Groupe de travail Réseau
Request for Comments : 5019
Catégorie : Sur la voie de la normalisation
Traduction Claude Brière de L'Isle

A. Deacon, VeriSign
R. Hurst, Microsoft
septembre 2007

Profil léger du protocole d'état de certificat en ligne (OCSP) pour environnements à gros volumes

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
La présente spécification définit un profil du protocole d'état de certificat en ligne (OCSP, *Online Certificate Status Protocol*) qui traite les problèmes d'adaptabilité inhérents lors de l'utilisation de OCSP dans des environnements d'infrastructure de clé publique (PKI, *Public Key Infrastructure*) à grande échelle (fort volume) et/ou des environnements de PKI qui exigent une solution légère pour minimiser la bande passante de communication et le traitement côté client.

Table des Matières

1. Introduction

Le protocole d'état de certificat en ligne [RFC2560] spécifie un mécanisme utilisé pour déterminer l'état des certificats numériques, au lieu d'utiliser des listes de révocation de certificat (CRL, *Certificate Revocation List*). Depuis sa définition en 1999, il a été déployé dans divers environnements et s'est révélé être un mécanisme utile de vérification de l'état des certificats. (Par souci de concision, on se réfère à OCSP comme étant utilisé pour vérifier l'état des certificats, mais seul l'état de révocation d'un certificat est vérifié via ce protocole.)

Aujourd'hui, de nombreux déploiements de OCSP ont été utilisés pour s'assurer d'informations en temps utile et sûres d'état de certificat pour des transactions d'informations électroniques de forte valeur ou d'informations très sensibles, comme dans les environnements de la banque et de la finance. À ce titre, l'exigence qu'un répondant OCSP réponde en "temps réel" (c'est-à-dire, en générant une nouvelle réponse OCSP pour chaque demande OCSP) a été importante. De plus, ces déploiements ont opéré dans des environnements où l'utilisation de la bande passante n'est pas un probème, et ont fonctionné sur des systèmes de client et serveur où la puissance de traitement n'est pas restreinte.

Comme l'utilisation de PKI continue de croître et d'entrer dans divers environnements, augmente aussi le besoin d'un mécanisme adaptable et de coût raisonnable de mécanisme d'état de certificat. Bien que OCSP tel que défini et déployé actuellement satisfasse les besoins des PKI de taille petite et moyenne qui fonctionnent sur les puissants systèmes des réseaux filaires, il y a des limites à l'adaptabilité de ces déploiements OCSP du point de vue à la fois de l'efficacité et du coût. Les environnements mobiles, où la bande passante du réseau peut être un souci majeur et où les appareils côté client subissent des contraintes de traitement, exigent une utilisation rationnelle de OCSP pour minimiser l'utilisation de la bande passante et la complexité du traitement côté client [OCSPMP].

PKI continue d'être déployé dans des environnements où des millions sinon des centaines de millions de certificats ont été produits. Dans nombre de ces environnements, un nombre encore plus grand d'utilisateurs (aussi appelés des consommateurs d'assertions *(relying parties)*) a besoin de s'assurer que le certificat sur lequel ils s'appuient n'a pas été révoqué. À ce titre, il est important que OCSP soit utilisé d'une façon telle qu'il assure que la charge qui pèse sur les répondants OCSP et l'infrastructure réseau requise pour héberger ces répondants reste minimale.

Le présent document traite des questions d'adaptabilité inhérentes à l'utilisation de OCSP dans les environnements de PKI décrits ci-dessus en définissant un profil de message et en précisant le comportement du client et du répondant OCSP qui va permettre :
1) la pré production et la distribution de la réponse OCSP ;
2) la réduction de la taille du message OCSP pour diminuer l'usage de bande passante ;
3) la mise en antémémoire du message de réponse dans le réseau et chez le client.

L'intention est que les exigences normatives définies dans ce profil soient adoptées par les clients et répondants OCSP opérant dans des environnements de PKI à très grande échelle (forts volumes) ou des environnments de PKI qui exigent une solution légère pour minimiser la bande passante et la puissance de traitement côté client (ou les deux) comme décrit ci-dessus. Comme OCSP n'a pas de moyen pour signaler les capacités du répondant dans le protocole, les clients qui ont besoin de différencier les réponses OCSP produites par les répondants qui se conforment à ce profil et celles de ceux qui ne le font pas, ont besoin de s'appuyer sur des mécanismes hors bande pour déterminer quand un répondant opère en accord avec le présent profil et, à ce titre, quand les exigences de ce profil s'appliquent. Dans le cas où des mécanismes hors bande ne peuvent pas être disponibles, ce profil assure que l'interopérabilité va quand même avoir lieu entre un client OCSP qui se conforme à la RFC2560 et un répondant qui opère dans un mode décrit dans cette spécification.

1.1 Terminologie des exigences

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].

