Network Working Group
Internet Draft: OTP SASL Mechanism
Updates: 2222
Document: draft-ietf-otp-sasl-02.txt

C. Newman
Innosoft

May 1998
Expires in six months

# The One-Time-Password SASL Mechanism

## Status of this memo



## Abstract

OTP [OTP] provides a useful authentication mechanism for situations where there is limited client or server trust. Currently, OTP is added to protocols in an ad-hoc fashion with heuristic parsing. This specification defines an OTP SASL [SASL] mechanism so it can be easily and formally integrated into many application protocols.

## 1. How to Read This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED" and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

This memo assumes the reader is familiar with OTP [OTP], OTP extended responses [OTP-EXT] and SASL [SASL].

## 2. Intended Use

The OTP SASL mechanism replaces the SKEY SASL mechanism [SASL]. OTP is a good choice for usage scenarios where the client is untrusted (e.g., a kiosk client), as a one-time password will only give the client a single opportunity to act on behalf of the user. OTP is also a good choice for situations where interactive logins are permitted to the server, as a compromised OTP authentication database is only subject to dictionary attacks, unlike authentication databases for other simple mechanisms such as CRAM-MD5 [CRAM-MD5]. It is important to note that each use of the OTP mechanism causes the authentication database entry for a user to be updated.

This SASL mechanism provides a formal way to integrate OTP into SASL-enabled protocols including IMAP [IMAP4], ACAP [ACAP], POP3 [POP-AUTH] and LDAPv3 [LDAPv3].

## 3. Profiling OTP for SASL

OTP [OTP] and OTP extended responses [OTP-EXT] offer a number of options. However, for authentication to succeed, the client and server need compatible option sets. This specification defines a single SASL mechanism: OTP. The following rules apply to this mechanism:

- The extended response syntax MUST be used.
- Servers MUST support the following four OTP extended responses: "hex", "word", "init-hex" and "init-word". Servers MUST support the "word" and "init-word" responses for the standard dictionary and SHOULD support alternate dictionaries. Servers MUST NOT require use of any additional OTP extensions or options.
- Clients SHOULD support display of the OTP challenge to the user and entry of an OTP in multi-word format. Clients MAY also support direct entry of the pass phrase and compute the "hex" or "word" response.
- Clients MUST indicate when authentication fails due to the sequence number getting too low and SHOULD offer the user the option to reset the sequence using the "init-hex" or "init-word" response.

Support for the MD5 algorithm is REQUIRED, and support for the SHA1 algorithm is RECOMMENDED.

## 4. OTP Authentication Mechanism

The mechanism does not provide any security layer.

The client begins by sending a message to the server containing the following two pieces of information.

(1) An authorization identity. When the empty string is used, this defaults to the authentication identity. This is used by system administrators or proxy servers to login with a different user identity. This field may be up to 255 octets and is terminated by a NUL (0) octet. US-ASCII printable characters are preferred, although UTF-8 [UTF-8] printable characters are permitted to support international names. Use of character sets other than US-ASCII and UTF-8 is forbidden.

(2) An authentication identity. The identity whose pass phrase will be used. This field may be up to 255 octets. US-ASCII printable characters are preferred, although UTF-8 [UTF-8] printable characters are permitted to support international names. Use of character sets other than US-ASCII and UTF-8 is forbidden.

The server responds by sending a message containing the OTP challenge as described in OTP [OTP] and OTP extended responses [OTP-EXT].

If a client sees an unknown hash algorithm name it will not be able to process a pass phrase input by the user. In this situation the client MAY prompt for the six-word format, issue the cancel sequence as specified by the SASL profile for the protocol in use and try a different SASL mechanism, or close the connection and refuse to authenticate. As a result of this behavior, a server is restricted to one OTP hash algorithm per user.

On success, the client generates an extended response in the "hex", "word", "init-hex" or "init-word" format. The client is not required to terminate the response with a space or a newline and SHOULD NOT include unnecessary whitespace.

Servers MUST tolerate input of arbitrary length, but MAY fail the authentication if the length of client input exceeds reasonable size.

## 5. Examples

In these example, "C:" represents lines sent from the client to the server and "S:" represents lines sent from the server to the client. The user name is "tim" and no authorization identity is

provided. The "<NUL>" below represents an ASCII NUL octet.

