Nonce Sense

Freshness and Security in OCSP Responses

Contact us at:
CoreStreet Ltd.
One Alewife Center
Suite 200
Cambridge, MA 02140

info@corestreet.com
Tel: 617-661-3554

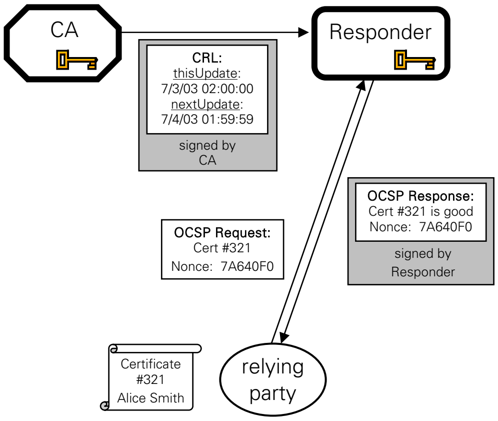
Introduction

The Online Certificate Status Protocol (OCSP) provides two alternate ways for a relying party to prove whether the contents of an OCSP response are up-to-date. Response "freshness" can be determined either using a challenge-response scheme or through secure timestamps within responses.

CoreStreet's RTC VA supports both techniques, but we feel that a large OCSP deployment requires a careful examination of the costs and risks associated with each freshness solution. This tech note analyzes the total system security factors for both approaches for large deployments.

Nonce-based OCSP freshness

An OCSP *nonce* offers an optional tool for a relying party to determine that it is receiving reasonably up-to-date certificate status information from a responder. A nonce is a random sequence of 20 bytes that is placed in an OCSP request, and the responder must use its secret key to sign a response containing that nonce:



In this configuration, the OCSP responder provides a fresh signed response for every request. The nonce proves to the relying party that the OCSP response contains status information that is as current as the data available to the responder (e.g. as current as the last CRL).

A small PKI can deploy a handful of responders which receive extremely frequent updates of certificate status changes. In these limited deployments, nonce-based responders may be able to offer fresh responses that contain status information that is only a few minutes out of date.

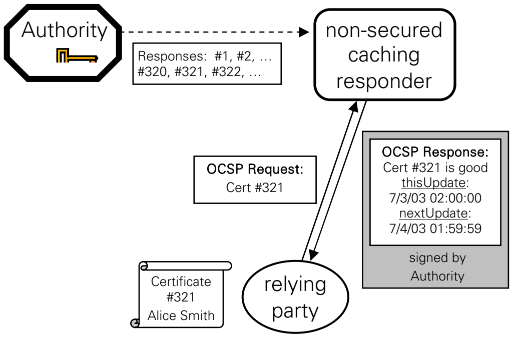
Larger deployments with millions of certificates or dozens of responder locations must rely on less frequent status updates. In a typical large PKI, each responder may receive a daily CRL which it will use for the next 24 hours.

In this large PKI, a nonce proves that the response is fresh from the responder, but the underlying status information (the CRL) may be hours out of date. This is true for any OCSP responder that is not directly integrated into the CA server, which is the only case where a responder can give truly up-to-date status responses.

Time-based freshness

Without nonces, an OCSP infrastructure can establish the current freshness of each response by marking its validity interval through use of `thisUpdate` and `nextUpdate` response fields. This allows a responder to use the same response for multiple requests rather than performing a new digital signature on every response.

In particular, this allows the caching of pre-signed responses by non-secured responders with no signature keys:



In this configuration, the responder provides a response that is fresh for a fixed period of time. A relying party will accept a response which is marked as valid at the time of the OCSP request, and reject any response that has expired. This is identical to the freshness used for CRLs, which provide reusable status information for a fixed number of minutes or hours.

The duration of a time-based response can be configured based on the acceptable minimum freshness allowed, so a medium security configuration could use responses that are good for 24 hours, and a high security configuration could use responses that are good for one hour.

In a large installation, the validity duration of an OCSP response will match the times for the original CRL that was used to determine certificate status, and the freshness security will match a CRL-based validation solution.

Security comparison

In a small installation with an integrated CA responder or a few nearby independent responders, nonce-based freshness can be used to ensure that relying parties receive information that is only a few seconds or minutes out of date. This may be important, for example, for high value financial transactions in low traffic infrastructures.

In installations with larger numbers of certificates and responders, responders will provide status information using CRLs that are cached for hours at a time, so the freshness of the status information in nonce-based responses is no more up-to-date than a similar time-based response.

In addition, an infrastructure relying on nonce-based freshness is uniquely vulnerable to two types of attacks.

First, an attacker may attempt to prevent a responder from providing responses by attacking its network service. In particular, an attacker may create a Denial of Service by sending more requests than can be handled by the responder. For every simple request sent by an attacker, a nonce-based responder must perform a 1024-bit RSA signature, which is a computationally intensive operation. An attacker with a 56kbps modem can easily generate enough requests (as few as 50 per second) to produce a Denial of Service in a nonce-based responder.

A time-based caching responder does not perform a digital signature for every response. It can typically provide twenty times as many responses per second as a nonce-based responder, and it can be replicated without incurring the cost of additional Hardware Security Modules, so a caching responder infrastructure is much more resistant to this type of Denial of Service attack.

Second, every nonce-based responder must contain a private key that is used for response signing. An attacker that gains use of the key can generate nonce-based responses to “validate” any revoked certificate or “revoke” any valid certificate. A hacker who gains physical or network access to the responder host could make the responder indicate that a stolen ID card is still valid, allowing it to be used to access a secure facility or network server.

A nonce-based responder infrastructure can reduce the ease of this attack with hardware security modules, firewalls, guarded hosting facilities, and intrusion detection systems, but responders will always be vulnerable to attacks through their OCSP protocol. Any successful attack exploiting a buffer overflow or other bug in the servers could be used to create a critical security breach.

A caching responder with time-based responses is immune to this severe type of attack, since it has no keys to protect. A successful attack on an non-secured caching responder can prevent it from issuing responses, but it cannot be made to sign forged responses.

For these reasons, an OCSP infrastructure using time-based caching responders offers significantly higher overall security for large deployments than a deployment with dozens of nonce-based responders with sensitive private keys. The risks from nonce-based responders are primarily justified in small deployments with protected networks and limited usage.