



XMPP

XEP-0438: Best practices for password hashing and storage

Sam Whited

<mailto:sam@samwhited.com>

<xmpp:sam@samwhited.com>

<https://blog.samwhited.com/>

2020-05-05

Version 0.1.1

Status	Type	Short Name
Experimental	Informational	passwords

This document outlines best practices for handling user passwords on the public Jabber network for both clients and servers.

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	1
3	Glossary	1
4	SASL Mechanisms	2
5	Client Best Practices	3
5.1	Mechanism Pinning	3
5.2	Storage	3
6	Server Best Practices	3
6.1	Additional SASL Requirements	3
6.2	Storage	4
6.3	Authentication and Rotation	4
7	PBKDF2 Parameters	4
8	Password Complexity Requirements	5
9	Security Considerations	5
10	IANA Considerations	6
11	XMPP Registrar Considerations	6

1 Introduction

Following best practices when hashing and storing passwords and other authenticator secrets impacts a great deal more than just a users identity. It also effects usability, and backwards compatibility by determining what authentication and authorization mechanisms can be used. Unfortunately, aside from mandating the use of SCRAM-SHA-1 in [RFC 6120](#)¹, and recommending at least 4096 rounds of PBKDF2 in [RFC 5802](#)² (a number which is now woefully inadequate), no general recommendations for best practices in password storage, transmission, or key derivation function tuning exist in the XMPP ecosystem.

Many of the recommendations in this document were taken from [Digital Identity Guidelines: Authentication and Lifecycle Management](#)³ and [Recommendation for Password-Based Key Derivation, Part 1: Storage Applications](#)⁴.

2 Requirements

This document makes specific recommendations for best practices on the public Jabber network for both clients and servers. It does not attempt to address private networks or proprietary services which may have different requirements, use cases, and security models. These recommendations include the hashing and storage of memorized secrets and other authenticators, authentication, and compatibility between clients and servers with respect to authentication.

To keep the length of this document manageable, we assume basic familiarity with password storage and handling, common terms, and cryptographic operations. For an overview of basic password security see the [OWASP Password Storage Cheat Sheet](#)⁵ maintained by the [Open Web Application Security Project \(OWASP\)](#)⁶.

3 Glossary

Many terms used in this document are defined in [Digital Identity Guidelines](#)⁷ Appendix A.1 and in [Recommendation for Password-Based Key Derivation, Part 1: Storage Applications](#)

¹RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

²RFC 5802: Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms <<http://tools.ietf.org/html/rfc5802>>.

³Digital Identity Guidelines: Authentication and Lifecycle Management, NIST Special Publication 800-63B <<https://doi.org/10.6028/NIST.SP.800-63b>>.

⁴Recommendation for Password-Based Key Derivation, Part 1: Storage Applications, NIST Special Publication 800-132 <<https://doi.org/10.6028/NIST.SP.800-132>>.

⁵OWASP Cheat Sheet Series for password storage <https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html>.

⁶The Open Web Application Security Project (OWASP, or OWASP Foundation) is a nonprofit foundation that works to improve the security of software. For further information, see <<https://owasp.org/>>.

⁷Digital Identity Guidelines, NIST Special Publication 800-63-3 <<https://doi.org/10.6028/NIST.SP.800-63-3>>.

⁸ §3.1. Throughout this document the term "password" is used to mean any password, passphrase, PIN, or other memorized secret.

Pepper A secret added to a password hash like a salt. Unlike a salt, peppers are secret and not unique. They must not be stored alongside the hashed password.

4 SASL Mechanisms

Clients and servers must already implement the SASL mechanisms listed in RFC 6120 §13.8.1 For Authentication Only. These mechanisms are:

- SCRAM-SHA-1
- SCRAM-SHA-1-PLUS

In addition, clients and servers SHOULD support the following SCRAM variants defined in RFC 7677 ⁹:

- SCRAM-SHA-256
- SCRAM-SHA-256-PLUS

Clients SHOULD NOT invent their own mechanisms that have not been standardized by the IETF, the XSF, or another reputable standards body.

Clients MUST NOT implement any mechanism with a usage status of "OBSOLETE", "MUST NOT be used", or "LIMITED" in the [IANA SASL Mechanisms Registry](#) ¹⁰. Similarly, any mechanism that depends on a hash function listed as "MUST NOT" in [Cryptographic Hash Function Recommendations for XMPP \(XEP-0414\)](#) ¹¹ MUST NOT be used. This means that the following mechanisms which were commonly used with XMPP in the past MUST NOT be supported:

- CRAM-MD5 ([RFC 2195](#) ¹²)
- DIGEST-MD5 ([RFC 6331](#) ¹³)

⁸Recommendation for Password-Based Key Derivation, Part 1: Storage Applications, NIST Special Publication 800-132 <<https://doi.org/10.6028/NIST.SP.800-132>>.

⁹RFC 7677: SCRAM-SHA-256 and SCRAM-SHA-256-PLUS Simple Authentication and Security Layer (SASL) Mechanisms <<http://tools.ietf.org/html/rfc7677>>.

¹⁰IANA registry of mechanisms used in the Simple Authentication and Security Layer protocol <<http://www.iana.org/assignments/sasl-mechanisms>>.

¹¹XEP-0414: Cryptographic Hash Function Recommendations for XMPP <<https://xmpp.org/extensions/xep-0414.html>>.

¹²RFC 2195: IMAP/POP AUTHorize Extension for Simple Challenge/Response <<http://tools.ietf.org/html/rfc2195>>.

¹³RFC 6331: Moving DIGEST-MD5 to Historic <<http://tools.ietf.org/html/rfc6331>>.

