Network Working Group M. Nottingham
Internet-Draft
Intended status: Standards Track M. Thomson
Expires: November 21, 2014 Mozilla
May 20, 2014

Opportunistic Encryption for HTTP URIs
draft-nottingham-http2-encryption-03

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
   1.1. Goals and Non-Goals 2
   1.2. Notational Conventions 3
2. Using HTTP URIs over TLS 3
3. Server Authentication 3
4. Interaction with "https" URIs 4
5. Requiring Use of TLS 4
   5.1. The HTTP-TLS Header Field 5
   5.2. Operational Considerations 6
6. Security Considerations 6
   6.1. Security Indicators 7
   6.2. Downgrade Attacks 7
   6.3. Privacy Considerations 7
7. References 7
   7.1. Normative References 7
   7.2. Informative References 8
Appendix A. Acknowledgements 8
Authors' Addresses 8

1. Introduction

This document describes a use of HTTP Alternative Services [I-D.ietf-httpbis-alt-svc] to decouple the URI scheme from the use and configuration of underlying encryption, allowing a "http" URI to be accessed using TLS [RFC5246] opportunistically.

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

A mechanism for limiting the potential for active attacks is described in Section 5. This provides clients with additional protection against them for a period after successfully connecting to a server using TLS. This does not offer the same level of protection as afforded to "https" URIs, but increases the likelihood that an active attack be detected.

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 should have a minimal impact upon performance, and should not require extensive 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 [I-D.ietf-httpbis-alt-svc] for a protocol identifier that uses TLS, such as "h2" [I-D.ietf-httpbis-http2].

A client that receives such an advertisement MAY direct future requests for the associated origin to the identified service (as specified by [I-D.ietf-httpbis-alt-svc]).

A client that places the importance of passive protections over performance might choose to withold requests until an encrypted connection is available. However, if such a connection cannot be successfully established, the client MAY 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. Clients with expired alternative services information could make a similar request in parallel to an attempt to contact an alternative service, to minimize the delays that might be incurred by failing to contact the alternative service.

3. Server Authentication

There are no existing expectations with respect to cryptographically strong server authentication when it comes to resolving HTTP URIs. Establishing it, as described in [RFC2818], creates a number of operational challenges. For these reasons, server authentication is not mandatory for HTTP URIs when using the mechanism described in this specification.

When connecting to an alternative service for an "http" URI, clients are required to perform the server authentication procedure described in Section 3.1 of [RFC2818]. The server certificate, if one is proffered by the alternative service, is not necessarily checked for validity, expiration, issuance by a trusted certificate authority or matched against the name in the URI. Therefore, the alternative service MAY provide any certificate, or even select TLS cipher suites that do not include authentication.

A client MAY perform additional checks on the certificate that it is offered (if the server does not select an unauthenticated TLS cipher suite). For instance, a client could examine the certificate to see if it has changed over time.

In order to retain the authority properties of "http" URIs, and as stipulated by [I-D.ietf-httpbis-alt-svc], clients MUST NOT use alternative services that identify a host other than that of the origin, unless the alternative service indication itself is strongly authenticated. This is not currently possible for "http" URIs on cleartext transports.

4. Interaction with "https" URIs

An alternative service that is discovered to support "http" URIs might concurrently support "https" URIs, because HTTP/2 permits the sending of requests for multiple origins (see [RFC6454]) on the one connection. Therefore, when using alternative services, both HTTP and HTTPS URIs might be sent on the same connection.

"https" URIs rely on server authentication. Therefore, if a connection is initially created without authenticating the server, requests for "https" resources cannot be sent over that connection until the server certificate is successfully authenticated. Section 3.1 of [RFC2818] describes the basic mechanism, though the authentication considerations in [I-D.ietf-httpbis-alt-svc] could 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.

5. Requiring Use of TLS

Editors’ Note: this is a very rough take on an approach that would provide a limited form of protection against downgrade attack. It’s unclear at this point whether the additional effort (and modest operational cost) is worthwhile.

The mechanism described in this specification is trival to mount an active attack against, for two reasons:

- A client that doesn’t perform authentication an easy victim of server impersonation, through man-in-the-middle attacks.

- A client that is willing to use cleartext to resolve the resource will do so if access to any TLS-enabled alternative services is blocked at the network layer.

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 alternate service 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 avilability, failing when it cannot be contacted. Effectively, this makes the alternative service "sticky" in the client.