2. Profil de message OCSP

Cette Section définit un sous ensemble de la fonctionnalité OCSPRequest et OCSPResponse définie dans la [RFC2560].

2.1 Profil de demande OCSP

2.1.1 Structure OCSPRequest

Les demandes OCSP qui se conforment à ce profil DOIVENT inclure seulement une demande dans la structure OCSPRequest.RequestList.

Les clients DOIVENT utiliser SHA1 comme algorithme de hachage pour les valeurs de CertID.issuerNameHash et de CertID.issuerKeyHash.

Les clients NE DOIVENT PAS inclure la structure singleRequestExtensions.

Les clients NE DEVRAIENT PAS inclure la structure requestExtensions. Si une structure requestExtensions est incluse, ce profil RECOMMANDE qu'il contienne seulement l'extension Nom occasionnel (id-pkix-ocsp-nonce). Voir à la Section 4 les problèmes concernant l'utilisation d'un nom occasionnel dans les environnements OCSP à fort volume.

2.1.2 OCSPRequests signées

Les clients NE DEVRAIENT PAS envoyer de demandes OCSP signées. Les répondants PEUVENT ignorer la signature sur les demandes OCSP.

Si la demande OCSP est signée, le client DEVRA spécifier son nom dans le champ OCSPRequest.requestorName ; autrement, les clients NE DEVRAIENT PAS inclure de champ requestorName dans la demande OCSP.Les serveurs OCSP DOIVENT être prêts à recevoir des demandes OCSP non signées qui contiennent le champ requestorName, mais doivent réaliser que la valeur fournie n'est pas authentifiée.

2.2 Profil de réponse OCSP

2.2.1 Structure OCSPResponse

Les répondants DOIVENT générer une BasicOCSPResponse comme identifiée par l'OID id-pkix-ocsp-basic. Les clients DOIVENT être capables d'analyser et accepter une BasicOCSPResponse. Les réponses OCSP qui se conforment à ce profil DEVRAIENT inclure seulement une réponse dans la structure ResponseData.responses, mais PEUVENT inclure des éléments SingleResponse supplémentaires si nécessaire pour améliorer les performances de pré-génération de réponse ou l'efficacité de la mise en antémémoire.

Le répondant NE DEVRAIT PAS inclure de responseExtensions. Comme spécifié dans la [RFC2560], les clients DOIVENT ignorer les extensiosn de réponse non reconnues non critiques dans la réponse.

Dans le cas où un répondant n'a pas la capacité de répondre à une demande OCSP contenant une option non prise en charge par le serveur, il DEVRAIT retourner la réponse la plus complète qu'il peut. Par exemple, dans le cas où un répondant prend seulement en charge des réponses pré-produites et n'a pas la capacité de répondre à une demande OCSP contenant un nom occasionnel, il DEVRAIT retourner une réponse qui n'inclut pas de nom occasionnel.

Les clients DEVRAIT tenter de traiter une réponse même si la réponse n'inclut pas de nom occasionnel. Voir à la Section 4 les détails de la validation des réponses qui ne contiennent pas de nom occasionnel. Voir aussi à la Section 7 les considérations de sécurité pertinentes.

Les répondants qui n'ont pas la capacité de répondre aux demandes OCSP qui contiennent une option non prise en charge comme un nom occasionnel PEUVENT transmettre la demande à un répondant OCSP capable de le faire.

Le répondant PEUT inclure des structures d'extension singleResponse.singleResponse.

2.2.2 OCSPResponses signées

Les clients DOIVENT valider la signature sur la réponse OCSP retournée.

Si la réponse est signée par un délégué de l'autorité de certification (CA, *certification authority*) productrice, un certificat valide du répondant DOIT être référencé dans la structure BasicOCSPResponse.certs.

Il est RECOMMANDÉ que le certificat du répondant OCSP contienne l'extension id-pkix-ocsp-nocheck, comme défini dans la [RFC2560], pour indiquer au client qu'il 'a pas besoin de vérifier l'état du certificat. De plus, il est RECOMMANDÉ que ni une extension OCSP authorityInfoAccess (AIA) ni une extension cRLDistributionPoints (CRLDP) ne soit incluse dans le certificat du répondant OCSP. En conséquence, le certificat signé du répondant DEVRAIT être d'une durée de vie relativement courte et renouvelé régulièrement.

Les clients DOIVENT être capables d'identifier les certificats de répondant OCSP en utilisant les choix byName et byKey de ResponseData.ResponderID. Les répondants DEVRAIENT utiliser byKey pour réduire la taille de la réponse dans les scénarios où la réduction de la bande passante est un problème.

