

APPSAWG  
Internet-Draft  
Obsoletes: 2388 (if approved)  
Intended status: Standards Track  
Expires: October 12, 2015

L. Masinter  
Adobe  
April 10, 2015

Returning Values from Forms: multipart/form-data  
draft-ietf-appsawg-multipart-form-data-11

Abstract

This specification defines the multipart/form-data Internet Media Type, which can be used by a wide variety of applications and transported by a wide variety of protocols as a way of returning a set of values as the result of a user filling out a form. It obsoletes RFC 2388.

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 October 12, 2015.

Copyright Notice

Copyright (c) 2015 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. Introduction . . . . .	2
2. percent-encoding option . . . . .	3
3. Advice for Forms and Form Processing . . . . .	3
4. Definition of multipart/form-data . . . . .	4
4.1. Boundary parameter of multipart/form-data . . . . .	4
4.2. Content-Disposition header for each part . . . . .	4
4.3. filename attribute of content-distribution part header . . . . .	4
4.4. Multiple files for one form field . . . . .	5
4.5. Content-Type header for each part . . . . .	5
4.6. The charset parameter for text/plain form data . . . . .	5
4.7. The _charset_ field for default charset . . . . .	6
4.8. Content-Transfer-Encoding deprecated . . . . .	6
4.9. Other Content- headers . . . . .	7
5. Operability considerations . . . . .	7
5.1. Non-ASCII field names and values . . . . .	7
5.1.1. Avoid non-ASCII field names . . . . .	7
5.1.2. Interpreting forms and creating form-data . . . . .	7
5.1.3. Parsing and interpreting form data . . . . .	8
5.2. Ordered fields and duplicated field names . . . . .	8
5.3. Interoperability with web applications . . . . .	8
5.4. Correlating form data with the original form . . . . .	9
6. IANA Considerations . . . . .	9
7. Security Considerations . . . . .	9
8. Media type registration for multipart/form-data . . . . .	10
9. References . . . . .	11
9.1. Normative References . . . . .	11
9.2. Informative References . . . . .	12
Appendix A. Changes from RFC 2388 . . . . .	12
Appendix B. Alternatives . . . . .	13
Author's Address . . . . .	13

## 1. Introduction

In many applications, it is possible for a user to be presented with a form. The user will fill out the form, including information that is typed, generated by user input, or included from files that the user has selected. When the form is filled out, the data from the form is sent from the user to the receiving application.

The definition of "multipart/form-data" is derived from one of those applications, originally set out in [RFC1867] and subsequently incorporated into HTML 3.2 [W3C.REC-html32-19970114], where forms are expressed in HTML, and in which the form data is sent via HTTP or

electronic mail. This representation is widely implemented in numerous web browsers and web servers.

However, "multipart/form-data" is also used for forms that are presented using representations other than HTML (spreadsheets, PDF, etc.), and for transport using means other than electronic mail or HTTP; it is used in distributed applications which do not involve forms at all, or do not have users filling out the form. For this reason, this document defines a general syntax and semantics independent of the application for which it is used, with specific rules for web applications noted in context.

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 BCP 14, RFC 2119 [RFC2119].

## 2. percent-encoding option

Within this specification, "percent-encoding" (as defined in [RFC3986]) is offered as a possible way of encoding characters in file names that are otherwise disallowed, including non-ASCII characters, spaces, control characters and so forth. The encoding is created replacing each non-ASCII or disallowed character with a sequence, where each byte of the UTF-8 encoding of the character is represented by a percent-sign (%) followed by the (case-insensitive) hexadecimal of that byte.

## 3. Advice for Forms and Form Processing

The representation and interpretation of forms and the nature of form processing is not specified by this document. However, for forms and form-processing that result in generation of multipart/form-data, some suggestions are included.

In a form, there is generally a sequence of fields, where each field is expected to be supplied with a value, e.g. by a user who fills out the form. Each field has a name. After a form has been filled out, and the form's data is "submitted": the form processing results in a set of values for each field-- the "form data".

In forms that work with multipart/form-data, field names could be arbitrary Unicode strings; however, restricting field names to ASCII will help avoid some interoperability issues (see Section 5.1).