The following is an example of the OTP mechanism using the ACAP [ACAP] profile of SASL. The pass phrase used in this example is:

This is a test.

```
C: a001 AUTHENTICATE "OTP" {4}
C: <NUL>tim
S: + "otp-md5 499 ke1234 ext"
C: "hex:5bf075d9959d036f"
S: a001 OK "AUTHENTICATE completed"
```

Here is the same example using the six-words response:

```
C: a001 AUTHENTICATE "OTP" {4}
C: <NUL>tim
S: + "otp-md5 499 ke1234 ext"
C: "word:BOND FOGY DRAB NE RISE MART"
S: a001 OK "AUTHENTICATE completed"
```

Here is the same example using the OTP-SHA1 mechanism:

```
C: a001 AUTHENTICATE "OTP" {4}
C: <NUL>tim
S: + "otp-sha1 499 ke1234 ext"
C: "hex:c90fc02cc488df5e"
S: a001 OK "AUTHENTICATE completed"
```

Here is the same example with the init-hex extended response

```
C: a001 AUTHENTICATE "OTP" {4}
C: <NUL>tim
S: + "otp-md5 499 ke1234 ext"
C: "init-hex:5bf075d9959d036f:md5 499 ke1235:3712dcb4aa5316c1"
S: a001 OK "OTP sequence reset, authentication complete"
```

The following is an example of the OTP mechanism using the IMAP [IMAP4] profile of SASL. The pass phrase used in this example is:

this is a test

```
C: a001 AUTHENTICATE OTP
S: +
C: AHRpbQ==
S: + b3RwLW1kNSAxMjMga2UxMjM0IGV4dA==
C: aGV4OjExZDRjMTQ3ZTIyN2MxZjE=
S: a001 OK AUTHENTICATE completed
```

Note that the lack of an initial client response and the base64

encoding are characteristics of the IMAP profile of SASL. The server challenge is "otp-md5 123 ke1234 ext" and the client response is "hex:11d4c147e227c1f1".

## 6. Security Considerations

This specification introduces no security considerations beyond those those described in SASL [SASL], OTP [OTP] and OTP extended responses [OTP-EXT]. A brief summary of these considerations follows:

This mechanism does not provide session privacy, server authentication or protection from active attacks.

This mechanism is subject to passive dictionary attacks. The severity of this attack can be reduced by choosing pass phrases well.

The server authentication database necessary for use with OTP need not be plaintext-equivalent.

Server implementations MUST protect against the race attack [OTP].

## 7. Multinational Considerations

As remote access is a crucial service, users are encouraged to restrict user names and pass phrases to the US-ASCII character set. However, if characters outside the US-ASCII chracter set are used in user names and pass phrases, then they are interpreted according to UTF-8 [UTF-8].

Server support for alternate dictionaries is strongly RECOMMENDED to permit use of the six-word format with non-English words.

## 8. IANA Considerations

Here is the registration template for the OTP SASL mechanism:

SASL mechanism name: OTP
Security Considerations: See section 6 of this memo
Published specification: this memo
Person & email address to contact for futher information:
see author's address section below
Intended usage: COMMON
Author/Change controller: see author's address section below

This memo also amends the SKEY SASL mechanism registration [SASL] by changing its intended usage to OBSOLETE.

## 9. References

[ACAP] Newman, Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, Innosoft, Netscape, November 1997.

[CRAM-MD5] Klensin, Catoe, Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", RFC 2195, MCI, September 1997.

[IMAP4] Crispin, M., "Internet Message Access Protocol - Version 4rev1", RFC 2060, University of Washington, December 1996.

[KEYWORDS] Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

[LDAPv3] Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, Critical Angle Inc., Netscape Communications Corp, Isode Limited, December 1997.

[MD5] Rivest, "The MD5 Message Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science, April 1992.

[OTP] Haller, Metz, Nesser, Straw, "A One-Time Password System", RFC 2289, Bellcore, Kaman Sciences Corporation, Nesser & Nesser Consulting, February 1998.

[OTP-EXT] Metz, "OTP Extended Responses", RFC 2243, The Inner Net, November 1997.

[POP-AUTH] Myers, "POP3 AUTHentication command", RFC 1734, Carnegie Mellon, December 1994.

[SASL] Myers, "Simple Authentication and Security Layer (SASL)", RFC 2222, Netscape Communications, October 1997.

[UTF-8] Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, Alis Technologies, January 1998.

## 10. Author's Address

Chris Newman
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790 USA

Email: chris.newman@innosoft.com