HTTP Working Group
Internet-Draft
Intended status: Experimental
Expires: December 23, 2016

M. Nottingham

M. Thomson
Mozilla

June 21, 2016

Opportunistic Security for HTTP
draft-ietf-httpbis-http2-encryption-06

Abstract

This document describes how "http" URIs can be accessed using Transport Layer Security (TLS) to mitigate pervasive monitoring attacks.

Note to Readers

Discussion of this draft takes place on the HTTP working group mailing list (ietf-http-wg@w3.org), which is archived at https://lists.w3.org/Archives/Public/ietf-http-wg/ .

Working Group information can be found at http://httpwg.github.io/ ; source code and issues list for this draft can be found at https://github.com/httpwg/http-extensions/labels/opp-sec .

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 December 23, 2016.

Copyright Notice

Table of Contents

1. Introduction

This document describes a use of HTTP Alternative Services [RFC7838] to decouple the URI scheme from the use and configuration of underlying encryption, allowing a "http" URI [RFC7230] to be accessed using Transport Layer Security (TLS) [RFC5246] opportunistically.

Serving "https" URIs require acquiring and configuring a valid certificate, which means that some deployments find supporting TLS difficult. This document describes a usage model whereby sites can serve "http" URIs over TLS without being required to support strong server authentication.

Opportunistic Security [RFC7435] does not provide the same guarantees as using TLS with "https" URIs; it is vulnerable to active attacks, and does not change the security context of the connection. Normally, users will not be able to tell that it is in use (i.e., there will be no "lock icon").

A mechanism for partially mitigating active attacks is described in Section 5.

1.1. Goals and Non-Goals

The immediate goal is to make the use of HTTP more robust in the face of pervasive passive monitoring [RFC7258].

A secondary goal is to limit the potential for active attacks. It is not intended to offer the same level of protection as afforded to "https" URIs, but instead to increase the likelihood that an active attack can be detected.

A final (but significant) goal is to provide for ease of implementation, deployment and operation. This mechanism is expected to have a minimal impact upon performance, and require a trivial administrative effort to configure.

1.2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Using HTTP URIs over TLS

An origin server that supports the resolution of "http" URIs can indicate support for this specification by providing an alternative service advertisement [RFC7838] for a protocol identifier that uses TLS, such as "h2" [RFC7540].

A client that receives such an advertisement MAY make future requests intended for the associated origin ([RFC6454]) to the identified service (as specified by [RFC7838]).

A client that places the importance of protection against passive attacks over performance might choose to withhold requests until an encrypted connection is available. However, if such a connection cannot be successfully established, the client can resume its use of the cleartext connection.

A client can also explicitly probe for an alternative service advertisement by sending a request that bears little or no sensitive information, such as one with the OPTIONS method. Likewise, clients with existing alternative services information could make such a request before they expire, in order minimize the delays that might be incurred.

Client certificates are not meaningful for URLs with the "http" scheme, and therefore clients creating new TLS connections to alternative services for the purposes of this specification MUST NOT present them. Established connections with client certificates MAY be reused, however.

3. Server Authentication

[RFC7838] requires that an alternative service only be used when there are "reasonable assurances" that it is under control of and valid for the whole origin.

As defined in that specification, a client can establish reasonable assurances when using a TLS-based protocol with the certificate checks defined in [RFC2818].

For the purposes of this specification, an additional way of establishing reasonable assurances is available when the alternative is on the same host as the origin, using the "http-opportunistic" well-known URI defined in Section 6.

This allows deployment without the use of valid certificates, to encourage deployment of opportunistic security. When it is in use, the alternative service can provide any certificate, or even select TLS cipher suites that do not include authentication.

When a client has a valid http-opportunistic response for an origin (as per Section 6), it MAY consider there to be reasonable assurances as long as:

- The origin and alternative service's hostnames are the same when compared in a case-insensitive fashion, and
- The origin object of the http-opportunistic response has a `tls-ports' member, whose value is an array of numbers, one of which matches the port of the alternative service in question, and
- The chosen alternative service returns the same representation as the origin did for the http-opportunistic resource.

For example, this request/response pair would constitute reasonable assurances for the origin "http://www.example.com" for an alternative service on port 443 or 8000 of the host "www.example.com":

```
GET /.well-known/http-opportunistic HTTP/1.1
Host: www.example.com

HTTP/1.1 200 OK
Content-Type: application/json
Connection: close

{
  "http://www.example.com": {
    "tls-ports": [443, 8000],
    "lifetime": 2592000
  }
}
```

Note that this mechanism is only defined to establish reasonable assurances for the purposes of this specification; it does not apply to other uses of alternative services unless they explicitly invoke it.