Within a given form, ensuring field names are unique is also helpful. Some fields may have default values or presupplied values in the form itself. Fields with presupplied values might be hidden or invisible;

this allows using generic processing for form data from a variety of actual forms.

#### 4. Definition of multipart/form-data

The media-type "multipart/form-data" follows the model of multipart MIME data streams as specified in [RFC2046] Section 5.1; changes are noted in this document.

A "multipart/form-data" body contains a series of parts, separated by a boundary.

##### 4.1. Boundary parameter of multipart/form-data

As with other multipart types, the parts are delimited with a boundary delimiter, constructed using CRLF, "--", the value of the boundary parameter. The boundary is supplied as a "boundary" parameter to the "multipart/form-data" type. As noted in [RFC2046] Section 5.1, the boundary delimiter MUST NOT appear inside any of the encapsulated parts, and it is often necessary to enclose the boundary parameter values in quotes on the Content-type line.

##### 4.2. Content-Disposition header for each part

Each part MUST contain a "content-disposition" header [RFC2183] and where the disposition type is "form-data". The "content-disposition" header MUST also contain an additional parameter of "name"; the value of the "name" parameter is the original field name from the form (possibly encoded; see Section 5.1). For example, a part might contain a header:

```
Content-Disposition: form-data; name="user"
```

with the body of the part containing the form data of the "user" field.

##### 4.3. filename attribute of content-distribution part header

For form data that represents the content of a file, a name for the file SHOULD be supplied as well, by using a "filename" parameter of the "content-disposition" header. The file name isn't mandatory for cases where the file name isn't available or is meaningless or private; this might result, for example, from selection or drag-and-drop or where the form data content is streamed directly from a device.

If a filename parameter is supplied, the requirements of [RFC2183] Section 2.3 for "receiving MUA" apply to receivers of "multipart/

form-data" as well: Do not use the file name blindly, check and possibly change to match local filesystem conventions if applicable, do not use directory path information that may be present.

In most multipart types, the MIME headers in each part are restricted to US-ASCII; for compatibility with those systems, file names normally visible to users MAY be encoded using the percent-encoding method in Section 2, following how a "file:" URI [I-D.ietf-appsawg-file-scheme] might be encoded.

NOTE: The encoding method described in [RFC5987], which would add a "filename\*" paramter to the "Content-Disposition" header, MUST NOT be used.

Some commonly deployed systems use multipart/form-data with file names directly encoded including octets outside the US-ASCII range. The encoding used for the file names is typically UTF-8, although HTML forms will use the charset associated with the form.

#### 4.4. Multiple files for one form field

The form data for a form field might include multiple files.

[RFC2388] suggested that multiple files for a single form field be transmitted using a nested multipart/mixed part. This usage is deprecated.

To match widely deployed implementations, multiple files MUST be sent by supplying each file in a separate part, but all with the same "name" parameter.

Receiving applications intended for wide applicability (e.g. multipart/form-data parsing libraries) SHOULD also support the older method of supplying multiple files.

#### 4.5. Content-Type header for each part

Each part MAY have an (optional) "content-type", which defaults to "text/plain". If the contents of a file are to be sent, the file data SHOULD be labeled with an appropriate media type, if known, or "application/octet-stream".

#### 4.6. The charset parameter for text/plain form data

In the case where the form data is text, the charset parameter for the "text/plain" Content-Type MAY be used to indicate the character encoding used in that part. For example, a form with a text field in

which a user typed "Joe owes <eu>100" where <eu> is the Euro symbol might have form data returned as:

```
--AaB03x
content-disposition: form-data; name="field1"
content-type: text/plain;charset=UTF-8
content-transfer-encoding: quoted-printable

Joe owes =E2=82=AC100.
--AaB03x
```

In practice, many widely deployed implementations do not supply a charset parameter in each part, but, rather, they rely on the notion of a "default charset" for a multipart/form-data instance. Subsequent sections will explain how the default charset is established.

