           HTTPS Token Binding with TLS Terminating Reverse Proxies
                       draft-ietf-tokbind-ttrp-00

Abstract

   This document defines common HTTP header fields that enable a TLS
   terminating reverse proxy to convey information about the validated
   Token Binding Message sent by the client to a backend server, which
   enables that backend server to bind, or verify the binding of,
   cookies and other security tokens to the client's Token Binding key.

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 January 22, 2018.

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 a key held by the browser or other HTTP client,
possession of which is proven on the TLS [RFC5246] connections over
which the tokens are used. 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 or protected 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 server. In a deployment where TLS is terminated by a reverse
proxy, however, the TLS connection is between the client and the
proxy while the backend server is likely the system that will issue
cookies or other 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 address it differently, which will make
interoperability between implementation difficult or impossible
without complex configurations or custom integrations.

This document standardizes HTTP header field names that a TLS
terminating reverse proxy (TTRP) adds to requests that it sends to
the backend servers. The headers contain the information from the
validated Token Binding Message sent by the client to the proxy with
the "Sec-Token-Binding" header, thus enabling the backend server to
bind, or verify the binding of, cookies and other security tokens to
the client's Token Binding key. The usage of the headers, both the
reverse proxy adding it and the application server using them to bind
cookies or other tokens, 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. HTTP Header Fields and Processing Rules

2.1. Token Binding ID HTTP Header Fields

The Token Binding Protocol [I-D.ietf-tokbind-protocol] recommends
that implementations make Token Binding IDs available to the
application as opaque byte sequences, enabling those applications to
use the Token Binding IDs when generating and verifying bound tokens.
In the context of a TLS terminating reverse proxy (TTRP) deployment,
the provided and referred Token Binding IDs are made available to the
backend application as the "Provided-Token-Binding-ID" and "Referred-
Token-Binding-ID" HTTP headers respectively. The value of both
headers is an "EncodedTokenBindingID", for which the ABNF [RFC5234]
syntax is shown in Figure 1 below. "EncodedTokenBindingID" 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. An "EncodedTokenBindingID" is only for use in
HTTP requests and MUST NOT to be used in HTTP responses.

```
EncodedTokenBindingID = *( DIGIT / ALPHA /  "-" / "_" )

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

Figure 1: Encoded Token Binding ID Header ABNF

The value of an "EncodedTokenBindingID" is a base64url encoding of
the TokenBindingID byte sequence (see section 3 of
[I-D.ietf-tokbind-protocol]) using the URL and filename safe alphabet
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.

2.2. Processing Rules

This section defines the applicable processing rules for a TLS
terminating reverse proxy (TTRP) and backend server(s) to provide
server side support of Token Binding over HTTP
[I-D.ietf-tokbind-https] using the HTTP headers described in
Section 2.1. Use of the technique is to be a configuration or
deployments option and the processing rules described herein are for
servers operating with that option enabled.

A TTRP negotiates the use of Token Binding with the client per
[I-D.ietf-tokbind-negotiation] and validates the Token Binding
Message as defined in The Token Binding Protocol
[I-D.ietf-tokbind-protocol] and Token Binding over HTTP
[I-D.ietf-tokbind-https] for each HTTP request on the underlying TLS
connection. Requests with a valid Token Binding Message (and meeting
any other authorization or policy requirements of the TTRP) are
dispatched to the backend server with the following modifications.

1. The "Sec-Token-Binding" header in the original incoming request
    MUST be removed from the request that is dispatched to the
    backend server.

2. The Token Binding ID of the provided Token Binding of the Token
    Binding Message MUST be placed in the "Provided-Token-Binding-ID"
    header field of the dispatched request using the format defined
    in Section 2.1.

3. If the Token Binding Message contains a referred Token Binding,
    the referred Token Binding ID MUST be placed in the "Referred-
    Token-Binding-ID" header field of the dispatched request using
    the format defined in Section 2.1. Otherwise, the "Referred-
    Token-Binding-ID" header field MUST NOT be present in the
    dispatched request.

4. Any occurrence of the "Provided-Token-Binding-ID" or "Referred-
    Token-Binding-ID" header in the original incoming request MUST be
    removed or overwritten before forwarding the request.

Requests made over a TLS connection where the use of Token Binding
was not negotiated MUST be sanitized by removing any occurrences of
the "Provided-Token-Binding-ID" and "Referred-Token-Binding-ID"
header fields prior to dispatching the request to the backend server.

Forward proxies and other intermediaries MUST NOT add the "Provided-
Token-Binding-ID" or "Referred-Token-Binding-ID" header to requests.

2.3. Examples

Extra line breaks and whitespace have been added to the following
examples for display and formatting purposes only.

2.3.1. Provided Token Binding ID

The following "Sec-Token-Binding" header is from an HTTP request made
over a TLS connection between the client and the TTRP where the use
of Token Binding has been negotiated (The base64url-encoded
representation of the exported keying material, which can be used to
validate the Token Binding Message, for that connection is
"AYVUayPTP9RmELNpGjFl6Ykm2CUx7pUMxe35yb11dgU"). The encoded Token
Binding Message has the provided Token Binding the client uses with
the server.

