INTERNET-DRAFT Viruthagiri Thirumavalavan
Intended Status: Proposed Standard Dombox, Inc.
Expires: Jun 04, 2020 Dec 04, 2019

SMTP Extension for Longer Email Address
draft-viruthagiri-email-address-length-01

Abstract

This memo defines an SMTP extension with keyword "EAML" whereby an SMTP server can declare that it is capable of handling longer email addresses without any local-part or domain-part restriction set by RFC 5321.

EAML stands for "Email Address Maximum Length".

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

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."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Definitions

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

The term "character" in this document refers to an "octet character".

2 Background

[RFC5321], Section 4.5.3.1.3 says:

```
+-------------------------------------------------------------------+
|                                                                   |
|   The maximum total length of a reverse-path or forward-path is   |
|   256 octets (including the punctuation and element separators).  |
|                                                                   |
+-------------------------------------------------------------------+
```

[RFC5321], Section 4.1.2 defines the path as:

```
+-------------------------------------------------------------------+
|                                                                   |
|               Path = "<" [ A-d-l ":" ] Mailbox ">"                |
|                                                                   |
+-------------------------------------------------------------------+
```

So the path will contain at least a pair of angle brackets in addition to the Mailbox. This limits the Mailbox (i.e. the email address) to 254 characters.

[RFC5321], Section 4.5.3.1 says:

```
+-------------------------------------------------------------------+
|                                                                   |
|  To the maximum extent possible, implementation techniques that   |
|  impose no limits on the length of these objects should be used.  |
|                                                                   |
+-------------------------------------------------------------------+
```

From a receiving server perspective, 254 maximum characters limitation is a "SOFT LIMIT" as per RFC 5321. i.e. A receiving server MUST support AT LEAST 254 maximum character email address.

While 254 characters is enough for RCPT TO email addresses, it's not enough for MAIL FROM addresses due to the widespread usage of Variable Envelope Return Path (VERP).

Bulk mailing systems use VERP for automatic bounce handling. For example, an end user subscribes to a list with 254 character email address which is a valid email address, but the rewritten VERP address would go beyond 254 characters.

In most cases, VERP addresses would not go beyond 254 characters. However local-part of VERP addresses would easily go beyond 64 characters. So this memo's primary concern is local-part 64 character limitation.

There is no standard way for a client to know whether a receiving server sticks to the 64 characters local-part limit or not.

Also there is no standard way for a client to know whether a receiving server supports email addresses that go beyond 254 characters. e.g. Gmail supports 900 character email addresses.

To address such issues, this memo defines an SMTP extension with keyword "EAML". The framework is defined in Section 5.

3. Relaxing the local-part limitation

[RFC5321], Section 4.5.3.1.1 says:

```
+-------------------------------------------------------------------+
|                                                                   |
|  The maximum total length of a user name or other local-part      |
|  is 64 octets.                                                    |
|                                                                   |
+-------------------------------------------------------------------+
```

Variable Envelope Return Path (VERP) and Sender Rewriting Scheme (SRS) are the two perfect examples where 64 character local-part limit actually causes issues. VERP and SRS are explained in Appendix A. and Appendix B. respectively.

Both VERP and SRS are designed to handle bounces. So the primary motivation for relaxing 64 character local-part limitation is to have better bounce handling.

A bounce message or just "bounce" is an automated message from an email system, informing the sender that the message had not been delivered (or some other delivery problem occurred). The original message is said to have "bounced". More formal terms for bounce message include "Non-Delivery Report" or "Non-Delivery Receipt" (NDR), "Delivery Status Notification" (DSN) message, or a "Non-Delivery Notification" (NDN).

A "bounce address" is also known as MAIL FROM, Return Path, Reverse Path, Envelope From, Envelope Sender etc.

This memo relaxes the local-part limitation explicitly with the help of EAML extension.

If overall email address length can be defined as "n", then the maximum total length of local-part is "n-2" characters. We allocated 1 character for the "@" symbol and 1 character for the minimum possible domain-part length. Note: We are using user@host address type for calculating this number. Refer Appendix C. for more info.

4. Relaxing the domain-part limitation

[RFC5321], Section 4.5.3.1.2 says:

```
+--------------------------------------------------------------------+
|                                                                    |
|  The maximum total length of a domain name or number is 255        |
|  octets.                                                           |
|                                                                    |
+--------------------------------------------------------------------+
```

Since the maximum total length of an email address is 254 characters, the domain-part limitation of 255 characters is flawed and redundant.

It's flawed because, any such limitation should be less than overall email address length.

i.e. domain-part length < overall email address length

It's redundant because, any email address that has 255 domain characters already an invalid email address due to overall email address length 254 characters. [We are assuming the server sticks to the default limit]

This memo relaxes the domain-part limitation explicitly with the help of EAML extension.

If overall email address length can be defined as "n", then the maximum total length of a domain name is "n-2" characters. We allocated 1 character for the "@" symbol and 1 character for the minimum possible local-part length.

