XEP-0480: SASL Upgrade Tasks

Thilo Molitor
mailto:thilo+xmpp@eightysoft.de
xmpp:thilo.molitor@juforum.de

2023-05-04
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	sut

This specification provides a way to upgrade to newer SASL mechanisms using SASL2 tasks.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

While Extensible SASL Profile (XEP-0388) [1] provides a modern and extensible way to use SASL in XMPP, it lacks support for SASL mechanism upgrades.
Modern XMPP server deployments typically store only the hash of a user's password, to improve account security. At times, it may be desirable for servers to upgrade to newer or different hash algorithms, e.g. so they can offer different authentication mechanisms for improved security or interoperability. Due to the security properties of one-way cryptographic hash algorithms, it is not possible for the server to extract the original data and simply hash it in a new format. To perform such upgrades, the cooperation of the client is necessary - because it has, or can obtain from the user, the original password to derive a hash from.
This specification fills that gap by providing a pluggable way to perform such SASL mechanism upgrades using Extensible SASL Profile (XEP-0388) [2] tasks to provide the server with the needed data it does not yet have.
This specification also provides a concrete definition of SCRAM upgrade tasks in Section 3.

2 Protocol

Clients capable of SASL mechanism upgrades defined herein MUST send the desired bare JID they want to authenticate for in the "from" attribute of the stream-header unless they don't know it (e.g. when using the GSS-API SASL mechanism etc.) according to section 4.7.1 of RFC 6120 [3]. Providing the bare JID in the "from" attribute, rather than introducing additional nonzas, saves one round-trip, see Extensible SASL Profile (XEP-0388) [4].

2.1 Server advertising possible upgrade tasks

To inform the client which SASL mechanism upgrades it supports, the server adds <upgrade/> elements in the namespace "urn:xmpp:sasl:upgrade:0", each containing the name of one upgrade task, to the SASL2 <authentication/> element inside the stream features.
Upgrade task names SHOULD have a prefix of "UPGR-" (to distinguish them from "normal" SASL mechanisms) followed by the SASL mechanism name to upgrade to, and if multiple mechanisms differ only in their support for channel-binding (e.g. SCRAM's -PLUS variants), implementations MUST use only the names of variants without channel-binding for the task names, because mechanism upgrades are independent of any channel-binding. Finally, upgrade tasks MUST NOT transmit plaintext passwords (or any reversible encoding of them) if the SASL mechanism to upgrade allows this to be avoided.

[1] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.
[2] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.
[3] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[4] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.

Listing 1: Server advertises support for SASL mechanism upgrades to fictional BLOOP2 and BLOOP-42 mechanisms

```
<!-{}- Client sending stream header -{}->
<stream:stream
  from='user@example.org'
  to='example.org'
  version='1.0'
  xml:lang='en'
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'>

<!-{}- Server responding with stream header and features -{}->
<stream:stream
  from='example.org'
  id='++TR84Sm6A3hnt3Q065SnAbbk3Y='
  to='user@example.org'
  version='1.0'
  xml:lang='en'
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'>
<stream:features>
  <authentication xmlns='urn:xmpp:sasl:2'>
    <mechanism>SCRAM-SHA-1</mechanism>
    <mechanism>SCRAM-SHA-1-PLUS</mechanism>
    <inline>
      <!-{}- Server indicates that XEP-0198 stream resumption can be
          done "inline" -{}->
      <resume xmlns='urn:xmpp:sm:3'/>
      <!-{}- Server indicates support for XEP-0386 Bind 2 -{}->
      <bind xmlns='urn:xmpp:bind2:1'/>
    </inline>
    <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-BLOOP2</upgrade>
    <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-BLOOP-42</upgrade>
  </authentication>
  <!-{}- Channel-binding information provided by XEP-0440 -{}->
  <sasl-channel-binding xmlns='urn:xmpp:sasl-cb:0'>
    <channel-binding type='tls-server-end-point'/>
    <channel-binding type='tls-exporter'/>
  </sasl-channel-binding>
</stream:features>
```

2.2 Client requesting SASL upgrades

The client SHOULD always request one or more upgrade tasks it recognises. To do this, it includes the <upgrade/> element namespaced to "urn:xmpp:sasl:upgrade:0" in its <authenticate/> element listing the upgrade tasks it wants to perform, as specified in the Initiation

section of Extensible SASL Profile (XEP-0388) [5], one <upgrade> element for each task.
Upon successfully authenticating the client (including any secondary authentication steps required for the account), but before the final <success/> would be sent, the server sends a <continue/> element, which MUST contain a single task, matching whatever was selected by the client. If the client selected more than one upgrade task, as sequence of upgrade tasks occur. The client then initiates this upgrade task by providing a corresponding <next/> element providing the task name and optionally including any further child-elements as defined by the specification for this concrete upgrade task.

