Groupe de travail Réseau
Request for Comments : 3760
Catégorie : Information
Traduction Claude Brière de L'Isle

D. Gustafson, Future Foundation
M. Just, Treasury Board of Canada
M. Nystrom, RSA Security
avril 2004

Disponibilité sécurisée des accréditifs (SACRED) Cadre de serveur d'accréditifs

Résumé
Avec l'augmentation du nombre, et plus particulièrement du nombre de types différents, d'appareils qui se connectent à l'Internet, la mobilité des accréditifs devient un problème pour la normalisation par l'IETF. Le présent document répond aux exigences sur les protocoles pour un échange sécurisé des accréditifs mentionnées dans la [RFC3157], en présentant un cadre de protocole abstrait.

Table des Matières

1. Introduction

Les accréditifs numériques, comme les clés privées et les certificats correspondants, sont utilisés pour prendre en charge divers protocoles de l'Internet, par exemple, S/MIME, IPsec, et TLS. Dans un certain nombre d'environnements, les usagers souhaitent utiliser les mêmes accréditifs sur les différents appareils d'utilisateur final. 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, ce n'est pas très pratique. De plus, avec certains appareils, en particulier les appareils sans fil et autres appareils avec de fortes contraintes, les outils requis n'existent simplement pas.

Le présent document propose un cadre général pour l'échange sécurisé de tels accréditifs et fournit les grandes lignes qui vont aider à guider le développement d'un ou plusieurs protocoles d'échange sécurisé des accréditifs disponibles (SACRED, *Securely Available CREDentials*).

2. Vue d'ensemble fonctionnelle

Les exigences pour SACRED sont pleinement décrites dans la [RFC3157]. Ces exigences supposent que deux architectures de réseau distinctement différentes seront créées pour prendre en charge l'échange d'accréditifs pour les usagers en itinérance :

a) Échange d'accréditifs client/serveur
b) Échange d'accréditifs d'homologue à homologue

Le présent document décrit le cadre pour un ou plusieurs protocoles d'échange d'accréditifs client/serveur.

Dans tous les cas, des méthodes d'authentification d'utilisateur adéquates seront utilisées pour s'assurer que les accréditifs ne sont pas divulgués à des tiers non autorisés. De même, des méthodes d'authentification de serveur adéquates seront utilisées pour s'assurer que les informations d'authentification de chaque client (voir au paragraphe 2.1) ne sont pas compromises, et pour s'assurer que les usagers en itinérance interagissent avec les serveurs d'accréditifs prévus/autorisés.

2.1 Définitions

Cette section donne les définitions de plusieurs termes ou phrases utilisés tout au long du document.

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

informations d'authentification de client : informations qui sont présentées par le client à un serveur pour authentifier le client. Cela peut inclure un jeton de mot de passe, une chaîne d'enregistrement qui peut avoir été reçue hors bande (et éventuellement utilisée pour enregistrer initialement un usager en itinérance) ou des données signées avec une clé de signature appartenant au client (par exemple, au titre de l'authentification de client TLS [RFC2246]).

accréditifs : objets cryptographiques et données en rapport utilisés pour prendre en charge des communications sûres sur l'Internet. Les accréditifs peuvent consister en paires de clés publique/privée, en clés symétriques, en certificats de clé publique X.509, en certificats d'attributs, et/ou en données d'application. Il existe plusieurs formats normalisés de représentation d'accréditifs, par exemple, [PKCS12], [PKCS15] (voir ci-dessous "accréditifs sécurisés").

clé de passe : clé symétrique, dérivée d'un mot de passe.

mot de passe : chaîne de caractères connue seulement d'un client et utilisée pour les besoins d'authentification auprès d'un serveur et/ou pour sécuriser des accréditifs. Un usager peut être obligé de mémoriser plus d'un mot de passe.

jeton de mot de passe : valeur dérivée d'un mot de passe en utilisant une fonction unidirectionnelle qui peut être utilisée par un client pour s'authentifier auprès d'un serveur. Un jeton de mot de passe peut être déduit d'un mot de passe en utilisant par exemple une fonction de hachage unidirectionnelle.

accréditifs sécurisés : ensemble d'un ou plusieurs accréditifs qui ont été sécurisés cryptographiquement, par exemple, chiffrés ou munis d'un code d'accès au message avec une clé de passe. Les accréditifs sécurisés peuvent être protégés en utilisant plus d'une couche de chiffrement, par exemple, l'accréditif est sécurisé avec une clé de passe correspondant à un mot de passe d'utilisateur et aussi par une clé connue seulement du serveur (la forme mémorisée de l'accréditif). Durant le transfert sur le réseau, l'accréditif protégé par une clé de passe peut être protégé par une couche de chiffrement supplémentaire en utilisant une clé symétrique choisie par le serveur d'accréditifs (par exemple, la forme transmise).

protocole de mot de passe fort : un protocole qui authentifie en toute sécurité les clients auprès des serveurs (voir par exemple [SPEKE] pour une définition plus détaillée de cela) où le client a seulement besoin de mémoriser un petit secret (un mot de passe) et ne porte pas d'autres informations secrètes, et où le serveur porte un vérificateur (jeton de mot de passe) qui lui permet d'authentifier le client. Un secret partagé est négocié entre client et serveur et est utilisé pour protéger les données échangées ensuite.

