Internet Engineering Task Force
Internet-Draft
Intended status: Best Current Practice
Expires: October 31, 2015

I. Barreira, Ed.
Izenpe
B. Morton, Ed.
Entrust

April 29, 2015

Trust models of the Web PKI
draft-ietf-wpkops-trustmodel-04

Abstract

This is one of a set of documents to define the operation of the Web PKI. It describes the currently deployed Web PKI trust.

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."

This Internet-Draft will expire on October 31, 2015.

Copyright Notice

Table of Contents

1. Introduction

This document defines the Web PKI trust model as it is currently implemented. The trust model is to support communications between the subscriber and the browser. It refers also to the current processing behaviours of cryptolibraries with the browsers they support, related to the communication between the server and the browser as indicated in the "Browser processing of server certificates" document. This document does not address future changes to the implemented trust model.

1.1. Requirements Language

The key words "REQUIRED", "MUST", "MUST NOT" and "MAY" in this document are to be interpreted as described in RFC 2119 [RFC2119]

1.2. Definitions

The use of PKI terminology is used as defined in RFC 5280 [RFC5280]. Additional definitions are provided below for interpretation of this document.

Certificate policy - per RFC 3647. [RFC3647]

Intermediate CA - is a non-root CA which issues certificates to issuing CAs.

Issuing CA - in relation to a particular subscriber certificate, the CA that issued the certificate.

Root CA - a CA with a self signed certificate and whose public key is included as a trust anchor in a root certificates store.

Root certificate - typically a self-signed certificate that identifies the root CA. The root certificate is a type of trust anchor.

Root certificates store - a set of root certificates which can be trusted by the operating system and/or a browser. Within the context of the present document the more general term Root Store is used in preference.

Root store policy - the governance policy provided by the root store provider.

Subscriber - per RFC 3647. [RFC3647]

Subscriber agreement - per RFC 3647. [RFC3647]

Trust Anchor - per RFC 5914. [RFC5914]

2. Trust model

This section describes the basic Web PKI trust model. Variants to the trust model are discussed in section 3.

In the Web PKI trust model, a browser uses a root store that contains one or more root CA public keys. Each entry in a browser's root store has been installed on an evaluation made by the browser vendor. Each root CA issues a certificate to one or more issuing CAs that are under the control of the same CA entity with the variant stated in 3.2.2. Each issuing CA accepts and responds to certificate requests from one or more subscribers via one or more registration authorities.

2.1. Root store provider

A root store provider (e.g., Microsoft or Mozilla) determines a root store policy. This policy must be met by a candidate root CA in order to be included in the root store. The root store provider installs and manages root certificates in its operating system or browser to support certificate chain validation. The root store provider establishes requirements for accepting a root certificate. These requirements may include legal agreements, security or audit reports or acceptance by another root store provider.

The root store provider may require the root CA to be subject to an annual compliance audit performed by a third party auditor as specified in [BR-certs]. The audit requirements are defined by the root store policy. The audit is based on an accepted schema of the standards (e.g., WebTrust or ETSI). A third party auditor generates an audit report which is provided to the root store provider. If the audit report states the root CA did not comply with the auditing standards, then the root CA will be required to take corrective actions. Once the corrective actions are completed, then an updated report is submitted to the root store provider. If the status of the root CA is not acceptable to the root store provider, then the root CA certificates may be removed from the root store or the indications from the browser (e.g., removal of https indicator) may change for certificates verified under that root CA.

2.2. CA Infrastructure

The CA infrastructure consists of a PKI hierarchy. Each organization acting as a CA entity is represented by one or more self-signed root certificates. The root CAs sign certificates for subordinate issuing CAs. The root CA may have subordinate intermediate CAs to manage

groups of subordinate issuing CAs. The CA entity manages root, intermediate, and issuing CAs and oversees operation of the certificate issuance and management system in accordance with a certificate policy.

2.2.1. Registration Authority

Each issuing CA operates a registration authority, with variations in 3.2.3 and 3.2.7, which authenticates requests for certificates in accordance with the certificate policy of the CA.