4. Interaction with "https" URIs

When using alternative services, requests for resources identified by both "http" and "https" URIs might use the same connection, because HTTP/2 permits requests for multiple origins on the same connection.

Since "https" URIs rely on server authentication, a connection that is initially created for "http" URIs without authenticating the server cannot be used for "https" URIs until the server certificate is successfully authenticated. Section 3.1 of [RFC2818] describes the basic mechanism, though the authentication considerations in Section 2.1 of [RFC7838] also apply.

Connections that are established without any means of server authentication (for instance, the purely anonymous TLS cipher suites) cannot be used for "https" URIs.

Because of the risk of server confusion about individual requests' schemes (see Section 8.4), clients MUST NOT mix "https" and "http" requests on the same connection unless the http-opportunistic response's origin object Section 6 has a "mixed-scheme" member whose value is "true".

5. Requiring Use of TLS

Even when the alternative service is strongly authenticated, opportunistically upgrading cleartext HTTP connections to use TLS is subject to active attacks. In particular:

- Because the original HTTP connection is in cleartext, it is vulnerable to man-in-the-middle attacks, and
- By default, if clients cannot reach the alternative service, they will fall back to using the original cleartext origin.

Given that the primary goal of this specification is to prevent passive attacks, these are not critical failings (especially considering the alternative - HTTP over cleartext). However, a modest form of protection against active attacks can be provided for clients on subsequent connections.

When an origin is able to commit to providing service for a particular origin over TLS for a bounded period of time, clients can choose to rely upon its availability, failing when it cannot be contacted. Effectively, this makes the choice to use a secured protocol "sticky".

5.1. Opportunistic Commitment

An origin can reduce the risk of attacks on opportunistically secured connections by committing to provide a secured, authenticated alternative service. This is done by including the optional "tls-commit" member in the origin object of the http-opportunistic well-known response (see Section 6).

This feature is optional due to the requirement for server authentication and the potential risk entailed (see Section 5.3).

When the value of the "tls-commit" member is "true" ([RFC7159], Section 3), it indicates that the origin makes such a commitment for the duration of the origin object lifetime.

```
{
  "http://www.example.com": {
    "tls-ports": [443,8080],
    "tls-commit": true,
    "lifetime": 3600
  }
}
```

Including "tls-commit" creates a commitment to provide a secured alternative service for the advertised period. Clients that receive this commitment can assume that a secured alternative service will be available for the origin object lifetime. Clients might however choose to limit this time (see Section 5.3).

5.2. Client Handling of A Commitment

The value of the "tls-commit" member MUST be ignored unless the alternative service can be strongly authenticated. The same authentication requirements that apply to "https://" resources SHOULD be applied to authenticating the alternative. Minimum authentication requirements for HTTP over TLS are described in Section 2.1 of [RFC7838] and Section 3.1 of [RFC2818]. As noted in [RFC7838], clients can impose other checks in addition to this minimum set. For instance, a client might choose to apply key pinning [RFC7469].

A client that receives a commitment and that successfully authenticates the alternative service can assume that a secured alternative will remain available for the origin object lifetime.

A client SHOULD avoid sending requests via cleartext protocols or to unauthenticated alternative services for the duration of the origin object lifetime, except to discover new potential alternatives.

A commitment is not bound to a particular alternative service. Clients are able to use alternative services that they become aware of. However, once a valid and authenticated commitment has been received, clients SHOULD NOT use an alternative service without both reasonable assurances (see Section 3) and strong authentication. Where there is an active commitment, clients SHOULD ignore advertisements for unsecured alternative services.

A client MAY send requests to an unauthenticated origin in an attempt to discover potential alternative services, but these requests SHOULD be entirely generic and avoid including credentials.

5.3. Operational Considerations

Errors in configuration of commitments has the potential to render even the unsecured origin inaccessible for the duration of a commitment. Initial deployments are encouraged to use short duration commitments so that errors can be detected without causing the origin to become inaccessible to clients for extended periods.

To avoid situations where a commitment causes errors, clients MAY limit the time over which a commitment is respected for a given origin. A lower limit might be appropriate for initial commitments;

the certainty that a site has set a correct value - and the corresponding limit on persistence - might increase as a commitment is renewed multiple times.