Listing 2: Client requests upgrades for fictional BLOOP2 and BLOOP-42 mechanisms

```
<!-{}- Client sends authentication request, requesting upgrades to
   BLOOP2 and BLOOP-42-{}->
<authenticate xmlns='urn:xmpp:sasl:2' mechanism='SCRAM-SHA-1-PLUS'>
  <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-BLOOP2</upgrade>
  <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-BLOOP-42</upgrade>
  <!-{}- Base64 of: 'p=tls-exporter,,n=user,r=12C4CD5C-E38E-4A98-8F6D
     -15C38F51CCC6' -{}->
  <initial-response>
     cD10bHMtZXhwb3J0ZXIsLG49dXNlcixyPTEyQzRDRDVDLUUzOEUtNEE5OC04RjZELTE1QzM4RjUxQ0ND
     ==</initial-response>
  <user-agent id='d4565fa7-4d72-4749-b3d3-740edbf87770'>
    <software>AwesomeXMPP</software>
    <device>Kiva's␣Phone</device>
␣␣</user-agent>
</authenticate>
```

Listing 3: Server initiates first upgrade task for BLOOP2

```
<!-{}- Client authenticates using SCRAM-SHA-1-PLUS (or whatever
   mechanism was selected)-{}->
[...]

<!-{}- The Server requests the client to perform the first requested
   upgrade task for BLOOP2 -{}->
<continue xmlns='urn:xmpp:sasl:2'>
  <additional-data>
    SSdtIGJvcmVkIG5vdy4=
  </additional-data>
  <tasks>
    <task>UPGR-BLOOP2</task>
  </tasks>
  <text>This account requires an upgrade to BLOOP2 as requested by the
      client</text>
</continue>
```

[5] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.

2.3 Performing the upgrade

Upon receiving the <next/> element for the upgrade, the server provides the elements and data needed for the client to calculate the requested data. The concrete elements and exchanges needed for the upgrade are specific to individual tasks. These tasks may be documented in other documents.

Listing 4: SASL mechanism upgrades to fictional BLOOP2 and BLOOP-42 mechanisms

```
<!-{}- The Client initiates the task requested by the server in the <
    continue/> element -{}->
<next xmlns='urn:xmpp:sasl:2' task='UPGR-BLOOP2'>
  <request xmlns='urn:xmpp:bloop2:example'>
    UGxlYXNlIHVwZ3JhZGUgbWUh
  </request>
</next>

<!-{}- The Server provides the needed data -{}->
<task-data xmlns='urn:xmpp:sasl:2'>
  <data xmlns='urn:xmpp:bloop2:example'>
    <data>U28sIG5leHQgRk9TREVNIC0gMjAxOCwgdGhhdCBpcy4uLg==</data>
    <description>BLOOP2 is cool!</description>
  </data>
</task-data>

<!-{}- The Client now responds -{}->
<task-data xmlns='urn:xmpp:sasl:2'>
  <upgrade xmlns='urn:xmpp:bloop2:example'>
    <response>
        Li4uSSdsbCBidXkgYSBiZWVyIGZvciB0aGUgZmlyc3QgcGVyc29uIHdoby4uLg
        ==</response>
    <finalize>2</finalize>
  </upgrade>
</task-data>

<!-{}- The Server requests the client to perform the second requested
    upgrade task for BLOOP-42 -{}->
<continue xmlns='urn:xmpp:sasl:2'>
  <additional-data>
    SSdtIGJvcmVkIG5vdy4=
  </additional-data>
  <tasks>
    <task>UPGR-BLOOP-42</task>
  </tasks>
  <text>This account requires an upgrade to BLOOP-42 as requested by
      the client</text>
</continue>

<!-{}- The Client initiates the task -{}->
```

```
<next xmlns='urn:xmpp:sasl:2' task='UPGR-BLOOP-42'>
  <request xmlns='urn:xmpp:bloop42:example'>
    UGxlYXNlIHVwZ3JhZGUgbWUh
  </request>
</next>

<!-{}- The Server provides the needed data -{}->
<task-data xmlns='urn:xmpp:sasl:2'>
  <data xmlns='urn:xmpp:bloop42:example'>
    <data>U28sIG5leHQgRk9TREVNIC0gMjAxOCwgdGhhdCBpcy4uLg==</data>
    <description>BLOOP-42 is cool!</description>
  </data>
</task-data>

<!-{}- The Client now responds -{}->
<task-data xmlns='urn:xmpp:sasl:2'>
  <upgrade xmlns='urn:xmpp:bloop42:example'>
    <response>
       Li4uSSdsbCBidXkgYSBiZWVyIGZvciB0aGUgZmlyc3QgcGVyc29uIHdoby4uLg
       ==</response>
    <finalize>42</finalize>
  </upgrade>
</task-data>

<!-{}- The upgrade was performed and the Server finishes
   authentication -{}->
<success xmlns='urn:xmpp:sasl:2'>
  <authorization-identifier>user@example.org</authorization-identifier
     >
</success>
```