2.2.2. Certificate status

Each CA provides certificate status in the form of a certificate revocation list (CRL) and/or an on-line certificate status protocol (OCSP) response. Updates and validity periods of the certificate status are provided in accordance with the certificate policy of the CA. The location of the CRL is provided in the certificate CDP (CRL Distribution Point) OID and the location of the OCSP response is provided in the AIA (Authority Info Access) OID of the issued certificate.

2.3. Subscriber

Each subscriber provides services through the browsers to relying parties. The subscriber identifies the on-line web location of its service using a domain name or IP address contained in a certificate.

The subscriber submits certificate requests in accordance with a CA's certificate policy. Once the certificate request has been accepted, the subscriber will receive the certificate and will manage the certificate in accordance with the subscriber agreement.

2.4. Browser

The browser accepts and manages certificates and performs related functions in accordance with the root store policy (e.g., [Mozilla-CP]).

3. Trust Model variants

This section defines variants to the roles of the parties as defined in section 2.

3.1. Root store provider variations

3.1.1. Browser adopts root store

The browser does not use its own root store, but uses the root store managed by a separate root store provider. For example, the Google Chrome browser operated on Windows uses the Windows root store.

The browser will provide its own trust and security indications. The browser may determine whether it will provide additional validation indications. The browser may also provide its own services to verify the status of the certificates.

3.2. CA Infrastructure variations

3.2.1. One root CA cross-certifies another root CA

Some browsers in active use do not possess the capability to be updated with new root certificates in the field. Consequently, these browsers do not accept new root certificates issued by CAs that came into existence after they were first deployed. The new root certificates are accepted by newer browsers and other browsers that can be updated in the field. As such newer CAs operate at a disadvantage to older CAs.

The disadvantage can be addressed by having trust extended to the new root certificate (that can belong to the older CA or be another CA), by having the public key of the new root certificate cross-signed by an older root CA which is already accepted in the older browsers. As the cross-certified root CA is also recognized directly by the root store provider, it operates in accordance with the requirements of that certificate policy to which the root CA conforms. In addition, the cross-certified CA complies to any requirements placed upon it by the contract between it and the cross-certifying root CA.

This contract should indicate also the adherence to the root store policy.

The [BR-certs] may be used as guidance for clarification.

3.2.2. Issuing CA is a third party to the root CA

An issuing CA may operate as a third party subordinate to the root CA. The issuing CA's behaviour is governed by its contract with the root CA, which commonly stipulates adherence to the root store policy. Unlike the situation in section 3.2.1, the subordinate issuing CA is not recognized independently by any relationship with the root store provider.

3.2.3. Registration authority is a third party to the issuing CA

A registration authority may operate as a third party to an issuing CA. A registration authority's behaviour is governed by its contract with the issuing CA, which commonly stipulates adherence to the root store policy to which the CA adheres. A third party registration authority is not identified in a CA certificate.

3.2.4. Root CA is operated by the government

In the case where a root CA is operated by a government department, a root store provider may rely upon an audit conducted in accordance with the government's own internal audit process.

3.2.5. Subscriber operates issuing CA

A subscriber may operate its own issuing CA. Typically, the subscriber is approved to issue certificates only within a specific region of the name-space, and this limitation is enforced by legal means or it can be also technically constrained. For example, the root CA may use the name constraints certificate extension to limit the region of the name-space in which the issuing CA can issue valid certificates.

This is often referred to as an enterprise-based subordinate CA relationship.

3.2.6. Subscriber sources management of issuing CA

A root CA may host an issuing CA on behalf of a subscriber. Typically, the subscriber is approved to issue certificates only within a specific region of the name-space, and this limitation is enforced by the host root CA either technically or by legal means. Examination of the certificate chain would indicate that the issuing CA was owned by the subscriber by viewing the organization name in the subject field.

This may also be an enterprise-based CA relationship; however, the entity operating the CA (rather than the enterprise subscriber) has immediate control of the CA and physical possession of the CA private key.

