BFCPBIS Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 3, 2016

Ram. Ravindranath
G. Salgueiro
Cisco

May 2, 2016

Session Description Protocol (SDP) WebSocket Connection URI Attribute
draft-ietf-bfcpbis-sdp-ws-uri-02

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

Copyright Notice

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) [RFC7235] 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 WebSocket subprotocol as a reliable transport mechanism between Session Initiation Protocol (SIP)[RFC3261] entities to enable use of SIP in web-oriented deployments. Additionally, [I-D.pd-dispatch-msrp-websocket] defines a new WebSocket sub-protocol 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, [I-D.ietf-bfcpbis-bfcp-websocket] defines a WebSocket sub-protocol as a reliable transport mechanism between Binary Floor Control Protocol (BFCP) [I-D.ietf-bfcpbis-rfc4582bis] entities to enable usage of BFCP in new scenarios.

As defined in Section 3 of [RFC2818], when using Secure WebSockets the Canonical Name (CNAME) of the Secure Sockets Layer (SSL) [RFC6101] certificate MUST match the WebSocket connection URI host. While it is possible to generate self-signed certificates with Internet Providers (IPs) as CNAME, in most cases it is not viable for certificates signed by well known authorities. Thus, there is a need to indicate the connection URI for 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", "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 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.

Applications that use SDP for negotiation and also use secure WebSocket as a transport protocol TLS MAY indicate the connection URI for the WebSocket Client via a new SDP a= media-level attribute defined in Section 3.3.

3.2. ws-uri SDP Attribute

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

Name : ws-uri

Value: ws-uri (defined in Section 3 of [RFC6455])

Usage Level: media

Mux Category: NOT RECOMMENDED

Charset Dependent: no

Example:

```
a=ws-uri:ws://example.com/chat
```

When the 'ws-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 port provided in the 'm= ' line SHALL be ignored too, as the 'a=ws-uri' SHALL provide port number when needed.

3.3. wss-uri SDP Attribute

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

Name : wss-uri

Value: wss-uri (defined in Section 3 of [RFC6455])

Usage Level: media

Mux Category: NOT RECOMMENDED

Charset Dependent: no

Example:

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

When the 'wss-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 secure WebSocket connection. The port provided in the 'm= ' line SHALL be ignored too, as the 'a=wss-uri' SHALL provide port number when needed.

3.4. ws-uri and wss-uri Multiplexing Considerations

Multiplexing characteristics of SDP attributes are described in [I-D.ietf-mmusic-sdp-mux-attributes]. Various SDP attribute multiplexing categories are introduced there.

Section 3.2 and Section 3.3 describes the multiplex category for the new attributes in this document.

There are no multiplexing rules specified for the ws-uri and wss-uri SDP media-level attributes. Additionally, the specification of multiplexing rules for the ws-uri and wss-uri attributes 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 and it is thus considered unlikely to be used in practice. Therefore, the ws-uri and wss-uri SDP media-level attributes defined in Section 3.2 and Section 3.3 for using WebSocket as a transport protocol are not likely to be used with SDP bundle and are consequently categorized as NOT RECOMMENDED for multiplexing.

If future extensions define how to bundle WebSocket then multiplexing rules for the "a=ws-uri:" and "a=wss-uri:" attributes 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=ws-uri" or "a=wss-uri" attribute in the media section 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.

4.2. Generating the Initial Offer

An SDP offerer in order to negotiate WebSocket as a transport MUST indicate the same in the "proto" field of the "m=" line. For example, to negotiate BFCP-over-WebSocket the "proto" value in the "m=" line MUST be TCP/WSS/BFCP if WebSocket is over TLS, else it MUST be TCP/WS/BFCP.

The offerer SHOULD assign the SDP "setup" attribute with a value of "active" (the offerer will be the initiator of the outgoing TCP connection), unless the offerer insists on being a receiver of an incoming connection, in which case the offerer SHOULD use a value of

"passive". The offerer MUST NOT assign an SDP "setup" attribute with a "holdconn" value. If the offerer assigns the SDP "setup" attribute with a value of "passive", the offerer MUST be prepared to receive an incoming TCP connection on the IP and port tuple advertised in the "c=" line and audio/video ports of the BFCP media stream before it receives the SDP answer.

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 setup the TLS /TCP connection and 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 IP address and port of the offerer using the procedures described in [RFC6455]. The answer MUST have an "a=ws-uri" or "a=wss-uri" attribute depending on whether the application is run of WS or WSS. This attribute MUST follow the syntax defined in Section 3. For BFCP application, the "proto" value in the "m=" line MUST be TCP/WSS/BFCP if WebSocket is run on TLS, else it MUST be TCP/WS/BFCP.

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 WebSockets is running over TLS, the server answers back with "a=wss-uri" attribute in the media section of SDP indicating the connection URI:

```
Answer (server):
m=application 50000 TCP/WSS/BFCP *
a=setup:passive
a=connection:new
a=wss-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 being active it MUST initiate the WebSocket connection handshake by sending a GET message on the negotiated media stream, towards the IP address and port of the answerer, as per the procedures described in [RFC6455].

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 "a=connection:existing" attribute in the media section of SDP following the rules mentioned in [RFC4145] if the ws-uri values 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 (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 setup attribute as "passive" so as to accept incoming connection and MUST include "a=wss-uri" or "a=ws-uri" attribute in the media section 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. Security Considerations

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

It is also RECOMMENDED that the application signaling traffic being transported over a WebSocket communication session be protected by using a secure WebSocket connection (using TLS [RFC5246] over TCP).

6. IANA Considerations

NOTE to RFC Editor: Please replace "XXXX" with the number of this RFC.

This document requests that IANA to register the attributes defined in Section 3.2 and Section 3.3 as new values for the SDP att-field under the Session Description Protocol (SDP) Parameters registry, with RFCXXXX as the reference.

7. 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 and Christer Holmberg for their invaluable suggestions and review comments.

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

[I-D.ietf-bfcpbis-bfcp-websocket]
Pascual, V., Roman, A., Cazeaux, S., Salgueiro, G., R, R., and S. Murillo, "The WebSocket Protocol as a Transport for the Binary Floor Control Protocol (BFCP)", draft-ietf-bfcpbis-bfcp-websocket-06 (work in progress), February 2016.

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

[I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-12 (work in progress), January 2016.

[I-D.pd-dispatch-msrp-websocket]
Dunkley, P., Llewellyn, G., Pascual, V., Salgueiro, G., and R. R, "The WebSocket Protocol as a Transport for the Message Session Relay Protocol (MSRP)", draft-pd-dispatch-msrp-websocket-11 (work in progress), May 2016.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[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>.

[RFC6101] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, DOI 10.17487/RFC6101, August 2011, <http://www.rfc-editor.org/info/rfc6101>.

[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>.

[RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, DOI 10.17487/RFC7235, June 2014, <http://www.rfc-editor.org/info/rfc7235>.

[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>.

[WS-API] W3C and I. Hickson, Ed., "The WebSocket API", May 2012.

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
US

Email: gsalguei@cisco.com