IETF 107
~~Vancouver~~ Virtual
March 2020

SECDISPATCH

Client-Cert HTTP Header

Conveying Client Certificate Information from TLS Terminating Reverse Proxies to Origin Server Applications

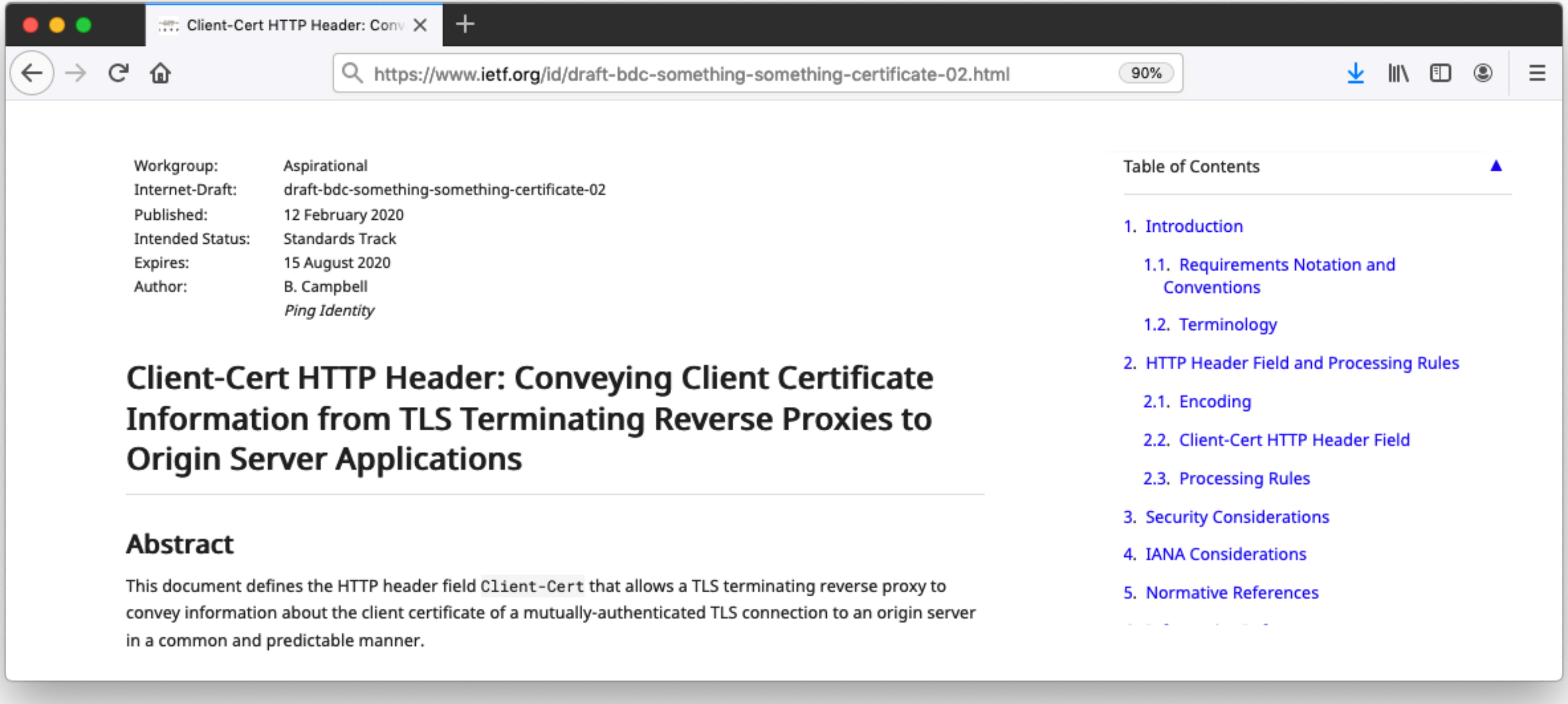
Brian Campbell
draft-bdc-something-something-certificate

~~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
  - Ingress controllers
- TLS client certificate authentication is sometimes used
  - In which case 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

[now] there’s a draft for that

https://datatracker.ietf.org/doc/draft-bdc-something-something-certificate/


Client-Cert HTTP Header: Conv
https://www.ietf.org/id/draft-bdc-something-something-certificate-02.html
90%

