HTTP
Internet-Draft
Intended status: Standards Track
Expires: September 16, 2016

M. Bishop
Microsoft

March 15, 2016

Secondary Server-Certificate Authentication in HTTP/2
draft-bishop-httpbis-http2-additional-certs-00

Abstract

Many HTTP servers host content from several origins. HTTP/2 [RFC7540] permits clients to reuse an existing HTTP connection to a server provided that certain conditions are satisfied. One of these conditions is the inclusion of the secondary origin in the certificate provided during the TLS [I-D.ietf-tls-tls13] handshake.

In many cases, origins will wish to maintain separate certificates for different origins but still desire the benefits of a shared HTTP connection. This draft describes how frames which were defined to transfer client certificates might be used to provide additional server certificates as well.

Status of This Memo

This Internet-Draft will expire on September 16, 2016.

Copyright Notice

Table of Contents

1. Introduction

Section 9.1.1 of [RFC7540] describes how connections may be reused as long as the server is authoritative for both origins. A server is considered authoritative for both origins if DNS resolves both origins to the IP address of the server and (for TLS) if the certificate presented by the server contains both origins, either as the Subject or contained in the Subject Alternative Names field.

[I-D.ietf-httpbis-alt-svc] enables a step of abstraction from the DNS resolution. If both hosts have provided an Alternative Service at hostnames which resolve to the IP address of the server, they are considered authoritative just as if DNS resolved the origin itself to that address.

The ORIGIN extension frame, defined in [I-D.nottingham-httpbis-origin-frame], provides a negative coalescing feature - a way for a server to request that a client _not_ reuse a connection for an origin, even when it might otherwise appear to be

supported. However, the ORIGIN frame does not currently permit a server to advertise the availability of origins which do not appear in the server's certificate as presented in the TLS handshake.

Servers which host many origins often would prefer to have separate certificates for some sets of origins. This may be for ease of certificate management (the ability to separately revoke or renew them), for legal reasons (a CDN acting on behalf of multiple origins), or any other factor which might drive this administrative decision. Clients connecting to such origins cannot currently reuse connections, even if both client and server would be willing to do so.

[I-D.thomson-http2-client-certs] defines certificate-related HTTP/2 frames, permitting a sender to offer a certificate chain along with proof that it possesses the corresponding private key to the end certificate. These frames are bound to the underlying TLS session, so that the certificates are as reliable as those provided at the TLS layer.

In this document, a mechanism for using these frames for secondary server authentication via HTTP/2 frames is defined. This mechanism can be implemented at the HTTP layer without requiring new TLS stack behavior and without breaking the existing interface between HTTP and applications above it. It primarily relaxes the one-way nature of the frames defined in [I-D.thomson-http2-client-certs], defining the processing of these frames in the reverse direction.

1.1. Origin Discovery

1.1.1. Client-driven discovery

As defined in [RFC7540], when a client finds that a https:// origin (or Alternative Service [I-D.ietf-httpbis-alt-svc]) to which it needs to make a request has the same IP address as a server to which it is already connected, it MAY check whether the TLS certificate provided contains the new origin as well, and if so, reuse the connection.

If not, but the server has advertised support for HTTP-layer certificates, the client MAY also send a "CERTIFICATE_REQUEST" frame Section 2.1 on stream zero requesting a certificate for the desired origin. The server responds with a series of "CERTIFICATE" frames containing the relevant certificate chain, if it possesses such a certificate. If not, the server responds with an empty "CERTIFICATE" frame.

1.1.2. Server-driven discovery

Because the approach in Section 1.1.1 requires an extra round-trip to the server before the client can determine whether a new TCP connection will be required, some origins will wish to proactively alert clients to certificates they possess. Servers might also wish to proactively prove their authority for an origin for which it intends to deliver pushed resources.

The server MAY send an "ORIGIN" frame including origins which are not in its TLS certificate. This represents an explicit claim by the server to possess the appropriate certificate - a claim the client MUST verify using the procedures in Section 1.1.1 before relying on the server's authority for the claimed origin.

The server might push resources from an origin for which it is authoritative but for which the client has not received the certificate. In this case, the client SHOULD verify the server's possession of an appropriate certificate by sending a "CERTIFICATE_REQUIRED" frame on the pushed stream and a "CERTIFICATE_REQUEST" on stream zero. The client MUST NOT use the pushed resource until an appropriate certificate has been received and validated.

2. Presenting Server Certificates at the HTTP/2 Framing Layer

{#certs-http2}

When a client wishes to obtain additional certificates from a server that has signaled support for HTTP certificate authentication (see Section 4), it does this by sending at least one "CERTIFICATE_REQUEST" frame (see Section 2.1) on stream zero. A client MAY send multiple concurrent "CERTIFICATE_REQUEST" frames. If server-initiated streams are blocked until the "CERTIFICATE_REQUEST" has been answered, the client SHOULD send "CERTIFICATE_REQUIRED" frames on those streams to inform the server.

Servers respond to certificate authentication requests by sending one or more "CERTIFICATE" frames (see Section 2.3) followed by a "CERTIFICATE_PROOF" frame, on stream zero.

2.1. The CERTIFICATE_REQUEST Frame

When the server has advertised support for HTTP certificate authentication (see Section 4), clients MAY send the "CERTIFICATE_REQUEST" frame. A server that has advertised support MUST NOT treat receipt of such a frame as a session error of type "PROTOCOL_ERROR".

The "CERTIFICATE_REQUEST" frame MUST be sent on stream zero. A "CERTIFICATE_REQUEST" frame received on any other stream MUST be rejected with a stream error of type "PROTOCOL_ERROR".

