                    SMTP Service Extension
                      for Authentication

Status of this Memo

    This document is an Internet-Draft. Internet-Drafts are working
    documents of the Internet Engineering Task Force (IETF), its areas,
    and its working groups. Note that other groups may also distribute
    working documents as Internet-Drafts.

    Internet-Drafts are draft documents valid for a maximum of six months
    and may be updated, replaced, or obsoleted by other documents at any
    time. It is inappropriate to use Internet-Drafts as reference
    material or to cite them other than as "work in progress."

    To view the entire list of current Internet-Drafts, please check the
    "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow
    Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern
    Europe), ftp.nis.garr.it (Southern Europe), munnari.oz.au (Pacific
    Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

    A revised version of this draft document will be submitted to the RFC
    editor as a Proposed Standard for the Internet Community. Discussion
    and suggestions for improvement are requested. This document will
    expire in six months. Distribution of this draft is unlimited.

1. Introduction

    This document defines an SMTP service extension [ESMTP] whereby an
    SMTP client may indicate an authentication mechanism to the server,
    perform an authentication protocol exchange, and optionally negotiate
    a security layer for subsequent protocol interactions. This
    extension is a profile of the Simple Authentication and Security
    Layer [SASL].

2. Conventions Used in this Document

   In examples, "C:" and "S:" indicate lines sent by the client and
   server respectively.

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

3. The Authentication service extension

   (1) the name of the SMTP service extension is "Authentication"

   (2) the EHLO keyword value associated with this extension is "AUTH"

   (3) The AUTH EHLO keyword contains as a parameter a space separated
       list of the names of supported SASL mechanisms.

   (4) a new SMTP verb "AUTH" is defined

   (5) an optional parameter using the keyword "AUTH" is added to the
       MAIL FROM command, and extends the maximum line length of the
       MAIL FROM command by 500 characters.

   (6) this extension is appropriate for the submission protocol
       [SUBMIT].

4. The AUTH command

   AUTH mechanism [initial-response]

     Arguments:
         a string identifying a SASL authentication mechanism.
         an optional base64-encoded response

     Restrictions:
         After an AUTH command has successfully completed, no more AUTH
         commands may be issued in the same session. After a successful
         AUTH command completes, a server MUST reject any further AUTH
         commands with a 503 reply.

         The AUTH command is not permitted during a mail transaction.

     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 security layer for subsequent protocol
interactions. If the requested authentication mechanism is not
supported, the server rejects the AUTH command with a 504
reply.

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 334 reply with the text part
containing 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 issues a line
with a single "*". If the server receives such an answer, it
MUST reject the AUTH command by sending a 501 reply.

The optional initial-response argument to the AUTH command is
used to save a round trip when using authentication mechanisms
that are defined to send no data in the initial challenge.
When the initial-response argument is used with such a
mechanism, the initial empty challenge is not sent to the
client and the server uses the data in the initial-response
argument as if it were sent in response to the empty challenge.
Unlike a zero-length client answer to a 334 reply, a zero-
length initial response is sent as a single equals sign ("=").
If the client uses an initial-response argument to the AUTH
command with a mechanism that sends data in the initial
challenge, the server rejects the AUTH command with a 535
reply.

If the server cannot BASE64 decode the argument, it rejects the
AUTH command with a 501 reply. If the server rejects the
authentication data, it SHOULD reject the AUTH command with a
535 reply unless a more specific error code, such as one listed
in section 6, is appropriate. Should the client successfully
complete the authentication exchange, the SMTP server issues a
235 reply.

The service name specified by this protocol's profile of SASL
is "smtp".

If a security layer is negotiated through the SASL
authentication exchange, it takes effect immediately following
the CRLF that concludes the authentication exchange for the
client, and the CRLF of the success reply for the server. Upon
a security layer's taking effect, the SMTP protocol is reset to

        the initial state (the state in SMTP after a server issues a
        220 service ready greeting). The server MUST discard any
        knowledge obtained from the client, such as the argument to the
        EHLO command, which was not obtained from the SASL negotiation
        itself. The client MUST discard any knowledge obtained from
        the server, such as the list of SMTP service extensions, which
        was not obtained from the SASL negotiation itself (with the
        exception that a client MAY compare the list of advertised SASL
        mechanisms before and after authentication in order to detect
        an active down-negotiation attack). The client SHOULD send an
        EHLO command as the first command after a successful SASL
        negotiation which results in the enabling of a security layer.

