

Conveying Client Certificate Information from TLS Terminating Reverse HTTP Proxies to Backend HTTP Applications

IETF 106
Singapore
Nov 2019
Brian Campbell



Problem Opportunity Statement

- HTTPS application deployments often have TLS ‘terminated’ by a reverse proxy somewhere in front of the actual HTTP(S) application
 - Old fashioned n-tier reverse proxy and origin server
 - CDN-as-a-service type offerings or application load balancing services
 - Microservices sidecar proxy
- TLS client certificate authentication is sometimes used
 - The actual application often needs to know about the client certificate
- In the absence of a standardized method of conveying the client certificate information, different implementations have done it differently or not at all

How we got to now

<https://mailarchive.ietf.org/arch/msg/oauth/jQ5MAZ1XCvxWbHwqlT3ITEEQoKo>



- ‘OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens’ is soon to be RFC and came up during list discussion of a different draft
- “... the proxy will pass the cert through to the AS in some undefined HTTP header with some undefined encoding. The mTLS spec should have defined this IMO, as it prevents interop ...” – WG list participant
- “It's not clear to me that mTLS should have defined a protocol from proxy to backend; that seems like it could be a fairly generic thing” – AD
- “agree ... it would be nice if such a thing existed but it would have much wider applicability than one narrow profile of OAuth.” - me
- “also agree. Would it be possible to get this pushed to http or tls? It would be more appropriate there, and very helpful to have a general spec for this.” – different WG list participant
- “...someone has to actually write a draft. Barring that, a HotRFC or secdispatch slot in Singapore would probably be good for drumming up interest.” – AD
- “I did put it some work on a conceptually similar issue around token binding with ‘HTTPS Token Binding with TLS Terminating Reverse Proxies’ ... lots of ways to approach the problem and solution but perhaps a lot could be taken from that document and applied to the similar client cert situation.” - me

A Simple Proposal

could potentially enable turn-key interoperable integration
between independent components

HTTP over a client
certificate mutually
authenticated TLS
connection

Sanitize headers and pass the client
certificate as new header with a
defined name and encoding

Client


GET /stuff HTTP/1.1
Host: example.com

Reverse
Proxy

```
GET /stuff HTTP/1.1
Host: ...
Something-something-certificate: MIIBBjCBraIBAjAKBggqhkJOPQ
QDAjAPMQ0wCwYDVQQDDARtdGxzMB4XDTE4MTAxODEyMzcwOVVoXDTIyMDUwM
jEyMzcwOVowDzENMAsGA1UEAwwEbXRsczBZMBMGBYqGSM49AgEGCCqGSM49
AwEHA0IABNcnxwqV6hY8QnhxxzFQ03C7HKW90y1MbnQZjjJ/Au08/coZwx
S7LFA4v0LS9WuneIXhbGGWvsDSb0tH6IxLm8wCgYIKoZIzj0EAwIDSQAwrG
IhAP0RC1E+vwJD/D1AGHGzuri+h1V/PpQEKTWUveORWz83AiEA5x2eXZOVb
U1JSGQgjdW5vaUaK1LR50Q2DmFfQj1L+SY=
```

Origin
Server

- Lots of disparate solutions already exist & retroactive adoption of a late-coming standard is uncertain
- Consensus here might prove surprisingly elusive
 - Aforementioned thread degenerated into strong opinions on properly securing such approaches and broadline personal attacks
 - Attacked (not personal) by an AD during #99 presentation of draft-ietf-tokbind-ttrp
 - RFC 7239 Forwarded HTTP Extension
 - draft-schwartz-tls-lb TLS Metadata for Load Balancers

A large cargo ship is sailing on a body of water at dusk. The ship's lights are on, and its reflection is visible in the water. In the background, a city skyline is visible, with buildings and lights reflecting on the water. The sky is dark and cloudy.

**But has the ship
already sailed?**

Looking ahead to IETF #107 in Vancouver

**To dispatch, or not to dispatch,
that is the question**