rfc5019

Profile OCSP pour les environnements à haut volume

OCSP spécifie un mécanisme utilisé pour déterminer le statut des certificats numériques, au lieu d'utiliser les CRL. Depuis sa définition en 1999, il a été déployé dans de nombreux environnements et prouvé son utilité.

À ce jour, de nombreux déploiements on été utilisé pour s'assurer les informations de statut de certificats de manière sécurisé et en temps pour les transactions électroniques ou hautement sensibles, tels que les environnements financiers. Un tel environnement exige un répondeur OCSP en temps réel. De plus, ces déploiements ont opérés dans des environnements où l'utilisation de la bande passante n'est pas un problème, et fonctionnement sur des systèmes client et serveur où la charge de traitement n'est pas une contrainte.

Vu que l'utilisation d'une PKI continue de grandir et d'aller dans divers environnements, on n'a donc besoin d'un mécanisme de statut de certificat efficace. Bien qu'OCSP a été définis et déployé dans de nombreuses petites et moyennes PKI qui opèrent sur des systèmes puissants et sur des réseaux câblés, il y a une limite sur l'agrandissement des déploiements. Les environnements mobiles, où la bande passante peut être une des principales contraintes, il exige une attention particulière de l'utilisation d'OCSP pour minimiser l'utilisation de la bande passante et la complexité de traitement côté client.

Les PKI continuent d'être déployés dans des environnements où des millions voir des centaines de millions de certificats ont été émis. Dans beaucoup de ces environnements, un aussi grand nombre d'utilisateurs on besoins de s'assurer que le certificat qu'ils souhaitent utiliser n'a pas été révoqué. Il est important que OCSP soit utilisé de manière à s'assurer que la charge des répondeurs OCSP et que l'infrastructure réseaux soient utilisés au minimum.

Ce document adresse les problèmes d'évolution inhérents en utilisant OCSP dans des environnements PKI décrits ci-dessus en définissant un profile de message et en clarifiant le comportement du client et du répondeur OCSP qui va permettre de :

1) pré-produire et distribuer des réponse OCSP

2) Réduire la taille de message OCSP pour réduire la bande passante

3) Le caching des messages de réponse dans le réseaux et dans le client

Il est prévu que les exigences définies dans ce profile soit adopté par les clients et les répondeurs OCSP opérant dans des environnements PKI à très grand volume ou les environnements PKI qui exigent une solution légère pour minimiser la bande passante de le traitement côté client. Vu que OCSP n'a pas de moyen de signaler les capacités du répondeur dans le protocole, les clients doivent différencier les réponses OCSP produites par les répondeurs conforme avec ce profile et ceux qui n'ont pas besoin de s'appuyer sur des mécanismes externe pour déterminer quand une réponse opère en accord avec ce profile et, quand les exigences de ce profile s'appliquent. Dans le cas où des mécanismes tiers ne sont pas disponible entre un client OCSP 2560 et un répondeur qui opère dans un mode décrit dans cette spécification.

Profile de demande OCSP

Structure OCSPRequest

OCSPRequest conforme a ce profile doit inclure seulement une Request dans la structure OCSPRequest.RequestList

Les clients ne doivent pas utiliser SHA1 comme algorithme de hash pour les valeurs CertID.issuerNameHash et 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 inclue, ce profile recommande qu'il ne contienne seulement que l'extension nonce (id-pkix-ocsp-nonce).

OCSPRequests signées

Les clients ne doivent pas envoyer d'OCSPRequests signés. Les répondeurs peuvent ignorer la signature dans les requêtes OCSPRequests.

Si OCSPRequest est signé, le client devrait spécifier son nom dans le champ OCSPRequest.requestorName, sinon, les clients ne devraient pas inclure le champ requestorName dans OCSPRequest. Les serveurs OCSP doivent être préparés à recevoir des requêtes OCSP non-signés qui contiennent le champ requestorName, mais doivent réaliser que la valeur fournie n'est pas authentifiée.

Profile de réponse OCSP

Structure OCSPResponse

Les répondeurs doivent générer un BasicOCSPResponse comme identifié par l'OID BasicOCSPResponse. Les OCSPResponses conformes à ce profile devraient inclure seulement un SingleResponse dans la structure ResponseData.responses, mais peuvent inclure des éléments SingleResponse additionnels si nécessaire pour améliorer les performances de pré-génération de réponse ou l'efficacité des caches.

