Network Working Group   A. Hathcock
Internet-Draft   J. Merkel
Intended Status: Informational   Alt-N Technologies
Expires: January 9, 2009   July 9, 2008

The Minger Email Address Verification Protocol
draft-hathcock-minger-05.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the Minger protocol. Minger is a protocol which allows a mail handling entity to query a remote service and ask the question "do you accept mail for this email address?" It includes security in the form of a hashed shared secret but can also be used anonymously if desired.

Requirements Language

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

Table of Contents

1. Introduction 3
1.1. The problem 3
1.2. Existing solutions 3
1.3. The Minger solution 4
2. The Minger protocol 4
2.1 The Minger query process 4
2.2 Description of query elements 5
3. Minger responses 5
3.1 Description of response elements 5
3.2 Example responses 6
4. Anonymous mode 6
5. Security Considerations 7
6. IANA Considerations 7
7. Informative References 7
Appendix A. Acknowledgements 8
Authors' Addresses 8
Intellectual Property and Copyright Statements 9

1. Introduction

1.1 The problem

It is common for elements within a typical email handling topology to be unaware of whether individual local-parts are valid for the mail it accepts. For example, so-called "edge" servers which provide security oriented services for downstream mail handling elements often do not have an exhaustive listing of all valid local-parts for a given domain. Thus, they are sometimes forced to accept and process messages which might otherwise be rejected as "user unknown". Similarly, entities offering "backup MX" mail services are rarely privy to a complete local-part listing and are therefore often decide to accept messages which might otherwise be rejected. Finally, even within a common administrative framework of several locally maintained and controlled SMTP servers in a load balanced configuration, it is not always possible for all servers to access a common local-part database.

1.2 Existing solutions

The need to determine whether an email address contains a valid local part has lead to the use of at least two existing mechanisms - Finger [RFC1288] and SMTP "call-forward".

Finger [RFC1288] describes a protocol for the exchange of user information. In theory, Finger could be used to determine whether an account exists by careful examination of the results of a Finger query. However, Finger suffers from a lack of security which makes its modern day use problematic when coupled with the user level detail it can provide. Also, Finger requires the use of TCP rather than UDP which seems ill suited to a simple verification scheme.

SMTP "call-forward" is a term used to describe a widespread practice whereby SMTP servers place an incoming SMTP session on hold while they attempt to use an outbound SMTP session to determine whether or not a given email address is valid. The theory behind this is as follows: if an SMTP server responds positively to an SMTP RCPT command [RFC2821] with a given email address then this potentially means that the address local-part is valid. One problem with such a scheme is the lack of efficiency inherent in the need to tear-up and tear-down an SMTP session over TCP. Also, because these types of SMTP sessions are not purposed to deliver mail, they typically drop connection after the RCPT command is processed. This leads to a large number of SMTP

sessions which appear in logs to have simply failed for no reason. This leads to situations in which SMTP transaction logs can no longer distinguish legitimate network errors from "call-forward" traffic.

SMTP includes a VRFY command which can be used to determine whether an email address exists. VRFY is not in wide-spread use and suffers from the same inefficiency concerns described in the discussion on SMTP "call-forward". Additionally, SMTP agents providing mail services to a domain are often not authoritative making VRFY requests potentially unreliable.

LDAP could be used to determine whether an email address exists. However, LDAP is overly complex to configure and maintain.

1.3 The Minger solution

What's needed is a protocol which is secure, has little overhead, and can be easily invoked to determine whether a given email address is valid or not. Minger achieves these goals using a shared secret for security and UDP to lower overhead.

2. The Minger protocol

Minger is a UDP protocol that operates on port 4069.

Syntax descriptions use the form described in Augmented Backus-Naur Form for Syntax Specifications (ABNF) [RFC4234].

2.1 The Minger query process

A Minger client constructs a query string as described below and transmits it over UDP to a Minger server. The format of the query is as follows:

```
  ABNF:

   query-string = mailbox [SP %x64 "=" digest] [SP tag-list]

   digest = base64                    ; digest for security
                                      ; base64 defined in [RFC5034]

   digest-text = shared-secret ":" mailbox ; input text for digest

   mailbox = Local-part "@" Domain    ; as defined in [RFC2821]

   shared-secret = 1*50(VCHAR)        ; password credential
```

```
   tag-list = tag-spec *(SP tag-spec) ; tag/value list

   tag-spec = tag-name "=" tag-value

   tag-name = 1 * ( ALPHA / DIGIT / "_") ; except 'd'

   tag-value = 1 * (ALPHA / DIGIT / "_")
```

2.2 Description of query elements

mailbox

This is the email address for which verification of existence is desired.

digest

This is the base64 encoding of the MD5 [RFC1321] hash of digest-text. Digest-text is constructed, the MD5 hash of that is computed, and that result is base64 encoded.

tag-list

Tag-list is provided so that future capability might be added in an easy way. Tag-names are case-sensitive and MUST NOT be used more than once.

3. Minger responses

Minger servers return a response string of the following form:

ABNF:

```
   response-string = mailbox status

   mailbox = Local-part "@" Domain   ; as defined in [RFC2821]

   status = %x30-35                  ; single digit result code
                                     ; from 0 - 5
```

3.1 Description of response elements

mailbox

This is the email address for which verification of existence is desired.

status

The following status codes are defined:

```
0 - invalid request (for example, malformed query string)
1 - access denied (for example, query from unauthorized IP)
2 - bad or missing credentials (returned when anonymous
    mode is disabled and no credentials were provided in the
    query string or when the credentials themselves are
    invalid)
3 - email address does not exist
4 - email address exists but can not receive mail (for example,
    the account associated with the email address has exceeded
    local storage constraints or it is otherwise disabled due
    to local policy)
5 - email address exists and is active (able to receive mail)
```

3.2 Example responses

Minger response returned when the queried email address does not exist:

```
arvel@example.com 3
```

Minger response returned for invalid credentials:

```
arvel@example.com 2
```

Minger response returned when the queried email address exists:

```
arvel@example.com 5
```

4. Anonymous mode

Minger clients MAY attempt anonymous queries; that is, queries which do not contain a shared secret digest within the query string. Minger servers MAY be configured to refuse anonymous queries. If so, they MUST respond with a status of "2".

5. Security Considerations

Minger is a protocol which is used to determine whether a given email address is valid or not. If a particular email infrastructure does not wish to advertise the email addresses that it services then this protocol should not be employed.

If a shared secret is employed to secure Minger from anonymous use that shared secret should be at least 128 bits.

6. IANA Considerations

IANA has assigned tcp & upd port 4069 for Minger.

7. Informative References

[RFC1288] Zimmerman, D., "The Finger User Information Protocol", RFC 1288, December 1991.

[RFC1734] Myers, J., "POP3 Authentication Command", RFC 1734, December 1994.

[RFC2821] Klensin, J., Editor, "Simple Mail Transfer Protocol", RFC 2821, March 2001.

[RFC4234] Crocker, D., Ed. And P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC1321] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992.

Appendix A. Acknowledgements

We wish to thank the members of the MDaemon Beta Community (md-beta-subscribe@altn.com) for their ideas and help and Paul Hoffman for his valuable feedback.

Authors' Addresses

Arvel Hathcock
Alt-N Technologies
http://www.altn.com

Email: arvel.hathcock@altn.com

Jonathan Merkel
Alt-N Technologies
http://www.altn.com

Email: jon.merkel@altn.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).