        The server is not required to support any particular
        authentication mechanism, nor are authentication mechanisms
        required to support any security layers. If an AUTH command
        fails, the client may try another authentication mechanism by
        issuing another AUTH command.

        If an AUTH command fails, the server MUST behave the same as if
        the client had not issued the AUTH command.

        The BASE64 string may in general be arbitrarily long. Clients
        and servers MUST be able to support challenges and responses
        that are as long as are generated by the authentication
        mechanisms they support, independent of any line length
        limitations the client or server may have in other parts of its
        protocol implementation.

   Examples:
        S: 220 smtp.example.com ESMTP server ready
        C: EHLO jgm.example.com
        S: 250-smtp.example.com
        S: 250 AUTH CRAM-MD5 DIGEST-MD5
        C: AUTH FOOBAR
        S: 504 Unrecognized authentication type.
        C: AUTH CRAM-MD5
        S: 334
        PENCeUxFREJoU0NnbmhNWitOMjNGNndAZWx3b29kLmlubm9zb2Z0LmNvbT4=
        C: ZnJlZCA5ZTk1YWVlMDljNDBhZjJiODRhMGMyYjNiYmFlNzg2ZQ==
        S: 235 Authentication successful.

5. The AUTH parameter to the MAIL FROM command

    AUTH=addr-spec

    Arguments:
        An addr-spec containing the identity which submitted the message
        to the delivery system, or the two character sequence "<>"
        indicating such an identity is unknown or insufficiently
        authenticated. To comply with the restrictions imposed on ESMTP
        parameters, the addr-spec is encoded inside an xtext. The syntax
        of an xtext is described in section 5 of [ESMTP-DSN].

    Discussion:
        The optional AUTH parameter to the MAIL FROM command allows
        cooperating agents in a trusted environment to communicate the
        authentication of individual messages.

        If the server trusts the authenticated identity of the client to
        assert that the message was originally submitted by the supplied
        addr-spec, then the server SHOULD supply the same addr-spec in an
        AUTH parameter when relaying the message to any server which
        supports the AUTH extension.

        A MAIL FROM parameter of AUTH=<> indicates that the original
        submitter of the message is not known. The server MUST NOT treat
        the message as having been originally submitted by the client.

        If the AUTH parameter to the MAIL FROM is not supplied, the
        client has authenticated, and the server believes the message is
        an original submission by the client, the server MAY supply the
        client's identity in the addr-spec in an AUTH parameter when
        relaying the message to any server which supports the AUTH
        extension.

        If the server does not sufficiently trust the authenticated
        identity of the client, or if the client is not authenticated,
        then the server MUST behave as if the AUTH=<> parameter was
        supplied. The server MAY, however, write the value of the AUTH
        parameter to a log file.

        If an AUTH=<> parameter was supplied, either explicitly or due to
        the requirement in the previous paragraph, then the server MUST
        supply the AUTH=<> parameter when relaying the message to any
        server which it has authenticated to using the AUTH extension.

        A server MAY treat expansion of a mailing list as a new
        submission, setting the AUTH parameter to the mailing list
        address or mailing list administration address when relaying the

message to list subscribers.

It is conforming for an implementation to be hard-coded to treat
all clients as being insufficiently trusted. In that case, the
implementation does nothing more than parse and discard
syntactically valid AUTH parameters to the MAIL FROM command and
supply AUTH=<> parameters to any servers to which it
authenticates using the AUTH extension.

Examples:
    C: MAIL FROM:<e=mc2@example.com> AUTH=e+3Dmc2@example.com
    S: 250 OK

6. Error Codes

The following error codes may be used to indicate various conditions
as described.

432 A password transition is needed

This response to the AUTH command indicates that the user needs to
transition to the selected authentication mechanism. This typically
done by authenticating once using the PLAIN authentication mechanism.

534 Authentication mechanism is too weak

This response to the AUTH command indicates that the selected
authentication mechanism is weaker than server policy permits for
that user.

538 Encryption required for requested authentication mechanism

This response to the AUTH command indicates that the selected
authentication mechanism may only be used when the underlying SMTP
connection is encrypted.

454 Temporary authentication failure

This response to the AUTH command indicates that the authentication
failed due to a temporary server failure.

530 Authentication required

This response may be returned by any command other than AUTH, EHLO,
HELO, NOOP, RSET, or QUIT. It indicates that server policy requires
authentication in order to perform the requested action.

