Groupe de travail Réseau
Request for Comments : 3767
Catégorie : En cours de normalisation

S. Farrell, éditeur, Trinity College Dublin
juin 2004
Traduction Claude Brière de L'Isle

Protocole de disponibilité sécurisée des accréditifs (SACRED)

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

Résumé
Le présent document décrit un protocole par lequel un utilisateur peut acquérir des accréditifs cryptographiques (par exemple, des clés privées, des structures PKCS n° 15) d'un serveur d'accréditifs, en utilisant une station de travail qui a un logiciel de confiance installé en local, mais sans configuration spécifique de l'utilisateur. La charge utile du protocole est décrite en XML. Le présent mémoire spécifie aussi un profil du protocole extensible d'échange de blocs (BEEP, *Blocks Extensible Exchange Protocol*). Les exigences de sécurité sont satisfaites en rendant obligatoire la prise en charge de TLS et/ou de résumés MD5 (à travers BEEP).

Table des Matières

1. Introduction

Des accréditifs numériques, comme des clés privées et les certificats correspondants, sont utilisés pour prendre en charge divers protocoles Internet, par exemple S/MIME, IPsec, et TLS. Dans un certain nombre d'environnements, les utilisateurs finaux souhaitent utiliser les mêmes accréditifs sur différents appareils d'extrémité. Dans un environnement "normal" d'ordinateur portable, l'utilisateur a déjà de nombreux outils à sa disposition pour permettre l'importation/exportation de ces accréditifs. Cependant, ceci n'est pas très pratique. De plus, avec certains appareils, en particulier les appareils sans fil et autres appareils avec plus de contraintes, les outils requis n'existent tout simplement pas.

Le présent document décrit un protocole pour l'échange sûr de tels accréditifs et est une réalisation du cadre abstrait de protocole décrit dans la [RFC3760].

De nombreux mots de passe choisis par l'utilisateur sont vulnérables aux attaques de dictionnaire. Le protocole SACRED est ainsi conçu pour ne pas donner d'information qu'un attaquant pourrait acquérir pour lancer une attaque de dictionnaire, que ce soit par espionnage ou en se faisant passer pour le client ou le serveur.

Le protocole permet aussi à un utilisateur de créer ou supprimer un compte, de changer le mot de passe et/ou les accréditifs d'un compte, et de charger les nouvelles valeurs sur le serveur. Le protocole assure que seulement quelqu'un qui connaissait le mot de passe de l'ancien compte est capable de modifier les accréditifs mémorisés sur le serveur d'accréditifs. Le protocole n'empêche pas de configurer un serveur à interdire certaines opérations (par exemple, le téléchargement d'accréditifs) pour certains utilisateurs. Les opérations de gestion de compte comme un tout sont de mise en œuvres facultative pour les serveurs d'accréditifs et les clients.

Noter qu'il y a potentiellement deux "mots de passe" impliqués lorsque on utilise ce protocole - le premier est utilisé pour authentifier l'usager au serveur d'accréditifs, et le second pour déchiffrer les accréditifs (des parties des accréditifs) à la suite d'une opération de téléchargement. Lorsque le contexte l'exige, on se réfère au premier comme au mot de passe de compte, et au second comme mot de passe d'accréditif.

Utiliser un protocole comme celui-ci est un peu moins sûr que d'utiliser une carte à mémoire, mais peut être utilisé jusqu'à ce que les cartes à mémoire et les lecteurs de carte à mémoire soient sur tous les postes de travail, et peut être utile même après que les cartes à mémoire seront partout, comme stratégie de sauvegarde lorsque la carte à mémoire d'un utilisateur est perdue ou fonctionne mal.

Le protocole est conçu pour satisfaire aux exigences de la [RFC3157]. les accréditifs cryptographiques peuvent prendre la forme de clés privées, PKCS n° 15 [PKCS15], ou de structures. À ce titre, un profil fondé sur BEEP [RFC3080] est spécifié pour le transport et la sécurité des messages (intégrité, authentification, et confidentialité). Dans ce cas, les exigences de sécurité sont satisfaites en rendant obligatoire la prise en charge (via BEEP) pour TLS [RFC2246] et/ou DIGEST-MD5 [RFC2831].

On suppose que les seules informations d'authentification disponibles à l'utilisateur sont un nom d'utilisateur et un mot de passe.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT" et "FACULTATIF" dans ce document sont à interpréter comme décrit dans la [RFC2119].

2. Le protocole

La présente section définit les opérations de gestion de compte et de "démarrage" pour le protocole SACRED.

Elle décrit aussi les formats de message utilisés, qui sont décrits en XML [XMLSCHEMA]. L'appendice A fournit un schéma XML pour ces éléments.

L'approche retenue ici est de définir les éléments SACRED qui sont compatibles avec ceux utilisés dans [XKMS] et la [RFC3275], afin qu'une mise en œuvre de ce protocole puisse aussi facilement prendre en charge XKMS, et vice versa.

Il est aussi prévu que d'autres instances de protocole SACRED (par exemple utilisant un schéma d'authentification, un format d'accréditif, ou protocole de transport, différents) pourraient réutiliser beaucoup des définitions données ici.

2.1 Opérations de gestion de compte

Ces opérations PEUVENT être mises en œuvre, c'est-à-dire, elles sont FACULTATIVES.

2.1.1 Demande d'informations

Cette opération N'EXIGE PAS l'authentification.

L'objet de cette opération est de fournir au client les valeurs requises pour une création de compte.

