Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2013

P. Hallam-Baker
Comodo Group Inc.

October 2, 2012

X.509v3 Extension: OCSP Stapling Required
draft-hallambaker-muststaple-00

Abstract

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

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 April 5, 2013.

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].

2. Purpose

[This could do with a re-write to start by explaining the intended use od OCSP, the reasons that browsers do not normally hard fail (DoS attack, flaky OCSP servers, efficiency) and that passing the OCSP data in band avoids dependence on an external service but does not allow the client to avoid the DoS attack or consequences of flaky servers unless it knows the OCSP data is passed in band.]

The purpose of the TLS Security Policy 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, the TLS Security Policy is only relevant to the validation of TLS protocol credentials. In particular to the revocation status of the credentials presented.

At the time of writing, the only TLS feature that is relevant to the revocation status of credentials is the Certificate Status Request extension (status_request) used to support in-band exchange of OCSP tokens, otherwise known as OCSP stapling. This extension is described in RFC 4366 [RFC4366].

The TLS Security Policy mechanism described in this document is designed to support policy statements that prevent a downgrade attack against the current OCSP stapling mechanism and possible future certificate revocation mechanisms such as stapling of multiple OCSP tokens.

The OCSP stapling mechanism described in RFC 4366 [RFC4366] permits a TLS server to provide evidence of valid certificate status inband and thus improve client response. A TLS Security Policy that advertises the status_request extension informs a client that if the status_request is specified in a TLS Client Helo, that a server compliant with the policy MUST respond with a valid OCSP token for the End Entity Certificate it presents.

Verification of the TLS Security Policy extension in a client permits

the client to avoid reliance on certificates that are revoked for the reasons that occur most frequently. In particular it allows a client to avoid mis-reliance on certificates that are revoked for cause or at the request of the subject (e.g. because of a compromised private key).

Since the TLS Security Policy 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 records that are designed to prevent or mitigate mis-issue. Nevertheless a Certification Authority MAY consider the presence or absence of a required security policy as one factor in determining the level of additional scruitiny a request should be subject to.

Any security policy specified in an End Entity certificate MUST be followed by the server or clients MAY refuse connection. It is important therefore that a Certification Authority only issue certificates that specify policies that match the configuration of the server and that the server is capable of verifying that its configuration is compatible with the security policy of the certificates it offers. Ideally, the TLS security policy would be specified by the client as part of the certificate issue process.

This document describes a mechanism that MAY be used to provide this communication in-band for the most commonly used certificate registration protocol.

3. Syntax

The TLS Security Policy extension has the following format:

```
cabf-tls-security-policy OBJECT IDENTIFIER ::=  { cabf 1 }

SecurityPolicy ::= SEQUENCE {
    minVersion       [0]     INTEGER,
    extensions       [1]     SEQUENCE OF INTEGER OPTIONAL}
```

Extension MAY be marked critical. Implementations that process the extension MUST ignore the criticality bit setting.

3.1. minVersion

The minVersion element constrains the minimum version of the TLS protocol to be offered as follows:

1 To be compliant with the policy a server MUST offer TLS version 1.0 or higher

2 To be compliant with the policy a server MUST offer TLS version 1.1 or higher

3 To be compliant with the policy a server MUST offer TLS version 1.2 or higher

n To be compliant with the policy a server MUST offer a TLS connection that specifies a version identifier of {3, m} where m >= n.

For historical reasons, TLS version 1.0 uses the protocol identifer {3,1}, TLS version 1.1 the identifier {3,2} and so on. The minVersion specifier is defined for consistency with the internal TLS protocol rather than the descriptive name.

It will be noted that this approach does not support major version number increments. This is intentional since a major version number increment signals an incompatible change to the specification which would almost certainly require a new Security Policy extension to be defined.

3.2. extensions

The extensions element lists a sequence of TLS extension identifiers that a server compliant with the policy MUST support and accept on client request.

If the TLS status_request extension is specified in the Client Hello, a compliant server MUST return a valid OCSP token for the specified End Entity certificate in the response.

