XEP-0236: Abuse Reporting

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

2008-05-09
Version 0.2

Status	Type	Short Name
Retracted	Standards Track	NOT YET ASSIGNED

This specification defines an XMPP protocol extension for reporting abusive traffic sent over an XMPP network. Note: This specification has been retracted in favor of XEP-0161, which now contains the content originally published in this specification.

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

Unfortunately, not all XMPP entities are well-behaved. Currently, if an XMPP entity (the "attacker") sends abusive stanzas to another XMPP entity (the "victim"), there is no way for the victim or the victim's server to inform the attacker's server that the attacker is generating abusive traffic. In practice, the victim's server may need to terminate the server-to-server connection (currently without explicitly informing the attacker's server about the reason for the termination) rather than continue to accept the abusive traffic.

This situation is far from desirable. Therefore, this specification defines three small XMPP protocol functions that can help to improve the reliability of server-to-server connections:

1. A method by which the receiving server can send an abuse report to the sending server, including the JID(s) of the sender(s).

2. An application-specific stanza error condition that can be combined with the standard <not-acceptable/> stanza error condition to inform the sending server that a particular XMPP stanza is considered abusive.

3. An application-specific stream error condition that can be combined with the standard <policy-violation/> stream error condition to inform the sending server about the reason for termination of an XML stream.

2 Syntax

An abuse report MUST be sent in an IQ stanza of type "set" containing an <abuse/> element qualified by the 'urn:xmpp:tmp:abuse' namespace (see Protocol Namespaces regarding issuance of one or more permanent namespaces). The allowable children of the <abuse/> element are:

- One or more <jid/> elements whose XML character data specifies the JID(s) of the abusive sender(s)

- An optional <reason/> element that specifies the reason for the abuse report, via a machine-readable abuse condition defined in this specification, (optionally) human-readable text about the report, and (optionally) an application-specific condition defined outside this specification.

This specification intentionally does not define exactly what constitutes abuse, since "abuse is in the eye of the beholder". However, the following machine-readable conditions are defined as children of the <reason/> element.

1

Condition	Definition
<gateway/>	Attempting to inappropriately use a gateway on the receiving server (see Gateway Interaction (XEP-0100) XEP-0100: Gateway Interaction <https://xmpp.org/extensions/xep-0100.html>.)
<muc/>	Attempting to take over or otherwise abuse Multi-User Chat (XEP-0045) XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>. rooms on the receiving server
<proxy/>	Attempting to inappropriately use a SOCKS5 Bytestreams (XEP-0065) XEP-0065: SOCKS5 Bytestreams <https://xmpp.org/extensions/xep-0065.html>. proxy, TURN server, or other proxy on the receiving server
<pubsub/>	Attempting to inappropriately use a Publish-Subscribe (XEP-0060) XEP-0060: Publish-Subscribe <https://xmpp.org/extensions/xep-0060.html>. service on the receiving server
<service/>	Attempting to inappropriately use any other kind of service on the receiving server
<spam/>	Sending spam (unsolicited bulk messages)
<stanza-too-big/>	Sending extremely large stanzas
<too-many-recipients/>	Sending messages that contain too many recipients (see Extended Stanza Addressing (XEP-0033) XEP-0033: Extended Stanza Addressing <https://xmpp.org/extensions/xep-0033.html>.)
<too-many-stanzas/>	Sending an extremely large number of stanzas
<unacceptable-payload/>	Sending messages that contain unacceptable payloads such as malicious executables
<unacceptable-text/>	Sending messages that contain unacceptable human-readable text
<undefined-abuse/>	The abuse condition is undefined (should be used with an application-specific condition)

Note: The foregoing list of conditions is not exhaustive. The list may be augmented or otherwise modified in a future version of this specification as a result of implementation and deployment experience.

3 Abuse Report

3.1 Generation

If an XMPP server receives abusive stanzas over a server-to-server connection, the receiving server SHOULD send an abuse report to the sending server.

Listing 1: Abuse Report

```
<iq from='example.org'
    id='rep1'
    to='example.com'
    type='set'>
  <abuse xmlns='urn:xmpp:tmp:abuse'>
    <jid>abuser@example.com/foo</jid>
    <reason>
      <condition>
        <muc/>
      </condition>
    </reason>
  </abuse>
</iq>
```

3.2 Processing

Upon receiving the abuse report, the sending server MUST proceed as follows.

3.2.1 Abuse Reporting Not Supported

If the sending server does not understand the abuse reporting protocol, it MUST return a <service-unavailable/> error to the receiving server.

Listing 2: Abuse reporting not supported

```
<iq from='example.com'
    id='rep1'
    to='example.org'
    type='error'>
  <error type='cancel'>
    <service-unavailable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

3.2.2 Sender(s) Not Found

If none of the JIDs contained in the abuse report exist at the sending server, the sending server MUST return an <item-not-found/> error to the receiving server.

Listing 3: Senders not found

```
<iq from='example.com'
    id='rep1'
    to='example.org'
```

```
    type='error'>
  <error type='cancel'>
    <item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
</iq>
```

3.2.3 Abuse Report Accepted

If the sending server accepts the abuse report for one or more JIDs, it MUST return an IQ stanza of type "result" to the receiving server.

Listing 4: Abuse report accepted

```
<iq from='example.com'
    id='rep1'
    to='example.org'
    type='result'/>
```

This specification does not define how a sending server shall behave when it receives an abuse report. In general it is expected that the sending server (1) will notify the human administrators of the server in some implementation-specific or deployment-specific fashion, and (2) may use the abuse report in an automated fashion (e.g., as input to a rate-limiting algorithm, reputation system, or decision about temporarily suspending the privileges of the sending entity or entities). In addition, the sending server MAY the report to trusted parties such as third-party reporting services.

4 Stanza Error

The receiving server MAY report that a particular stanza is considered abusive. The stanza error condition MUST be <not-acceptable/> and the error stanza MUST include an application-specific error condition of <abuse/> qualified by the 'urn:xmpp:tmp:abuse' (see Protocol Namespaces regarding issuance of one or more permanent namespaces). The <abuse/> element MUST include one or more <jid/> elements whose XML character data specifies the JID(s) of the abusive sender(s).

Listing 5: Abusive stanza

