Internet Engineering Task Force (IETF)
Internet-Draft
Intended Status: Standards Track
Expires: November 13, 2014

Phillip Hallam-Baker
Comodo Group Inc.
David Chadwick
University of Kent

May 12, 2014

Web PKI Operations: Revocation and Status
draft-wpkops-revocation-00

Abstract

This document describes the certificate status mechanisms supported in the Web PKI

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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.

[Introduce CRL Sets here, once I find a citation]

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 important 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 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 trust the certificate subject. Since it is the behavior of the subject rather than their identity that is at issue, there may not be sufficient reason for the CA to revoke the certificate. There is thus a case for parties other than the certificate subject and issuer having the ability to revoke certificates in certain circumstances. 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 Certificate Signing Certificates 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 (although they are X.509v3 certificates). 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 either because the original information is wrong or circumstances have changed. For example the subject's affiliation has changed. 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 why 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 have 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, once issued, a PKIX certificate has five potential states, a single valid state and four invalid states:

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

Valid
: The certificate was issued and is valid.

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

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 cannot be contacted, does not reply or rejects the request, etc.

Hard-Fail
: The client attempts to obtain revocation status from external sources and will reject certificates unless either an affirmative assertion of validity or an affirmative assertion of not revoked 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.

An OCSP service MAY return the following responses to a request:

Success [[+ CRL Status Code]
: The OCSP status request succeeded and the service returned one of the CRL status codes described above.

Refused
: The OCSP responder refused to answer the request.

Unknown
: The certificate for which status was requested was not found or the status is not determined.

Invalid
: The OCSP server returned an answer that was not understood.

Fail
: The service failed to answer the request or the client was unable to contact the OCSP service.

The OCSP response reflects the success or failure of the OCSP transaction rather than the status of the certificate being queried. Thus a client whose behavior is Soft-Fail will only reject a certificate if the OCSP response Success and an Invalid certificate status is returned. Thus an OCSP server that responds to a request for status for a certificate that is known to have never been issued with 'Invalid' will cause soft-fail clients to accept the certificate.

Note that [RFC6960] does not differentiate the results Success/Valid and Unknown. CAs are however required to differentiate these responses under the CABForum Basic Requirements [TBS].

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.

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

In the lightweight distribution mode of operation specified in [RFC5019], 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

Most browsers employ a 'blacklist' to block certificates known to be mis-issued. The number of entries supported in such lists is typically small. In some cases the list is hardcoded in the browser or platform code and is only updated with the browser or platform. In other cases the blacklist is updatable separately.

Q: Which Web browsers support update of the list without updating the browser.

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 from log servers. Moreover a client can determine that a certificate has been issued transparently (i.e. is in the log) or not. Any certificate the client receives that is not in the log should be treated as suspicious or invalid by the client.

In order for CT to work, a CA registers a certificate in a log server and is returned a signed time stamp by the log server. This signed time stamp must be given to the certificate subject, and it must send this to RPs along with its certificate. This requires the TLS handshake to be enhanced to pass the signed time stamp, and clients and servers to be enhanced to receive and send it. This will take time to be rolled out to the entire Internet, so CT is not a short term solution.

Monitor servers regularly scan the logs to look for suspicious or unauthorised certificates that have been deposited there.

One potential problem with CT is that it is not described how the monitors will operate and determine whether a certificate is suspicious or unauthorised. How will they know if a certificate is suspicious or not? How will clients be notified of this? What does suspicious actually mean to a client? How should a client behave when it is told that a certificate is dodgy by a monitor? Who arbitrates on this if there is a disagreement between monitors or monitors and a CA? Is it the issuing CA? If so, this will not stop a compelled certificate creation attack since it is the subject that has to say which certificate is false and which is not, since both were issued by the same CA or its subordinate.

3. Status Acquisition Mechanisms

3.1. CRLSets

[Working on getting a citable description]

(Only supported in the Chrome Browser)

3.2. SCVP

Not supported in any Web PKI application or service.

3.3. XKMS

Not supported in any Web PKI application or service.

4. Cryptography Platforms (to be completed once the survey is finished)

[Should expand this noting that while support for a feature in the platfor is often a necessary for support in applications, it is not necessarily sufficient.]

4.1. Checklist

4.2. cryptlib

4.3. Microsoft Windows

4.4. Network Security Services

Used in Firefox and Chrome

4.5. OpenSSL

5. Web Server Status (TBC once the survey is finished)

Web Server support for revocation is fairly straightforward since the Web Server is only involved in revocation in the case of stapled OCSP tokens and this is supported in the latest versions of all the servers surveyed.