#### 4.7. The `_charset_` field for default charset

Some form processing applications (including HTML) have the convention that the value of a form entry with entry name "`_charset_`" and type "hidden" is automatically set when the form is opened; the value is used as the default charset of text field values (see form-charset in Section 5.1.2). In such cases, the value of the default charset for each text/plain part without a charset parameter is the supplied value. For example:

```
--AaB03x
content-disposition: form-data; name="_charset_"

iso-8859-1
--AaB03x--
content-disposition: form-data; name="field1"

...text encoded in iso-8859-1 ...
AaB03x--
```

#### 4.8. Content-Transfer-Encoding deprecated

Previously, it was recommended that senders use a "Content-Transfer-Encoding" encoding (such as "quoted-printable") for each non-ASCII part of a multipart/form-data body, because that would allow use in transports that only support a "7BIT" encoding. This use is deprecated for use in contexts that support binary data such as HTTP. Senders SHOULD NOT generate any parts with a "Content-Transfer-Encoding" header.

Currently, no deployed implementations that send such bodies have been discovered.

#### 4.9. Other Content- headers

The "multipart/form-data" media type does not support any MIME headers in the parts other than Content-Type, Content-Disposition, and (in limited circumstances) Content-Transfer-Encoding. Other headers MUST NOT be included and MUST be ignored.

### 5. Operability considerations

#### 5.1. Non-ASCII field names and values

Normally, MIME headers in multipart bodies are required to consist only of 7-bit data in the US-ASCII character set. While [RFC2388] suggested that non-ASCII field names be encoded according to the method in [RFC2047], this practice doesn't seem to have been followed widely.

This specification makes three sets of recommendations for three different states of workflow.

##### 5.1.1. Avoid non-ASCII field names

For broadest interoperability with existing deployed software, those creating forms SHOULD avoid non-ASCII field names. This should not be a burden, because in general the field names are not visible to users. The field names in the underlying need not match what the user sees on the screen.

If non-ASCII field names are unavoidable, form or application creators SHOULD use UTF-8 uniformly. This will minimize interoperability problems.

##### 5.1.2. Interpreting forms and creating form-data

Some applications of this specification will supply a character encoding to be used for interpretation of the multipart/form-data body. In particular, HTML 5 [W3C.REC-html5-20141028] uses:

- o The content of a '\_charset\_' field, if there is one.
- o the value of an accept-charset attribute of the <form> element, if there is one,
- o the character encoding of the document containing the form, if it is US-ASCII compatible,

- o otherwise UTF-8.

Call this value the form-charset. Any text, whether field name, field value, or (text/plain) form data which uses characters outside the ASCII range MAY be represented directly encoded in the form-charset.

#### 5.1.3. Parsing and interpreting form data

While this specification provides guidance for creation of multipart/form-data, parsers and interpreters should be aware of the variety of implementations. File systems differ as to whether and how they normalize Unicode names, for example. The matching of form elements to form-data parts may rely on a fuzzier match. In particular, some multipart/form-data generators might have followed the previous advice of [RFC2388] and used the [RFC2047] "encoded-word" method of encoding non-ASCII values:

```
encoded-word = "=?" charset "?" encoding "?" encoded-text "=?"
```

Others have been known to follow [RFC2231], to send unencoded UTF-8, or even strings encoded in the form-charset.

For this reason, interpreting "multipart/form-data" (even from conforming generators) may require knowing the charset used in form encoding, in cases where the `_charset_` field value or a charset parameter of a text/plain Content-Type header is not supplied.

#### 5.2. Ordered fields and duplicated field names

Form processors given forms with a well-defined ordering SHOULD send back results in order (note that there are some forms which do not define a natural order.) Intermediaries MUST NOT reorder the results. Form parts with identical field names MUST NOT be coalesced.

#### 5.3. Interoperability with web applications

Many web applications use the "application/x-url-encoded" method for returning data from forms. This format is quite compact, e.g.:

```
name=Xavier+Xantico&verdict=Yes&colour=Blue&happy=sad&Utf%F6r=Send
```

However, there is no opportunity to label the enclosed data with content type, apply a charset, or use other encoding mechanisms.