6. The "http-opportunistic" well-known URI

This specification defines the "http-opportunistic" well-known URI [RFC5785]. A client is said to have a valid http-opportunistic response for a given origin when:

- The client has obtained a 200 (OK) response for the well-known URI from the origin, and it is fresh [RFC7234] (potentially through revalidation [RFC7232]), and
- That response has the media type "application/json", and
- That response's payload, when parsed as JSON [RFC7159], contains an object as the root, and
- The root object contains a member whose name is a case-insensitive character-for-character match for the origin in question, serialised into Unicode as per Section 6.1 of [RFC6454], and whose value is an object (hereafter, the "origin object"),
- The origin object has a "lifetime" member, whose value is a number indicating the number of seconds which the origin object is valid for (hereafter, the "origin object lifetime"), and
- The origin object lifetime is greater than the "current_age" (as per [RFC7234], Section 4.2.3).

Note that origin object lifetime might differ from the freshness lifetime of the response.

7. IANA Considerations

This specification registers a Well-Known URI [RFC5785]:

- URI Suffix: http-opportunistic
- Change Controller: IETF
- Specification Document(s): Section 6 of [this specification]
- Related Information:

8. Security Considerations

8.1. Security Indicators

User Agents MUST NOT provide any special security indicia when an "http" resource is acquired using TLS. In particular, indicators that might suggest the same level of security as "https" MUST NOT be used (e.g., a "lock device").

8.2. Downgrade Attacks

A downgrade attack against the negotiation for TLS is possible. With commitment (see Section 5), this is limited to occasions where clients have no prior information (see Section 8.3), or when persisted commitments have expired.

For example, because the "Alt-Svc" header field [RFC7838] likely appears in an unauthenticated and unencrypted channel, it is subject to downgrade by network attackers. In its simplest form, an attacker that wants the connection to remain in the clear need only strip the "Alt-Svc" header field from responses.

Downgrade attacks can be partially mitigated using the "tls-commit" member of the http-opportunistic well-known resource, because when it is used, a client can avoid using cleartext to contact a supporting server. However, this only works when a previous connection has been established without an active attacker present; a continuously present active attacker can either prevent the client from ever using TLS, or offer its own certificate.

8.3. Privacy Considerations

Cached alternative services can be used to track clients over time; e.g., using a user-specific hostname. Clearing the cache reduces the ability of servers to track clients; therefore clients MUST clear cached alternative service information when clearing other origin-based state (i.e., cookies).

8.4. Confusion Regarding Request Scheme

HTTP implementations and applications sometimes use ambient signals to determine if a request is for an "https" resource; for example, they might look for TLS on the stack, or a server port number of 443.

This might be due to limitations in the protocol (the most common HTTP/1.1 request form does not carry an explicit indication of the URI scheme), or it may be because how the server and application are

implemented (often, they are two separate entities, with a variety of possible interfaces between them).

Any security decisions based upon this information could be misled by the deployment of this specification, because it violates the assumption that the use of TLS (or port 443) means that the client is accessing a HTTPS URI, and operating in the security context implied by HTTPS.

Therefore, servers need to carefully examine the use of such signals before deploying this specification.

8.5. Server Controls

Because this specification allows "reasonable assurances" to be established by the content of a well-known URI, servers SHOULD take suitable measures to assure that its content remains under their control. Likewise, because the Alt-Svc header field is used to describe policies across an entire origin, servers SHOULD NOT permit user content to set or modify the value of this header.

9. References

9.1. Normative References

[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>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <http://www.rfc-editor.org/info/rfc6454>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

[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>.

[RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <http://www.rfc-editor.org/info/rfc7232>.

[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>.

[RFC7838] Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", RFC 7838, DOI 10.17487/RFC7838, April 2016, <http://www.rfc-editor.org/info/rfc7838>.

9.2. Informative References

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <http://www.rfc-editor.org/info/rfc7258>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <http://www.rfc-editor.org/info/rfc7469>.

Appendix A. Acknowledgements

Mike Bishop contributed significant text to this document.

Thanks to Patrick McManus, Stefan Eissing, Eliot Lear, Stephen Farrell, Guy Podjarny, Stephen Ludin, Erik Nygren, Paul Hoffman, Adam

Langley, Eric Rescorla, Julian Reschke, Kari Hurtta, and Richard Barnes for their feedback and suggestions.

Authors' Addresses

Mark Nottingham

Email: mnot@mnot.net
URI: https://www.mnot.net/

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com