LAMPS
Internet-Draft
Intended status: Informational
Expires: 18 December 2025

P. Liu, Ed.
Pengcheng Laboratory
R. Fu, Ed.
China Unicom
R. Chen, Ed.
China Mobile
X. Liu, Ed.
Y. Zhang, Ed.
Pengcheng Laboratory
16 June 2025

Simple Local Web PKI Certificate Resource Preservation Management for Internet Browser
draft-liu-lamps-browser-webpki-cert-preservation-06

Abstract

The management of Web PKI certificate resources presents a challenge when the misalignment of ownership and management rights over certificate resources of one organization creating a risk of unilateral suspension and revocation by another competing organizations. This situation undermines the stability of critical infrastructure and affects the integrity of authentication systems. To address these concerns, this document proposes a mechanism that allows Internet browsers to create a customized management view of certificate resources, enabling them to override the verification results from specific certification authorities as needed. This approach enhances security, facilitates stakeholder collaboration, and preserves the operational integrity of foundational industry systems.

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 18 December 2025.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The current SSL/TLS certificate architecture faces challenges due to the increasing centralization of major certificate authorities (CAs), which are controlled by a few institutions. Many critical systems worldwide depend on these centralized CAs, making their verification processes vulnerable to external influences from OCSP, CRL, and CT systems that may have competing interests. This reliance can lead to malicious revocation of certificates, rendering important industry services inaccessible and impacting the foundational economy of a nation.

The misalignment of ownership and management rights over certificate resources of one organization (can be a nation or a large group) creates a risk of unilateral suspension and revocation by another competing organizations (can be also a nation or a large group).

Addressing this issue requires developing a certificate resource management technology that is compatible with existing authentication systems while allowing organizations to replace the certificate verification results provided by certain certification authority CAs when necessary. Currently, no specific standards exist for these scenarios, some Internet browsers may provide related configurations that ignore all certificate errors or are similar to whitelists. However, generally ignoring all SSL/TLS certificate verification errors is considered unsecure and poses serious security risks.

This document proposes a straightforward mechanism for Internet browsers to create a local customized management view of Web PKI certificate resources. It enables organizations to overwrite certificate data or verification results from certain CAs when needed, thus retaining control over their critical certificates. By implementing this local preservation mechanism, organizations can mitigate the risks associated with malicious revocation or the failure to reissue expired certificates. Users can independently assess the validity of certificates, ensuring the stable operation of essential systems and enhancing the overall network security across various industries.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Design Goals

The local Web PKI certificate resource preservation mechanism aims to achieve two main goals.

1. After the basic certificate verification process defined in RFC5280 is completed, if a specified error occurs, such as a certificate being revoked or expired, the organizational user (can be nation or a large group) can autonomously decide which certificates are valid.

2. As needed, organizational user can define their own untrusted certificates, regardless of whether they are verified as legitimate or not during the standard verification process.

