%%%--
S: .

Exemple n° 12 : Pare-feu, cachant les noms système derrière le pare-feu :
C: TRACK <12345-20010101@example.com> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S: Content-Type: multipart/related; boundary=%%%%; type=tracking-status
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user1@example1.com
S: Final-Recipient: rfc822; user1@example1.com
S: Action: relayed
S: Status: 2.1.9
S: Remote-MTA: dns; example2.com
S: Last-Attempt-Date: Mon, 1 Jan 2001 19:15:03 -0500
S:
S: --%%%%
S: Content-Type: message/tracking-status
S:
S: Original-Envelope-Id: 12345-20010101@example.com
S: Reporting-MTA: dns; example2.com
S: Arrival-Date: Mon, 1 Jan 2001 15:15:15 -0500
S:
S: Original-Recipient: rfc822; user2@example1.com
S: Final-Recipient: rfc822; user4@example1.com
S: Action: delivered
S: Status: 2.5.0
S:
S: --%%%%--
S: .

5. Commande COMMENT

Syntaxe :
comment-command = "COMMENT" opt-text CRLF
opt-text = [WSP *(VCHAR / WSP)]

Lorsque le client produit la commande COMMENT, le serveur MTQP DOIT répondre par une réponse de réussite (+OK ou +OK+). Tout le texte facultatif fourni avec la commande COMMENT est ignoré.

6. Commande STARTTLS

Syntaxe :
starttls-command = "STARTTLS" 1*WSP domaine *WSP CRLF
domaine = (sous-domaine 1*("." sous-domaine))

TLS [RFC2246] est un mécanisme populaire pour améliorer les communications TCP avec la protection de la

confidentialité et l'authentification. Tous les serveurs MTQP DOIVENT mettre en œuvre TLS. Cependant, TLS PEUT être désactivé par un administrateur de serveur, soit explicitement, soit en omettant d'installer des certificats pour que TLS les utilise. Si un serveur MTQP prend en charge TLS et a un ou plusieurs certificats disponibles, il DOIT inclure "STARTTLS" dans la liste des spécifications d'option au démarrage du protocole.

Note : TLS DEVRAIT être activé sur les serveurs MQTP chaque fois que possible.

Le paramètre DOIT être un nom de domaine pleinement qualifié (FQDN, *fully qualified domain name*). Un client DOIT spécifier le nom d'hôte auquel il pense parler afin que le serveur puisse répondre avec le certificat TLS approprié. Ceci est utile pour les serveurs virtuels qui fournissent le suivi de message pour plusieurs domaines (c'est-à-dire, l'hébergement virtuel).

Si le serveur retourne une réponse négative, il PEUT utiliser un des codes de réponse suivants :

"/" "unsupported"
"/" "unavailable"
"/" "tls-in-progress"
"/" "bad-fqdn"

Si TLS n'est pas pris en charge, un code de réponse de "/unsupported" DEVRAIT alors être utilisé. Si TLS n'est pas disponible pour quelque autre raison, un code de réponse de "/unavailable" DEVRAIT alors être utilisé. Si une session TLS est déjà en cours, c'est alors une erreur de protocole et "-BAD" DOIT être retourné avec un code de réponse de "/tls-in-progress". Si il y a discordance entre le FQDN fourni et le FQDN qui se trouve dans le champ dNSName de l'extension subjectAltName du certificat du serveur [RFC3280], c'est une erreur de protocole et "-BAD" DOIT être retourné avec un code de réponse de "/bad-fqdn".

Après avoir reçu une réponse positive à une commande STARTTLS, le client DOIT commencer la négociation TLS avant de donner aucune autre commande MTQP.

Si le client MTQP utilise le traitement en parallèle (voir ci-dessous) la commande STARTTLS doit être la dernière commande dans un groupe.

6.1 Traitement après la commande STARTTLS

Si la prise de contact TLS échoue, le serveur DEVRAIT interrompre la connexion.

Après l'achèvement de la prise de contact TLS, les deux parties DOIVENT immédiatement décider si elles continuent sur la base de l'authentification et de la confidentialité réalisées. Le client et le serveur MTQP peuvent décider de continuer même si la négociation TLS s'est terminée sans authentification et/ou sans confidentialité parce que la plupart des services MTQP sont effectués sans authentification ni confidentialité, mais certains clients ou serveurs MTQP peuvent vouloir ne continuer que si un certain niveau d'authentification et/ou de confidentialité a été réalisé.

Si le client MTQP décide que le niveau d'authentification ou de confidentialité n'est pas assez élevé pour qu'il continue, il DEVRAIT produire une commande MTQP QUIT immédiatement après l'achèvement de la négociation TLS.