Le client envoie un message InfoRequest (qui n'a pas de contenu).

Le serveur répond par un message InfoResponse qui contient les paramètres du mécanisme d'authentification pour le serveur et la liste des types de ProcessInfo pris en charge. Pour DIGEST-MD5, cela consiste en la liste des domaines

(chacun comme un élément XML nommé "Realm") que le serveur prend en charge. Il DOIT y avoir au moins un domaine spécifié.

Les clients DOIVENT être capables de choisir un des domaines de la liste et DOIVENT être capables de ne pas tenir compte des autres informations présentes (pour l'extensibilité).

2.1.2 Création de compte

Cette opération EXIGE l'authentification du serveur.

L'objet de cette opération est d'établir un nouveau compte sur le serveur. Les informations requises pour un "nouveau" compte vont dépendre du mécanisme SASL [RFC2222] utilisé.

Le client envoie une demande CreateAccountRequest *(demande de création de compte)*, qui contient le nom du compte (par exemple le nom d'utilisateur). Il contient aussi les éléments requis pour créer un compte pour un mécanisme d'authentification particulier. Les informations réelles sont définies conformément au mécanisme d'authentification. Pour DIGEST-MD5, cela consiste en le vérificateur de mot de passe (le nom d'utilisateur haché, le mot de passe et le domaine) et le domaine choisi. Bien que plus d'un ensemble de telles données soit permis par les structures de données définies dans l'appendice, les clients DEVRAIENT ici en inclure seulement un.

Le serveur répond par un message d'erreur ou d'accusé de réception.

2.1.3 Suppression de compte

Cette opération EXIGE l'authentification mutuelle.

L'objet de cette opération est de supprimer le compte entier.
Le client envoie un message RemoveAccountRequest (qui n'a pas de contenu) au serveur.

Le serveur DOIT supprimer toutes les informations relatives au compte et répondre par un message d'erreur ou d'accusé de réception.

2.1.4 Modification de compte

Cette opération EXIGE l'authentification mutuelle.

L'objet de cette opération est de permettre au client de changer les informations requises pour l'authentification. Les informations requises vont dépendre de la méthode d'authentification utilisée.

Le client envoie un message ModifyAccountRequest *(demande de modification de compte)*, qui contient les éléments requis pour changer les informations d'authentification pour le compte, pour un mécanisme d'authentification particulier. Les informations réelles sont définies conformément au mécanisme d'authentification. Pour la [RFC2831], elles vont consister en un domaine et une valeur de vérificateur de mot de passe.

Une fois que les informations du compte ont été changées, le serveur va répondre par un message d'erreur ou d'accusé de réception.

2.2 Opérations au démarrage

Ces opérations DOIVENT être prises en charge par toutes les mises en œuvre conformes.

2.2.1 Chargement d'accréditifs

Cette opération EXIGE l'authentification mutuelle.

L'objet de cette opération est de permettre au client de mettre en dépôt un accréditif chez le serveur.

Le client envoie au serveur un message UploadRequest *(demande de chargement)* qui DOIT contenir un accréditif.

Si il existe déjà pour le compte un accréditif avec le même champ de sélecteur d'accréditif que dans la demande UploadRequest (un accréditif "correspondant") alors cet accréditif est remplacé par le nouvel accréditif provenant de

UploadRequest. Autrement un "nouvel" accréditif est associé à ce compte. Si un nouvel accréditif est chargé, le client DEVRAIT alors inclure (dans LastModified) son concept local de l'heure (si il en a un) ou l'indication qu'il n'a pas d'horloge. La valeur réelle de LastModified peut être n'importe quoi, (mais l'élément doit être présent) car cela sera écrasé par le serveur dans tous les cas.

Si un changement est fait aux accréditifs mémorisés associés au compte, le serveur DOIT alors mettre à jour la valeur correspondante de LastModified (retournée dans le message DownloadResponse) à l'heure en cours (chez le serveur).

La valeur de LastModified dans UploadRequest DOIT être celle qui a été reçue le plus récemment dans une DownloadResponse correspondante pour cet accréditif. Cela signifie qu'il est fortement RECOMMANDÉ aux clients de ne produire une demande UploadRequest que sur la base des accréditifs récemment téléchargés, car autrement la valeur de LastModified peut être périmée.

La valeur de LastModified peut aussi être utile pour détecter les conflits. Par exemple, charger sur la plateforme A, télécharger de la plateforme B, mettre à jour à partir de B, mettre à jour à partir de A. Le serveur pourrait détecter un conflit sur le second chargement. Dans ce cas, le serveur DOIT répondre par une erreur BEEP (qui DEVRAIT être StaleCredential *(accréditif périmé)*).

Le serveur remplace la valeur de LastModified fournie par l'heure courante au serveur avant de mémoriser l'accréditif. (Noter que cela signifie qu'il ne serait pas raisonnable pour un client d'inclure le champ LastModified dans une signature numérique ClientInfo qui est calculée sur le CredentialType.)

Le serveur répond par un message d'erreur ou d'accusé de réception.

2.2.2 Téléchargement d'accréditifs

Cette opération EXIGE l'authentification mutuelle.

L'objet de cette opération est de permettre à un client d'obtenir un ou plusieurs accréditifs d'un serveur (ce qui est en réalité l'objet de tout le protocole !).

Le client envoie un message DownloadRequest au serveur qui PEUT contenir une chaîne de sélecteur d'accréditif pour l'accréditif. Pas de sélecteur d'accréditif, ou un sélecteur d'accréditif vide signifie que la demande est pour tous les accréditifs associés à ce compte.