Many form-interpreting programs (primarily web browsers) now implement and generate multipart/form-data, but an existing



application might need to optionally support both the application/x-url-encoded format as well.

#### 5.4. Correlating form data with the original form

This specification provides no specific mechanism by which multipart/form-data can be associated with the form that caused it to be transmitted. This separation is intentional; many different forms might be used for transmitting the same data. In practice, applications may supply a specific form processing resource (in HTML, the ACTION attribute in a FORM tag) for each different form. Alternatively, data about the form might be encoded in a "hidden field" (a field which is part of the form but which has a fixed value to be transmitted back to the form-data processor.)

#### 6. IANA Considerations

Please update the Internet Media Type registration of multipart/form-data to point to this document, using the template in Section 8. In addition, please update the registrations of the "name" parameter and the "form-data" value in the "Content Disposition Values and Parameters" registry to both point to this document.

#### 7. Security Considerations

All form processing software should treat user supplied form-data with sensitivity, as it often contains confidential or personally identifying information. There is widespread use of form "auto-fill" features in web browsers; these might be used to trick users to unknowingly send confidential information when completing otherwise innocuous tasks. Multipart/form-data does not supply any features for checking integrity, ensuring confidentiality, avoiding user confusion, or other security features; those concerns must be addressed by the form-filling and form-data-interpreting applications.

Applications which receive forms and process them must be careful not to supply data back to the requesting form processing site that was not intended to be sent.

It is important when interpreting the filename of the Content-Disposition header to not overwrite files in the recipient's file space inadvertently.

User applications that request form information from users must be careful not to cause a user to send information to the requestor or a third party unwillingly or unwittingly. For example, a form might request 'spam' information to be sent to an unintended third party,

or private information to be sent to someone that the user might not actually intend. While this is primarily an issue for the representation and interpretation of forms themselves (rather than the data representation of the form data), the transportation of private information must be done in a way that does not expose it to unwanted prying.

With the introduction of form-data that can reasonably send back the content of files from a user's file space, the possibility arises that a user might be sent an automated script that fills out a form and then sends one of the user's local files to another address. Thus, additional caution is required when executing automated scripting where form-data might include a user's files.

Files sent via multipart/form-data may contain arbitrary executable content, and precautions against malicious content are necessary.

The considerations of [RFC2183] Sections 2.3 and 5 with respect to the filename parameter of the Content-Disposition header also apply to its usage here.

#### 8. Media type registration for multipart/form-data

This section is the [RFC6838] media type registration.

Type name: multipart

Subtype name: form-data

Required parameters: boundary

Optional parameters: none

Encoding considerations: Common use is BINARY.

In limited use (or transports that restrict the encoding to 7BIT or 8BIT each part is encoded separately using Content-Transfer-Encoding Section 4.8).

Security considerations: See Section 7 of this document.

Interoperability considerations: This document makes several recommendations for interoperability with deployed implementations, including Section 4.8.

Published specification: This document.

Applications that use this media type: Numerous web browsers, servers, and web applications.

Fragment identifier considerations: None: Fragment identifiers are not defined for this type.

Additional information: None: no deprecated alias names, magic numbers, file extensions or Macintosh sssfile type codes.

Person & email address to contact                      for further information  
Author of this document.

Intended Usage: COMMON

Restrictions on usage: none

Author: Author of this document.

Change controller: IETF

Provisional registration: N/A

## 9. References

### 9.1. Normative References

- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2183] Troost, R., Dorner, S., and K. Moore, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.
- [RFC2231] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

## 9.2. Informative References

- [I-D.ietf-appsawg-file-scheme]  
Kerwin, M., "The file URI Scheme", draft-ietf-appsawg-file-scheme-00 (work in progress), January 2015.
- [RFC1867] Nebel, E. and L. Masinter, "Form-based File Upload in HTML", RFC 1867, November 1995.
- [RFC2388] Masinter, L., "Returning Values from Forms: multipart/form-data", RFC 2388, August 1998.
- [RFC5987] Reschke, J., "Character Set and Language Encoding for Hypertext Transfer Protocol (HTTP) Header Field Parameters", RFC 5987, August 2010.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.
- [W3C.REC-html32-19970114]  
Raggett, D., "HTML 3.2 Reference Specification", World Wide Web Consortium Recommendation REC-html32-19970114, January 1997, <<http://www.w3.org/TR/REC-html32-19970114>>.
- [W3C.REC-html5-20141028]  
Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, E., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014, <<http://www.w3.org/TR/2014/REC-html5-20141028>>.

