XEP-0297: Stanza Forwarding

Matthew Wild
mailto:mwild1@gmail.com
xmpp:me@matthewwild.co.uk

Kevin Smith
mailto:kevin@kismith.co.uk
xmpp:kevin@doomsong.co.uk

2013-10-02
Version 1.0

Status	Type	Short Name
Draft	Standards Track	forward

This document defines a protocol to forward a stanza from one entity to another.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 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 <https://xmpp.org/about/xsf/ipr-policy> or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1 Introduction 1

2 Requirements 1

3 Forwarding a stanza 1
- 3.1 Overview 1
- 3.2 Business rules 2

4 Determining Support 3

5 IANA Considerations 4

6 XMPP Registrar Considerations 4
- 6.1 Protocol Namespaces 4
- 6.2 Protocol Versioning 4

7 Security Considerations 4
- 7.1 As-is 4
- 7.2 Extensions 5

8 XML Schema 5

9 Acknowledgements 6

1 Introduction

There are many situations is which an entity needs to forward a previously sent stanza to another entity, such as forwarding an interesting message to a friend, or a server forwarding stored messages from an archive. Here we specify a simple encapsulation method for such forwards. This format can be used in other specifications requiring the forwarding of stanzas, or used in isolation for a user to forward a message to another user (rather like email forwards).

2 Requirements

Several properties are desirable when forwarding stanzas:

- The original sender and receiver should be identified.
- Most extension payloads should be included (not only a message <body/>).
- It should be possible to annotate the stanza (e.g. with a timestamp) without ambiguity as to the original stanza contents.
- It should allow features such as encryption or signing of the original stanza to survive the forwarding.

3 Forwarding a stanza

3.1 Overview

Let us suppose that a Romeo receives a message from Juliet:

Listing 1: Receiving a message

```
<message from='juliet@capulet.lit/orchard'
         id='0202197'
         to='romeo@montague.lit'
         type='chat'>
  <body>Yet I should kill thee with much cherishing.</body>
  <mood xmlns='http://jabber.org/protocol/mood'>
    <amorous/>
  </mood>
</message>
```

To forward this to Mercutio, Romeo would send a new message with a <forwarded/> payload of namespace 'urn:xmpp:forward:0'.

Listing 2: Forwarding a message

```
<message to='mercutio@verona.lit' from='romeo@montague.lit/orchard'
   type='chat' id='28gs'>
  <body>A most courteous exposition!</body>
  <forwarded xmlns='urn:xmpp:forward:0'>
    <delay xmlns='urn:xmpp:delay' stamp='2010-07-10T23:08:25Z'/>
    <message from='juliet@capulet.lit/orchard'
             id='0202197'
             to='romeo@montague.lit'
             type='chat'
             xmlns='jabber:client'>
      <body>Yet I should kill thee with much cherishing.</body>
      <mood xmlns='http://jabber.org/protocol/mood'>
        <amorous/>
      </mood>
    </message>
  </forwarded>
</message>
```

3.2 Business rules

1. Forwarded stanzas SHOULD include all relevant child elements of the original stanza by default. However, an implementation MAY omit elements it deems irrelevant and safe to discard. An example would be omitting Chat State Notifications (XEP-0085) [1] elements from <message> stanzas which typically do not make sense outside the context of a conversation session. However it should be noted that removing such elements can invalidate any digital signature on a stanza. If preserving a signature is important in the context this extension is used then child elements SHOULD NOT be removed.

2. The forwarding entity SHOULD add a <delay/> child to the <forwarded/> element to indicate to the recipient the date/time that the forwarding entity received the original stanza. The format of this element is described in Delayed Delivery (XEP-0203) [2].

3. The namespace of the forwarded stanza MUST be preserved (this is typically 'jabber:client'). If no 'xmlns' is set for the stanza then as per XML namespacing rules it would inherit the 'urn:xmpp:forward:0' namespace, which is wrong.

4. When this extension is employed simply for a user to forward a given message to a contact, the outer <message/> SHOULD contain a body (even if empty) and a receiving client should pay particular attention to ensure it renders both the sender's text and

[1] XEP-0085: Chat State Notifications <https://xmpp.org/extensions/xep-0085.html>.
[2] XEP-0203: Delayed Delivery <https://xmpp.org/extensions/xep-0203.html>.

the forwarded message unambiguously.

