The Hashed Token SASL Mechanism

Abstract

This I-D is developed at https://github.com/Flowdalic/xeps/tree/master/draft-schmaus-kitten-sasl-ht

This document specifies the family of Hashed Token SASL mechanisms which enable a proof-of-possession-based authentication scheme and are meant to be used for quick re-authentication of a previous session. The Hashed Token SASL mechanism's authentication sequence consists of only one round-trip. The usage of short-lived, exclusively ephemeral hashed tokens is achieving the single round-trip property. The SASL mechanism specified herein further provides hash agility, mutual authentication and is secured by channel binding.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 April 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

This specification describes the family of Hashed Token (HT) Simple Authentication and Security Layer (SASL) [RFC4422] mechanisms, which enable a proof-of-possession-based authentication scheme. The HT mechanism is designed to be used with short-lived, exclusively ephemeral tokens, called SASL-HT tokens, and allow for quick, one round-trip, re-authentication of a previous session.

Further properties of the HT mechanism are 1) hash agility, 2) mutual authentication, and 3) being secured by channel binding.

Clients are supposed to request SASL-HT tokens from the server after being authenticated using a "strong" SASL mechanism like SCRAM [RFC5802]. Hence a typical sequence of actions using HT may look like the following:
A) Client authenticates using a strong mechanism (e.g., SCRAM)
B) Client requests secret SASL-HT token
C) Service returns SASL-HT token
   <normal client-server interaction here>
D) Connection between client and server gets interrupted,
   for example because of a WiFi ↔ GSM switch
E) Client resumes the previous session using HT and token from C)
F) Service revokes the successfully used SASL-HT token
   [goto B]

The HT mechanism requires an accompanying, application protocol
specific, extension, which allows clients to requests a new SASL-HT
token (see Section 5 (Section 5)). One example for such an
application protocol specific extension based on HT is [XEP-0397].
This XMPP [RFC6120] extension protocol allows, amongst other things,
B) and C),

Since the SASL-HT token is not salted, and only one hash iteration
is used, the HT mechanism is not suitable to protect long-lived
shared secrets (e.g. "passwords"). You may want to look at [RFC5802]
for that.

1.1. Conventions and Terminology

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
BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here. These words may also appear in this
document in lower case as plain English words, absent their
normative meanings.

1.2. Applicability

Because this mechanism transports information that should not be
controlled by an attacker, the HT mechanism MUST only be used over
channels protected by Transport Layer Security (TLS, see [RFC5246]),
or over similar integrity-protected and authenticated channels.
Also, the application protocol specific extension which requests a
new SASL-HT token SHOULD only be used over similarly protected
channels.

Also, when TLS is used, the client MUST successfully validate the
server's certificate ([RFC5280], [RFC6125]).

The family of HT mechanisms is not applicable for proxy
authentication since they can not carry an authorization identity
string (authzid).
2. The HT Family of Mechanisms

Each mechanism in this family differs by choice of the hash algorithm and the choice of the channel binding \([\text{RFC5929}]\) type.

An HT mechanism name is a string beginning with "HT-" followed by the capitalised name of the used hash, followed by "-", and suffixed by one of 'ENDP' and 'UNIQ'.

Hence each HT mechanism has a name of the following form:

HT-<hash-alg>-<cb-type>

Where <hash-alg> is the capitalised "Hash Name String" of the IANA "Named Information Hash Algorithm Registry" \([\text{iana-hash-alg}]\) as specified in \([\text{RFC6920}]\), and <cb-type> is one of 'ENDP', 'UNIQ', or 'EXPR' denoting the channel binding type. In the case of 'ENDP', the tls-server-end-point channel binding type is used. In the case of 'UNIQ', the tls-unique channel binding type is used. In the case of 'EXPR', the tls-exporter \([\text{RFC9266}]\) channel binding type is used. Valid channel binding types are defined in the IANA "Channel-Binding Types" registry \([\text{iana-cbt}]\) as specified in \([\text{RFC5056}]\).

