Network Working Group
Internet-Draft
Updates: 5231, 6409 (if approved)
Intended status: Standards Track
Expires: January 7, 2016

K. Moore
Network Heretics
C. Newman
Oracle

July 6, 2015

SMTP and SUBMISSION Service Extensions For Address Query
draft-moore-email-addrquery-00.txt

Abstract

This document defines several mechanisms which can be used by a client such as a Mail User Agent or Mail Submission Agent, to query an SMTP server which is configured to accept incoming mail for a mail domain, to obtain information associated with an email address based in that domain. Among other purposes, these mechanisms are intended to facilitate discovery of senders' and/or recipients' public keys for use in automatic verification of whole-message digital signatures and automatic whole-message encryption of email sent to recipients.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 7, 2016.

Copyright Notice

Table of Contents

1. Introduction

At least since the introduction of MIME [RFC1321] there has been a desire to allow message senders to discover capabilities of email recipients, so that senders could avoid sending message contents to recipients who were unable to make use of such contents. Similarly, deployment of per-message encryption (e.g. PEM [RFC1113], S/MIME [RFC5751], and OpenPGP [RFC4880]) has long been hampered for lack of a standard and widely supported means to discover and verify authenticity of senders' and recipients' public key(s).

The issue surfaced recently as part of the DANE working group discussion in Dallas, and specifically in an effort to adapt TLSA DNS

records [RFC6698] for use in discovery of email recipients' public keys. The problem there was that there's no clean way to map recipient email addresses onto DNS labels, because the interpretation of a local-part of an email address is entirely left to the SMTP server(s) that accept incoming mail for that address's mail domain,

and different mail domains have configured their SMTP servers to interpret their email addresses in different ways. The "local parts" of email addresses may be case-sensitive or case-insensitive, subaddresses may be allowed, there may be some sort of fuzzy matching, an address may be forwarded elsewhere, and so on. Also, having public keys for email recipients advertised in DNS would have facilitated email traffic analysis by an observer watching DNS queries and responses in cleartext.

Since the knowledge of how to interpret an email address is inherently embedded in the code and configuration of the SMTP servers that accept incoming mail for that address's email domain, it appears that the best way to advertise public keys and other information associated with email addresses is to do so using the same SMTP servers that accept such incoming mail. That way, the logic that maps from address to associated information will be the same logic that maps from recipient address to recipient mailbox (or forwarding address). A separate lookup service could be used, but this would introduce a high probablility that the service would interpret the address differently than that mail domain's SMTP servers, if for no other reason than configuration errors. However as a compromise for large mail service providers, and especially those that serve large numbers of mail domains, the proposed SMTP extension also includes a "redirect" mechanism that can be used to refer a client to a separate service which then provides the requested information. Finally, this document defines an extension to the SUBMISSION service which allows that service to perform an address information lookup operation on behalf of its authenticated client, which can be useful to circumvent the common practice of blocking outbound port 25 traffic.

2. Conventions and Terminology Used In This Document

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].

This specification expresses syntax using the Augmented Backus-Naur Form (ABNF) as described in [RFC5234], including the core rules in Appendix B and rules from [RFC5322].

In examples illustrating protocol interactions, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

3. SMTP Service Extension for Address Query

This section defines a service extension to the Simple Message Transfer Protocol (SMTP) [RFC5321] which can be used by a client to query the server for information about an email address for which the server accepts incoming mail.

- The Name of this extension is "Address Query".
- Servers implementing this extension advertise an additional EHLO keyword of "ADDRQUERY", which has no associated parameters.
- This extension introduces one new SMTP command, AQRY, described below.
- This extension does not alter any existing SMTP commands, nor does this extension change the minimum line length that an implementation of SMTP including this extension must support.

3.1. AQRY SMTP Command

The AQRY SMTP command is used to query an SMTP server about an address containing a domain name for which the server is configured to act as a mail exchanger, i.e. to accept incoming mail for delivery. A SMTP server which accepts incoming mail for a domain is in a unique position to interpret email addresses contianing that domain, since only such a server can reliably know whether the local part of that email address is case-sensitive (i.e. whether Joe@example.com and joe@example.com are distinct users), whether

subaddressing applies to that domain (e.g. whether joe+xyz@example.com refers to the same user as joe@example.com), whether a particular recipient has mail forwarded, and so on. Therefore an SMTP server MUST reject an AQRY command which contains an address for which the server is not explicitly configured to accept incoming mail.