3.2.7. Subscriber manages registration authority

A subscriber may manage a registration authority. The subscriber is approved to issue certificates only within a specific region of the name-space, and this limitation is enforced by the issuing CA through technical or legal means.

This is often referred to an enterprise-based registration authority relationship with the issuing CA.

3.2.8. Subscriber certificate issued by a root CA

Some legacy situations demand that a certificate be issued directly by a root CA, without the involvement of intermediate issuing CAs.

3.3. Subscriber

3.3.1. Subscriber uses agent

A subscriber may use a third party agent to manage its certificates. The third party will request certificates from a registration authority and manage the certificates in accordance with the subscriber agreement on the subscriber's behalf.

3.4. Browser

3.4.1. Browser directly trusts issuing CA key

A browser may allow a relying party to designate a CA key as a trust anchor for the purpose of evaluating subscriber certificates.

3.4.2. Browser directly trusts subscriber entity key

A browser may allow a relying party to designate a subscriber's certificate as a trust anchor.

3.4.3. Browser makes root CA public key unusable

A browser may allow a relying party to remove the trust of a root CA by deleting the root certificate from the root store. In some cases the trust removal may only be temporary as the browser or operating system may update the root store and restore the trust of the root CA.

3.4.4. Browser supports public key pinning

A browser may limit the set of public keys used to verify a certificate containing a domain name. Limitation can be done by including the set of accepted public keys in the browser or by respecting an HTTP header provided by the subscriber.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

The trust models described here exhibit several vulnerabilities that could adversely affect the reliability of the authentication they provide.

5.1. HTTPS is optional

The subscriber does not have to support HTTPS for the web site. The subscriber may provide HTTPS in some cases and not in other cases. As such, the trust model is optional for each web site. In the event of no HTTPS, the browser could more easily be attacked. This attack can be mitigated by supporting HSTS in accordance with RFC 6797 [RFC6797]. HSTS allows the subscriber to declare to the browser that interactions shall only be done using HTTPS connections.

5.2. automatic update of root certificates

The end user may remove or add some or all root certificates provided in a root store provider and then when an automatic update takes place it may be reinstated the removed ones and remove the added ones causing a posible denial of service and introducing some vulnerabilities.

5.3. Naming of subscribers

Subscriber names with any of the following characteristics can be used in an impersonation attack.

- homographic name
- mixed-alphabet name
- name that contains a string termination character
- Internet non-unique name (e.g. an internal server name)

With the exception of non-unique names, CAs in the Web PKI are required to screen out requests for certificates with any of these characteristics. CAs are required to phase out the practice of issuing non-unique names by 2015 per [BR-certs].

Technically, unless constrained by an upstream CA to issue certificates only in a specific region of the name-space, any CA in

the Web PKI can issue an apparently legitimate certificate for any name, whether or not the legitimate holder of that name is aware of or approves the issuance. Furthermore, the legitimate holder of that name may not discover that such a certificate has been issued.

5.4. Root CA compromise

In the event of a detected compromise of a root CA, its key is blacklisted by means of a software update. This has the effect of invalidating every certificate that is subordinate to that root CA, whether or not the certificate was issued while the compromise existed. This step would have a severe impact upon the CA and its subscribers; this is a step not likely to be taken without being very careful.

6. References

6.1. IETF Normative 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, May 2008.

[RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, June 2010.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, November 2012.

6.2. IETF Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, November 2003.

Appendix A. Other references

[BR-certs] - CA/Browser Forum, Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates. https://cabforum.org/baseline-requirements-documents/

[Mozilla-CP] - Mozilla CA Certificate Policy. https://www.mozilla.org/projects/security/certs/policy/

Authors' Addresses

Inigo Barreira (editor)
Izenpe
Beato Tomas de Zumarraga 71, 1. 01008 Vitoria-Gasteiz. Spain

Email: i-barreira@izenpe.net

Bruce Morton (editor)
Entrust
1000 Innovation Drive. Ottawa, Ontario. Canada K2K 3E7

Email: bruce.morton@entrust.com