## Appendix A. Changes from RFC 2388

The handling of non-ASCII field names changed-- no longer recommending the RFC 2047 method, instead suggesting senders send UTF-8 field names directly, and file names directly in the form-charset.

The handling of multiple files submitted as the result of a single form field (e.g. HTML's <input type=file multiple> element) results in each file having its own top level part with the same name parameter; the method of using a nested "multipart/mixed" from [RFC2388] is no longer recommended for creators, and not required for receivers as there are no known implementations of senders.

The `_charset_` convention and use of an explicit form-data charset is documented.

'boundary' is a required parameter in Content-Type.

The relationship of the ordering of fields within a form and the ordering of returned values within multipart/form-data was not defined before, nor was the handling of the case where a form has multiple fields with the same name.

Editorial: Removed obsolete discussion of alternatives in appendix. Update references. Move outline of form processing into Introduction.

## Appendix B. Alternatives

There are numerous alternative ways in which form data can be encoded; many are listed in [RFC2388] section 5.2. The multipart/form-data encoding is verbose, especially if there are many fields with short values. In most use cases, this overhead isn't significant.

More problematic are the differences introduced when implementors opted to not follow [RFC2388] when encoding non-ASCII field names (perhaps because "may" should have been "MUST"). As a result, parsers need to be more complex for matching against the possible outputs of various encoding methods.

## Author's Address

Larry Masinter  
Adobe

Email: [masinter@adobe.com](mailto:masinter@adobe.com)  
URI: <http://larry.masinter.net>

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 21, 2014

W. Mills  
Yahoo! Inc.  
M. Kucherawy  
Facebook, Inc.  
April 19, 2014

The Require-Recipient-Valid-Since Header Field and SMTP Service  
Extension  
draft-ietf-appsawg-rrvs-header-field-11

## Abstract

This document defines an extension for the Simple Mail Transfer Protocol called RRVs, to provide a method for senders to indicate to receivers a point in time when the ownership of the target mailbox was known to the sender. This can be used to detect changes of mailbox ownership, and thus prevent mail from being delivered to the wrong party. This document also defines a header field called Require-Recipient-Valid-Since that can be used to tunnel the request through servers that do not support the extension.

The intended use of these facilities is on automatically generated messages, such as account statements or password change instructions, that might contain sensitive information, though it may also be useful in other applications.

## 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 October 21, 2014.

## Copyright Notice

Copyright (c) 2014 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. Introduction . . . . .	4
2. Definitions . . . . .	4
3. Description . . . . .	5
3.1. The 'RRVS' SMTP Extension . . . . .	5
3.2. The 'Require-Recipient-Valid-Since' Header Field . . . . .	6
3.3. Timestamps . . . . .	6
4. Use By Generators . . . . .	6
5. Handling By Receivers . . . . .	7
5.1. SMTP Extension Used . . . . .	8
5.1.1. Relays . . . . .	9
5.2. Header Field Used . . . . .	9
5.2.1. Design Choices . . . . .	10
5.3. Clock Synchronization . . . . .	11
6. Relaying Without RRVS Support . . . . .	11
6.1. Header Field Conversion . . . . .	12
7. Header Field with Multiple Recipients . . . . .	13
8. Special Use Addresses . . . . .	13
8.1. Mailing Lists . . . . .	14
8.2. Single-Recipient Aliases . . . . .	14
8.3. Multiple-Recipient Aliases . . . . .	14
8.4. Confidential Forwarding Addresses . . . . .	15
8.5. Suggested Mailing List Enhancements . . . . .	15
9. Continuous Ownership . . . . .	15
10. Digital Signatures . . . . .	16
11. Authentication-Results Definitions . . . . .	16
12. Examples . . . . .	17
12.1. SMTP Extension Example . . . . .	17
12.2. Header Field Example . . . . .	18
12.3. Authentication-Results Example . . . . .	18
13. Security Considerations . . . . .	18
13.1. Abuse Countermeasures . . . . .	18
13.2. Suggested Use Restrictions . . . . .	19
13.3. False Sense of Security . . . . .	19
13.4. Reassignment of Mailboxes . . . . .	19
14. Privacy Considerations . . . . .	20
14.1. The Trade-Off . . . . .	20

