This specification defines a method to manage client certificates that can be used with SASL External to allow clients to log in without a password.
Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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).
2 Certificate Management

To manage their certificates, this protocol describes a way for clients to store, query and remove certificates on their server. These certificates can be generated by the client itself, for example to replace a password-based login, or the certificates can be imported by the user.

2.1 Add a X.509 Certificate

When a user wants to add a certificate, the client uploads it to the server. It does this by sending an <iq/> with an <append/> payload qualified by the 'urn:xmpp:saslcert:1' namespace. The

---

<append/> element MUST include an <x509cert/> element containing the Base64 encoded DER data of the certificate. The client also MUST provide a unique name for the certificate as a <name/> element to make it possible for the user to manage their different client certificates.

**Listing 1: Client Uploads an X.509 Certificate. Whitespace only added for presentation.**

```
<iq type='set'
   from='hamlet@shakespeare.lit/denmark'
   id='append'>
   <append xmlns='urn:xmpp:saslcert:1'>
     <name>Mobile Client</name>
     <x509cert>
       MIICCTCCAXKgAwIBAgIJALhU0Id6xxwQA0GCSqGSIb3QEJBQUAMA4xDADAKBgNV
       BAMTA2ZvzbAGIjalhU0Id6xxwQA0GCSqGSIb3QEJBQUAMA4xDADAKBgNV
       BAMTA2ZvzbAGIjalhU0Id6xxwQA0GCSqGSIb3QEJBQUAMA4xDADAKBgNV
       RMINLKr+CxqozF14DqKXlwGzQyR49kK6abZ9sSfspT8ja2uNpwI1U32748t
       qSA6paCWbHCc+e7/hm5KymX8Hl3Vjfb/dl0w3rtxj19JRFgryWyxvndNZUpN2s3D
       hKDFvgy9sP/xZp8d/ubbaRxqZKzCAwEAAaNvMG0wHQYDVR0OBBYEJFJwFq3SRGcx
       QXm9qFgF+XBWkeML4MD4GA1UdwQ3MDWAFJWwFq3SRGcXYXm9qFgF+XBWkeML4orRkK
       EDAOMQwwCgYDQwDEwNmb2+CCQC4VCNChesc EDAMoGwNhRMHMBTADAHQ/M8GCSqG
       S1b3QEJBQUAMA4GBAiI1eG20d0msNVXyWAXg21RsJt9INHJQTCJMoUeTtaRjypp
       ffJtuopguNBNdN+MjEp2/+/zLNMaDYLxCaTvmbF6zvY0hzB9i8kNTh23F5b5j+yK
       QChPXXu00EGCaODWhfKhRNdseUozfNWOz91TgMGw8eYNL11QRL//IAOF0r/8
     </x509cert>
   </append>
</iq>
```

The server either returns an empty result or an error. The server MUST return an error when a client tries to add a certificate using a name that is already used by a certificate. From then on the user can use that certificate to authenticate using SASL EXTERNAL.

The client adding the certificate does not need to be the client using it. A user may use a client to upload a certificate for a bot. The bot creates its certificate and private key, and the user uploads the certificate to the server. After that procedure the bot can log in to the XMPP network without ever knowing the user’s password.

An optional element <no-cert-management/> inside the append element indicates that a client logged in with that certificate is not allowed to add or remove certificates from the list. A server MAY allow such a client to query the list of certificates.

**Listing 2: X.509 Certificate Upload (no-cert-management)**

```
<iq type='set'
   from='hamlet@shakespeare.lit/denmark'
   id='nocert'>
   <append xmlns='urn:xmpp:saslcert:1'>
     <name>Simple Bot</name>
     <no-cert-management/>
     <x509cert>
       ...
     </x509cert>
   </append>
</iq>
```
2 CERTIFICATE MANAGEMENT

```xml
<iq type='get'
    from='hamlet@shakespeare.lit/denmark'
    id='query'>
  <items xmlns='urn:xmpp:saslcert:1'/>
</iq>
```

2.2 Request a List of all Certificates

A user may want to get a list of all certificates that can be used for SASL EXTERNAL. The client can query the list of the items by sending an <iq/> stanza with an <items/> payload qualified by the 'urn:xmpp:saslcert:1' namespace.

Listing 3: Client Requests List of X.509 Certificates

```xml
<iq type='result'
    to='hamlet@shakespeare.lit/denmark'
    id='query'>
  <items xmlns='urn:xmpp:saslcert:1'>
    <item>
      <name>Mobile Client</name>
      <x509cert>...
    </item>
    <item>
      <name>Laptop</name>
      <x509cert>...
    </item>
  </items>
</iq>
```

The server then returns the list of all known certificates including the user provided name. For every certificate, the server MAY include a <users/> element with a <resource/> for every resource that is currently logged in and used that certificate with SASL EXTERNAL.

Listing 4: Server Sends List of Known X.509 Certificates
2.3 Disable a Certificate

To make it easier to transition to a new certificate when the current one expires, without requiring the user to enter their password again, a client can upload a new certificate to replace its current certificate. After the new certificate is added to the server, it MAY want to disable the old certificate to keep the list of certificates short. Without that, the list will grow indefinitely, making the certificate handling for the user more difficult. Once a certificate is removed it can no longer be used for SASL EXTERNAL. A server MAY automatically remove expired certificates for this list.