In addition, to ensure the integrity of the information provided to the client and to deter both passive and active attacks, any SMTP server supporting ADDRQUERY MUST also support the STARTTLS service extension, and MUST reject any AQRY command not appearing in a TLS-protected session. Clients using the AQRY command MUST support the TLS Server Name Indication (SNI) [RFC6066] extension, and MUST supply the host name of the server to which they wish to connect in the ServerNameList portion of the extension_data field of the extended client hello message. (This requirement also applies to SUBMISSION servers that implement the Address Query Proxy extension.) This host name will either be the target of the MX record associated with the address being queried, or the "host" field as obtained from an AQRY

or AQPX redirect response as defined below. Servers supporting the Address Query extension SHOULD support SNI and use it to provide an appropriate server certificate, if available.

The syntax of the AQRY command is as follows:

```
aqry = "AQRY" SP "<" Mailbox ">"
       [ "RRVS=" date-time ] [ "COOKIE=" Atom ] CRLF
```

where Mailbox is as defined in [RFC5321], and date-time is as defined in [RFC3339], with the added restriction that a "time-secfrac" MUST NOT be used.

The AQRY command requests that the SMTP server return public information about the email address ("Mailbox") specified in the command. If the optional RRVS parameter is included, it specifies that the email address must have been valid at least since that date and time. If the server knows that the address has not been valid that long, it MUST return either an error, or a redirect to a server that will return an "address not found" error.
(Note: Although the RRVS parameter to the AQRY command has the same syntax as the RRVS parameter to the RCPT command as defined in

[RFC7293], the two are separate and have different purposes. An SMTP server MAY support the Address Query extension even if it does not support the RRVS extension.)

The COOKIE parameter is used only in redirects, as described below.

3.1.1. Client Use of AQRY command

Clients wishing to query for email address information MUST first perform a DNS [RFC1035] lookup with query type of MX, specifying the domain name that appears in the email address. The selection of SMTP servers among those returned from the DNS query follows the same algorithm used for selection of SMTP servers to be used for forwarding mail [RFC5321]: servers with lower MX precedence values are queried before servers with higher MX precedence values.

Clients MUST NOT send an AQRY command to a server that isn't listed in DNS as a mail exchanger for the mail domain of the address to be queried. (Exception: a client MAY send an AQRY command to an arbitrary SMTP server without first obtaining that from a DNS MX lookup, if this is done specifically and entirely for the purpose of fault diagnosis or configuration checking and the results are not used to encrypt email nor validate a digital signature.)

Clients wishing to use AQRY MUST first negotiate use of TLS encryption using the STARTTLS command [RFC3207]. If the server does

not advertise STARTTLS, or the TLS negotiation fails, the client MUST NOT attempt to use AQRY. Furthermore, the client MUST NOT attempt to use AQRY before first establishing the identity of the server using the server's certificate, and in particular, that the server's TLS certificate contains a subjectAltName of dNSName type [RFC5280] matches either the DNS name that is the target of the MX record, or the DNS name appearing in the email address for which information is being requested. (Exception: the check of the TLS certificate MAY be skipped if the AQRY operation is done specifically and entirely for the purpose of fault diagnosis or configuration checking, and the results are not used to encrypt email nor validate a digital signature.)

In response to an AQRY command, the server MUST return one of: a normal response, a redirect response, or an error response.

A normal response contains information about the email address for which the request was issued which is specific to that email address, and/or information about the mail domain name which appears in that email address. A normal response MAY also contain information such as address(es) to which incoming mail will be forwarded. In some such cases the client will need to perform additional AQRY operations, perhaps of other SMTP servers serving other domains, in order to learn information about the addresses that would eventually receive mail sent to the originally queried address.

A redirect response does not contain information about the requested email address, but does contain one or more URLs which may then be queried to learn about that address and/or its mail domain.

3.1.2. Normal AQRY Response

The normal (non-redirect, non-error) response to a valid AQRY command consists of multiple lines. Each line but the last line of the response begins with "212-". The remainder of each line beginning with "212-" consists of JSON text [RFC7159] subsequently encoded in BASE64 format as defined in [RFC2045]. BASE64 is used to avoid the need for the server to produce JSON text which conforms to SMTP line-length restrictions.

A normal response is not an indication that the address supplied in the AQRY command is valid. An implementation that does not wish to disclose whether recipients are valid MAY return "fake" information in response to AQRY requests for nonexistent recipients. However the implementation MUST NOT return "fake" information for valid recipients.