Noter la distinction entre un "mot de passe de compte" et un "mot de passe d'accréditif". Un mot de passe de compte (et le jeton de mot de passe correspondant) est utilisé pour authentifier un serveur d'accréditifs et pour négocier une clé qui fournit le chiffrement de niveau session entre client et serveur.

Un mot de passe d'accréditif est utilisé pour déduire une clé de passe qui sera utilisée pour fournir un chiffrement et une authentification persistants pour un accréditif mémorisé. Les documents de normes d'accréditif sécurisés applicables (par exemple [PKCS15]) décrivent les détails techniques des techniques spécifiques de chiffrement fondé sur le pot de passe (PBE, *password-based-encryption*) qui sont utilisées pour protéger les accréditifs contre l'utilisation non autorisée.

Bien que la même valeur de mot de passe puisse être utilisée pour fournir les deux services, il est probable que des clés de passe différentes, spécifiques d'un algorithme, seront générées à partir du mot de passe (c'est-à-dire, à cause de valeurs de sel différentes, etc.).

De plus, bien qu'il puisse être plus convenable pour un utilisateur de se rappeler d'un seul mot de passe, des politiques de sécurité différentes (par exemple, les règles de mot de passe) entre le serveur d'accréditifs et le producteur d'accréditifs peuvent résulter en ce qu'un utilisateur doive se rappeler de multiples mots de passe.

2.2 Accréditifs

Le présent document traite de l'échange sécurisé et de la gestion en ligne des accréditifs dans un environnement d'itinérance ou mobile. Les accréditifs PEUVENT être utilisables avec tout appareil d'utilisateur final qui peut se connecter à l'Internet, comme :

- un ordinateur portable ou de bureau
- un téléphone mobile
- un assistant numérique personnel (PDA, *personal digital assistant*)
- etc.

Le système d'utilisateur final peut, facultativement, mémoriser ses informations d'accréditif sur un appareil spécial qui fournit une portabilité et une protection améliorées des accréditifs d'utilisateur.

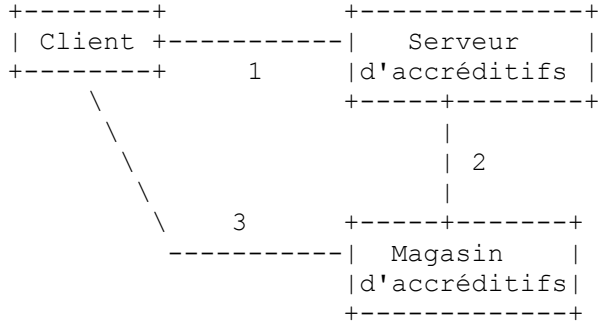
Comme l'accréditif contient généralement des informations sensibles qui ne sont connues que du détenteur de l'accréditif, les accréditifs NE DOIVENT PAS être envoyés en clair durant la transmission réseau et NE DEVRAIENT PAS être en clair lors de la mémorisation sur un appareil d'utilisateur final comme une disquette ou un disque dur. Pour cette raison, on définit un accréditif sécurisé. Tout au long du présent document, on suppose que, au moins du point de vue du protocole, un accréditif sécurisé est un objet de données opaque (et au moins partiellement protégé quant à la confidentialité et l'intégrité) qui peut être utilisé par un appareil connecté au réseau. Une fois téléchargés, les clients doivent être capables de récupérer leurs accréditifs à partir de ce format opaque.

Au minimum, tous les formats d'accréditif pris en charge DEVRAIENT assurer la protection de la confidentialité et de l'intégrité pour les clés privées, les clés secrètes, et tous les autres objets de données qui doivent être protégés de la divulgation ou de la modification. Normalement, ces capacités de sécurité font partie du format de base d'accréditif de façon que l'accréditif (par exemple, un fichier de données) soit protégé lorsque il est mémorisé sur un disque dur, une disquette souple, etc.

Durant la transmission sur le réseau, l'accréditif sécurisé est protégé par une seconde (externe) couche de chiffrement. La couche de chiffrement externe est créée en utilisant une clé de chiffrement de niveau session qui est déduite durant le processus d'authentification mutuelle. Effectivement, les accréditifs sécurisés traversent un "tunnel chiffré" qui assure une couche supplémentaire de protection de la confidentialité pour les informations d'accréditifs (et toutes les autres informations) échangées.

2.3 Architecture réseau

Le diagramme de réseau ci-dessous montre les composants impliqués dans le cadre client/serveur SACRED.



Client – entité qui veut restituer ses accréditifs d'un serveur d'accréditifs.

Serveur d'accréditifs - serveur qui télécharge des accréditifs sécurisés et les charge à partir du client. Le serveur est responsable de l'authentification du client pour s'assurer que les accréditifs sécurisés ne sont échangés qu'avec un utilisateur final approprié. Le serveur d'accréditifs est authentifié auprès du client pour assurer que les informations d'authentification du client ne sont pas compromises et qu'ainsi l'utilisateur peut faire confiance aux accréditifs restitués.

Magasin d'accréditifs – répertoire des accréditifs sécurisés. Il peut y avoir des dispositifs de contrôle d'accès mais ils ne sont généralement pas suffisants par eux-mêmes pour sécuriser les accréditifs. Le serveur d'accréditifs peut être capable de partager les accréditifs entre plusieurs magasins d'accréditifs pour la redondance ou pour fournir des niveaux supplémentaires de protection pour les accréditifs d'utilisateur.

Protocole 1 - protocole utilisé pour authentifier le client et le serveur d'accréditifs, et charger et télécharger les accréditifs d'utilisateur à partir d'un serveur d'accréditifs.