Si le serveur MTQP décide que le niveau d'authentification ou de confidentialité n'est pas assez élevé pour qu'il continue, il PEUT interrompre la connexion. Si il décide que le niveau d'authentification ou de confidentialité n'est pas assez élevé pour qu'il continue, et si il n'interrompt pas la connexion, il DEVRAIT répondre à toute commande MTQP provenant du client (autre qu'une commande QUIT) par une réponse négative "-ERR" et un code de réponse de "/insecure".

6.2 Résultat de la commande STARTTLS

À l'achèvement de la prise de contact TLS, le protocole MTQP est remis dans l'état initial (l'état de MTQP après qu'un serveur démarre). Le serveur DOIT éliminer toute information obtenue du client avant la négociation TLS elle-même. Le client DOIT éliminer toute information obtenue du serveur, comme la liste des options MTQP, qui n'ont pas été obtenues de la négociation TLS elle-même.

À l'achèvement de la prise de contact TLS, le serveur agit comme si la connexion avait été initiée et il répond avec une réponse d'état initial et, facultativement, une liste d'options de serveur. La liste des options de serveur MTQP reçue après la prise de contact TLS DOIT être différente de celle retournée avant la prise de contact TLS. En particulier, un serveur NE DOIT PAS retourner l'option STARTTLS dans la liste des options de serveur après l'achèvement de la prise de contact

TLS.

Le client et le serveur DOIVENT tous deux savoir si il y a une session TLS active. Un client NE DOIT PAS tenter de commencer une session TLS si il y en a déjà une d'active.

7. Commande QUIT

Syntaxe :
quit-command = "QUIT" CRLF

Lorsque le client produit une commande QUIT, la session MTQP se termine. La commande QUIT n'a pas de paramètre. Le serveur DOIT répondre avec une réponse de succès. Le client PEUT clore la session à partir de son extrémité immédiatement après avoir produit cette commande (si le client est sur un système d'exploitation où cela ne pose pas de problème).

8. Traitement en parallèle

Le client MTQP peut choisir de transmettre des groupes de commandes MTQP en lots sans attendre une réponse à chaque commande individuelle. Le serveur MTQP DOIT traiter les commandes dans leur ordre de réception.

Des commandes spécifiques peuvent faire peser d'autres contraintes sur le traitement en parallèle. Par exemple, STARTTLS doit être la dernière commande d'un lot de commandes MTQP.

8.1 Exemples

Les deux exemples suivants sont identiques :

Exemple n° 13 :
C: TRACK <tracking-id> YWJjZGVmZ2gK
S: +OK+ Voici les informations de suivi
S:
S: ... tracking details #1 aller là ...
S: .
C: TRACK <tracking-id-2> QUJDREVGR0gK
S: +OK+ Voici les informations de suivi
S:
S: ... tracking details #2 aller là ...
S: .

Exemple n° 14 :
C: TRACK <tracking-id> YWJjZGVmZ2gK
C: TRACK <tracking-id-2> QUJDREVGR0gK
S: +OK+ Voici les informations de suivi
S:
S: ... tracking details #1 aller là ...
S: .
S: +OK+ Voici les informations de suivi
S:
S: ... tracking details #2 aller là ...
S: .

9. Schéma d'URI MTQP

9.1 Utilisation prévue

Le schéma d'URI MTQP est utilisé pour désigner les serveurs MTQP sur les hôtes Internet accessibles en utilisant le

protocole MTQP. Il effectue une interrogation MTQP et retourne les informations d'état de suivi.

9.2 Nom de schéma d'URI

Le nom du schéma d'URI est "mtqp".

9.3 Syntaxe de schéma d'URI

Un URI MTQP prend une des formes suivantes :

mtqp://<mserver>/track/<unique-envid>/<mtrk-secret>
mtqp://<mserver>:<port>/track/<unique-envid>/<mtrk-secret>

La première forme est utilisée pour se référer à un serveur MTQP sur l'accès standard, tandis que la seconde forme spécifie un accès non standard. Ces deux formes spécifient que la commande TRACK est à produire en utilisant l'identifiant de suivi (unique-envid) et le secret d'autorisation (mtrk-secret) donnés. L'élément de chemin "/track/" DOIT être traité comme insensible à la casse, mais unique-envid et mtrk-secret NE le DOIVENT PAS.

