INTERNET-DRAFT S. Santesson (AAA-sec)
Intended Status: Proposed Standard
Expires September 2009 March 2009

TLS Cached Certificates Extension
<draft-santesson-tls-certcache-00.txt>

Status of this Memo

Abstract

This document defines a Transport Layer Security (TLS) extension for cached certificates. This extension allows the TLS client to inform a server of a previously cached server certificate path, allowing the server to omit sending an identified certificate chain to the client during the TLS handshake protocol exchange.

1. Introduction

A server certificate sent to the client during a TLS handshake 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 CertCache TLS extension, which may be used by a client to and a server to omit sending known certificate data in the Server Certificate message.

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 [N1].

2 Cached Certs Extension

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

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

The "extension_data" field of this extension SHALL contain "CachedCerts" containing a hash of cached server certificates:

```
struct {
      opaque certificate_hash; <1..2^8-1>
} CachedCerts;
```

The certificate_hash value MUST include at least one hash value calculated over an expected certificate_list element of a server side Certificate message.

The hash algorithm used to generate hash included in certificate_hash MUST be SHA-1.

4 Message flow

In order to allow negotiation to omit certificate data in the Server Certificate message, the client MUST include an extension of type "cached_certs" in the (extended) client hello, which SHALL contain at

least one certificate hash as specified in section 2.

Servers that receive an extended client hello containing a "cached_certs" extension, MAY indicate that they are willing to accept omitting certificate data in the Server Certificate message by including an extension of type "cached_certs" in the (extended) server hello, which SHALL contain a hash received in the cached_certs extension from the client, which is a complete hash calculated over all omitted certificates.

After negotiation of the use of cached certificates has been successfully completed (by exchanging hello messages including "cached_certs" extensions), the server MAY replace the certificate_list element in its Certificate message with the hash included in the cached_certs extension of the server hello message.

All operations of the handshake protocol will be processed as if the hash value carried in the certificate_list element is the actual bits of the server certificate path, with the only exception that all public key operations will be done using the real certificates and associated keys identified by the hash value. For example hash and length values in the Finished message will be calculated over the modified Server Certificate message (with omitted certificate data) that was sent to the client.

5 Security Considerations

The use of hash algorithm in this specification requires reasonable random properties in order to provide unique identifiers. No security threat requires the hash algorithm to have strong collision resistance. Consequently, there is no reason to provide hash agility at the cost of protocol complexity.

6 IANA Considerations

TBD

7 Normative References

[N1] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

TBD

Authors' Addresses

Stefan Santesson
AAA-sec AB
Bjornstorp 744
247 98 Genarp
Sweden

EMail: stefan@aaa-sec.com

Full Copyright Statement