Workgroup: Aspirational
Internet-Draft: draft-bdc-something-something-certificate-02
Published: 12 February 2020
Intended Status: Standards Track
Expires: 15 August 2020
Author: B. Campbell
Ping Identity

Client-Cert HTTP Header: Conveying Client Certificate Information from TLS Terminating Reverse Proxies to Origin Server Applications

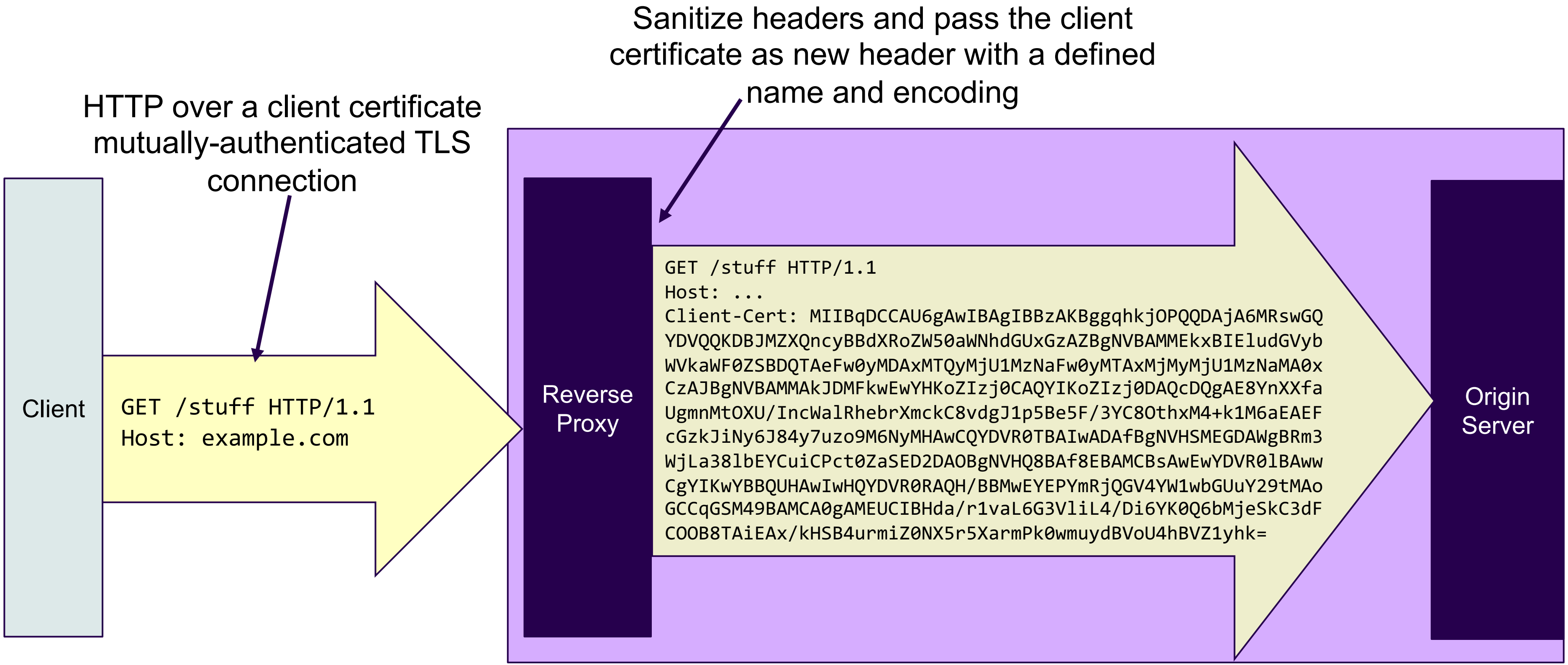
Abstract

This document defines the HTTP header field `Client-Cert` that allows a TLS terminating reverse proxy to convey information about the client certificate of a mutually-authenticated TLS connection to an origin server in a common and predictable manner.