Le répondeur ne devrait pas inclure responseExtensions. Comme spécifié dans OCSP, les clients doivent ignorer les responsesExtensions non-critiques dans la réponse.

Dans le cas où un répondeur n'a pas la possibilité de répondre à une demande OCSP contenant une option non supportée par le serveur, il devrait retourner la réponse la plus complète qu'il peut. Par exemple, dans le cas où un répondeur ne supporte que les réponses pré-produites et n'a pas la possibilité de répondre à une demande OCSP contenant a nonce, il devrait retourner une réponse qui n'inclut pas un nonce.

Les clients devraient tenter de traiter une réponse même si la réponse n'inclue pas un nonce.

Les répondeurs qui n'ont pas la capacité de répondre aux requêtes OCSP qui contiennent une option non-supportée telle que nonce peuvent renvoyer la requête à un répondeur OCSP qui en a la capacité.

Le répondeur peut inclure la structure d'extensions singleResponse.singleResponse.

OCSPResponses signés

Les clients doivent valider la signature de l'OCSPResponse retourné. Si la réponse est signée par un délégué de l'autorité de certification émettrice, un certificat de répondeur valide doit être référencé dans la structure BasicOCSPResponse.certs.

Il est recommandé que le certificat du répondeur OCSP contienne l'extension id-pkix-ocsp-nocheck, comme définis dans OCSP, pour indiquer au client qu'il ne doit pas vérifier le statut du certificat. De plus, il est recommandé que ni une extension authorityInfoAccess ni une extension cRLDistributionPoints ne soit inclus dans le certificat du répondeur OCSP. En accord avec ça, le certificat de signature du répondeur devrait être relativement de courte durée et renouvelé régulièrement.

Les clients doivent être capable d'identifier les certificats de répondeur OCSP en utilisant les chois ResponseData.ResponderID byName

et byKey. Les répondeur devraient utiliser byKey pour réduire ensuite la taille de la réponse dans les scénarios où la réduction de la bander passante est un problème.

Valeurs OCSPResponseStatus

Tant que l'infrastructure OCSP a des enregistrements autoritatifs pour un certificat particulier, un OCSPResponseStatus "Success" sera retourné. Quand l'accès aux enregistrements autoritatifs pour un certificat particulier n'est pas disponible, le répondeur doit retourner un OCSPResponseStatus "unautorized". Ce profile étend la rfc2560 de "unauthorized" comme suit :

La réponse "unauthorized" est retournée dans le cas où le client n'est pas autorisé à faire cette demande à ce serveur ou quand le serveur n'est pas capable de répondre de manière autoritative.

Par exemple, les répondeurs OCSP qui n'ont pas accès aux enregistrements autoritatifs pour un certificat, tel que ceux qui génèrent et distribuent les réponses OCSP à l'avance et donc n'ont pas la capacité de répondre correctement avec une réponse "success" ni "unknown", vont répondre avec un OCSPResponseStatus "unauthorized". Également, pour s'assurer que la base d'information de révocation ne grandis pas indéfiniment dans le temps, le répondeur peut supprimer les enregistrements des certificats expirés. Les demandes des clients pour les certificat dont l'enregistrement a été supprimé vont résulter dans un OCSPResponseStatus "unauthorized".

thisUpdate, nextUpdate, et producedAt

En pré-produisant les messages OCSPResponse, le répondeur doit définir les temps thisUpdate, nextUpdate et producedAt comme suit :

thisUpdate La date à laquelle le statut indiqué est connu comme étant correcte

nextUpdate La date à laquelle ou avant laquelle de nouvelles informations seront disponibles sur le statut des certificats. Les répondeur doivent toujours inclure cette valeur pour aider dans le caching des réponses.

producedAt La date à laquelle la réponse OCSP a été signée

Note : Dans de nombreux cas la valeur de thisUpdate et producedAt seront les même.

Pour ce profile, les valeurs GeneralizedTime encodé ASN.1 pour thisUpdate, nextUpdate et producedAt doivent être exprimés en GMT et doivent inclure les secondes, même quand le nombre de secondes est 0. Les valeurs GeneralizedTime ne doivent pas inclure les fractions de secondes.

Comportement du client

Découverte du répondeur OCSP

Les clients doivent supporter l'extension authorityInfoAccess comme définis dans PKIX et doivent reconnaître la méthode d'accès id-ad-ocsp. Cela permet d'informer les clients sur la manière de contacter le service OCSP.

