Internet Engineering Task Force	B. Campbell
Internet-Draft	Ping Identity
Intended status: Standards Track	January 11, 2017
Expires: July 15, 2017	

HTTPS Token Binding and TLS Terminating Reverse Proxies

draft-campbell-tokbind-tls-term-00

Abstract

This document defines an HTTP header field that enables a TLS terminating reverse proxy to convey the information a backend server needs in order for it to process and validate a Token Binding Message sent by the client.

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 15, 2017.

Copyright Notice

Table of Contents

1. Introduction

Token Binding over HTTP [I-D.ietf-tokbind-https] provides a mechanism that enables HTTP servers to cryptographically bind cookies and other security tokens to TLS [RFC5246] connections. When Token Binding is negotiated in the TLS handshake [I-D.ietf-tokbind-negotiation] the client sends an encoded Token Binding Message [I-D.ietf-tokbind-protocol] as a header in each HTTP request, which proves possession of one or more private keys held by the client. The public portion of the keys are represented in the Token Binding IDs of the Token Binding Message and for each one there is a signature over some data, which includes the exported keying material [RFC5705] of the TLS connection. An HTTP server issuing cookies or other security tokens can associate them with the Token Binding ID, which ensures those tokens cannot be used successfully over a different TLS connection or by a different client than the one to which they were issued.

A fairly common deployment architecture for HTTPS applications is to have the backend HTTP application servers sit behind a reverse proxy that terminates TLS. The proxy is accessible to the internet and dispatches client requests to the appropriate backend server within a private network. The backend servers are not directly accessible outside the private network and are only reachable through the reverse proxy. The details of such deployments are typically opaque to clients who make requests to the proxy server and see responses as though they originated from the proxy server itself. TLS connections for HTTPS are established between each client and the reverse proxy server.

Token Binding facilitates a binding of security tokens to a key held by the client by way of the TLS connection between that client and the sever. In a TLS terminating reverse proxy deployment, however, the TLS connection is between the client and the proxy while the backend server is likely the system that will issue security tokens. Additional steps are therefore needed to enable the use of Token Binding in such deployment architectures. In the absence of a standardized approach, different implementations will will address it differently, which will make interoperability between implementation difficult or impossible without complex configurations or custom integrations.

This document standardizes an HTTP header field named Token-Binding-Context that a TLS terminating reverse proxy adds to requests that it sends to the backend servers. The value of the header contains the information from its connection with the client that is necessary for the backend server to process and validate the Token Binding Message also in the request. The usage of the header, both the reverse proxy adding it and the application server using it rather than information from its inbound connection, are to be configuration options of the respective systems as they will not always be applicable.

1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. The Token-Binding-Context HTTP Header Field

When configured to do so, a reverse proxy that terminates TLS and negotiates Token Binding over HTTP [I-D.ietf-tokbind-https] with a client adds a Token-Binding-Context HTTP header field to the request that is dispatched to a backend server.

The Token-Binding-Context is a single HTTP header field-value as defined in Section 3.2 of [RFC7230], which MUST NOT have a list of values or occur multiple times in a request. The Token-Binding-Context header is only for use in HTTP requests and MUST NOT to be used in HTTP responses. The header field value is defined in ABNF [RFC5234] syntax as:

```
Token-Binding-Context = EncodedTBContextMessage
EncodedTBContextMessage = 47*( DIGIT / ALPHA /  "-" / "_" )

DIGIT = <Defined in Section B.1 of [RFC5234]>
ALPHA = <Defined in Section B.1 of [RFC5234]>
```

The header field name is Token-Binding-Context and its value is a base64url encoding of a Token Binding Context Message using the URL- and filename-safe character set described in Section 5 of [RFC4648], with all trailing pad characters '=' omitted and without the inclusion of any line breaks, whitespace, or other additional characters.

The Token Binding Context Message is a byte sequence that contains the concatenation of the negotiated Token Binding Protocol Version and Key Parameters as well as the exported keying material (EKM) from the TLS connection between the client and reverse proxy. The first two bytes are the ProtocolVersion, as defined in Section 2 of [I-D.ietf-tokbind-negotiation], that the reverse proxy negotiated with the client. The third byte is the negotiated TokenBindingKeyParameters (also defined in Section 2 of [I-D.ietf-tokbind-negotiation]). The remaining 32 or more bytes are the EKM from the TLS connection between the client and the reverse proxy, as defined in Section 3.3 of [I-D.ietf-tokbind-protocol].