2.2.3 Valeurs de OCSPResponseStatus

Tant que l'infrastructure OCSP a des enregistrements d'autorité pour un certificat particulier, un OCSPResponseStatus de "réussite" va être retourné. Quand l'accès aux enregistrements d'autorité pour un certificat particulier n'est pas disponible, le répondant DOIT retourner un OCSPResponseStatus de "non autorisé". À ce titre, le présent profil étend la définition de la [RFC2560] de "non autorisé" comme suit : la réponse "non autorisé" est retournée dans les cas où le client n'est pas autorisé à faire cette interrogation à ce serveur ou le serveur n'est pas capable de faire une réponse d'autorité.

Par exemple, les répondants OCSP qui n'ont pas accès aux enregistrements d'autorité pour un certificat demandé, comme ceux qui génèrent et distribuent en avance les réponses OCSP et donc n'ont pas la capacité de répondre correctement avec une réponse signée "réussite" mais "inconnu", vont répondre avec un OCSPResponseStatus de "non autorisé". Aussi, afin de s'assurer que la base de données des informations de révocation ne croît pas sans limite au fil du temps, le répondant PEUT supprimer les enregistrements d'état des certificats arrivés à expiration. Les demandes des clients pour des certificats dont l'enregistrement a été supprimé vont résulter en un OCSPResponseStatus de "non autorisé".

Les considérations de sécurité concernant l'utilisation de réponses non signées sont discutées dans la [RFC2560].

2.2.4 thisUpdate, nextUpdate, et producedAt

Quand il pré-produit des messages OCSPResponse, le répondant DOIT régler les instants thisUpdate, nextUpdate, et producedAt comme suit :

thisUpdate : instant auquel l'état indiqué est connu pour être correct.

nextUpdate : instant auquel ou avant lequel des informations plus récentes vont être disponibles sur l'état du certificat. Les répondants DOIVENT toujours inclure cette valeur pour aider à la mise en antémémoire des réponses. Voir à la Section 6 des informations supplémentaires sur la mise en antémémoire.

producedAt : instant où la réponse OCSP a été signée.

Note : dans de nombreux cas, la valeur de thisUpdate et de producedAt va être la même.

Pour les besoins de ce profil, des valeurs de GeneralizedTime codées en ASN.1 comme thisUpdate, nextUpdate, et producedAt DOIVENT être exprimées en temps moyen de Greenwich (GMT, *Greenwich Mean Time*) (dit aussi Zoulou) et DOIVENT inclure les secondes (c'est-à-dire, les temps sont dans le format AAAAMMJJHHMMSSZ) même lorsque le nombre de secondes est zéro. Les valeurs de GeneralizedTime NE DOIVENT PAS inclure de fraction de seconde.

3. Comportement du client

3.1 Découverte de répondant OCSP

Les clients DOIVENT prendre en charge l'extension authorityInfoAccess définie dans la [RFC3280] et DOIVENT reconnaître la méthode d'accès id-ad-ocsp. Cela permet aux CA d'informer les clients de la façon dont ils peuvent contacter le service OCSP.

Dans le cas où un client veut vérifier l'état d'un certificat qui contient à la fois une extension authorityInformationAccess (AIA) pointant sur un répondant OCSP et une extension cRLDistributionPoints pointant sur une CRL, le client DEVRAIT tenter de contacter d'abord le répondant OCSP. Les clients PEUVENT tenter de restituer la CRL si aucune réponse OCSP

n'est reçue du répondant après l'expiration d'un temporisatteur et un nombre d'essais configurés en local.

3.2 Envoi d'une demande OCSP

Pour éviter du trafic réseau inutile, les applications DOIVENT vérifier la signature des données signées avant de demander à un client OCSP de vérifier l'état des certificats utilisés pour vérifier les données. Si la signature est invalide ou si l'application n'est pas capable de la vérifier, une vérification OCSP NE DOIT PAS être demandée.

De même, une application DOIT valider la signature sur les certificats dans une chaîne, avant de demander à un client OCSP de vérifier l'état du certificat. Si la signature du certificat est invalide ou si l'application n'est pas capable de la vérifier, une vérification OCSP NE DOIT PAS être demandée. Les clients NE DEVRAIENT PAS faire une demande de vérification de l'état de certificats expirés.

4. S'assurer qu'une réponse OCSP est fraîche

Afin de s'assurer qu'un client n'accepte pas une réponse périmée qui indique un "bon" état alors qu'en fait il y a une réponse plus à jour qui spécifie le statut de "révoqué", un client doit s'assurer que les réponses qu'il reçoit sont fraîches.

En général, deux mécanismes sont disponibles aux clients pour s'assurer qu'une réponse est fraîche. La première utilise des noms occasionnels, et la seconde se fonde sur l'heure. Pour que les mécanismes fondés sur l'heure fonctionnent, les clients et les répondants DOIVENT avoir accès à une source horaire précise.

