Opportunistic Security for HTTP/2

Abstract

   This document describes how "http" URIs can be accessed using
   Transport Layer Security (TLS) and HTTP/2 to mitigate pervasive
   monitoring attacks. This mechanism not a replacement for "https"
   URIs; it is vulnerable to active attacks.

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. It allows an "http" URI [RFC7230] to be
   accessed using HTTP/2 and Transport Layer Security (TLS) [RFC5246]
   with Opportunistic Security [RFC7435].

   This document describes a usage model whereby sites can serve "http"
   URIs over TLS, thereby avoiding the problem of serving Mixed Content
   (described in [W3C.CR-mixed-content-20160802]) while still providing
   protection against passive attacks.

   Opportunistic Security does not provide the same guarantees as using
   TLS with "https" URIs, because 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").

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 (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
trivial administrative effort to configure.

Preventing active attacks (such as man-in-the-middle attacks) is a
non-goal for this specification. Furthermore, this specification is
not intended to replace or offer an alternative to "https", since
"https" both prevents active attacks and invokes a more stringent
security model in most clients.

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]. Such a protocol MUST include an
explicit indication of the scheme of the resource. This excludes
HTTP/1.1; HTTP/1.1 clients are forbidden from including the absolute
form of a URI in requests to origin servers (see Section 5.3.1 of
[RFC7230]).

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]), provided that the alternative
service opts in as described in Section 2.1.

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; therefore, clients creating new TLS connections to
alternative services for the purposes of this specification MUST NOT
present them. A server that also provides "https" resources on the
same port can request a certificate during the TLS handshake, but it
MUST NOT abort the handshake if the client does not provide one.

2.1. Alternative Server Opt-In

For various reasons, it is possible that the server might become
confused about whether requests' URLs have an "http" or "https"
scheme (see Section 4.4). To ensure that the alternative service has
opted into serving "http" URLs over TLS, clients are required to
perform additional checks before directing "http" requests to it.

Clients MUST NOT send "http" requests over a secured connection,
unless the chosen alternative service presents a certificate that is
valid for the origin as defined in [RFC2818]. Using an authenticated
alternative service establishes "reasonable assurances" for the
purposes of [RFC7838]. In addition to authenticating the server, the
client MUST have obtained a valid "http-opportunistic" response for
an origin (as per Section 2.3) using the authenticated connection.
An exception to the latter restriction is made for requests for the
"http-opportunistic" well-known URI.

For example, assuming the following request is made over a TLS
connection that is successfully authenticated for those origins, the
following request/response pair would allow requests for the origins
"http://www.example.com" or "http://example.com" to be sent using a
secured connection:

```
HEADERS
  + END_STREAM
  + END_HEADERS
    :method = GET
    :scheme = http
    :authority = example.com
    :path = /.well-known/http-opportunistic

HEADERS
    :status = 200
    content-type = application/json
DATA
  + END_STREAM
[ "http://www.example.com", "http://example.com" ]
```

This document describes multiple origins, but only for operational
convenience. Only a request made to an origin (over an authenticated
connection) can be used to acquire the "http-opportunistic" resource
for that origin. Thus, in the example, the request to
"http://example.com" cannot be assumed to also provide a
representation of the "http-opportunistic" resource for
"http://www.example.com".

2.2. Interaction with "https" URIs

   Clients MUST NOT send "http" and "https" requests on the same
   connection. Similarly, clients MUST NOT send "http" requests for
   multiple origins on the same connection.

2.3. 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:

   o The client has requested the well-known URI from the origin over
      an authenticated connection and a 200 (OK) response was provided,

   o That response is fresh [RFC7234] (potentially through revalidation
      [RFC7232]),

   o That response has the media type "application/json",

   o That response's payload, when parsed as JSON [RFC7159], contains
      an array as the root, and

   o The array contains a string that is a case-insensitive, character-
      for-character match for the origin in question, serialized into
      Unicode as per Section 6.1 of [RFC6454].

   A client MAY treat an "http-opportunistic" resource as invalid if
   values it contains are not strings.

   This document does not define semantics for "http-opportunistic"
   resources on an "https" origin, nor does it define semantics if the
   resource includes "https" origins.

   Allowing clients to cache the "http-opportunistic" resource means
   that all alternative services need to be able to respond to requests
   for "http" resources. A client is permitted to use an alternative
   service without acquiring the "http-opportunistic" resource from that
   service.

   A client MUST NOT use any cached copies of an "http-opportunistic"
   resource that was acquired (or revalidated) over an unauthenticated
   connection. To avoid potential errors, a client can request or
   revalidate the "http-opportunistic" resource before using any
   connection to an alternative service.

   Clients that use cached "http-opportunistic" responses MUST ensure
   that their cache is cleared of any responses that were acquired over
   an unauthenticated connection. Revalidating an unauthenticated
   response using an authenticated connection does not ensure the
   integrity of the response.

3. IANA Considerations

   This specification registers the following well-known URI [RFC5785]:

   o URI Suffix: http-opportunistic

   o Change Controller: IETF

   o Specification Document(s): Section 2.3 of RFC 8164

   o Related Information:

4. Security Considerations

4.1. Security Indicators

   User agents MUST NOT provide any special security indicators 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").

4.2. Downgrade Attacks

   A downgrade attack against the negotiation for TLS is possible.

   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.

4.3. 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).

4.4. 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 expected limitations in the protocol (the most
   common HTTP/1.1 request form does not carry an explicit indication of
   the URI scheme, and the resource might have been developed assuming
   HTTP/1.1), or it may be because of 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 an HTTPS URI and operating in the security context implied
   by HTTPS.

   Therefore, server implementers and administrators need to carefully
   examine the use of such signals before deploying this specification.

4.5. Server Controls

   This specification requires that a server send both an alternative
   service advertisement and host content in a well-known location to
   send HTTP requests over TLS. Servers SHOULD take suitable measures
   to ensure that the content of the well-known resource 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.

5. References

5.1. 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>.

5.2. 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>.

   [W3C.CR-mixed-content-20160802]
              West, M., "Mixed Content", World Wide Web Consortium CR
              CR-mixed-content-20160802, August 2016,
              <https://www.w3.org/TR/2016/CR-mixed-content-20160802>.

Authors' Addresses

   Mark Nottingham

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

   Martin Thomson
   Mozilla

   Email: martin.thomson@gmail.com