

X.509v3 TLS Feature Extension
draft-hallambaker-tlsfeature-10

Abstract

The purpose of the TLS feature extension is to prevent downgrade attacks that are not otherwise prevented by the TLS protocol. In particular, the TLS feature extension may be used to mandate support for revocation checking features in the TLS protocol such as OCSP stapling. Informing clients that an OCSP status response will always be stapled permits an immediate failure in the case that the response is not stapled. This in turn prevents a denial of service attack that might otherwise be possible.

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 January 7, 2016.

Copyright Notice

Copyright (c) 2015 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 (<http://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. Definitions

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

1.2. TLS Feature, X.509 Extension

In order to avoid the confusion that would occur in attempting to describe an X.509 extension describing the use of TLS extensions, in this document the term 'extension' is reserved to refer to X.509v3 extensions and the term ?TLS feature extension? is used to refer to what the TLS specification [[RFC5246](#)] refers to as an ?extension?.

2. Purpose

The purpose of the TLS feature extension is to prevent downgrade attacks that are not otherwise prevented by the TLS protocol.

Since the TLS protocol itself provides strong protection against most forms of downgrade attack including downgrade attacks against cipher suite choices offered and client credentials, the TLS feature extension is only relevant to the validation of TLS protocol credentials.

At the time of writing, the only TLS feature extensions that are relevant to the revocation status of credentials are the Certificate Status Request extension (status_request) and the Multiple Certificate Status Extension (status_request_v2). These extensions are used to support in-band exchange of Online Certificate Status Protocol (OCSP) tokens, otherwise known as OCSP stapling. These extensions are described in [[RFC6066](#)] and [[RFC6961](#)].

The OCSP stapling mechanism described in [[RFC6066](#)] permits a TLS server to provide evidence of valid certificate status inband. When this information is provided inband, the privacy, performance and reliability concerns arising from the need to make a third party connection during the TLS handshake are eliminated. A client cannot however draw any conclusion from the absence of inband status information unless it knows that the legitimate server would have provided it. The status information might have been omitted because the server does not support the extension or because the server is

withholding the information intentionally, knowing the certificate to be invalid.

The inclusion of a TLS feature extension advertising the status_request feature in the server end entity certificate permits a client to fail immediately if the certificate status information is not provided by the server. The need to query the OCSP responder is eliminated entirely. This improves client efficiency and more importantly prevents a denial of service attack against the client by either blocking the OCSP response or mounting a denial of service attack against the OCSP responder.

Since the TLS feature extension is an option, it is not likely that an attacker attempting to obtain a certificate through fraud will choose to have a certificate issued with this extension. Such risks are more appropriately addressed by mechanisms such as Certification Authority Authorization DNS records [RFC 6844](#) [[RFC6844](#)] that are designed to prevent or mitigate mis-issue.

A server offering an end entity certificate with a TLS feature extension MUST satisfy a client request for the specified feature unless this would be redundant as described below. Clients MAY refuse to accept the connection if the server does not accept a request for a specified feature.

A Certification Authority SHOULD NOT issue certificates that specify a TLS feature extension advertising features that the server does not support.

A server MAY advise a Certification Authority that it is capable of supporting a feature by including the corresponding TLS feature extension in a Certificate Signing Request [[RFC2986](#)]. A server SHOULD verify that its configuration supports the features advertised in the credentials presented to a client requesting connection.

This document describes the use of the TLS feature in PKIX end entity certificates and certificate signing certificates. A mechanism that MAY be used to describe support for the specified features in-band for the most commonly used certificate registration protocol is also provided.

3. Syntax

See [Appendix A](#) for an ASN.1 module

The TLS feature extension has the following format:

tls-feature OBJECT IDENTIFIER ::= { id-pe TBD2 }

Features ::= SEQUENCE OF INTEGER

The extnValue of the tls-feature extension is the ASN.1 DER encoding of the Features structure.

The TLS feature extension SHOULD NOT be marked critical. [RFC 5280](#) [[RFC5280](#)] requires that implementations that do not understand critical extensions MUST reject the certificate. Marking the TLS feature extension critical breaks backward compatibility and is not recommended unless this is the desired behavior.

[3.1.](#) TLS Feature

The object member Features is a sequence of TLS extension identifiers (features, in this specification's terminology) as specified in the IANA Transport Layer Security (TLS) Extensions registry. If these features are requested by the client in its ClientHello message, then the server MUST return a ServerHello message that satisfies this request.

This specification does not require a TLS client to offer or support any TLS feature regardless of whether it is specified in the server certificate's TLS feature extension or not. In particular a client MAY request and a server MAY support any TLS extension regardless of whether it is specified in a TLS feature extension or not.

A server that offers a certificate that contains a TLS feature extension MUST support the features specified and comply with the corresponding requirements.

[3.2.](#) Use

[3.2.1.](#) Certificate Signing Request

If the certificate issue mechanism makes use of the PKCS#10 Certificate Signing Request (CSR) [[RFC2986](#)], the CSR MAY specify a TLS feature extension as a CSR Attribute as defined in [\[RFC2986\] section 4.1](#). A server or server administration tool should only generate key signing requests that it knows can be supported by the server for which the certificate is intended.

[3.2.2.](#) Certificate Signing Certificate

When present in a Certificate Signing Certificate (i.e., Certification Authority certificate with the key usage extension value set to keyCertSign), the TLS feature extension specifies a

constraint on valid certificate chains. Specifically, a certificate that is signed by a Certificate Signing Certificate that contains a TLS feature extension MUST contain a TLS feature extension which offer the same set or a superset of the features advertised in the signing certificate.