Parce que le présent profil spécifie que les clients NE DEVRAIENT PAS inclure de structure requestExtensions dans les demandes OCSP (voir au paragraphe 2.1) les clients DOIVENT être capables de déterminer la fraîcheur de la réponse OCSP sur la base d'une source horaire précise. Les clients qui optent pour l'inclusion d'un nom occasionnel dans la demande NE DEVRAIENT PAS rejeter une réponse OCSP correspondante sur la seule base de la non existence du nom occasionnel attendu, mais DOIVENT revenir à la validation de la réponse OCSP fondée sur l'heure.

Les clients qui n'incluent pas de nom occasionnel dans la demande DOIVENT ignorer tout nom occasionnel qui pourrait être présent dans la réponse.

Les clients DOIVENT vérifier l'existence du champ nextUpdate et DOIVENT s'assurer que l'heure actuelle, exprimée en temps GMT comme décrit au paragraphe 2.2.4, tombe entre les instants thisUpdate et nextUpdate. Si le champ nextUpdate est absent, le client DOIT rejeter la réponse.

Si le champ nextUpdate est présent, le client DOIT s'assurer qu'il n'est pas plus tôt que l'heure actuelle. Si l'heure actuelle chez le client est plus tard que l'heure spécifiée dans le champ nextUpdate, le client DOIT rejeter la réponse comme périmée. Les clients PEUVENT permettre la configuration d'une petite période de tolérance pour l'acceptation des réponses après nextUpdate pour tenir compte de différences d'horloge mineures relatives aux répondants et aux antémémoires. Cette période de tolérance devrait être choisie en fonction de la précision de la technologie de synchronisation disponible dans l'environnement de l'application appelante. Par exemple, les homologues Internet avec des connexions à faible latence vont normalement s'attendre à ce que la synchronisation de NTP les gardent dans une précision d'une fraction de seconde ; des environnements à plus forte latence ou lorsque un outil analogue à NTP n'est pas disponible peuvent devoir être plus tolérants.

Voir les considérations sur la sécurité à la Section 7 pour des détails supplémentaires sur les attaques en répétition et par interposition.

5. Profil de transport

Le répondant OCSP DOIT prendre en charge les demandes et les réponses sur HTTP. Quand ils envoient des demandes qui sont inférieures ou égales à un total de 255 octets (après codage) incluant le schéma et les délimiteurs (http://) le nom de serveur et la structure OCSPRequest codée en base64, les clients DOIVENT utiliser la méthode GET (pour permettre la mise en antémémoire de la réponse). Les demandes OCSP de plus de 255 octets DEVRAIENT être soumises en utilisant la méthode POST. Dans tous les cas, les clients DOIVENT suivre les descriptions de l'annexe A.1.1 de la [RFC2560] lors de la construction de ces messages.

Quand ils construisent un message GET, les clients OCSP DOIVENT coder en base64 la structure OCSPRequest et l'ajouter à l'URI spécifié dans l'extension AIA [RFC3280]. Les clients NE DOIVENT PAS inclure de caractères CR ou LF dans la chaîne codée en base64. Les clients DOIVENT coder comme un URL approprié la demande OCSP codée en base64. Par exemple :

http://ocsp.example.com/MEowSDBGMEQwQjAKBggqhkiG9w0CBQQQ7sp6GTKpL2dAdeGaW267owQQqInESWQD0mGeBArSgv%2FBWQIQLJx%2Fg9xF8oySYzol80Mbpg%3D%3D

En réponse aux demandes OCSP correctement formatées qui peuvent être mises en antémémoire (c'est-à-dire, les réponses qui contiennent une valeur nextUpdate) le répondant va inclure la valeur binaire du codage DER de la réponse OCSP précédée des en-têtes HTTP [RFC2616] suivants :

content-type : application/ocsp-response
content-length : <longueur de la réponse OCSP>
last-modified : <date producedAt [RFC2616]>
ETag : "<un valideur fort>"
expires : <date nextUpdate [RFC2616]>
cache-control : max-age=<n>, public, no-transform, must-revalidate
date : <date actuelle [RFC2616]>

Voir au paragraphe 6.2 les détails de l'utilisation de ces en-têtes.

6. Recommandations pour la mise en antémémoire

La capacité de mettre en antémémoire les réponses OCSP partout dans le réseau est un facteur important dans les déploiements OCSP à fort volume. Cette section discute du comportement recommandé de mise en antémémoire des clients OCSP et des mandataires HTTP et des étapes qui devraient être suivies pour minimiser le nombre de fois où les clients OCSP "vont sur le réseau". De plus, le concept d'inclure les réponses OCSP dans les échanges de protocole (autrement dit l'agrafage ou le portage) comme défini dans TLS, est aussi exposé.

6.1 Mise en antémémoire chez le client

Pour minimiser l'usage de la bande passante, les clients DOIVENT mettre en antémémoire locale les réponses OCSP d'autorité (c'est-à-dire, une réponse avec une signature qui a été validée avec succès et qui indique un OCSPResponseStatus de "réussite").

