



# XMPP

## XEP-0035: SSL/TLS Integration

Robert Norris

<mailto:rob@cataclysm.cx>

<xmpp:rob@cataclysm.cx>

2003-11-05

Version 0.2

| Status    | Type            | Short Name |
|-----------|-----------------|------------|
| Retracted | Standards Track | N/A        |

NOTE WELL: this specification was retracted on 2003-11-05 since the topic is addressed definitively in XMPP Core. Please refer to XMPP Core for further information.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 - 2011 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 <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                     | 1 |
| 2   | Protocol                         | 1 |
| 2.1 | Overview . . . . .               | 1 |
| 2.2 | Stream initialization . . . . .  | 1 |
| 2.3 | TLS negotiation . . . . .        | 2 |
| 3   | Certificate-based authentication | 3 |

## 1 Introduction

The TLS protocol <sup>1</sup> (formerly known as SSL) provides a way to secure an application protocol from tampering and eavesdropping. The option of using such security is desirable for Jabber due to common connection eavesdropping and hijacking attacks. <sup>23</sup>

Traditionally, Jabber servers has supported TLS by utilising a "wrapper" around the standard protocol stream. This wrapper usually listens on a port other than those listed in the IANA registry <sup>4</sup> (commonly 5223 for client-to-server communications and 5270 for server-to-server communications). In the case of client-to-server communications, clients must initiate a TLS session immediately after connecting, before beginning the normal XML stream. This method of utilising TLS is typical of many servers that implement stream-based protocols, but has a number of flaws, which are outlined in section 7 of RFC 2595. Accordingly, the use of port 5223 and port 5270 for secure sessions is deprecated.

This document describes an extension to the Jabber XML stream that provides a "STARTTLS" command which clients may invoke on an insecure stream to secure it. This extension is modelled on RFC 2595, which describes the STARTTLS extension for the IMAP <sup>5</sup>, POP3 <sup>6</sup> and ACAP <sup>7</sup> protocols. A future document (or an addition to this document) will define TLS support within server-to-server streams.

## 2 Protocol

### 2.1 Overview

This protocol operates over the standard Jabber client connection on port 5222.

The namespace identifier for this protocol ipath The following examples show the dialogue between a client [C] and a server [S].

### 2.2 Stream initialization

The client begins by requesting the use of STARTTLS as part of the normal Jabber stream negotiation. The server responds by informing the client whether or not it supports STARTTLS. It does this in the normal stream negotiation response:

---

<sup>1</sup>RFC 2246

<sup>2</sup>See RFC 1704, "On Internet Authentication."

<sup>3</sup>This paragraph adapted from RFC 2595, section 1.

<sup>4</sup>The Internet Assigned Numbers Authority defines jabber-client (port 5222) and jabber-server (port 5269) as the standard Jabber ports; see <http://www.iana.org/assignments/port-numbers>.

<sup>5</sup>RFC 2060

<sup>6</sup>RFC 1939

<sup>7</sup>RFC 2244

Listing 1: Stream initialization

```
C: <stream:stream xmlns='jabber:client'
      xmlns:stream='http://etherx.jabber.org/streams'
      xmlns:tls='http://www.ietf.org/rfc/rfc2595.txt'
      to='jabber.org'>
S: <stream:stream xmlns='jabber:client'
      xmlns:stream='http://etherx.jabber.org/streams'
      xmlns:tls='http://www.ietf.org/rfc/rfc2595.txt'
      id='12345678'>
```

In the event that a server does not support the STARTTLS extension, it will respond with the normal stream negotiation response:

Listing 2: Stream initialization; server not supporting STARTTLS

```
C: <stream:stream xmlns='jabber:client'
      xmlns:stream='http://etherx.jabber.org/streams'
      xmlns:tls='http://www.ietf.org/rfc/rfc2595.txt'
      to='jabber.org'>
S: <stream:stream xmlns='jabber:client'
      xmlns:stream='http://etherx.jabber.org/streams'
      id='12345678'>
```

### 2.3 TLS negotiation

To begin the TLS negotiation, the client issues the STARTTLS command:

Listing 3: STARTTLS request

```
C: <tls:starttls/>
```

When the server is ready to begin the TLS negotiation, it will close the XML stream, but will keep the underlying connection to the client open:

Listing 4: STARTTLS response

```
S: </stream:stream>
```

The client now begins a normal TLS negotiation by sending the TLS ClientHello command. Upon completion of the TLS negotiation, the client reissues the XML stream initialization:

Listing 5: Stream initialization

```
C: <stream:stream xmlns='jabber:client'
      xmlns:stream='http://etherx.jabber.org/streams'
      to='jabber.org'>
S: <stream:stream xmlns='jabber:client'
```

```
xmlns:stream='http://etherx.jabber.org/streams'  
id='12345678'>
```

This is necessary, since any information about the stream presented by the server or the client may have been modified by an attacker.

Note that once the secure channel has been established, the server must not advertise or allow the use of the STARTTLS command.

### 3 Certificate-based authentication

TLS allows clients to be authenticated by verifying the certificate that they present during the TLS negotiation. This can be done in conjunction with the Jabber SASL profile (see [SASL Integration](#)<sup>8</sup>) and the EXTERNAL mechanism.

If a client authenticates with a certificate using the TLS authentication, and the client requests the use of SASL in the second XML stream negotiation (over the secure channel), servers supporting certificate-based authentication should add the EXTERNAL mechanism to the list of supported authentication mechanisms. If the client then requests this mechanism, the server should automatically inform the user that authentication was successful. See [RFC 2222](#)<sup>9</sup> and XEP-0034 for more information.

Servers implementing STARTTLS functionality are not required to implement certificate-based authentication.

---

<sup>8</sup>XEP-0034: SASL Integration <<http://xmpp.org/extensions/xep-0034.html>>.

<sup>9</sup>RFC 2222: Simple Authentication and Security Layer (SASL) <<http://tools.ietf.org/html/rfc2222>>.