5. When a forwarded stanza forms part of an encapsulating protocol, the <forwarded/> element SHOULD be a child of a tag of that protocol, and SHOULD NOT be included as a direct child of the transmitted stanza.

4 Determining Support

Clients that implement this specification to display simple forwarded messages (i.e. those not part of another extension) SHOULD indicate their support via the 'urn:xmpp:forward:0' feature.

Listing 3: Client requests features of a contact

```
<iq xmlns='jabber:client'
    type='get'
    from='romeo@montague.lit/orchard'
    to='juliet@capulet.lit/balcony'
    id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 4: Contact responds with forwarding feature

```
<iq xmlns='jabber:client'
    type='result'
    to='romeo@montague.net/orchard'
    from='juliet@capulet.lit/balcony'
    id='info1'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:forward:0'/>
    ...
  </query>
</iq>
```

Senders SHOULD NOT forward messages using this protocol to recipients that have not indicated support for it. However they may still format the forward as plain text inside the <body> of a standard message stanza for compatibility with clients lacking support. Such a plain text version SHOULD NOT be included in a stanza using this extension (e.g. as a 'fallback'), as receiving entities are expected to display the <body> of a message as well as any forwarded stanza therein.

5 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [3].

6 XMPP Registrar Considerations

6.1 Protocol Namespaces

The XMPP Registrar [4] includes 'urn:xmpp:forward:0' in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

6.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

7 Security Considerations

Forwarding stanzas can reveal information about the original sender, including possible presence leaks as well as the stanza payloads themselves. Any extensions using this format must therefore consider the implications of this.
Forwarding can either be used as-is, or in the context of another specification, with different security considerations as described below.

7.1 As-is

Receipt of a forwarded stanza from a third-party does not guarantee that the original stanza was actually received, or that the content has not been modified, by the forwarder. Integrity of the original stanza can only be determined through cryptographic signatures for example, which are beyond the scope of this specification.
Considering the above an end-user client should take special care in its rendering of forwarded

[3] The 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/>.

[4] The 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 <https://xmpp.org/registrar/>.

stanzas, such as forwarded messages, to ensure that the user cannot mistake it for a message received directly from the original sender.
An entity SHOULD NOT trust that forwards are genuine when receiving them unprovoked (i.e. outside the scope of another specification).
Forwarded stanzas MUST NOT be processed as if they were non-forwarded stanzas. Furthermore a client SHOULD ignore non-<message> stanzas that are not embedded within another extension.

7.2 Extensions

While security considerations are ultimately dependent upon the specifications using the format defined herein, forwarding introduces scope for stanza forgery such that authors of derivative specifications will need to address security considerations themselves. These need to cover which entities a client should accept forwards from, and which entities those are permitted to forward stanzas for. For example, a specification may choose to only trust forwards if they are received from the user's client, another client on the bare JID, or the user's server.

8 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:forward:0'
    xmlns='urn:xmpp:forward:0'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0297: http://www.xmpp.org/extensions/xep-0297.html
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace='jabber:client'
             schemaLocation='http://xmpp.org/schemas/jabber-client.xsd
                '/>
  <xs:import namespace='jabber:server'
             schemaLocation='http://xmpp.org/schemas/jabber-server.xsd
                '/>
  <xs:import namespace='urn:xmpp:delay'
             schemaLocation='http://xmpp.org/schemas/delay.xsd'/>
```

```
  <xs:element name='forwarded'>
    <xs:complexType>
      <xs:sequence xmlns:delay='urn:xmpp:delay'
                   xmlns:client='jabber:client'
                   xmlns:server='jabber:server'>
        <xs:element ref='delay:delay' minOccurs='0' maxOccurs='1' />
        <xs:choice minOccurs='0' maxOccurs='1'>
          <xs:choice>
            <xs:element ref='client:message'/>
            <xs:element ref='client:presence'/>
            <xs:element ref='client:iq'/>
          </xs:choice>
          <xs:choice>
            <xs:element ref='server:message'/>
            <xs:element ref='server:presence'/>
            <xs:element ref='server:iq'/>
          </xs:choice>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

9 Acknowledgements

Thanks to Kim Alvefur, Jefry Lagrange, Matt Miller, Peter Saint-Andre, Kurt Zeilenga, and Florian Zeitz for their feedback.