La plupart des clients OCSP vont envoyer des demandes OCSP à ou près du moment nextUpdate (quand une réponse en antémémoire arrive à expiration). Pour éviter de grosses pointes de la charge de celui qui répond qui pourraient survenir quand de nombreux clients rafraîchissent les réponses mises en antémémoire pour un certificat recherché, les répondants PEUVENT indiquer quand le client devrait aller chercher une réponse OCSP mise à jour en utilisant la directive "cache-control:max-age". Les clients DEVRAIENT aller chercher la réponse OCSP mise à jour à ou après le moment max-age. Pour s'assurer que les clients reçoivent une réponse OCSP mise à jour, les répondants OCSP DOIVENT rafraîchir la réponse OCSP avant le moment max-age.

6.2 Mandataires HTTP

Le répondant DEVRAIT régler les en-têtes HTTP de la réponse OCSP d'une manière telle qu'elle permette une utilisation intelligente des serveurs mandataires HTTP intermédiaires. Voir dans la [RFC2616] la définition complète de ces en-têtes et le format approprié de toutes les valeurs de date et d'heure.

En-tête HTTP	Description
date	date et heure à laquelle le serveur OCSP a généré la réponse HTTP.
last-modified	cette valeur spécifie la date et l'heure à laquelle le répondant OCSP a fait la dernière modification de la réponse. Cette date et heure va être la même que l'horodatage thisUpdate dans la demande elle-même.
expires	spécifie pendant combien de temps la réponse est considérée fraîche. Cette date et heure va être la même que l'horodatage nextUpdate dans la réponse OCSP elle-même.
ETag	chaîne qui identifie une version particulière des données associées. Ce profil RECOMMANDE que la valeur de ETag soit la représentation ASCII HEX du hachage SHA1 de la structure OCSPResponse.

cache-control contient des directives de mise en antémémoire.
* max-age=<n> n est une valeur d'heure plus tard que thisUpdate mais plus tôt que nextUpdate.
* public rend normalement non mettable en antémémoire une réponse qui peut être mise en antémémoire par des antémémoires partagées et non partagées.
* no-transform spécifie qu'une antémémoire de mandataire ne peut pas changer le type, la longueur, ou le codage du contenu de l'objet.
* must-revalidate -empêche les antémémoires de retourner intentionnellement des réponses périmées.

Les répondants OCSP NE DOIVENT PAS inclure d'en-tête "Pragma: no-cache", "Cache-Control: no-cache", ou "Cache-Control: no-store" dans des réponses OCSP d'autorité.

Les répondants OCSP DEVRAIENT inclure un ou plusieurs de ces en-têtes dans les réponses OCSP non d'autorité.

Par exemple, si on suppose qu'une réponse OCSP a les valeurs d'horodatage suivantes :
thisUpdate = May 1, 2005 01:00:00 GMT
nextUpdate = May 3, 2005 01:00:00 GMT
productedAt = May 1, 2005 01:00:00 GMT
et qu'un client OCSP demande la réponse du 2 mai 2005 à 01:00:00 GMT. Dans ce scénario, la réponse HTTP peut ressembler à :

```
content-type: application/ocsp-response
content-length: 1000
date: Fri, 02 May 2005 01:00:00 GMT
last-modified: Thu, 01 May 2005 01:00:00 GMT
ETag: "c66c0341abd7b9346321d5470fd0ec7cc4dae713"
expires: Sat, 03 May 2005 01:00:00 GMT
cache-control: max-age=86000,public,no-transform,must-revalidate
<...>
```

Les clients OCSP NE DOIVENT PAS inclure un en-tête no-cache dans les messages de demande OCSP, sauf si le client rencontre une réponse expirée qui peut être le résultat d'un mandataire intermédiaire qui a mis en antémémoire des données périmées. Dans cette situation, les clients DEVRAIENT renvoyer la demande en spécifiant que les mandataires devraient être outrepassés en incluant un en-tête HTTP approprié dans la demande (c'est-à-dire, Pragma: no-cache ou Cache-Control: no-cache).

6.3 Mise en antémémoire aux serveurs

Dans certains scénarios, il est avantageux d'inclure des informations de réponse OCSP dans le protocole utilisé entre le client et le serveur. Inclure des réponses OCSP de cette manière a quelques effets intéressants.

D'abord, cela permet la mise en antémémoire des réponses OCSP sur le serveur, diminuant donc le nombre de prises du répondant OCSP.

Ensuite, cela permet la validation des certificats dans le cas où le client n'est pas connecté à un réseau et donc élimine le besoin pour les clients d'établir une nouvelle session HTTP avec le répondant.

Troisièmement, cela réduit le nombre d'allers-retours que doit faire le client pour terminer une prise de contact.

