INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: August 22, 2010

S. Santesson (3xA Security)

February 18, 2010

Transport Layer Security (TLS) Cached Information Extension
<draft-ietf-tls-cached-info-03.txt>

Status of this Memo

Copyright and License Notice

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.

Table of Contents

1 Introduction 3
2 Cached Information Extension 4
4 Message flow 5
5 Security Considerations 5
6 IANA Considerations 6
7 Normative References 6
Authors' Addresses 7

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, when included in the client hello, 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..2^16-1>;
} 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.

The client MUST NOT include hashes for multiple objects in the same CachedObject structure. If more than one hash is present in the CachedObject structure, they MUST be hashes over the same information object using different hash algorithms.

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. Clients MAY need the ability to cache different values depending on other information in the Client Hello that modify what values the server uses, in particular the Server Name Indication [RFC4366] value.

Servers that receive an extended client hello containing a "cached_information" extension, MAY indicate that they support caching of information objects by including an extension of type "cached_information" with an empty extension_data field in their (extended) server hello.

Following a successful exchange of "cached_information" extensions, the server may replace data objects identified through the client extension with any of the CachedInformationHash values received from the client, which matches the replaced object.

The handshake protocol will proceed using the cached data as if it was 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

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

NOTE: RFC 4366 will be updated by RFC4366bis, currently in IESG process.

Authors' Addresses

Stefan Santesson

3xA Security AB
Bjornstorp 744
247 98 Genarp

Sweden

EMail: sts@aaa-sec.com