Network Working Group
Request for Comments: 2095
Category: Standards Track

J. Klensin
R. Catoe
P. Krumviede
MCI

January 1997

IMAP/POP AUTHorize Extension for Simple Challenge/Response

Status of this Memo

Abstract

While IMAP4 supports a number of strong authentication mechanisms as described in RFC 1731, it lacks any mechanism that neither passes cleartext, reusable passwords across the network nor requires either a significant security infrastructure or that the mail server update a mail-system-wide user authentication file on each mail access. This specification provides a simple challenge-response authentication protocol that is suitable for use with IMAP4. Since it utilizes Keyed-MD5 digests and does not require that the secret be stored in the clear on the server, it may also constitute an improvement on APOP for POP3 use as specified in RFC 1734.

1. Introduction

Existing Proposed Standards specify an AUTHENTICATE mechanism for the IMAP4 protocol [IMAP, IMAP-AUTH] and a parallel AUTH mechanism for the POP3 protocol [POP3-AUTH]. The AUTHENTICATE mechanism is intended to be extensible; the four methods specified in [IMAP-AUTH] are all fairly powerful and require some security infrastructure to support. The base POP3 specification [POP3] also contains a lightweight challenge-response mechanism called APOP. APOP is associated with most of the risks associated with such protocols: in particular, it requires that both the client and server machines have access to the shared secret in cleartext form. CRAM offers a method for avoiding such cleartext storage while retaining the algorithmic simplicity of APOP in using only MD5, though in a "keyed" method.

At present, IMAP [IMAP] lacks any facility corresponding to APOP. The only alternative to the strong mechanisms identified in [IMAP-AUTH] is a presumably cleartext username and password, supported through the LOGIN command in [IMAP]. This document describes a simple challenge-response mechanism, similar to APOP and PPP CHAP [PPP], that can be used with IMAP (and, in principle, with POP3).

This mechanism also has the advantage over some possible alternatives of not requiring that the server maintain information about email "logins" on a per-login basis. While mechanisms that do require such per-login history records may offer enhanced security, protocols such as IMAP, which may have several connections between a given client and server open more or less simultaneous, may make their implementation particularly challenging.

2. Challenge-Response Authentication Mechanism (CRAM)

The authentication type associated with CRAM is "CRAM-MD5".

The data encoded in the first ready response contains an presumptively arbitrary string of random digits, a timestamp, and the fully-qualified primary host name of the server. The syntax of the unencoded form must correspond to that of an RFC 822 'msg-id' [RFC822] as described in [POP3].

The client makes note of the data and then responds with a string consisting of the user name, a space, and a 'digest'. The latter is computed by applying the keyed MD5 algorithm from [KEYED-MD5] where the key is a shared secret and the digested text is the timestamp (including angle-brackets).

This shared secret is a string known only to the client and server. The 'digest' parameter itself is a 16-octet value which is sent in hexadecimal format, using lower-case ASCII characters.

When the server receives this client response, it verifies the digest provided. If the digest is correct, the server should consider the client authenticated and respond appropriately.

Keyed MD5 is chosen for this application because of the greater security imparted to authentication of short messages. In addition, the use of the techniques described in [KEYED-MD5] for precomputation of intermediate results make it possible to avoid explicit cleartext storage of the shared secret on the server system by instead storing the intermediate results which are known as "contexts".

CRAM does not support a protection mechanism.

Example:

The examples in this document show the use of the CRAM mechanism with the IMAP4 AUTHENTICATE command [IMAP-AUTH]. The base64 encoding of the challenges and responses is part of the IMAP4 AUTHENTICATE command, not part of the CRAM specification itself.

```
S: * OK IMAP4 Server
C: A0001 AUTHENTICATE CRAM-MD5
S: + PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1jaS5uZXQ+
C: dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
S: A0001 OK CRAM authentication successful
```

In this example, the shared secret is the string 'tanstaaftanstaaf'. Hence, the Keyed MD5 digest is produced by calculating

```
MD5((tanstaaftanstaaf XOR opad),
    MD5((tanstaaftanstaaf XOR ipad),
    <1896.697170952@postoffice.reston.mci.net>))
```

where ipad and opad are as defined in the keyed-MD5 Work in Progress [KEYED-MD5] and the string shown in the challenge is the base64 encoding of <1896.697170952@postoffice.reston.mci.net>. The shared secret is null-padded to a length of 64 bytes. If the shared secret is longer than 64 bytes, the MD5 digest of the shared secret is used as a 16 byte input to the keyed MD5 calculation.

This produces a digest value (in hexadecimal) of

```
b913a602c7eda7a495b4e6e7334d3890
```

The user name is then prepended to it, forming

```
tim b913a602c7eda7a495b4e6e7334d3890
```

Which is then base64 encoded to meet the requirements of the IMAP4 AUTHENTICATE command (or the similar POP3 AUTH command), yielding

```
dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWI0ZTZlNzMzNGQzODkw
```

3. References

[CHAP] Lloyd, B., and W. Simpson, "PPP Authentication Protocols", RFC 1334, October 1992.

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

[IMAP-AUTH] Myers, J., "IMAP4 Authentication Mechanisms", RFC 1731, Carnegie Mellon, December 1994.

[KEYED-MD5] Krawczyk, H., "HMAC-MD5: Keyed-MD5 for Message Authentication", Work in Progess.

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

[POP3] Myers, J., and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, Carnegie Mellon, May 1996.

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

4. Security Considerations

It is conjectured that use of the CRAM authentication mechanism provides origin identification and replay protection for a session. Accordingly, a server that implements both a cleartext password command and this authentication type should not allow both methods of access for a given user.

While the saving, on the server, of "contexts" (see section 2) is marginally better than saving the shared secrets in cleartext as is required by CHAP [CHAP] and APOP [POP3], it is not sufficient to protect the secrets if the server itself is compromised. Consequently, servers that store the secrets or contexts must both be protected to a level appropriate to the potential information value in user mailboxes and identities.

As the length of the shared secret increases, so does the difficulty of deriving it.

While there are now suggestions in the literature that the use of MD5 and keyed MD5 in authentication procedures probably has a limited effective lifetime, the technique is now widely deployed and widely understood. It is believed that this general understanding may assist with the rapid replacement, by CRAM-MD5, of the current uses of permanent cleartext passwords in IMAP. This document has been

deliberately written to permit easy upgrading to use SHA (or whatever alternatives emerge) when they are considered to be widely available and adequately safe.

Even with the use of CRAM, users are still vulnerable to active attacks. An example of an increasingly common active attack is 'TCP Session Hijacking' as described in CERT Advisory CA-95:01 [CERT95].

See section 1 above for additional discussion.

5. Acknowledgements

This memo borrows ideas and some text liberally from [POP3] and [RFC-1731] and thanks are due the authors of those documents. Ran Atkinson made a number of valuable technical and editorial contributions to the document.

6. Authors' Addresses

John C. Klensin
MCI Telecommunications
800 Boylston St, 7th floor
Boston, MA 02199
USA

EMail: klensin@mci.net
Phone: +1 617 960 1011

Randy Catoe
MCI Telecommunications
2100 Reston Parkway
Reston, VA 22091
USA

EMail: randy@mci.net
Phone: +1 703 715 7366

Paul Krumviede
MCI Telecommunications
2100 Reston Parkway
Reston, VA 22091
USA

EMail: paul@mci.net
Phone: +1 703 715 7251