The data structure encoded in the JSON object is further described in section Section 5.

The last line of the response is of the form:

"212" SP "." CRLF

To produce the normal response to an AQRY command, the server first

produces or obtains the requested information in JSON format. The server then encodes the entire JSON object using the BASE64 algorithm, such that each line of the BASE64 output does not exceed 76 characters, not including the CRLF character sequence that terminates each line. The server then prepends "212-" to the beginning of each line of the BASE64 output. Finally, the server appends a single line consisting of "212 ." to the output. Per normal SMTP convention, each line of the reply MUST be terminated by CRLF.

Note: If a address is configured to forward mail to one or more other addresses, this can affect the contents of the JSON object or result in an error. See Section 5.

To recover the JSON from the AQRY reply text, the client first collects the text and ensures that the terminating "212 ." line is present. The terminating line is then discarded, and the "212-" prefix is removed from each of the preceding lines. The resulting text is then fed to the BASE64 decoder to produce a JSON object. The resulting JSON object may then be interpreted.

3.1.3. Redirect AQRY Response

In the case where the SMTP server is configured to accept incoming mail for the address presented in the AQRY command, but either of the following two conditions apply:

(a) in the currently active TLS session, the SMTP server did not present a server certificate with a subjectAltName with dNSName type that matches the domain name portion of the email address presented in the AQRY command; OR

(b) the SMTP server is configured to return a redirect for other reasons, e.g. to shed load from the SMTP server to another server which is better equipped to service that kind of query;

the SMTP server MAY return a multi-line redirect response with a response code of 213. Similar in presentation format to the normal response, the redirect response consists of BASE64-encoded JSON, with each line of the BASE64 text preceded by "213-" and the last line of

the response consisting entirely of "213 ." followed by CRLF.

However, the data structure represented in JSON for a redirect response is different than that of a normal response. The data structure encoded in a redirect response consists of an array of objects describing SMTP servers to which the query can be referred. Each such object may contain the following elements:

host
: DNS name, IPv4 address, or IPv6 address of an SMTP server.

port
: Optional port number to be used to contact the SMTP server. Port 25 is assumed if this element is not supplied.

cookie
: Optional cookie to be passed in the COOKIE parameter to the AQRY command when querying the server. This parameter may be used for any purpose by mutual agreement between the server issuing the redirect response, and the server to which the redirect response refers. For example: it may be used to encode an encrypted database record identifier of the named recipient; or it may be used to encode an encrypted timestamp at which the referral was issued by the server, so that the referred-to server can refuse to return a response if that timestamp is missing or not recent.

There is no significance to the order in which the list items, or the elements of any of the objects in the list, appear in the JSON.

Example: A client issues a query for information about joe@example.com, and the server returns a redirect response:

```
C: AQRY <joe@example.com>
S: 213-W3siaG9zdCI6ICJmb28uZXhhbXBsZS5jb20iLCAiY29va2llIjogImxranNl
S: 213-b3J1IiwgInBvcnQiOiA5ODc2fSwgeyJob3N0IjogIjEwLjEuMi4zIiwgImNv
S: 213-b2tpZSI6ICJzZndlcnYzMyJ9LCB7Imhvc3QiOiAiMjAwMTpEQjg6YWJjZDo6
S: 213-MToyIiwgImNvb2tpZSI6ICJsa2pzZW9ydSIsICJwb3J0IjogNDMyNX1d
S: 213 .
```

The client decodes this and obtains the following data structure (formatted for readability below):

```
[
   { "host": "foo.example.com", "port": 9876,
      "cookie": "lkjseoru" },
   { "host": "10.1.2.3", "cookie": "sfwerv33" },
   { "host": "2001:DB8:abcd::1:2", "port": 4325,
     "cookie": "lkjseoru" }
]
```

The client could then obtain the requested information via any of the following:

- Open a connection to foo.example.com, port 9876, negotiate STARTTLS, then issue the command: "AQRY <joe@example.com> COOKIE=lkjseoru",

- Open a connection to 10.1.2.3, port 25, negotiate STARTTLS, then issue the command: "AQRY <joe@example.com> COOKIE=sfwerv33", OR

- Open a connection to 2001:DB8:abcd::1:2, port 4325, negotiate STARTTLS, then issue the command "AQRY <joe@example.com> COOKIE=lkjseoru" .

