kitten
Internet-Draft
Intended status: Informational
Expires: 1 September 2025

B. Bucksch
Beonex
S. Farrell
Trinity College Dublin

28 February 2025

SASL Remember Me
draft-ietf-kitten-sasl-rememberme-00

Abstract

Introduces a SASL mechanism that allows the application to stay logged in and re-login without user interaction, after completing a time-consuming SASL login mechanism that involves the user.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-kitten-sasl-rememberme/.

Discussion of this document takes place on the kitten Working Group mailing list (mailto:kitten@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/kitten/. Subscribe at https://www.ietf.org/mailman/listinfo/kitten/.

Source for this draft and an issue tracker can be found at https://github.com/benbucksch/sasl-rememberme.

Status of This Memo

Table of Contents

1. Introduction

A client application might at first log in to a server using Passkey or multi-factor authentication. However, these mechanisms require complicated user interaction. These are too costly to perform at every login. They can realistically only be completed once during setup. To stay logged in, this SASL mechanism allows the server to create a client-specific token that the client application can then use to log in to the same account, without further user interaction.

The token is opaque to the client and the server can choose how to implement it. It may be a custom random string that the server stores, or an application-specific password generated for this client together with the username, or a JWT token, or a refresh token which does not expire.

2. Creation of the token

The client requests the token from the server using either 1. application-specific commands, like IMAP REMEMBERME, which are to be defined in other standards. 2. the success response of another SASL mechanism.

The client stores the token, instead of a password, in a client-specific storage that is as secure as a password storage.

3. Login using the token

If the client needs to log in and has a token for this user and host, uses the SASL REMEMBERME mechanism to log in. REMEMBERME mechanism starts with the client sending the initial client response, which has the following format defined using ABNF:

```
rememberme-client-step1 = token
token                   = 1*OCTET
```

If the server accepts the response as valid and allows login, it responds with a SASL success response. The user is logged in. (ALTERNATIVE EXPIRY: .... it responds with a SASL success response, a new token, and the expiry time of the new token. The user is logged in. To avoid race conditions in clients that open multiple connections at the same time, the previously used token MUST be valid for at least 30 more seconds. Likewise, if the server returned a new token, then it must return the same new token in response to the same old token for the next 1 hour. Reason: If the client opens 5 connections at the same time, using the same token, but the server were to respond with 5 different new tokens, and it were to allow only 1 of them, then the client would not know which one to store, due to race conditions in the network response.)

If the response is invalid, the server responds with a SASL error and a human-readable error message for the end user.

```
   server-final-message = server-error "," server-error-message
           ; Only returned on error. Omitted on success.

   server-error = "e=" server-error-value

   server-error-value = "invalid-encoding" /
                        "unknown-user" /
                        "invalid-username-encoding" /
                          ; invalid username encoding (invalid UTF-8 or
                          ; SASLprep failed)
                        "other-error" /
                        server-error-value-ext
           ; Unrecognized errors should be treated as "other-error".
           ; In order to prevent information disclosure, the server
           ; may substitute the real reason with "other-error".

   server-error-value-ext = value
           ; Additional error reasons added by extensions
           ; to this document.

   server-error-message = "m=" server-error-message-value

   server-error-message-value = 1*OCTET
           ; Human readable error message in UTF-8
```

4. IMAP Example

In IMAP, the exchange would be:

```
S: * OK ACME IMAP Server v1.23 is ready
C: 22 CAPABILITY
S: 22 CAPABILITY IMAP4rev1 IMAP4rev2 AUTH=PASSKEY AUTH=REMEMBERME
C: 23 AUTHENTICATE REMEMBERME QUVDNjU3NjU3NjU1Nwo=
S: 23 OK AUTHENTICATE completed
```

In the above example the token is "AEC6576576557" which is base64-encoded according to IMAP SASL profile.

5. Requirements on the token

1. The token MUST also allow the server to infer the authentication identity (e.g. username or email address). The token alone must be sufficient to log in. This SASL mechanism doesn’t support separate authorization identities.

2. The token MUST NOT expire. (ALTERNATIVE EXPIRY: The token MUST be valid for at least 30 days.)

3. The server SHOULD be able to revoke the tokens, in case this specific user account or device was compromised. In this case, the authentication MUST fail and the SASL error message MUST explain the situation to the user and give instructions how to remedy the situation.

6. Conventions and Definitions

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.

7. Security Considerations

Clients should treat the token like a password and store it securely.

TODO Security

8. IANA Considerations

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

To: iana@iana.org
Subject: Registration of a new SASL mechanism REMEMBERME

SASL mechanism name (or prefix for the family): REMEMBERME
Security considerations: Section YY of [RFCXXXX]
Published specification (optional, recommended): [RFCXXXX]
Person & email address to contact for further information:
IETF Kitten WG <kitten@ietf.org>
Intended usage: COMMON
Owner/Change controller: IESG <iesg@ietf.org>
Note:

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4422] Melnikov, A., Ed. and K. Zeilenga, Ed., "Simple Authentication and Security Layer (SASL)", RFC 4422, DOI 10.17487/RFC4422, June 2006, <https://www.rfc-editor.org/rfc/rfc4422>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Ben Bucksch
Beonex
Email: ben.bucksch@beonex.com

Stephen Farrell
Trinity College Dublin
Email: stephen.farrell@cs.tcd.ie