XEP-0379: Pre-Authenticated Roster Subscription

Georg Lukas
mailto:georg@op-co.de
xmpp:georg@yax.im

2021-03-04
Version 0.3.3

Status	Type	Short Name
Proposed	Standards Track	pars

This document defines a protocol and URI scheme for pre-authenticated roster links that allow a third party to automatically obtain the user's presence subscription. The goal of this is to make onboarding of new XMPP IM contacts as easy as possible.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

Romeo is an active XMPP IM (Instant Messaging) user. He convinces Juliet (who doesn't have an XMPP account yet) to install a client and register with some server. Now, Romeo only needs to create a mutual presence subscription with her, without yet knowing her JID.
This specification allows Romeo to create an out-of-band link (URI) which, when opened in Juliet's (or another user's) client, will:

- Add Romeo to Juliet's roster (with a display name optionally specified by Romeo)
- Add Juliet to Romeo's roster (without a predefined display name)
- Establish a mutual presence subscription between Romeo and Juliet

The perceivable effect is that with a single click, Romeo and Juliet become "friends" in terms of XMPP presence subscription.

2 Requirements

This specification makes use of XMPP URIs. The basic URI scheme for XMPP is defined in RFC 5122 [1] and extended in XMPP URI Query Components (XEP-0147) [2] to support different actions like "roster" for roster addition and "subscribe" for presence subscription.

3 Pre-Authenticated Roster Subscription

The process of mutual roster addition and subscription involves multiple steps:

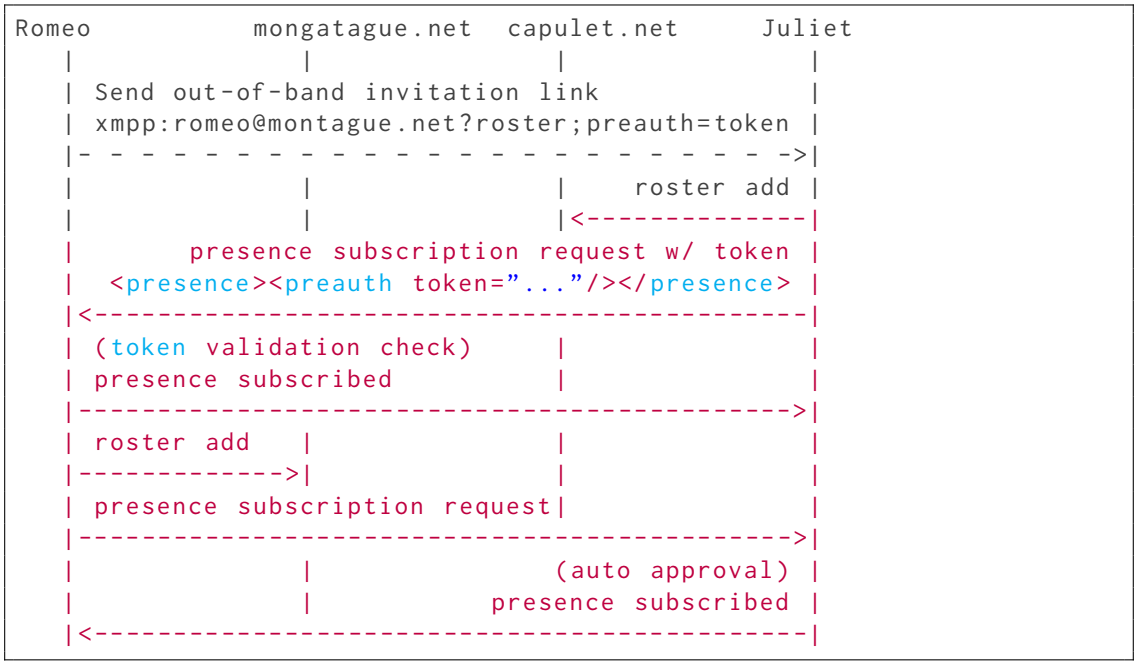
1. Generation of invitation link
2. Out-of-band transmission and presentation of the link
3. Subscription request to the user by the link receiver (new contact)
4. Approval by the user and mutual subscription request
5. Approval by the new contact

The general idea of the protocol and the details of the individual steps are outlined in the following.

[1] RFC 5122: Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP) <http://tools.ietf.org/html/rfc5122>.
[2] XEP-0147: XMPP URI Query Components <https://xmpp.org/extensions/xep-0147.html>.

3.1 General Idea

As Romeo doesn't know Juliet's JID, he needs to send an out-of-band invitation. Later, his client needs to match an incoming subscription request to this invitation, so it can perform a secure automatic roster addition and subscription approval. This matching is accomplished by means of an authentication token, which is generated by Romeo's client, added to the invitation link and then carried over into the subscription request eventually made by Juliet's client. Romeo's client can then compare the token received in a subscription request to the list of issued tokens, and automatically approve the subscription.



3.2 Generation of Invitation Link

Whenever Romeo wishes to invite somebody to his roster, his client will generate an invitation link that contains a new authentication token. This document extends the "roster" URI action defined in XMPP URI Query Components (XEP-0147) [3] with a new key-value parameter named "preauth" to store the generated token. Romeo's client will create an **xmpp:** link containing Romeo's JID, the "roster" action, the "preauth" parameter with the token value, and optionally a "name" parameter with the suggested display name.

Listing 1: Invitation Link with Roster Action and Authentication Token

```
xmpp:romeo@montague.net?roster;preauth=1tMFqYDdKhfe2pwp;name=Romeo+
    Montague
```

[3] XEP-0147: XMPP URI Query Components <https://xmpp.org/extensions/xep-0147.html>.

If the "preauth" parameter is present, the processing client is supposed not only to add the user to the roster, but also to automatically send a subscription request containing the authentication token.

Server-side implementation: it is possible (but out-of-scope for this document), to let the user's server handle generation of links as well as automatic approval of qualified subscription requests. One such approach is documented in Easy User Onboarding (XEP-0401) [4].

3.3 Out-of-band Transmission and Presentation of the Link

As Romeo doesn't know Juliet's JID in advance, he needs to use an out-of-band method (like e-mail, QR codes or NFC) to transmit the invitation link to Juliet. While these methods allow transmission of **xmpp:** URIs, there is no mechanism to ensure that Juliet actually has a client installed that can open the URI.
One way to solve this problem is to present Juliet with a web-based landing page that contains the following elements:

- A short text that this is an XMPP invitation from Romeo.
- A client recommendation (based on the detected web browser/OS) with download links.
- A prominent button that activates the actual **xmpp:** link.

There are multiple options where such a landing page could be hosted:

1. **XSF:** a central place would provide a common ground for a curated client list and ensure long-term availability. However, the operator would be able to collect metadata and abuse authentication tokens.
2. **Client developer:** the developer of Romeo's client can provide a landing page for invitation requests created with this client. This is a feasible short-term solution and allows to recommend the same client as used by the link sender. However, it shares the privacy objections of the XSF solution and can not guarantee long-term availability of the service. If the client development shuts down, invitation links created with the client will cease working.
3. **User's server:** this is the optimal long-term solution, as the user's server is already entrusted with the relevant metadata and will exist at least as long as the user's account on that server. However, this requires an additional server component to query for invitation URIs and a web server hosting the landing page. Furthermore, the server operator needs to maintain the list of recommended clients.

[4] XEP-0401: Easy User Onboarding <https://xmpp.org/extensions/xep-0401.html>.