XEP-0488: MUC Token Invite

Maxime Buquet
mailto:pep@bouah.net
xmpp:pep@bouah.net

2024-03-11
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	muc-token-invite

This specification provides a way to generate tokens to invite users to a MUC room.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 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 Use Cases **1**

3.1 Discovering support 1
3.2 Token generation 1
3.3 Limits 2
3.4 Inviting users 3
3.5 Using a token 3
3.6 Listing tokens 3
3.7 Revoking a token 4

4 Implementation Notes **4**

5 Accessibility Considerations **5**

6 Internationalization Considerations **5**

7 Security Considerations **5**

8 IANA Considerations **5**

9 XMPP Registrar Considerations **5**

10 XML Schema **6**

1 Introduction

This specification provides a way to request invite tokens to a MUC room in order to invite users whose address is unknown to a member-only Multi-User Chat (XEP-0045) [1] room.

2 Requirements

- Allow tokens to be generated, optionally with constraints.
- Allow tokens to be revoked.
- Don't prevent affiliated users of a room to join if they don't possess a token.
- Don't require clients receiving tokens to have any specific implementation.

3 Use Cases

3.1 Discovering support

Supporting entities MUST advertise the urn:xmpp:muc-token-invite:0 Service Discovery (XEP-0030) [2] feature.

3.2 Token generation

An entity may request a token from a Multi-User Chat (XEP-0045) [3] service by sending an iq of type set containing a <request> element in the urn:xmpp:muc-token-invite:0 namespace.

Listing 1: Requesting an invitation token

```
<iq type='set' to='news@commons.example.org' id='request1'>
  <request xmlns='urn:xmpp:muc-token-invite:0'/>
</iq>
```

The MUC room MUST reply to the request with a <token> element in the urn:xmpp:muc-token-invite:0 namespace, containing the token as text node. The token MUST be an opaque string but does not need to be unique within a room.

Listing 2: Room successfully replies with a token

```
<iq type='result' from='news@commons.example.org' to='louise@example.
    org' id='request1'>
```

[1] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[2] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[3] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

```
  <token xmlns='urn:xmpp:muc-token-invite:0'>
     lyQZ1RzacYTlf3svGODYq1xVabNnMc2x</token>
</iq>
```

Implementations MUST reply an error ot type auth/forbidden if the requesting entity isn't allowed to generate a token.

Listing 3: Room replies with an error because user lacks permissions

```
<iq type='result' from='news@commons.example.org' to='louise@example.
    org' id='request1'>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

3.3 Limits

It is possible to create tokens that may be used only a specific number of times to grant users affiliations, and/or may have an expiry time.
To constrain the token to a number of times after which it expires, the counter attribute (xs:unsignedInt) can be used in the <request> element.
To constrain the token to a time limit, the delay attribute (xs:unsignedInt) can be used in the <request> element.
if both attributes are combined, whichever constraint is reached first expires the token.

Listing 4: A client requests a token with a month time limit and 5 users

```
<iq type='set' to='news@commons.example.org' id='request2'>
  <request xmlns='urn:xmpp:muc-token-invite:0' delay='2678400' counter
     ='5'/>
</iq>
```

The reply from the service MUST contain at least the requested delay and counter attributes. Requested values for these attributes MAY be altered by the server. This may be useful to implement a default server policy (maximum time, and/or counter). Values returned indicate current values that apply to the issued token.
Services may want to automatically limit issued tokens even with the request doesn't have any. In the following example, the MUC service enforces a maximum time limit of a week as a policy.

Listing 5: Reply from the room

```
<iq type='result' from='news@commons.example.org' to='louise@example.
    org' id='request2'>
  <token xmlns='urn:xmpp:muc-token-invite:0'
         delay='604800'
```

```
        counter='5'
        >lyQZ1RzacYTlf3svGODYq1xVabNnMc2x</token>
</iq>
```

3.4 Inviting users

Integration with Mediated Invites or Direct MUC Invitations (XEP-0249) [4] is not described in this document as invite tokens generated this way may not be used when the invitee's address is known.
Clients may include generated tokens in the password parameter of a URI as such:

Listing 6: URI with an invite token

```
xmpp:news@commons.example?join;password=TOKEN
```

3.5 Using a token

Receiving entities will follow the usual flow of joining password protected-rooms.
When a token is used by a participant who doesn't have any affiliation, a server MUST give them an affiliation level of member.
If an expired token is used by someone who isn't affiliated yet, the room MAY additionally include in the presence error an <expired-token/> element in the urn:xmpp:muc-token-invite:0 namespace, as a sibling of the <not-authorized/> element.

3.6 Listing tokens

It is possible for room participants to list tokens by sending an iq of type get containing a <tokens/> element in the urn:xmpp:muc-token-invite:0 namespace.
The room MUST reply with all tokens that the participant is allowed to revoke, each listed in <token> elements within a <tokens> wrapper element. Individual token elements MUST contain updated attibute values, that is, if a token has been issued with counter set to 5 and has been used twice (2), listing tokens at this point will show this specific token with a counter attribute value of 3.

Listing 7: Requesting the token list

```
<iq type='get' to='news@commons.example.org' id='request3'>
  <tokens xmlns='urn:xmpp:muc-token-invite:0'/>
</iq>

<iq type='result' to='louise@example.org' id='request3'>
  <tokens xmlns='urn:xmpp:muc-token-invite:0'>
```

[4] XEP-0249: Direct MUC Invitations <https://xmpp.org/extensions/xep-0249.html>.