Protocole 2 - protocole utilisé par le serveur d'accréditifs pour mémoriser et restituer les accréditifs d'utilisateur (LDAP, LDAP/SSL, ou autres).

Protocole 3 - protocole utilisé par le client pour mémoriser et restituer les accréditifs d'utilisateur du magasin d'accréditifs (LDAP, LDAP/SSL, ou autres).

Ce cadre décrit les grandes lignes de la conception du protocole 1. Les protocoles 2 et 3 sont en relation étroite (mais sortent du domaine d'application du présent document) et pourraient être mis en œuvre en utilisant des protocoles standard, comme LDAP ou LDAP sécurisé, ou d'autres protocoles standard ou propriétaires. Noter aussi que tout protocole de serveur d'administrateur-accréditif est supposé être spécifique du fabricant du serveur et ne fait pas pour l'instant l'objet de l'effort de normalisation SACRED.

Il n'est pas interdit aux clients d'échanger des accréditifs directement avec un magasin d'accréditifs (ou tout autre serveur de leur choix). Cependant, l'authentification mutuelle avec les usagers en itinérance et un niveau cohérent de protection des données d'accréditif lorsque elles sont mémorisées sur les serveurs du réseau et pendant le transit sont fournis par les protocoles SACRED échangés avec le serveur d'accréditifs. Selon la conception du serveur d'accréditifs, les accréditifs d'utilisateur peuvent s'écouler à travers le serveur d'accréditifs vers le magasin d'accréditifs ou directement entre le client et le magasin d'accréditifs.

Aussi, les utilisateurs peuvent télécharger leurs accréditifs sur plusieurs serveurs d'accréditifs pour obtenir des niveaux de disponibilité améliorés. La coordination (répétition automatique) des informations d'utilisateur ou de données d'accréditifs entre plusieurs serveurs d'accréditifs sort actuellement du domaine d'application du présent document.

3. Cadre du protocole

Cette section donne les grandes lignes de la description des protocoles client/serveur qui peuvent être utilisés pour échanger et gérer des accréditifs SACRED.

Le protocole d'échange d'accréditif client/serveur se fonde sur trois opérations de base abstraites : "GET", "PUT", et "DELETE". Le protocole d'échange d'accréditif sécurisé se réalise comme suit :

connexion - le client initie une connexion à un serveur d'accréditifs dans le but d'échanger des accréditifs sécurisés.

authentification mutuelle/négociation de clé – en utilisant un protocole de mot de passe fort (ou équivalent) le client s'authentifie auprès du serveur, le serveur s'authentifie auprès du client, et une clé de chiffrement de niveau session est négociée. Les détails de l'échange de protocole d'authentification mutuelle dépendent de la méthode d'authentification particulière utilisée. Dans tous les cas, le résultat final est d'authentifier le client auprès du serveur et le serveur auprès du client, et d'établir un fort secret partagé entre les deux parties.

demandes du client – le client SACRED produit une ou plusieurs demandes générales d'échange d'accréditif (par exemple, GET, PUT, ou DELETE).

réponses de serveur - le serveur d'accréditifs SACRED répond à chaque demande, en effectuant avec succès l'opération ou en indiquant une erreur appropriée.

clôture - le client indique qu'il n'a pour l'instant plus de demande pour le serveur. Le contexte de sécurité entre client et serveur n'est plus nécessaire. La clôture est une opération logique de gestion de session.

déconnexion - les parties déconnectent la connexion de niveau transport entre client et serveur. Noter que "connexion" et "déconnexion" sont des opérations logiques de couche transport qui incluent l'échange de protocole entre les deux processus communicants.

Chaque opération de haut niveau d'échange d'accréditif est constituée d'une série de paires de demande-réponse. Le client initie chaque demande, que le serveur traite avant de retourner la réponse appropriée. Chaque demande doit se terminer (le serveur fait rapport de la réussite ou de l'échec) avant que le client produise la demande suivante. Le serveur DEVRAIT vouloir servir au moins une demande de chargement ou téléchargement à la suite d'une authentification mutuelle réussie mais l'une ou l'autre partie peut terminer à tout moment la connexion logique.

Dans les paragraphes qui suivent, les accréditif sécurisés et les valeurs qui s'y rapportent sont représentés à l'aide de la notation suivante :

SC-x est le fichier d'accréditifs sécurisés, qui inclut un champ Identifiant de format et des données d'accréditif. Les données d'accréditif sont un objet de données chiffrées opaque (par exemple, un fichier PKCS#15 ou PKCS#12). L'identifiant de format est nécessaire pour analyser correctement les données d'accréditif.

Name-x est un sélecteur ou localisateur défini par compte (un nom facile à mémoriser) qui est utilisé pour indiquer un accréditif sécurisé spécifique. Le nom de chaque accréditif mémorisé sous un certain compte d'utilisateur DOIT être unique, par exemple, il peut y avoir un accréditif appelé "financier" et un autre appelé "santé", etc. Au minimum, les noms d'accréditif DOIVENT être uniques sur un certain nom de compte/usager. Lorsque aucun nom n'est fourni pour une opération GET, tous les accréditifs mémorisées pour ce nom d'utilisateur seront retournés.

