Internet Engineering Task Force	P. Hallam-Baker
Internet-Draft	Comodo Group Inc.
Intended status: Standards Track	June 07, 2013
Expires: December 09, 2013	

PKIX Operations: Certificate Status

draft-hallambaker-pkixstatus-00

Abstract

[abs]

Status of This Memo

Copyright Notice

Table of Contents

1. Certificate Status

A certificate is issued with a predetermined validity interval. It is common practice to specify a validity interval that starts a few hours or days before the instant of issue so as to avoid rejection by machines with clocks running behind the current time or otherwise mis-set. In normal operation the certificate will remain valid until it expires.

The CA that issued a certificate has primary responsibility for maintaining the certificate life cycle and reporting changes to certificate status. But other parties can and in some cases do report status for third party certificates. In particular client and platform providers have revoked certificates known to have been mis-issued or in a case of a CA breach.

1.1. Operational Certificate Lifecycle Model

PKIX does not describe a certificate lifecyle model. Instead the certificate lifecycle model is a consequence of the issue of PKIX Certificates and CRLs. While this is sufficient for describing PKIX it is not satisfactory as a reference model for describing operations. Not least because modern PKIX operations are frequently based on the use of OCSP rather than CRLs and differences in the semantics of CRLs and OCSP are one of the features we would want to measure. The distinction between an operational model and PKIX semantics is illustrated by considering the difference between the operational concept of direct/indirect status assertions and the PKIX semantics of direct/indirect CRLs.

1.1.1. Direct and Indirect Status Assertions

PKIX CRLs may be marked as direct or indirect to indicate that they are issued by the same CA that issued the original certificate (a direct CRL) or by a third party (an indirect CRL).

In the corresponding operational model we define a direct status assertion as being by the same CA that issued the original certificate and an indirect status assertion as being any status assertion that is not direct.

The difference between the operational and PKIX models has imporant practical consequences. The CA that originally issued an assertion naturally holds a privileged position when it comes to revoking it. A direct CRL thus has a privileged position when considering the question of certificate validity. A direct status assertion thus has a privileged position when considering revocation status. A direct CRL carries an implicit claim that it is a direct status assertion but this is merely a claim unless the client validating the CRL takes steps to verify it. For example by verifying that the CRL signature has valid trust chain to the same trust anchor as the certificate.

CRLs introduce a further complication as a CRL contains a list of explicit statements declaring that a certificate is invalid. In the case of a direct CRL there is an implicit assertion that any issued, unexpired certificate not listed was valid at the time the CRL was issued. The processing rules specified in [RFC5280] appear to limit this implicit assertion to direct CRLs but this is does not appear to be called out in the text.

One of the main use cases that might motivate the issue of an indirect status assertion is the case where a third party notices that a certificate is being used for malicious purposes and intends to advise relying parties that they should not rely on that certificate. There is thus a case for granting third parties the ability to revoke certificates but does granting this ability also confer the ability to (implicitly) declare certificates valid?

[Operational question: Do clients interpret indirect CRLs as substitutes for the direct CRL or as adjuncts providing additional information.]

1.1.2. Trust Path Processing

One of the operational questions we would like to understand is the extent to which it is possible to revoke EE certificates by revoking one or more of the CSCs in the certification path.

Self Signed certificates used to transport Trust Anchors are not actually PKIX certificates and are not governed by the PKIX model. One important consequence of this is that relying parties do not use PKIX mechanisms to check the validity of Trust Anchors.

CSCs signed by the trust anchor are potentially subject to revocation. Do the status checking mechanisms employed in browsers support this in practice?

[OCSP and CRLs raise separate issues here. In the case of an OCSP responder should we require signed OCSP tokens for each cert in the path? Is it possible to use a mix of CSCs and OCSP in stapled tokens?]

1.1.3. Revocation Reasons

A status declarer may declare a certificate invalid (i.e. revoke the certificate) before its scheduled expiry for a variety of reason that include:

Subject requested revocation:
: The certificate subject requested revocation.

Subject requested correction:
: The certificate subject requested information in a certificate be corrected. Such corrections are typically made by revoking the original certificate and issuing a replacement.

Payment declined:
: A CA may issue a certificate before payment has cleared. If the payment is subsequently declined, the certificate is revoked.