3 SCRAM upgrade tasks

For upgrades of SCRAM mechanisms as defined in RFC 5802 [6], the server has to provide the needed data for the client to calculate the SaltedPassword as defined in this RFC (or some RFC updating it), namely the iteration count and salt. To do so the server sends a <salt/> element namespaced to "urn:xmpp:scram-upgrade:0" containing the salt and an attribute named "iteration" containing the iteration count as defined in that RFC, omitting the "s=" and "i=" prefix. The <salt/> element is contained within a <task-data/> wrapper element as defined in Extensible SASL Profile (XEP-0388) [7].
The client then calculates the SaltedPassword and sends back its base64 encoded value inside a <hash/> element namespaced to "urn:xmpp:scram-upgrade:0". The <hash/> element is contained within a <task-data/> wrapper element as defined in Extensible SASL Profile

[6] RFC 5802: Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms <http://tools.ietf.org/html/rfc5802>.
[7] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.

(XEP-0388) [8].
The name of the upgrade task MUST NOT conain the "-PLUS" suffix, because channel-binding is not relevant for upgrade tasks.

Listing 5: SCRAM hash upgrade task for SCRAM-SHA-256 after successful SCRAM-SHA-1-PLUS authentication

```
<!-{}- Client sending stream header -{}->
  <stream:stream
    from='user@example.org'
    to='example.org'
    version='1.0'
    xml:lang='en'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'>

  <!-{}- Server responding with stream header and features -{}->
  <stream:stream
    from='example.org'
    id='++TR84Sm6A3hnt3Q065SnAbbk3Y='
    to='user@example.org'
    version='1.0'
    xml:lang='en'
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'>
  <stream:features>
    <authentication xmlns='urn:xmpp:sasl:2'>
      <mechanism>SCRAM-SHA-1</mechanism>
      <mechanism>SCRAM-SHA-1-PLUS</mechanism>
      <inline>
        <!-{}- Server indicates that XEP-0198 stream resumption can be
            done "inline" -{}->
        <resume xmlns='urn:xmpp:sm:3'/>
        <!-{}- Server indicates support for XEP-0386 Bind 2 -{}->
        <bind xmlns='urn:xmpp:bind2:1'/>
      </inline>
      <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-SCRAM-SHA-256</
          upgrade>
      <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-SCRAM-SHA-512</
          upgrade>
    </authentication>
    <!-{}- Channel-binding information provided by XEP-0440 -{}->
    <sasl-channel-binding xmlns='urn:xmpp:sasl-cb:0'>
      <channel-binding type='tls-server-end-point'/>
      <channel-binding type='tls-exporter'/>
    </sasl-channel-binding>
  </stream:features>
```