```
 Sec-Token-Binding: AIkAAgBBQKzyIrmcY_YCtHVoSHBut69vrGfFdy1_YKTZfFJv
   6BjrZsKD9b9FRzSBxDs1twTqnAS71M1RBumuihhI9xqxXKkAQEtxe4jeUJU0WezxlQ
   XWVSBFeHxFMdXRBIH_LKOSAuSMOJ0XEw1Q8DE248qkOiRKzw3KdSNYukYEPmO21bQi
   3YYAAA
```

                Figure 2: Header in HTTP Request to TTRP

After validating the Token Binding Message, the TTRP removes the
"Sec-Token-Binding" header and adds the following "Provided-Token-
Binding-ID" header with the provided Token Binding ID to the request
that is dispatched to the backend server.

    Provided-Token-Binding-ID: AgBBQKzyIrmcY_YCtHVoSHBut69vrGfFdy1_YKTZ
      fFJv6BjrZsKD9b9FRzSBxDs1twTqnAS71M1RBumuihhI9xqxXKk

            Figure 3: Header in HTTP Request to Backend Server

2.3.2. Provided and Referred Token Binding IDs

   The following "Sec-Token-Binding" header is from an HTTP request made
   over a TLS connection between the client and the TTRP where the use
   of Token Binding has been negotiated (The base64url-encoded
   representation of the exported keying material, which can be used to
   validate the Token Binding Message, for that connection is
   "wEWWCP1KPxfq-QL4NxYII_P4ti_9YYqrTpGs28BZEqE"). The encoded Token
   Binding Message has the provided Token Binding the client uses with
   the server as well as the referred Token Binding that it uses with a
   different server.

     Sec-Token-Binding: ARIAAgBBQCfsI1D1sTq5mvT_2H_dihNIvuHJCHGjHPJchPav
       NbGrOo26-2JgT_IsbvZd4daDFbirYBIwJ-TK1rh8FzrC-psAQMyYIqXj7djGPev1dk
       jV9XxLYGCyqOrBVEtBHrMUCeo22ymLg3OiFcl_fmOPxJbjxI6lKcF0lyfy-dSQmPIe
       zQ0AAAECAEFArPIiuZxj9gK0dWhIcG63r2-sZ8V3LX9gpNl8Um_oGOtmwoP1v0VHNI
       HEOzW3BOqcBLvUzVEG6a6KGEj3GrFcqQBAHQm0pzgUTXKLRamuKE1pmmP9I3UBVpoe
       1DBCe9H2l1VPpsImakUa6crAqZ-0CGBmji7bYzQogpKcyxTTFk5zdwAA

                Figure 4: Header in HTTP Request to TTRP

   After validating the Token Binding Message, the TTRP removes the
   "Sec-Token-Binding" header and adds the following "Provided-Token-
   Binding-ID" and "Referred-Token-Binding-ID" headers, with the
   provided and referred Token Binding IDs respectively, to the request
   that is dispatched to the backend server.

     Provided-Token-Binding-ID: AgBBQCfsI1D1sTq5mvT_2H_dihNIvuHJCHGjHPJc
       hPavNbGrOo26-2JgT_IsbvZd4daDFbirYBIwJ-TK1rh8FzrC-ps
     Referred-Token-Binding-ID: AgBBQKzyIrmcY_YCtHVoSHBut69vrGfFdy1_YKTZ
       fFJv6BjrZsKD9b9FRzSBxDs1twTqnAS71M1RBumuihhI9xqxXKk

            Figure 5: Headers in HTTP Request to Backend Server

3. Security Considerations

   The headers described herein enable a reverse proxy and backend
   server to function together as though they are single logical server
   side deployment of HTTPS Token Binding. Use of the headers 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 headers and in relying on their
   value.

Producing and consuming the headers 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 headers thus requiring an "opt-in" to the functionality.

Reverse proxies SHOULD only add the headers to requests that are forwarded to trusted backend servers.

Backend servers MUST only accept the headers from trusted reverse proxies. And reverse proxies MUST sanitize the incoming request before forwarding it on by removing or overwriting any existing instances of the headers. Otherwise arbitrary clients can control the header values 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 headers occurring only on 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

4.1. HTTP Message Header Field Names Registration

This document specifies the following new HTTP header fields, registration of which is requested in the "Permanent Message Header Field Names" registry defined in [RFC3864].

o Header Field Name: "Provided-Token-Binding-ID"
o Applicable protocol: HTTP
o Status: standard
o Author/change Controller: IETF
o Specification Document(s): [[this specification]]

o Header Field Name: "Referred-Token-Binding-ID"
o Applicable protocol: HTTP
o Status: standard

o Author/change Controller: IETF
o Specification Document(s): [[this specification]]

[5](). References