<table>
<thead>
<tr>
<th>cb-type</th>
<th>Channel Binding Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDP</td>
<td>tls-server-end-point</td>
</tr>
<tr>
<td>UNIQ</td>
<td>tls-unique</td>
</tr>
<tr>
<td>EXPR</td>
<td>tls-exporter</td>
</tr>
</tbody>
</table>

Table 1: Mapping of cb-type to Channel Binding Types

The following table lists some examples of HT SASL mechanisms registered by this document.

<table>
<thead>
<tr>
<th>Mechanism Name</th>
<th>HT Hash Algorithm</th>
<th>Channel-binding unique prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>HT-SHA-512-ENDP</td>
<td>SHA-512</td>
<td>tls-server-end-point</td>
</tr>
<tr>
<td>HT-SHA-512-UNIQ</td>
<td>SHA-512</td>
<td>tls-unique</td>
</tr>
<tr>
<td>HT-SHA3-512-ENDP</td>
<td>SHA3-512</td>
<td>tls-server-end-point</td>
</tr>
<tr>
<td>HT-SHA-256-UNIQ</td>
<td>SHA-256</td>
<td>tls-unique</td>
</tr>
</tbody>
</table>

Table 2: Examples of HT SASL mechanisms

3. The HT Authentication Exchange

The mechanism consists of a simple exchange of precisely two messages between the initiator and responder.

The following syntax specifications use the Augmented Backus-Naur form (ABNF) notation as specified in \([\text{RFC5234}]\).
3.1. Initiator First Message

The HT mechanism starts with the initiator-msg, send by the initiator to the responder. The following lists the ABNF grammar for the initiator-msg:

\[\text{initiator-msg} = \text{authcid NUL initiator-hashed-token}\]

- \[\text{authcid} = 1^*\text{SAFE} ; \text{MUST} \text{ accept up to 255 octets}\]
- \[\text{initiator-hashed-token} = 1^*\text{OCTET}\]

\[\text{NUL} = \%0x00 ; \text{The null octet}\]
\[\text{SAFE} = \text{UTF1} / \text{UTF2} / \text{UTF3} / \text{UTF4}\]

\[;\text{any UTF-8 encoded Unicode character except NUL}\]
\[\text{UTF1} = \%x01-7F ;; \text{except NUL}\]
\[\text{UTF2} = \%xC2-DF \text{UTF0}\]
\[\text{UTF3} = \%xE0 \%xA0-BF \text{UTF0} / \%xE1-EC 2(UTF0) / \%xED \%x80-9F \text{UTF0} / \%xEE-EF 2(UTF0)\]
\[\text{UTF4} = \%xF0 \%x90-BF 2(UTF0) / \%xF1-F3 3(UTF0) / \%xF4 \%x80-8F 2(UTF0)\]
\[\text{UTF0} = \%x80-BF\]

The initiator first message starts with the authentication identity (authcid, see [RFC4422]) as UTF-8 [RFC3629] encoded string. It is followed by initiator-hashed-token separated by as single null octet.

The value of the initiator-hashed-token is defined as follows:

\[\text{initiator-hashed-token} := \text{HMAC(token, "Initiator" || cb-data)}\]

HMAC() is the function defined in [RFC2104] with H being the selected HT hash algorithm, 'cb-data' represents the data provided by the selected channel binding type, and 'token' are the UTF-8 encoded octets of the SASL-HT token string which acts as a shared secret between initiator and responder.

The initiator-msg \textbf{MAY} be included in TLS 1.3 0-RTT early data, as specified in [RFC8446]. If this is the case, then the initiating entity \textbf{MUST NOT} include any further application protocol payload in the early data besides the HT initiator-msg and potential required framing of the SASL profile. The responder \textbf{MUST} abort the SASL authentication if the early data contains additional application protocol payload.

SASL-HT hence allows exploiting TLS 1.3 early data for "0.5 Round Trip Time (RTT)" resumption of the application protocol's session. Using TLS early data requires extra care when implementing: The early data should only contain the SASL-HT payload, i.e., the initiator-msg, and not an application protocol...
specific payload. The reason for this is that the early data could be replayed, and thus needs to carry an idempotent operation. On the other hand, if the responding entity can verify the early data, then it can send additional application protocol payload together with the "resumption successful" response to the initiating entity.

