Internet Engineering Task Force (IETF)
Request for Comments: 8124
Category: Standards Track
ISSN: 2070-1721

R. Ravindranath
G. Salgueiro
Cisco

March 2017

The Session Description Protocol (SDP) WebSocket Connection URI Attribute

Abstract

The WebSocket protocol enables bidirectional real-time communication between clients and servers in web-based applications. This document specifies extensions to Session Description Protocol (SDP) for application protocols using WebSocket as a transport.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc8124.

Table of Contents

1. Introduction

The WebSocket protocol [RFC6455] enables bidirectional message exchange between clients and servers on top of a persistent TCP connection (optionally secured with Transport Layer Security (TLS) [RFC5246]). The initial protocol handshake makes use of Hypertext Transfer Protocol (HTTP) [RFC7230] semantics, allowing the WebSocket protocol to reuse existing HTTP infrastructure.

Modern web browsers include a WebSocket client stack compliant with the WebSocket API [WS-API] as specified by the W3C. It is expected that other client applications (e.g., those running on personal computers, mobile devices, etc.) will also make a WebSocket client stack available. Several specifications have been written that define how different applications can use a WebSocket subprotocol as a reliable transport mechanism.

For example, [RFC7118] defines a WebSocket subprotocol as a reliable transport mechanism between Session Initiation Protocol (SIP)[RFC3261] entities to enable use of SIP in web-oriented deployments. Additionally, [RFC7977] defines a new WebSocket subprotocol as a reliable transport mechanism between Message Session Relay Protocol (MSRP) clients and relays. [RFC7395] defines a WebSocket subprotocol for the Extensible Messaging and Presence Protocol (XMPP). Similarly, [BFCP-WEBSOCKET] defines a WebSocket subprotocol as a reliable transport mechanism between Binary Floor Control Protocol (BFCP) [BFCP] entities to enable usage of BFCP in new scenarios.

When a WebSocket subprotocol is used as a transport mechanism between a server and client, there needs to be a way to indicate the connection URI from the server to the WebSocket client. For applications that use Session Description Protocol (SDP) [RFC4566] to negotiate, the connection URI can be indicated by means of an SDP attribute. This specification defines new SDP attributes to indicate the connection URI for the WebSocket client. Applications that use SDP for negotiation and WebSocket as a transport protocol can use this specification to advertise the WebSocket client connection URI.

2. Terminology

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

3. SDP Considerations

3.1. General

Applications that use the SDP Offer/Answer mechanism [RFC3264] for negotiating media and also use WebSocket or secure WebSocket as a transport protocol MAY indicate the connection URI for the WebSocket client via a new SDP "a=" media-level attribute defined in Section 3.2.

3.2. "websocket-uri" SDP Attribute

This section defines a new SDP media-level attribute, "websocket-uri", which can appear in any of the media sections.

Example:

```
a=websocket-uri:wss://example.com/chat
```

Where "wss://example.com/chat" is the ws-URI defined in Section 3 of [RFC6455].

When the "websocket-uri" attribute is present in the media section of the SDP, the IP address in "c=" line SHALL be ignored and the full URI SHALL be used instead to open the WebSocket connection. The clients MUST ensure that they use the URI to open the WebSocket connection and ignore the IP address in the "c=" line and the port in the "m=" line.

3.3. "websocket-uri" Multiplexing Considerations

Multiplexing characteristics of SDP attributes are described in [SDP-MUX]. Various SDP attribute multiplexing categories are introduced there.

- The multiplexing category of the "a=websocket-uri" attribute is CAUTION.

There are no multiplexing rules specified for the "websocket-uri" SDP media-level attribute. Additionally, the specification of multiplexing rules for the "websocket-uri" attribute is outside the scope of this document.

While it is technically possible to bundle WebSocket, there are a variety of reasons that make it impractical; thus, it is considered unlikely to be used in practice. Therefore, the "websocket-uri" SDP media-level attribute defined in Section 3.2 for using WebSocket as a transport protocol is not likely to be used with SDP bundle and is consequently categorized as CAUTION for multiplexing.

If future extensions define how to bundle WebSocket, then multiplexing rules for the "a=websocket-uri" attribute need to be defined as well, for instance, in an extension of this SDP based WebSocket negotiation specification.

4. SDP Offer/Answer Procedures

4.1. General

An endpoint (i.e., both the offerer and the answerer) that wishes to negotiate WebSocket as transport protocol MUST indicate that it wishes to use WebSocket or secure WebSocket in the "proto" field of the "m=" line. Furthermore, the server side, which could be either the offerer or answerer, MUST add an "a=websocket-uri" attribute in the media section whose value can be either "ws-URI" or "wss-URI", as defined in Section 3 of [RFC6455], depending on whether it wishes to use WebSocket or secure WebSocket. This new attribute MUST follow the syntax defined in Section 3. The procedures in this section apply to an "m=" line associated with any media stream that uses WebSocket or secure WebSocket as transport.