[7](). Formal Syntax

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

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.

```
UPALPHA         = %x41-5A               ;; Uppercase: A-Z

LOALPHA         = %x61-7A               ;; Lowercase: a-z

ALPHA           = UPALPHA / LOALPHA  ;; case insensitive

DIGIT           = %x30-39               ;; Digits 0-9

HEXDIGIT        = %x41-46 / DIGIT     ;; hexidecimal digit (uppercase)

hexchar         = "+" HEXDIGIT HEXDIGIT

xchar           = %x21-2A / %x2C-3C / %x3E-7E
                    ;; US-ASCII except for "+", "=", SPACE and CTL

xtext           = *(xchar / hexchar)

AUTH_CHAR       = ALPHA / DIGIT / "-" / "_"

auth_type       = 1*20AUTH_CHAR

auth_command    = "AUTH" SPACE auth_type [SPACE (base64 / "=")]
                    *(CRLF [base64]) CRLF

auth_param      = "AUTH=" xtext
                      ;; The decoded form of the xtext MUST be either
                      ;; an addr-spec or the two characters "<>"

base64          = base64_terminal /
                    ( 1*(4base64_CHAR) [base64_terminal] )

base64_char     = UPALPHA / LOALPHA / DIGIT / "+" / "/"
                    ;; Case-sensitive

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

continue_req    = "334" SPACE [base64] CRLF
```

```
CR              = %x0C           ;; ASCII CR, carriage return

CRLF            = CR LF

CTL             = %x00-1F / %x7F ;; any ASCII control character and DEL

LF              = %x0A           ;; ASCII LF, line feed

SPACE           = %x20           ;; ASCII SP, space
```

8. References

[ABNF] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium, Demon Internet Ltd, November 1997.

[CRAM-MD5] Klensin, J. et al, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", RFC 2195, September 1997.

[ESMTP] Klensin et al, "SMTP Service Extensions", RFC 1869, November 1995.

[ESMTP-DSN] Moore, K, "SMTP Service Extension for Delivery Status Notifications", RFC 1891.

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

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

[SUBMIT] Gellens, R., Klensin, J., "Message Submission", draft-gellens-submit-13.txt, October 1998

[RFC821] Postel, J., "Simple Mail Transfer Protocol", RFC 821, August 1982.

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", RFC 822, August 1982.

9. **Security Considerations**

    Security issues are discussed throughout this memo.

    If a client uses this extension to get an encrypted tunnel through an
    insecure network to a cooperating server, it needs to be configured
    to never send mail to that server when the connection is not mutually
    authenticated and encrypted. Otherwise, an attacker could steal the
    client's mail by hijacking the SMTP connection and either pretending
    the server does not support the Authentication extension or causing
    all AUTH commands to fail.

    Before the SASL negotiation has begun, any protocol interactions are
    performed in the clear and may be modified by an active attacker.
    For this reason, clients and servers MUST discard any knowledge
    obtained prior to the start of the SASL negotiation upon completion
    of a SASL negotiation which results in a security layer.

    This mechanism does not protect the TCP port, so an active attacker
    may redirect a relay connection attempt to the submission port
    [SUBMIT]. The AUTH=<> parameter prevents such an attack from causing
    an relayed message without an envelope authentication to pick up the
    authentication of the relay client.

    A message submission client may require the user to authenticate
    whenever a suitable SASL mechanism is advertised. Therefore, it may
    not be desirable for a submission server [SUBMIT] to advertise a SASL
    mechanism when use of that mechanism grants the client no benefits
    over anonymous submission.

    This extension is not intended to replace or be used instead of end-
    to-end message signature and encryption systems such as S/MIME or
    PGP. This extension addresses a different problem than end-to-end
    systems; it has the following key differences:

      (1) it is generally useful only within a trusted enclave

      (2) it protects the entire envelope of a message, not just the
          message's body.

      (3) it authenticates the message submission, not authorship of the
          message content

      (4) it can give the sender some assurance the message was
          delivered to the next hop in the case where the sender
          mutually authenticates with the next hop and negotiates an
          appropriate security layer.

Additional security considerations are mentioned in the SASL
specification [SASL].

10. Author's Address:

John Gardiner Myers
Netscape Communications
501 East Middlefield Road
Mail Stop MV-029
Mountain View, CA 94043

Email: jgmyers@netscape.com