14.2. Probing Attacks . . . . .	20
14.3. Envelope Recipients . . . . .	20
14.4. Risks with Use . . . . .	21
15. IANA Considerations . . . . .	21
15.1. SMTP Extension Registration . . . . .	21
15.2. Header Field Registration . . . . .	21
15.3. Enhanced Status Code Registration . . . . .	21
15.4. Authentication Results Registration . . . . .	22
16. References . . . . .	23
16.1. Normative References . . . . .	23
16.2. Informative References . . . . .	24
Appendix A. Acknowledgments . . . . .	24



## 1. Introduction

Email addresses sometimes get reassigned to a different person. For example, employment changes at a company can cause an address used for an ex-employee to be assigned to a new employee, or a mail service provider (MSP) might expire an account and then let someone else register for the local-part that was previously used. Those who sent mail to the previous owner of an address might not know that it has been reassigned. This can lead to the sending of email to the correct address, but the wrong recipient. This situation is of particular concern with transactional mail related to purchases, online accounts, and the like.

What is needed is a way to indicate an attribute of the recipient that will distinguish between the previous owner of an address and its current owner, if they are different. Further, this needs to be done in a way that respects privacy.

The mechanisms specified here allow the sender of the mail to indicate how "old" the address assignment is expected to be. In effect, the sender is saying, "I know that the intended recipient was using this address at this point in time. I don't want this message delivered to anyone else" A receiving system can then compare this information against the point in time at which the address was assigned to its current user. If the assignment was made later than the point in time indicated in the message, there is a good chance the current user of the address is not the correct recipient. The receiving system can then prevent delivery and, preferably, notify the original sender of the problem.

The primary application is transactional mail (such as account information, password change requests, and other automatically generated messages) rather than user-authored content. However, it may be useful in other contexts; for example, a personal address book could record the time an email address was added to it, and thus use that time with this extension.

Because the use cases for this extension are strongly tied to privacy issues, attention to the Security Considerations (Section 13) and the Privacy Considerations (Section 14), is particularly important. Note, especially, the limitation described in Section 13.3.

## 2. Definitions

For a description of the email architecture, consult [EMAIL-ARCH].

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 [KEYWORDS].

### 3. Description

To address the problem described in Section 1, a mail sending client (usually an automated agent) needs to indicate to the server to which it is connecting that it expects the destination address of the message to have been under continuous ownership (see Section 9) since a specified point time. That specified time would be the time when the intended recipient gave the address to the message author, or perhaps a more recent time when the intended recipient reconfirmed ownership of the address with the sender.

Two mechanisms are defined here: an extension to the Simple Mail Transfer Protocol [SMTP] and a new message header field. The SMTP extension permits strong assurance of enforcement by confirming support at each handling step for a message, and the option to demand support at all nodes in the handling path of the message (and returning of the message to the originator otherwise). The header field can be used when the Message Delivery Agent (MDA) supports this function, but an intermediary system between the sending system and the MDA does not. However, the header field does not provide the same strong assurance described above, and is more prone to exposure of private information (see Section 14.1).

The SMTP extension is called "RRVS" (Require Recipient Valid Since), and adds a parameter to the SMTP "RCPT" command that indicates the most recent point in time when the message author believed the destination mailbox to be under the continuous ownership of a specific party. Similarly, the Require-Recipient-Valid-Since header field includes an intended recipient coupled with a timestamp indicating the same thing.

#### 3.1. The 'RRVS' SMTP Extension

Extensions to SMTP are described in Section 2.2 of [SMTP].

