Certificate Limitation Policy

draft-belyavskiy-certificate-limitation-policy-07

Abstract

The document provides a specification of the application-level trust model. Being provided at the application level, the limitations of trust can be distributed separately using cryptographically protected format instead of hardcoding the checks into the application itself.

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 https://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."

This Internet-Draft will expire on May 29, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

Binary trust model standardized as a set of trusted anchors and CRLs/OCSP services does not cover all corner cases in the modern crypto world. There is a need in more differentiated limitations. Some of them are suggested [1] by Google when it limits the usage of Symantec’s certificates. The CRL profile does not fit the purpose of such limitations. The CRLs are issued by the same CAs that are subject to be limited.

Currently the set of CAs trusted by OS or browsers can be used for the validation purposes. In case when a large enough CA becomes untrusted, it cannot be deleted from the storage of trusted CAs because it may cause error of validation of many certificates. The measures usually taken in such cases usually include application-level limitation of certificates lifetimes, refusing to accept EV-certificates in other way than DV, requirements to use Certificate Transparency, etc.

This document suggests a cryptographically signed format dubbed Certificate Limitation Profile (CLP) designed for description of such limitations. This format can be used by applications that use system-wide set of trust anchors for validating purposes or by applications with own wide enough set of trusted anchors in case when the trust anchor for the entity found misbehaving cannot be revoked.

Currently the only way to provide such limitations is hard coding them in application itself. Using of CLPs does not allow to completely avoid hard coding but allows to hard code only the minimal set of rarely changing data:

- the fact that application uses CLP
- the certificate to verify the signature under the CLP file
- minimal date of the CLP to be used for the current version of application.

It will be possible to move the checks for the limitations to the external cryptographical libraries, such as OpenSSL, instead of checking them at the application level.

2. Certificate Limitations Profile

A proposed syntax and overall structure of CLP is very similar to the one defined for CRLs [2].
CertificateList ::= SEQUENCE {
  tbsCertList     TBSCertList,
  signatureAlgorithm AlgorithmIdentifier,
  signatureValue   BIT STRING }

TBSCertList ::= SEQUENCE {
  version         Version,
  signature       AlgorithmIdentifier,
  issuer          Name,
  thisUpdate      Time,
  nextUpdate      Time,
  limitedCertificates  SEQUENCE OF SEQUENCE {
    userCertificate CertificateSerialNumber,
    certificateIssuer  Name,
    limitationDate     Time,
    limitationPropagation  Enum,
    fingerprint  SEQUENCE {
      fingerprintAlgorithm AlgorithmIdentifier,
      fingerprintValue    OCTET STRING 
    } OPTIONAL,
    limitations  Limitations,
  } OPTIONAL,
};

Limitations ::= SEQUENCE SIZE (1..MAX) OF Limitation

Limitation ::= SEQUENCE {
  limitID        OBJECT IDENTIFIER,
  LimitationValue OCTET STRING
  -- contains the DER encoding of an ASN.1 value
  -- corresponding to the limitation type
  -- identified by limitID
}
TBSCertList – the sequence of individual certificates to be limited;

signatureAlgorithm – the OID of the signature algorithm used for signature;

signatureValue – the bit string representing signature of the TBSCertList.

2.2. CLP signature

The key used for signing the CLP files should have a special Key Usage value and/or an Extended Key Usage value.

2.3. CLP entry fields

Each entry in list contains the following fields:

The issuer of the certificate with limited trust.

The serial of the certificate with limited trust.

The fingerprint of the certificate with limited trust (optional).

and a subset of the following limitations:

issuedNotAfter – do not trust the certs issued after the specified date

trustNotAfter – do not trust the certs after the specified date

validityPeriod, days – take minimal value from "native" validity period and specified in the limitation file

ignoredX509Extensions – list of X.509 extensions of limited certificate that MUST be ignored for the specified certificate (e.g. EV-indicating extensions)

requiredX509Extensions – list of X.509 extensions that MUST be present in the certificate to be trusted.

requiredNativeChecking – list of the CA-provided checks that MUST be applied

applicationNameConstraints – list of domains allowed to be issued by this certificate
excludedIssueIntermediary - disallow issuing of the Intermediary certificates

The limitations are identified by OIDs

2.3.1. Limitations

2.3.1.1. issuedNotAfter

When this limitation is present, any certificate matching the entry and issued after the specified date MUST NOT be trusted.

The issuedNotAfter limitation is identified by OID TBA.

issuedNotAfter ::= SEQUENCE {
    IssuedNotAfter    Time
}

2.3.1.2. trustNotAfter

When this limitation is present, any certificate matching the entry MUST NOT be trusted after the specified date.

The trustNotAfter limitation is identified by OID TBA.

trustNotAfter ::= SEQUENCE {
    TrustNotAfter    Time
}

2.3.1.3. validityPeriod

When this limitation is present, no certificate matching the entry should be treated as valid after specified period from its validFrom.

The validityPeriod is measured in days.

