Groupe de travail Réseau
Request for Comments : 4752
RFC rendue obsolète : 2222
Catégorie : Sur la voie de la normalisation

A. Melnikov, éditeur, Isode
novembre 2006

Traduction Claude Brière de L'Isle

Mécanisme Kerberos V5 ("GSSAPI") simple authentification et couche de sécurité (SASL)

Résumé
L'authentification simple et couche de sécurité (SASL, *Simple Authentication et Security Layer*) est un cadre pour ajouter la prise en charge de l'authentification aux protocoles en mode connexion. Le présent document décrit la méthode pour utiliser l'interface de programme d'application de service de sécurité générique (GSS-API, *Generic Security Service Application Program Interface*) Kerberos V5 dans SASL.

Le présent document remplace le paragraphe 7.2 de la RFC 2222, la définition du mécanisme SASL "GSSAPI". Le présent document, avec la RFC 4422, rend obsolète la RFC 2222.

Table des matières

1. Introduction

La présente spécification documente le mécanisme actuellement déployé d'authentification simple et de couche de sécurité (SASL, *Simple Authentication et Security Layer)* [RFC4422] qui prend en charge le mécanisme Kerberos V5 [RFC4120] d'interface de programme d'application de service de sécurité générique (GSS-API, *Generic Security Service Application Program Interface)* [RFC2743], [RFC4121]. La séquence d'authentification est décrite à la Section 3. Noter que la séquence d'authentification décrite a des limitations connues, en particulier, elle manque de liens de canal et le nombre d'allers-retours requis pour réaliser l'échange d'authentification n'est pas minimal. Le groupe de travail SASL travaille sur un document distinct qui devrait traiter ces limitations.

1.1 Relation aux autres documents

Le présent document, avec la RFC 4422, rend obsolète la RFC 2222 dans sa totalité. Le présent document remplace le paragraphe 7.2 de la RFC 2222. Le reste est rendu obsolète comme précisé au paragraphe 1.2 de la RFC 4422.

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].

3. Mécanisme GSS-API Kerberos V5

Le nom du mécanisme SASL pour le mécanisme Kerberos V5 GSS-API [RFC4121] est "GSSAPI". Bien que connu sous le nom de mécanisme SASL GSSAPI, le mécanisme est spécifiquement lié à Kerberos V5 et au mécanisme Kerberos V5 de GSS-API.

Le mécanisme GSSAPI SASL est un mécanisme SASL où "le client passe en premier" ; c'est-à-dire, il commence par l'envoi d'une "réponse" par le client créée comme décrit dans le paragraphe suivant.

La mise en œuvre PEUT établir tous fanions ou arguments GSS-API non mentionnés dans la présente spécification comme nécessaire pour que la mise en œuvre applique sa politique de sécurité.

Noter que les codes d'état majeurs retournés par GSS_Init_sec_context() ou GSS_Accept_sec_context() autres que GSS_S_COMPLETE ou GSS_S_CONTINUE_NEEDED causent l'échec de l'authentification. Les codes d'état majeurs retournés par GSS_Unwrap() autres que GSS_S_COMPLETE (sans aucun code d'état supplémentaire) causent l'échec de l'authentification et/ou de la couche de sécurité.

3.1 Côté client de l'échange de protocole d'authentification

Le client invoque GSS_Init_sec_context *(initiation de contexte de sécurité GSS)* passant une input_context_handle *(bride de contexte d'entrée)* de 0 (initialement), mech_type *(type de mécanisme)* du mécanisme Kerberos V5 GSS-API [RFC1964], un chan_binding *(lien de canal)* de NULL, et un targ_name *(nom de cible)* égal au output_name *(nom de résultat)* provenant du GSS_Import_Name invoqué avec le input_name_type *(type de nom d'entrée)* de GSS_C_NT_HOSTBASED_SERVICE (*) et une input_name_string *(chaîne de nom d'entrée)* de "service@hostname" où "service" est le nom de service spécifié dans le profil du protocole, et "hostname" est le nom d'hôte pleinement qualifié du serveur. Quand il invoque le GSS_Init_sec_context, le client DOIT passer le fanion integ_req_flag de VRAI (**). Si le client veut demander une couche de sécurité, il DOIT aussi fournir au GSS_Init_sec_context un fanion mutual_req_flag de VRAI, et un fanion sequence_req_flag de VRAI. Si le client veut demander une couche de sécurité assurant la protection de la confidentialité, il DOIT aussi fournir au GSS_Init_sec_context un fanion conf_req_flag de VRAI. Le client répond alors avec le jeton output_token résultant. Si GSS_Init_sec_context retourne GSS_S_CONTINUE_NEEDED, le client devrait alors s'attendre à ce que le serveur produise un jeton dans un défi ultérieur. Le client doit passer le jeton à une autre invocation de GSS_Init_sec_context, répétant les actions de ce paragraphe.