Both offerer or answerer can initiate a WebSocket connection. It is expected that, based on the topology (for example, if the client is behind NAT and server is on global IP address), the offerer and answerer applications decide on who will initiate the WebSocket connection and appropriately set the "setup" attribute in SDP following the procedures of [RFC4145].

4.2. Generating the Initial Offer

In order to negotiate WebSocket as a transport, an SDP offerer MUST indicate that it wishes to use it in the "proto" field of the "m=" line. For example, to negotiate BFCP-over-WebSocket, the "proto" value in the "m=" line is TCP/WSS/BFCP if WebSocket is over TLS; else, it is TCP/WS/BFCP, as specified in [BFCP-WEBSOCKET].

The offerer SHOULD assign the SDP "setup" attribute with a value of "active" (the offerer will be the initiator of the outgoing TCP connection) or "passive" if the offerer wishes to be a receiver of an incoming connection. The offerer MUST NOT assign an SDP "setup" attribute with a "holdconn" value. The offerer MUST follow the procedures described in [RFC4145] while using the "setup" attribute. If the "setup" attribute has a value of "passive", it MUST have a URI in the "a=websocket-uri" attribute.

The following is an example of an "m=" line for a BFCP connection:

```
Offer (browser):
m=application 9 TCP/WSS/BFCP *
a=setup:active
a=connection:new
a=floorctrl:c-only
m=audio 55000 RTP/AVP 0
m=video 55002 RTP/AVP 31
```

In the above example, the client is intending to set up the TLS/TCP connection; hence, the port is set to a value of 9, which is the discard port.

4.3. Generating the Answer

If the answerer accepts the offered WebSocket transport connection, in the associated SDP answer, the answerer MUST assign an SDP "setup" attribute with a value of either "active" or "passive", according to the procedures in [RFC4145]. The answerer MUST NOT assign an SDP "setup" attribute with a value of "holdconn".

If the answerer assigns an SDP "setup" attribute with a value of "active", the answerer MUST initiate the WebSocket connection handshake by acting as client on the negotiated media stream, towards the URI specified in the "a=websocket-uri" SDP attribute using the procedures described in Section 4 of [RFC6455].

If the answerer assigns an SDP "setup" attribute with a value of "passive", then it MUST have a value of "ws-URI" or "wss-URI", as defined in Section 3 of [RFC6455] in an "a=websocket-uri" SDP attribute, depending on whether the application uses WebSocket or secure WebSocket. This attribute MUST follow the syntax defined in Section 3.

The following example shows a case where the server responds with a BFCP media stream over a WebSocket connection running TLS. It shows an answer "m=" line for the BFCP connection. In this example, since WebSocket is running over TLS, the server answers back with an "a=websocket-uri" attribute in the media section of SDP having a "wss-URI" connection URI:

```
Answer (server):
m=application 50000 TCP/WSS/BFCP *
a=setup:passive
a=connection:new
a=websocket-uri:wss://bfcp-ws.example.com?token=3170449312
a=floorctrl:s-only
a=confid:4321
a=userid:1234
a=floorid:1 m-stream:10
a=floorid:2 m-stream:11
m=audio 50002 RTP/AVP 0
a=label:10
m=video 50004 RTP/AVP 31
a=label:11
```

4.4. Offerer Processing of the Answer

When the offerer receives an SDP answer, if the offerer ends up initiating the TCP connection, then it MUST follow the procedures in Section 5.

4.5. Modifying the Session

Once an offer/answer exchange has been completed, either endpoint MAY send a new offer in order to modify the session. The endpoints can reuse the existing WebSocket connection by adding an "a=connection:existing" attribute in the media section of the SDP following the rules mentioned in [RFC4145], if the "websocket-uri" SDP value and the transport parameters indicated by each endpoint are unchanged. Otherwise, following the rules for the initial offer/answer exchange, the endpoints can negotiate and create a new WebSocket connection on top of TLS/TCP or TCP.

4.6. Offerless INVITE Scenarios

In some scenarios, an endpoint (e.g., a browser) originating the call (a User Agent Client or UAC) can send an offerless INVITE to the server. The server will generate an offer in response to the INVITE. In such cases, the server MUST send an offer with the "setup" attribute with a value of "passive" so as to accept incoming connection and MUST include an "a=websocket-uri" attribute in the media section whose value MUST be either "ws-URI" or "wss-URI", depending on whether the server wishes to use WebSocket or secure WebSocket. The SDP offer sent by the server will look like the example in Section 4.3.

5. Procedures at WebSocket Client

The WebSocket client MUST always initiate the outgoing TCP connection; hence, the SDP "setup" attribute MUST always be "active" for the WebSocket client in its SDP offer/answer. In the example below, the WebSocket client is the offerer; hence, it assigns a "setup" attribute with a value of "active".

The WebSocket server is a server on the Internet; hence, it MUST always assign an SDP "setup" attribute with a value of "passive". This also avoids the need to use Interactive Connectivity Establishment (ICE) between WebSocket client and WebSocket server, as the connection model here would be a typical client-to-server web connection.