The validityPeriod limitation is identified by OID TBA.

validityPeriod ::= SEQUENCE {
    Days INTEGER
}

2.3.1.4. ignoredX509Extensions

When this limitation is present, the extensions listed in this element should be ignored for the matching certificate.

The ignoredX509Extensions limitation is identified by OID TBA.
ignoredX509Extensions ::= SEQUENCE SIZE (1..MAX) OF ExtenID
ExtenID ::= OBJECT IDENTIFIER

2.3.1.5.  requiredX509extensions

When this limitation is present, the extensions listed in this element should be present for the matching certificate.

The requiredX509extensions limitation is identified by OID TBA.

requiredX509extensions ::= SEQUENCE SIZE (1..MAX) OF ExtenID
ExtenID ::= OBJECT IDENTIFIER

2.3.1.6.  requiredNativeChecking

When this limitation is present, it specifies that the certificates issued by this CA SHOULD be checked against CRL and/or OCSP, depending on contents of the extension.

The requiredNativeChecking limitation is identified by OID TBA.

requiredNativeChecking ::= SEQUENCE {
  RequiredCRLChecking BOOLEAN,
  RequiredOCSPChecking BOOLEAN
}

2.3.1.7.  applicationNameConstraints

This limitation are applied like Name Constraints [3] limitation specified in RFC 5280.

This section implies 2 variants of checks:

- The list of names that are allowed for the CA to issue certificates for
- The list of names that are forbidden for the CA to issue certificates for

The applicationNameConstraints limitation is specified according to RFC 5280, 4.2.1.10 and reuses OID specified in RFC 5280.
id-ce-nameConstraints OBJECT IDENTIFIER ::=  { id-ce 30 }

NameConstraints ::= SEQUENCE {
    permittedSubtrees       [0]     GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]     GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                    GeneralName,
    minimum         [0]     BaseDistance DEFAULT 0,
    maximum         [1]     BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)

2.3.1.7.1.  excludedIssueIntermediary

When this limitation is present, the intermediate certificates issued by this CA MUST NOT be trusted.

The excludedIssueIntermediary limitation is identified by OID TBA.

3. Verification of CLP

The verification of CLP SHOULD be performed by the application. The application should check whether the provided CLP matches the internal requirements and is correctly signed by the specified key.

4. Verification with CLP

In case of using CLP the checks enforced by CLP should be applied after the other checks.

The limitation provided by CLP MUST NOT extend the trustworthy of the checked certificate.

The limitations are applied after cryptographic validation of the certificate and during building its chain of trust. If the certificate or any of its ascendants in the chain of trust matches any record in the CLP, the limitations are applied from the ascendant to descendants. The issuedNotAfter and trustNotAfter limitations are applied to find out the actual validity periods for the any certificate in the chain of trust. If the CLP prescribes to have a particular extension(s) and the certificate does not have it, the certificate MUST NOT be trusted.
Application MAY use more than one CLPs (e.g. app-wide, set of system-wide, user-defined). When multiple CLPs are in use, the limitations are applied simultaneously.

In case when more than one chain of trust are valid for a certificate, if any of this chains is valid after applying the limitations, the certificate MUST be treated as valid.

5. ASN.1 notation

TBD

6. Real-world considerations

6.1. Expected sources and consumers of CLPs

Public CLPs can be created and distributed by such parties as OS vendors, browser vendors and other parties treated as worth trusting.

Usage of CLPs is reasonable for applications establishing TLS connections with unpredictable sets of peers. The main examples of such applications are web-browsers and MTAs.

6.2. Size limitation

To avoid uncontrolled growth of CLPs, the limitations are applied to root and intermediate CA certificates.

7. Security considerations

In case when an application uses CLP, it is recommended to specify the minimal date of issuing of the CLP document somewhere in code. It allows to avoid an attack of CLP rollback when the stale version of CLP is used.

It is recommended to distribute CLPs using the channels that are used for distribution of the applications themselves to avoid possible DoS consequences.

If application checks for fresh CLPs, it SHOULD check that nextUpdate field in a fresh one is newer than in the current one. The application MAY accept a CLP with nextUpdate in past. If an application is failing to get updates, then it can continue to run with what it has.
7.1. Unsigned CLP

In case of trusted environment signing CLP can be reluctant. If CLP is delivered via application bundle, it can be verified together with other application data. But it makes sense to separate trust to the source of the content from trust to the content itself. On the other hand it is not a problem to create a local CLP signed by a locally created key.

8. IANA considerations

TBD

9. Acknowledgements


10. References

The current version of the document is available on GitHub
https://github.com/beldmit/clp

11. References

11.1. URIs

[1] https://groups.google.com/a/chromium.org/forum/#!msg/blink-dev/eUAKwjihhBs/rpxMXjZHCQAJ


Author’s Address

Dmitry Belyavskiy
Technical Centre of Internet
8 Marta str., 1 bld. 12
Moscow 127083
RU

Email: beldmit@gmail.com