(*) Les clients PEUVENT utiliser des types de noms autres que GSS_C_NT_HOSTBASED_SERVICE pour importer les noms d'accepteur des serveurs, mais seulement quand ils ont connaissance a priori que les serveurs prennent en charge des types de noms de remplacement. Autrement, les clients DOIVENT utiliser GSS_C_NT_HOSTBASED_SERVICE pour importer les noms d'accepteur.

(**) Noter que les mises en œuvre de la [RFC2222] ne vont pas fonctionner avec les mises en œuvre de GSS-API qui exigent que integ_req_flag soit vrai. Aucune mise en œuvre de la [RFC1964] ou de la [RFC4121] qui exige que le fanion integ_req_flag soit vrai n'existe et il est estimé que toute future mise à jour de la [RFC4121] exigera que la protection de l'intégrité soit disponible même si elle n'est pas explicitement demandée par l'application.

Quand GSS_Init_sec_context retourne GSS_S_COMPLETE, le client examine le contexte pour s'assurer qu'il fournit un niveau de protection permis par la politique de sécurité du client. En particulier, si le fanion integ_avail n'est pas établi dans le contexte, alors aucune couche de sécurité ne peut être offerte ou acceptée.

Si le fanion conf_avail n'est pas établi dans le contexte, aucune couche de sécurité avec confidentialité ne peut être offerte ou acceptée. Si le contexte est acceptable, le client effectue les tâches suivantes : si la dernière invocation de GSS_Init_sec_context a retourné un jeton output_token, alors le client répond avec le output_token, autrement, le client répond sans données. Le client devrait alors s'attendre à ce que le serveur produise un jeton dans un défi suivant. Le client passe ce jeton à GSS_Unwrap et interprète le premier octet du texte en clair résultant comme un gabarit binaire qui spécifie les couches de sécurité prises en charge par le serveur et les second au quatrième octets comme la taille maximum de message de sortie que le serveur est capable de recevoir (dans l'ordre des octets du réseau). Si le texte en clair résultant n'est pas long de 4 octets, le client fait échouer la négociation. Le client vérifie que la mémoire tampon maximum du serveur est 0 si le serveur n'annonce pas la prise en charge d'une couche de sécurité.

Le client construit alors les données, avec le premier octet contenant le gabarit binaire qui spécifie la couche de sécurité choisie, les second au quatrième octets contenant dans l'ordre des octets du réseau la taille maximum du message de sortie que le client est capable de recevoir (qui DOIT être 0 si le client ne prend en charge aucune couche de sécurité) et les octets restants contenant l'identité d'autorisation codée en UTF-8 [RFC3629]. (Note de mise en œuvre : l'identité d'autorisation n'est pas terminée par un octet de valeur zéro (%x00) (par exemple, le codage UTF-8 du caractère NUL (U+0000))). Le client passe les données à GSS_Wrap avec le fanion conf_req_flag réglé à FAUX et répond avec le message de sortie généré. Le client peut alors considérer que le serveur est authentifié.

3.2 Côté serveur de l'échange de protocole d'authentification

Un serveur NE DOIT PAS annoncer la prise en charge du mécanisme "GSSAPI" SASL décrit dans ce document si il n'a pas d'accréditif d'accepteur pour le mécanisme Kerberos V5 GSS-API [RFC1964].

Le serveur passe la réponse initiale du client à GSS_Accept_sec_context comme jeton d'entrée *(input_token)* réglant input_context_handle à 0 (initialement), un chan_binding de NULL, et un acceptor_cred_handle convenable (voir ci-dessous). Si GSS_Accept_sec_context retourne GSS_S_CONTINUE_NEEDED, le serveur retourne le jeton de sortie *(output_token)* généré au client dans un défi et passe la réponse résultante à une autre invocation de GSS_Accept_sec_context, répétant les actions de ce paragraphe.

Les serveurs DEVRAIENT utiliser un accréditif obtenu en invoquant GSS_Acquire_cred ou GSS_Add_cred pour le nom désiré de GSS_C_NO_NAME et l'identifiant d'objet (OID, *Object Identifier*) du mécanisme Kerberos V5 GSS-API [RFC1964] (*). Les serveurs PEUVENT utiliser GSS_C_NO_CREDENTIAL comme bride d'accréditif d'accepteur. Les serveurs PEUVENT utiliser un accréditif obtenu en invoquant GSS_Acquire_cred ou GSS_Add_cred pour le ou les noms principaux du serveur (**) et le mécanisme Kerberos V5 GSS-API [RFC1964].