Le serveur répond par un message DownloadResponse ou d'erreur. Une DownloadResponse contient une ou plusieurs charges utiles d'accréditif, incluant l'heure de LastModified qui représente l'heure (au serveur) du dernier changement de chaque accréditif associé au compte (par exemple à la suite d'une demande UploadRequest).

2.2.3 Suppression d'accréditifs

Cette opération EXIGE l'authentification mutuelle.

L'objet de cette opération est de permettre au client de supprimer un ou tous les accréditifs associés au compte.

Le client envoie un message DeleteRequest au serveur qui peut contenir un élément CredentialSelector *(sélecteur d'accréditif)* ou All *(tout)*.

Si la demande de suppression DeleteRequest contient un élément All, alors tous les accréditifs associés à ce compte sont supprimés.

Si la DeleteRequest contient un CredentialSelector, la demande PEUT alors inclure une valeur LastModified. Si la valeur LastModified est présente dans la DeleteRequest, elle DOIT alors être la valeur qui a été le plus récemment reçue dans une DownloadResponse correspondante pour cet accréditif. Si la valeur ne correspond pas, le serveur NE DOIT alors PAS supprimer les accréditifs.

Si il n'existe pas d'accréditif "correspondant", le serveur retourne une erreur.

Le serveur répond à cette demande par un message d'erreur ou d'accusé de réception.

2.3 Divers

2.3.1 Sécurité de session

Six opérations SACRED sont définies ci-dessus. Dans ce paragraphe, on spécifie les exigences de sécurité pour chacune des opérations (lorsque elles sont prises en charge).

Opération	**Sécurité EXIGÉE**
Demande d'information	Aucune
Création de compte	Authentification du serveur, confidentialité, intégrité
Suppression de compte	Authentification mutuelle, confidentialité, intégrité
Modification de compte	Authentification mutuelle, confidentialité, intégrité
Chargement d'accréditif	Authentification mutuelle, confidentialité, intégrité
Téléchargement d'accréditif	Authentification mutuelle, confidentialité, intégrité
Suppression d'accréditif	Authentification mutuelle, confidentialité, intégrité

Les exigences de sécurité peuvent être satisfaites par plusieurs mécanismes. Le présent document EXIGE que les serveurs d'accréditifs prennent en charge TLS et DIGEST-MD5. Les clients DOIVENT accepter DIGEST-MD5 et TLS avec l'authentification de serveur.

La suite de chiffrement de mise en œuvre obligatoire TLS pour SACRED est TLS_RSA_WITH_3DES-EDE_CBC_SHA. Les mises en œuvre DEVRAIENT aussi prendre en charge TLS_RSA_WITH_AES_128_CBC_SHA [RFC3268].

Lorsque on effectue l'authentification mutuelle en utilisant DIGEST-MD5 pour le client, DIGEST-MD5 DOIT seulement être utilisé "au sein" d'un "tuyau" TLS authentifié par le serveur, et DOIT seulement être utilisé pour l'authentification du client. C'est-à-dire qu'on n'utilise pas les services de sécurité DIGEST-MD5 (confidentialité, intégrité, etc.).

2.3.2 Traitement de plusieurs accréditifs pour un compte

Lorsque plus d'un accréditif est mémorisé sous un seul compte, le client peut choisir un seul accréditif en utilisant la chaîne facultative de sélecteur d'accréditif.

Il n'y a pas de concept d'un "accréditif par défaut" – tous les accréditifs DOIVENT avoir un sélecteur unique associé pour ce compte. Le sélecteur est EXIGÉ pour les demandes de chargement et FACULTATIF pour les demandes de téléchargement. Si le sélecteur est omis dans une demande de téléchargement, cela DOIT être interprété comme une demande pour tous les accréditifs mémorisés.

Une valeur de chaîne de sélecteur vide (c'est-à-dire "") dans une demande de téléchargement d'accréditif est à interpréter comme si la chaîne de sélecteur était omise, c'est-à-dire une demande de téléchargement contenant cela est une demande pour tous les accréditifs.

C'est une erreur d'avoir plus d'un accréditif mémorisé sous le même compte lorsque tous ont la même chaîne de sélecteur d'accréditif.

2.3.3 Champs communs

Tous les messages envoyés au serveur PEUVENT contenir des valeurs de ProcessInfo. Ce champ PEUT être utilisé par d'autres spécifications ou pour des extensions de fabricant. Par exemple, un serveur peut exiger des clients qu'ils incluent un numéro de téléphone dans ce champ. Les messages de réponse d'informations contiennent une liste des types de ProcessInfo que le serveur prend en charge. Ce schéma d'extensibilité est similaire à celui utilisé dans [XKMS] et [XBULK].

Lorsque aucun message spécifique de réponse n'est défini pour une opération (par exemple pour UploadRequest), le transport va alors indiquer le succès ou l'échec.

Tous les messages de réponse définis ici PEUVENT contenir une chaîne Status, contenant une valeur destinée à la lecture par l'homme.