ID-x est un indicateur de version d'accréditif distinct qui PEUT être utilisé pour demander une opération GET/PUT/DELETE conditionnelle. Cette valeur d'identifiant d'accréditif DEVRAIT contenir la date et l'heure de "dernière modification" du serveur (par exemple, l'heure à laquelle cette version d'accréditif particulière a été mémorisée sur le serveur) et PEUT contenir des informations supplémentaires comme un numéro de séquence on une empreinte (complète ou partielle) d'accréditif qui sera utilisée pour s'assurer que l'identifiant d'accréditif est unique par rapport aux autres versions d'accréditif mémorisées sous le même compte d'utilisateur et nom d'accréditif.

Tous les accréditifs désignés par un nom peuvent être accédés par authentification sous un seul nom d'utilisateur. Si un utilisateur a besoin d'utiliser ou préfère utiliser plus d'un mot de passe d'authentification (et/ou méthode d'authentification) distinct pour protéger l'accès à plusieurs accréditifs sécurisés, il DEVRAIT enregistrer ces accréditifs sous des noms d'utilisateur/comptes distincts, un pour chaque différente méthode d'authentification utilisée.

3.1 Chargement d'accréditif

L'objet de l'opération de chargement d'accréditif est de permettre à un client d'enregistrer de nouveaux accréditifs, ou de remplacer des accréditifs actuellement mémorisés (par exemple, des accréditifs qui peuvent avoir été mis à jour par le client en utilisant un logiciel de gestion de clés approprié).

Le cadre pour le chargement d'accréditif, tel que mis en œuvre en utilisant l'opération PUT, est :
- le client et le serveur établissent une session mutuellement authentifiée et négocient un secret partagé ;
- le client produit ensuite un message PUT qui contient l'accréditif chargé et les champs de données qui s'y rapportent ;
- le serveur va répondre au PUT, indiquant que l'accréditif a bien été mémorisé sur le serveur ou qu'une erreur s'est produite.

La demande PUT du client PEUT contenir un champ d'identifiant facultatif (identifiant d'accréditif). S'il est présent, le nouvel accréditif ne sera mémorisé que si un accréditif et identifiant d'accréditif du même nom est actuellement mémorisé sur le serveur (par exemple, une opération logique REPLACE est effectuée). Le serveur DOIT retourner une erreur si un client tente de remplacer un accréditif qui n'existe pas sur le serveur.

La réponse du serveur d'accréditifs à une demande PUT DOIT contenir un identifiant de version d'accréditif (identifiant d'accréditif) pour l'accréditif nouvellement mémorisé qui PEUT être utilisé par les clients pour optimiser les opérations suivantes de téléchargement et éviter des discordances de version d'accréditif.

3.1.1 Séquence de protocole de chargement d'accréditif

Voici un exemple de séquence de protocole de "chargement d'accréditif" :

```
Client                          Serveur
  < connexion >                 -->
  <--- authentification mutuelle --->
  < PUT SC-1, Nom-1, [ID-1] > -->
                  <--    < Nom-1, nouvel-ID-1 >
  < PUT SC-2, Nom-2, [ID-2] > -->
                  <--    < Nom-2, nouvel-ID-2 >
                     ...
  < close >                  -->
                  <--    OK (+ déconnexion)
```

nouvel-ID-x est l'identifiant d'accréditif de l'accréditif nouvellement mémorisé.

3.2 Téléchargement d'accréditif

Les clients en itinérance peuvent télécharger leurs accréditifs à tout moment après qu'ils ont été chargés sur le serveur.

Le cadre pour le téléchargement d'un accréditif, tel que mis en œuvre en utilisant l'opération GET, est :
- le client DEVRAIT authentifier le serveur,
- l'usager DOIT être authentifié (par le serveur),
- une demande GET est produite pour le téléchargement de l'accréditif,
- la réponse contient l'accréditif et l'identifiant de format.

L'accréditif d'utilisateur spécifique qui est demandé peut être identifié par son nom dans le message envoyé au serveur d'accréditifs. Si cela réussit, la réponse DOIT contenir l'élément de données d'accréditif demandé (identifiant de format et données) comme défini ci-dessus.

Si l'utilisateur produit une demande GET avec un champ Nom d'accréditif NUL, le serveur DEVRAIT retourner tous les accréditifs mémorisés sous le compte d'utilisateur actuel.

Facultativement, le client PEUT inclure un identifiant d'accréditif pour indiquer une demande de téléchargement conditionnel. Dans ce cas, le serveur ne va retourner l'accréditif demandé que si et seulement si l'identifiant de l'accréditif actuellement mémorisé sur le serveur NE correspond PAS à l'identifiant spécifié.

Le serveur devrait retourner l'accréditif demandé ou une réponse distincte indiquant que le téléchargement conditionnel n'a pas été effectué (par exemple, le client a déjà une copie de cet accréditif là).

3.2.1 Séquence de protocole de téléchargement d'accréditif

Voici un exemple d'une séquence de protocole de "téléchargement d'accréditif" :

```
Client                          Serveur
  < connexion >                  -->
  <--- authentification mutuelle -->
  < GET Nom-1, [ID-1] >         -->
                <--    < SC-1, ID-1' >
  < GET Nom-2, [ID-2] >          -->
                <--    < réponse GET >
                   ...
  < close >                      -->
                <--    OK (+ déconnexion)
```

Noter que pour la seconde demande, aucun accréditif n'a été retourné car ID-2, tel qu'inclus dans la demande du client, correspondait à l'identifiant pour l'accréditif Nom-2.

