HTTPS Token Binding with TLS Terminating Reverse Proxies

https://tools.ietf.org/html/draft-campbell-tokbind-ttrp-00

Brian Campbell

from IETF 93 and as seen on https://www.ietf.org/meeting/99/index.html

IETF 99
Prague
July 2017

Problem Statement

- HTTPS application deployments often have TLS ‘terminated’ by a reverse proxy (TTRP) sitting in front of the actual application
- For applications in such deployments to take advantage of token binding, some information needs to be communicated from the TLS layer to the application (in the general case anyway)
- In the absence of a standard means of doing this, different implementations will do it differently
  - Terrible for interoperability
  - A boon to unneeded complexity
  - Improved opportunity to get things wrong
  - i.e. client certificate authentication

Confirmation Bias...

- I’ve been out spreading the good word of Token Binding
- Often the question of “what if my application isn't the piece that does TLS?” comes up

Token Binding
Enables a long-lived binding via client generated public-private key pair used to sign TLS exported keying material and sent as an HTTP header
CLOUD IDENTITY SUMMIT 2017

A Short History

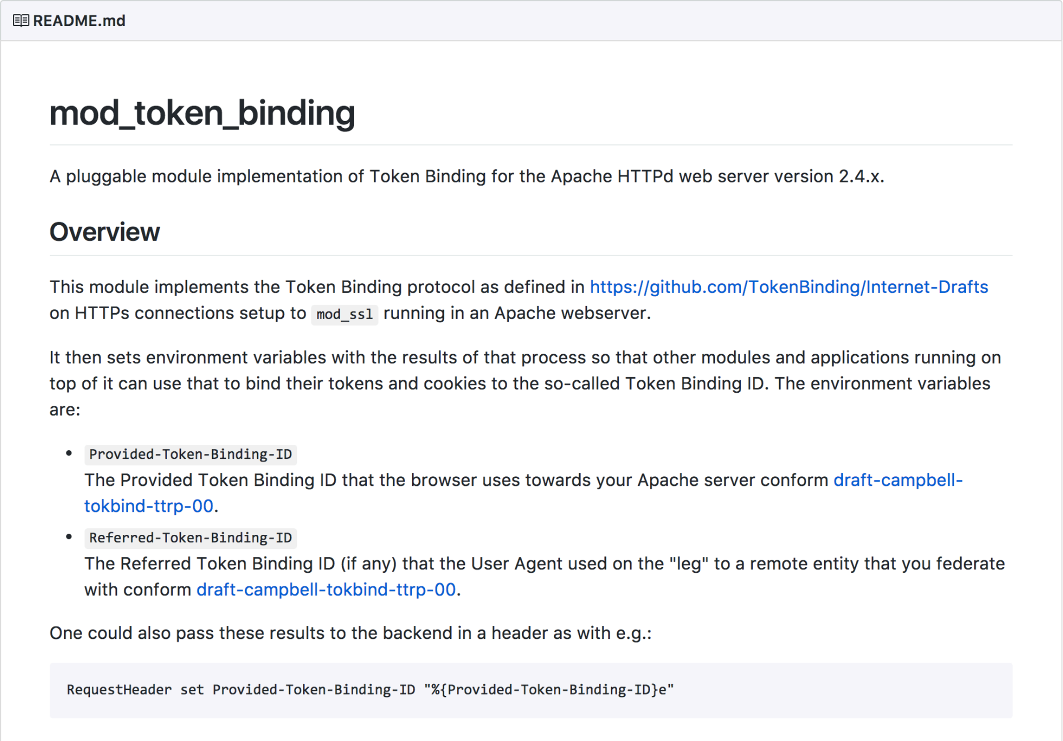
- IETF 97 Seoul: 'consensus to work on the problem’
  - Two general approaches possible:
    - Expose Token Binding ID(s)
    - Expose EKM
- draft-campbell-tokbind-tls-term-00 exposes EKM to backend as header
- TTRP acronym coined by =JeffH
- Received some pushback on approach (primarily from implementers working with NGINX and Apache)
- IETF 98 Chicago: rushed & cut short in main session due to time
  - But held an open side meeting later in the week
    - That group clearly favored approach of exposing Token Binding IDs
- draft-campbell-tokbind-ttrp-00 exposes Token Binding IDs to backend as headers

HTTPS Token Binding with TLS Terminating Reverse Proxies

draft-campbell-tokbind-ttrp-00

- Defines HTTP headers that enable a TTRP and backend server to function together as a single logical server side deployment of HTTPS Token Binding
- TTRP validates the TokenBindingMessage from the `Sec-Token-Binding` header and removes it from dispatched request
- `Provided-Token-Binding-ID` header with base64url encoded provided TokenBindingID added to dispatched request
- `Referred-Token-Binding-ID` header with encoded referred TokenBindingID (if applicable) added to dispatched request
- Trust between the TTRP and backend server
- TTRP required to sanitize headers
- Original TokenBindingMessage not provided to backend

… and Running Code



README.md

mod_token_binding

A pluggable module implementation of Token Binding for the Apache HTTPd web server version 2.4.x.