2.3.4 Format d'accréditif

Un certain nombre de messages impliquent l'élément Credential. Il a les champs suivants (tous les champs facultatifs peuvent se produire exactement zéro ou une fois sauf mention contraire) :

- CredentialSelector contient une chaîne par laquelle cet accréditif particulier (pour ce compte) peut être identifié.
- PayLoad contient soit un ds:KeyInfo, soit une autre forme d'accréditif. Les mises en œuvre DOIVENT prendre en charge la forme PKCS #15 de ds:KeyInfo définie ci-dessous (l'élément SacredPKCS15).
- LastModified est une chaîne qui contient l'heure (chez le serveur) de la dernière modification de cet accréditif.
- TimeToLive (facultatif) est une indication que les clients DEVRAIENT respecter, qui spécifie le nombre de secondes pendant lequel l'accréditif téléchargé est utilisable.
- ProcessInfo (facultatif) PEUT contenir toute information (typée) que le serveur est destiné à traiter. Si le serveur ne prend en charge aucune des données de ProcessInfo, il PEUT ignorer ces données.
- ClientInfo (facultatif) PEUT contenir toute information (typée) que le client est destiné à traiter, mais que le serveur DOIT ignorer. Si le client ne prend en charge aucune des données de ClientInfo, il PEUT ignorer ces données (par exemple si les ClientInfo sont spécifiques de l'appareil).

3. Profil BEEP pour SACRED

Le protocole décrit dans le présent mémoire est réalisé comme un profil de la [RFC3080].

De futurs mémoires pourront définir d'autres versions de profil BEEP pour SACRED. Lorsque un homologue BEEP envoie son accueil, il indique quels profils il veut prendre en charge. En conséquence, lorsque le client BEEP demande à commencer un canal, il indique les versions qu'il accepte, et si l'une d'elles est acceptable au serveur BEEP; ce dernier spécifie quel profil il démarre.

Identification de profil : http://iana.org/beep/sacred

Messages échangés durant la création de canal :
InfoRequest *(demande d'informations)*
CreateAccountRequest *(demande de création de compte)*
RemoveAccountRequest *(demande de suppression de compte)*
ModifyAccountRequest *(demande de modification de compte)*
DownloadRequest *(demande de téléchargement)*
UploadRequest *(demande de chargement)*
DeleteRequest *(demande de suppression)*
InfoResponse *(réponse d'informations)*
DownloadResponse *(réponse de téléchargement)*
error *(erreur)*
ok

Messages débutant des échanges biunivoques :
InfoRequest,
CreateAccountRequest,
RemoveAccountRequest,
ModifyAccountRequest,
DownloadRequest,
UploadRequest,
DeleteRequest

Messages dans des réponses positives :
ok,
InfoResponse,
DownloadResponse

Messages dans des réponses négatives : erreur

Messages dans des changements de un à plusieurs : aucun

Syntaxe de message : voir la Section 3

Sémantique de message : voir la Section 2

Informations de contact : voir la section Adresse de l'éditeur du présent mémoire

3.1 Initialisation de profil

Parce que toutes les opérations du profil SACRED sauf une ont des exigences de sécurité (voir le paragraphe 2.3.1) avant de commencer le profil SACRED, la session BEEP va probablement être réglée à utiliser soit http://iana.org/beep/TLS, soit http://iana.org/beep/TLS suivi par http://iana.org/SASL/DIGEST-MD5.
L'appendice B donne un exemple de réglage d'une session BEEP qui utilise DIGEST-MD5 (c'est-à-dire qu'il montre comment régler la sécurité BEEP).

De toutes façons, à l'achèvement du processus de négociation, une reprise du réglage se produit dans laquelle les deux homologues BEEP produisent un nouvel accueil. Consulter la Section 3 de la [RFC3080] pour un exemple de la façon dont un homologue BEEP peut choisir de produire des accueils différents selon que la confidentialité est ou non utilisée.
Tout message figurant sur la liste du paragraphe 3.2 ci-dessous peut être échangé durant l'initialisation du canal (voir le paragraphe 2.3.1.2 de la [RFC3080]), par exemple,

```
C: <start number='1'>
C:   <profile uri='http://iana.org/beep/sacred'>
C:          <![CDATA[<DownloadRequest ...>]]>
C:    </profile>
C: </start>

S: <profile uri='http://iana.org/beep/sacred'>
S:   <![CDATA[<DownloadResponse ...>]]>
S: </profile>
```

Noter que BEEP impose le codage et les limitations de longueur aux messages qui sont portés durant l'initialisation du canal.

3.2 Échange de profil

Tous les messages sont échangés comme "application/beep+xml" (voir au paragraphe 6.4 de la [RFC3080]) :

Rôle	Message	Réponse	Erreur
I	InfoRequest	InfoResponse	erreur
I	CreateAccountRequest	ok	erreur
I	RemoveAccountRequest	ok	erreur
I	ModifyAccountRequest	ok	erreur
I	DownloadRequest	DownloadResponse	erreur
I	UploadRequest	ok	erreur
I	DeleteRequest	ok	erreur

3.3 Traitement d'erreur

Le message "erreur" du paragraphe 2.3.1.5 de la [RFC3080] est utilisé pour porter les informations d'erreur. Normalement, après avoir marqué une erreur, un homologue va initier une libération en douceur de la session BEEP.
Les codes de réponse d'erreur BEEP suivants, tirés de la [RFC3080] sont à utiliser :

Code	Signification
421	service indisponible
450	action demandée non effectuée (par exemple, verrouillage déjà utilisé)
451	action demandée interrompue (par exemple, erreur locale de traitement)
454	échec temporaire d'authentification
500	erreur générale de syntaxe (par exemple, XML mal formé)
501	erreur de syntaxe des paramètres (par exemple, XML non valide)
504	paramètre non mis en œuvre
530	authentification exigée
534	mécanisme d'authentification insuffisant (par exemple, trop faible, séquence épuisée, etc.)
535	échec d'authentification
537	action non autorisée pour l'utilisateur
538	le mécanisme d'authentification exige le chiffrement
550	action demandée non effectuée (par exemple, aucun des profils demandés n'est acceptable)
553	paramètre invalide
554	échec de transaction (par exemple, violation de politique)

Les codes de réponse d'erreur spécifiques de SACRED suivants peuvent aussi être utilisés :

Code	Signification
555	Extension (ProcessInfo) utilisée non acceptée
556	Extension exigées (ProcessInfo) non présentes
557	StaleCredential (une mauvaise valeur de LastModified était contenue dans une UploadRequest).

