Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 22, 2016

M. Thomson
Mozilla
G. Eriksson
C. Holmberg
Ericsson

March 21, 2016

Caching Secure HTTP Content using Blind Caches
draft-thomson-http-bc-00

Abstract

A mechanism is described whereby a server can use client-selected shared cache.

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 September 22, 2016.

Table of Contents

1. Shared Caching for HTTPS

Shared caches allow an HTTP server to offload the responsibility for delivering certain content. Content in the shared cache can be accessed efficiently by multiple clients, saving the origin server from having to serve those requests and ensuring that clients receive responses to cached requests more quickly.

Proxy caching is the most common configuration for shared caching. A proxy cache is either explicitly configured by a client, discovered as a result of being automatically configured, or interposed automatically by an on-path network entity (this latter case being called a transparent proxy).

HTTPS [RFC2818] prevents the use of proxies by creating an authenticated end-to-end connection to the origin server or its gateway that is authenticated. This provides a critical protection against man-in-the-middle attacks, but it also prevents the proxy from acting as a shared cache.

Thus, clients use the CONNECT pseudo-method (Section 4.3.6 of [RFC7231]) with any explicitly configured proxies to create an end-to-end tunnel and will refuse to send a query for an "https" URI to a proxy.

This document describes a method for conditionally delegating the hosting of secure content to the same server. This delegation allows a client to send a request for an "https" resource via a proxy rather than insisting on an end-to-end TLS connection. This enables shared caching for a limited set of "https" resources, as selected by the server.

1.1. Notational Conventions

The words "MUST", "MUST NOT", "SHOULD", and "MAY" are used in this document. It's not shouting; when they are capitalized, they have the special meaning defined in [RFC2119].

This document uses the term "proxy cache" to refer to a proxy [RFC7230] that operates an HTTP cache [RFC7234].

2. Same-Host Secure Content Delegation

The secure content delegation mechanism defined in [SCD] is used to create a separate resource that contains encrypted and integrity protected content.

A client that signals a willingness to support this feature can be provided an response with an out-of-band encoding [I-D.reschke-http-oob-encoding] that identifies this resource. The client can then make a request for that content to a proxy cache rather than directly to the origin server.

In this document, the origin server is able to act in the role of the CDN in [SCD]. However, all of the considerations that apply to having a third party host content apply to the proxy cache. Thus, integrity and confidentiality protections against the proxy cache are the primary consideration.

2.1. Signaling Presence of a Proxy

Without a clear signal from the client that a caching proxy is present, an origin is unable to send a response with out-of-band encoding. A value of "out-of-band" in the Accept-Encoding header field might only indicate willingness to use the secure content delegation mechanism.

The BC header field indicates that a client is connected to a proxy cache that it is willing to use for out-of-band requests. The value of the BC header field is a simple boolean, represented as a "0" or "1". A value that is present and set to "1" indicates that a proxy cache is present and available for use. This header field can be used even if the current request was not routed via a proxy.

```
BC = "0" / "1"
```

Issue: What signal do we need from the proxy cache that it supports this mode of operation? Can we expect that a proxy cache will happily accept a request for an HTTPS URL?

Issue: Do we want to identify the proxy so that the origin can make some sort of judgment about the proxy? Probably not. We shouldn't be relying on the origin server making judgments about the character of proxies.

2.2. Enabling Proxy Use

It is not sufficient to couple the acceptance and use of out-of-band content encoding with the use of a proxy. Without an additional signal, a resource using secure content delegation to a CDN [SCD] could trigger a request via a proxy.

The security properties of delegation via a CDN and via a caching proxy are similar only to the extent that a third party is involved. However, it might be the case that the CDN has a stronger relationship with the origin server and additional constraints on its actions, such as contractual limitations. Such constraints might make delegation to the CDN acceptable to the origin server. A caching proxy might not be considered acceptable.

Therefore, a clear signal from the origin server is needed to allow the client to identify which resources are safe to retrieve from a proxy-cache. A "proxy" extension to the JSON format defined in [I-D.reschke-http-oob-encoding] is added that signals to the client that the out-of-band content MAY be retrieved by making a request to a proxy.