Once the offer/answer is complete, the client MUST initiate the WebSocket connection handshake by sending a GET message on the negotiated media stream, towards the URI specified in an "a=websocket-uri" attribute, as per the procedures described in [RFC6455]. When no port is passed in the "a=websocket-uri" attribute, the default port (80 or 443) is used depending on whether the value was "ws-URI" or "wss-URI".

6. Security Considerations

An attacker may attempt to add, modify, or remove an "a=websocket-uri" attribute from a session description. This could result in an application behaving undesirably. Consequently, it is RECOMMENDED that integrity protection be applied to the SDP session descriptions. For session descriptions carried in SIP [RFC3261], S/MIME is available to provide such end-to-end integrity protection.

As described in Section 10 of [RFC6455], application signalling traffic being transported over WebSocket MUST support secure WebSocket and SHOULD employ it when communicating with their peers.

The WebSocket clients have to initiate the TCP connection to the WebSocket server identified by the Fully Qualified Domain Name (FQDN) in an "a=websocket-uri" attribute. Further, as with any other web connection, the clients will verify the server's certificate. The WebSocket client MUST follow the procedures in [RFC7525] (including host name verification as per Section 6.1 in [RFC7525]) while setting up a TLS connection with a WebSocket server.

7. IANA Considerations

7.1. Registration of the "websocket-uri" SDP Media Attribute

This document defines a new SDP media-level attribute "websocket-uri" in Section 3.2; IANA has registered the following SDP att-field under the "Session Description Protocol (SDP) Parameters" registry as follows:

Attribute name:	websocket-uri
Long-form attribute name:	WebSocket Connection URI
Type of attribute:	media
Mux category:	CAUTION
Charset Dependent:	No
Purpose:	The "websocket-uri" attribute is intended to be used as a connection URI for opening the WebSocket connection.
Appropriate values:	A ws-URI or wss-URI, as defined in Section 3 of [RFC6455]
Contact name:	Gonzalo Salgueiro
Contact email:	gsalguei@cisco.com
Reference:	RFC 8124

8. References

8.1. 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>.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <http://www.rfc-editor.org/info/rfc4145>.

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <http://www.rfc-editor.org/info/rfc6455>.

8.2. Informative References

[BFCP] Camarillo, G., Drage, K., Kristensen, T., Ott, J., and C. Eckel, "The Binary Floor Control Protocol (BFCP)", Work in Progress, draft-ietf-bfcpbis-rfc4582bis-16, November 2015.

[BFCP-WEBSOCKET] Pascual, V., Roman, A., Cazeaux, S., Salgueiro, G., and R. R, "The WebSocket Protocol as a Transport for the Binary Floor Control Protocol (BFCP)", Work in Progress, draft-ietf-bfcpbis-bfcp-websocket-15, February 2017.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[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>.

[RFC7118] Baz Castillo, I., Millan Villegas, J., and V. Pascual, "The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP)", RFC 7118, DOI 10.17487/RFC7118, January 2014, <http://www.rfc-editor.org/info/rfc7118>.

[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>.

[RFC7395] Stout, L., Ed., Moffitt, J., and E. Cestari, "An Extensible Messaging and Presence Protocol (XMPP) Subprotocol for WebSocket", RFC 7395, DOI 10.17487/RFC7395, October 2014, <http://www.rfc-editor.org/info/rfc7395>.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <http://www.rfc-editor.org/info/rfc7525>.

[RFC7977] Dunkley, P., Llewellyn, G., Pascual, V., Salgueiro, G., and R. Ravindranath, "The WebSocket Protocol as a Transport for the Message Session Relay Protocol (MSRP)", RFC 7977, DOI 10.17487/RFC7977, September 2016, <http://www.rfc-editor.org/info/rfc7977>.

[SDP-MUX] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", Work in Progress, draft-ietf-mmusic-sdp-mux-attributes-16, December 2016.

[WS-API] Hickson, I., Ed., "The WebSocket API", W3C Candidate Recommendation, September 2012, <https://www.w3.org/TR/2012/CR-websockets-20120920/>.

Acknowledgements

Thanks to Christer Holmberg for raising the need for a BFCP-independent SDP attribute for WebSocket Connection URI.

The authors wish to acknowledge Paul Kyzivat, Suhas Nandakumar, Christer Holmberg, Charles Eckel, Dan Wing, Alissa Cooper, and Joel Halpern for their invaluable suggestions and review comments.

Thanks to Mirja Kuehlewind, Alexey Melnikov, Ben Campbell, and Kathleen Moriarty for their comments and feedback during IESG reviews.

Authors' Addresses

Ram Mohan Ravindranath
Cisco Systems, Inc.
Cessna Business Park,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: rmohanr@cisco.com

Gonzalo Salgueiro
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
United States of America

Email: gsalguei@cisco.com