Profiles for Initial Server Settings
draft-montenegro-httpbis-http2-server-profiles-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on November, 2013.

Copyright

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

Peon, Montenegro et. al. [Page 1]
1. Introduction

In HTTP/2.0, if running over TLS, the client is the first to transmit HTTP/2.0 frames on the new session. With its first transmission to the server, the client can already overrun some of the server settings and preferences, leading to failure conditions and unnecessary complexity. This document proposes a solution to this problem based on a very small set of server profiles for initial settings sent within the TLS handshake.
This document proposes a profile-based approach, based on defining server profiles for initial settings sent by the server within the TLS handshake: a compact server profile vs a normal server profile.

The profiles are communicated as part of the exchanged application-layer protocol names, precisely as supported already by ALPN.

Of course, the compact profile is not limited to embedded/constrained servers. It could be used by a regular web server if it wishes to reduce its initial resource usage for new connections for whatever reason. These profiles are just means to communicate initial SETTINGS. These SETTINGS are no different from any other, and can be subsequently modified by another SETTINGS frames, per HTTP/2.0. For example, a server may use a compact profile for the beginning of a session, after which it may send an updated set of SETTINGS to the client, increasing the use of resources to match (or exceed) those used by the normal profile.

A point bears repeating: this issue is only with server SETTINGS in the TLS case. The profiles are only for server initial SETTINGS. The client does not have this issue, as it starts the HTTP/2.0 session by sending its SETTINGS (within the connection header), so when the server gets a chance to initially transmit on the new session it is guaranteed to have received the client SETTINGS already. As an aside, in the HTTP non-TLS case (which is not the subject of this document), HTTP/2.0 already fixed that by allowing the client to send its settings in an HTTP/1.1 header (HTTP2-Settings).

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Definition of Server Profiles for Initial Settings

There are two profiles for server initial settings: the compact server profile and the normal server profile.

Profile 1 – constrained server profile:

- Negotiation strings:
  - "H2c" for final HTTP/2.0 specification
  - "HTTP-draft-06/2.0c" for draft version -06
(Appending a "c" to the negotiation string denotes the "compact" profile.)

Profile 2 - normal server profile

- Negotiation strings:
  - "H2" for final HTTP/2.0 specification
  - "HTTP-draft-06/2.0" for draft version -06
    (Unchanged string implies the web server profile)
  - SETTINGS_MAX_CONCURRENT_STREAMS: value: 100
  - SETTINGS_INITIAL_WINDOW_SIZE: value: 64K
  - SETTINGS_FLOW_CONTROL_OPTIONS: 0 (flow control on)
  - SETTINGS_MAX_BUFFER_SIZE: 4K

These profiles are to be defined in the base HTTP/2.0 document, and
the negotiation strings would be registered with IANA in the ALPN
registry. Only the strings corresponding to the final HTTP/2.0
specification are to be registered in IANA.

Other experimental profiles could be defined for use in testing new
features, without need to register them in IANA.

3. Usage

The client uses ALPN to include the negotiation strings for both
profiles: in the TLS client hello, the client includes both of these:

Example 1 - Profiles based on final HTTP/2.0 specification:
  - H2c
  - H2

Example 2 - Profiles based on draft versions of the HTTP/2.0
specifications, e.g., for draft version -06, the client includes
both of these:
  - HTTP-draft-06/2.0c
  - HTTP-draft-06/2.0
The server uses ALPN to respond with the negotiation string it selects. For example 1, to select the compact profile the server includes this in its Hello: "H2c". Alternatively, if the client had proposed draft -06 (Example 2), the server would respond with "HTTP-draft-04/2.0c".

The client interprets the returned string and sets the SETTINGS for the server accordingly. The client can initiate the HTTP/2.0 session (beginning with the client’s SETTINGS frame) and MUST respect the server SETTINGS.

In examples above, the client sets the server SETTINGS per the compact server profile. Per usual HTTP/2.0 rules, either endpoint is free to adjust their preferences by sending additional SETTINGS frames.

Upon receipt of the client’s first transmission on the HTTP/2.0 session, the server responds with its own SETTINGS frame, which can already supersede any SETTINGS set via the server profile. This is in keeping with HTTP/2.0 rules.

4. Pros and Cons of Server Profiles

Server profiles for initial settings are not the only possible approach for the server to communicate its settings in advance of the HTTP/2.0 session. This section compares to the other alternative that has been discussed, namely, the option of augmenting the TLS handshake to allow embedding the server SETTINGS (e.g., as ancillary data to the application-layer protocol negotiation itself).

Pros:

. No need to modify TLS handshake any further. This is a HUGE benefit, considering that ALPN has entered working group last call in the TLS WG.
. No need to engage TLS WG (less extraneous dependencies).
. Simple to spec (once the profiles are agreed upon).
. HTTP/2.0 only needs to define the strings augmented with the profiles and register those in the ALPN registry (as well as other HTTPbis registries that may be applicable).
. This approach of profiling could be used to distinguish experimental from production-grade servers.
As compared to attempting to define default values for the server settings, the profile approach mitigates this by not requiring ONE set of defaults that must work for every type of server (which we fruitlessly attempted before), but 2 sets of defaults depending on the server profile. Targeting a limited set of 2 profiles should make this manageable (normal web server vs compact/embedded server).

Cons:

- Increases the size of the client TLS Hello as each profile would imply a separate string per ALPN specs, e.g., "H2c" and "H2". Not a big concern as long as we limit it to, say, 2 profiles. At any rate, other TLS extensions are growing this anyway.

5. Security Considerations

None.

6. IANA Considerations

The negotiation strings must be registered with IANA’s ALPN registry: Application Layer Protocol Negotiation protocol byte strings within the TLS section. This is not new, as HTTP/2.0 has to do this anyways. This proposal just gives a certain format to the strings that would ultimately be registered by HTTP/2.0. In particular, the compact server profile would be denoted by the string "H2c". The normal server profile would be denoted by the string "H2".

Notice that in the interest of terseness, this proposal departs from the notation currently used in HTTP/2.0. Current HTTP/2.0 would have the compact and normal server profiles be registered as "HTTP/2.0c" and "HTTP/2.0".

7. Acknowledgments

Thanks to Mark Nottingham for useful discussions in this space.

This document was prepared using 2-Word-v2.0.template.doc.
8. References

8.1. Normative References


8.2. Informative References

None.

Author’s Addresses

Roberto Peon
Google, Inc

Email: fenix@google.com

Gabriel Montenegro
Microsoft Corporation

Phone:
Email: gabriel.montenegro@microsoft.com