The "proxy" attribute is a boolean value. In its absence, the value is assumed to be false. If present and set to true, a client can send the request for the out-of-band content to a proxy instead of the identified server.

Clients MUST NOT send a request via a proxy if the message containing the out-of-band content encoding does not include header fields for message integrity and encryption, such as the M-I header field [I-D.thomson-http-mice] or the Crypto-Key header field [I-D.ietf-httpbis-encryption-encoding]. Absence of these header fields indicate an error on the part of the origin server, since integrity and confidentiality protection are mandatory.

Alternative: The "proxy" attribute might be replaced by a rule that stated that same-origin out-of-band encoding implied permission to route via a proxy. However, the gain here is minimal, it saves only on the explicit indication.

2.3. Proxy Identification and Authentication

This mechanism does not work with a transparent caching proxy. Since the request is made over end-to-end HTTPS in the absence of a proxy, the feature will not be used unless the proxy is known to the client.

A proxy cache MUST therefore be expressly configured or discovered. This produces a name and possibly a port number for the proxy. The proxy MUST be contacted using HTTPS [RFC2818] and authenticated using the configured or discovered domain name.

3. Performance Optimizations

As noted in [SCD], the secondary request required by out-of-band content encoding imposes a performance penalty. This can be mitigated by priming clients with information about the location and disposition of resources prior to the client making a request. A resource map described in [SCD] might be provided to clients to eliminate the latency involved in making requests of the origin server for resources that might be cached.

3.1. Proxy Cache Priming

A client that makes a request of an origin server via an unprimed proxy cache will suffer additional latency as a consequence of the cache having to make a request to the origin server.

The following options are possible:

- Clients can speculatively make requests to a proxy cache based on information it learns from a resource map. To avoid a potential waste of resources as a result of receiving complete responses, these might either be limited to HEAD requests; HTTP/2 [RFC7540] flow control might be used to allow only limited information to be sent.

- The origin server might provide the proxy cache with "prefetch" link relations in responses to requests for secondary resources. These link relations might identify other resources that the proxy might retrieve speculatively. This does not improve the latency of the initial request, but could improve subsequent requests.

4. Security Considerations

All the considerations of [SCD] apply. In particular, content that is distributed with the assistance of a proxy cache MUST include integrity and confidentiality protection. That means that the M-I header field [I-D.thomson-http-mice] and the Crypto-Key header field

[I-D.ietf-httpbis-encryption-encoding] or equivalent information MUST be present in responses that include an out-of-band content encoding.

Clients that receive a response without the information necessary to ensure integrity and confidentiality protection against a proxy cache MUST NOT make a request to a proxy to retrieve that response. Clients could treat such a response as failed, make the request directly to the origin server, or retry a request without the out-of-band token in the Accept-Encoding header field (for idempotent methods only).

5. IANA Considerations

This document has no IANA actions. It should.

6. References

6.1. Normative References

[I-D.ietf-httpbis-encryption-encoding]
Thomson, M., "Encrypted Content-Encoding for HTTP", draft-ietf-httpbis-encryption-encoding-01 (work in progress), March 2016.

[I-D.thomson-http-mice]
Thomson, M., "Merkle Integrity Content Encoding", draft-thomson-http-mice-00 (work in progress), January 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[RFC7234] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Caching", RFC 7234, DOI 10.17487/RFC7234, June 2014, <http://www.rfc-editor.org/info/rfc7234>.

[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>.

[SCD] Ericsson, G., Holmberg, C., and M. Thomson, "An Architecture for Secure Content Delegation using HTTP", February 2016, <draft-eriksson-http-scd.html>.

6.2. Informative References

[I-D.reschke-http-oob-encoding]
Reschke, J. and S. Loreto, "'Out-Of-Band' Content Coding for HTTP", draft-reschke-http-oob-encoding-04 (work in progress), March 2016.

[RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <http://www.rfc-editor.org/info/rfc7231>.

Authors' Addresses

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com

Goeran AP Eriksson
Ericsson

Email: goran.ap.eriksson@ericsson.com

Christer Holmberg
Ericsson

Email: christer.holmberg@ericsson.com