Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 14, 2014

M. Nottingham
December 11, 2013

Opportunistic Encryption for HTTP URIs
draft-nottingham-http2-encryption-02

Abstract

This document proposes two changes to HTTP/2.0; first, it suggests using ALPN Protocol Identifies to identify the specific stack of protocols in use, including TLS, and second, it proposes a way to opportunistically encrypt HTTP/2.0 using TLS for HTTP URIs.

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 June 14, 2014.

Table of Contents

1. Introduction

In discussion at IETF87, it was proposed that the current means of bootstrapping encryption in HTTP [I-D.ietf-httpbis-p1-messaging] - using the "HTTPS" URI scheme - unintentionally gives the server disproportionate power in determining whether encryption (through use of TLS [RFC6246]) is used.

This document proposes using the new "alternate services" layer described in [I-D.nottingham-httpbis-alt-svc] to decouple the URI scheme from the use and configuration of underlying encryption, allowing a "http://" URI to be upgraded to use TLS opportunistically.

Additionally, because using TLS requires acquiring and configuring a valid certificate, some deployments may find supporting it difficult. Therefore, this document also proposes a "relaxed" profile of HTTP/2.0 over TLS that does not require strong server authentication, specifically for use with "http://" URIs.

1.1. Goals and Non-Goals

The immediate goal is to make HTTP URIs more robust in the face of passive monitoring.

Such passive attacks are often opportunistic; they rely on sensitive information being available in the clear. Furthermore, they are often broad, where all available data is collected en masse, being analyzed separately for relevant information.

It is not a goal of this document to address active or targeted attacks, although future solutions may be complementary.

Other goals include ease of implementation and deployment, with minimal impact upon performance (in keeping with the goals of HTTP/2.0).

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. Proposal: Indicating Security Properties in Protocol Identifiers

In past discussions, there has been general agreement to reusing the ALPN protocol identifier [I-D.ietf-tls-applayerprotoneg] for all negotiation mechanisms in HTTP/2.0, not just TLS.

This document proposes putting additional information into them to identify the use of encryption as well as configuration of that encryption, independent of the URI scheme in use.

Thus, we won't have just one protocol identifier for HTTP/2.0, but two; one with and one without the use of TLS. As such, the following identifiers are recommended if this approach is adopted:

- h1 - http/1.x over TCP
- h1t - http/1.x over TLS over TCP (as per [RFC2818])
- h2 - http/2.x over TCP
- h2t - http/2.x over TLS over TCP (as per [RFC2818])
- h2r - http/2.x over TLS over TCP (see Section 2.1)

Draft implementations could be indicated with a suffix; e.g., h2t-draft10.

Most of these are already latently defined by HTTP/2.0, with the exception being h2r, defined below. Note that the focus of this proposal is on the semantics of the identifiers; an exact syntax for them is not part of it.

By indicating the use of TLS in the protocol identifier allows a client and server to negotiate the use of TLS for "http://" URIs; if the server offers h2t, the client can select that protocol, start TLS and use it.

Note that, as discussed in Section 3.1, there may be situations (e.g,. ALPN) where advertising some of these profiles are inapplicable or inadvisable. For example, in an ALPN negotiation for a "https://" URI, it is only sensible to offer h1t and h2t.

If adopted, this proposal would be effected by adjusting the text in Section 3 of [I-D.ietf-httpbis-http2] ("Starting HTTP/2.0") along the lines described above. Note that the specific protocol identifiers above are suggestions only.

2.1. Proposal: The "h2r" Protocol

If the proposal above is adopted, a separate proposal is to define a separate protocol identifier for "relaxed" TLS operation.

Servers that support the "h2r" protocol indicate that they support TLS for access to URIs with the "http" URI scheme using HTTP/2.0 or greater.

Servers MAY advertise the "h2r" profile for resources with a "http" origin scheme; they MUST NOT advertise it for resources with a

"https" origin.

When a client connects to an "h2r" alternate service, it MUST use TLS1.1 or greater, and MUST use HTTP/2.x. HTTP/2.0 SHOULD be used as soon as TLS negotiation is completed; i.e., the "Upgrade dance" SHOULD NOT be performed.

When connecting to an "h2r" service, the algorithm for authenticating the server described in [RFC2818] Section 3.1 changes; the client does not necessarily validate its certificate for expiry, hostname match or relationship to a known certificate authority (as it would with "normal" HTTPS).

However, the client MAY perform additional checks on the certificate and make a decision as to its validity before using the server. Definition of such additional checks are out of scope for this specification.

Upon initial adoption of this proposal, it is expected that no such additional checks will be performed. Therefore, the client MUST NOT use the "h2r" profile to connect to alternate services whose host does not match that of the origin (as per [I-D.nottingham-httpbis-alt-svc]), unless additional checks are performed.

Servers SHOULD use the same certificate consistently over time, to aid future extensions for building trust and adding other services.

[TODO: define "same"; likely not the same actual certificate.]

When the h2r protocol is in use, User Agents MUST NOT indicate the connection has the same level of security as https:// (e.g. using a "lock device").

If this proposal is adopted, the "h2r" protocol could be defined in [I-D.ietf-httpbis-http2] (most likely, Section 3), or in a separate document.

3. Security Considerations

3.1. Downgrade Attacks

A downgrade attack against the negotiation for TLS is possible, depending upon the properties of the negotiation mechanism.

For example, because the Alt-Svc header field [I-D.nottingham-httpbis-alt-svc] appears in the clear for "http://"

URIs, it is subject to downgrade by attackers that are able to Man-in-the-Middle the network connection; in its simplest form, an attacker that wants the connection to remain in the clear need only strip the Alt-Svc header from responses.