Table of Contents

draft-bdc-something-something-certificate:

a simple proposal that could potentially enable turn-key interoperable integration between independent components



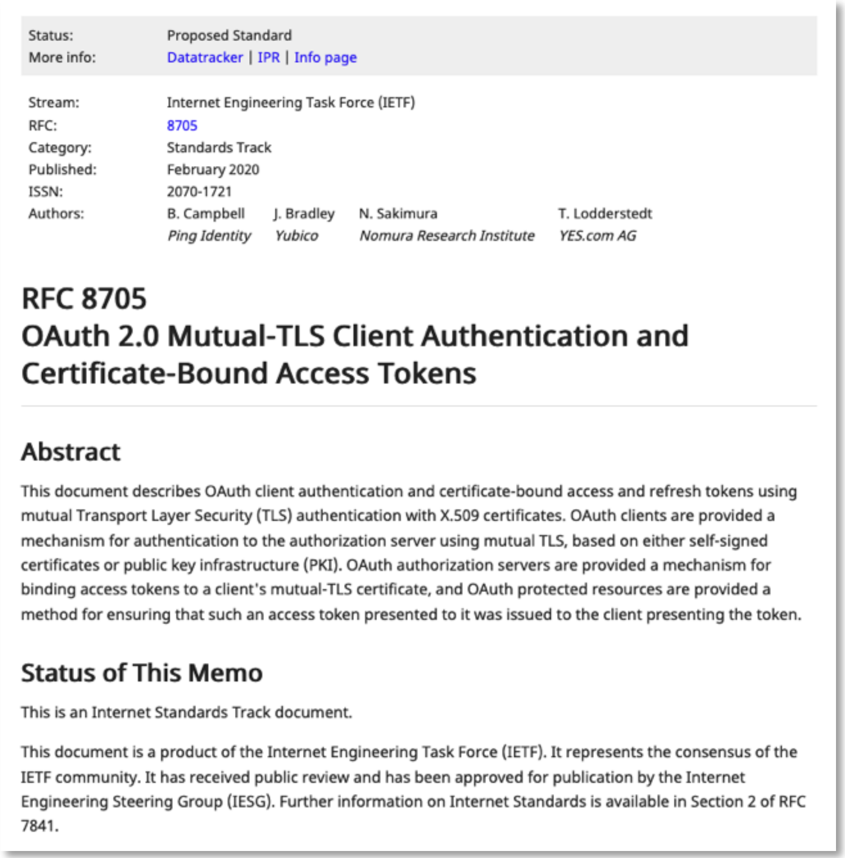
HTTP over a client certificate mutually-authenticated TLS connection

```
GET /stuff HTTP/1.1
Host: example.com
```

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

```
GET /stuff HTTP/1.1
Host: ...
Client-Cert: MIIBqDCCAU6gAwIBAgIBBzAKBggqhkjOPQQDAjA6MRswGQ
YDVQQKDBJMZXQncyBBdXRoZW50aWNhdGUxGzAZBgNVBAMMEkxBIEludGVyb
WVkaWF0ZSBDQTAeFw0yMDAxMTQyMjU1MzNaFw0yMTAxMjMyMjU1MzNaMA0x
CzAJBgNVBAMMAkJDMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE8YnXXfa
UgmnMtOXU/IncWalRhebrXmckC8vdgJ1p5Be5F/3YC8OthxM4+k1M6aEAEF
cGzkJiNy6J84y7uzo9M6NyMHAwCQYDVR0TBAIwADAfBgNVHSMEGDAWgBRm3
WjLa38lbEYCuiCPct0ZaSED2DAOBgNVHQ8BAf8EBAMCBsAwEwYDVR0lBAww
CgYIKwYBBQUHAwIwHQYDVR0RAQH/BBMwEYEPYmRjQGV4YW1wbGUuY29tMAo
GCCqGSM49BAMCA0gAMEUCIBHda/r1vaL6G3VliL4/Di6YK0Q6bMjeSkC3dF
COOB8TAiEAx/kHSB4urmiZ0NX5r5XarmPk0wmuydBVoU4hBVZ1yhk=
```

Backstory: Oct 2019



Status:	Proposed Standard
More info:	Datatracker | IPR | Info page

Stream:	Internet Engineering Task Force (IETF)			
RFC:	8705			
Category:	Standards Track			
Published:	February 2020			
ISSN:	2070-1721			
Authors:	B. Campbell *Ping Identity*	J. Bradley *Yubico*	N. Sakimura *Nomura Research Institute*	T. Lodderstedt *YES.com AG*

RFC 8705
OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens

Abstract

This document describes OAuth client authentication and certificate-bound access and refresh tokens using mutual Transport Layer Security (TLS) authentication with X.509 certificates. OAuth clients are provided a mechanism for authentication to the authorization server using mutual TLS, based on either self-signed certificates or public key infrastructure (PKI). OAuth authorization servers are provided a mechanism for binding access tokens to a client's mutual-TLS certificate, and OAuth protected resources are provided a method for ensuring that such an access token presented to it was issued to the client presenting the token.

