



XMPP

XEP-0246: End-to-End XML Streams

Peter Saint-Andre
<mailto:stpeter@jabber.org>
<xmpp:stpeter@jabber.org>
<https://stpeter.im/>

2008-06-18
Version 0.1

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This specification defines methods for communicating via end-to-end XML streams over a logical or physical connection that provides a reliable transport between two endpoints.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1	Introduction	1
2	Initiating an e2e Stream	1
3	Stream Encryption	2
4	Exchanging Stanzas	3
5	Ending an e2e Stream	3
6	Security Considerations	4
7	IANA Considerations	4
8	XMPP Registrar Considerations	4

1 Introduction

XMPP as defined in [XMPP Core](#)¹ does not support direct interaction between endpoints, since it requires a client to authenticate an XML stream with a "home" server and send all of its outbound XML stanzas through that server (which potentially can route those stanzas through a peer server for delivery to the intended recipient). However, in some scenarios it is desirable to establish end-to-end XML streams between two endpoints instead of relying on the standard client-server architecture. These scenarios include:

- Two endpoints cannot access an XMPP server
- Two endpoints want to enforce end-to-end encryption
- Two endpoints want to send a high volume of XMPP traffic but the intermediate servers enforce rate limits

The first scenario is addressed by [Link-Local Messaging](#)². The second and third scenarios are addressed by [Jingle XML Streams](#)³. Both of those technologies result in the establishment of a direct or mediated connection between two endpoints, such as a direct TCP connection, a bytestream through SOCKS5 ([SOCKS5 Bytestreams](#)⁴) or XMPP itself ([In-Band Bytestreams](#)⁵), or other future transport methods such as [TCP Candidates with Interactive Connectivity Establishment \(ICE\)](#)⁶.

Once two endpoints have opened a direct or mediated connection, they can establish an XML stream over that connection for end-to-end ("e2e") communication. We call this an "e2e stream".

2 Initiating an e2e Stream

The initiator and recipient essentially follow the process defined in RFC 6120 to establish XML streams between themselves.

First, the initiator opens an XML stream to the recipient over the negotiated transport.

Listing 1: Opening a Stream

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
```

¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

²XEP-0174: Link-Local Messaging <<http://xmpp.org/extensions/xep-0174.html>>.

³XEP-0247: Jingle XML Streams <<http://xmpp.org/extensions/xep-0247.html>>.

⁴XEP-0065: SOCKS5 Bytestreams <<http://xmpp.org/extensions/xep-0065.html>>.

⁵XEP-0047: In-Band Bytestreams <<http://xmpp.org/extensions/xep-0047.html>>.

⁶TCP Candidates with Interactive Connectivity Establishment (ICE) <<http://tools.ietf.org/html/draft-ietf-mmusic-ice-tcp>>. Work in progress.

```

from='romeo@forza'
to='juliet@pronto'
version='1.0'>

```

In accordance with RFC 6120, the initial stream header SHOULD include the 'to' and 'from' attributes. In the case of XEP-0174, these SHOULD be the username@machine-name advertised in the PTR record. In the case of Jingle XML Streams, these SHOULD be the bare JIDs (<localpart@domain.tld> or <domain.tld>) of the entities as communicated via XMPP.

If the initiator supports stream features and the other stream-related aspects of XMPP 1.0 as specified in RFC 6120, then it SHOULD include the version='1.0' flag as shown in the previous example.

The recipient then responds with a stream header as well:

Listing 2: Stream Header Response

```

<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  from='juliet@pronto'
  to='romeo@forza'
  version='1.0'>

```

If both the initiator and recipient included the version='1.0' flag, the recipient SHOULD also send stream features as specified in RFC 6120:

Listing 3: Recipient Sends Stream Features

```

<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
</stream:features>

```

3 Stream Encryption

The mere exchange of stream headers results in an unencrypted and unauthenticated channel between the two entities. The entities SHOULD upgrade the channel to an encrypted stream using the XMPP STARTTLS command defined in [XMPP Core](#)⁷ using [RFC 5246](#)⁸, optionally followed by SASL negotiation for mutual authentication (see [RFC 4422](#)⁹).

End-to-end XML streams can be negotiated between two XMPP clients, between an XMPP client and a remote XMPP service (i.e., a service with which a client does not have a direct XML stream, such as a remote [Multi-User Chat](#)¹⁰ room), or between two remote XMPP services. Therefore, if standard X.509 certificates are used then a party to an e2e XML stream will

⁷RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

⁸RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 <<http://tools.ietf.org/html/rfc5246>>.

⁹RFC 4422: Simple Authentication and Security Layer (SASL) <<http://tools.ietf.org/html/rfc4422>>.

¹⁰XEP-0045: Multi-User Chat <<http://xmpp.org/extensions/xep-0045.html>>.

present either a client certificate or a server certificate as appropriate. If X.509 certificates are used, they MUST at a minimum be generated and validated in accordance with the certificate guidelines provided in [RFC 6120](#)¹¹; however, applications of end-to-end XML streams MAY define supplemental guidelines for certificate validation in the context of particular architectures, such as XEP-0174 for link-local streams and XEP-0247 for direct or mediated streams negotiated through XMPP servers.

To ease the transition from the PGP-based object encryption method specified in [Current Jabber OpenPGP Usage](#)¹², clients using TLS for e2e streams MAY use the OpenPGP TLS extension defined in [RFC 5081](#)¹³ (if available).

Use of other TLS extensions MAY be appropriate as well, including those defined in [RFC 5246](#)¹⁴ and [RFC 5054](#)¹⁵.

4 Exchanging Stanzas

Once the streams are established, either entity then can send XMPP message, presence, and IQ stanzas, with or without 'to' and 'from' addresses.

Listing 4: Sending a Message

```
<message from='romeo@forza' to='juliet@pronto'>
  <body>M' lady ,_I_would_be_pleased_to_make_your_acquaintance.</body>
</message>
```

Listing 5: A Reply

```
<message from='juliet@pronto' to='romeo@forza'>
  <body>Art thou not Romeo, and a Montague?</body>
</message>
```

5 Ending an e2e Stream

To end the stream, either party closes the XML stream:

Listing 6: Closing the Stream

```
</stream:stream>
```

¹¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

¹²XEP-0027: Current Jabber OpenPGP Usage <<http://xmpp.org/extensions/xep-0027.html>>.

¹³RFC 5081: Using OpenPGP Keys for Transport Layer Security (TLS) Authentication <<http://tools.ietf.org/html/rfc5081>>.

¹⁴RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2 <<http://tools.ietf.org/html/rfc5246>>.

¹⁵RFC 5054: Using the Secure Remote Password (SRP) Protocol for TLS Authentication <<http://tools.ietf.org/html/rfc5054>>.

The other party then closes the stream in the other direction as well:

Listing 7: Closing the Stream

```
</stream:stream>
```

Both parties then SHOULD close the logical or physical connection between them.

6 Security Considerations

End-to-end streams SHOULD be encrypted; see the [Stream Encryption](#) section of this document.

7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)¹⁶.

8 XMPP Registrar Considerations

This document requires no interaction with the [XMPP Registrar](#)¹⁷.

¹⁶The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

¹⁷The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <http://xmpp.org/registrar/>.