Enfin, cela simplifie la mise en œuvre de OCSP côté client en permettant une situation où le client a seulement besoin de la capacité d'analyser et reconnaître les réponses OCSP.

Cette fonctionnalité a été spécifiée comme une extension au protocole TLS [RFC4346] au paragraphe 3.6 de la [RFC4366], mais peut être appliquée à tout protocole client-serveur.

Ce profil RECOMMANDE que les clients et serveurs TLS mettent tous deux en œuvre le mécanisme d'extension de demande d'état de certificat pour TLS.

Plus d'informations concernant les problèmes de mise en antémémoire peuvent être obtenues dans la [RFC3143].

7. Considérations sur la sécurité

Les considérations suivantes s'appliquent en plus des considérations sur la sécurité de la Section 5 de la [RFC2560].

7.1 Attaques en répétition

Parce que l'utilisation de noms occasionnels dans ce profil est facultative, il y a une possibilité qu'une réponse OCSP périmée puisse être répétée, causant donc l'acceptation d'un réponse comme bonne alors qu'en fait il y a une réponse plus à jour qui spécifie son état comme révoqué. Afin d'atténuer cette attaque, les clients DOIVENT avoir accès à une source d'heure précise et s'assurer que les réponses OCSP qu'ils reçoivent sont suffisamment fraîches.

Les clients qui n'ont pas une source précise de date et heure sont vulnérables à l'interruption de service. Par exemple, un client avec une horloge suffisamment rapide peut rejeter une réponse OCSP fraîche. De même, un client avec une horloge suffisamment lente peut incorrectement accepter des réponses valides expirées pour des certificats qui peuvent en fait être révoqués.

De futures versions du protocole OCSP pourraient fournir un moyen pour que le client sache si le serveur prend en charge ou non les noms occasionnels. Si un client peut determiner que le serveur prend en charge les noms occasionnels, il DOIT rejeter une réponse qui ne contient pas un nom occasionnel attendu. Autrement, les clients qui optent pour l'inclusion de nom occasionnel dans la demande NE DEVRAIENT PAS rejeter une réponse OCSP correspondante seulement sur la base de la non existence du nom occasionnel attendu, mais DOIVENT revenir à la validation de la réponse OCSP sur la base de l'heure.

7.2 Attaques par interposition

Pour atténuer les risques associés à cette classe d'attaques, le client doit valider correctement la signature de la réponse.

L'utilisation de réponses signées dans OCSP sert à authentifier l'identité du répondant OCSP et à vérifier qu'il est autorisé à signer les réponses au nom de la CA.

Les clients DOIVENT s'assurer qu'ils sont en communication avec un répondant autorisé selon les règles décrites au paragraphe 4.2.2.2 de la [RFC2560].

7.3 Attaques en usurpation d'identité

L'utilisation de réponses signées dans OCSP sert à authentifier l'identité du répondant OCSP.

Comme précisé dans la [RFC2560], les clients doivent valider correctement la signature de la réponse OCSP et la signature sur le certificat du signataire de la réponse OCSP pour s'assurer qu'un répondant autorisé l'a créée.

7.4 Attaques de déni de service

Les répondants OCSP devraient prendre des mesures pour empêcher ou atténuer les attaques de déni de service. Comme ce profil spécifie l'utilisation de demandes OCSP non signées, l'accès au répondant peut être implicitement donné par quiconque peut envoyer une demande à un répondant, et donc la capacité de monter une attaque de déni de service via une inondation de demande peut être augmentée. Par exemple, un répondant pourrait limiter le taux de demandes entrantes provenant d'une adresse IP particulière si un comportement discutable est détecté.

7.5 Modification des en-têtes HTTP

Les valeurs incluses dans les en-têtes HTTP, comme décrit aux Sections 5 et 6, ne sont pas protégées cryptographiquement ; elles peuvent être manipulées par un attaquant. Les clients ne DEVRAIENT utiliser ces valeurs que comme indication pour la mise en antémémoire et ne DEVRAIENT en fin de compte s'appuyer que sur les valeurs présentes dans la réponse OCSP signée. Les clients NE DEVRAIENT PAS s'appuyer sur des réponses mises en antémémoire au delà de nextUpdate.

7.6 Authentification et autorisation de demande

L'utilisation suggérée de demandes non signées dans cet environnement supprime une option qui permet au répondant de déterminer l'authenticité de la demande entrante. Donc, l'accès au répondant peut être donné implicitement à quiconque peut envoyer une demande à un répondant. Les environnements où une autorisation explicite d'accès au répondant OCSP est nécessaire peuvent utiliser d'autres mécanismes pour authentifier les demandeurs ou restreindre ou mesurer le service.

8. Remerciements

Les auteurs tiennent à remercier Magnus Nystrom de RSA Security, Inc., Jagjeet Sondh de Vodafone Group R&D, et David Engberg de CoreStreet, Ltd. de leurs contributions à la présente spécification.

