Token Binding Working Group G. Mandyam
Internet-Draft L. Lundblade
Intended status: Standards Track J. Azen
Expires: April 3, 2017 Qualcomm Technologies Inc.
September 30, 2016

Attested TLS Token Binding
draft-mandyam-tokbind-attest-00

Abstract

Token binding allows HTTP servers to bind bearer tokens to TLS connections. In order to do this, clients or user agents must prove possession of a private key. However, proof-of-possession of a private key becomes truly meaningful to a server when accompanied by an attestation statement. This specification describes extensions to the existing token binding protocol to allow for attestation statements to be sent along with the related token binding messages.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-tokbind-protocol] and [I-D.ietf-tokbind-negotiation] describe a framework whereby servers can leverage cryptographically-bound authentication tokens to verify TLS connections. This is useful for prevention of man-in-the-middle attacks on TLS sessions, and provides a mechanism by which identity federation systems can be leveraged by a relying party to verify a client based on proof-of-possession of a private key.

Once the use of token binding is negotiated as part of the TLS handshake, an application layer message (the Token Binding message) may be sent from the client to the relying party whose primary purpose is to encapsulate a signature over a value associated with the current TLS session (Exported Key Material, i.e. EKM - see [I-D.ietf-tokbind-protocol]).

Proof-of-possession of a private key is useful to a relying party, but the associated signature in the Token Binding message does not provide an indication as to how the private key is stored and in what kind of environment the associated cryptographic operation takes place. This information may be required by a relying party in order to satisfy requirements regarding client platform integrity. Therefore, attestations are sometimes required by relying parties in order for them to accept signatures from clients. As per the definition in [I-D.birkholz-tuda], "remote attestation describes the attempt to determine the integrity and trustworthiness of an endpoint -- the attestee -- over a network to another endpoint -- the verifier -- without direct access." Attestation statements are therefore widely used in any server verification operation that leverages client cryptography.

TLS token binding can therefore be enhanced with remote attestation statements. The attestation statement can be used to augment Token Binding message. Moreover, the attestation may optionally be included by the client as part of TLS negotiation [I-D.ietf-tokbind-negotiation]. This could be used by a relying party for several different purpose, including (1) to determine whether to accept token binding messages from the associated client, or (2) require an additional mechanism for binding the TLS connection to an authentication operation by the client. In addition, the attestation can accompany the token binding message as a separate application protocol message.

2. Attestation Enhancement to TLS Token Binding Negotiation

[I-D.ietf-tokbind-negotiation] provides the necessary extensions tothe TLS handshake that allows for TLS token binding to be negotiated as part of any connection. It is necessary that the TLS client and server agree on the parameters that attach to the token binding session, and these extensions to TLS messaging make that possible.

A new TLS extension would be defined, "attested token binding", and used in the client hello.

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

Based on this extension, the "TokenBindingParameters" extension data is modified to include attestation:

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

enum {
  (255)
} TokenBindingKeyParameters

enum {
  packed(0), tpmv1 (1), tpmv2 (2),(255)
} AttestationType

struct {
  ProtocolVersion token_binding_version;
  AttestationType token_binding_attestation_type;
  TokenBindingKeyParameters key_parameters_list<1...2^8-1>;
  attestation_length_bytes<1..2^8-1>;
  attestation_data<1..2^(8*attestation_length_bytes)>
} TokenBindingParameters;
```

3. Attestation Enhancement to TLS Token Binding Message

The attestation statement can be processed 'in-band' as part of the Token Binding Message itself. However, many attestation statements include a signature. Therefore including attestation data as part of the Token Binding Message does not appear to provide any discernible advantage, while introducing additional complexity in server processing of the Token Binding message. Therefore a new HTTP header field is defined to accompany the Sec-Token-Binding header defined in [I-D.ietf-tokbind-https]:

```
Sec-Token-Binding-Attestation:  <base64url-encoded AttestationData>
```

The attestation data itself is determined as:

```
enum {
  packed(0), tpmv1 (1), tpmv2 (2),(255)
} AttestationType;
struct {
  AttestationType token_binding_attestation_type;
  attestation_length_bytes<1..2^8-1>;
  attestation_data<1..2^(8*attestation_length_bytes)>
} AttestationData;
```

4. Attestation Suppression

It may be desirable to suppress attestation after the initial TLS handshake when the attestation is originally sent. This can be desirable if the attestation statement does not change over time. In this case, the TLS extension to be used would be "attested token binding with suppression", and would be used in the client hello.

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

The "TokenBindingParameters" extension data is as defined previously. However, after the initial TLS handshake, the Sec-Token-Binding-Attestation header will not be sent in ensuing HTTP transactions corresponding to this TLS negotiation.

5. Example - Platform Attestation for Anomaly Detection

An example of where a platform-based attestation is useful can be for remote attestation based on client traffic anomaly detection. Many network infrastructure deployments employ network traffic monitors for anomalous pattern detection. Examples of anomalous patterns detectable in the TLS handshake could be unexpected cipher suite negotiation for a given source/destination pairing. In this case, it may be desirable for a client-enhanced attestation reflecting for instance that an expected offered cipher suite in the client hello message is present or the originating browser integrity is intact through a hash over the browser application package. This attestation could also be delivered as part of an application-encapsulated message, but this attestation may not be available to network traffic monitors that cannot decrypt application-layer traffic. Due to the presence of the remote attestation in the client hello, a network traffic monitor can verify the attestation and potentially emit alerts based on an unexpected attestation.

6. IANA Considerations

This memo includes no request to IANA.

7. References

7.1. Normative References

[I-D.ietf-tokbind-https]
Popov, A., Nystrom, M., Balfanz, D., Langley, A., and J. Hodges, "Token Binding over HTTP", draft-ietf-tokbind-https-05 (work in progress), July 2016.

[I-D.ietf-tokbind-negotiation]
Popov, A., Nystrom, M., Balfanz, D., and A. Langley, "Transport Layer Security (TLS) Extension for Token Binding Protocol Negotiation", draft-ietf-tokbind-negotiation-03 (work in progress), July 2016.

[I-D.ietf-tokbind-protocol]
Popov, A., Nystrom, M., Balfanz, D., Langley, A., and J. Hodges, "The Token Binding Protocol Version 1.0", draft-ietf-tokbind-protocol-08 (work in progress), July 2016.

7.2. Informative References

[I-D.birkholz-tuda]
Fuchs, A., Birkholz, H., McDonald, I., and C. Bormann, "Time-Based Uni-Directional Attestation", draft-birkholz-tuda-02 (work in progress), July 2016.

Authors' Addresses

Giridhar Mandyam
Qualcomm Technologies Inc.
5775 Morehouse Drive
San Diego, California 92121
USA

Phone: +1 858 651 7200
Email: mandyam@qti.qualcomm.com

Laurence Lundblade
Qualcomm Technologies Inc.
5775 Morehouse Drive
San Diego, California 92121
USA

Phone: +1 858 658 3584
Email: llundbla@qti.qualcomm.com

Jon Azen
Qualcomm Technologies Inc.
5775 Morehouse Drive
San Diego, California 92121
USA

Phone: +1 858 651 9476
Email: jazen@qti.qualcomm.com