```
<message from='abuser@example.org/foo'
         to='victim@example.org'>
  [ ... some abusive payload here ... ]
</message>
```

Listing 6: Stanza error

```xml
<message from='example.com'
         to='example.org'>
  <error type='cancel'>
    <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  </error>
  <abuse xmlns='urn:xmpp:tmp:abuse'>
    <jid>abuser@example.com/foo</jid>
    <reason>
      <condition>
        <unacceptable-payload/>
      </condition>
    </reason>
  </abuse>
</message>
```

5 Stream Error

If the sending entity continues to generate abusive stanzas via the sending server, the receiving server MAY close the stream between the receiving server and the sending server. The stream error condition MUST be <policy-violation/> and the stream error MUST include an application-specific error condition of <abuse/> qualified by the 'urn:xmpp:tmp:abuse'. The <abuse/> element MUST include one or more <jid/> elements whose XML character data specifies the JID(s) of the abusive sender(s).

Listing 7: Stream Error

```xml
<stream:error>
  <policy-violation xmlns='urn:ietf:params:xml:ns:xmpp-streams'/>
  <abuse xmlns='urn:xmpp:tmp:abuse'>
    <jid>abuser@example.com/foo</jid>
    <reason>
      <condition>
        <too-many-stanzas/>
      </condition>
    </reason>
  </abuse>
</stream:error>
</stream:stream>
```

The receiving entity then SHOULD terminate the TCP connection between the receiving server and the sending server.

6 Discovering Support

If a server supports the abuse reporting protocol, it MUST report that fact by including a service discovery feature of "urn:xmpp:tmp:abuse" (see Protocol Namespaces regarding issuance of one or more permanent namespaces) in response to a Service Discovery (XEP-0030) [1] information request:

Listing 8: Service Discovery information request

```
<iq from='example.org'
    id='disco1'
    to='example.com'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'/>
</iq>
```

Listing 9: Service Discovery information response

```
<iq from='example.com'
    id='disco1'
    to='example.org'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    ...
    <feature var='urn:xmpp:tmp:abuse'/>
    ...
  </query>
</iq>
```

7 Security Considerations

7.1 Denial of Service Attacks

It is possible for an abusive sender to launch a denial of service attack against legitimate users of the sending server by generating abusive traffic over the server-to-server connection (in fact such attacks have already been observed on XMPP networks). Although use of the abuse reporting protocol does not completely prevent such attacks, it may at least enable sending servers to react to abusive traffic in close to real time, thus helping to "heal" the network when denial of service attacks are launched.

7.2 Man in the Middle Attacks

If a malicious entity can inject information into the server-to-server connection, it can falsely send abuse reports to the sending server. Therefore the connection SHOULD be encrypted

---

[1]XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.

using Transport Layer Security as specified in XMPP Core [2].

8 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA) [3].

9 XMPP Registrar Considerations

9.1 Protocol Namespaces

Until this specification advances to a status of Draft, its associated namespace shall be "urn:xmpp:tmp:abuse"; upon advancement of this specification, the XMPP Registrar [4] shall issue a permanent namespace in accordance with the process defined in Section 4 of XMPP Registrar Function (XEP-0053) [5].

9.2 Application-Specific Errors

The XMPP Registrar shall add <abuse/> to its registry of application-specific error conditions (see <https://xmpp.org/registrar/errors.html>), where the element is qualified by the 'urn:xmpp:tmp:abuse' namespace (see Protocol Namespaces regarding issuance of one or more permanent namespaces).

The registry submission is as follows:

```
<condition>
  <ns>urn:xmpp:tmp:abuse</ns>
  <element>abuse</element>
  <desc>the sending entity has generated traffic that the receiving
      server considers abusive</desc>
  <doc>XEP-xxxx</doc>
</condition>
```

---

[2]RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <http://tools.ietf.org/html/rfc6120>.

[3]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/>.

[4]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/>.

[5]XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

10 XML Schema

```xml
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:tmp:abuse'
    xmlns='urn:xmpp:tmp:abuse'
    elementFormDefault='qualified'>

  <xs:element name='abuse'>
    <xs:complexType>
      <xs:sequence>
        <xs:element name='jid' type='xs:string' minOccurs='1'
            maxOccurs='unbounded'/>
        <xs:element ref='reason' minOccurs='0'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='reason'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='condition' minOccurs='0' maxOccurs='1'/>
        <xs:element name='text' type='xs:string' minOccurs='0'
            maxOccurs='1'/>
        <xs:any namespace='##other' minOccurs='0' maxOccurs='1'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='condition'>
    <xs:complexType>
      <xs:choice>
        <xs:element name='gateway' type='empty'/>
        <xs:element name='muc' type='empty'/>
        <xs:element name='proxy' type='empty'/>
        <xs:element name='pubsub' type='empty'/>
        <xs:element name='service' type='empty'/>
        <xs:element name='spam' type='empty'/>
        <xs:element name='stanza-too-big' type='empty'/>
        <xs:element name='too-many-recipients' type='empty'/>
        <xs:element name='too-many-stanzas' type='empty'/>
        <xs:element name='unacceptable-payload' type='empty'/>
        <xs:element name='unacceptable-text' type='empty'/>
        <xs:element name='undefined-abuse' type='empty'/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
```

```
  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```