3.3 Suppression d'accréditif

Le cadre pour la suppression d'accréditif, tel que mis en œuvre avec l'opération DELETE, est :

- le serveur d'accréditif DOIT être authentifié (par le client) en utilisant une séquence de protocole qui dépend de la méthode,
- l'usager DOIT être authentifié (par le serveur) en utilisant une séquence de protocole qui dépend de la méthode,
- l'usager envoie alors un message de demande DELETE qui contient le nom de l'accréditif à supprimer.
- Facultativement, le client peut inclure un identifiant d'accréditif dans la demande DELETE. Dans ce cas, l'accréditif sera supprimé si l'identifiant de la demande correspond à l'identifiant de l'accréditif actuellement mémorisé sur le serveur. Cela peut être fait pour s'assurer qu'un client qui a l'intention de supprimer son accréditif mémorisé ne supprime pas par erreur une version différente de l'accréditif.

3.3.1 Séquence de protocole de suppression d'accréditif

Voici un exemple de séquence de protocole de "suppression d'accréditif" :

```
Client                          Serveur
 < connexion >                  -->
 <----- authentification mutuelle ---->
 < DEL Nom-1, [ID1] >     -->
            <--     < Nom-1 supprimé >
 < DEL Nom-2, [ID2] >     -->
              <--     < Nom-2 supprimé >
                      ...
 < close >                      -->
              <--     OK (+ déconnexion)
```

3.4 Gestion d'accréditif

Noter que les trois opérations définies ci-dessus (GET, PUT, DELETE) peuvent être utilisées pour effectuer les opérations de base de gestion d'accréditif :

- ajout d'un nouvel accréditif sur le serveur,
- mise à jour (remplacement) d'un accréditif existant,
- suppression d'un accréditif existant.

Les informations fournies pour ces opérations de base peuvent être utilisées pour aider à guider la conception d'opérations plus complexes comme l'enregistrement d'un usager (ajouter un compte), le désenregistrement d'un usager (supprimer un compte), le changement d'un mot de passe de compte, ou établir la liste de tous les accréditifs.

Noter que, dans le cas où il existe déjà un accréditif avec le même nom sur le serveur, télécharger un accréditif NUL est équivalent logiquement à supprimer un accréditif mémorisé précédemment.

4. Considérations de protocole

4.1 Formats d'accréditifs sécurisés

Pour s'assurer que les accréditifs créés sur un appareil, et qui en sont téléchargés, peuvent être chargés et utilisés sur tout autre appareil, il est nécessaire de définir un seul format d'accréditif "de mise en œuvre obligatoire" qui doit être accepté par toutes les mises en œuvre de client conformes.

Au moins deux formats d'accréditif bien définis sont disponibles à ce jour : [PKCS12] et [PKCS15].

D'autres formats d'accréditif facultatifs peuvent aussi être pris en charge si nécessaire. Par exemple, des formats d'accréditif supplémentaires pourraient être définis pour être utilisés avec des appareils spécifiques (compatibles) d'un client. Chaque format d'accréditif DOIT fournir une protection adéquate de la confidentialité pour les accréditifs d'utilisateur lorsque ils sont mémorisés sur une disquette souple, un disque dur, etc.

Tout au long du présent document, l'accréditif est traité comme un objet de données opaque (chiffré) et, à ce titre, le format d'accréditif n'affecte pas le protocole d'échange d'accréditif de base.

4.2 Méthodes d'authentification

L'authentification est d'une importance vitale pour assurer que les accréditifs sont acceptés du seul utilisateur final autorisé,

et livrés à lui seul. Si un accréditif non sécurisé est livré à un tiers, l'accréditif peut être plus facilement compromis. Si un accréditif est accepté d'une partie non autorisée, l'utilisateur peut être conduit à utiliser un accréditif qui a été substitué par un attaquant (par exemple, un attaquant pourrait remplacer un accréditif récent par un plus ancien qui appartient au même utilisateur).

Idéalement, la liste des méthodes d'authentification devrait rester ouverte, permettant que de nouvelles méthodes soient ajoutées lorsque des besoins sont identifiés et qu'elles sont disponibles. Pour tous les accréditifs, la méthode d'authentification de l'usager et les données sont définies lorsque un utilisateur est d'abord enregistré auprès du serveur d'accréditifs et peut être mis à jour de temps en temps par la suite par l'utilisateur autorisé.

Pour protéger adéquatement les accréditifs d'utilisateur contre la divulgation non autorisée ou la modification dans un environnement d'itinérance, toutes les méthodes d'authentification SACRED DOIVENT assurer la protection des accréditifs d'utilisateur dans les environnements réseau où des attaquants peuvent tenter d'exploiter de potentielles vulnérabilités de la sécurité. Voir les exigences pour SACRED [RFC3157], paragraphe 3.1, "Vulnérabilités".

Au minimum, chaque méthode d'authentification SACRED DEVRAIT assurer que :
- le serveur authentifie le client,
- le client authentifie le serveur,
- le client et le serveur négocient en toute sécurité (ou déduisent) une clé secrète, cryptographiquement forte, (par exemple, une clé de session),
- l'échange d'un ou plusieurs accréditifs d'utilisateur est protégé en utilisant cette clé de session.

On s'attend à ce que tous les protocoles client/serveur SACRED fournissent toutes ces fonctions de sécurité de base. Certains protocoles existants d'authentification qui pourraient être utilisés à cette fin incluent :
- des protocoles de mot de passe forts,
- TLS

Les paragraphes 4.2.1 et 4.2.2 donnent quelques lignes directrices sur le moment où utiliser ces méthodes d'authentification sur la base des capacités génériques de sécurité qu'elles fournissent et les éléments de sécurité (mots de passe, paires de clés, certificats d'utilisateur, certificats de CA) qui doivent être disponibles au client SACRED.

