HTTPBIS
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

M. Thomson
Mozilla

July 4, 2014

Client Authentication over New TLS Connection
draft-thomson-httpbis-cant-01

Abstract

This document defines an HTTP authentication scheme that can be added to an error response to indicate to a client that a successful response will only be provided over a new TLS connection, and only if the client has provided a certificate on that connection.

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 5, 2015.

Table of Contents

1. Introduction

Client authentication in HTTP sometimes relies on certificate-based authentication of clients in Transport Layer Security (TLS) ([RFC5246], [I-D.ietf-tls-tls13]). Some existing uses of client authentication rely on TLS renegotiation, triggering renegotiation and a request for a client certificate in response to a request for a particular resource.

HTTP/2 [I-D.ietf-httpbis-http2] forbids the use of renegotiation, except for at the very beginning of a connection. TLS 1.3 [I-D.ietf-tls-tls13] does not support renegotiation at all. Both of these restrictions result in a server being unable to use renegotiation to conditionally request certificate-based authentication for clients in those protocol versions.

This document defines a new authentication scheme, "ClientCertificate", for use in HTTP authentication challenges [I-D.ietf-httpbis-p7-auth]. In combination with the 401 (Unauthorized) status code, this indicates that client authentication at the TLS layer is required in order to access the resource.

1.1. Conventions and Terminology

At times, this document falls back on shorthands for establishing interoperability requirements on implementations: the capitalized words "MUST", "SHOULD" and "MAY". These terms are defined in [RFC2119].

2. Client Certificate Challenge

A new authentication scheme ([I-D.ietf-httpbis-p7-auth]) for the "WWW-Authenticate" and "Proxy-Authenticate" header fields is defined with the name "ClientCertificate".

This challenge cannot be satisfied by constructing an Authorization header field [I-D.ietf-httpbis-p7-auth], it can only be satisfied by making the request on a TLS connection where an appropriate certificate has been provided by the client.

This authentication scheme cannot be used for "http" URIs.

To effectively use this authentication scheme, a new connection is needed for every protection space used by a given origin server. A client can use the "ClientCertificate" challenge as a trigger to open a new connection and to use client authentication on that connection.

For the new connection, a client can use the mechanism in [I-D.thomson-tls-care] to prompt the server to request a client certificate, to avoid the problem where the server doesn't know to make a CertificateRequest.

[[TBD: In TLS 1.3 a client can unilaterally provide authentication information without a request from the server.]]

[[TBD For versions of TLS prior to 1.3,]] a client can immediately request renegotiation immediately after the initial handshake. A server that supports conditional client authentication MUST request a client certificate if it receives a renegotiation request prior to receiving any application data.

3. Client Certificate Challenge Parameters

In addition to the "realm" parameter, a challenge with this authentication scheme MAY include parameters that provide a client with information that assists with selecting an appropriate certificate to offer. This can be necessary, since the necessary context that the server relies on is derived from the request and this is not available at the server when establishing a new connection.

3.1. Distinguished Name Parameter ("dn")

The "dn" attribute includes a distinguished name for the certificate authority. This matches the "certificate_authorities" value sent in a TLS CertificateRequest handshake message. The "dn" parameter is repeated for every distinguished name that is permitted.

A distinguished name is defined in Abstract Syntax Notation number 1 (ASN.1) [X680] and encoded using Distinguished Encoding Rules (DER) [X690] when used in TLS. The value of the "dn" parameter is a DER-encoded distinguished name that is further encoded using base 64 [RFC4648] with the URL and filename safe alphabet. Multiple alternative distinguished names are carried by repeating the "dn" parameter.

```
dn-parameter = base64url
base64url    = ALPHA / DIGIT / "_" / "~"
```

[[Issue: We could use the string encoding defined in RFC 4514, which could be friendlier for production and consumption. That means that this would need to use quoted-string, RFC 4514 escaping would need to be escaped twice, and non-ASCII characters in the DN would need to be escaped. Since the fingerprint parameters are strictly better anyway, and it's also highly likely that selection criteria are unnecessary due to clients rarely having more than one certificate for any given site, I'm not inclined to support that level of complication.]]

