Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: July 11, 2016

A. Popov, Ed.
M. Nystroem
Microsoft Corp.
D. Balfanz
A. Langley
Google Inc.

January 8, 2016

Transport Layer Security (TLS) Extension for Token Binding Protocol Negotiation
draft-ietf-tokbind-negotiation-02

Abstract

This document specifies a Transport Layer Security (TLS) [RFC5246] extension for the negotiation of Token Binding protocol [I-D.ietf-tokbind-protocol] version and key parameters.

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 July 11, 2016.

Copyright Notice

Table of Contents

1. Introduction

In order to use the Token Binding protocol [I-D.ietf-tokbind-protocol], the client and server need to agree on the Token Binding protocol version and the parameters (signature algorithm, length) of the Token Binding key. This document specifies a new TLS extension to accomplish this negotiation without introducing additional network round-trips.

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 [RFC2119].

2. Token Binding Negotiation Client Hello Extension

The client uses the "token_binding" TLS extension to indicate the highest supported Token Binding protocol version and key parameters.

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

The "extension_data" field of this extension contains a "TokenBindingParameters" value.

```
struct {
    uint8 major;
    uint8 minor;
} ProtocolVersion;

enum {
    rsa2048_pkcs1.5(0), rsa2048_pss(1), ecdsap256(2), (255)
} TokenBindingKeyParameters;

struct {
    ProtocolVersion token_binding_version;
    TokenBindingKeyParameters key_parameters_list<1..2^8-1>
} TokenBindingParameters;
```

"token_binding_version" indicates the version of the Token Binding protocol the client wishes to use during this connection. This SHOULD be the latest (highest valued) version supported by the client. [I-D.ietf-tokbind-protocol] describes version {1, 0} of the protocol. Prototype implementations of Token Binding drafts can indicate support of a specific draft version, e.g. {0, 1} or {0, 2}.

"key_parameters_list" contains the list of identifiers of the Token Binding key parameters supported by the client, in descending order of preference.

3. Token Binding Negotiation Server Hello Extension

The server uses the "token_binding" TLS extension to indicate support for the Token Binding protocol and to select the protocol version and key parameters.

The server that supports Token Binding and receives a client hello message containing the "token_binding" extension, will include the "token_binding" extension in the server hello if all of the following conditions are satisfied:

1. The server supports the Token Binding protocol version offered by the client or a lower version.

2. The server finds acceptable Token Binding key parameters on the client’s list.

3. The server is also negotiating Extended Master Secret [RFC7627] and Renegotiation Indication [RFC5746] TLS extensions. This requirement only applies when TLS 1.2 or an older TLS version is used (see security considerations section below for more details).

The server will ignore any key parameters that it does not recognize. The "extension_data" field of the "token_binding" extension is structured the same as described above for the client "extension_data".

"token_binding_version" contains the lower of the Token Binding protocol version offered by the client in the "token_binding" extension and the highest version supported by the server.

"key_parameters_list" contains exactly one Token Binding key parameters identifier selected by the server from the client's list.

4. Negotiating Token Binding Protocol Version and Key Parameters

It is expected that a server will have a list of Token Binding key parameters identifiers that it supports, in preference order. The server MUST only select an identifier that the client offered. The server SHOULD select the most highly preferred key parameters identifier it supports which is also advertised by the client. In the event that the server supports none of the key parameters that the client advertises, then the server MUST NOT include "token_binding" extension in the server hello.

The client receiving the "token_binding" extension MUST terminate the handshake with a fatal "unsupported_extension" alert if any of the following conditions are true:

1. The client did not include the "token_binding" extension in the client hello.

2. "token_binding_version" is higher than the Token Binding protocol version advertised by the client.

3. "key_parameters_list" includes more than one Token Binding key parameters identifier.

4. "key_parameters_list" includes an identifier that was not advertised by the client.

5. TLS 1.2 or an older TLS version is used, but Extended Master Secret [RFC7627] and Renegotiation Indication [RFC5746] TLS extensions are not negotiated (see security considerations section below for more details).