3.4 Identité d'autorisation SASL

L'utilisation de l'identité d'autorisation SASL dans le présent protocole est spécifique de la mise en œuvre. Si elle est utilisée, l'identité d'autorisation n'est pas un substitut du champ Sélecteur d'accréditif, mais peut être utilisée pour affecter l'autorisation d'accès aux accréditifs.

4. Considérations relatives à l'IANA

L'IANA a enregistré le profil BEEP spécifié à la Section 4 : http://iana.org/beep/sacred

Le protocole SACRED DEVRAIT fonctionner sur l'accès 1118.

Le nom de service GSSAPI (exigé quand on utilise SASL) pour ce protocole DEVRA être "sacred".

5. Considérations sur la sécurité

La [RFC3157] demande que les spécifications déclarent comment elles traitent les vulnérabilités mentionnées ci-dessous.

V1. Un attaquant passif peut observer tous les paquets sur le réseau et mener ensuite une attaque de dictionnaire.
- L'utilisation de DIGEST-MD5 et/ou TLS contre cette vulnérabilité.

V2. Un attaquant peut tenter de se faire passer pour un serveur d'accréditifs afin d'amener un client à révéler en ligne des informations permettant une attaque de dictionnaire ultérieure.
- L'utilisation de l'authentification de serveur ou mutuelle contre cette vulnérabilité.

V3. Un attaquant peut tenter d'amener un client à déchiffrer un "texte chiffré" choisi et d'obtenir du client qu'il utilise le texte source résultant – l'attaquant peut alors être capable de conduire une attaque de dictionnaire (par exemple si le texte source résultant du "déchiffrage" d'une chaîne aléatoire est utilisé comme clé privée DSA).
- L'utilisation de l'authentification de serveur ou mutuelle contre cette vulnérabilité.

V4. Un attaquant pourrait écraser une entrée de répertoire de façon que lorsqu'un usager utilise ensuite ce qu'il pense être un bon accréditif, il expose les informations sur son mot de passe (et donc l'accréditif "réel").
- Les mises en œuvre de serveurs DEVRAIENT prendre des mesures pour protéger la base de données. Les clients PEUVENT utiliser le champ ClientInfo pour mémoriser, par exemple, une signature sur l'accréditif, qu'ils vont alors vérifier avant d'utiliser le composant privé.

V5. Un attaquant peut copier le répertoire d'un serveur d'accréditifs et mener une attaque de dictionnaire.
- Les mises en œuvre de serveurs DEVRAIENT prendre des mesures pour protéger la base de données.

V6. Un attaquant peut tenter de se faire passer pour un client afin d'essayer de faire révéler au serveur des informations permettant une attaque de dictionnaire ultérieure.
- Les exigences d'authentification mutuelle de ce protocole contrent cette vulnérabilité dans une grande mesure. De plus, les serveurs d'accréditifs PEUVENT choisir de fournir des mécanismes qui protègent contre les attaques de dictionnaire en ligne contre les mots de passe de compte d'utilisateur, soit par des tentatives d'accès répétées sur le compte d'un seul utilisateur (en faisant varier le mot de passe) soit en tentant d'accéder à de nombreux comptes d'usagers en utilisant le même mot de passe.

V7. Un attaquant peut persuader un serveur qu'une connexion réussie s'est produite, même si ce n'est pas vrai.
- L'authentification de client empêche cela.

V8. Au chargement, un attaquant peut écraser les accréditifs de quelqu'un d'autre sur le serveur.
- Seulement si il connaît déjà le mot de passe de compte (grâce à l'authentification mutuelle).

V9. Lors de l'utilisation de l'authentification fondée sur le mot de passe, un attaquant peut forcer un changement de mot de passe en un mot de passe connu ou "faible".
- L'authentification de client contre ceci.

V10. Un attaquant peut tenter une attaque par interposition *(man-in-the-middle)* pour différentes raisons...
- L'authentification mutuelle et le chiffrement des messages suivants empêche cela.

V11. L'utilisateur entre le mot de passe au lieu du nom.

- Comme le mécanisme DIGEST-MD5 n'est utilisé qu'après le réglage TLS, le nom d'utilisateur est aussi protégé.

V12. Un attaquant pourrait tenter diverses attaques de déni de service.

- Il n'y a pas de contre mesures spécifiques contre les attaques de DoS.

Si le message CreateAccountRequest a été envoyé sur un canal en clair (ou exposé par ailleurs) un attaquant pourrait alors monter une attaque de dictionnaire et récupérer le mot de passe de compte. C'est pourquoi le transport TLS authentifié par le serveur est EXIGÉ pour cette opération.