3.2. Fingerprint Parameters ("sha-256", ...)

A server can instead include the fingerprint of a certificate that is on the certificate chain of an acceptable client certificate. For instance, this might include the fingerprint of an acceptable end-entity certificate, though what is more likely is that this includes the fingerprint of a certificate issuer.

The name of fingerprint parameters is taken from the hash function textual names registry defined in [RFC4572]. Clients MUST support a "sha-256" parameter, which indicates that the fingerprint is calculated using SHA-256 [RFC6234]. The value of a fingerprint parameter is encoded using base 64 [RFC4648] with the URL and filename safe alphabet.

```
sha-256-parameter = base64url
```

Like the "dn" parameter, fingerprint parameters can be repeated to provide clients with alternative values.

For example, a server could indicate that a set of permissible certificates based on a SHA-256 fingerprint, as follows:

```
WWW-Authenticate: ClientCertificate realm="home",
                  sha-256=NjUwZjA0Mzcy..., sha-256=MzJiMTQ3ODF...
```

These fingerprint values could be matched against an end-entity certificate or any issuer in the certificate chain. Note that line breaks are added to this example for formatting reasons only.

Where fingerprints are provided with multiple hash function names, a client can use any of the provided algorithms to determine which certificate to provide.

Note that strong collision-resistance is not important for the hash function that is used for certificate fingerprints, since clients only use this value to select between available certificates. The only consequence of a collision is that it becomes more difficult for the client to select the correct certificate.

4. Security Considerations

Clients that support this authentication scheme will create a new connection each time that they see a challenge. This could be exploited in order to generate additional load from connections on both server and client. This authentication scheme MUST only be used for "https" URIs.

Using new connections for client authentication has additional processing costs to the client in proving access to the private keys associated with the client certificate; and to the server in proving access to the private keys associated with their certificate twice in the case that the client opts for confidentiality protection on the client certificate (though only for TLS versions prior to 1.3, see Section 5).

HTTP/2 [I-D.ietf-httpbis-http2] allows clients to use the same connection for multiple domains. Certificate-based client authentication as defined by this specification is bound to a single canonical root URI (see [I-D.ietf-httpbis-p7-auth]). This could create issues where the security properties of a connection become unclear. Clients MUST ensure that a client-authenticated connection is only used for the canonical root URI for which it was created.

5. Privacy Considerations

TLS version 1.2 and prior do not provide confidentiality protection for client certificates in the initial handshake. Confidentiality protection for handshake messages, including the client certificate, is provided only for renegotiation handshakes.

Clients can initiate renegotiation immediately after the TLS connection is established to ensure that passive observers aren't able to view the selected certificate.

Revealing that a certificate is in use could alert a passive observer to the fact that a client has requested particular resources, thereby aiding traffic analysis. While renegotiation hides the contents of a client certificate, the presence of a new connection could reveal that some form of client authentication is being used. This is an especially strong signal in HTTP/2, where new connections are discouraged and are therefore exceptional.

Clients MUST avoid offering their client certificate if it will lack confidentiality protection, unless they are explicitly configured to send credentials in the clear.

6. IANA Considerations

IANA is requested to create an entry in the HTTP Authentication Scheme Registry with the following information:

Authentication Scheme Name: ClientCertificate

Pointer to specification text: This document

Notes This scheme does not rely on the Authorization header field.

7. Acknowledgements

Eric Rescorla helped identify the problem and formulate this mechanism. Julian Reschke and Michael Koeller provided excellent feedback. Andrei Popov observed that the information in the TLS CertificateRequest message is needed so that clients can select an appropriate certificate.

8. References

8.1. Normative References

[I-D.ietf-httpbis-p7-auth]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Authentication", draft-ietf-httpbis-p7-auth-26 (work in progress), February 2014.

[I-D.ietf-tls-tls13]
Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-01 (work in progress), April 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[X680] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ISO/IEC 8824-1:2002, 2002.

[X690] ITU-T, "Information technology - ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ISO/IEC 8825-1:2002, 2002.

8.2. Informational References

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-13 (work in progress), June 2014.

[I-D.thomson-tls-care]
Thomson, M., "Client Authentication Request Extension for (D)TLS", draft-thomson-tls-care-00 (work in progress), March 2014.

Author's Address

Martin Thomson
Mozilla
331 E Evelyn Street
Mountain View, CA 94041
US

Email: martin.thomson@gmail.com