XEP-0178: Best Practices for Use of SASL EXTERNAL with Certificates

Peter Saint-Andre
mailto:peter@andyet.net
xmpp:stpeter@stpeter.im
https://stpeter.im/

Peter Millard

2011-05-25
Version 1.1

Status	Type	Short Name
Active	Informational	N/A

This document specifies best practices for XMPP usage of the SASL EXTERNAL mechanism in the context of PKIX certificates.

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 Client-to-Server Recommendation 1

3 Server-to-Server Recommendation 5

4 Security Considerations 8

5 IANA Considerations 8

6 XMPP Registrar Considerations 9

7 Acknowledgements 9

8 Author Note 9

1 Introduction

XMPP as specified in RFC 3920 [1] and updated in RFC 6120 [2] allows the use of any SASL (RFC 4422 [3]) mechanism in the authentication of XMPP entities. This document specifies a recommended protocol flow for use of the SASL EXTERNAL mechanism with PKIX (RFC 5280 [4]) certificates [5], especially when an XMPP service indicates that TLS is mandatory-to-negotiate.

2 Client-to-Server Recommendation

As specified in RFC 3920 and updated in RFC 6120, during the stream negotiation process an XMPP client can present a certificate (a "client certificate"). If a JabberID is included in a client certificate, it is encapsulated as an id-on-xmppAddr Object Identifier ("xmppAddr"), i.e., a subjectAltName entry of type otherName with an ASN.1 Object Identifier of "id-on-xmppAddr" as specified in Section 13.7.1.4 of RFC 6120.
There are three possible cases:

1. The certificate includes one xmppAddr.
2. The certificate includes more than one xmppAddr.
3. The certificate includes no xmppAddr.

This specification includes recommendations that address all three cases.
The RECOMMENDED protocol flow for client-to-server use of SASL EXTERNAL with client certificates is as follows:

1. Client initiates stream to server.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:client'
        from='juliet@example.com'
        to='example.com'
        version='1.0'>
```

[1] RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc3920>.
[2] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[3] RFC 4422: Simple Authentication and Security Layer (SASL) <http://tools.ietf.org/html/rfc4422>.
[4] RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile <http://tools.ietf.org/html/rfc5280>.
[5] This specification focuses on the use of the SASL EXTERNAL mechanism with X.509 certificates. Future specifications might document best practices for use of SASL EXTERNAL outside the context of the X.509 infrastructure, for example via Internet Protocol Security (IPSec) as specified in RFC 4301 [6].

2. Server replies with stream header.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:client'
        id='c2s_234'
        from='example.com'
        to='juliet@example.com'
        version='1.0'>
```

3. Server advertises TLS stream feature, which might indicate that TLS is mandatory-to-negotiate.

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

4. Client sends STARTTLS command to server.

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

5. Server informs client to proceed.

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

6. Server requests, and client presents, the client certificate during TLS negotiation.

7. Server and client successfully complete the TLS negotiation and client initiates a new initial stream header to server over the encrypted TCP connection.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:client'
        from='juliet@example.com'
        to='example.com'
        version='1.0'>
```

8. Server replies with response stream header.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:client'
        id='c2s_345'
        from='example.com'
        to='juliet@example.com'
        version='1.0'>
```

9. Server advertises SASL mechanisms. Because the client presented a client certificate, here the server offers the SASL EXTERNAL mechanism (see Section 6.3.4 of RFC 6120 for recommendations regarding the conditions under which to offer the SASL EXTERNAL mechanism).

```
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>EXTERNAL<mechanism>
    <mechanism>DIGEST-MD5<mechanism>
    <mechanism>ANONYMOUS<mechanism>
  </mechanisms>
</stream:features>
```

10. Client considers EXTERNAL to be its preferred SASL mechanism so it attempts to complete SASL negotiation using that mechanism. The following paragraphs illustrate several possible paths, depending on whether the client includes an authorization identity (for the official rules regarding when to include the authorization identity, see Section 6.3.8 of RFC 6120).

    a) If the client certificate contains only one JID, then the client MAY include an authorization identity, but only if it desires to be authorized as a JID other than the address in the client certificate; else it MUST NOT include an authorization identity (this is shown in the following example by setting the XML character data of the <auth/> element to "="):

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
      mechanism='EXTERNAL'>=</auth>
```

    b) If the client certificate contains more than one JID, then the client MUST include an authorization identity so that the server can determine which JID to use (this is shown in the following example by setting the XML character data of the <auth/> element to "anVsaWV0QGV4YW1wbGUuY29t", which is the base 64 encoding for "juliet@example.com").

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
      mechanism='EXTERNAL'>anVsaWV0QGV4YW1wbGUuY29t</auth>
```

c) If the client certificate does not contain a JID, then the client MAY include an authorization identity, but only if it desires to be authorized as a JID other than the address specified during SASL negotiation; else it MUST NOT include an authorization identity (this is shown in the following example by setting the XML character data of the <auth/> element to "="):

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
      mechanism='EXTERNAL'>=</auth>
```

11. Server determines whether to allow authentication and authorization of user.

a) If (1) the certificate presented by the client contains only one valid XMPP address that corresponds to a registered account on the server and (2) the client did not pass an authorization identity in the SASL exchange, then the server SHOULD allow authentication and authorization of that JID.

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

b) If the certificate contains more than one valid XMPP address that corresponds to a registered account on the server (e.g., because the server offers virtual hosting) and during the SASL exchange the client specified an authorization identity that corresponds to one of the JIDs presented in the client certificate, then the server SHOULD allow authentication and authorization of the JID specified as the authorization identity.

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

If no authorization identity is included, then the server SHOULD return a SASL failure case of <invalid-authzid/> and close the stream.

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  <invalid-authzid/>
</failure>
</stream:stream>
```

c) If the certificate does not contain an XMPP address, then the server MAY attempt to determine if there is a registered account associated with the user, for example by performing an LDAP lookup based on the Common Name or other information presented by the client in the certificate; if such a JID mapping is successful and

the mapped JID matches the authorization identity provided, then the server SHOULD allow authentication and authorization of that mapped JID.

