
Abstract


Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7590.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Introduction

The Extensible Messaging and Presence Protocol (XMPP) [RFC6120] (along with its precursor, the so-called "Jabber protocol") has used Transport Layer Security (TLS) [RFC5246] (along with its precursor, Secure Sockets Layer or SSL) since 1999. Both [RFC6120] and its predecessor [RFC3920] provided recommendations regarding the use of TLS in XMPP. In order to address the evolving threat model on the Internet today, this document provides stronger recommendations.

In particular, this document updates [RFC6120] by specifying that XMPP implementations and deployments MUST follow the best current practices documented in the "Recommendations for Secure Use of TLS and DTLS" [RFC7525]. This includes stronger recommendations regarding SSL/TLS protocol versions, fallback to lower versions, TLS-layer compression, TLS session resumption, cipher suites, public key lengths, forward secrecy, and other aspects of using TLS with XMPP.

2. Terminology

Various security-related terms are to be understood in the sense defined in [RFC4949].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
3. Recommendations

The best current practices documented in the "Recommendations for Secure Use of TLS and DTLS" [RFC7525] are included here by reference. Instead of repeating those recommendations here, this document mostly provides supplementary information regarding secure implementation and deployment of XMPP technologies.

3.1. Support for TLS

Support for TLS (specifically, the XMPP profile of STARTTLS) is mandatory for XMPP implementations, as already specified in [RFC6120] and its predecessor [RFC3920].

The server (i.e., the XMPP receiving entity) to which a client or peer server (i.e., the XMPP initiating entity) connects might not offer a stream feature of <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>. Although in general this stream feature indicates that the server supports and offers TLS, this stream feature might be stripped out by an attacker (see Section 2.1 of [RFC7457]). Similarly, the <required/> child element of the <starttls/> stream feature is used to indicate that negotiation of TLS is mandatory; however, this could also be stripped out by an attacker. Therefore, the initiating entity MUST NOT be deterred from attempting TLS negotiation even if the receiving entity does not advertise support for TLS. Instead, the initiating entity SHOULD (based on local policy) proceed with the stream negotiation and attempt to negotiate TLS.

3.2. Compression

XMPP supports an application-layer compression technology [XEP-0138]. Although this XMPP extension might have slightly stronger security properties than TLS-layer compression (since it is enabled after Simple Authentication and Security Layer (SASL) authentication, as described in [XEP-0170]), this document neither encourages nor discourages use of XMPP-layer compression.

3.3. Session Resumption

To improve the reliability of communications over XMPP, it is common practice for clients and servers to implement the stream management extension [XEP-0198]. Although that specification includes a method for resumption of XMPP streams at the application layer, also using session resumption at the TLS layer further optimizes the overall process of resuming an XMPP session (see [XEP-0198] for detailed information). Whether or not XEP-0198 is used for application-layer
session resumption, implementations MUST follow the recommendations provided in [RFC7525] regarding TLS-layer session resumption.

3.4. Authenticated Connections

Both the core XMPP specification [RFC6120] and the CertID specification [RFC6125] provide recommendations and requirements for certificate validation in the context of authenticated connections. This document does not supersede those specifications (e.g., it does not modify the recommendations in [RFC6120] regarding the Subject Alternative Names or other certificate details that need to be supported for authentication of XMPP connections using PKIX certificates).

Wherever possible, it is best to prefer authenticated connections (along with SASL [RFC4422]), as already stated in the core XMPP specification [RFC6120]. In particular:

- Clients MUST authenticate servers.
- Servers MUST authenticate clients.
- Servers SHOULD authenticate other servers.

This document does not mandate that servers need to authenticate peer servers, although such authentication is strongly preferred. Unfortunately, in multi-tenant environments it can be extremely difficult to obtain and deploy PKIX certificates with the proper Subject Alternative Names (see [XMPP-DNA] and [PKIX-POSH] for details). To overcome that difficulty, the Domain Name Associations (DNAs) specification [XMPP-DNA] describes a framework for XMPP server authentication methods, which include not only PKIX but also DNS-Based Authentication of Named Entities (DANE) as defined in [DANE-SRV] and PKIX over Secure HTTP (POSH) as defined in [PKIX-POSH]. These methods can provide a basis for server identity verification when appropriate PKIX certificates cannot be obtained and deployed.