4.2.1 Protocoles de mots de passe forts

Des protocoles de mots de passe forts tels que ceux décrits dans [RFC2945], [BM92], [BM94], et [SPEKE] PEUVENT être utilisés pour assurer l' authentification mutuelle et la confidentialité pour les protocoles SACRED.

Tous les protocoles de mots de passe forts exigent que des valeurs spécifiques de l'utilisateur (c'est-à-dire, un jeton de passe et les valeurs qui s'y rapportent) soient configurées au sein du serveur. Seul celui qui connaît le mot de passe peut calculer la valeur du vérificateur. Elle doit être livrée en toute sécurité au serveur au moment où le client établit une relation avec le serveur. Au moment de la connexion, les messages sont échangés entre les deux parties et des algorithmes complémentaires sont utilisés pour calculer une valeur partagée commune connue du seul utilisateur légitime et du serveur. Les deux parties déduisent une clé forte (symétrique) qui peut être utilisée pour sécuriser les communications entre les deux parties.

4.2.2 Authentification TLS

L'authentification TLS peut être soit mutuelle entre le client et le serveur, soit unilatérale lorsque seul le serveur est authentifié auprès du client. Ces options sont décrites dans les deux paragraphes suivants. Dans les deux cas, TLS peut être utilisé pour authentifier le serveur chaque fois que le client TLS a été préconfiguré avec les certificats indispensables pour valider la chaîne de certificats du serveur (incluant la vérification de l'état de révocation).

Authentification de serveur TLS (sTLS)

TLS fournit une capacité de base de session sécurisée (parfois appelée TLS côté serveur) par laquelle le client authentifie le serveur et une paire de clés de chiffrement de niveau session ²qui est échangée en toute sécurité entre client et serveur. À la suite de l'authentification du serveur et de l'établissement du contexte de sécurité, toutes les demandes du client et les réponses du serveur échangées sont protégées quant à leur intégrité et leur confidentialité.

Les concepteurs de protocoles et leurs mises en œuvre devraient avoir conscience que la souplesse de la méthode d'authentification du serveur par TLS fondée sur le certificat crée des risques pour la sécurité qui doivent être maîtrisés. Précisément, il est nécessaire de s'assurer que l'usager est connecté au serveur d'accréditifs prévu (site sûr) et à aucun autre. La norme TLS v1.0 [RFC2246] identifie les bases de la gestion de ce risque en son paragraphe F.3 (voir aussi le paragraphe 5.2 du présent document) : "Les mises en œuvre et les usagers doivent être prudents quand ils décident quels certificats et

autorités de certificats sont acceptables ; une autorité de certificat malhonnête peut causer des dommages terrifiants."

Noter aussi qu'une mise en œuvre fautive de traitement de la chaîne de certificats de serveur TLS (de complexité croissante) par le client SACRED, pourrait conduire à une compromission similaire, permettant la réussite d'une attaque par usurpation d'identité du serveur d'accréditifs ou par interposition.

Une approche d'ingénierie qui fournit une méthode améliorée ou augmentée d'authentification du serveur peut être garantie pour les conceptions de protocole SACRED. Il est aussi important de comprendre que la simple mise en couches de protocole de sécurité développée de façon indépendante (par exemple, en utilisant BEEP ou des techniques de mise en couche similaires) produit un protocole de sécurité multi couches complexe qui peut être facilement vaincu par une attaque spécifique de la combinaison capable d'exposer et exploiter les faiblesses connues du ou des protocoles individuels.

Lorsque nécessaire, et après l'établissement d'une session TLS entre les deux parties, le serveur d'accréditifs peut demander que le client fournisse ses informations d'identifiant et mot de passe d'utilisateur pour authentifier l'utilisateur distant. De préférence, client et serveur peuvent coopérer pour effectuer l'opération d'authentification qui permet au serveur d'authentifier le client (et peut-être vice-versa) dans un "processus à connaissance zéro". Dans ce cas, le client n'a pas besoin d'un accréditif de sécurité.

TLS avec authentification du client (cTLS)

TLS fournit une capacité de session sûre facultative (parfois appelée TLS côté client) par laquelle le serveur TLS peut demander l'authentification du client en vérifiant la signature numérique du client.

Afin d'utiliser cTLS pour fournir l'authentification mutuelle, le client doit aussi être configuré avec au moins un accréditif de sécurité acceptable au serveur TLS pour les besoin de l'authentification à distance du client.

4.2.3 Autres méthodes d'authentification

D'autres méthodes d'authentification qui fournissent les capacités de sécurité nécessaires PEUVENT aussi convenir pour l'utilisation avec les protocoles d'échange d'accréditif SACRED.

4.3 Suites de protocoles de transport

Il est prévu qu'une ou plusieurs piles de protocoles sous-jacentes puissent porter les protocoles d'échange d'accréditif SACRED. Il est reconnu pour commencer que l'utilisation de plusieurs suites de protocoles sous-jacentes, bien que non idéale du point de vue de l'interopérabilité, peut bien être requise pour prendre en charge la grande variété de besoins prévue.