[5.1](). 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-09 (work in progress), April 2017.

[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-08 (work in progress), April 2017.

[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-14 (work in progress), April 2017.

[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>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data
          Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006,
          <http://www.rfc-editor.org/info/rfc4648>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
          Specifications: ABNF", STD 68, RFC 5234,
          DOI 10.17487/RFC5234, January 2008,
          <http://www.rfc-editor.org/info/rfc5234>.

[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>.

   [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer
              Protocol (HTTP/1.1): Message Syntax and Routing",
              RFC 7230, DOI 10.17487/RFC7230, June 2014,
              <http://www.rfc-editor.org/info/rfc7230>.

5.2. Informative References

   [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration
              Procedures for Message Header Fields", BCP 90, RFC 3864,
              DOI 10.17487/RFC3864, September 2004,
              <http://www.rfc-editor.org/info/rfc3864>.

Appendix A. Acknowledgements

   The author would like to thank the following people for their various
   contributions to the specification: Vinod Anupam, Dirk Balfanz, John
   Bradley, Jeff Hodges, Subodh Iyengar, Leif Johansson, Yoav Nir,
   Andrei Popov, Eric Rescorla, Piotr Sikora, Martin Thomson, Hans
   Zandbelt and others (please let me know, if you've contributed and
   I've forgotten you).

Appendix B. Open Issues

   o During discussions at a side meeting in Chicago (IETF 98) there
      seemed to be general support for having the TTRP rename the "Sec-
      Token-Binding" header to something else and pass the full original
      EncodedTokenBindingMessage to the backend server via a different
      header (maybe "TTRP-Token-Binding" or something) in addition to
      the "Provided-Token-Binding-ID" and if applicable the "Referred-
      Token-Binding-ID" headers defined herein. The idea was largely
      that the backend server "might need it for something" so pass the
      whole thing along just in case. However, as I sat down to write
      this draft, I couldn't bring myself to add it in the main text.
      On thinking about it more, it feels inefficient/duplicative and
      rather inelegant. And without the EKM, much of the data not
      already made available via the Token Binding IDs is meaningless
      (e.g. the signature value). Data in TokenBinding.extensions, if
      extensions are present, might be useful to the backend server.
      But might also only be useful/meaningful at the TTRP where the
      initial TLS connection is terminated. I really don't know.
      Perhaps any extensions, if present, should be passed to the
      backend via different header(s)? Or maybe it would be more
      appropriate to not attempt to cover TokenBinding.extensions in
      this document and defer to the definition of individual extensions
      to say how/if they are to be handled in a TTRP type deployment?

Appendix C. **Document History**

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

   draft-ietf-tokbind-ttrp-00

   o Initial WG draft from draft-campbell-tokbind-ttrp.

   draft-campbell-tokbind-ttrp-01

   o Minor editorial fixes.

   o Add to the Acknowledgements.

   draft-campbell-tokbind-ttrp-00

   o Initial draft based on 'consensus to work on the problem' from the
      Seoul meeting [1][2] and reflecting the consensus approach from
      discussions at the Chicago meeting [3].

      [1] https://www.ietf.org/proceedings/97/minutes/minutes-97-
      tokbind-01.txt (minutes from Seoul)
      [2] https://www.ietf.org/proceedings/97/slides/slides-97-tokbind-
      reverse-proxies-00.pdf (slides from Seoul)
      [3] https://mailarchive.ietf.org/arch/msg/
      unbearable/_ZHI8y2Vs5WMP8VMRr7zroo_sNU (summary of discussion)

Author's Address

   Brian Campbell
   Ping Identity

   Email: brian.d.campbell@gmail.com