In each of the above instances, the client will supply the "host" parameter from the object as the TLS Server Name Indication (SNI) HostName. Any RRVS parameter appearing in the original AQRY command is also supplied when issuing the AQRY command to the redirect servers.

Since the SMTP servers returned in a referral response are not expected to be able to process incoming mail, they are not required to implement the full SMTP protocol. They need only implement the following commands: EHLO (advertising STARTTLS and ADDRQUERY), STARTTLS, AQRY, and QUIT.

3.1.4. Other response codes

In addition to reply codes defined in [RFC5321], the following reply codes SHOULD be used to indicate the error conditions described below:

411 database lookup temporary failure
: This failure occurs whenever the SMTP server must consult some external database or other service in order to provide the requested information, and that service fails to respond within a reasonable time. The client may reasonably retry the command after some interval. [[XXX specify timeout for AQRY]]

511 no information available for this address
: The address appears to be valid but there is no information available that is associated with it.

513 server not configured to handle AQRY for this domain
: The server is configured to accept incoming mail for the domain name appearing in the address, but the server is not configured to

perform queries for addresses in that domain.

550 no such address
The address does not exist.

551 server does not accept incoming mail for this domain
The server is not configured to accept incoming mail for the domain name appearing in the address.

555 command not supported for this recipient
The address may be valid but the AQRY command is not supported for this recipient.

559 TLS required but not negotiated
This reply code is returned whenver a client attempts an AQRY command in a SMTP session that is not protected by TLS.

4. SUBMISSION Service Extension for Address Query Proxy

This section defines a service extension to the Mail Submission Protocol [RFC6409] which can be used by an authenticated, authorized client to query an SMTP server on port 25 for information about an email address. This is intended only as a workaround for port 25 blocking, so the extension is minimally tailored for that purpose.

- The Name of this extension is "Address Query Proxy".
- Servers implementing this extension advertise an additional EHLO keyword of "ADDRQUERYPROXY", which has no associated parameters.
- This extension introduces one new Submission command, AQPX, described below.
- This extension does not alter any existing Submission service commands, nor does this extension change the minimum line length that an implementation of the Submission protocol including this extension must support.

4.1. AQPX Command

The AQPX command is used to query an Submission server for

information about an email address. The client user MUST have already been authenticated and verified to be authorized to use that Submission server. Use of this command by a mail client (such as a Mail User Agent) is OPTIONAL; this specification does not prohibit a client directly contacting an SMTP server. However, it is expected that clients will often need a service as a workaround for the common practice of blocking outbound traffic on TCP port 25.

When this command is received, the Submission server will then:

- verify that the user is authenticated via a TLS-protected session
- consult the SMTP server specified in the AQPX command,
- negotiate a TLS session using STARTTLS,
- verify that the server's certificate is valid and has an appropriate subjectAltName for the address, and if so,
- issue an AQRY command to that server, and
- return the response from the AQRY command.

If some error occurs in the process of performing the above, the Submission server will return an appropriate response code.

The syntax of the AQPX command is as follows:

```
aqpx = "AQPX" SP "<" Mailbox ">"
       "SERVER=" ( Domain / address-literal )
       [ RRVS=date-time ] [ COOKIE=cookie ] CRLF
```

The SERVER parameter specifies an SMTP server to consult. Since this may be any server included in either a response to a DNS MX query, or a server returned in a redirect from a previous query to an SMTP server, the SUBMISSION server SHOULD NOT restrict the servers to which a client may issue a query. There is no provision for specifying the port at which the SMTP server is to be contacted; the client is assumed to be able to directly contact servers on ports other than 25. The RRVS and COOKIE parameters are passed to the AQRY command issued to the SMTP server.

4.2. AQPX responses

Since this is a proxy service that is intended to return a response from a remote SMTP server, any valid response to the SMTP AQRY command (including a normal response, redirect response, or error response) is also a valid response to a Submission service AQPX command.

The submission service SHOULD NOT follow redirects returned by an SMTP server, and MUST return the SMTP server's response intact and without modification.

In addition, the following AQPX-specific response codes are permitted:

- 431 connection or query to SMTP server timed out

- 541 invalid remote server certificate received from <smtp-server-name>

- 542 server certificate for <smtp-server-name> does not match <domain>

5. Address Query Information Data Model

Note: This is preliminary and is expected to need considerable work, and probably a separate document.

The response to the AQRY command is a single JSON object. This JSON object contains zero or more members each of which is itself an object. The members of the outer object are named either for a domain (which does not contain an "@") or an email address (which contains an "@"). In either case the domain or email address are in pure ASCII format, as would be used in a SMTP MAIL command without any domain or address internationalization extensions.