Dans le cas où un client vérifie le statut d'un certificat qui contient une extension authorityInformationAccess pointant vers un répondeur OCSP et une extension cRLDistributionPoints pointant vers une CRL, le client devrait tenter de contacter le répondeur OCSP en premier. Les clients peuvent tenter de récupérer la CRL si aucune OCSPResponse n'est reçu du répondeur.

Envoyer une demande OCSP

Pour éviter le trafic réseaux inutile, les applications doivent vérifier la signature d'une donnée signée avant de demander à un client OCSP de vérifier le statut des certificats utilisé pour vérifier la donnée. Si la signature est invalide ou l'application n'est pas capable de la vérifier, une vérification OCSP ne doit pas être tentée.

Similairement, une application doit valider la signature dans les certificats dans une chaîne, avant de demander à un client OCSP de vérifier le statut d'un certificat. Si la signature du certificat est invalide ou que l'application n'est pas capable de le vérifier, une vérification OCSP ne doit pas être faite. Les clients ne devraient pas faire de demande de statut des certificats expirés.

S'assure qu'un OCSPResponse est récent

Pour s'assurer qu'un client n'accepte pas de réponse expirée qui indique un statut 'good' il y a une réponse plus récente qui spécifie un statut 'revoked', un client doit s'assurer que les réponses qu'ils reçoivent sont récentes.

En général, 2 mécanismes sont disponibles aux clients pour s'assurer qu'une réponse est récente. La première utilise nonces, et la seconde est basée sur la date. Pour que les mécanismes basés sur les dates fonctionnent, les clients et les répondeurs doivent avoir accès à une source de temps précise.

Parce que ce profile spécifie que les clients ne devraient pas inclure une structure requestExtensions dans OCSPRequests, les clients doivent être capable de déterminer que OCSPResponse est récent basé sur une source de temps précise. Les clients qui optent pour inclure un nonce dans le demande ne doivent pas rejeter une OCSPResponse correspondante seulement sur la base de la non-existance du nonce attendu, mais doivent valider l'OCSPResponse basé sur le temps.

Les clients qui n'incluent pas un nonce dans la requête doivent ignorer tout nonce qui peut être présent dans la réponse.

Les clients doivent vérifier l'existence du champ nextUpdate et doivent s'assurer de la date courante, exprimée en temps GMT, qui doit être entre le thisUpdate et le nextUpdate. Si le champ nextUpdate est absent, le client doit rejeter la réponse.

Si le champs nextUpdate est présent, le client doit s'assurer qu'il n'est pas antérieur la date courante. Si la date courante dans le client est ultérieur au temps spécifié dans le champ nextUpdate, le client doit rejeter la réponse. Les clients peuvent autoriser la configuration d'une petite période de tolérance pour accepter des réponses après nextUpdate pour gérer les différences mineurs d'horloge relative aux répondeurs et aux caches. Cette période de tolérance devrait être choisis sur la précision de la technologie de synchronisation de temps disponible dans l'environnement de l'application appelante. Par exemple, les paires internet avec connexion à faibles latences s'attendent à une synchronisation du temps NTP avec une précision élevée ; les environnements à haute latence ou quand un NTP n'est pas disponible peuvent avoir besoin d'une tolérance plus élevée.

Profile de transport

Le répondeur OCSP doit supporter les requêtes et réponses sur HTTP. En envoyant les requêtes qui sont inférieur ou égal à 255 octets au total (après encodage) incluant le schéma et les délimiteurs (http ://), le nom du serveur et la structure OCSPRequest encodé en base 64, les clients doivent utiliser la méthode GET (pour permettre le caching de réponse OCSP). Les requêtes OCSP supérieurs à 255 octets devraient être envoyés en utilisant la méthode POST. Dans tous les cas, les clients doivent suivre les descriptions définis dans la norme OCSP en construisant leurs messages.