Les membres de la liste de diffusion SACRED ont discuté de plusieurs suites de protocoles qui ont été examinées selon leurs mérites techniques, chacun avec des avantages et des coûts de conception/mise en œuvre de protocole distincts. Parmi ces protocoles, on a :
- TCP
- BEEP
- HTTP

Toutes les suites de protocoles mentionnées ici dépendent de TCP pour fournir un protocole de couche transport fiable, de bout en bout. Chacune de ces approches de construction donne une façon différente pour traiter les questions restantes de couche d'application (gestion de base de session, sécurité de niveau session, présentation/formatage, fonctionnalités d'application).

4.3.1 TCP

Cette approche (la mise en couche d'un protocole d'échange d'accréditif SACRED directement par dessus une connexion TCP) exige le développement d'un protocole commercial de messagerie d'échange d'accréditifs qui fasse l'interface d'une connexion/prise TCP. Le principal avantage de cette approche est la capacité de fournir exactement la fonction de protocole nécessaire et rien de plus. La plupart des environnements de développement de serveur et client fournissent déjà l'API de niveau prise nécessaire.

4.3.2 BEEP

Cette approche s'appuie sur le protocole extensible d'échanges de blocs (BEEP, *Blocks Extensible Exchange Protocol*)

décrit dans la [RFC3080]. BEEP fournit un échange de messages d'homologue à homologue d'utilisation générale sur tout mécanisme de transport où les transpositions nécessaire de couche transport ont été définies pour le fonctionnement sur TCP, TLS, etc. Voir aussi la [RFC3081].

BEEP fournit les capacités nécessaires d'authentification d'utilisateur/sécurité de session et gestion de session pour la prise en charge des opérations d'échange d'accréditif SACRED.

4.3.3 HTTP

Cette approche s'appuie sur le protocole de transport hypertexte (HTTP, *Hypertext Transport Protocol*) décrit dans les [RFC1945] et [RFC2616]. HTTP fournit une frappe d'utilité générale et une négociation de la représentation des données, permettant de construire les systèmes indépendamment des objets de données transférés. La prise en charge de HTTP est disponible sur une grande variété de plateformes de serveur et de clients, incluant les appareils portables qui s'appliquent aux environnements d'itinérance (ordinateurs portables, tablettes, téléphones mobiles, etc.).

HTTP est mis en couche sur TCP et peut facultativement être utilisé avec TLS pour fournir une sécurité par authentification au niveau session. L'une ou l'autre des options d'authentification de TLS, sTLS ou cTLS, peut être utilisée chaque fois que TLS est pris en charge.

5. Considérations sur la sécurité

Les considérations sur la sécurité qui suivent identifient les observations et précautions générales à considérer pour un cadre de prise en charge de la mobilité des accréditifs. Lors de la conception ou la mise en œuvre d'un protocole de prise en charge de ce cadre, on devrait suivre ces considérations sur la sécurité, et de plus consulter le document sur les exigences pour la disponibilité sécurisée des accréditifs SACRED [RFC3157].

5.1 Sécurité des communications

Une PDU SACRED va contenir des informations relevant de l'authentification du client ou serveur, ou la communication des accréditifs. Ces informations sont l'objet des soucis de sécurité traditionnels identifiés ci-dessous.

5.1.1 Confidentialité

Le mot de passe ou vérificateur de mot de passe devrait être protégé lors de la communication du client au serveur d'accréditifs. La valeur communiquée devrait être résistante à un attaque de dictionnaire.

De même la confidentialité des accréditifs d'entité doit être protégée lorsque ils sont communiqués du client au serveur et vice-versa. La valeur communiquée devrait aussi résister à une attaque de dictionnaire.

5.1.2 Intégrité

L'intégrité de la communication entre le client et le serveur d'accréditifs est exigée. De cette façon, les opérations prévues du client ne peuvent pas être altérées (par exemple, d'une mise à jour à une suppression d'accréditifs) ni de "vieux" accréditifs être par malveillance donnés au client (par exemple, éventuellement par un attaquant qui répète un téléchargement d'accréditifs antérieurs).

5.1.3 Authentification d'entité

Une authentification appropriée du client et du serveur est exigée pour réaliser la confidentialité et l'intégrité de la communication.

Le serveur doit authentifier correctement le client, afin que les accréditifs ne soient pas révélés par erreur à un attaquant. Le client doit s'assurer de l'identification appropriée du serveur d'accréditifs afin d'empêcher la révélation de leur mot de passe à un attaquant. Cet objectif peut être réalisé implicitement avec un protocole fondé sur un mot de passe fort ou explicitement. Si le serveur est identifié explicitement, l'utilisateur ou le client doit s'assurer que le mot de passe d'utilisateur est convoyé à un serveur de confiance. Ceci peut être réalisé en installant des clés de confiance appropriées chez le client.

5.1.4 Non répudiation

Il n'est pas exigé que le protocole SACRED prenne en charge par lui-même la non répudiation, bien que le contexte dans lequel les accréditifs sont utilisés puisse avoir de telles exigences.

5.2 Sécurité des systèmes

La sécurité des systèmes est concernée par la protection des points d'extrémité du protocole (c'est-à-dire, le client et le serveur) et des informations mémorisées dans le serveur pour la prise en charge du protocole SACRED.

5.2.1 Sécurité du client

Comme avec la plupart des protocoles de sécurité, une utilisation sûre du client s'appuie souvent, en partie, sur un comportement sûr de la part de l'utilisateur. Dans le cas d'un protocole SACRED fondé sur le mot de passe, les utilisateurs devraient être formés, ou guidés par une politique, à choisir des mots de passe avec une quantité raisonnable d'entropie. De plus, les utilisateurs devraient être informés de l'importance de la protection de la confidentialité de leur mot de passe de compte.

