Network Working Group
Request for Comments: 1734
Category: Standards Track

J. Myers
Carnegie Mellon
December 1994

POP3 AUTHentication command

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

1. Introduction

This document describes the optional AUTH command, for indicating an authentication mechanism to the server, performing an authentication protocol exchange, and optionally negotiating a protection mechanism for subsequent protocol interactions. The authentication and protection mechanisms used by the POP3 AUTH command are those used by IMAP4.

2. The AUTH command

AUTH mechanism

Arguments:
a string identifying an IMAP4 authentication mechanism, such as defined by [IMAP4-AUTH]. Any use of the string "imap" used in a server authentication identity in the definition of an authentication mechanism is replaced with the string "pop".

Restrictions:
may only be given in the AUTHORIZATION state

Discussion:
The AUTH command indicates an authentication mechanism to the server. If the server supports the requested authentication mechanism, it performs an authentication protocol exchange to authenticate and identify the user. Optionally, it also negotiates a protection mechanism for subsequent protocol interactions. If the requested authentication mechanism is not supported, the server

should reject the AUTH command by sending a negative response.

The authentication protocol exchange consists of a series of server challenges and client answers that are specific to the authentication mechanism. A server challenge, otherwise known as a ready response, is a line consisting of a "+" character followed by a single space and a BASE64 encoded string. The client answer consists of a line containing a BASE64 encoded string. If the client wishes to cancel an authentication exchange, it should issue a line with a single "*". If the server receives such an answer, it must reject the AUTH command by sending a negative response.

A protection mechanism provides integrity and privacy protection to the protocol session. If a protection mechanism is negotiated, it is applied to all subsequent data sent over the connection. The protection mechanism takes effect immediately following the CRLF that concludes the authentication exchange for the client, and the CRLF of the positive response for the server. Once the protection mechanism is in effect, the stream of command and response octets is processed into buffers of ciphertext. Each buffer is transferred over the connection as a stream of octets prepended with a four octet field in network byte order that represents the length of the following data. The maximum ciphertext buffer length is defined by the protection mechanism.

The server is not required to support any particular authentication mechanism, nor are authentication mechanisms required to support any protection mechanisms. If an AUTH command fails with a negative response, the session remains in the AUTHORIZATION state and client may try another authentication mechanism by issuing another AUTH command, or may attempt to authenticate by using the USER/PASS or APOP commands. In other words, the client may request authentication types in decreasing order of preference, with the USER/PASS or APOP command as a last resort.

Should the client successfully complete the authentication exchange, the POP3 server issues a positive response and the POP3 session enters the TRANSACTION state.

Possible Responses:
+OK maildrop locked and ready
-ERR authentication exchange failed

Examples:

```
S: +OK POP3 server ready
C: AUTH KERBEROS_V4
S: + AmFYig==
C: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4DT
   +nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
   WwuQ1MWiy6IesKvjL5rL9WjXUb9MwT9bpObYLGOKi1Qh
S: + or//EoAADZI=
C: DiAF5A4gA+oOIALuBkAAmw==
S: +OK Kerberos V4 authentication successful
   ...
C: AUTH FOOBAR
S: -ERR Unrecognized authentication type
```

Note: the line breaks in the first client answer are for editorial clarity and are not in real authenticators.

3. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) notation as specified in RFC 822.

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
ATOM_CHAR       ::= <any CHAR except atom_specials>

atom_specials   ::= "(" / ")" / "{" / SPACE / CTLs / "%" / "*" /
                    <"> / "\"

auth            ::= "AUTH" 1*(SPACE / TAB) auth_type *(CRLF base64)
                    CRLF

auth_type       ::= 1*ATOM_CHAR

base64          ::= *(4base64_CHAR) [base64_terminal]

base64_char     ::= "A" / "B" / "C" / "D" / "E" / "F" / "G" / "H" /
        "I" / "J" / "K" / "L" / "M" / "N" / "O" / "P" /
                    "Q" / "R" / "S" / "T" / "U" / "V" / "W" / "X" /
                    "Y" / "Z" /
                    "a" / "b" / "c" / "d" / "e" / "f" / "g" / "h" /
                    "i" / "j" / "k" / "l" / "m" / "n" / "o" / "p" /
                    "q" / "r" / "s" / "t" / "u" / "v" / "w" / "x" /
                    "y" / "z" /
                    "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" /
                    "8" / "9" / "+" / "/"
                    ;; Case-sensitive

base64_terminal ::= (2base64_char "==") / (3base64_char "=")

CHAR            ::= <any 7-bit US-ASCII character except NUL,
                     0x01 - 0x7f>

continue_req    ::= "+" SPACE base64 CRLF

CR              ::= <ASCII CR, carriage return, 0x0C>

CRLF            ::= CR LF

CTL             ::= <any ASCII control character and DEL,
                     0x00 - 0x1f, 0x7f>
```

```
LF              ::= <ASCII LF, line feed, 0x0A>

SPACE           ::= <ASCII SP, space, 0x20>

TAB             ::= <ASCII HT, tab, 0x09>
```

4. References

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

5. Security Considerations

Security issues are discussed throughout this memo.

6. Author's Address

John G. Myers
Carnegie-Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213

EMail: jgm+@cmu.edu