(*) À la différence de GSS_Add_cred, GSS_Acquire_cred utilise un ensemble d'OID du mécanisme de GSS-API comme paramètre d'entrée. L'ensemble d'OID peut être créé en utilisant GSS_Create_empty_OID_set et GSS_Add_OID_set_member. Il peut être libéré en invoquant le GSS_Release_oid_set.

(**) L'utilisation des noms principaux d'un serveur qui a le type de nom GSS_C_NT_HOSTBASED_SERVICE et du format "service@hostname", où "service" est le nom de service spécifié dans le profil du protocole, et "hostname" est le nom d'hôte pleinement qualifié du serveur, est RECOMMANDÉE. Le nom du serveur est généré en invoquant GSS_Import_name avec le type de nom d'entrée de GSS_C_NT_HOSTBASED_SERVICE et la chaîne de nom d'entrée de "service@hostname".

À l'établissement réussi du contexte de sécurité (c'est-à-dire, GSS_Accept_sec_context retourne GSS_S_COMPLETE) le serveur DEVRAIT vérifier que le mécanisme GSS-API négocié est bien Kerberos V5 [RFC1964]. Ceci est fait en examinant la valeur du paramètre mech_type retourné de l'invocation de GSS_Accept_sec_context. Si la valeur diffère, l'authentification SASL DOIT être interrompue.

À l'établissement réussi du contexte de sécurité et si le serveur a utilisé GSS_C_NO_NAME/GSS_C_NO_CREDENTIAL pour créer la bride d'accréditif d'accepteur, le serveur DEVRAIT aussi vérifier en utilisant le GSS_Inquire_context que le nom cible *(target_name)* utilisé par le client correspond à

- la syntaxe de nom "service@hostname" de GSS_C_NT_HOSTBASED_SERVICE, où "service" est le nom de service spécifié dans le profil de protocole d'application, ou à
- la syntaxe de nom GSS_KRB5_NT_PRINCIPAL_NAME [RFC1964] pour un principal à deux composants où le premier composant correspond au nom de service spécifié dans le profil de protocole d'application.

Quand GSS_Accept_sec_context retourne GSS_S_COMPLETE, le serveur examine le contexte pour s'assurer qu'il fournit

un niveau de protection permis par la politique de sécurité du serveur. En particulier, si le fanion integ_avail n'est pas établi dans le contexte, alors aucune couche de sécurité ne peut être offerte ou acceptée. Si le fanion conf_avail n'est pas établi dans le contexte, aucune couche de sécurité avec protection de la confidentialité ne peut être offerte ni acceptée.

Si le contexte est acceptable, le serveur effectue les actions suivantes : si la dernière invocation de GSS_Accept_sec_context a retourné un jeton de résultat *(output_token)* le serveur le retourne au client dans un défi et attend du client une réponse sans données. Qu'un jeton de résultat soit ou non retourné (et après reception de toute réponse du client à un tel jeton de résultat) le serveur construit quatre octets de données, avec le premier octet contenant un gabarit binaire qui spécifie les couches de sécurité prises en charge par le serveur et les trois octets suivants contenant dans l'ordre des octets du réseau la taille maximum du jeton de résultat que le serveur est capable de recevoir (qui DOIT être 0 si le serveur ne prend pas en charge de couche de sécurité). Le serveur doit alors passer le texte en clair à GSS_Wrap avec le fanion conf_req_flag réglé à FAUX et produire au client dans un défi le message de résultat généré.

Le serveur doit alors passer la réponse résultante à GSS_Unwrap et interpréter le premier octet du texte en clair résultant comme le gabarit binaire pour la couche de sécurité choisie, les trois octets suivants comme la taille maximum du message de résultat que le client est capable de recevoir (dans l'ordre des octets du réseau) et les octets restants comme l'identité d'autorisation. Le serveur vérifie que le client a choisi une couche de sécurité qui était offerte et que la taille maximum de mémoire tampon du client est 0 si aucune couche de sécurité n'a été choisie. Le serveur doit vérifier que le nom de source *(src_name)* est autorisé à agir comme l'identitié d'autorisation. Après ces vérifications, le processus d'authentification est complet. Le serveur n'est pas supposé retourner de données supplémentaires avec l'indication de succès.

3.3 Couche de sécurité

Les couches de sécurité et leurs gabarits binaires correspondants sont comme suit :
1 Pas de couche de sécurité
2 Protection de l'intégrité. L'envoyeur invoque GSS_Wrap avec le fanion conf_req_flag réglé à FAUX
4 Protection de la confidentialité. L'envoyeur invoque GSS_Wrap avec le fanion conf_req_flag réglé à VRAI.

