XEP-0175: Best Practices for Use of SASL ANONYMOUS

Peter Saint-Andre
mailto:stpeter@stpeter.im
xmpp:stpeter@jabber.org
https://stpeter.im/

2009-09-30
Version 1.2

Status	Type	Short Name
Active	Informational	N/A

This document specifies best practices for use of the SASL ANONYMOUS mechanism in the context of client authentication with an XMPP server.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

XMPP Core [1] allows XMPP server implementations to support any SASL mechanism (see RFC 4422 [2]) when authenticating clients. This document provides recommendations regarding use of the SASL ANONYMOUS mechanism (see RFC 4505 [3]) in XMPP systems.

2 Deployment Types

XMPP server implementations can be deployed in a variety of settings. Although it is difficult to provide recommendations for every kind of XMPP deployment, this document attempts to strike a balance between more and less controlled settings by defining three different deployment types:

- Public deployments, such as well-known instant messaging (IM) services on the open Internet.
- Private deployments, such as enterprise IM services, technical support departments, and helplines.
- Specialized deployments that typically will be accessed in a controlled fashion, such as gaming services, members-only websites, and applications that are not used directly by human users.

3 Recommendations

An XMPP server implementation SHOULD NOT enable the SASL ANONYMOUS mechanism by default, but instead SHOULD force the administrator of a given service to explicitly enable support in the context of that deployment.
When a client authenticates using SASL ANONYMOUS, an XMPP server SHOULD assign a temporary, unique bare JID <localpart@domain.tld> to the client. Although the method for ensuring the uniqueness of the localpart is a matter of implementation, it is RECOMMENDED for the localpart to be a UUID as specified in RFC 4122 [4].
Although RFC 4505 [5] allows the client to provide so-called ”trace data” when authenticating via SASL ANONYMOUS, it is NOT RECOMMENDED for the client to include trace data as the XML character data of the <auth/> element (instead, the <auth/> element SHOULD be empty). However, if trace data is included, the server MUST NOT use it for any purpose other than tracing (e.g., not use it as the resource identifier of the anonymous user’s full JID).

[1] RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.
[2] RFC 4422: Simple Authentication and Security Layer (SASL) <http://tools.ietf.org/html/rfc4422>.
[3] RFC 4505: The SASL ANONYMOUS Mechanism <http://tools.ietf.org/html/rfc4505>.
[4] RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace <http://tools.ietf.org/html/rfc4122>.
[5] RFC 4505: The SASL ANONYMOUS Mechanism <http://tools.ietf.org/html/rfc4505>.

Because an anonymous user is unknown to the server, the server SHOULD appropriately restrict the user's access in order to limit the possibility of malicious behavior (such as denial of service attacks as described in Best Practices to Discourage Denial of Service Attacks (XEP-0205) [6]), especially on public deployments. The following restrictions are encouraged on public deployments. Administrators of private deployments and specialized deployments are advised to take these restrictions into account when configuring their services, but can reasonably relax these restrictions if they have appropriate access controls in place or their deployment requirements cannot be met using the more restrictive profile applied in public deployments.

1. During resource binding, the server MAY ignore the resource identifier provided by the client (if any) and instead assign a resource identifier that it generates on behalf of the client.

2. The server SHOULD NOT allow the client to initiate communication with entities hosted at remote servers.

3. The server MAY allow the client to establish relationships with local services and users; such relationships might include presence subscriptions and roster additions (see XMPP IM [7]), Multi-User Chat (XEP-0045) [8] registrations, and Publish-Subscribe (XEP-0060) [9] subscriptions. (Note that allowing presence subscriptions and roster additions can create a sub-optimal user experience for the added contacts.) However, if the server permits such relationships, it MUST cancel them when the client's session ends.

4. The server MAY allow the client to store information on the server for the purpsoe of providing an optimal user experience (e.g., storage of client preferences using Private XML Storage (XEP-0049) [10]). However, if the server allows this, it SHOULD remove such information when the client's session ends.

5. The server SHOULD NOT allow the client to send large numbers of XMPP stanzas or otherwise use large amounts of system resources (e.g., by binding multiple resource identifiers or creating multiple SOCKS5 Bytestreams (XEP-0065) [11] sessions).

[6] XEP-0205: Best Practices to Discourage Denial of Service Attacks <https://xmpp.org/extensions/xep-0205.html>.
[7] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.
[8] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[9] XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>.
[10] XEP-0049: Private XML Storage <https://xmpp.org/extensions/xep-0049.html>.
[11] XEP-0065: SOCKS5 Bytestreams <https://xmpp.org/extensions/xep-0065.html>.

4 Protocol Flow

The RECOMMENDED protocol flow following TLS negotiation (refer to RFC 3920 [12]) is as follows:

1. Client initiates stream to server.

Listing 1: Stream initiation

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

2. Server replies with stream header.

Listing 2: Stream header reply

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

3. Server advertises stream features.

Listing 3: Stream features advertisement

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

4. Client requests SASL ANONYMOUS mechanism.

Listing 4: Requesting SASL ANONYMOUS

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

[12] RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc3920>.

5. Server sends <success/>.

Listing 5: Sending success

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

6. Client opens new stream.

Listing 6: Initiating a new stream

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

7. Server tells client that resource binding is required.

Listing 7: Stream header reply with features

```
<stream:stream
        xmlns:stream='http://etherx.jabber.org/streams'
        xmlns='jabber:client'
        id='c2s_345'
        from='example.com'
        version='1.0'>
<stream:features>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'/>
</stream:features>
```

8. Client requests that server create a resource for it.

Listing 8: Requesting resource creation

```
<iq type='set' id='bind_1'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'/>
</iq>
```

9. Server replies with full JID.

Listing 9: Server informs client of full JID

```
<iq type='result' id='bind_1'>
  <bind xmlns='urn:ietf:params:xml:ns:xmpp-bind'>
    <jid>59BEC12A-9BAB-452B-88F8-D1563F09E549@example.com/2384
        F02A7E01</jid>
```

```
    </bind>
</iq>
```

5 Service Discovery

A server MUST reply to a Service Discovery (XEP-0030) [13] information ("disco#info") request sent to the bare JID <localpart@domain.tld> of the user with an identity of "account/anonymous", as shown in the following example.

Listing 10: Server informs client of full JID

```
<iq from='59BEC12A-9BAB-452B-88F8-D1563F09E549@example.com'
    id='kj37vd95'
    to='requester@example.com/foo'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity category='account' type='anonymous'/>
    <feature var='http://jabber.org/protocol/disco#info'/>
    <feature var='http://jabber.org/protocol/disco#items'/>
  </query>
</iq>
```

6 Security Considerations

The security considerations discussed in RFC 3920 [14] and RFC 4505 apply to the use of SASL ANONYMOUS in XMPP; specific suggestions regarding usage restrictions for anonymous users are provided under the Recommendations section of this document.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [15].

[13] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[14] RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc3920>.
[15] 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/>.

8 XMPP Registrar Considerations

This document requires no interaction with the XMPP Registrar [16].

9 Acknowledgements

Thanks to Dave Cridland, Tuomas Koski, Jack Moffitt, Andy Skelton, and Kurt Zeilenga for their feedback.

[16] 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/>.