Just to be clear,

[RFC1034], Section 3.5 says, A domain label must be 63 characters or

less. For example, www.example.com contains 3 labels. i.e. www, example and com.

This memo relaxes only the domain-part soft limitation set by [RFC5321]. Not the domain label hard limitation set by [RFC1034].

5. Framework for the EAML Extension

The following service extension is defined:

(1) The name of the SMTP service extension is "Email Address Maximum Length";

(2) The EHLO keyword value associated with this extension is "EAML";

(3) One OPTIONAL parameter is allowed with this EHLO keyword value, a three digit decimal number indicating the maximum email address length in octets that the server will accept.

(4) [RFC5322] header field data is still acceptable to 1000 characters. i.e. 998+EOL. where EOL refers to the control codes <CR><LF>. Carriage Return and Line Feed. In order to avoid hitting that limit and give some room for header keys, EAML parameter value MUST NOT be greater than 900. i.e. 900 octets is the maximum allowed email address length.

(5) The parameter value SHOULD be from 254-900 (inclusive).

```
S: 220 smtp.example.com ESMTP Ready
C: EHLO bob.example.org
S: 250-smtp.example.com
S: 250-EXPN
S: 250-EAML 500
S: 250-PIPELINING
S: 250 HELP
```

(6) The syntax of the parameter is as follows, using the ABNF notation of [RFC5322]:

```
eaml-param ::= [ 3DIGIT ]
```

(7) If the parameter is omitted or the parameter value is 0-253 (inclusive) or the value is greater than 900, then the value MUST be treated as 254.

(8) Servers offering this extension MUST remove the 64 characters local-part limit. That effectively means, maximum characters allowed in local-part is n-2. Where n is the EAML parameter

value.

(9) Servers offering this extension MUST remove the 255 characters domain-part limit. That effectively means, maximum characters allowed in domain-part is n-2. Where n is the EAML parameter value.

(10) The parameter value is OPTIONAL.

```
S: 220 smtp.example.com ESMTP Ready
C: EHLO bob.example.org
S: 250-smtp.example.com
S: 250-EXPN
S: 250-EAML
S: 250-PIPELINING
S: 250 HELP
```

(11) The parameter omitted EAML keyword says:

[*] No local-part limitation.
[*] No domain-part limitation.
[*] 254 maximum characters

(12) No additional SMTP verbs are defined by this extension.

6. IANA Considerations

IANA is hereby requested to register the EAML extension.

7. Security Considerations

This RFC does not discuss security issues and is not believed to raise any security issues not already endemic in electronic mail and present in fully conforming implementations of SMTP.

8. References

8.1. Normative References

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <http://www.rfc-editor.org/info/rfc5321>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <http://www.rfc-editor.org/info/rfc5322>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <http://www.rfc-editor.org/info/rfc1034>.

[RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, DOI 10.17487/RFC3464, January 2003, <https://www.rfc-editor.org/info/rfc3464>.

[VERWIKI] Wikipedia, "Variable envelope return path", December 2019, <http://en.wikipedia.org/w/index.php?title=Variable_envelope_return_path&oldid=902866095>.

[SRSWIKI] Wikipedia, "Sender Rewriting Scheme", December 2019, <http://en.wikipedia.org/w/index.php?title=Sender_Rewriting_Scheme&oldid=920076660>.

[VERP] Bernstein, D., "Variable Envelope Return Paths", February 1997, <http://cr.yp.to/proto/verp.txt>.

[SRS] Shevek, "The Sender Rewriting Scheme", June 2004, <https://www.libsrs2.org/srs/srs.pdf>.

Appendix A. Variable Envelope Return Path (VERP)

Most bounce messages have historically been designed to be read by human users, not automatically handled by software. They all convey the same basic idea (the message from X to Y could not be delivered because of reason Z) but with so many variations that it would be nearly impossible to write a program to reliably interpret the meaning of every bounce message. [RFC3464] defines a standard format to fix this problem, but support for the standard is far from universal.

VERP solves the bounce handling problem.

The hard part of bounce handling is matching up a bounce message with the undeliverable address that caused the bounce. If the mailing list

software can see that a bounce resulted from an attempt to send a message to user@example.com then it doesn't need to understand the rest of the information in the bounce. It can simply count how many messages were recently sent to user@example.com, and how many bounces resulted, and if the proportion of bounced messages is too high, the address is removed from the list.

While bounce message formats in general vary wildly, there is one aspect of a bounce message that is highly predictable: the address to which it will be sent. VERP takes full advantage of this. In a mailing list that uses VERP, a different MAIL FROM address is used for each recipient.

The mailing list manager knows that it sent a message from X to Y, so if a bounce message is received at address X, it can only be because address Y was undeliverable, because nothing was sent from X to any other address. Thus the important information has been extracted from the bounce message, without any need to understand its contents, which means the person in charge of the list does not need to deal with it manually.