```
    <token counter='3'>lyQZ1RzacYTlf3svGODYq1xVabNnMc2x</token>
    <token delay='183'>HIFac1EUx3gDA1TEXlblwQ2izGIqAUab</token>
  </tokens>
</iq>
```

3.7 Revoking a token

It is possible to revoke a token early by sending an iq containing a <revoke> element in the urn:xmpp:muc-token-invite:0 namespace, with the token as the text node. The room MUST then reply successfully with an empty iq.
If the user is unauthorized to issue tokens, the room should reply with an iq error type auth/forbidden. If the user is unauthorized to revoke the specified token, or if the token doesn't exist, the room should reply with an iq error of type cancel/item-not-found.

Listing 8: A client revokes a token

```
<iq type='set' to='news@commons.example.org' id='request4'>
  <revoke xmlns='urn:xmpp:muc-token-invite:0'>
     lyQZ1RzacYTlf3svGODYq1xVabNnMc2x</revoke>
</iq>
```

Listing 9: The revocation succeeds

```
<iq type='result' to='louise@example.org' id='request4'/>
```

Listing 10: The revocation fails because the user is not allowed to revoke specified token

```
<iq type='result' to='louise@example.org' id='request4'>
  <error type='cancel'>
    <item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

4 Implementation Notes

Tokens may be added to bookmark storage by receiving entities and as such implementing MUC rooms SHOULD ignore tokens provided during join when a user is already affiliated with the room. In this case, if a counter was attached it SHOULD NOT be decremented.
Tokens with no constraint are not equivalent to passwords. A token is only required to be supplied once as opposed to passwords, which need to be specified at every join independently of user affiliation.
The Using a token section describes a way for clients to know they may have used an invalid token by adding an error specific to this document. It is likely that tokens aren't stored indefinitely but rather removed from storage not long after they expire, which makes it

hard for MUC services to distinguish between a password for the room before configuration change, and an expired token. This specification assumes that it was an expired token as the room isn't password protected.
Possible extensions to this spec could include broadcasting information about the inviter in a new participant's join presence, as well as issuing tokens with specific affiliations and/or Hats (XEP-0317) [5].

5 Accessibility Considerations

None?

6 Internationalization Considerations

None?

7 Security Considerations

Leaking tokens may lead to inviting unwelcomed people to a room. Token limits and revocations provide users a way to reduce harm in such a case. A service SHOULD also enforce a reasonable maximum value as a time or usage constraint (24h, a week, a year, etc.).
Issuing tokens may be locked down by service operators, or by room administrators via the muc#roomconfig_allowinvites Multi-User Chat (XEP-0045) [6] configuration option.
It is RECOMMENDED that room moderators be able to list and revoke tokens generated by every other participant.

8 IANA Considerations

None.

9 XMPP Registrar Considerations

None.

[5] XEP-0317: Hats <https://xmpp.org/extensions/xep-0317.html>.
[6] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.

10 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:muc-token-invite:0'
  xmlns='urn:xmpp:muc-token-invite:0'
  elementFormDefault='qualified'>

  <xs:annotations>
    <xs:documentation>
      The protocol documented by this schema is defined
      in XEP-xxxx: https://xmpp.org/extensions/xep-xxxx.html.
    </xs:documentation>
  </xs:annotations>

  <xs:element name='request'>
    <xs:complexType>
      <xs:attribute name='delay' type='xs:unsignedInt' use='optional'/
          >
      <xs:attribute name='counter' type='xs:unsignedInt' use='optional
          '/>
    </xs:complexType>
  </xs:element>

  <xs:element name='token' type='xs:string'>
    <xs:complexType>
      <xs:attribute name='delay' type='xs:unsignedInt' use='optional'/
          >
      <xs:attribute name='counter' type='xs:unsignedInt' use='optional
          '/>
      <xs:attribute name='creator' type='xs:string' use='optional'/>
    </xs:complexType>
  </xs:element>

  <xs:element name='tokens'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='token' minOccurs='1'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='revoke' type='xs:string'/>

  <xs:element name='expired-token'>
    <xs:simpleType>
      <xs:restriction base='xs:string'>
        <xs:enumeration value=''/>
```

```
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```