A domain object potentially contains two objects: one named "transmit" which contains attributes describing the domain's default behavior when sending messages, the other named "receive" which contains attributes describing the domain's default behavior when receiving messages. An address object also contains two object: one named "sender" which contains attributes describing that address's

behavior when sending messages, and one named "recipient" which contains attributes describing that address's behavior when receiving messages. The separation of the transmit/receive and sender/recipient roles allow for separate keys and policies to be specified for each. In contrast to the per-address set of attributes, the domain set of attributes provides the capability for the mail domain to supply attributes independently of any recipient, including the ability of the mail domain to sign messages originated by a recipient, and the ability of the mail domain to receive encrypted messages on behalf of a recipient whose mail user agent cannot decrypt mail, and decrypt those messages prior to the recipient's mail user agent obtaining them.

Examples of attributes within these objects include:

domain transmit
"signing_policy", "signing_key_list", "encryption_policy"

domain receive
"accept_encryption", "accept_signature", "encryption_key_list", "alias"

address sender

"signing_policy", "signing_key_list", "encryption_policy"

address recipient
"accept_encryption", "accept_signature", "encryption_key_list", "forwarding_address_list"

Each key would be an array consisting of a string key type identifier consisting of a message type and a key type for that message (e.g. "openpgp-rsa"), followed by the key or certificate itself in the format normally used with that message type. The accept_* attributes would be arrays of strings, each string indicating an encryption or signature format accepted by the domain or recipient.

6. Trustworthiness Of Address Query Responses

As described above, the JSON object returned in response to AQRY may itself contain multiple member objects, each for a separate email address or mail domain. The trustworthiness of each member MUST be

evaluated separately. A member object of an AQRY response MUST NOT be considered trustworthy for any purpose, unless the TLS server certificate used to authenticate the session in which the information was obtained contained a subjectAltName extension specifying a dNSName matching either the domain used to name the section, or the domain portion of the email address used to name the section.

A Submission server implementing the AQPX extension server MUST evaluate the trustworthiness of each named object in the response and only return those sections which are verified to be trustworthy according to the above rule.

7. Enhanced Status Codes for Address Query and Address Query Proxy Extensions

(This section needs to be written.)

8. Security Considerations

- This service relies on the SMTP server's TLS server certificate to authenticate per-domain and per-address information, including potentially public keys for use with encryption and digital signatures. While it appears to have the advantage of being deployable, as most service providers will already be familiar with TLS and X.509 certficicate management, the Address Query service may invest more trust in such servers and their key management practices than was designed for.

- The Address Query Proxy extension to the Submission service inherently requires the client and user to trust the Submission

server to do correct validation and correct name matching of SMTP servers' certificates, as there is no good way to transfer the integrity and authenticity assurances provided by the TLS protocol to the Submission server from the remote SMTP server, to the Submission client.

- With AQRY as it's currently specified, a mail service provider supporting multiple client domains will either need to manage multiple certificates and private keys (one or more for each client domain), or refer queries to a separate server, managed by the client, for each client. A client may prefer to not expose

its private key to a mail service provider. However this choice MAY be made on a per-client basis.

- Especially since the Address Query SMTP service extension does not require authentication, and since it may potentially provide arbitrary information about an email address or mail domain, attackers may attempt to use it to "mine" or "harvest" information about arbitrary mail addresses and their users. It is recommended that only the minimum information necessary for the desired level of mail operation be exposed through this service. In addition, servers MAY return "fake" information for nonexistent recipients in order to discourage probing of arbitrary addresses. Servers MAY also implement rate limiting of AQRY command processing, though this may not be effective against distributed information gathering networks.

- As compared to most uses of SMTP and Submission protocols which primarily transmit data from client to server, this extension is specifically designed to transmit potentially-significant amounts of data from the server to the client. As such, client implementations MUST NOT fail or corrupt internal data when receiving large amounts of data in an SMTP response, nor when processing the returned data (whether or not correctly formatted).

- There is some potential for the AQPX Submission service extension to be used as a means of traffic laundering when attacking other services. However this potential is believed to be minimal (except for data harvesting attack described above) as this service only communicates with other hosts on TCP port 25 and is limited to a very specific SMTP command sequence. Submission servers MUST require authentication before accepting AQPX commands, SHOULD implement rate limiting of such commands or other mechanism to prevent single clients from overusing the service, and SHOULD log at least the number of AQPX queries on a per-user basis.