En construisant un message GET, les clients OCSP doivent encoder en base 64 la structure OCSPRequest et l'ajouter à l'URI spécifiée dans l'extension authorityInfoAccess. Les clients ne doivent pas inclure les caractères CR ou LF dans la chaîne encodée base 64. Les clients doivent encoder l'url proprement, par exemple :
`http://ocsp.example.com/MEowSDBGMEQwQjAKBggqhkiG9w0CBQQQ7sp6GTKpL2dAdeGaW267owQQqInESWQD0mGeBArSgv%2FBWQIQLJx%2Fg9x`

En réponse aux OCSPRequests formatés proprement qui sont capable d'être mis en cache (par exemple, les réponses qui contiennent une valeur nextUpdate), le répondeur va inclure la valeur binaire de l'encodé DER de l'OCSPResponse précédé par l'en-tête HTTP suivant :

```
content-type: application/ocsp-response
content-length: <OCSP response length>
last-modified: <producedAt [HTTP] date>
ETag: "<strong validator>"
expires: <nextUpdate [HTTP] date>
cache-control: max-age=<n>, public, no-transform, must-revalidate
date: <current [HTTP] date>
```

Recommandations de mise en cache

La capacité de mise en cache des réponses OCSP via le réseaux est un facteur important dans les déploiements à haut volume. Cette section discute de comportement de la mise en cache des clients OCSP et des proxy HTTP et les étapes qui devraient être faites pour minimiser le nombre de fois que les clients OCSP utilisent le réseaux. De plus, le concept de réponses OCSP incluses dans les échanges de protocole (stapling ou piggybacking), tel que définis dans TLS, sont également discutés.

Mise en cache par le client

Pour minimiser l'utilisation de la bande passante, les clients doivent mettre en cache localement les réponses OCSP autoritatives (ex : une réponse avec une signature qui a été validée avec succès et qui indique un OCSPResponseStatus 'successful').

Beaucoup de clients OCSP vont envoyer des OCSPRequests à ou près du temps nextUpdate (quand une réponse mise en cache expire). Pour éviter des pointes importantes dans la charge des répondeurs qui peuvent se produire quand de nombreux clients rafraîchissent les réponses en cache pour un certificat populaire, les répondeurs peuvent indiquer quand le client devrait chercher une réponse OCSP mise à jours en utilisant la directive cache-control :max-age. Pour s'assurer que les clients reçoivent une réponse mises à jours, les répondeurs OCSP doivent rafraîchir la réponse OCSP avant le temps max-age.

Proxy HTTP

Le répondeur devrait définir l'en-tête HTTP dans la réponse OCSP de manière à permettre une utilisation intelligente des serveurs proxy intermédiaire. Voir la norme HTTP pour la définition de ces en-têtes et le format correcte.

HTTP Header___Description

date La date et l'heure à laquelle le serveur OCSP a généré la réponse HTTP.

last-modified Cette valeur spécifie la date et l'heure à laquelle le répondeur OCSP à modifié la réponse. Cette date sera la même que thisUpdate dans la requête elle-même.

expires Spécifie la durée de à laquelle la réponse est considérée comme récente. Cette date est la même que nextUpdate dans la réponse OCSP.

ETag Une chaîne qui identifie une version particulière de la donnée associée. Ce profile recommande que la valeur ETag soit une représentation hexa ASCII du hash SHA1 de la structure OCSPResponse

cache-control Contient des directives de mise en cache :

* **max-age=<n>** où n est la valeur de temps ultérieur à thisUpdate mais inférieur à nextUpdate.
* **public** rend la réponse non cachable
* **no-transform** Spécifie qu'une proxy cache ne peut pas changer le type, longueur, ou encodage du contenu de l'objet.
* **must-revalidate** empêche les caches de retourner des réponses périmées intentionnellement

Les répondeurs OCSP ne doivent pas inclure d'en-tête "Pragma : no-cache", "Cache-Control : no-cache", ou "Cache-Control : no-store" dans les réponses OCSP autoritatives. Les répondeurs OCSP devraient inclure un ou plusieurs de ces en-têtes dans les réponses non-autoritatives.

Par exemple, supposons une réponse OCSP ayant les horodatages suivant :

```
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 que les clients demandent la réponse le 2 may 2005 01 :00 :00 GMT. Dans ce scénario, la réponse HTTP peut ressembler à ceci :

```
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 d'en-tête no-cache dans les demandes, sauf si le client rencontre une réponse expirée qui peut être le résultat d'une donnée périmée d'un proxy cache. Dans cette situation, les clients devraient renvoyer le demande en spécifiant que les proxy devraient être bypassés en incluant un en-tête HTP approprié dans la demande (ex. Pragma : no-cache ou Cache-Control : no-cache).