9. Références

9.1 Références normatives

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

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin et C. Adams, "Protocole d'état de certificat en ligne d'infrastructure de clé publique X.509 pour l'Internet - OCSP", juin 1999. *(P.S.) (Remplacée par* RFC6960)

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

[RFC3280] R. Housley, W. Polk, W. Ford et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002. *(Obsolète, voir* RFC5280)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4366] S. Blake-Wilson et autres, "Extensions de sécurité de la couche Transport (TLS)", avril 2006. (*Obsolète,* RFC5246) *(P.S.)*

9.2 Références pour information

[OCSPMP] Open Mobile Alliance, "OCSP Mobile Profile V1.0", www.openmobilealliance.org.

[RFC3143] I. Cooper, J. Dilley, "Problèmes connus de mandataire/antémémoire dans HTTP", juin 2001. *(Information)*

Appendice A. Exemple de messages OCSP

A.1 Demande OCSP

```
SEQUENCE {
  SEQUENCE {
    SEQUENCE {
      SEQUENCE {
        SEQUENCE {
          SEQUENCE {
            SEQUENCE {
              OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
              NULL
              }
            OCTET STRING
              C0 FE 02 78 FC 99 18 88 91 B3 F2 12 E9 C7 E1 B2
              1A B7 BF C0
```

```
        OCTET STRING
         0D FC 1D F0 A9 E0 F0 1C E7 F2 B2 13 17 7E 6F 8D
         15 7C D4 F6
        INTEGER
         09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
       }
      }
     }
    }
   }
```

