XEP-0101: HTTP Authentication using Jabber Tickets

Richard Dobson
mailto:richard@dobson-i.net
xmpp:richard@dobson-i.net

2004-01-18
Version 0.2

Status	Type	Short Name
Deferred	Standards Track	Not yet assigned

This document defines a protocol for authenticating HTTP requests using Jabber Tickets.

Legal

Copyright

Permissions

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 <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

Contents

1 Introduction

Jabber Ticket Authentication is a method of authenticating with HTTP servers using your jabber identification.
This allows you to login to websites using your jabber address in a single sign-on fashion similar to .NET Passport, but unlike .NET Passport is not locked into a single authentication provider.
Tickets also mean the jabber ticket provider and the web server do not need to be tightly integrated for authentication to work, also because its not tightly integrated it means webmasters do not need to setup their own jabber server to provide tickets, they can use a third party provider even a central ”tickets.jabber.org”. Also because tickets are not tightly integrated it makes it far easier for webmasters to integrate with Jabber, it also makes web farms far more scalable and reliable.

2 Requirements

The motivations for this document are:

- To provide a method of using a jabber connections authenticated stream to provide a method of authenticating with an HTTP server.
- To provide this authentication without needing the jabber ticket component and the webserver to be tightly coupled, this is essential in a web farm environment for scalability.
- To make the communication between the jabber client and the server(s) as simple as possible.

3 Use Cases

3.1 Client web browser window requests a Jabber Ticket Authentication protected web page

Listing 1: Request for page

```
GET http://www.webserver.com/webpage.html HTTP/1.1
```

Listing 2: The server responds with a 401 and WWW-Authenticate header

```
401 Unauthorised HTTP/1.1
WWW-Authenticate: JabberTicket realm="ticket.server.com"
```

The realm is the JID you need to request your JabberTicket from.

3.2 Client requests JabberTicket

Listing 3: Request for ticket

```
<iq
    to=’ticket.server.com’
    type=’get’
    id=’ticket1’>
  <query xmlns=”http://jabber.org/protocol/ticket”/>
</iq>
```

Listing 4: Server responds with jabber ticket

```
<iq
    to=’user@domain.com/resource’
    from=’ticket.server.com’
    type=’result’
    id=’ticket1’>
  <query xmlns=”http://jabber.org/protocol/ticket”>
    54yudvjhssa76dta6sgdst78r4sadsfjdhs...
  </query>
</iq>
```

The ticket is encrypted data represented as a string, the client does not need to decode it since it is passed to the webserver unaltered.

3.3 Client replies to 401 HTTP error

Listing 5: Client HTTP request

```
GET http://www.webserver.com/webpage.html HTTP/1.1
Authorization: JabberTicket 54yudvjhssa76dta6sgdst78r4sadsfjdhs...
```

3.4 Server responds and allows or denies access to the file

Listing 6: Server allows access

```
200 OK HTTP/1.1
Content-Type: text/html
```

Listing 7: Server denies access

```
403 Forbidden HTTP/1.1
```

4 Implementation Notes

The following guidelines may assist developers.

- The ticket can be encrypted however the provider likes since only they will need to understand it.
- The ticket must somewhere contain in it the JID of the end user (or some method of knowing who the user is), so that the webserver knows who it is.
- It is recommended that your tickets also use an extra level of authentication such as ensuring the User-Agent is the same across requests, that the ip address is the same across requests.

5 Security Considerations

5.1 Man in the middle

This form of HTTP authentication is susceptable to man in the middle attack where the ticket could be captured and retransmitted by someone else, but this can be solved by using an encrypted jabber connection (e.g. HTTPS) and an HTTPS connection to the webserver.

5.2 Key length

It is recommended the encryption key length for the ticket be long enough to make it hard to crack the ticket.

5.3 Ticket expiration

It is recommended the ticket has an expiration and that it be between a few minutes and a few hours, e.g. 60 minutes.

6 IANA Considerations

The HTTP authentication scheme ”JabberTicket” may need to be registered with IANA.

7 XMPP Registrar Considerations

The XMPP Registrar [1] will need to register the new namespace of ”http://jabber.org/protocol/ticket”.

[1] 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 <http://xmpp.org/registrar/>.