5.1. Checklist

Support for OCSP Stapling?

5.2. Apache

Support for OCSP Stapling?
Since version 2.3.

5.3. IIS

Support for OCSP Stapling?
Yes

5.4. LiteSpeed

Support for OCSP Stapling?
Since version 4.2.4.

5.5. nginx

Support for OCSP Stapling?
Since version 1.3.7.

6. Web Client Status (TBC once the survey is finished)

[Need to consider further dimensions here. In particular Chrome behaves differently depending on the platform it is on and several browsers have different revocation checking for EV vs other certificate policies.]

6.1. Checklist

Supported Revocation Checking Mechanisms
To Do: Some browsers do not support CRL DP. Others give preference to OCSP, but fall back to CRL DP if the necessary AIA is missing. Some browsers give priority to OCSP, but switch to CRL DP when a particular issuer's revocation information is retrieved frequently.

User Experience for Certificate Status Invalid

User Experience for Certificate Status Unknown

What sources are permitted to sign CRLs or OCSP responses for a certificate, can any trusted CA sign or only the CA that issued the certificate?

6.2. Chrome

The Chrome browser automatically updates itself to the latest version unless this feature is explicitly disabled by the user.

Supported Revocation Checking Mechanisms
OCSP (at present).

User Experience for Certificate Status Invalid

User Experience for OCSP Fail
Soft fail

6.3. Firefox

Supported Revocation Checking Mechanisms
OCSP checked by default since Firefox 3. OCSP Stapling has been added to nightly builds but is not yet in production releases.

User Experience for Certificate Status Invalid

User Experience for Certificate Status Unknown
Soft fail

6.4. Internet Explorer

Supported Revocation Checking Mechanisms
CRLs and OCSP.

User Experience for Certificate Status Invalid
IE5: If the certificate has a valid trust path, the user is presented with a dialog box that says "The Security certificate for this site has been revoked. This site should not be trusted"

If however, a valid trust path cannot be found, the user receives the error message for the invalid trust path instead: "The user is presented with a dialogue box that tells them that 'Information you exchange with this site cannot be viewed or changed by others. However there is a problem with the site's security certificate."

IE??: Warning Page "There is a problem with this website's security certificate"

CVE-2011-0199 : Chris Hawk and Wan-Teh Chang of Google MS ? ?the revocation date is determined by comparing the current date with the RevocationDate field in the CRL or the OCSP response? For Windows Vista with Service Pack 1 and Windows Server 2008, the OCSP signing certificate may chain up to any trusted root CA as long as the certificate chain includes the OCSP Signing EKU extension. CryptoAPI first determines whether a time valid version of the revocation object exists in the CryptoAPI disk cache.

6.5. Opera

Supported Revocation Checking Mechanisms
OCSP checked by default since Opera version 8

User Experience for Certificate Status Invalid

User Experience for Certificate Status Unknown
Hard fail

6.6. Safari

Supported Revocation Checking Mechanisms
CRL: all, OCSP checking is enabled by default as of Mac OSX 10.7 (Lion). Prior to that it had to be enabled in the Keychain

preferences.

User Experience for Certificate Status Invalid
Dialog Box: Offers Continue/Cancel/Show Certificate

User Experience for Certificate Status Unknown

User Experience for OCSP-FAIL

For Apple?s OS X, OCSP and CRL checking can be configured via Keychain Access -> Preferences ?> Certificates. A dialog box opens with three rows: OCSP, CRL, and Priority. Under OCSP and CRL, the three allowed options (a grayed-out one said something like ?always?) were: ?Off?, ?Best Attempt?, and ?Required if certificate indicates?. ?Best Attempt? was the default. Under Priority, the options were ?OCSP? ?CRL? and ?Require both?. ?OCSP? was the default.

IE5: Popup dialog box: "Revocation information for the security certificate for this site is not available. Do you want to proceed?"

7. CA 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.

7.1. Checklist

Are CRLs or OCSP supported

Is the CDP extension filled?

Is the AIA extension filled?

7.2. CA-Browser Forum Requirements

8. Security Considerations

Put something here?

9. IANA Considerations

None

10. References

10.1. Normative References

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

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

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

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

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

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

Authors' Addresses

Phillip Hallam-Baker
Comodo Group Inc.

philliph@comodo.com

David Chadwick
University of Kent

d.w.chadwick@kent.ac.uk