For example, below is an encoded Token Binding Context Message indicating version 1.0 of the protocol, ecdsap256(2) key parameters, and a 32 byte EKM:

```
AQACltcPRPoACC9N9lW5ESCvw4e6_6oISR38bwc2ddz7fFs4i
```

A backend server that receives a request from a trusted reverse proxy containing the Token-Binding-Context and Sec-Token-Binding headers decodes the Token Binding Context Message and uses its content to validate the encoded Token Binding Message as described in Section 2 of Token Binding over HTTP [I-D.ietf-tokbind-https] in place of information that otherwise would have come from the TLS connection.

Reverse proxies MUST only add the Token-Binding-Context header when explicitly configured to do so and MUST only dispatch requests containing it to trusted backend servers. Any occurrence of the Token-Binding-Context header in the request from the client MUST be removed or overwritten before forwarding the request. Backend servers MUST only accept the Token-Binding-Context header when explicitly configured to do so and only from trusted reverse proxies.

Forward proxies and other intermediaries MUST NOT add the Token-Binding-Context header to requests.

3. Security Considerations

The Token-Binding-Context header enables a reverse proxy and backend server to function together as though they are single logical deployment of HTTPS Token Binding. Use of the header outside that intended use case, however, may undermine the protections afforded by Token Binding. Therefore steps must be taken to prevent unintended use, both in sending the header and in relying on its value.

Producing and consuming the Token-Binding-Context header should be a configurable option, respectively, in a reverse proxy and backend server (or individual application in that server). The default configuration for both should be to not use the Token-Binding-Context header thus requiring an "opt-in" to its usage.

Reverse proxies should only add the header to requests that are forwarded to trusted backend servers. Otherwise a legitimate EKM value might be disclosed to an unintended party.

Backend servers should only accept the header from trusted reverse proxies. And reverse proxies need to sanitize the incoming request before forwarding it on by removing or overwriting any existing instances of the Token-Binding-Context header. Otherwise arbitrary clients can control the EKM value as seen and used by the backend server.

The communication between a reverse proxy and backend server needs to be secured against eavesdropping and modification by unintended parties.

The configuration options and request sanitization are necessarily functionally of the respective servers. The other requirements can be met in a number of ways, which will vary based on specific deployments. The communication between a reverse proxy and backend server, for example, might be over a mutually authenticated TLS with the insertion and consumption of the Token-Binding-Context header occurring only on for that connection. Alternatively the network topology might dictate a private network such that the backend application is only able to accept requests from the reverse proxy and the proxy can only make requests to that server. Other deployments that meet the requirements set forth herein are also possible.

4. IANA Considerations

This document specifies the Token-Binding-Context HTTP header field, registration of which is requested in the "Permanent Message Header Field Names" registry defined in [RFC3864].

- Header Field Name: Token-Binding-Context
- Applicable protocol: http
- Status: standard
- Author/change Controller: IETF
- Specification Document(s): Section 2 of [[this specification]]

5. Normative References

[I-D.ietf-tokbind-https] Popov, A., Nystrom, M., Balfanz, D., Langley, A. and J. Hodges, "Token Binding over HTTP", Internet-Draft draft-ietf-tokbind-https-07, November 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", Internet-Draft draft-ietf-tokbind-negotiation-06, November 2016.

[I-D.ietf-tokbind-protocol] Popov, A., Nystrom, M., Balfanz, D., Langley, A. and J. Hodges, "The Token Binding Protocol Version 1.0", Internet-Draft draft-ietf-tokbind-protocol-11, November 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.

[RFC3864] Klyne, G., Nottingham, M. and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008.

[RFC5246]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008.
[RFC5705]	Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010.
[RFC7230]	Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014.

Appendix A. Acknowledgements

The author would like to thank the following people for their contributions to the specification: Dirk Balfanz, John Bradley, Subodh Iyengar, Andrei Popov, Martin Thomson and others (please let me know, if you've contributed and I've forgotten you).

Appendix B. Open Issues

- might need this...

Appendix C. Document History

[[to be removed by the RFC Editor before publication as an RFC]]

draft-campbell-tokbind-tls-term-00

- Initial draft based on 'consensus to work on the problem' at the Seoul meeting. Slides and minutes from the meeting, respectively: https://www.ietf.org/proceedings/97/slides/slides-97-tokbind-reverse-proxies-00.pdf https://www.ietf.org/proceedings/97/minutes/minutes-97-tokbind-01.txt

Author's Address

Brian Campbell
Ping Identity
EMail: brian.d.campbell@gmail.com