Si quelqu'un vole la base de données du serveur, il peut lancer une attaque de dictionnaire. Si l'attaque de dictionnaire réussit, l'attaquant peut déchiffrer les accréditifs de l'usager. Un attaquant qui a découvert le mot de passe de compte d'un utilisateur peut aussi charger de nouveaux accréditifs, en supposant que l'usager soit autorisé à modifier les accréditifs, parce que quelqu'un qui connaît le mot de passe de compte de l'usager est supposé être l'usager. Cependant, si quelqu'un vole la base de données du serveur et ne réussit pas à obtenir le mot de passe de compte de l'usager par une attaque de dictionnaire, il ne sera pas capable de télécharger les nouveaux accréditifs.

Les serveurs d'accréditifs DEVRAIENT incorporer des mesures qui agissent contre les attaques de déni de service. En particulier, ils DEVRAIT éliminer les connexions inactives et minimiser l'utilisation des ressources par des connexions non authentifiées. Un certain nombre de recommandations figurent dans [DDOS].

Diverses opérations du protocole SACRED dépendent de l'authentification du serveur fournie par le serveur authentifié par TLS. Les clients SACRED DEVRAIENT veiller à ce que le serveur correct soit à l'extrémité distante du "tuyau" TLS en effectuant les vérifications qui sont énumérées au paragraphe 3.1 de la [RFC2818]. Les clients DEVRAIENT aussi inclure le champ facultatif Nom de serveur BEEP dans leur message "start" et DEVRAIENT ensuite s'assurer que le nom de serveur BEEP est cohérent avec les vérifications sur le serveur TLS décrites dans la RFC 2818. Manquer à effectuer des vérifications pourrait permettre l'accès d'un serveur déguisé aux accréditifs de l'utilisateur.

Si le mot de passe de compte SACRED devait être utilisé dans un autre protocole moins sûr, utilisant DIGEST-MD5, il pourrait alors se produire une attaque par interposition (MITM, *man-in-the-middle*). Cependant, ce n'est pas le cas dans la mesure où le hachage de client DIGEST-MD5 comporte une "digest-uri-value" choisie par le client, qui dans le cas de SACRED sera "sacred/<serverName>". Dans une attaque de MITM, ces valeurs seront autre chose. Une attaque de MITM telle que décrite est donc déjouée, parce que digest-uri-value ne va pas correspondre à ce qu'attend le serveur SACRED.

6. Références

6.1 Références normatives

[PKCS15] "PKCS #15 v1.1: Cryptographic Token Information Syntax Standard," RSA Laboratories, juin 2000.

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

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

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

[RFC2831] P. Leach et C. Newman, "Utilisation de l'authentification par résumé comme mécanisme SASL", mai 2000. (*Obsolète, voir* RFC6331)

[RFC3080] M. Rose, "Cœur du protocole extensible d'échange de blocs (BEEP)", mars 2001. *(P.S.)*

[RFC3157] A. Arsenault, S. Farrell, "Disponibilité sécurisée des accréditifs - exigences", août 2001. *(Information)*

[RFC3268] P. Chown, "Suites de chiffrement de la norme de chiffrement évolué (AES) pour la sécurité de la couche Transport (TLS)", juin 2002. (*Obsolète, voir* RFC5246) *(P.S.)*

[RFC3275] D. Eastlake 3rd , J. Reagle, D. Solo, "Syntaxe et traitement de signature en langage de balisage extensible (XML)", mars 2002. *(D.S.)*

[XMLSCHEMA] D. Beech, M. Maloney, N. Mendelsohn, et H. Thompson. "XML Schema Part 1: Structures", Recommandation W3C, mai 2001. http://www.w3.org/TR/2001/REC- xmlschema-2-20010502/

6.2 Références pour information

[DDOS] "Recommendations for the Protection against Distributed Denial-of-Service Attacks in the Internet", http://www.iwar.org.uk/comsec/resources/dos/ddos_en.htm

[RFC2818] E. Rescorla, "HTTP sur TLS", mai 2000. *(Information, MàJ par* RFC5785, RFC7230*)*

[RFC3760] D. Gustafson, M. Just, M. Nystrom, "Disponibilité sécurisée des accréditifs (SACRED) – cadre du serveur d'accréditifs", avril 2004. *(Information)*

[XKMS] Hallam-Baker, P. (ed), "XML Key Management Specification", http://www.w3.org/TR/xkms2/

[XBULK] Hughes, M (ed), "XML Key Management Specification - Bulk Operation", http://www.w3.org/TR/xkms2-xbulk/

Remerciements

Radia Perlman (radia.perlman@sun.com) et Charlie Kaufman (charliek@microsoft.com) sont les co-auteurs des versions antérieures du présent document. Michael Zolotarev (mzolotar@tpg.com.au) a fait une grande partie du travail initial, adaptant une version antérieure pour l'utilisation de SRP (bien que SRP ait été ensuite abandonnée, une grande partie du cadre survit). Marshall Rose (mrose@dbc.mtview.ca.us) a beaucoup aidé, en particulier, avec le profil BEEP. Ei les personnes suivantes ont été impliquées activement dans les discussions de la liste de diffusion qui ont conduit au présent document : David Chizmadia, Dave Crocker (dcrocker@brandenburg.com), Lawrence Greenfield (leg+@andrew.cmu.edu), Dale Gustafson (degustafson@comcast.net), Mike Just (just.mike@tbs-sct.gc.ca), John Linn (jlinn@rsasecurity.com), Neal McBurnett (neal@bcn.boulder.co.us), Keith Moore (moore@cs.utk.edu), RL "Bob" Morgan (rlmorgan@washington.edu), Magnus Nystrom (magnus@rsasecurity.com), Eamon O'Tuathail (eamon.otuathail@clipcode.com), Gareth Richards (grichards@rsasecurity.com)

Bien sûr, toutes les erreurs restent de la responsabilité de l'éditeur.