4. The local Web PKI certificate resource preservation mechanism based on Internet browser

4.1. Local Web PKI Certificate Resource Preservation Process in Internet Browser Certificate Verification

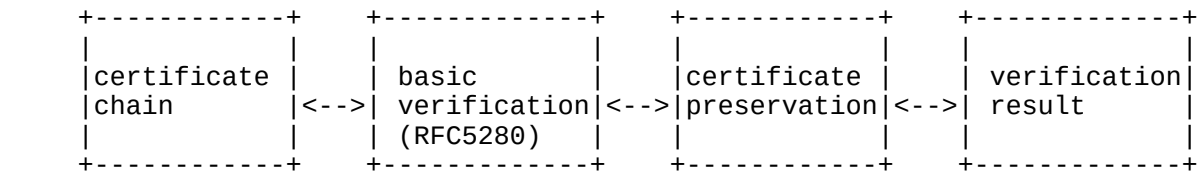

Figure 1: the certificate preservation process based on Internet Browser

The process of Web PKI certificate resource preservation based on Internet Browser is shown in Figure 1. After completing basic certificate verification following [RFC5280] in the browser, the browser uses the "LocalCertWhiteFilters" defined in section 4.2 to verify whether the serial number "SerialNumber" of the target leaf certificate exists in the whitelist certificate filter field "CertWhiteFilters" of the current local certificate preservation repository, or whether the subject name "subjectName" of the target leaf certificate exists in the whitelist certificate filter field "CertWhiteFilters" of the current local certificate preservation repository, or if the subject alternative name extension "subjectAltName" of the target leaf certificate exists, to verify whether the subject alternative name exists in the whitelist certificate filter item "WhiteCertFilters" of the current local certificate preservation repository. As long as one of the above conditions is met, it is matched to the whitelist certificate filter field "CertWhiteFilters" of the current local certificate preservation repository. If a specific error number "ErrorNo" is found during the basic certificate verification process, such as the target leaf certificate being revoked or expired, the current target leaf certificate should be considered valid.

According to the definition of "LocalCertBlackAssertions" in Section 4.2 of the local certificate preservation repository, verify whether the serial number "SerialNumber" of the target leaf certificate exists in the blacklist certificate assertion field "CertBlackAssertions" of the current local certificate preservation repository, or verify whether the subject name "subjectName" of the target leaf certificate exists in the blacklist certificate assertion field "CertBlackAssertions" of the current local certificate preservation repository, or if the subject alternative name extension of the target leaf certificate exists, verify whether the "subjectAltName" exists in the blacklist certificate assertion field "CertBlackAssertions" of the current local certificate preservation

repository, or verify whether the issuer name "Issuer" exists in the current local certificate preservation repository’s blacklist certificate assertion field "CertBlackAssertions", Alternatively, if the issuer alternative name extension of the target leaf certificate exists, verify whether the "issuerAltName" exists in the current local certificate preservation repository’s blacklist certificate assertion field "CertBlackAssertions". As long as one of the conditions is met, regardless of whether any errors occur during the basic certificate verification process, the current target leaf certificate should be considered invalid and the browser user should be given a corresponding network certificate status prompt.

4.2. Local certificate preservation repository

This mechanism defines and introduces the format of the certificate preservation repository based on JSON-formated file. The local Web PKI certificate resource preservation mechanism or service in the certificate verification process of Internet browsers that comply with this mechanism shall comply with the format specification defined in this section. This mechanism uses ASN. 1 Specific Encoding Rules (DER) to encode the basic fields of the following certificate parameter items and form the specific certificate parameter data structures. ASN. 1 DER encoding is an encoding system for the tag, length, and value of each element. For certificate preservation repository based on JSON-formated file, its local whitelist certificate filters "LocalCertWhiteFilters" and blacklist certificate assertions "LocalCertBlackAssertions" members are specified in JSON format [RFC8259]. JSON member not defined here is not allowed to be used in certificate preservation repository file. The implementation of Internet browsers must treat any deviation from this specification as an error. For the functional version update related to this specification in this document, this mechanism increments the "Version" field in the JSON file to indicate the version and functional differences of the local certificate preservation repository.

The certificate preservation repository file format based on JSON format is as follows:

```
{
   “Version” : 1,
   “LocalCertWhiteFilters” : {
     “ErrorNo” : ERR_CERT_REVOKED,
     “CertWhiteFilters” : [],
  },
   “LocalCertBlackAssertions” : {
      “CertBlackAssertions” : [],
  }
}
```

The "Version" member is set to 1 and encoded as a number in this standard version.

For the local whitelist certificate filter member "LocalCertWhiteFilters" in the certificate preservation repository file, Internet browser users can configure zero or more whitelist certificate filter entries "CertWhiteFilters" that have passed the basic certificate verification logic, where each whitelist certificate filter entry in "CertWhiteFilters" can include the "SerialNumber" of the certificate, the "SerialNumber" of the certificate, and/or the "subjectName" of the certificate. This mechanism suggests that each whitelist certificate filter entry should contain an explanatory note "comment", so as to explain the relevant matching information to Internet browser users as much as possible and facilitate their understanding.

