Web PKI OPPS
Internet Draft
Intended status: Informational
Expires: April 2014

B. Wilson
Digicert
S. Chokhani
Cygnacom
R. Alden
Comodo

October 18, 2013

Browser processing of server certificates
draft-wilson-wpkops-browser-processing-00.txt

Abstract

This is one of a set of documents to define the operation of the Web PKI. It describes common variations in browser behavior related to processing server certificates.

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 ??.

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 April 18, 2014.

Table of Contents

1. Introduction

This document defines the current processing behaviors of browsers with respect to SSL/TLS session establishment between a server and a browser, including signature verification, certificate parsing, chain processing, revocation checking, and other processes described in RFC 5280 and the SSL/TLS protocol.

The information presented in this document is based on user experience and should not be construed as exhaustive. In other words, it is based on observed behavior and is not based on any comprehensive testing. The product vendors and reviewers are encouraged to provide additional information that sheds light on the observations made in this document or to provide additional observations.

This document does not address future changes to the implemented trust model.

1.1. Definitions

PKI terminology is as defined in RFC 5280. Other definitions are defined below for interpretation of this document.

Blacklisting - A concept where a browser does not trust a public key in the blacklist. A trust anchor, intermediate CA or TLS Server public key can be included in the blacklist. When a public key appears in the black list, the browser should not trust that public

key regardless of where the public key appears in the certification path (be it trust anchor, CA certificate or the TLS Server certificate). Some browsers support the blacklisting of public keys. Since blacklisting is a revocation mechanism, discussion of blacklisting is outside the scope of this document.

Bypassable error - A behavior in which the browser detects an abnormal condition and asks the user whether to proceed with (i.e. click-through to) the SSL/TLS connection.

Fatal error - A behavior in which the browser detects an abnormal condition and halts (or technically cannot complete) session negotiation and drops the connection or otherwise blocks the user from continuing (also referred to as "hard fail").

Internationalized Name - The Punycode-encoded ASCII representation of Unicode characters prefaced with the ASCII Compatible Encoding (ACE) prefix, xn--.

Name mismatch - A condition detected by a browser in which no name in the common name or subject alternative name for the subject in the certificate matches the hostname sought by the client (i.e. the client's reference identity - usually a Fully Qualified Domain Name - is not in the certificate).

Pinned - A condition in which the association between two or more aspects of the entity-public-key relationship (e.g. server name, public key, CA, certificate) are configured and set in the browser before initiation of a TCP connection.

Stapled - A condition in which information related to the server's certificate (e.g. OCSP response) is delivered by the server to the client as part of the SSL/TLS handshake, and not by direct communication with the issuing CA. Not all browsers request stapled responses. Since OCSP stapling is directly related to revocation, discussion of OCSP stapling is outside the scope of this document.

Visual indicator - A behavior in which the browser changes the color(s) and/or intensity of pixels on a screen in the browser chrome to indicate a changed condition. Visual indicators also include

error pages, pop-up dialogs, and warning messages.

Wildcard character - An asterisk - * (Unicode 2A).

1.2. Scope

The scope of this document excludes revocation checking. Revocation checking is addressed in another document.

The scope of the initial version of this document excludes the browser behavior for client authentication TLS. For example, a legitimate client certificate may not be presented or selected under

certain circumstances. One case in point is when the browser and the TLS Server have different trust anchors.

The scope of the initial version of this document excludes dealing with the following lesser-used X.509 certificate extensions: issuer alternative name; subject directory attribute; policy constraint; inhibit any policy; and subject information access.

In addition, due to revocation checking being out of scope, the discussion of the following extensions is out of scope: CRL Distribution Point; Freshest CRL; and OCSP field in the Authority Information Access extension. Discussion of OCSP stapling is outside the scope of this document.

The initial focus of this paper is on the following browsers and platforms:

	IExplorer	Firefox	Opera	Chrome	Safari
Windows XP	X	X	X	X	X
Windows 7+	X	X	X	X	X
Mac OS X	N/A	X	N/A	X	X
Linux	N/A	X	X	X	N/A

Additional platforms such as Android will be added.

1.3. Document Organization

This Section 1 has provided the introduction.

Section 2 describes the requirements for certification path development in order to establish trust in the server public key.