5 Client Best Practices

5.1 Mechanism Pinning

Clients maintain a list of preferred SASL mechanisms, generally ordered by perceived strength to enable strong authentication (RFC 6120¹⁴ §6.3.3 Mechanism Preferences). To prevent downgrade attacks by a malicious actor that has successfully man in the middle a connection, or compromised a trusted server's configuration, clients SHOULD implement "mechanism pinning". That is, after the first successful authentication with a strong mechanism, clients SHOULD make a record of the authentication and thereafter only advertise and use mechanisms of equal or higher perceived strength.

For reference, the following mechanisms are ordered by their perceived strength from strongest to weakest with mechanisms of equal strength on the same line. This list is a non-normative example and does not indicate that these mechanisms should or should not be supported:

1. EXTERNAL
2. SCRAM-SHA-1-PLUS, SCRAM-SHA-256-PLUS
3. SCRAM-SHA-1, SCRAM-SHA-256
4. PLAIN
5. DIGEST-MD5, CRAM-MD5

5.2 Storage

Clients SHOULD always store authenticators in a trusted and encrypted keystore such as the system keystore, or an encrypted store created specifically for the clients use. They SHOULD NOT store authenticators as plain text.

If clients know that they will only ever authenticate using a mechanism such as SCRAM where the original password is not needed (for example if the mechanism has been pinned) they SHOULD store the SCRAM bits or the hashed and salted password instead of the original password. However, if backwards compatibility with servers that only support the PLAIN mechanism or other mechanisms that require using the original password is required, clients MAY choose to store the original password so long as an appropriate keystore is used.

6 Server Best Practices

6.1 Additional SASL Requirements

Servers MUST NOT support any mechanism that would require authenticators to be stored in such a way that they could be recovered in plain text from the stored information. This

¹⁴RFC 6120: Extensible Messaging and Presence Protocol (XMPP): Core <<http://tools.ietf.org/html/rfc6120>>.

includes mechanisms that store authenticators using reversible encryption, obsolete hashing mechanisms such as MD5, and hashes that are unsuitable for use with authenticators such as SHA256.

6.2 Storage

Servers **MUST** always store passwords only after they have been salted and hashed. If multiple hashes are supported for use with SCRAM, for example SCRAM-SHA-1 and SCRAM-SHA-256, separate salted and hashed passwords **SHOULD** be calculated and stored for each mechanism so that users can log in with multiple clients that support only some of the mechanisms.

A distinct salt **SHOULD** be used for each user, and each SCRAM family supported. Salts **MUST** be generated using a cryptographically secure random number generator. The salt **MAY** be stored in the same datastore as the password. If it is stored alongside the password, it **SHOULD** be combined with a pepper stored in the application configuration, an environment variable, or some other location other than the datastore containing the salts.

6.3 Authentication and Rotation

When authenticating using PLAIN or similar mechanisms that involve transmitting the original password to the server the password **MUST** be hashed and compared against the salted and hashed password in the database using a constant time comparison.

Each time a password is changed or reset, a new random salt should be created and the iteration count and pepper (if applicable) should be updated to the latest value required by server policy.

If a pepper is used, consideration should be taken to ensure that it can be easily rotated. For example, multiple peppers could be stored with new passwords and reset passwords using the latest pepper. A hash of the pepper using a cryptographically secure hash function such as SHA256 could then be stored in the database next to the salt so that future logins can identify which pepper in the list was used. This is just one example, pepper rotation schemes are outside the scope of this document.

7 PBKDF2 Parameters

Because the PBKDF2 key derivation function ([RFC 8018](#) ¹⁵) is used by SCRAM-SHA-1 which is mandated for use in XMPP, this document recommends it for password storage. Servers **SHOULD** use the following parameters when applying PBKDF2:

Minimum iterations 10,000 (100,000 for higher security environments)

¹⁵RFC 8018: PKCS #5: Password-Based Cryptography Specification Version 2.1 <<http://tools.ietf.org/html/rfc8018>>.

Minimum salt length 16 bytes

Minimum pepper length 32 bytes

The minimum iteration count may be tuned to the specific system on which password hashing is taking place.

8 Password Complexity Requirements

Clients and servers SHOULD enforce a minimum length of 8 characters for user passwords. If using a mechanism such as PLAIN where the server performs hashing on the original password, a maximum length between 64 and 128 characters MAY be imposed to prevent denial of service (DoS) attacks. Passwords SHOULD be required to conform to the Opaque String profile of [RFC 8265](#)¹⁶. No other password restrictions should be applied.

9 Security Considerations

The SCRAM suite of SASL mechanisms are recommended in this document, however, there is currently no way to force a password reset. This reduces upgrade agility if a weakness is discovered in SCRAM and means that new, untested, SCRAM-based or SCRAM-like mechanisms should be added with caution.

This document mentions many hash functions that are already in use in the XMPP ecosystem, or that have been used in the past. It does not make recommendations for what functions should or should not be used in new applications. For recommendations about the use of hash functions and their security implications, see [Cryptographic Hash Function Recommendations for XMPP \(XEP-0414\)](#)¹⁷

This document contains recommendations that are likely to change over time. It should be reviewed yearly to ensure that it remains accurate and up to date. Many of the recommendations in this document were taken from the [OWASP Password Storage Cheat Sheet](#)¹⁸, which can be used as a reference when making updates.

¹⁶RFC 8265: Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords <<http://tools.ietf.org/html/rfc8265>>.

¹⁷XEP-0414: Cryptographic Hash Function Recommendations for XMPP <<https://xmpp.org/extensions/xep-0414.html>>.

¹⁸OWASP Cheat Sheet Series for password storage <https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html>.

10 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)¹⁹.

11 XMPP Registrar Considerations

No namespaces or parameters need to be registered with the [XMPP Registrar](#)²⁰ as a result of this document.

¹⁹The 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/>.

²⁰The 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/>.