- The proposed data model anticipates this service being useful for any of several modes of per-message encryption. In addition to end-to-end encryption (in which encryption is done by the sender's MUA and decryption is done by the recipient's MUA), it is also

possible for encryption to be done by the sender's Submission agent, or for the decryption to be done by the recipient's SMTP server or delivery agent or message store (if the sender's MUA or Submission agent encrypt the message for the recipient domain's encryption key). While end-to-end encryption is in some sense the ideal situation, as it theoretically minimizes the potential for exposure of messages, there are several "real world" barriers to its universal adoption. One such barrier is that the majority of mail users today use webmail services in which end-to-end is a fairly meaningless concept. Another such barrier is the widespread use of spam filters and message filtering firewalls which require exposure to received messages in cleartext to be useful. Another such barrier is the legal or other requirement that many organizations have for archival of email communications. Finally, many kinds of personal computer are notoriously insecure, so a user's messages and credentials might actually be better protected on a well-managed server than on his or her own PC. By permitting flexibility in how email encryption is done it is hoped that encryption may be more widely deployed and that it will provide an upgrade path to optimal security for everyone.

9. IANA Considerations

(this section requires elaboration.)

- Register AQRY SMTP service extension
- Register AQPX Submission service extension
- Register new Enhanced Status Codes
- Register new SMTP reply codes (if there is not such a registry, there should be)
- create registry for AQRY data model elements
- possibly reserve port number for use in AQRY redirects

10. References

10.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

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

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

[RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

10.2. Informative References

[RFC1113] Linn, J., "Privacy enhancement for Internet electronic mail: Part I - message encipherment and authentication procedures", RFC 1113, August 1989.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

[RFC7293] Mills, W. and M. Kucherawy, "The Require-Recipient-Valid-Since Header Field and SMTP Service Extension", RFC 7293, July 2014.

Appendix A. Rationale For Design Choices

This section is not normative.

- As described above, the choice of using an SMTP extension for this purpose, and using mail exchangers for the authoritiatve sources of this information, resulted from the observation that only the SMTP servers for incoming mail for a mail domain reliably know how to interpret an email address from that mail domain.

- The redirect response was included because many mail service providers accept incoming mail for large numbers of mail domains, and that it is infeasible and generally inappropriate for a large mail service provider to maintain server certificates that name each of the mail domains for which it provides service. The redirect response thus permits referral of a request to a specific server for each mail domain. The redirect response also may be useful in the case where the listed MX servers for a mail domain do not handle incoming mail directly, but rather forward it to or through one or more internal servers (e.g. firewalls, spam filters) before the message reaches the server responsible for address interpretation and delivery. Finally, the redirect

response may be useful in allowing a heavily-loaded server to devote its resources to mail delivery by referring queries about email address information elsewhere.

- Originally the redirect response contained https URLs, and the queries to other servers were to use http/1 or http/2. This appeared to make client implementations unnecessarily complex, for several reasons: differences in error reporting between SMTP and

HTTP required two sets of error codes and different logic on the client side for each, the existence of HTTP redirects coupled with the need to verify subjectAltName in server certificates appeared to make it difficult to reuse ordinary HTTP library routines. So redirects were changed to specify the DNS name or address, and port, of one or more SMTP servers, thus allowing reuse of the same code on the client for both kinds of query.

- As indicated above, the SUBMISSION extension was created as a workaround for the common practice of blocking outbound TCP traffic to a destination port of 25. However, it also seemed appropriate for a SUBMISSION server to support this functionality based on an anticipated desire for a SUBMISSION server to support additional extensions (not defined in this document) for server-side signing and/or encryption of submitted mail.

- The SUBMISSION AQPX command doesn't support arbitrary ports because it seemed like too much of an opportunity for clients to use that facility for malicious purposes, even if the clients do have to be authenticated. It might be worth considering reserving a specific port for SMTP AQRY referrals.

- The SUBMISSION AQPX command doesn't handle MX lookup, referrals, or retries because of concern over timeout hazards, and because it seemed better to let clients perform these operations than to burden servers with them.

Authors' Addresses

Keith Moore
Network Heretics
PO Box 1934
Knoxville, TN 37901

US

Email: moore@network-heretics.com

Chris Newman
Oracle
440 E. Huntington Dr., Suite 400
Arcadia, CA 91006
US

Email: chris.newman@oracle.com