Appendice A. Schéma XML

```
<?xml version="1.0" encoding="UTF-8"?>
  <schema
    targetNamespace="urn:sacred-2002-12-19"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:sacred="urn:sacred-2002-12-19"
    xmlns="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation=
    "http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd"/>
    <!-- extensibility holes -->
    <complexType name="ProcessInfoType">
     <sequence maxOccurs="unbounded">
      <any namespace="##other"/>
     </sequence>
    </complexType>
    <element name="ProcessInfo" type="sacred:ProcessInfoType"/>
    <complexType name="ClientInfoType">
     <sequence maxOccurs="unbounded">
      <any namespace="##other"/>
     </sequence>
    </complexType>
    <element name="ClientInfo" type="sacred:ClientInfoType"/>
    <!—Où mette les informations d'authenentication -->
    <complexType name="AuthInfoType">
     <choice maxOccurs="unbounded">
```

```
  <element name="DigestMD5AuthInfo">
   <complexType>
    <sequence>
     <element name="PasswordVerifier" type="base64Binary"/>
     <element name="Realm" type="string" />
    </sequence>
   </complexType>
  </element>
  <any namespace="##other"/>
 </choice>
</complexType>
<element name="AuthInfo" type="sacred:AuthInfoType"/>
<!-- paramètres de mécanisme d'authentification -->
<complexType name="AuthParamsType">
 <choice maxOccurs="unbounded">
  <element name=" DigestMD5AuthParams">
   <complexType>
    <sequence>
     <element name="Realm" type="string"
      minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
   </complexType>
  </element>
  <any namespace="##other"/>
 </choice>
</complexType>
<element name="AuthParams" type="sacred:AuthParamsType"/>
<!-- Messsages du protocole -->
<!-- Opérations de "traitement de compte" -->
<!—Demande d'informations -->
<element name="InfoRequest"/>
<element name="InfoResponse">
 <complexType>
  <sequence>
   <element name="Status" type="string" minOccurs="0"/>
   <element name="ServerId" type="string"/>
   <element ref="sacred:AuthParams"/>
   <element ref="sacred:ProcessInfo" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
<!—Demande de création de compte -->
<element name="CreateAccountRequest">
 <complexType>
  <sequence>
   <element name="UserId" type="string"/>
   <element ref="sacred:AuthInfo"/>
   <element ref="sacred:ProcessInfo" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
<!—Suppression de demande de compte -->
<element name="RemoveAccountRequest">
 <complexType>
  <sequence>
   <element ref="sacred:ProcessInfo" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
<!— Demande de changement de mot de passe -->
<element name="ModifyAccountRequest">
 <complexType>
  <sequence>
```

```
   <element ref="sacred:AuthInfo"/>
   <element ref="sacred:ProcessInfo" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
<!-- "run-time" operations -->
<!—Demande de téléchargement -->
<element name="DownloadRequest">
 <complexType>
  <sequence>
   <element name="CredentialSelector" type="string"
    minOccurs="0"/>
   <element ref="sacred:ProcessInfo" minOccurs="0"/>
  </sequence>
 </complexType>
</element>
<!—Réponse de téléchargement -->
<element name="DownloadResponse">
 <complexType>
  <sequence>
   <element name="Status" type="string" minOccurs="0"/>
   <element name="Credential" type="sacred:CredentialType"
    maxOccurs="unbounded"/>
  </sequence>
 </complexType>
</element>
<!—Demande de chargement -->
<element name="UploadRequest">
 <complexType>
  <sequence>
   <element name="Credential" type="sacred:CredentialType"/>
  </sequence>
 </complexType>
</element>
<element name="DeleteRequest">
  <complexType>
    <sequence>
      <choice>
        <sequence>
          <element name="CredentialSelector" type="string"/>
          <element name="LastModified" type="dateTime"
               minOccurs="0"/>
        </sequence>
        <element name="All"/>
      </choice>
      <element ref="sacred:ProcessInfo" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
<!-- Structures en rapport avec les accréditifs -->
<!-- A new ds:KeyInfo thing -->
<element name="SacredPKCS15" type="base64Binary"/>
<!-- accréditif -->
<complexType name="CredentialType">

 <sequence>
  <element name="CredentialSelector" type="string"/>
  <element name="LastModified" type="dateTime"/>
  <element name="Payload" type="ds:KeyInfoType" minOccurs="0"/>
  <element name="TimeToLive" type="string" minOccurs="0"/>
  <element ref="sacred:ProcessInfo" minOccurs="0"/>
  <element ref="sacred:ClientInfo" minOccurs="0"/>
 </sequence>
```

```
    </complexType>
</schema>
```

Appendice B. Exemple de réglage avec BEEP

Voici à quoi ressemble un réglage de BEEP pour l'authentification et la confidentialité en utilisant le DIGEST-MD5 de TLS et SASL :

```
L: <attente de connexion entrante>
I: <connexion ouverte>
```

... chaque homologue envoie un accueil indiquant les services qu'il offre ...

```
L: RPY 0 0 . 0 233
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/SASL/DIGEST-MD5' />
L:    <profile uri='http://iana.org/beep/TLS' />
L:    <profile uri='http://iana.org/beep/sacred' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

... l'initiateur commence un cannal pour TLS et porte une demande de commencer la négociation TLS...