Mise en cache des serveurs

Dans certains scénarios, il est avantageux d'inclure les information de réponse OCSP dans le protocole utilisé entre le client et le serveur. En incluant les réponses OCSP de cette manière a quelques effet désirable.

D'abord, il permet la mise en cache des réponses OCSP dans le serveur diminuant ainsi le nombre de requêtes au répondeur.

Ensuite, il permet la validation de certificat dans le cas où le client n'est pas connecté à un réseaux et donc élimine le besoin pour les clients d'établir une nouvelle session HTTP avec le répondeur.

Il réduit le nombre d'aller-retours que le client a besoin pour compléter un handshake

Enfin, il simplifie l'implémentation OCSP côté client en permettant une situation où le client n'a seulement besoin de la capacité de parser et reconnaître les réponses OCSP.

Cette fonctionnalité a été spécifiée comme extension du protocole TLS, mais peut être appliqué à tout protocole client-serveur.

Ce profile recommande que les clients et les serveurs TLS implémentent le mécanisme d'extension de demande de statut de certificat pour TLS.

D'autres informations sur les problèmes de mise en cache peuvent être obtenus depuis la rfc3143.

Considérations de sécurité

Les considérations suivantes s'appliquent en plus des considérations de sécurité spécifié dans la norme OCSP.

Attaques Replay

Puisque l'utilisation de nonce dans ce profile est optionnel, il y a possibilité qu'une réponse OCSP périmée puisse être rejouée, causant le client à accepter une réponse 'good' quand en fait il a été révoké. Pour réduire cette attaque, les clients doivent avoir accès à une source de temps précise et s'assurer que les réponses OCSP qu'ils reçoivent sont suffisamment récentes.

Les clients qui n'ont pas de source de temps précise sont vulnérables. Par exemple, un client avec une horloge suffisamment rapide peut rejeter une réponse OCSP récente et accepter une réponses valide qui est expirée pour des certificat qui sont en fait révoqués.

De future versions du protocole OCSP peuvent fournir une manière pour le client de savoir si le serveur supporte nonce ou non. Si un client peut déterminer que le serveur supporte nonce, il doit rejeter une réponse qui ne contient pas le nonce attendu. Sinon, les client qui optent pour inclure un nonce dans la demande ne devraient pas rejeter l'OCSPResponse correspondant uniquement sur la base de la non-existance du nonce, mais doivent également valider OCSPResponse sur la base du temps.

Attaques Man-In-The-Middle

Pour réduire les risques associés avec ce type d'attaque, le client doit valider correctement la signature de la réponse. L'utilisation de réponses signées dans OCSP sert à authentifier l'identité du répondeur OCSP et de vérifier qu'il est autorisé à signer les réponses de la CA.

Attaques pas usurpation d'identité

L'utilisation de réponses signées dans le serveur OCSP sert à authentifier l'identité du répondeur OCSP. Comme détaillé dans OCSP, les clients doivent valider correctement la signature de la réponse OCSP et la signature du certificat du signataire OCSP pour s'assurer que le répondeur est autorisé.

Attaques pas deni de service

Les répondeurs OCSP devraient prendre des mesures pour empêcher ou réduire les attaques par déni de service. Vu que ce profile spécifie l'utilisation de OCSPRequests non-signés, l'accès au répondeur peut être donné implicitement par toute personne qui souhaite envoyer une demande à un répondeur, et ainsi la capacité de monter des attaques par déni de service via des demandes en rafales. Par exemple, un répondeur peut limiter le taux de demande entrante pour une adresse IP particulière si un comportement douteux est détecté.

Modification des en-têtes HTTP

Les valeurs incluses dans les en-têtes HTTP ne sont pas protégées cryptographiquement ; elles peuvent être manipulées par un attaquant. Les clients devraient utiliser ces valeurs pour la mise en cache seulement après s'assurer que les valeurs sont présentes dans le OCSPResponse signé. Les clients ne devraient pas valider les réponses au-delà de la date nextUpdate.

Authentification et autorisation des demandes

La suggestion d'utiliser des demandes non-signée dans cet environnement supprime une option qui permet au répondeur de déterminer l'authenticité de la requête entrante. Donc, l'accès au répondeur peut être implicitement donné à toute personne qui peut envoyer une demande à un répondeur. Les environnement où l'autorisation explicite pour accéder au répondeur OCSP est nécessaire peuvent utiliser

d'autres mécanismes pour authentifier les demandeurs, ou restreindre le service.