Status of This Memo

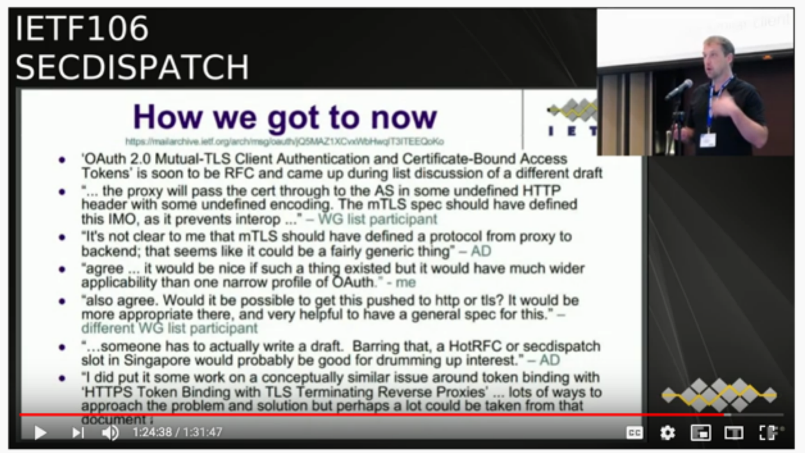
This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

- "... 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 ..." – OAUTH WG list participant [1]
- "WTF?" – [paraphrasing] me+[2]
- "possible to get this pushed to http or tls? … more appropriate there, and very helpful to have a general spec" – different OAUTH WG list participant [3]
- "…HotRFC or secdispatch…" – a Sec AD [4]

[1] https://mailarchive.ietf.org/arch/msg/oauth/mHIklRNEXjsLx6duoq_NcC0EMXU/
[2] https://mailarchive.ietf.org/arch/msg/oauth/WtR-VfkfSGHB90i70gDrfMdWau0/
https://mailarchive.ietf.org/arch/msg/oauth/n9OitgKy0iE7aM3pDv_OWAv64FA/
[3] https://mailarchive.ietf.org/arch/msg/oauth/RUzYeToHWDxLn7GlxbpXFLSNC1c/
[4] https://mailarchive.ietf.org/arch/msg/oauth/Rz_ndhksas1pTPTJlQAArnuCwBA/

Backstory: Singapore Nov 2019



How we got to now

https://mailarchive.ietf.org/arch/msg/oauth/jQ5MAZ1XCvxWbHwqT3ITEEQoKo

- '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."

jetlag & rushed presentation with too many words + no draft = come back later

Backstory: Jan/Feb/Mar 2020

- Wrote drafts -00, -01, & -02
- Shared ‘em SECDISPATCH (maybe should have been DISPATCH?)
- Received a positive, if somewhat underwhelming, reception [1]
  - “think it is useful”
  - “I support this effort! … lack of … has been a pain point for migrating applications with client-cert driven auth mechanisms into the cloud.”
  - “Good luck. If you need a vote, please let me know.” [off list]
  - “surprised it wasn't already a thing actually. I can't see why it would any place other than HTTPbis.”
  - “that would have been useful two years ago” [coworker off list]

[1] https://mailarchive.ietf.org/arch/browse/secdispatch/?q=draft-bdc-something-something-certificate

- Lots of disparate solutions already exist & retroactive adoption of late-coming standards uncertain (at best)
- Consensus here might prove surprisingly elusive
  - Aforementioned OAUTH thread degenerated into strong opinions on properly securing approaches and borderline personal attacks
  - Attacked (not personal but…) by an AD during #99 presentation of draft-ietf-tokbind-ttrp on similar security issues
  - Likely contention about exactly what and how much certificate data to convey
  - RFC 7239: Forwarded HTTP Extension
  - draft-schwartz-tls-lb: TLS Metadata for Load Balancers
  - All the other things that I don't know that I don't know…
    - draft-ietf-httpbis-http2-secondary-certs: Secondary Certificate Authentication in HTTP/2
    - ?

But… has that ship already sailed?

North ~~Vancouver~~

Before 108 ~~Madrid?~~, 109 Bangkok(~~?~~) or 110 Prague!?!...

To dispatch, or not to dispatch? that is the question
Whether 'tis better in TLS or HTTPbis or something else or not at all