One drawback with this approach is that clients need to strongly authenticate the alternative service to act upon such a commitment; otherwise, an attacker could create a persistent denial of service.

5.1. The HTTP-TLS Header Field

A alternative service can make this commitment by sending a "HTTP-TLS" header field:

```
HTTP-TLS     = 1#parameter
```

When it appears in a HTTP response from a strongly authenticated alternative service, this header field indicates that the availability of the origin through TLS-protected alternative services is "sticky", and that the client MUST NOT fall back to cleartext protocols while this information is considered fresh.

For example:

```
HTTP/1.1 200 OK
Content-Type: text/html
Cache-Control: 600
Age: 30
Date: Thu, 1 May 2014 16:20:09 GMT
HTTP-TLS: ma=3600
```

Note that the commitment is not bound to a particular alternative service; clients SHOULD use other alternative services that they become aware of, as long as the requirements regarding authentication and avoidance of cleartext protocols are met.

When this header field appears in a response, clients MUST strongly authenticate the alternative service, as described in Section 3.1 of [RFC2818], noting the additional requirements in [I-D.ietf-httpbis-alt-svc]. The header field MUST be ignored if strong authentication fails.

Persisted information expires after a period determined by the value of the "ma" parameter. See Section 4.2.3 of [I-D.ietf-httpbis-p6-cache] for details of determining response age.

```
ma-parameter     = delta-seconds
```

Requests for an origin that has a persisted, unexpired value for "HTTP-TLS" MUST fail if they cannot be made over an authenticated TLS connection.

5.2. Operational Considerations

To avoid situations where a persisted value of "HTTP-TLS" causes a client to be unable to contact a site, clients SHOULD limit the time that a value is persisted for a given origin. A hard limit might be set to a month. A lower limit might be appropriate for initial observations of "HTTP-TLS"; the certainty that a site has set a correct value - and the corresponding limit on persistence - can increase as the value is seen more over time.

Once a server has indicated that it will support authenticated TLS, a client MAY use key pinning [I-D.ietf-websec-key-pinning] or any other mechanism that would otherwise be restricted to use with HTTPS URIs, provided that the mechanism can be restricted to a single HTTP origin.

6. Security Considerations

6.1. 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., using a "lock device").

6.2. Downgrade Attacks

A downgrade attack against the negotiation for TLS is possible. With the "HTTP-TLS" header field, this is limited to occasions where clients have no prior information (see Section 6.3), or when persisted commitments have expired.

For example, because the "Alt-Svc" header field [I-D.ietf-httpbis-alt-svc] 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.

As long as a client is willing to use cleartext TCP to contact a server, these attacks are possible. The "HTTP-TLS" header field provides an imperfect mechanism for establishing a commitment. The advantage is that this only works if a previous connection is established where an active attacker was not present. A continuously present active attacker can either prevent the client from ever using TLS, or offer a self-signed certificate. This would prevent the client from ever seeing the "HTTP-TLS" header field, or if the header field is seen, from successfully validating and persisting it.

6.3. Privacy Considerations

Clients that persist state for origins can be tracked over time based on their use of this information. Persisted information can be cleared to reduce the ability of servers to track clients. A browser client MUST clear persisted all alternative service information when clearing other origin-based state (i.e., cookies).

7. References

7.1. Normative References

[I-D.ietf-httpbis-alt-svc]
Nottingham, M., McManus, P., and J. Reschke, "HTTP Alternative Services", draft-ietf-httpbis-alt-svc-01 (work in progress), April 2014.

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", draft-ietf-httpbis-http2-12 (work in progress), April 2014.

[I-D.ietf-httpbis-p6-cache]
Fielding, R., Nottingham, M., and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Caching", draft-ietf-httpbis-p6-cache-26 (work in progress), February 2014.

[I-D.ietf-websec-key-pinning]
Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", draft-ietf-websec-key-pinning-13 (work in progress), May 2014.

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

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

7.2. Informative References

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, December 2011.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, May 2014.

Appendix A. Acknowledgements

Thanks to Patrick McManus, Eliot Lear, Stephen Farrell, Guy Podjarny, Stephen Ludin, Erik Nygren, Paul Hoffman, Adam Langley, Eric Rescorla and Richard Barnes for their feedback and suggestions.

Authors’ Addresses

Mark Nottingham

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

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com