```
I: MSG 0 1 . 52 149
I: Content-Type: application/beep+xml
I:
I: <start number='1' serverName="sacred.example.org">
I:    <profile uri='http://iana.org/beep/TLS'>
I:       &amp;lt;ready />
I:    </profile>
I: </start>
I: END
```

... l'écoutant crée le canal et porte qu'il est prêt à commencer TLS ...

```
L: RPY 0 1 . 233 112
L: Content-Type: application/beep+xml
L:
L: <profile uri='http://iana.org/beep/TLS'>
L:    &amp;lt;proceed />
L: </profile>
L: END
```

... à réception de la réponse, l'initiateur démarre TLS ...

... négociation réussie de la sécurité du transport ...

... un nouvel accueil est envoyé (voir la Section 9 de la RFC 3080) noter que l'écoutant n'annonce plus TLS (il fonctionne déjà)...

```
L: RPY 0 0 . 0 186
L: Content-Type: application/beep+xml
L:
L: <greeting>
```

```
L:    <profile uri='http://iana.org/beep/SASL/DIGEST-MD5' />
L:    <profile uri='http://iana.org/beep/sacred' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

... l'initiateur démarre un canal pour DIGEST-MD5 et porte les informations d'initialisation pour le mécanisme ...

```
I: MSG 0 1 . 52 178
I: Content-Type: application/beep+xml
I:
I: <start number='1'>
I:    <profile uri='http://iana.org/beep/SASL/DIGEST-MD5'>
I:       &lt;blob> ... &lt;/blob>
I:    </profile>
I: </start>
I: END
```

... l'écoutant crée le canal et porte un défi ...

```
L: RPY 0 1 . 186 137
L: Content-Type: application/beep+xml
L:
L: <profile uri='http://iana.org/beep/SASL/DIGEST-MD5'>
L:     &lt;blob> ... &lt;/blob>
L: </profile>
L: END
```

... l'initiateur envoie une réponse au défi ...

```
I: MSG 1 0 . 0 58
I: Content-Type: application/beep+xml
I:
I: <blob> ... </blob>
I: END
```

... l'écoutant accepte le défi et dit à l'initiateur qu'il est maintenant authentifié ...

```
L: RPY 1 0 . 0 66
L: Content-Type: application/beep+xml
L:
L: <blob status='complete' />
L: END
```

... l'initiateur démarre un canal pour SACRED et porte sa demande SACRED initiale ...

```
I: MSG 0 2 . 230 520
I: Content-Type: application/beep+xml
I:
I: <start number='3'>
I:    <profile uri='http://iana.org/beep/sacred' />
I:       &lt;?xml version="1.0" encoding="UTF-8"?>
I:       &lt;sacred:DownloadRequest
I:         xmlns:sacred="urn:sacred-2002-12-19"
I:         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
I:         xsi:schemaLocation="urn:sacred-2002-12-19 sacred.xsd">
I:         &lt;CredentialSelector>
I:                 magnus-credentials&lt;/CredentialSelector>
I:       &lt;/sacred:DownloadRequest>
```

```
I: </start>
I: END

... l'écoutant crée le canal et porte la réponse à la demande initiale SACRED

L: RPY 0 2 . 323 805
L: Content-Type: application/beep+xml
L:
L: <profile uri='http://iana.org/beep/sacred' />
L:    &lt;?xml version="1.0" encoding="UTF-8"?>
L:    &lt;sacred:DownloadResponse
L:      xmlns:sacred="urn:sacred-2002-12-19"
L:      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
L:      xsi:schemaLocation="urn:sacred-2002-12-19 sacred.xsd">
L:      &lt;Status>Success&lt;/Status>
L:      &lt;Credential>
L:        &lt;CredentialSelector>
L:            magnus-credential&lt;/CredentialSelector>
L:        &lt;LastModified>2002-11-22T00:00:08Z&lt;/LastModified>
L:        &lt;Payload>
L:           &lt;sacred:SacredPKCS15
L:             xmlns:sacred="urn:sacred-2002-12-19">GpM7
L:           &lt;/sacred:SacredPKCS15>
L:        &lt;/Payload>
L:      &lt;/Credential>
L:    &lt;/sacred:DownloadResponse>
L: </profile>
L: END
```

Appendice C. Fourniture de SACRED avec d'autres protocoles

SACRED peut être mis en œuvre dans un environnement non BEEP, pourvu qu'avant l'envoi de toute PDU SACRED, le protocole d'application soit protégé selon les exigences de sécurité du paragraphe 2.3.

Par exemple, si SACRED est provisionné comme charge utile d'un protocole d'application qui prend en charge SASL et TLS, la négociation SASL et/ou TLS appropriée doit alors se faire avec succès avant l'échange de PDU Sacred.

Autrement, si le protocole d'application ne prend pas en charge SASL, une ou plusieurs PDU sont alors définies pour faciliter une négociation SASL, et la négociation appropriée doit survenir avant d'échanger des PDU SACRED.

Adresse de l'éditeur

Stephen Farrell,
Distributed Systems Group,
Computer Science Department,
Trinity College Dublin,
IRELAND
téléphone : +353-1-608-3070
mél : stephen.farrell@cs.tcd.ie

Déclaration complète de droits de reproduction

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org .

Remerciement

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