3.2. Initiator Authentication

Upon receiving the initiator-msg, the responder calculates itself the value of initiator-hashed-token and compares it with the received value found in the initiator-msg. If both values are equal, then the initiator has been successfully authenticated. Otherwise, if both values are not equal, then authentication MUST fail.

If the responder was able to authenticate the initiator, then the used token MUST be revoked immediately.

3.3. Final Responder Message

After the initiator was authenticated the responder continues the SASL authentication by sending the responder-msg to the initiator.

The ABNF for responder-msg is:

responder-msg = 1*OCTET

The responder-msg value is defined as follows:

responder-msg := HMAC(token, "Responder" || cb-data)

The initiating entity MUST verify the responder-msg to achieve mutual authentication.

4. Compliance with SASL Mechanism Requirements

This section describes compliance with SASL mechanism requirements specified in Section 5 of [RFC4422].

1. "HT-SHA-256-ENDP", "HT-SHA-256-UNIQ", "HT-SHA-3-512-ENDP" and "HT-SHA-3-512-UNIQ".

2. Definition of server-challenges and client-responses: a) HT is a client-first mechanism. b) HT does send additional data with success (the responder-msg).

3. HT is not capable of transferring authorization identities from the client to the server.
4. HT does not offer any security layers (HT offers channel binding instead).

5. HT does not protect the authorization identity.

5. Requirements for the Application-Protocol Extension

It is REQUIRED that the application-protocol specific extension provides a mechanism to request a SASL-HT token in form of a Unicode string. The returned token MUST have been newly generated by a cryptographically secure random number generator and MUST contain at least 128 bit of entropy.

It is RECOMMENDED that the protocol allows the requestor to signal the name of the SASL mechanism which he intends to use with the token. If a token is used with a different mechanism than the one which was signalled upon requesting the token, then the authentication MUST fail. This allows pinning the token to a SASL mechanism, which increases the security because it makes it impossible for an attacker to downgrade the SASL mechanism.

6. Security Considerations

To be secure, the HT mechanism MUST be used over a TLS channel that has had the session hash extension [RFC7627] negotiated, or session resumption MUST NOT have been used.

It is RECOMMENDED that implementations periodically require a full authentication using a strong SASL mechanism which does not use the SASL-HT token.

It is of vital importance that the SASL-HT token is generated by a cryptographically secure random generator. See [RFC4086] for more information about Randomness Requirements for Security.

7. IANA Considerations

IANA is requested to add the following family of SASL mechanisms to the SASL Mechanism registry established by [RFC4422]:

To: iana@iana.org

Subject: Registration of a new SASL family HT

SASL mechanism name (or prefix for the family): HT-*

Security considerations: Section 6 of draft-schmaus-kitten-sasl-ht
Published specification (optional, recommended): draft-schmaus-kitten-sasl-ht-XX (TODO)

Person & email address to contact for further information: IETF SASL WG kitten@ietf.org

Intended usage: COMMON

Owner/Change controller: IESG iesg@ietf.org

Note: Members of this family MUST be explicitly registered using the "IETF Review" [RFC5226] registration procedure. Reviews MUST be requested on the Kitten WG mailing list kitten@ietf.org (or a successor designated by the responsible Security AD).

8. References

8.1. Normative References


[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/


8.2. Informative References


Acknowledgments

This document benefited from discussions on the KITTEN WG mailing list. The authors would like to especially thank Thijs Alkemade, Sam Whited and Alexey Melnikov for their comments on this topic. Furthermore, we would like to thank Alexander Wuerstlein, who came up with the idea to pin the token to a SASL mechanism for increased security.

Authors' Addresses

Florian Schmaus
FAU Erlangen-Nürnberg
Email: flow@cs.fau.de

Christoph Egger
FAU Erlangen-Nürnberg
Email: egger@cs.fau.de