De plus, l'interface du client devrait être conçue pour déjouer la "lecture par dessus l'épaule" où un attaquant peut observer le mot de passe pendant qu'il est entré par l'usager. Cela se fait souvent en ne faisant pas écho des caractères exacts du mot de passe lors de sa frappe.

Aussi, l'interface devrait encourager la frappe du mot de passe dans le champ d'interface approprié afin que les protections puissent être correctement mises en application. Par exemple, un usager devrait être prévenu de ne pas entrer par erreur son mot de passe dans le champ "nom d'utilisateur" (car le mot de passe va probablement être reflété par l'écran dans ce cas, et ne pourra pas être chiffré lors de la communication au serveur). Cela peut se faire, par exemple, via l'insertion automatique du nom d'utilisateur ou de plusieurs choix de nom d'utilisateur dans le champ de dialogue approprié sur l'écran.

5.2.2 Sécurité du client, authentification du serveur TLS

Lorsque TLS est utilisé comme protocole de transport SACRED, l'interface de client devrait être conçue pour permettre à l'usager de vérifier qu'elle est connecté au serveur d'accréditifs prévu. Par exemple, le logiciel client devrait permettre l'affichage visuel des composants d'identification du certificat X.509 du serveur TLS, comme le nom du serveur, l'empreinte du certificat, etc.

Les utilisateurs devraient être guidés pour vérifier régulièrement ces informations, permettant une reconnaissance directe des serveurs d'accréditifs de confiance. De plus, les usagers devraient être informés de l'importance de la vérification de l'identité de leur serveur d'accréditifs avant d'initier toute opération d'échange d'accréditifs.

Un client SACRED DEVRAIT n'être configuré qu'avec les ancres de confiance SACRED qui sont à utiliser par le client. La réutilisation d'ancres de confiance provenant d'autres applications, par exemple, les navigateurs Internet, N'EST PAS RECOMMANDÉE.

5.2.3 Sécurité du serveur

Les vérificateurs de mots de passe et les accréditifs d'utilisateur doivent recevoir un haut niveau de protection au serveur d'accréditifs. En plus du salage et du super chiffrement de chacun (pour assurer la résistance aux attaques de dictionnaire hors ligne) un système devrait s'assurer que les clés du serveur d'accréditifs sont protégées en utilisant des contrôles d'accès de procédure et physiques suffisants.

La connexion au serveur d'accréditifs devrait être résistante aux attaques en répétition.

Les tentatives en ligne d'accès à un compte d'utilisateur particulier devraient être contrôlées, ou au moins surveillées. Le contrôle peut être mis en application en incorporant un délai après un certain nombre d'échecs de connexion sur un certain compte, ou éventuellement le verrouillage du compte. Autrement, on peut simplement enregistrer dans le journal des événements les tentatives non réussies lorsque une notice administrative est produite une fois qu'est atteint un seuil d'échec de tentatives d'accès aux accréditifs.

5.2.4 Déni de service

Comme avec la plupart des protocoles, les questions de déni de service (DoS) doivent aussi être prises en compte. Dans le cas de SACRED, la plupart des question de DoS sont un problème pour le protocole de transport sous jacent. Cependant,

certains problèmes peuvent quand même être atténués.

Le service à un usager peut être dénié dans le cas ou le compte est verrouillé après de nombreux échecs de tentative de connexion. La possibilité d'une protection contre les attaques en ligne doit donc être prise en considération (comme décrit ci-dessus). Une authentification d'utilisateur appropriée devrait s'assurer qu'un attaquant ne s'approprie pas, par malveillance, les accréditifs d'un utilisateur. Les serveurs d'accréditifs devraient être méfiants à l'égard de connexions répétées sur un compte particulier (ce qui identifie aussi une possible faille de la sécurité, comme décrit ci-dessus) ou des volumes anormaux de demandes sur un certain nombre de comptes (identifiant éventuellement une attaque de déni de service).

6. Références

6.1 Références normatives

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

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

6.2 Références pour information

[BM92] Bellovin, S. and M. Merritt, "Encrypted Key Exchange: Password-based protocols secure against dictionary attacks", Proceedings of the IEEE Symposium on Research in Security and Privacy, mai 1992.

[BM94] Bellovin, S. and M. Merritt, "Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks et Password File Compromise", ATT Labs Technical Report, 1994.

[PKCS12] "PKCS 12 v1.0: Personal Information Exchange Syntax", RSA Laboratories, 24 juin 1999.

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

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk, "Protocole de transfert Hypertext -- HTTP/1.0", mai 1996. *(Info.)*

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

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC2945] T. Wu, "Système SRP d'authentification et d'échange de clés", septembre 2000. *(P.S.)*

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

[RFC3081] M. Rose, "Transposition du cœur BEEP en TCP", mars 2001. *(P.S.)*

[SPEKE] Jablon, D., "Strong Password-Only Authenticated Key Exchange", septembre 1996.

7. Adresse des auteurs

Dale Gustafson
Future Foundation Inc.
mél : degustafson@comcast.net

Mike Just
Treasury Board of Canada, Secretariat
mél : Just.Mike@tbs-sct.gc.ca

Magnus Nystrom
RSA Security Inc.
mél : magnus@rsasecurity.com

8. Déclaration complète de droits de reproduction

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