In file format, the content of the local whitelist certificate filter member "LocalCertWhiteFilters" is represented as the values of the members "ErrorNo" and "CertWhiteFilters". The member "ErrorNo" describes the type of certificate error that occurs in the basic certificate verification logic, which is a positive integer with a value of a number in JSON format. Its default value is ERR_CERT_REVOKED (value 201, the types and value definitions of basic certificate errors defined in this mechanism can be found in Appendix A). Any algorithm implementation that complies with this mechanism can extend and define its own error types. The member "CertWhiteFilters" is represented by an array of zero or more objects, each entry must contain at least one of the following members, or a combination of these members.

1. "serialNumber": the serial number, is of type INTEGER, and in JSON format, its value is a number.

2. "subjectName": the subject name, is of type Name. In JSON format, its value is the Base64 encoding of the certificate's subject name, without the trailing character '='. This means that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

3. "subjectAltName": the alternative name for the subject, is of type GeneralNames. In JSON format, its value is the Base64 encoding of the certificate's subject alternative name, with no trailing character '=', meaning that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

4. "comment": a configuration item related annotation, whose value is a string in JSON format, used to explain the annotation of this configuration item.

For the local blacklist certificate assertion member "LocalCertBlackAssertions" in the certificate security repository file, Internet browser users can configure zero or more blacklist certificate assertion device entries "CertBlackAssertions" that have passed the basic certificate verification logic, where each blacklist certificate assertion entry "CertBlackAssertions" can include the "SerialNumber" of the certificate, the "subjectName" of the certificate, the "subjectAltName" of the certificate, the "issuer" name of the certificate, and/or the issuer alternative name "issuerAltName" of the certificate. This mechanism recommends that each blacklist certificate assertion entry contain an explanatory not "comment", so as to explain relevant matching information to Internet browser users as much as possible and facilitate users' understanding.

In terms of format, the content of "LocalCertBlackAssertions", a local blacklist certificate assertion, is represented as the value of a "CertBlackAssertions" member, which is an array of zero or more objects. Each entry must contain at least one of the following members, or a combination of the following members.

1. "serialNumber": the serial number, is of type INTEGER, and in JSON format, its value is a number.

2. "subjectName": the subject name, is of type Name. In JSON format, its value is the Base64 encoding of the certificate's subject name, without the trailing character '='. This means that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

3. "subjectAltName": the alternative name for the subject, is of type GeneralNames. In JSON format, its value is the Base64 encoding of the certificate's subject alternative name, with no trailing character '=', meaning that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

4. "issuerName": the issuer name, is of type Name, and in JSON format, its value is the Base64 encoding of the certificate issuer's name, without the trailing character '=', meaning that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

5. "issuerAltName": the alternative name for the issuer, is of type GeneralNames, with field formats specified in RFC 5280. In JSON format, its value is the Base64 encoding of the certificate issuer's alternative name, followed by no trailing character '=', meaning that the value of this member is an ASN. 1 byte string (OCTET STRING) without an ASN. 1 tag and length field.

6. "comment": a configuration item related annotation, whose value is a string in JSON format, used to explain the annotation of this configuration item.

It should be noted that in actual systems, multiple local certificate preservation repository files based on JSON format may be supported for simultaneous use. To ensure consistency in local functionality, the configuration of multiple local certificate preservation repository files must ensure that there are no conflicts. In other words, the Internet browser should check the local whitelist certificate filter field "LocalCertWhiteFilters" and the local blacklist certificate assertion field "LocalCertBlackAssertions" in each local certificate preservation repository file to ensure that there are no overlapping configuration entries, and also ensure that there are no overlapping configuration entries in the local whitelist certificate filter field "LocalCertWhiteFilters" and the local blacklist certificate assertion field "LocalCertBlackAssertions" between different local certificate preservation repository files. If a conflict is detected, an error should be reported to the user who created the relevant local certificate preservation repository file. Internet browsers should de-duplicate multiple local certificate preservation repository files as a collection. If there is any overlap between the local whitelist certificate filter field "LocalCertWhiteFilters" and the local blacklist certificate assertion field "LocalCertBlackAssertions" between the local certificate preservation repository files, the entire collection must be rejected.

