INTERNET-DRAFT   S. Santesson (3xA Security)
Intended Status: Proposed Standard   Q. Dang   (NIST)
Expires January 2010   June 2009

Transport Layer Security (TLS) Cached Information Extension

<draft-ietf-tls-cached-info-00.txt>

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

This document defines a Transport Layer Security (TLS) extension for cached information. This extension allows the TLS client to inform a server of cached information from previous TLS sessions, allowing the server to omit sending cached static information to the client during the TLS handshake protocol exchange.

1 Introduction

TLS handshakes often include fairly static information such as server certificate and a list of trusted Certification Authorities (CAs). Static information such as a server certificate can be of considerable size. This is the case in particular if the server certificate is bundled with a complete certificate path, including all intermediary certificates up to the trust anchor public key.

Significant benefits can be achieved in low bandwidth and high latency networks, in particular if the communication channel also has a relatively high rate of transmission errors, if a known and previously cached server certificate path can be omitted from the TLS handshake.

This specification defines the Cached Information TLS extension, which may be used by a client and a server to exclude transmission of known cached parameters from the TLS handshake.

1.1 Terminology

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 Cached Information Extension

A new extension type (cached_information(TBD)) is defined and used in both the client hello and server hello messages. The extension type is specified as follows.

```
enum {
     cached_information(TBD), (65535)
} ExtensionType;
```

The "extension_data" field of this extension SHALL contain "CachedInformation" according to the following structure:

```
enum {
     certificate_chain(1), trusted_cas(2), (255)
} CachedInformationType;

struct {
     HashAlgorithm hash;
     opaque hash_value<1..255>;
} CachedInformationHash;

struct {
     CachedInformationType type;
     CachedInformationHash hashes<1..2^16-1>;
} CachedObject;
```

```
      struct {
           CachedObject cached_info<1..2^24-1>;;
      } CachedInformation;
```

Hash algorithm identifiers are provided by the RFC 5246 [RFC5246] HashAlgorithm registry. Compliant implementations MUST support sha1(2) as HashAlgorithm.

When CachedInformationType identifies certificate_chain, then hash_value MUST include at least one hash value calculated over the certificate_list element of a server side Certificate message.

When CachedInformationType identifies trusted_cas, then hash_value MUST include at least one hash value calculated over the certificate_authorities element of a server side CertificateRequest message.

Other specifications MAY define more CachedInformationType types.

4 Message flow

Clients MAY include an extension of type "cached_information" in the (extended) client hello, which SHALL contain at least one CachedObject as specified in section 2.

Servers that receive an extended client hello containing a "cached_information" extension, MAY indicate that they support one or more of the cached information objects by including an extension of type "cached_information" in the (extended) server hello, which SHALL contain at least one CachedObject received from the client. The CachedObject's returned by the server MUST include the types the server supports and has accepted to replace with a hash of the cached data.

After negotiation of the use of cached certificates has been successfully completed (by exchanging hello messages including "cached_certs" extensions), the server MUST replace agreed cached information objects in its handshake messages with a corresponding hash_value from CachedInformationHash that was included in the cached_information extension of the server hello message.

The handshake protocol will proceed using the cached data as if it they were provided in the handshake protocol. The finished message will however be calculated over the actual data exchanged in the handshake protocol. That is, the finished message will be calculated over the hash values of cached information objects and not over the cached objects that were omitted from transmission.

5 Security Considerations

Hash algorithms used in this specification are required to have reasonable random properties in order to provide reasonably unique identifiers. Failure of a provided hash to correctly and uniquely identify the correct set of hashed parameters may at most lead to a failed TLS handshake followed by a new attempt without the cached information extension. No serious security threat requires selected hash algorithms to have strong collision resistance.

6 IANA Considerations

1) Create an entry, cached_information(TBD), in the existing registry for ExtensionType (defined in RFC 5246 [RFC5246]).

2) Establish a registry for TLS CachedInformationType values. The first entries in the registry are certificate_chain(1) and trusted_cas(2). TLS CachedInformationType values in the inclusive range 0-63 (decimal) are assigned via RFC 5226 [RFC5226] Standards Action. Values from the inclusive range 64-223 (decimal) are assigned via RFC 5226 Specification Required. Values from the inclusive range 224-255 (decimal) are reserved for RFC 5226 Private Use.

7 Normative References

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

[RFC5226] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.

[RFC5246] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008

Authors' Addresses

Stefan Santesson

3xA Security AB
Bjornstorp 744
247 98 Genarp
Sweden

EMail: sts@aaa-sec.com

Quynh Dang

NIST
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20899-8930
USA

Email: quynh.dang@nist.gov

Full Copyright Statement

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/licenseinfo). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.