3.3. Use

3.3.1. Certificate Signing Request

If the certificate issue mechanism makes use of the PKCS#10 Certificate Signing Request (CSR) RFC 4366 [RFC4366], the CSR MAY specify a TLS Security Policy extension as a CSR attribute. A server or server administration tool should only generate key signing requests that it knows can be supperted by the server for which the certificate is intended.

3.3.2. Certificate Signing Certificate

When present in a Certificate Signing Certificate, the TLS Security Policy extension specifies a constraint on valid certificate chains. Specifically, a certificate chain is only valid if each certificate in the chain specifies a TLS Security Policy that is at least as restrictive as that specified in the certificate for the key used to sign it.

While relying clients MAY reject certificates that do not comply with this particular requirement, the use of TLs Security Policy 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.

3.3.3. End Entity Certificate

When specified in an End Entity Certificate, the TLS Security Policy extension specifies criteria that a server MUST meet to be compliant with the policy.

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

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

3.4. Processing

3.4.1. Certification Authority

A CA SHOULD NOT issue certs with the extension unless there is an affirmative statement to the effect that stapling is requested.

For example the use of the extension in the CSR or through an out of band communication.

3.4.2. Server

A server SHOULD verify that its configuration is compatible with the TLS Security Policy 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 NOT override local configuration options to offer use

of cipher suites that have been otherwise deprecated as insecure but MAY override local configuration to offer stronger cipher suites that would otherwise have been the case. For example, a server that would normally only offer DES might offer 3DES but not vice versa.

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

3.4.3. Client

A compliant client SHOULD reject an attempt to establish a TLS connection with security properties that are inconsistent with the specified TLS Security Policy extension.

4. Acknowledgements

[List of CABForum and PKIX contributors]

5. Security Considerations

5.1. Alternative Certificates and Certificate Issuers

Use of the TLS Security Policy 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 Security Policy is compromised after issue but not in the case that the attacker obtains an unmarked certificate from an issuer through fraud.

TLS Security Policy 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 security policy 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 Security Policy extension does not provide protection against a cipher suite downgrade attack. This is left to the existing controls in the TLS protocol itself.

6. For discussion.

[RFC EDITOR: DELETE PRIOR TO PUBLICATION]

During the design of the extension, various proposals were made to add functionality. This section explains the reasons that particular functionality was chosen to be supported. It should be deleted by the RFC editor prior to publication.

6.1. Cipher Suite Downgrade Attack

The TLS protocol provides controls designed to prevent a cipher suite downgrade attack. Should these be found to be inadequate, the appropriate response would be a modification of the TLS protocol and assignment of a new minor version number or a required extension.

6.2. Remove Version attribute

The TLS protocol arguably provides a sufficient degree of protection against a version number downgrade attack and thus it could be argued that this particular attribute is unnecessary.

The reason the attribute was included is that it provides a failsafe for the single most important aspect of the protocol. While TLS is intended to be secure against downgrade attack this is not established by means of a formal proof, nor is it likely that such proof would be possible without making assumptions as to the security of the cryptographic algorithms used to authenticate packets.

6.3. Mandated client behavior

Many certificate issuers (including this one) would like to mandate a particular set of client behavior when a certificate is processed. For example, requiring that a certificate 'hard fail' in cases where the server is unable to obtain OCSP status.

Desirable though this functionality is to certificate issuers, it is hard to see how client providers are likely to provide support. In particular the browser providers are already aware that CAs would prefer that they 'hard fail' on OCSP status. Will expressing that request in a certificate make them any more likely to comply?

6.4. Other protocols and applications

Should we describe a similar extension for code signing?

While such an extension would clearly be useful, execution would require a specification for code signing to refer to.

7. IANA Considerations

No action by IANA is required.

8. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

[X.509] International Telecommunication Union, "ITU-T Recommendation X.509 (11/2008): Information technology - Open systems interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, November 2008.

[X.680] International Telecommunication Union, "ITU-T Recommendation X.680 (11/2008): Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T Recommendation X.680, November 2008.

Author's Address

Phillip Hallam-Baker
Comodo Group Inc.

Email: philliph@comodo.com