A client can disable a certificate by sending an `<iq/>` stanza containing a `<disable/>` element qualified by the `urn:xmpp:saslcert:1` namespace and containing a `<name/>` element with the user-provided name for the certificate.

Listing 5: Client disables an X.509 Certificate

```
<iq type='set'
    from='hamlet@shakespeare.lit/denmark'
    id='disable'>
  <disable xmlns='urn:xmpp:saslcert:1'>
    <name>Mobile Client</name>
  </disable>
</iq>
```

2.4 Revoke a Certificate

The user may want to revoke a certificate from a stolen or compromised device. The mechanism is similar to disabling a certificate. The difference is that if a client is logged in with that compromised certificate using SASL EXTERNAL, the server SHOULD close the stream to that client forcing a log out of the client.

A client can revoke a certificate by sending an `<iq/>` stanza containing a `<revoke/>` element qualified by the `urn:xmpp:saslcert:1` namespace and containing a `<name/>` element with the user provided name for the certificate.

Listing 6: Client revokes an X.509 Certificate

```
<iq type='set'
    from='hamlet@shakespeare.lit/denmark'
    id='revoke'>
  <revoke xmlns='urn:xmpp:saslcert:1'>
    <name>Mobile Client</name>
  </revoke>
</iq>
```
3 SASL EXTERNAL

When using SASL EXTERNAL, the certificate does not need to be signed by a trusted entity if the certificate was uploaded by a user. The server still MUST reject the certificate if it is expired. In a company environment the server MAY only accept signed certificates; the behavior depends on the company’s security policy. A free public XMPP server MUST allow self-signed certificates and certificates signed by a CA unknown to the server. The client certificate SHOULD include a JID as defined in sections 17.2.1.2. and 17.2.1.3. in RFC 6120: a JID MUST be represented as an XmppAddr, i.e., as a UTF8String within an otherName entity inside the subjectAltName.

Listing 7: subjectAltName using the “id-on-xmppAddr” format

subjectAltName=otherName:id-on-xmppAddr;UTF8:hamlet@shakespeare.lit

The protocol flow is almost the same as described in XEP-0178, except for the resource binding in step 12. If the client used a certificate with a single XmppAddr field and that field contained a full JID, or if the certificate had multiple XmppAddr fields and the address that was used as authorization identity was a full JID, then the server should only allow the client to use the resource of that JID. If a client with the same resource is currently logged in it SHOULD be logged out by the server.

4 Determining Support

If a server supports storage of client side certificates, it MUST advertise that fact by including the feature “urn:xmpp:saslcert:1” in response to a Service Discovery (XEP-0030) request:

Listing 8: Client queries for server features

<iq type='get' id='disco1' to='shakespeare.lit' from='hamlet@shakespeare.lit/denmark'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>

Listing 9: Server responds with features

<iq type='result' id='disco1' from='shakespeare.lit' to='hamlet@shakespeare.lit/denmark'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:saslcert:1'/>
    ...
  </query>
</iq>

5 Security Considerations

5.1 Stream Characteristics

This specification allows the user to manipulate an alternative way to log into the server. The certificates are not required to be signed and any certificate can be used. Therefore the server MUST reject any communication described in this document if the link between client and server is not secured with both STARTTLS and SASL to prevent a man-in-the-middle from modifying the certificate.

5.2 Changing the Password

In-Band Registration (XEP-0077) \(^7\) defines a mechanism to change the password without knowing the current one. If the server supports password change it MUST return not-authorized for clients logged in using SASL EXTERNAL and MAY include a password change form requiring the old password.

---

**Listing 10: Password Change**

```xml
<iq type='set' to='shakespeare.lit' id='change1'>
  <query xmlns='jabber:iq:register'>
    <username>hamlet</username>
    <password>newpass</password>
  </query>
</iq>
```

**Listing 11: Server Returns Password Change Form With Error**

```xml
<iq type='error' from='shakespeare.lit' to='hamlet@shakespeare.lit/denmark' id='change1'>
  <query xmlns='jabber:iq:register'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Password Change</title>
      <instructions>Use this form to change your password.</instructions>
      <field type='hidden' var='FORM_TYPE'>
        <value>jabber:iq:register:changepassword</value>
      </field>
      <field type='text-single' label='Username' var='username'>
        <required/>
      </field>
      <field type='text-private' label='Old Password' var='old_password'>
        <required/>
      </field>
    </x>
  </query>
</iq>
```

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) 8.

7 XMPP Registrar Considerations

7.1 Protocol Namespaces

This specification defines the following XML namespace:

- urn:xmpp:saslcert:1

Upon advancement of this specification from a status of Experimental to a status of Draft, the XMPP Registrar 9 shall add the foregoing namespace to the registry located at <https://xmpp.org/registrar/namespaces.html>, as described in Section 4 of XMPP Registrar Function (XEP-0053) 10.

7.2 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.

---

8 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/>.

9 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/>.

8 XML Schema

The XML schema will be provided in a later version of this document.