Workgroup: Network Working Group
Internet-Draft: draft-ietf-kitten-scram-2fa-01
Published: 25 January 2022
Intended Status: Standards Track
Expires: 29 July 2022
Authors:
A. Melnikov
Isode Ltd

Extensions to Salted Challenge Response (SCRAM) for 2 factor authentication

Abstract

This specification describes an extension to family of Simple Authentication and Security Layer (SASL; RFC 4422) authentication mechanisms called the Salted Challenge Response Authentication Mechanism (SCRAM), which provides support for 2 factor authentication. It also includes a separate extension for quick reauthentication.

This specification also gives an example of how TOTP (RFC 6238) can be used as the second factor.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Conventions Used in This Document
  2.1. Terminology
  2.2. Notation
3. SCRAM Extension for 2FA
4. SCRAM Extension for reauthentication
5. Formal Syntax
6. Use of TOTP with SCRAM
7. Example
8. Open Issues
9. Security Considerations
10. IANA Considerations
11. Acknowledgements
12. Normative References
Author's Address

1. Introduction

SCRAM [RFC5802] is a password based SASL [RFC4422] authentication mechanism that provides (among other things) mutual authentication and binding to an external security layer such as TLS.

Two-factor authentication (2FA) is a way to add additional security to an authentication exchange. The first "factor" is a password. The second "factor" is a verification code retrieved from an application on a mobile device or computer. 2FA is conceptually similar to a security token device that banks in some countries require for online banking. Other names for 2FA systems include OTP (one-time password) and TOTP (Time-based One-time Password algorithm, such as [RFC6238]).

This specification describes an extension to SCRAM to provide 2 factor authentication. SCRAM already relies on passwords for authentication. This document specifies how second "factors" can be incorporated into SCRAM authentication. It also includes a separate (but frequently used together with the 2 factor authentication) extension for quick reauthentication.

2. Conventions Used in This Document

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

Formal syntax is defined by [RFC5234] including the core rules defined in Appendix B of [RFC5234].

Example lines prefaced by "C:" are sent by the client and ones prefaced by "S:" by the server. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only, and are not part of the actual protocol exchange.

2.1. Terminology

This document uses several terms defined in [RFC4949] ("Internet Security Glossary") including the following: authentication, authentication exchange, authentication information, brute force, challenge-response, cryptographic hash function, dictionary attack, eavesdropping, hash result, keyed hash, man-in-the-middle, nonce, one-way encryption function, password, replay attack and salt. Readers not familiar with these terms should use that glossary as a reference. Other terms defined in [RFC5802] are also used in this document.

2.2. Notation

This document reuses notation defined in SCRAM.

3. SCRAM Extension for 2FA

This extension doesn't add any extra roundtrips to SCRAM authentication. SCRAM was designed to be extensible, so it allows for optional and mandatory attributes, which covered by MAC codes. Second "factors" are conveyed in the second message ("client-final-message-without-proof" ABNF production) sent from the client to the server.

This extension doesn't change how the client authenticates the server.

The server authenticates the client after receiving the second message as described in Section 3 of [RFC5802] If the client included "type" and "second-factor" attributes defined in this document (see Section 5) and the server supports the specified second factor type, the server verifies content of the "second-factor" according to the "type". If the second factor verification fails, the server MUST fail authentication and SHOULD return either "replayed-second-factor" or "invalid-second-factor" error in the "e" attribute. [[It would be possible to make the extra attributes mandatory by using SCRAM's "m=", but the text above doesn't do that. This is one of open issues to resolve.]]

4. SCRAM Extension for reauthentication

This reauthentication extension to SCRAM allows the server to return a token that can be used for quick reauthentication and bypasses 2 factor authentication prompt to the user. The reauthentication token is randomly generated value. The reauthentication token is returned in the "o" attribute that is appended to the end of the "server-final-message".

[[Note: it would be possible to extend SCRAM itself to do reauthentication, by including an earlier received reauthentication token in the "client-first-message" of a subsequent SCRAM authentication. This will also turn off the server checking for 2 factor authentication information, unless the reauthentication attempt is rejected by the server. In the meantime, this document presents a couple of other alternatives on how to use other SASL mechanisms with the reauthentication token.]]

When the CLIENT-KEY/CLIENT-KEY-PLUS mechanism (see draft-cridland-kitten-clientkey) is used for the reauthentication after a successful SCRAM authentication, the reauthentication token is the Client Secret Key. [[Need to also somehow convey token expiration?]]

When the HT-* mechanism (see draft-schmaus-kitten-sasl-ht) is used for the reauthentication after a successful SCRAM authentication, the reauthentication token is the draft-schmaus-kitten-sasl-ht token. [[Note that the HT hash should probably match the SCRAM hash used.]]

5. Formal Syntax

This document defines the following new SCRAM attributes:

* t: This attribute specifies the type of second factor. [[Create IANA registry for these?]] This document defines one type: "totp". If this attribute is specified, the "f" attribute MUST also be specified.