The summary of all members and types supported by the JSON-formated local certificate preservation repository file in this mechanism is as follows:

```
{
   “Version” : Type Int,
   “LocalCertWhiteFilters” : {
     “ErrorNo” : Type Int,
     “CertWhiteFilters” [
      {
         “serialNumber” : Type Int,
         “subjectName” : "<Base 64 of some subjectName>",
         “subjectAltName” : "<Base 64 of some subjectAltName>",
         “comment” : Type String,
      }
    ],
 },
  “LocalCertBlackAssertions” : {
   “CertBlackAssertions” : [
    {
       “serialNumber” : Type Int,
       “subjectName” : "<Base 64 of some subjectName>",
       “subjectAltName” : "<Base 64 of some subjectAltName>",
       “issuerName” : "<Base 64 of some issuerName>",
       “issuerAltName” : "<Base 64 of some issuerAltName>",
       “comment” : Type String,
    }
  ],
 }
}
```

5. Appendix A

The meaning of the following errors can be literally seen from common browsers.

1. ERR_CERT_INVALID: 201, The certificate is invalid, such as incorrect format, unsupported fields, etc.

2. ERR_SSL_PINNED_KEY_NOT_IN_CERT_CHAIN: 202, The certificate used on the website does not match the HTTP public key of the built-in certificate, which may cause the website to be hijacked. It is necessary to check the website’s DNS analysis to restore normal HTTPS access; It is also possible that the HPKP Google Chrome error is due to incorrect settings.

3. ERR_CERT_REVOKED: 203, The certificate used by the website has been revoked. The certificate issuing authority has revoked the certificate due to changes in enterprise information or violations of website content. The certificate has been added to the certificate revocation list CRL. We need to reapply for the certificate and deploy it correctly.

4. ERR_CERT_AUTHORITY_INVALID: 204, The website is using a certificate issued by an invalid certificate authority. This error indicates that the root certificate of the certificate used by the website is not trusted by the browser, which may be due to the user using a self-signed certificate or the root certificate of the certificate being revoked. The solution is to reapply for a certificate issued by a certificate authority trusted by the browser.

5. ERR_CERT_COMMON_NAME_INVALID: 205, The certificate used by the website does not match the domain name, and the domain name supported by the certificate does not match the website domain name. In other words, the website used the wrong certificate. The solution is to reapply for a new SSL certificate with the same domain name as the website.

6. ERR_CERT_WEAK_SIGNATURE_ALGORITHM: 206, The website certificate uses an insecure signature algorithm, and the digital signature algorithm is used for identity verification between communication parties. If the insecure SHA-1 signature algorithm is used, the browser will report an error, and the SHA-256 signature algorithm should be used.

7. ERR_CERT_DATE_INVALID: 207, SSL certificate has expired. The certificate has expired and been deleted. Applying for a new certificate and installing it correctly can solve the error.

8. ERR_CERT_VALIDITY_TOO_LONG: 208, The validity period of the website certificate is too long. As time goes by, the longest lifespan of public trust certificates gradually shortens. According to the Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates after version V2.0.8, SSL/TLS Certificates issued before 15 March 2026 SHOULD NOT have a Validity Period greater than 397 days and MUST NOT have a Validity Period greater than 398 days; SSL/TLS Certificates issued on or after 15 March 2026 and before 15 March 2027 SHOULD NOT have a Validity Period greater than 199 days and MUST NOT have a Validity Period greater than 200 days; SSL/TLS Certificates issued on or after 15 March 2027 and before 15 March 2029 SHOULD NOT have a Validity Period greater than 99 days and MUST NOT have a Validity Period greater than 100 days;

SSL/TLS Certificates issued on or after 15 March 2029 SHOULD NOT have a Validity Period greater than 46 days and MUST NOT have a Validity Period greater than 47 days. For the purpose of calculations, a day is measured as 86,400 seconds. Any amount of time greater than this, including fractional seconds and/or leap seconds, shall represent an additional day. For this reason, SSL/TLS Certificates SHOULD NOT be issued for the maximum permissible time by default, in order to account for such adjustments.

9. ERR_CERT_NO_REVOCATION_MECHANISM: 209, The certificate does not have a revocation mechanism, meaning there is no CRL or OCSP reference. This error may pose a security risk as the browser cannot verify whether the certificate has been revoked. Mainly, some CAs issue short-term certificates, so they do not provide a method to revoke them. Some intermediate boxes (such as Palo Alto Networks or Cisco firewalls) or antivirus software on computers are used on the network, which will intercept HTTPS and replace certificates with their own certificates.

