Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 13, 2014

P. Hallam-Baker
Comodo Group Inc.

June 11, 2014

X.509v3 TLS Feature Extension
draft-hallambaker-tlsfeature-04

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 December 13, 2014.

Table of Contents

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

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 'feature' is used to refer to a TLS 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 is only relevant to the validation of TLS protocol credentials. In particular to the revocation status of the server credentials presented.

At the time of writing, the only TLS feature extensions that are relevant to the revocation status of credentials is the Certificate Status Request extension (status_request) Multiple Certificate Status Extension (status_request_v2) These extensions are used to support in-band exchange of OCSP tokens, otherwise known as OCSP stapling. These extensions are described in [RFC6066] and [draft-pettersen-tls-ext-multiple-ocsp-03].

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 witholding 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 approrpriately addressed by mechanisms such as Certificate Authority Authorization DNS records RFC 6844 [RFC6844] that are designed to prevent or mitigate mis-issue. Nevertheless a Certification Authority MAY consider the presence or absence of a

required TLS feature as one factor in determining the level of additional scruitiny a request should be subject to.

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. Otherwise clients MAY refuse connection. It is important therefore that a Certification Authority only issue certificates that specify features that match the configuration of the server and that the server is capable of verifying that its configuration is compatible with the feature declaration of the certificates it offers. Ideally, the TLS feature declaration would be specified by the certificate request generator as part of the certificate issue process.

A client feature request is redundant if the purpose of the request is fully satisfied by another feature. For example, a server need not satisfy a client request for the status_request feature if the status_request_v2 is offered and satisfied.

In the case that the cached_information feature is offered and satisfied, a client request for the status_request or status_request_v2 features is satisfied if and only if the cached credentials referenced include the OCSP status information necessary to establish the certificate status.

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

3. Syntax

The TLS Feature extension has the following format:

```
tls-feature OBJECT IDENTIFIER ::=  { id-pe  1 }

Features ::= SEQUENCE OF INTEGER
```

The TLS Feature Extension SHOULD NOT be marked critical. RFC 5280 [RFC5280] requires that implementations that do not understand the extension MUST reject the certificate. Marking the TLS Feature Extension critical breaks backward compatibility and is not recommended unless this is the desired behavior. Implementations that process the extension MUST ignore the criticality bit setting.

3.1. TLS Feature

The TLS Feature extension lists a sequence of TLS extension identifiers (features) that a TLS server compliant with the feature declaration MUST support and satisfy on client 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.

If a TLS Feature extension specifies a TLS feature, a server offering the certificate MUST support the extension specified and MUST comply with any specific requirements specified for that feature in this document or in the document that specifies the TLS feature.

3.1.1. status_request

If the TLS status_request feature is specified in the TLS Feature extension and a TLS client specifies the status_request feature in the Client Hello, a server MUST return a valid OCSP token for the specified server's End Entity certificate in the response.

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. 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., CA 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 MUST offer the same set or a superset of the features advertised in the signing certificate.

While relying parites (i.e., clients) MAY reject certificates that do not comply with this requirement, the use of TLS Feature extension in Certificate Signing Certificates is primarily intended for use by

parties seeking to evaluate the performance of certificate issuers and MAY be ignored by clients.

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.

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.

3.3. Processing

3.3.1. Certification Authority

A CA 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.

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

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

3.3.3. Client

A compliant client SHOULD reject a TLS connection with security properties that are inconsistent with the specified TLS Feature extension. A compliant client MAY accept such a TLS connection request however if it is determined that doing so is appropriate in particular circumstances.

4. Acknowledgements

[List of CABForum and PKIX contributors]

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 risks of such refusal would appear to be negligible since a Certificate 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
23	id-pe-tlsfeature	{this RFC}

7. Normative References

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

[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.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 6844, January 2013.

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

Email: philliph@comodo.com