* f: This attribute specifies the value of the second factor. For "t=totp" it is 6 digit decimal number. [[Use 8 digits per Rick van Rein?]] This attribute MUST be ignored unless the "t" attribute is also specified.

* o: This attribute specifies the base64-encoded value of the reauthentication token.

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [RFC5234].

```
type              = "t=" type-value
                ; Complies with "attr-val" syntax.
type-value        = "totp" / value
                ; Type of second factor.
                ; Should be registered with IANA.
second-factor     = "f=" second-factor-value
                ; Complies with "attr-val" syntax.
second-factor-value = 6DIGIT / value
                ; 6DIGIT when "t=totp"
server-error-value-ext =
        "replayed-second-factor" /
        "invalid-second-factor" /
        "second-factor-value-missing"

value = <as defined in RFC 5802>


reauth-token = "o=" base64
                  ;; base64 encoding of reauthentication
                  ;; token.
```

6. Use of TOTP with SCRAM

When TOTP is used with SCRAM, the following values for "t" and "f" attributes (see Section 5 for their generic syntax) are used:

* t: This attribute specifies the type of second factor. For TOTP the value is "totp". If this attribute is specified, the "f" attribute MUST also be specified.

* f: This attribute specifies the value of the second factor. For "t=totp" it is 6 digit decimal number. This attribute MUST be ignored unless the "t" attribute is also specified.

A TOTP URI is specified with the following ABNF:

```
totp-uri = "otpauth" "://" "totp/" label "?secret=" secret
           "&issuer=" issuer
label = issuer (":" / "%3A") identity
identity = 1*CHAR ; URI-encoded SASL identity
secret = 40 * HEXCHAR ; Base32 (hex) encoded secret with no padding.
issuer = 1*CHAR ; Issuer name.
```

7. Example

The following example extends the example from Section 5 of [RFC5802] to demonstrate use of TOTP:

```
C: n,,n=user,r=fyko+d2lbbFgONRv9qkxdawL
S: r=fyko+d2lbbFgONRv9qkxdawL3rfcNHYJY1ZVvWVs7j,s=QSXCR+Q6sek8bf92,
   i=4096
C: c=biws,r=fyko+d2lbbFgONRv9qkxdawL3rfcNHYJY1ZVvWVs7j,
   t=totp,f=776804,p=v0X8v3Bz2T0CJGbJQyF0X+HI4Ts=
S: v=lz59pqV8S7suAoZWja4dJRkFsKQ=
```

Please note that TOTP extension described in this document works in the same way with SCRAM-SHA-256/SCRAM-SHA-256-PLUS, SCRAM-SHA-512/SCRAM-SHA-512-PLUS or any other SCRAM variants that use other hash functions.

8. Open Issues

Simon Josefsson: should this be a new SASL mechanism name, e.g. CROTP-SHA-256?

Simon Josefsson: cookie option for fast reauthentication? Alexey: can do or just used CLIENT-KEY (draft-cridland-kitten-clientkey)?

Rick van Rein: specify a HOTP variant as well?

Rick van Rein: use TOTP with 6 or 8 digits? Register both variants?

9. Security Considerations

Unless an external security layer (such as TLS) is also used, the OTP value is sent in unencrypted/unhashed form from the client to the server, which allows an attacker to read the OTP value and perform a race with the server to validate the OTP.

TBD

10. IANA Considerations

IANA is requested to update the definition of the SASL family SCRAM in the SASL Mechanism registry established by [RFC4422] to also point to this document.

IANA is also requested to create a new subregistry of "SASL mechanism" for registering second factor schemes used in the "t" attribute as specified in this document.

The registration template is as follows:

```
SCRAM Second Factor Scheme Name:
Pointer to specification text:
Notes (optional):
```

The registration procedure for the above subregistry is Expert Review.

IANA is requested to register a new value in the subregistry defined above:

```
SCRAM Second Factor Scheme Name: TOTP
Pointer to specification text: [[ this document ]]
Notes (optional): (none)
```

11. Acknowledgements

Thank you to Stephen Farrell for motivating creation of this document and to Dave Cridland for describing how TOTP can be used with XMPP in XEP-0400. Thank you to Rick van Rein and Simon Josefsson for comments and corrections, but all final errors in this document remain mine.

12. 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/info/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/info/rfc4422>.

[RFC4949]
Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC5802] Newman, C., Menon-Sen, A., Melnikov, A., and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms", RFC 5802, DOI 10.17487/RFC5802, July 2010, <https://www.rfc-editor.org/info/rfc5802>.

[RFC6238] M'Raihi, D., Machani, S., Pei, M., and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm", RFC 6238, DOI 10.17487/RFC6238, May 2011, <https://www.rfc-editor.org/info/rfc6238>.

Author's Address

Alexey Melnikov
Isode Ltd

Email: Alexey.Melnikov@isode.com