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

If JID mapping is unsuccessful, then the server SHOULD return a SASL failure condition of <not-authorized/> and close the stream.

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  <not-authorized/>
</failure>
</stream:stream>
```

If JID mapping is successful but the mapped JID does not match the authorization identity provided (if any), then the server SHOULD return a SASL failure condition of <invalid-authzid/> and close the stream.

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  <invalid-authzid/>
</failure>
</stream:stream>
```

12. If SASL authentication succeeded, the client opens a new stream, then client and server proceed with resource binding as described in RFC 6120.

3 Server-to-Server Recommendation

RFC 3920 specified that if a JabberID is included in a certificate intended for use by an XMPP server (a "server certificate"), it shall be encapsulated as an xmppAddr. That recommendation is updated in RFC 6120 through a reference to RFC 6125 [7], which prefers use of a dNSName and/or SRVName entry in the Subject Alternative Name. The DNS domain name contained in the certificate can be a fully-qualified domain name ("FQDN") or a so-called "wildcard" with the '*' character as the complete left-most label (see RFC 6125 for complete details).
The RECOMMENDED protocol flow for server-to-server use of SASL EXTERNAL with server (domain) certificates is as follows:

1. Server1 initiates stream to server2.

[7] RFC 6125: Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) <http://tools.ietf.org/html/rfc6125>.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='conference.example.org'
        to='example.com'
        version='1.0'>
```

2. Server2 replies with stream header.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='s2s_234'
        from='example.com'
        to='conference.example.org'
        version='1.0'>
```

3. Server2 advertises TLS stream feature, which might indicate that TLS is mandatory-to-negotiate.

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

4. Server1 sends STARTTLS command to Server2.

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

5. Server2 informs Server1 to proceed.

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

6. Server2 requests, and Server1 presents, Server1's certificate during TLS negotiation.

7. Server2 validates certificate in accordance with the rules from RFC 6120 and RFC 6125.

   a) If certificate is unacceptable for the reasons explained in RFC 6120 and RFC 6125, Server2 closes Server1's TCP connection.

b) Else Server2 completes successful TLS negotiation and Server1 sends a new initial stream header to Server2 over the encrypted TCP connection.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        from='conference.example.org'
        to='example.com'
        version='1.0'>
```

8. Server2 replies with stream header.

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:server'
        id='s2s_345'
        from='example.com'
        to='conference.example.org'
        version='1.0'>
```

9. Server2 advertises SASL mechanisms. If the 'from' attribute of the stream header sent by Server1 can be matched against one of the identifiers provided in the certificate following the matching rules from RFC 6125, Server2 SHOULD advertise the SASL EXTERNAL mechanism. If no match is found, Server2 MAY either close Server1's TCP connection or continue with a Server Dialback (XEP-0220) [8] negotiation.

```
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>EXTERNAL<mechanism>
  </mechanisms>
</stream:features>
```

10. Server1 considers EXTERNAL to be its preferred SASL mechanism. For server-to-server authentication, the <auth/> element MAY include an authorization identity, however a future version of this specification might disallow use of the authorization identity in server-to-server authentication (in the following example, Server1 includes an empty response of "=" as shown in RFC 6120).

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl'
      mechanism='EXTERNAL'>=</auth>
```

[8] XEP-0220: Server Dialback <https://xmpp.org/extensions/xep-0220.html>.

Interoperability Note: Previous versions of this specification stated that the receiving server always relied on the connecting server's inclusion of the authorization identity. Even though this is no longer required, the connecting server SHOULD include the authorization identity for backward compability.

11. Server2 determines if hostname is valid.

    a) If the 'from' attribute of stream header sent by Server1 can be matched against one of the identifiers provided in the certificate following the matching rules from RFC 6125, Server2 returns success.

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

    Implementation Note: If Server2 needs to assign an authorization identity during SASL negotiation, it SHOULD use the value of the 'from' attribute of the stream header sent by Server1.

    b) Else Server2 SHOULD return a <not-authorized/> stream error and close the stream.

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
  <not-authorized/>
</failure>
</stream:stream>
```

4 Security Considerations

This document introduces no security considerations or concerns above and beyond those discussed in RFC 6120 and RFC 6125.

5 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [9].

[9] 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/>.

6 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [10].

7 Acknowledgements

Thanks to Dave Cridland, Philipp Hancke, Joe Hildebrand, Justin Karneges, Chris Newton, Rob Norris, and Matthias Wimmer for their comments.

8 Author Note

Peter Millard, co-author of the initial version of this specification, died on April 26, 2006. The remaining author appreciates his assistance in defining the best practices described herein.

[10] 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/>.