10. ERR_CERT_UNABLE_TO_CHECK_REVOCATION: 210, If the local browser has enabled the Required Online Revocation ChecksForLocalAnchors policy and the CRL of the website certificate does not comply with the requirements of RFC 5280.

11. ERR_CERTIFICATE_TRANSPARENCY_REQUIRED: 211, It is specific to Google Chrome, mainly because the website certificate has not yet been added to the transparency log by the issuing authority. There are two situations where the certificate is not added to the transparent log. The first scenario is that the issuing authority did not add the certificate, possibly due to their negligence. The second scenario is that the website owner may have requested the certificate authority not to add their certificate to the log.

12. ERR_CERT_SYMANTEC_LEGACY: 212, Google Chrome no longer trusts Symantec certificates issued before June 1, 2016. Symantec has sold its CA business to Digicol and no longer issues SSL certificates, so the likelihood of encountering this error is very low.

13. ERR_CERT_NAME_CONSTRAINT_VIOLATION: 213, The domain name of the certificate does not comply with the rules and violates the name constraints in the certificate. This error may occur when the name constraint of the certificate is not met, which may be due to configuration errors or unauthorized use.

14. ERR_CERT_KNOWN_INTERCEPTION_BLOCKED: 214, This error indicates that a known SSL/TLS interception attempt has been blocked. When a third party attempts to intercept a secure connection, this situation may occur, endangering the user's privacy and security. You need to investigate the potential security software or network configuration that caused this interception and ensure a secure connection.

15. ERR_CERT_WEAK_KEY: 215, It indicates that using weak keys for encryption in certificates poses a security risk. This vulnerability may expose users to potential security risks. To address this error, website administrators need to update their certificates with more powerful key algorithms to improve security.

16. ERR_CERT_STATUS_NON_UNIQUE_NAME: 216, When the website certificate is missing a unique theme name, it indicates that it does not have a unique name and is generally not mapped to an error. It is considered a warning to downgrade SSL UI.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This mechanism defines the reference specification for the Internet browser to locally preserve and manage the customized Web PKI certificate resources, and provides a simple mechanism to enable the Internet browser (in its implementation form, it may also be a browser proxy) to establish a local customized management view of the Web PKI certificate resources, and to overwrite the certificate data or verification results published by certain certification authority CAs when necessary. This mechanism addresses the security issues of domain name certificate resources in network infrastructure, namely the risk of unilateral suspension and revocation of certificate ownership due to the mismatch between ownership and management rights of certificate resources; Cleverly resolving the contradiction between unity and autonomy, key infrastructure improvement and stability, compatible with the contradiction between existing and smooth substitution, compatible with existing authentication systems, enabling stakeholders in the network to smoothly replace existing authentication, cope with the impact of malicious revocation of important industry certificates, and ensure the safe and normal operation of important industry systems. For this reason, Internet browsers (relying parties or their agents) conforming to this mechanism can autonomously decide and process any certificate and its verification results asserted by the local certificate preservation

database according to local management requirements. This mechanism is applicable to the implementation and application of the Internet browser certificate resource preservation system based on Web PKI, and it is applicable to ensuring the smooth operation of the secure network access related to the business of an organization, without being subject to the management and control of other organizations that may have competitive interests; The Internet browser local certificate preservation specifications defined in this section are universal, and can also be applied to other similar network security applications and environments of different types based on PKI mechanism.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[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, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/rfc/rfc5280>.

Authors' Addresses

Penghui Liu (editor)
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: liuph@pcl.ac.cn

Yu Fu (editor)
China Unicom
No.1 Zhonghe Street
Beijing
100000
China
Email: fuy186@chinaunicom.cn

Meiling Chen (editor)
China Mobile
No.32 Xuanwumen West Street
Beijing
100000
China
Email: chenmeiling@chinamobile.com

Xiang Liu (editor)
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: liux15@pcl.ac.cn

Yu Zhang (editor)
Pengcheng Laboratory
No.2 Xingke 1 Street
Shenzhen
518055
China
Email: zhangy08@pcl.ac.cn