This proposal does not offer a remedy for this risk. However, it's important to note that it is no worse than current use of unencrypted HTTP in the face of such active attacks.

Future proposals might attempt to address this risk.

4. References

4.1. Normative References

[I-D.ietf-httpbis-http2]
Belshe, M., Peon, R., Thomson, M., and A. Melnikov, "Hypertext Transfer Protocol version 2.0", draft-ietf-httpbis-http2-08 (work in progress), November 2013.

[I-D.ietf-tls-applayerprotoneg]
Friedl, S., Popov, A., Langley, A., and S. Emile, "Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension", draft-ietf-tls-applayerprotoneg-03 (work in progress), October 2013.

[I-D.nottingham-httpbis-alt-svc]
Nottingham, M., "HTTP Alternate Services", draft-nottingham-httpbis-alt-svc-00 (work in progress), October 2013.

[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.

4.2. Informative References

[I-D.ietf-httpbis-p1-messaging]
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", draft-ietf-httpbis-p1-messaging-25 (work in progress), November 2013.

[I-D.mbelshe-httpbis-spdy]
Belshe, M. and R. Peon, "SPDY Protocol", draft-mbelshe-httpbis-spdy-00 (work in progress), February 2012.

[RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804, May 2000.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, August 2002.

[RFC6246] Sajassi, A., Brockners, F., Mohan, D., and Y. Serbest, "Virtual Private LAN Service (VPLS) Interoperability with Customer Edge (CE) Bridges", RFC 6246, June 2011.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[firesheep]
Butler, E., "Firesheep", 2010, <http://codebutler.com/firesheep/>.

[streetview]
Kravets, D., "The Anatomy of Google's Wi-Fi Sniffing Debacle", 2012, <http://www.wired.com/threatlevel/2012/05/google-wifi-fcc-investigation/>.

[xkeyscore]
Greenwald, G., "NSA tool collects 'nearly everything a user does on the internet'", 2013, <http://www.theguardian.com/world/2013/jul/31/nsa-top-secret-program-online-data>.

Appendix A. Acknowledgements

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

Appendix B. Recent History and Background

One of the design goals for SPDY [I-D.mbelshe-httpbis-spdy] was increasing the use of encryption on the Web, achieved by only

supporting the protocol over a connection protected by TLS [RFC5246].

This was done, in part, because sensitive information - including not only login credentials, but also personally identifying information (PII) and even patterns of access - are increasingly prevalent on the Web, being evident in potentially every HTTP request made.

Attacks such as FireSheep [firesheep] showed how easy it is to gather such information when it is sent in the clear, and incidents such as Google's collection of unencrypted data by its StreetView Cars [streetview] further illustrated the risks.

In adopting SPDY as the basis of HTTP/2 [I-D.ietf-httpbis-http2], the HTTPbis Working Group agreed not to make TLS mandatory to implement (MtI) or mandatory to use (MtU) in our charter, despite an IETF policy to prefer the "best security available" [RFC3365].

There were a variety of reasons for this, but most significantly, HTTP is used for much more than the traditional browsing case, and encryption is not needed for all of these uses. Making encryption MtU or MtI was seen as unlikely to succeed because of the wide deployment of HTTP URIs.

However, since making that decision, there have been developments that have caused the Working Group to discuss these issues again:

1. Active contributors to some browser implementations have stated that their products will not use HTTP/2 over unencrypted connections. If this eventuates, it will prevent wide deployment of the new protocol (i.e., it couldn't be used with those products for HTTP URIs; only HTTPS URIs).
2. It has been reported that surveillance of HTTP traffic takes place on a broad scale [xkeyscore]. While the IETF does not take a formal, moral position on wiretapping, we do have a strongly held belief "that both commercial development of the Internet and adequate privacy for its users against illegal intrusion requires the wide availability of strong cryptographic technology" [RFC2804]. This requirement for privacy is further reinforced by [RFC6973].

As a result, we decided to revisit the issue of how encryption is used in HTTP/2.0 at IETF87.

Appendix C. Frequently Asked Questions

C.1. Will this make encryption mandatory in HTTP/2.0?

Not in the sense that this proposal would have it required (with a MUST) in the specification.

What might happen, however, is that some browser implementers will take the flexibility that this approach grants and decide to not negotiate for HTTP/2.0 without one of the encryption profiles. That means that servers would need to implement one of the encryption-enabling profiles to interoperate using HTTP/2.0 for HTTP URIs.

C.2. No certificate checks? Really?

h2r has the effect of relaxing certificate checks on "http://" - but not "https://" - URIs when TLS is in use. Since TLS isn't in use for any "http://" URIs today, there is no net loss of security, and we gain some privacy from passive attacks.

This makes TLS significantly simpler to deploy for servers; they are able to use a self-signed certificate.

Additionally, it is possible to detect some attacks by remembering what certificate is used in the past "pinning" or third-party verification of the certificate in use. This may offer a way to gain stronger authentication of the origin server's identity, and mitigate downgrade attacks (although doing so is out of the scope of this document).

C.3. Why do this if a downgrade attack is so easy?

There are many attack scenarios (e.g., third parties in coffee shops) where active attacks are not feasible, or much more difficult.

Additionally, active attacks can often be detected, because they change protocol interactions; as such, they bring a risk of discovery.

C.4. Why Have separate relaxed protocol identifiers?

If all implementations agree that using TLS for "http://" URIs always means that the certificate checks are "relaxed", it could be that there is no need for a separate protocol identifier. However, this needs to be discussed.

Author's Address

Mark Nottingham

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