Overview

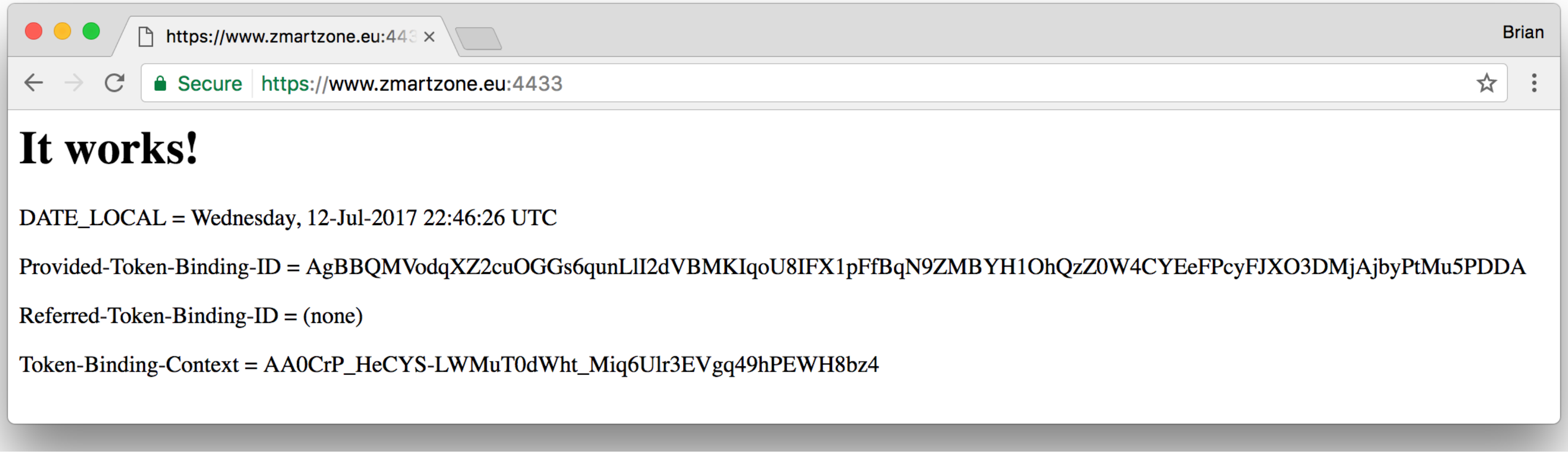
This module implements the Token Binding protocol as defined in https://github.com/TokenBinding/Internet-Drafts on HTTPs connections setup to `mod_ssl` running in an Apache webserver.

It then sets environment variables with the results of that process so that other modules and applications running on top of it can use that to bind their tokens and cookies to the so-called Token Binding ID. The environment variables are:

- `Provided-Token-Binding-ID`
  The Provided Token Binding ID that the browser uses towards your Apache server conform draft-campbell-tokbind-ttrp-00.
- `Referred-Token-Binding-ID`
  The Referred Token Binding ID (if any) that the User Agent used on the "leg" to a remote entity that you federate with conform draft-campbell-tokbind-ttrp-00.

One could also pass these results to the backend in a header as with e.g.:

```
RequestHeader set Provided-Token-Binding-ID "%{Provided-Token-Binding-ID}e"
```



It works!

DATE_LOCAL = Wednesday, 12-Jul-2017 22:46:26 UTC

Provided-Token-Binding-ID = AgBBQMVodqXZ2cuOGGs6qunLlI2dVBMKIqoU8IFX1pFfBqN9ZMBYH1OhQzZ0W4CYEeFPcyFJXO3DMjAjbyPtMu5PDDA

Referred-Token-Binding-ID = (none)

Token-Binding-Context = AA0CrP_HeCYS-LWMuT0dWht_Miq6Ulr3EVgq49hPEWH8bz4

Next Steps

- The people are waiting this…

… I'd ask that the WG consider a Call for Adoption

from IETF 93