Typically, an email from Alice to Bob in the above example will have headers like the following:

```
+-------------------------------------------------------------------+
|                                                                   |
|    From: alice@example.org                                        |
|    To: bob@example.com                                            |
|    Return-Path: bounces@example.org                               |
|                                                                   |
+-------------------------------------------------------------------+
```

A very simple VERP address for a mail to bob@example.com will be:

```
+-------------------------------------------------------------------+
|                                                                   |
|    MAIL FROM:<bounces-bob=example.com@example.org>                |
|                                                                   |
+-------------------------------------------------------------------+
```

bob is a 3 character local-part, so there won't be any issues.

Let's just assume a user with 62 character local-part subscribe to the list.

A normal mail would look like this.

```
+-------------------------------------------------------------------+
```

```
   |                                                                    |
   |    From: alice@mailchimp.com                                       |
   |    To: this-electronic-mail-address-local-part-                    |
   |          contains-62-characters@example.com                        |
   |    Return-Path: bounces@example.org                                |
   |                                                                    |
   +--------------------------------------------------------------------+
```

The [VERP] address would look like:

```
   +--------------------------------------------------------------------+
   |                                                                    |
   |     MAIL FROM:<bounces-this-electronic-mail-address-local-part-    |
   |                contains-62-characters=example.com@example.org>     |
   |                                                                    |
   +--------------------------------------------------------------------+
```

Above VERP address local-part contains 82 characters. If example.com sticks to the RFC 5321 local-part limit 64 characters, then the above VERP address structure won't work.

[VERP] was introduced by Daniel J. Bernstein in 1997 and today many software supports VERP.

To name a few,

AmazonSES, CiviCRM, Courier Mail Server, Discourse, exim, ezmlm, GNU Mailman, Inxmail, Mercury Mail Transport System, mlmmj, Mahara, Mailchimp, MediaWiki, Moodle, postfix, qmail, Sendmail, STEdb, StrongMail, Sympa, Thexyz, Zimbra, Target Box, NotifyBC.

See Wikipedia's page [VERWIKI] to learn more about VERP.

Appendix B. Sender Rewriting Scheme (SRS)

Sender Rewriting Scheme ([SRS]) is a scheme for rewriting the envelope sender address of an email message, in view of remailing it. In this context, remailing is a kind of email forwarding. SRS was devised in order to forward email without breaking the Sender Policy Framework (SPF), back in 2003.

[SRS] is a form of variable envelope return path ([VERP]) inasmuch as it encodes the original envelope sender in the local part of the rewritten address. Consider example.com forwarding a message originally destined to bob@example.com to his new address <bob@example.net>

```
+--------------------------------------------------------------------+
|                                                                    |
|    ORIGINAL                                                        |
|    envelope sender:    alice@example.org                           |
|    envelope recipient: bob@example.com                             |
|                                                                    |
|    REWRITTEN                                                       |
|    envelope sender:    SRS0=HHH=TT=example.org=alice@example.com   |
|    envelope recipient: bob@example.net                             |
|                                                                    |
+--------------------------------------------------------------------+
```

With respect to VERP, the local part (alice) is moved after her domain name (example.org), further adding a prefix (SRS0), a hash (HHH), and a timestamp (TT)

SRS provides for another prefix, SRS1, to be used for rewriting an already rewritten address, in a multi-hop scenario. If example.net has to forward the message in turn, it can spare adding another timestamp and repeating the original local part (alice). That is, each new forwarder adds just its own hash (HHH) and the domain name of the preceding forwarder:

```
+--------------------------------------------------------------------+
|                                                                    |
|    FURTHER REWRITTEN                                               |
|    envelope sender:    SRS1=HHH=example.com==HHH=TT=example.org    |
|                        =alice@example.net                          |
|    envelope recipient: bob@further.example                         |
|                                                                    |
+--------------------------------------------------------------------+
```

SRS1 incorporates 2 domains in the local-part. So 64 characters are not enough.

See Wikipedia's page [SRSWIKI] to learn more about SRS.

Appendix C. Email Address Types

INFORMATIONAL:

The term "email address" is a generic term. It can refer to an Internet address, Intranet address, X.400 address etc.

Internet Address:

An email address on the Internet would look like a@b.c
e.g. john@example.com [user@domain]

Intranet Address:

An email address on the Intranet may look like a@b
e.g. alice@machine50 [user@host]

X.400 Address:

An X.400 address is technically referred to as an Originator/Recipient (OR) address. It consists of several elements, including:

C	(Country name)
ADMD	(Administration Management Domain, short-form A), usually a public mail service provider
PRMD	(Private Management Domain, short-form P)
O	(Organization name)
OU	(Organizational Unit Names), OU is equivalent to OU0, can have OU1, OU2...
G	(Given name)
I	(Initials)
S	(Surname)

e.g. G=Harald;S=Alvestrand;O=Uninett;PRMD=Uninett;A=;C=no

IP address Literal:

jsmith@[192.168.2.1]
jsmith@[IPv6:2001:db8::1]

Authors' Addresses

Viruthagiri Thirumavalavan
Dombox, Inc.

EMail: giri@dombox.org