9.3.1 Syntaxe formelle

Voici une description ABNF de l'URI MTQP.

mtqp-uri = "mtqp://" authority "/track/" unique-envid "/" mtrk-secret

9.4 Règles de codage

Le codage de unique-envid est discuté dans la [RFC3885]. Mtrk-secret doit obligatoirement être codé en base64. Si les octets "/", "?" et "%" apparaissent dans unique-envid ou mtrk-secret, il est de plus obligatoire de les représenter par un "%" suivi de deux caractères hexadécimaux. (Les deux caractères donnent la représentation hexadécimale de cet octet).

10. Considérations relatives à l'IANA

Le numéro d'accès système 1038 a été alloué au protocole d'interrogation de suivi de message par l'Autorité d'allocation des numéros de l'Internet (IANA).
Le nom de service "MTQP" a été enregistré auprès de l'IANA.
L'IANA a aussi enregistré le gabarit d'enregistrement d'URI donné à l'Appendice A conformément à la [RFC2717].

Le présent document demande que l'IANA tienne un nouveau registre : MTQP options. L'objet du registre est d'enregistrer les options du présent protocole. Les options dont le nom ne commence pas par "vnd." DOIVENT être définies dans une RFC en cours de normalisation ou expérimentale approuvée par l'IESG. Les nouvelles options MTQP DOIVENT inclure les informations suivantes au titre de leur définition :

identifiant d'option
paramètre d'option
commandes ajoutées
commandes standard affectées
référence de spécification
discussion

Une option MTQP est définie dans le présent document, avec la définition d'enregistrement suivante :

Identifiant d'option : STARTTLS
Paramètres d'option : aucun
Commandes ajoutées : STARTTLS
Commandes standard affectées : aucune
Spécification de référence : RFC 3887
Discussion : voir la RFC 3887

Des options supplémentaires spécifiques d'un fabricant pour le présent protocole ont des noms qui commencent par "vnd.". Après le "vnd." va apparaître le nom de domaine inversé du fabricant, un autre point ".", et un nom pour l'option elle-même. Par exemple, "vnd.com.example.extinfo" pourrait représenter une extension spécifique d'un fabricant donnant des suppléments d'information de la part du propriétaire du domaine "example.com". Ces noms PEUVENT être enregistrés auprès de l'IANA.

11. Considérations sur la sécurité

Si le générateur d'un message devait déléguer sa demande de suivi à un tiers, cela serait vulnérable à l'espionnage sur des sessions non chiffrées. L'utilisateur peut décider message par message si ce risque est acceptable.

La sécurité des informations de suivi dépend du caractère aléatoire du secret choisi pour chaque message et du niveau d'exposition de ce secret. Si des secrets différents sont utilisés pour chaque message, l'exposition maximum provenant du suivi d'un message sera celle d'un seul message pendant le temps où les informations de suivi sont conservées sur un serveur MTQP. Si ce niveau d'exposition est trop élevé, TLS peut être utilisé pour réduire un peu l'exposition.

On devrait noter que le suivi de message n'est pas un mécanisme de bout en bout. Donc, si une paire client/serveur MTQP décide d'utiliser la confidentialité TLS, les interrogations de suivi ne sont pas sécurisée auprès des serveurs MTQP antérieurs ou postérieurs.

Le client et le serveur MTQP doivent tous deux vérifier le résultat de la négociation TLS pour voir si une authentification ou confidentialité acceptable a été réalisée. Ignorer cette étape invaliderait complètement l'utilisation de TLS pour la sécurité. La décision sur l'acceptabilité de l'authentification ou de la confidentialité est prise en local ; elle dépend de la mise en œuvre, et sort du domaine d'application du présent document.

Le client et le serveur MTQP devraient noter avec soin le résultat de la négociation TLS. Si la négociation résulte en pas de confidentialité, ou si elle résulte en une confidentialité qui utilise des algorithmes ou des longueurs de clé qui sont réputés trop faibles, ou si l'authentification n'est pas assez bonne pour l'une ou l'autre partie, le client peut choisir de mettre fin à la session MTQP avec une commande QUIT immédiate, ou le serveur peut choisir de ne plus accepter de commande MTQP.

Une attaque par interposition peut être lancée en supprimant la réponse d'option "STARTTLS" provenant du serveur. Cela ferait que le client n'essayerait pas une session TLS. Un client MTQP peut se protéger contre cette attaque en enregistrant le fait qu'un certain serveur MTQP offre TLS durant une session et en générant une alarme si cette offre n'apparaît pas dans une réponse d'option pour une session ultérieure.

De façon similaire, l'identité du serveur telle qu'exprimée dans le certificat d'un serveur devrait être mise en antémémoire, et une alarme devrait être générée si elle ne correspond pas dans une session ultérieure.