Given the pervasiveness of eavesdropping [RFC7258], even an encrypted but unauthenticated connection might be better than an unencrypted connection in these scenarios (this is similar to the "better-than-nothing security" approach for IPsec [RFC5386]). Encrypted but unauthenticated connections include connections negotiated using anonymous Diffie-Hellman mechanisms or using self-signed certificates, among others. In particular for XMPP server-to-server interactions, it can be reasonable for XMPP server implementations to accept encrypted but unauthenticated connections when Server Dialback keys [XEP-0220] are used; such keys on their own provide only weak
identity verification (made stronger through the use of DNSSEC [RFC4033]), but this at least enables encryption of server-to-server connections. The DNA prooftypes mentioned above are intended to mitigate the residual need for encrypted but unauthenticated connections in these scenarios.

3.5. Server Name Indication

Although there is no harm in supporting the TLS Server Name Indication (SNI) extension [RFC6066], this is not necessary since the same function is served in XMPP by the ‘to’ address of the initial stream header as explained in Section 4.7.2 of [RFC6120].

3.6. Human Factors

It is strongly encouraged that XMPP clients provide ways for end users (and that XMPP servers provide ways for administrators) to complete the following tasks:

- Determine if a given incoming or outgoing XML stream is encrypted using TLS.
- Determine the version of TLS used for encryption of a given stream.
- If authenticated encryption is used, determine how the connection was authenticated or verified (e.g., via PKI, DANE, POSH, or Server Dialback).
- Inspect the certificate offered by an XMPP server.
- Determine the cipher suite used to encrypt a connection.
- Be warned if the certificate changes for a given server.

4. Security Considerations

The use of TLS can help to limit the information available for correlation between the XMPP application layer and the underlying network and transport layers. As typically deployed, XMPP technologies do not leave application-layer routing data (such as XMPP ‘to’ and ‘from’ addresses) at rest on intermediate systems, since there is only one hop between any two given XMPP servers. As a result, encrypting all hops (sender’s client to sender’s server, sender’s server to recipient’s server, and recipient’s server to recipient’s client) can help to limit the amount of metadata that might leak.
It is possible that XMPP servers themselves might be compromised. In that case, per-hop encryption would not protect XMPP communications, and even end-to-end encryption of (parts of) XMPP stanza payloads would leave addressing information and XMPP roster data in the clear. By the same token, it is possible that XMPP clients (or the end-user devices on which such clients are installed) could also be compromised, leaving users utterly at the mercy of an adversary.

This document and related actions to strengthen the security of the XMPP network are based on the assumption that XMPP servers and clients have not been subject to widespread compromise. If this assumption is valid, then ubiquitous use of per-hop TLS channel encryption and more significant deployment of end-to-end object encryption technologies will serve to protect XMPP communications to a measurable degree, compared to the alternatives.

This document covers only communication over the XMPP network and does not take into account gateways to non-XMPP networks. As an example, for security considerations related to gateways between XMPP and the Session Initiation Protocol (SIP), see [RFC7247] and [RFC7572].

5. References

5.1. Normative References


5.2. Informative References


Appendix A. Implementation Notes

Some governments enforce legislation prohibiting the export of strong cryptographic technologies. Nothing in this document ought to be taken as advice to violate such prohibitions.

Acknowledgements

The authors would like to thank the following individuals for their input: Dave Cridland, Philipp Hancke, Olle Johansson, Steve Kille, Tobias Markmann, Matt Miller, and Rene Treffer.

Roni Even caught several important issues in his review on behalf of the General Area Review Team.

Ben Campbell, Spencer Dawkins, and Barry Leiba provided helpful input during IESG review.

Thanks to Leif Johansson and Orit Levin as chairs of the UTA WG, Ben Campbell and Joe Hildebrand as chairs of the XMPP WG, and Stephen Farrell as the sponsoring Area Director.

Authors’ Addresses

Peter Saint-Andre
&yet

EMail: peter@andyet.com
URI: https://andyet.com/

Thijs Alkemade

EMail: me@thijsalkema.de