HTTPS Token Binding with TLS Terminating Reverse Proxies

Brian Campbell

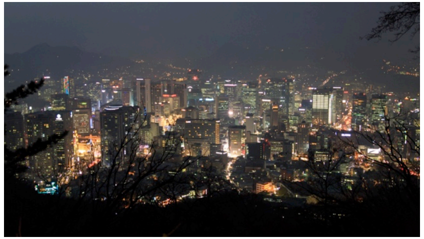
IETF 100
Singapore
November 2017

draft-ietf-tokbind-ttrp

https://tools.ietf.org/html/draft-ietf-tokbind-ttrp-01

Problem Statement

- HTTPS application deployments often have TLS ‘terminated’ by a reverse proxy in front of the actual application
  - products, open source, services
- 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



A Brief 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 the backend as header
- TTRP acronym coined by =JeffH for **T**LS **T**erminating **R**everse **P**roxy
- 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 announced and 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
- draft-campbell-tokbind-ttrp-01 just editorial

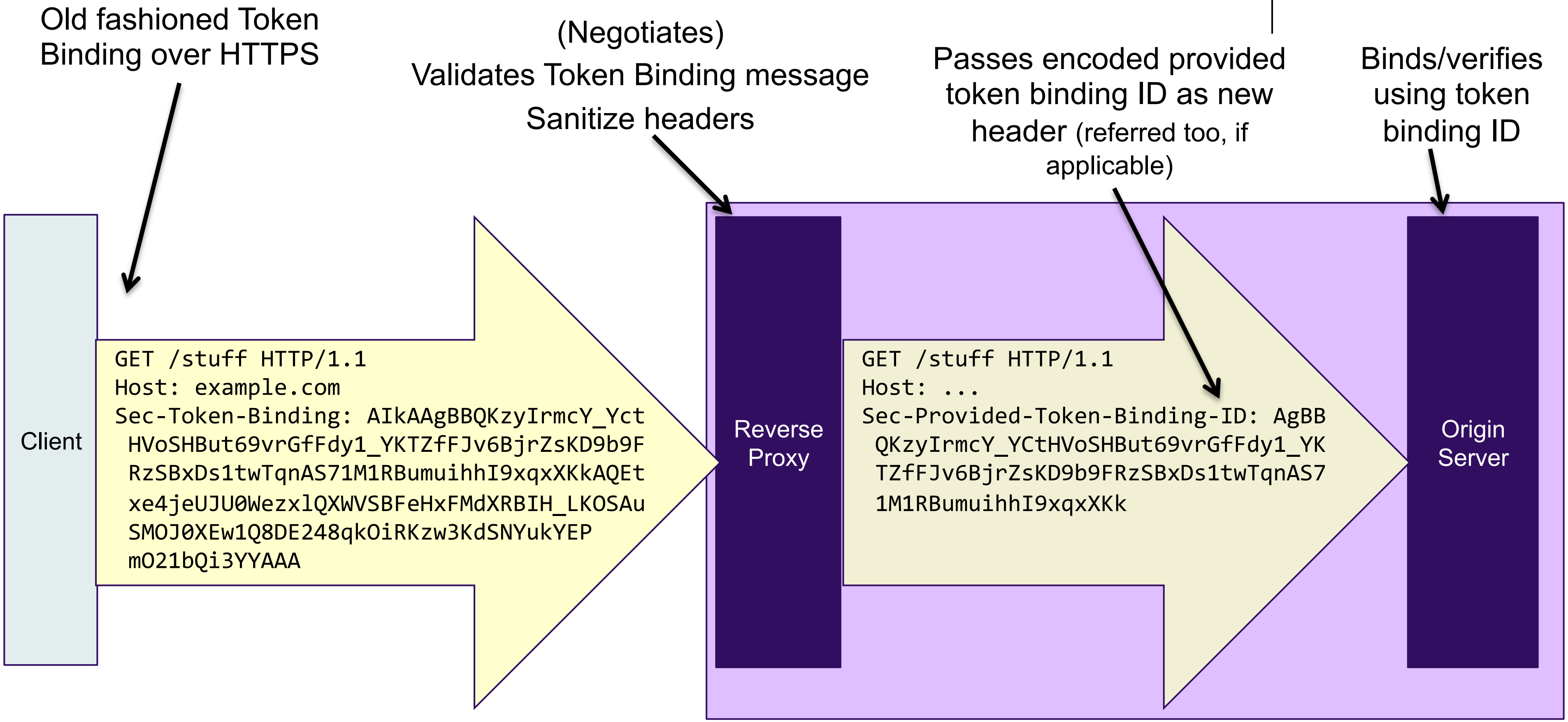
A Brief History cont.

- (shortly after) IETF 99 Prague: Adopted as WG document
- draft-ietf-tokbind-ttrp-01 added `Sec-` prefix to headers

Details of draft-ietf-tokbind-ttrp-01

- 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
- `Sec-Provided-Token-Binding-ID` header with base64url encoded provided TokenBindingID added to dispatched request
- `Sec-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

A Picture is (maybe) Worth a Thousand Words



Old fashioned Token Binding over HTTPS

(Negotiates)
Validates Token Binding message
Sanitize headers

Passes encoded provided token binding ID as new header (referred too, if applicable)

Binds/verifies using token binding ID

```
GET /stuff HTTP/1.1
Host: example.com
Sec-Token-Binding: AIkAAgBBQKzyIrmcY_Yct
 HVoSHBut69vrGfFdy1_YKTZfFJv6BjrZsKD9b9F
 RzSBxDs1twTqnAS71M1RBumuihhI9xqxXKkAQEt
 xe4jeUJU0WezxlQXWVSBFeHxFMdXRBIH_LKOSAu
 SMOJ0XEw1Q8DE248qkOiRKzw3KdSNYukYEP
 mO21bQi3YYAAA
```

```
GET /stuff HTTP/1.1
Host: ...
Sec-Provided-Token-Binding-ID: AgBB
 QKzyIrmcY_YCtHVoSHBut69vrGfFdy1_YK
 TZfFJv6BjrZsKD9b9FRzSBxDs1twTqnAS7
 1M1RBumuihhI9xqxXKk
```

The Elephant in the Room

- Concern expressed in Prague about header sanitization as means to prevent client injection
  - doesn’t fail safe, if improperly implemented/deployed
- Client header injection not at all unique to the functionality of this draft
  - inappropriate for -tokbind-ttrp to define a one-off mechanism
- Stripping/sanitizing headers is de facto means of dealing with this kind of situation in practice today
  - sufficient when properly implemented
  - normatively required by -tokbind-ttrp
- The unsafe failure mode is far from catastrophic
  - lose protections afforded by token binding, which is not ideal, but it is the current state of just about everything on the web today

Support for Other Token Binding Types?

- -01 currently only supports provided and referred
  - `Sec-Provided-Token-Binding-ID`
  - `Sec-Referred-Token-Binding-ID`
- #99 Prague minutes: “have usecases that require > 2 token bindings”
- Use-case description requested
  - (no details provided to the WG yet)
- Looking for WG input/consensus

Until next time... Questions/Comments?

from IETF 89