[8] XEP-0388: Extensible SASL Profile <https://xmpp.org/extensions/xep-0388.html>.

```
  <!-{}- Client sends authentication request, requsting an upgrade to
      SCRAM-SHA-256 -{}->
  <authenticate xmlns='urn:xmpp:sasl:2' mechanism='SCRAM-SHA-1-PLUS'>
    <upgrade xmlns='urn:xmpp:sasl:upgrade:0'>UPGR-SCRAM-SHA-256</
        upgrade>
    <!-{}- Base64 of: 'p=tls-exporter,,n=user,r=12C4CD5C-E38E-4A98-8
        F6D-15C38F51CCC6' -{}->
    <initial-response>
        cD10bHMtZXhwb3J0ZXIsLG49dXNlcixyPTEyQzRDRDVDLUUzOEUtNEE5OC04RjZELTE1QzM4RjUxQ0
        ==</initial-response>
    <user-agent id='d4565fa7-4d72-4749-b3d3-740edbf87770'>
      <software>AwesomeXMPP</software>
      <device>Kiva's␣Phone</device>
␣␣␣␣</user-agent>
␣␣</authenticate>

␣␣<!-{}-␣Client␣authenticates␣using␣SCRAM-SHA-1-PLUS␣(or␣whatever␣
    mechanism␣was␣selected)-{}->
␣␣[...]

␣␣<!-{}-␣The␣Server␣provides␣the␣SCRAM␣upgrade␣task␣requested␣by␣the␣
    client␣in␣its␣<authenticate/>␣element␣-{}->
␣␣<continue␣xmlns='urn:xmpp:sasl:2'>
␣␣␣␣<!-{}-␣Base64␣of:␣'v=msVHs/BzIOHDqXeVH7EmmDu9id8='␣-{}->
␣␣␣␣<additional-data>
␣␣␣␣␣␣dj1tc1ZIcy9CeklPSERxWGVWSDdFbW1EdTlpZDg9
␣␣␣␣</additional-data>
␣␣␣␣<tasks>
␣␣␣␣␣␣<task>UPGR-SCRAM-SHA-256</task>
␣␣␣␣</tasks>
␣␣</continue>

␣␣<!-{}-␣The␣Client␣initiates␣the␣task␣-{}->
␣␣<next␣xmlns='urn:xmpp:sasl:2'␣task='UPGR-SCRAM-SHA-256'/>

␣␣<!-{}-
␣␣␣␣The␣server␣sends␣the␣required␣salt␣and␣iteration␣count.
␣␣-{}->
␣␣<task-data␣xmlns='urn:xmpp:sasl:2'>
␣␣␣␣<salt␣xmlns='urn:xmpp:scram-upgrade:0'␣iterations='4096'>
␣␣␣␣␣␣A_SXCRXQ6sek8bf_Z
␣␣␣␣</salt>
␣␣</task-data>

␣␣<!-{}-␣The␣client␣responds␣with␣the␣base64␣encoded␣SaltedPassword.␣
    -{}->
␣␣<task-data␣xmlns='urn:xmpp:sasl:2'>
␣␣␣␣<hash␣xmlns='urn:xmpp:scram-upgrade:0'>
```

```
      BzOnw3Pc5H4bOLlPZ/8JAy6wnTpH05aH21KW2+Xfpaw=
    </hash>
  </task-data>

  <!-{}-
    Finally, the server sends a success after adding the salted SHA
      -256 hash to its database.
    A SASL2 success always includes the authorization identifier.
  -{}->
  <success xmlns='urn:xmpp:sasl:2'>
    <authorization-identifier>user@example.org</authorization-
      identifier>
  </success>
```

4 Business Rules

For compatibility purposes, the server SHOULD keep the older authentication data (password hashes etc.) of all configured mechanisms to continue offering the same mechanisms as before.

5 Security Considerations

Clients SHOULD use channel-binding, if available, when requesting an upgrade to make sure no MITM can eavesdrop that hash and subsequently use it for authentication. Note that a client can always choose to not upgrade SASL mechanisms if it can not use channel-binding or the connection is otherwise deemed not secure enough.

6 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/>.

7 XMPP Registrar Considerations

This specification does not need any interaction with the XMPP Registrar [10].

8 XML Schema

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'
           targetNamespace="urn:xmpp:sasl:upgrade:0"
           xmlns="urn:xmpp:sasl:upgrade:0"
           elementFormDefault="qualified">

    <xs:element name="upgrade" type="SaslMechName" minOccurs="0"
        maxOccurs="unbounded"/>

    <xs:simpleType name="SaslMechName">
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="20"/>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'
           targetNamespace="urn:xmpp:scram-upgrade:0"
           xmlns="urn:xmpp:scram-upgrade:0"
           elementFormDefault="qualified">

  <xs:element name="parameters">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="salt" type="base64Data" minOccurs="1"
            maxOccurs="1"/>
        <xs:element name="iterations" type="iterationCount" minOccurs=
            "1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="hash" type="base64Data"/>
```

[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/>.

```
  <xs:simpleType name="base64Data">
      <xs:restriction base="xs:base64Binary"/>
  </xs:simpleType>

  <xs:simpleType name="iterationCount">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

9 Acknowledgements

Thanks to Matthew Wild and Dave Cridland for all of their valuable feedback and improvements.