Section 2 also contains the nuances of the popular browsers in terms of their ability to meet these requirements and security implications of these nuances.

Section 3 describes requirements for certification path validation in order to establish trust in the server public key. Section 3 also contains the nuances of the popular browsers in terms of their ability to meet these requirements and security implications of these nuances.

Section 4 describes the requirements for processing the Server certificate. Section 4 also contains the nuances of the popular browsers in terms of their ability to meet these requirements and security implications of these nuances.

Section 5 describes the browser user interface indicators.

Section 6 lists IANA Considerations.

Section 7 summarizes Security Considerations discussed throughout this document.

Section 8 contains references.

2. Certification Path Development

2.1. Basic Requirements

This section lists the resources that a browser should be able to use for the development of certification path.

A browser should only use its trust anchor store to determine the trust anchor for a Server's certification path.

A browser should be able to use its local cache of certificates for certification path development.

A browser should be able to use the certificates sent by the TLS Server in the TLS handshake for certification path development.

A browser should be able to use the caIssuers field in the Authority Information Access extension in order to build the certification

path. Specifically, the browser should be able to use unsecure HTTP and unsecure LDAP method. The browser should be able to handle HTTP single certificate payload and multiple certificate payload as described in RFC 5280. The browser should be able to handle LDAP pointer to caCertificate and crossCertificatePair attribute as described in RFC 5280.

2.2. Additional Requirements

None

2.3. Browser Observations

We have observed that Firefox on any of the platforms listed in the scope section does not use caIssuers field in the Authority Information Access extension. This may result in undesired effect of rejecting a valid certificate since a path to the certificate was not

built. When a path cannot be built, Firefox gives a negative visual indication as a bypassable error as described in Section 5.6.

Inputs are sought from the working group participants and vendors to identify additional sources of certificates and additional exceptions.

3. Certification Path Validation

3.1. Basic Requirements (based on RFC 5280)

A browser should only use one or more trust anchors from its trust anchor store for certification path validation.

A browser should perform certification path validation in accordance with Section 6 of RFC 5280.

3.2. Additional Requirements

The public exponent for all RSA keys in SSL/TLS certificates must be an odd number and cannot be "1."

None of the browsers check the public exponent to verify that it is odd and it is not one.

After December 31, 2013, all public RSA keys must be at least 2048 bits.

Currently, if an RSA key size is less than 900 bits, Opera presents the user with a negative visual indicator and a bypassable dialog. If the RSA key size is greater 900 bits but less than 1,000 bits it removes the padlock indicator.

Microsoft allows manual configuration of minimum key lengths by editing the registry, using a certificate utility or other mechanisms. See http://support.microsoft.com/kb/2661254. Thus, IE and Chrome on Windows platform can enforce the 2,048 bit requirement. It is not known if Firefox on Windows or other browsers listed in the scope section address the 2,048 bit key size requirement.

3.3. Browser Observations

3.3.1. Name Constraints

We have observed that Safari does not process name constraints. Thus, if the name constraints extension is non-critical, Safari provides no visual indicator of any anomaly. If the name constraint is critical, Safari will reject the certification path due to an unrecognized critical extension, but will give the user a choice to proceed with the connection.

The following additional observations are made with respect to name constraint violations:

. Microsoft IE on Windows platforms enforces name constraints (in both the CN and in the Subject Alternative Name), but gives the user a choice to proceed with the connection.

. Firefox on all platforms enforces name constraints (in both the CN and in the Subject Alternative Name) and does not permit the user to proceed.

. Chrome on the Windows platform enforces name constraints (in both the CN and in the Subject Alternative Name), and does not permit the user to proceed.

. Chrome on Mac OSX seems to follow the behavior of Safari in not recognizing critical extension and does not permit the user to proceed.

. Chrome on Linux enforces name constraints in the Subject Alternative Name and does not enforce the name constraint on the CN. Furthermore, in the case of name constraint failure on Linux, Chrome gives the user a choice to proceed with the connection.

3.3.2. Validity Period

The browser may display a warning indicating that a certificate in the certification path is outside of its validity interval or expired. The user may be given the choice to proceed to the content. The trust indicator may be suppressed. In some cases, there may be no warning, but the trust indicator is simply suppressed.