Si TLS n'est pas utilisé, une demande de suivi est vulnérable aux attaques en répétition, de telle sorte qu'un espion peut ultérieurement répéter la même prise de contact pour éventuellement glaner plus d'informations sur l'état d'un message.

Avant qu'ait commencé la prise de contact TLS, 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 toute information obtenue avant le début de la prise de contact TLS lors de l'achèvement de cette prise de contact.

Si une paire client/serveur effectue une prise de contact TLS et si le serveur fait un enchaînement de références, le serveur DEVRAIT alors tenter de négocier TLS au même (ou mieux) niveau de sécurité au prochain bond. Dans un scénario bond par bond, STARTTLS est une demande de sécurité "au mieux" et devrait être traité comme telle.

SASL n'est pas utilisé parce que l'authentification est par message plutôt que par utilisateur.

12. Syntaxe du protocole

Voici une description collectée en ABNF du protocole MTQP.

mtqp-uri = "mtqp://" authority "/track/" unique-envid "/" mtrk-secret

conversation = command-response *(client-command command-response)

; côté client
client-command = track-command / starttls-command / quit-command /comment-command

track-command = "TRACK" 1*WSP unique-envid 1*WSP mtrk-secret CRLF
mtrk-secret = base64

starttls-command = "STARTTLS" 1*WSP domain *WSP CRLF
domain = (sub-domain 1*("." sub-domain))

quit-command = "QUIT" CRLF

comment-command = "COMMENT" opt-text CRLF

; côté serveur
command-response = success-response / temp-response / error-response / bad-response

temp-response = "-TEMP" response-info opt-text CRLF

opt-text = [WSP *(VCHAR / WSP)]

error-response = "-ERR" response-info opt-text CRLF

bad-response = "-BAD" response-info opt-text CRLF

success-response = single-line-success / multi-line-success

single-line-success = "+OK" response-info opt-text CRLF

multi-line-success = "+OK+" response-info opt-text CRLF
*dataline dotcrlf

dataline = *998OCTET CRLF

dotcrlf = "." CRLF

NAMECHAR = ALPHA / DIGIT / "-" / "_"

response-info = *("/" ("admin" / "unavailable" / "unsupported" / "tls-in-progress" / "insecure" / "tls-required" / 1*NAMECHAR))

13. Remerciements

La description de STARTTLS se fonde sur la [RFC3207].

14. Références

[RFC0977] B. Kantor et P. Lapsley, "Protocole de transfert des nouvelles du réseau", février 1986. *(Obsolète, voir* RFC3977*)*

[RFC1939] J. Myers, M. Rose, "Protocole Post Office - version 3", mai 1996. (*MàJ par* RFC1957, RFC2449) (STD0053)

[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.)*

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

[RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir* RFC5234)

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999.

[RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir* RFC3986)

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

[RFC2717] R. Petke, I. King, "Procédures d'enregistrement des noms de schéma d'URL", novembre 1999. (*Obsolète, voir* RFC4395) (BCP0035)*)*

[RFC2782] A. Gulbrandsen, P. Vixie et L. Esibov, "Enregistrement de ressource DNS pour la spécification de la localisation des services (DNS SRV)", février 2000.

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

[RFC3174] D. Eastlake 3 et P. Jones, "Algorithme US de hachage sécurisé n° 1 (SHA1)", sept. 2001. *(Info, MàJ par* 4634 *et* 6234*)*

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

[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)

[RFC3885] E. Allman, T. Hansen, "Extension de service SMTP pour le suivi de message", septembre 2004. *(P.S.)*

[RFC3886] E. Allman, "Format de message extensible pour les réponses de suivi de message", septembre 2004. *(P.S.)*

[RFC3888] T. Hansen, "Modèle et exigences du suivi de message", septembre 2004. *(Information)*

Appendice A. Gabarit d'enregistrement d'URI MTQP

Nom du schéma : mtqp
Syntaxe du schéma : voir le paragraphe 9.1
Considérations de codage des caractères : voir le paragraphe 9.4
Utilisation prévue : voir le paragraphe 9.3
Applications et/ou protocoles qui utilisent ce schéma : MTQP
Considérations d'interopérabilité : comme spécifié pour MTQP
Considérations de sécurité : voir le paragraphe 11.0
Publications pertinentes : [RFC3885], [RFC3888], [RFC3886]
Contact : Groupe de travail MSGTRK
Auteur/Contrôleur des modifications : IESG

Adresse de l'auteur

Tony Hansen
AT&T Laboratories
Middletown, NJ 07748
USA

téléphone : +1.732.420.8934
mél : tony+msgtrk@maillennium.att.com

Déclaration complète de droits de reproduction

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