A.2 Réponse OCSP

```
SEQUENCE {
  ENUMERATED 0
  [0] {
   SEQUENCE {
    OBJECT IDENTIFIER ocspBasic (1 3 6 1 5 5 7 48 1 1)
    OCTET STRING, encapsulates {
     SEQUENCE {
      SEQUENCE {
       [0] {
        INTEGER 0
        }
       [1] {
        SEQUENCE {
         SET {
          SEQUENCE {
           OBJECT IDENTIFIER organizationName (2 5 4 10)
           PrintableString 'Example Trust Network'
           }
          }

         SET {
          SEQUENCE {
           OBJECT IDENTIFIER
            organizationalUnitName (2 5 4 11)
           PrintableString 'Example, Inc.'
           }
          }
         SET {
          SEQUENCE {
           OBJECT IDENTIFIER
            organizationalUnitName (2 5 4 11)
           PrintableString
          'Example OCSP Responder'
           }
          }
         }
        }
       GeneralizedTime 07/11/2005 23:52:44 GMT
       SEQUENCE {
        SEQUENCE {
         SEQUENCE {
          SEQUENCE {
           SEQUENCE {
            OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
            NULL
            }
           OCTET STRING
           C0 FE 02 78 FC 99 18 88 91 B3 F2 12 E9 C7 E1 B2
```

```
      1A B7 BF C0
      OCTET STRING
      0D FC 1D F0 A9 E0 F0 1C E7 F2 B2 13 17 7E 6F 8D
      15 7C D4 F6
      INTEGER
      09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
      }
     [0]
      Erreur : L'objet a une longueur de zéro.
      GeneralizedTime 07/11/2005 23:52:44 GMT
     [0] {
      GeneralizedTime 14/11/2005 23:52:44 GMT
      }
     }
    }
   }
  SEQUENCE {
   OBJECT IDENTIFIER
    sha1withRSAEncryption (1 2 840 113549 1 1 5)
   NULL
   }

  BIT STRING
   0E 9F F0 52 B1 A7 42 B8 6E C1 35 E1 0E D5 A9 E2
   F5 C5 3C 16 B1 A3 A7 A2 03 8A 2B 4D 2C F1 B4 98
   8E 19 DB BA 1E 1E 72 FF 32 F4 44 E0 B2 77 1C D7
   3C 9E 78 F3 D1 82 68 86 63 12 7F A4 6F F0 4D 84
   EA F8 E2 F7 5D E3 48 44 57 28 80 C7 57 3C FE E1
   42 0E 5E 17 FC 60 D8 05 D9 EF E2 53 E7 AB 7F 3A
   A8 84 AA 5E 46 5B E7 B8 1F C6 B1 35 AD FF D1 CC
   BA 58 7D E8 29 60 79 F7 41 02 EA E0 82 0E A6 30
  [0] {
   SEQUENCE {
    SEQUENCE {
     SEQUENCE {
      [0] {
       INTEGER 2
       }
      INTEGER
      49 4A 02 37 1B 1E 70 67 41 6C 9F 06 2F D8 FE DA
      SEQUENCE {
       OBJECT IDENTIFIER
        sha1withRSAEncryption (1 2 840 113549 1 1 5)
       NULL
       }
      SEQUENCE {
       SET {
        SEQUENCE {
         OBJECT IDENTIFIER
          organizationName (2 5 4 10)
         PrintableString 'Example Trust Network'
         }
        }
       SET {
        SEQUENCE {
         OBJECT IDENTIFIER
          organizationalUnitName (2 5 4 11)
         PrintableString 'Example, Inc.'
         }
        }
       SET {
```

```
    SEQUENCE {
     OBJECT IDENTIFIER
      organizationalUnitName (2 5 4 11)
     PrintableString
       'Example CA'
     }
    }
   }
  SEQUENCE {
   UTCTime 08/10/2005 00:00:00 GMT
   UTCTime 06/01/2006 23:59:59 GMT
   }
  SEQUENCE {
   SET {
    SEQUENCE {
     OBJECT IDENTIFIER
      organizationName (2 5 4 10)
     PrintableString 'Example Trust Network'
     }
    }
   SET {
    SEQUENCE {
     OBJECT IDENTIFIER
      organizationalUnitName (2 5 4 11)
     PrintableString 'Example, Inc.'
     }
    }
   SET {
    SEQUENCE {
     OBJECT IDENTIFIER
      organizationalUnitName (2 5 4 11)
     PrintableString
      'Example OCSP Responder'
     }
    }
   }
  SEQUENCE {
   SEQUENCE {
    OBJECT IDENTIFIER
     rsaEncryption (1 2 840 113549 1 1 1)
    NULL
    }
   BIT STRING, encapsulates {
    SEQUENCE {
     INTEGER
 00 AF C9 7A F5 09 CA D1 08 8C 82 6D AC D9 63 4D
 D2 64 17 79 CB 1E 1C 1C 0C 6E 28 56 B5 16 4A 4A
 00 1A C1 B0 74 D7 B4 55 9D 2A 99 1F 0E 4A E3 5F
 81 AF 8D 07 23 C3 30 28 61 3F B0 C8 1D 4E A8 9C
 A6 32 B4 D2 63 EC F7 C1 55 7A 73 2A 51 99 00 D5
 0F B2 4E 11 5B 83 55 83 4C 0E DD 12 0C BD 7E 41
 04 3F 5F D9 2A 65 88 3C 2A BA 20 76 1D 1F 59 3E
 D1 85 F7 4B E2 81 50 9C 78 96 1B 37 73 12 1A D2
       [ Les autres octets de 1 sont sautés ]
     INTEGER 65537
     }
    }
   }
  [3] {
   SEQUENCE {
    SEQUENCE {
```

```
            OBJECT IDENTIFIER
              basicConstraints (2 5 29 19)
            OCTET STRING, encapsulates {
              SEQUENCE {}
              }
            }
           SEQUENCE {
            OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
            OCTET STRING, encapsulates {
              SEQUENCE {
                OBJECT IDENTIFIER
                  ocspSigning (1 3 6 1 5 5 7 3 9)
                }
              }
            }
           SEQUENCE {
            OBJECT IDENTIFIER keyUsage (2 5 29 15)
            OCTET STRING, encapsulates {
              BIT STRING 7 unused bits
                '1'B (bit 0)
              }
            }
           SEQUENCE {
            OBJECT IDENTIFIER
              ocspNoCheck (1 3 6 1 5 5 7 48 1 5)
            OCTET STRING, encapsulates {
              NULL
              }
            }
          }
         }
        }
       SEQUENCE {
        OBJECT IDENTIFIER
          sha1withRSAEncryption (1 2 840 113549 1 1 5)
        NULL
        }
       BIT STRING
        3A 68 5F 6A F8 87 36 4A E2 22 46 5C C8 F5 0E CE
        1A FA F2 25 E1 51 AB 37 BE D4 10 C8 15 93 39 73
        C8 59 0F F0 39 67 29 C2 60 20 F7 3F FE A0 37 AB
        80 0B F9 3D 38 D4 48 67 E4 FA FD 4E 12 BF 55 29
        14 E9 CC CB DD 13 82 E9 C4 4D D3 85 33 C1 35 E5
        8F 38 01 A7 F7 FD EB CD DE F2 F7 85 86 AE E3 1B
        9C FD 1D 07 E5 28 F2 A0 5E AC BF 9E 0B 34 A1 B4
        3A A9 0E C5 8A 34 3F 65 D3 10 63 A4 5E 21 71 5A
      }
     }
    }
   }
  }
 }
}
```

Adresse des auteurs

Alex Deacon
VeriSign, Inc.
487 E. Middlefield Road
Mountain View, CA 94043
USA
téléphone : 1-650-426-3478
mél : alex@verisign.com

Ryan Hurst
Microsoft
One Microsoft Way
Redmond, WA 98052
USA
téléphone : 1-425-707-8979
mél : rmh@microsoft.com

Déclaration complète de droits de reproduction