If the "token_binding" extension is included in the server hello and the client supports the Token Binding protocol version selected by the server, it means that the version and key parameters have been negotiated between the client and the server and SHALL be definitive

for the TLS connection. In this case, the client MUST use the negotiated key parameters in the "provided_token_binding" as described in [I-D.ietf-tokbind-protocol].

If the client does not support the Token Binding protocol version selected by the server, then the connection proceeds without Token Binding.

Please note that the Token Binding protocol version and key parameters are negotiated for each TLS connection, which means that the client and server include their "token_binding" extensions both in the full TLS handshake that establishes a new TLS session and in the subsequent abbreviated TLS handshakes that resume the TLS session.

5. IANA Considerations

This document defines a new TLS extension "token_binding", which needs to be added to the IANA "Transport Layer Security (TLS) Extensions" registry.

This document establishes a registry for identifiers of Token Binding key parameters entitled "Token Binding Key Parameters" under the "Token Binding Protocol" heading.

Entries in this registry require the following fields:

- Value: The octet value that identifies a set of Token Binding key parameters (0-255).
- Description: The description of the Token Binding key parameters.
- Specification: A reference to a specification that defines the Token Binding key parameters.

This registry operates under the "Expert Review" policy as defined in [RFC5226]. The designated expert is advised to encourage the inclusion of a reference to a permanent and readily available specification that enables the creation of interoperable implementations using the identified set of Token Binding key parameters.

An initial set of registrations for this registry follows:

Value: 0

Description: rsa2048_pkcs1.5

Specification: this document

Value: 1

Description: rsa2048_pss

Specification: this document

Value: 2

Description: ecdsap256

Specification: this document

6. Security Considerations

6.1. Downgrade Attacks

The Token Binding protocol version and key parameters are negotiated via "token_binding" extension within the TLS handshake. TLS prevents active attackers from modifying the messages of the TLS handshake, therefore it is not possible for the attacker to remove or modify the "token_binding" extension. The signature algorithm and key length used in the TokenBinding of type "provided_token_binding" MUST match the parameters negotiated via "token_binding" extension.

6.2. Triple Handshake Vulnerability in TLS 1.2 and Older TLS Versions

The Token Binding protocol relies on the TLS Exporters [RFC5705] to associate a TLS connection with a Token Binding. The triple handshake attack [TRIPLE-HS] is a known TLS protocol vulnerability allowing the attacker to synchronize exported keying material between TLS connections. The attacker can then successfully replay bound tokens. For this reason, the Token Binding protocol MUST NOT be negotiated with these TLS versions, unless the Extended Master Secret [RFC7627] and Renegotiation Indication [RFC5746] TLS extensions have also been negotiated.

7. Acknowledgements

This document incorporates comments and suggestions offered by Eric Rescorla, Gabriel Montenegro, Martin Thomson, Vinod Anupam, Bill Cox, Nick Harper and others.

8. References

8.1. Normative References

[I-D.ietf-tokbind-protocol]
Popov, A., Nystrom, M., Balfanz, D., and A. Langley, "The Token Binding Protocol Version 1.0", draft-ietf-tokbind-protocol-03 (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <http://www.rfc-editor.org/info/rfc5705>.

[RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, DOI 10.17487/RFC5746, February 2010, <http://www.rfc-editor.org/info/rfc5746>.

[RFC7627] Bhargavan, K., Ed., Delignat-Lavaud, A., Pironti, A., Langley, A., and M. Ray, "Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension", RFC 7627, DOI 10.17487/RFC7627, September 2015, <http://www.rfc-editor.org/info/rfc7627>.

8.2. Informative References

[TRIPLE-HS]
Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Pironti, A., and P. Strub, "Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS. IEEE Symposium on Security and Privacy", 2014.

Authors' Addresses

Andrei Popov (editor)
Microsoft Corp.
USA

Email: andreipo@microsoft.com

Magnus Nystroem
Microsoft Corp.
USA

Email: mnystrom@microsoft.com

Dirk Balfanz
Google Inc.
USA

Email: balfanz@google.com

Adam Langley
Google Inc.
USA

Email: agl@google.com