When the browser detects that the current system time is beyond the validity period of a certificate in the certification path, a warning is displayed. Some browsers indicate that a certificate has expired and present a bypassable error asking whether or not to proceed or allowing the user to view the certificate with a certificate viewer. Some browsers also alert the user to the possibility that the error is not caused by an expired certificate, but by incorrect system time, and display the system time. For example, "Your computer's clock currently indicates it is Monday, October 14, 2013, 4:00 AM. Does this look right? If not, you should correct the error and refresh this page."

We plan to enhance this section with additional and more complete information in terms of validity period for certificates in the certification path (i.e., handling expired certificates).

3.3.3. Areas for Future Work

Future work will include a review of the basic constraints extension.

We also plan to discuss how pinning interplays with certification path development and validation. Some browsers support the pinning of public keys.

Most browsers perform certificate policy extension processing appropriately. We have not examined if the policy mapping, inhibit any policy, and policy constraints extensions are processed correctly. Most browsers as a result of certificate policies extension processing provide a visual indication when they detect that all the certificates in the certification path contains the correct policy OID for Extended Validation. We plan to quantify this characteristics for the browsers listed in the scope section.

Inputs are sought from the working group participants and vendors to identify additional path validation rules and additional exceptions.

4. Server certificate processing

This section focuses on how the browsers use Server certificates. While some of these checks should be part of certification path validation, these checks are discussed here as part of TLS Server certificate processing in order to emphasis how the browsers use the information in TLS Server certificates.

4.1. Subject Names

SSL/TLS certificates contain at least one subject name to bind the public key in the certificate with the server that possesses corresponding private key. The subject name appears in the subject alternative name extension as dNSName name type and often in the common name field. The latter practice of using the common name was deprecated by RFC 2818. A browser processes the subject name in the certificate to determine whether it matches the expected server name. Browsers are known to successfully connect with servers whose DNS name appears in the Subject CN only and when subject alternative name extension is absent. As discussed earlier, the enforcement of name constraint on the DNS name appearing in CN varies with the browser.

Browser processing of internationalized names in subject names of certificates allow browsers to either process the Internationalized

Name back into Unicode or display the Internationalized Name in ASCII as xn--.

In addition to the use of names for SSL/TLS processing, certificate distinguished name fields may provide further identification of the subject through domain-component naming and X.500 naming (e.g. country, organization, etc.). When name constraints are used on the DN, the entire subject distinguished name (not just the CN) needs to pass the name constraints.

Section 3.1 of RFC 2818 states that in the case of a certificate name mismatch, a browser "MUST either notify the user (clients MAY give the user the opportunity to continue with the connection in any case) or terminate the connection with a bad certificate error."

Typical browser behavior will provide a message box that reads, "Security Error: Domain Name Mismatch" with treatment as a bypassable error with options such as "View Certificate," "OK" or "Cancel."

Some browsers still prioritize common name processing over subject alternative name processing even though use of the common name has been deprecated. Another scenario is when the common name is not one of the names listed as a subject alternative name. When either of these occur, a browser might throw a domain name mismatch even though the name to be used for the SSL/TLS session is in either the common name field or the subject alternative name of the certificate but not in both.

Most browsers display a warning, but allow the user to proceed to viewing the contents of the web site.

Some systems, such as Keychain Access in Apple OS X, allow the user to override certificate name mismatches by explicitly trusting a certificate for a particular domain name that is not contained in the certificate.

4.2. Wildcard character

Some browsers support a wildcard character in the leftmost position. We plan to quantify wildcard behaviors for the browsers listed above in the scope section.

4.3. Key Usage Extension

The browsers should use the server public key for key encryption, key agreement or digital signature verification depending on the TLS cipher suite selected. Below are a few examples:

- TLS_RSA_WITH_AES_128_CBC_SHA: The Server key is used for encrypting the master secret and thus, the Server certificate should have the key encipherment bit set if the Server certificate contains the key usage extension.

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA: The Server key is used for authentication of ephemeral DH key thus, the Server certificate should have the digital signature bit set if the Server certificate contains the key usage extension.

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA: The Server key is used for authentication of ephemeral DH key thus, the Server certificate should have the digital signature bit set if the Server certificate contains the key usage extension.

- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA: The Server key is used for ECDH key agreement/exchange. Thus, the Server certificate should have the key agreement bit set if the Server certificate contains the key usage extension.

We plan to add information how the browsers adhere to these requirements.

4.4. Extended Key Usage

If the extended key usage extension is present in the Server certificate, it should have one of the following OIDs: Server Authentication or anyExtendedKeyUsage.

We plan to add information how the browsers adhere to this requirement.

5. Browser Human Interface (Visual) Indicators

This section describes the typical kinds of browser/OS behaviors when processing SSL/TLS certificates.

5.1. Visual indicators

The most commonly used visual indicator of SSL/TLS security is the padlock icon. Variations of the icon include the closed padlock, the open padlock, and the padlock superimposed with a red slash or X.

5.2. Positive visual indicators

Commonly used visual indicators that are considered positive indications of web site authentication or security are a closed padlock icon, use of the color green, and the display of additional information about issuer or subject of the certificate.

Some of these indicators are called EV indicators because of their use when displaying a website that presents an Extended Validation certificate to the browser.

5.3. Negative visual indicators

Visual indicators used by browsers to convey warnings include use of the color red, a slash (/) or X across a positive indicator (a red slash or X across the padlock icon and/or the "https"), a message box, or the removal of a positive indicator (e.g. removal of the padlock).

5.4. Message boxes, dialog boxes and error pages

A message box is generally used not just as negative indicator, but also to convey more context-specific guidance to the end user. They can provide warnings or explain why an SSL/TLS connection cannot be

completed. Dialog boxes are used when the browser encounters an uncertain environmental condition (for gray areas where the security threat is not black or white). Some dialog boxes provide a simple binary choice (a) proceed or (b) "get me out of here." This type of browser behavior can be referred to as a "single bypassable error." Other dialog boxes can exhibit more complex behavior, such as multiple branches, additional nested bypassable errors, helpful

information, and decisions to be made by the user.

An error page is another mechanism used by browsers to provide certificate-related information to users.

Some error messages provide an option to view the certificate. Clicking on the offer launches the browser's certificate viewer.

5.5. Certificate viewers

Most browsers provide a means to examine the SSL/TLS certificate of the web site and the chain of certificates leading up to the root certificate. Some browsers block viewing the certificate in circumstances determined by the browser to be insecure.

5.6. Certification Path Development and Validation Indication

If the certification path cannot be validated, some browsers will alert the user about the inability to complete the server's certificate chain.

Most browsers will provide a warning when a certificate is signed by an unknown CA. The warning usually states that an unknown authority issued the certificate. Additional warnings include that if the user has connected to the site previously without errors, it may mean an attacker is trying to impersonate the site and intercept confidential communications. Users are advised not to continue unless they are sure.

With some browsers, this error can be bypassed for the session or the user can explicitly trust the certificate permanently. When a certification path fails because the issuer is not in the certificate/key store, most browsers will still allow the user to explicitly trust the certificate or the issuing CA. The number of steps required to explicitly trust an untrusted certificate vary from browser to browser.

5.7. Configurables

Most browsers provide the ability to configure certain certificate-related behaviors. In Mozilla Firefox a user can change some options using Tools -> Options -> Advanced -> Certificates or by typing

"about:config" in the address window and editing security preferences. Changes in Microsoft's Internet Explorer settings can be made under Tools -> Internet options -> Advanced -> Security or by editing the registry. In Apple OS X, configuration changes are performed by accessing preferences for certificates in the Keychain, but since the only configurations available are related to revocation checking (CRLs and OCSP), they are outside the scope of this draft.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

The operations described above exhibit several vulnerabilities that could adversely affect the reliability of the authentication and security provided by SSL/TLS certificates. These vulnerabilities have been discussed throughout this RFC and are summarized below:

These items will be provided when the draft becomes more stable.

8. Normative 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, Nov 2003.

[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.

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

[W3C-WSC] Web Security Context: User Interface Guidelines, W3C Recommendation 12 August 2010.

Authors' Addresses

Ben Wilson
Email: ben@digicert.com

Santosh Chokhani
Email: schokhani@cygnacom.com

Robin Alden
Email: robin@comodo.com