D'autres gabarits binaires pourront être définis à l'avenir ; les bits qui ne sont pas compris doivent être retirés de la négociation.

Lors du décodage de toutes données reçues avec GSS_Unwrap, un major_status autre que GSS_S_COMPLETE DOIT être traité comme erreur fatale.

Noter que SASL négocie la taille maximum du message de résultat à envoyer. Les mises en œuvre peuvent utiliser l'invocation GSS_Wrap_size_limit pour déterminer la taille maximum correspondante du message d'entrée.

4. Considérations relatives à l'IANA

L'IANA a modifié l'enregistrement existant pour "GSSAPI" comme suit :
Famille de mécanismes SASL : NO
Nom de mécanisme SASL : GSSAPI
Considérations de sécurité : voir la Section 5 de la RFC 4752
Spécification publiée : RFC 4752
Adresse de la personne à contacter pour plus d'informations : Alexey Melnikov <Alexey.Melnikov@isode.com>
Usage prévu : COMMUN
Contrôleur des changements : iesg@ietf.org
Informations supplémentaires : ce mécanisme est pour le mécanisme GSS-API de Kerberos V5.

5. Considérations sur la sécurité

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

Quand il construit la chaîne de nom d'entrée *(input_name_string)* le client NE DEVRAIT PAS canoniser le nom de domaine pleinement qualifié du serveur en utilisant un service de répertoire non sûr ou pas de confiance.

Pour la compatibilité avec le logiciel déployé, le présent document exige que le paramètre chan_binding *(liens de canaux)*

de GSS_Init_sec_context et GSS_Accept_sec_context soit NUL, donc interdisant l'utilisation de la prise en charge de GSS-API pour les liens de canaux. Les liens de canaux GSS-API dans SASL sont supposés être pris en charge via une nouvelle famille GSS-API de mécanismes SASL (qui seront introduits dans un futur document).

Des considérations sur la sécurité supplémentaires sont dans la [RFC4422] et la [RFC2743]. Des considérations sur la sécurité supplémentaires pour le mécanisme GSS-API se trouvent dans les [RFC1964] et [RFC4120].

6. Remerciements

Le présent document remplace le paragraphe 7.2 de la [RFC2222] écrite par John G. Myers. Il a aussi contribué de façon significative à la présente révision.

Lawrence Greenfield a converti le texte de ce document au format XML.

Les contributions de nombreux membres de la liste de diffusion SASL ont été très utiles, en particulier les commentaires de Chris Newman, Nicolas Williams, Jeffrey Hutzelman, Sam Hartman, Mark Crispin, et Martin Rex.

7. Changements par rapport à la RFC 2222

La [RFC2078] spécifie la version de GSS-API utilisée par la [RFC2222], qui fournissait la version originale de cette spécification. Cette version de GSS-API ne fournissait pas le fanion integ_integ_avail comme entrée à GSS_Init_sec_context. Au lieu de cela, la protection de l'intégrité était toujours demandée. La [RFC4422] exige que lorsque possible, la négociation de couche de sécurité soit protégée en intégrité. Pour satisfaire cette exigence et au titre du passage de la [RFC2078] à la [RFC2743], la présente spécification exige que les clients demandent la protection de l'intégrité à GSS_Init_sec_context afin qu'ils puissent utiliser GSS_Wrap pour protéger la négociation de la couche de sécurité. La présente spécification n'exige pas que le mécanisme offre la couche de sécurité d'intégrité, mais simplement que la négociation de couche de sécurité soit enveloppée.

8. Références

8.1 Références normatives

[RFC1964] J. Linn, "Mécanisme GSS-API de Kerberos version 5", juin 1996. (*MàJ par* RFC4121 et RFC6649)

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

[RFC2743] J. Linn, "Interface générique de programme d'application de service de sécurité, version 2, mise à jour 1", janvier 2000. (*MàJ par* RFC5554)

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

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

[RFC4121] L. Zhu et autres, "Version 2 du mécanisme d'interface de programme d'application de service de sécurité générique (GSS-API) de Kerberos version 5", juillet 2005. (*MàJ* RFC1964) (*MàJ par les* RFC6542, 6649, 8062)) *(P.S.)*

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

8.2 Références pour information

[RFC2078] J. Linn, "Interface générique de programme d'application de service de sécurité, version 2", janvier 1997.

((Obsolète, voir la RFC2743)

[RFC2222] J. Myers, "Authentification simple et couche de sécurité (SASL)", octobre 1997. (*Obsolète, voir* RFC4422, RFC4752) (*MàJ par* RFC2444) *(P.S.)*

Adresse de l'éditeur

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

mél : Alexey.Melnikov@isode.com
URI : http://www.melnikov.ca/

Déclaration complète de droits de reproduction

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