This behavior provides a means of requiring support for a particular set of features for certificates issued under a particular Certificate Signing Certificate without requiring TLS clients to verify compliance with TLS feature extensions in multiple certificates.

3.2.3. End Entity Certificate

When specified in a server end entity Certificate (i.e. a certificate that specifies the id-kp-server EKU), the TLS feature extension specifies criteria that a server MUST meet to be compliant with the feature declaration.

In the case that a client determines that the server configuration is inconsistent with the specified feature declaration it MAY reject the TLS configuration.

3.2.3.1. TLS status_request

In the case that a client determines that the server configuration is inconsistent with a feature declaration specifying support for the TLS status_request extension it SHOULD reject the TLS configuration.

A client MAY accept a TLS configuration despite it being inconsistent with the TLS feature declaration if the validity of the certificate chain presented can be established through other means. For example, by successfully obtaining the OCSP data from another source.

There are certain situations in which the alternative to establishing a connection with imperfect TLS security is to transmit the same information with no security controls whatsoever. Accordingly, a client MAY accept a TLS configuration despite it being inconsistent with the TLS feature declaration but MUST NOT distinguish that connection as secure.

3.3. Processing

Advertising a TLS feature extension may change the expectations of relying parties. If these expectations are not met, a valid certificate may be rejected as invalid. Particular attention is required at the start of a certificate lifecycle. A server will be unable to comply with a TLS feature extension if the certificate is

issued and released to the subject before the corresponding status token is published.

3.3.1. Certification Authority

A Certification Authority SHOULD NOT issue certs with a TLS feature extension unless there is an affirmative statement to the effect that the end entity intends to support the specified features. For example the use of a feature extension in the CSR or through an out of band communication.

A Certification Authority SHOULD ensure that the certificate provisioning process for certificates containing a TLS feature extension permits the certificate subject to meet the requirements. For example ensuring that OCSP tokens are published before the corresponding certificate is released to the subscriber.

3.3.2. Server

A TLS server certificate containing a TLS feature extension MAY be used with any TLS server that supports the specified features. It is not necessary for the server to provide support for the TLS feature extension itself. Such support is nevertheless desirable as it can reduce the risk of administrative error.

A server SHOULD verify that its configuration is compatible with the TLS feature extension expressed in a certificate it presents. When an existing certificate is to be replaced by a new one, the server SHOULD NOT begin using the new certificate until the necessary OCSP status token(s) are available.

A server MAY override local configuration options if necessary to ensure consistency but SHOULD inform the administrator whenever such an inconsistency is discovered.

A server SHOULD support generation of the feature extension in CSRs if key generation is supported.

3.3.3. Client

A client MUST treat a certificate with a TLS feature extension as an invalid certificate if the features offered by the server do not contain all features present in both the client's ClientHello message and the TLS feature extension

In the case that use of TLS with a valid certificate is mandated by explicit security policy, application protocol specification or other means, the client MUST refuse the connection. If the use of TLS with

a valid certificate is optional, a client MAY accept the connection but MUST NOT treat the certificate as valid.

4. Acknowledgements

This proposal incorporates text and other contributions from participants in the IETF and CA-Browser forum. In particular, Robin Alden, Richard Barnes, Viktor Dukhovni, Stephen Farrell, Gervase Markham, Yoav Nir, Tom Ritter, Jeremy Rowley, Stefan Santesson, Ryan Sleevi, Brian Smith, Rob Stradling and Sean Turner

5. Security Considerations

5.1. Alternative Certificates and Certificate Issuers

Use of the TLS feature extension to mandate support for a particular form of revocation checking is optional. This control can provide protection in the case that a certificate with a TLS feature is compromised after issue but not in the case that the attacker obtains an unmarked certificate from an issuer through fraud.

The TLS feature extension is a post-issue security control. Such risks can only be addressed by security controls that take effect before issue.

5.2. Denial of Service

A certificate Issuer could issue a certificate that intentionally specified a feature statement that they knew the server could not support.

The consequences of such refusal would appear to be limited since a Certification Authority could equally refuse to issue the certificate.

5.3. Cipher Suite Downgrade Attack

The TLS feature extension does not provide protection against a cipher suite downgrade attack. This is left to the existing controls in the TLS protocol itself.

6. IANA Considerations

On approval, IANA shall add in the SMI Security for PKIX Certificate Extension (1.3.6.1.5.5.7.1) registry the following entry:

| Decimal | Description | References |
|---------|------------------|------------|
| ----- | ----- | ----- |
| TBD2 | id-pe-tlsfeature | {this RFC} |

7. [Appendix A](#): ASN.1 module

```

TLS-Feature-Module-2015 {
    iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-tls-feature-2015(TBD1)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

    IMPORTS -- From RFC 5912

    id-pe
    FROM PKIX1Explicit-2009 {
        iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-pkix1-explicit-02(51)}

    EXTENSION
    FROM PKIX-CommonTypes-2009 {
        iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-pkixCommon-02(57)}
    ;

    CertExtensions EXTENSION ::= {
        ext-TLSFeatures, ... }

    -- TLS Features Extension

    ext-TLSFeatures EXTENSION ::= { SYNTAX
        Features IDENTIFIED BY id-pe-tls-features }

    id-pe-tls-features OBJECT IDENTIFIER ::= { id-pe TBD2 }

    Features ::= SEQUENCE OF INTEGER

END

```

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", [RFC 6066](#), January 2011.
- [RFC6961] Pettersen, Y., "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension", [RFC 6961](#), June 2013.
- [RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", [RFC 6844](#), January 2013.
- [RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", [RFC 2986](#), November 2000.
- [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.

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

Email: philliph@comodo.com