Declined extension:
: The certificate was originally issued on condition that use beyond an initial period would

require an additional fee which the subject did not pay.

Terms of Use:
: The subject was determined to have breached the terms of use

Fraudulent Request:
: The application was determined to be fraudulent after issue

CA compromise:
: The certificate can no longer be trusted because the operations of the CA were compromised.

The ability to provide a reason for revocation is defined without explaining the reason a CA should provide this information or how relying parties should behave differently according to the revocation reason given. Revoked certificates are to be considered invalid regardless of the reason for revocation.

PKIX does not define an order of severity. In cases where multiple reasons apply, the CA may pick any. There is no obligation to report a reason at all let alone report severity.

Once a certificate is revoked the certificate lifecycle is complete as far as the CA is concerned and there is no obligation on the CA to update the revocation reason after the fact to reflect the discovery of a more serious cause.

In the case of a subject request the CA only has reliable knowledge of the fact of the request and not the reason(s) the request was made. A certificate subject might have requested the certificate be revoked because they no further use for it or because they know the associated private key has been compromised. Even if the CA asks for the revocation reason there is no reason to expect the subject to answer. The subject may not wish to report that a private key has been compromised.

The net effect of these limitations is that revocation reasons only provide a lower bound on the severity of the cause for which a certificate was revoked.

1.1.4. Operational Certificate States

From an operational point of view, the lifecycle of a PKIX certificate has five potential states:

Valid
: The certificate was issued and is valid.

Invalid
: No certificate was issued or the certificate issued is no longer valid.

  Nonexistent
  : The certificate does not exist. This may be because the certificate has not yet been issued or it will never be issued.

  Hold
  : The certificate exists but has been suspended with the possibility of reinstatement.

  Revoked
  : The certificate exists but has been declared to be invalid with permanent effect.

  Expired
  : The certificate existed in the past but the expiry date specified at issue has passed.

The Hold state has been found to be of little or no practical value since issuing a new certificate is simpler and more effective than attempting to cancel a previous instruction to put the certificate on hold.

CRLs and certain OCSP configurations do not permit a client to distinguish between the states Valid and Invalid/Nonexistent. The CRL mechanism was designed to allow a relying party to check the validity of a known certificate. It was thus unnecessary to distinguish the states Valid and Nonexistent as that would be verified by checking the signature. Accordingly a CRL contains only a list of invalid certificates.

In the case of a CA Breach, key compromise or cryptanalytic attack, a certificate may be created that has a valid signature but was not issued by the CA. Such a certificate is 'Nonexistent' as far as the CA is concerned. Requiring a CA to distinguish these states in reporting certificate status provides a limited degree of transparency in CA operations. A CA that reports 'Nonexistent' in response to a status request for an unexpired certificate that has a valid signature has a defective or breached issue process. A CA that reports valid in response to a status request for a non-existent certificate has a defective or breached revocation mechanism.

1.2. Client Behavior

WebPKI clients are advised but not required to check certificate status before relying on the assertions they contain. Waiting to obtain status information from an external source before relying on a certificate may cause delay or even rejection of a valid certificate.

Excluding the possibility that a client requests revocation status then ignores the result, the options available to a Web PKI client are therefore:

Ignore
: The client does not process revocation status from any source

Local
: The client only process revocation status that is available from local sources. For example hardcoded 'do not trust' lists.

Soft-Fail
: The client attempts to obtain revocation status from external sources and will reject certificates reported as revoked but will accept a certificate as valid if the external source does not reply.

Hard-Fail
: The client attempts to obtain revocation status from external sources and will reject certificates unless an affirmative assertion of validity is obtained.

2. Status Assertion Mechanisms

2.1. CRLs

The PKIX CRL mechanism for asserting certificate status is described in [RFC5280].

2.1.1. Status Model

A CRL only provides a list of certificates that have been revoked. An issued, unexpired certificate is presumed to be valid if it does not appear in the CRL. The certificate states supported by the CRL mechanism are thus:

UNREVOKED
: Corresponds to operational states Valid, Nonexistent and Expired.

UNDETERMINED
: Occurs when no CRL with a corresponding scope is available.

REVOKED
: Corresponds to operational state Revoked.

HOLD
: Corresponds to operational state Hold.