When sent from client to server, the "CERTIFICATE_REQUEST" frame has the same layout, with one change to the field definitions. The "CA-Count" and "Certificate-Authorities" fields are replaced by "Origin-Count" and "Origins" fields, with the same length and format. "Origins" is the distinguished name of the origin for which the client wishes to obtain a certificate, represented in DER-encoded [X690] format. The number of such structures is given by the 16-bit "Origin-Count" field, which MUST be one (0x01).

2.2. The CERTIFICATE_REQUIRED frame

The "CERTIFICATE_REQUIRED" frame is sent by clients to indicate that processing of a server-initiated stream (for example, a pushed resource) is blocked pending certificate authentication. The frame includes a request identifier which can be used to correlate the stream with a "CERTIFICATE_REQUEST" frame received on stream zero. The layout and fields are unmodified from [I-D.thomson-http2-client-certs].

When the server has advertised support for HTTP certificate authentication (see Section 4), clients MAY send the "CERTIFICATE_REQUIRED" frame. A server that has advertised support MUST NOT treat receipt of such a frame as a stream error of type "PROTOCOL_ERROR".

The client MUST NOT send a "CERTIFICATE_REQUIRED" frame on stream zero or a client-initiated stream. A server that receives a "CERTIFICATE_REQUIRED" frame on an inappropriate stream SHOULD treat this as a connection error of type "PROTOCOL_ERROR".

2.3. The CERTIFICATE frame

The "CERTIFICATE" frame allows the sender to present a certificate which should be used as authentication for previous or subsequent requests.

The payload of a "CERTIFICATE" frame contains elements of a certificate chain, terminating in an end certificate. The layout, fields, and processing are unmodified from [I-D.thomson-http2-client-certs].

2.4. The CERTIFICATE_PROOF Frame

The "CERTIFICATE_PROOF" frame allows the sender to prove possession of a certificate which should be used as authentication for previous or subsequent requests. The payload of a "CERTIFICATE_PROOF" frame contains proof of possession of the private key corresponding to an end certificate previously presented in a series of "CERTIFICATE" frames. The layout, fields, and processing are unmodified from [I-D.thomson-http2-client-certs].

Servers MUST set the "AUTOMATIC_USE" flag when sending a "CERTIFICATE_PROOF" frame.

2.5. The USE_CERTIFICATE Frame

The "USE_CERTIFICATE" frame is sent by servers to indicate that processing of a server-initiated stream should use a certificate provided in a previous series of "CERTIFICATE" and "CERTIFICATE_PROOF" frames. The frame includes a certificate identifier which can be used to correlate the stream with a certificate received on stream zero.

A "USE_CERTIFICATE" frame with no payload expresses the server's choice to proceed without providing a certificate. Clients SHOULD process the request as authenticated solely by the certificate provided at the TLS layer, likely by discarding the pushed resource and terminating the stream.

Otherwise, the "USE_CERTIFICATE" frame contains a single octet, which is the authentication request identifier. A server that receives a "USE_CERTIFICATE" of any other length MUST treat this as a stream error of type "PROTOCOL_ERROR". Frames with identical request identifiers refer to the same certificate chain.

The server MUST NOT send a "USE_CERTIFICATE" frame on stream zero or a client-initiated stream. A client that receives a "USE_CERTIFICATE" frame on an inappropriate stream SHOULD treat this as a connection error of type "PROTOCOL_ERROR".

3. Indicating failures during Certificate Authentication

The errors defined by [I-D.thomson-http2-client-certs] MAY be used by either clients or servers, as appropriate.

4. Indicating Support for HTTP-Layer Certificate Authentication

Servers that support HTTP-layer certificate authentication indicate this using the HTTP/2 "SETTINGS_HTTP_CERT_AUTH" setting defined in [I-D.thomson-http2-client-certs].

The initial value for the "SETTINGS_HTTP_CERT_AUTH" setting is 0, indicating that the server does not support HTTP-layer certificate authentication. A server sets the "SETTINGS_HTTP_CERT_AUTH" setting to a value of 1 to indicate support for HTTP-layer certificate authentication as defined in this document. Any value other than 0 or 1 MUST be treated as a connection error (Section 5.4.1 of [RFC7540]) of type "PROTOCOL_ERROR".

5. Security Considerations

This mechanism defines an alternate way to obtain server certificates other than the TLS handshake. While the signature of exporter values is expected to be equally secure, it is important to recognize that a vulnerability in this code path is equal to a vulnerability in the TLS handshake.

This draft defines a mechanism which could be used to probe servers for origins they support, but opens no new attack versus making repeat TLS connections with different SNI values. Servers SHOULD impose similar denial-of-service mitigations (e.g. request rate limits) to "CERTIFICATE_REQUEST" frames as to new TLS connections.

6. IANA Considerations

No changes are made to the registrations in [I-D.thomson-http2-client-certs].

7. Acknowledgements

8. References

8.1. Normative References

[I-D.thomson-http2-client-certs]
Thomson, M. and M. Bishop, "Reactive Certificate-Based Client Authentication in HTTP/2", draft-thomson-http2-client-certs-01 (work in progress), January 2016.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <http://www.rfc-editor.org/info/rfc7540>.

[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 ISO/IEC 8825-1:2002, 2002, <http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>.

8.2. Informative References

[I-D.ietf-httpbis-alt-svc]
mnot, m., McManus, P., and J. Reschke, "HTTP Alternative Services", draft-ietf-httpbis-alt-svc-12 (work in progress), February 2016.

[I-D.ietf-tls-tls13]
Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-11 (work in progress), December 2015.

[I-D.nottingham-httpbis-origin-frame]
mnot, m. and E. Nygren, "The ORIGIN HTTP/2 Frame", draft-nottingham-httpbis-origin-frame-01 (work in progress), January 2016.

Author's Address

Mike Bishop
Microsoft

Email: michael.bishop@microsoft.com