The CRL result 'UNREVOKED' thus corresponds to three states in the Operational model of which one is Valid and the other two are Invalid states. A client that does not have a source of trusted time available may use the issue time of the CRL as the basis for checking expiry. The CRL mechanism does not provide a means of determining that a certificate was legitimately issued

2.1.2. Revocation Reasons

[RFC5280] requires that a CRL entry specify a reason code but not the circumstances in which a code should be raised. [This is however specified in X.509v3] The following reason codes are defined:

unspecified

keyCompromise

cACompromise

affiliationChanged

superseded

cessationOfOperation

privilegeWithdrawn

aACompromise

2.2. OCSP

OCSP is defined in [RFC6960]. [RFC5019] (lightweight) and TLS Stapling [RFC6066] Section 8.

The OCSP protocol permits responses to be signed in advance [static] or provide a proof of freshness by returning a nonce presented by the client.

The protocol only permits static responses to report the status of individual certificates. There is no feature analagous to the NSEC3 feature of DNSSEC which permits the non-existence of an entry in a particular range to be asserted.

[CABForum (expected to) mandate distinction of Valid / Nonexistent]

2.2.1. CRL Responder

An OCSP responder may generate responses from CRLs. Such a responder can generate most but not all the responses required in advance by generating revoked responses for all the certificates listed in the CRL and valid responses for all the certificate serial numbers presented in previous requests.

Such a responder cannot distinguish between Valid and nonexistent states unless provided with additional information not in the CRL.

2.2.2. Lightweight Distribution

[RFC5019]

In the lightweight distribution mode of operation, the CA generates OCSP responses for all unexpired certificates that it has issued. The signed tokens are then passed to a separate network for distribution. For example, a Content Delivery Network with a large number of delivery points.

One of the main strengths of this model is that all the signing of OCSP tokens is done offline and no signing key is ever exposed to an external network. One consequence of this model is that responses for nonexistent certificates cannot be signed.

2.2.3. OCSP Stapling

One of the principle limitations of the traditional OCSP model is that each TLS transaction becomes a three party communication. To complete the TLS connection the client must communicate with the server being contacted and the OCSP service. This approach introduces unnecessary delay and an additional potential point of failure and is therefore unsatisfactory.

OCSP stapling permits a TLS server to provide a client that supports the stapling extension to provide the OCSP token together with the certificate it corresponds to. This permits a client to establish a TLS communication without the need for a three party communication in the case that the client and server both support stapling.

The chief drawback to stapling is that support for stapling is optional. thus a client that does not receive a stapled token must attempt to obtain it from the OCSP service and is therefore subject to the same Softfail/hardfail dilemma described above.

2.3. Other

2.3.1. Hardcoded/Indirect Revocation List

[Commonly employed in browsers]

[Some earlier versions could only be updated by changing all the code. Very inflexible.]

[Updatable revocation lists]

2.3.2. DANE

DANE assertions [RFC6698] may be used to cancel a certificate. [describe]

2.3.3. Certificate Transparency

CT [RFC6962] provides a means of auditing the operation of a CA using only information that is available to the public. Moreover a client can determine that a certificate has been issued transparently or not. [describe]

[Allows another way to distinguish Valid and nonexistent and thus CA breach.]

3. Status Acquisition Mechanisms

3.1. Google's Status Mechanism

3.2. SCVP

3.3. XKMS

4. Status

Historical behavior is only of interest to the extent that it affects current operations.

Every PKIX certificate has a built in expiry date. Thus we are only interested in CA operations from the date at which their oldest unexpired certificate is still valid.

4.1. CAs

Describe survey methodology here (self reporting)

4.1.1. CA-Browser Forum Requirements

Here put the common requirements.

4.2. Servers

4.3. Clients

5. Security Considerations

Put something here?

6. IANA Considerations

None

7. References

7.1. Normative References

[RFC5019] Deacon, A. and R. Hurst, "The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments", RFC 5019, September 2007.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6277] Santesson, S. and P. Hallam-Baker, "Online Certificate Status Protocol Algorithm Agility", RFC 6277, June 2011.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S. and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, June 2013.

7.2. Non Normative References

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

[RFC6962] Laurie, B., Langley, A. and E. Kasper, "Certificate Transparency", RFC 6962, June 2013.

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.
EMail: philliph@comodo.com