The name of the extension is "RRVS", an abbreviation of "Require Recipient Valid Since". Servers implementing the SMTP extension advertise an additional EHLO keyword of "RRVS", which has no associated parameters, introduces no new SMTP commands, and does not alter the MAIL command.

A Message Transfer Agent (MTA) implementing RRVS can transmit or accept one new parameter to the RCPT command. An MDA can also accept this new parameter. The parameter is "RRVS", and the value is a timestamp expressed as a "date-time" as defined in [DATETIME], with the added restriction that a "time-secfrac" MUST NOT be used. The

timestamp MAY optionally be followed by a semi-colon character and a letter (known as the "no-support action") indicating the action to be taken when a downstream MTA is discovered that does not support the extension. Valid actions are "R" (reject; the default) and "C" (continue).

Formally, the new parameter and its value are defined as follows:

```
rrvs-param = "RRVS=" date-time [ ";" ( "C" / "R" ) ]
```

Accordingly, this extension increases the maximum command length for the RCPT command by 33 characters.

The meaning of this extension, when used, is described in Section 5.1.

### 3.2. The 'Require-Recipient-Valid-Since' Header Field

The general constraints on syntax and placement of header fields in a message are defined in Internet Message Format [MAIL].

Using Augmented Backus-Naur Form [ABNF], the syntax for the field is:

```
rrvs = "Require-Recipient-Valid-Since:" addr-spec ";" date-time  
      CRLF
```

"date-time" is defined in Section 3.3, and "addr-spec" is defined in Section 3.4.1, of [MAIL].

### 3.3. Timestamps

The header field version of this protocol has a different format for the date and time expression than the SMTP extension does. This is because message header fields use a format to express time and date that is specific to message header fields, and this is consistent with that usage.

Use of both date and time is done to be consistent with how current implementations typically store the timestamp, and to make it easy to include the time zone. In practice, granularity beyond the date may or may not be useful.

## 4. Use By Generators

When a message is generated whose content is sufficiently sensitive that an author or author's Administrative Management Domain (ADMD; see [EMAIL-ARCH]) wishes to protect against misdelivery using this protocol, it determines for each recipient mailbox on the message a

timestamp at which it last confirmed ownership of that mailbox. It then applies the SMTP extension when sending the message to its destination.

In cases where the outgoing MTA does not support the extension, the header field defined above can be used to pass the request through that system. However, use of the header field is only a "best-effort" approach to solving the stated goals, and it has some shortcomings:

1. The positive confirmation of support at each handling node, with the option to return the message to the originator when end-to-end support cannot be confirmed, will be unavailable;
2. The protocol is focused on affecting delivery (that is, the transaction) rather than on content, and therefore use of a header field in the content is generally inappropriate;
3. The mechanism cannot be used with multiple recipients without unintentionally exposing information about one recipient to the others (see Section 7; and
4. There is a risk of the timestamp parameter being inadvertently forwarded, automatically or intentionally by the user (since user agents might not reveal the presence of the header field), and therefore exposed to unintended recipients. (See Section 14.4.)

Thus, the header field format MUST NOT be used unless the originator or relay has specific knowledge that the receiving MDA or an intermediary MTA will apply it properly. In any case, it SHOULD NOT be used for the multi-recipient case.

Use of the header field mechanism is further restricted by the practices described in Section 7.2 of [SMTP], Section 3.6.3 of [MAIL], and Section 7 of this document.

## 5. Handling By Receivers

If a receiver implements this specification, then there are two possible evaluation paths:

1. The sending client uses the extension, and so there was an RRVS parameter on a RCPT TO command in the SMTP session and the parameters of interest are taken only from there (and the header field, if present, is disregarded); or
2. The sending client does not use the extension, so the RRVS parameter was not present on the RCPT TO commands in the SMTP

session, but the corresponding header field might be present in the message.

When the continuous ownership test fails for transient reasons (such as an unavailable database or other condition that is likely temporary), normal transient failure handling for the message is applied.

If the continuous ownership test cannot be completed because the necessary datum (the mailbox creation or reassignment date/time) was not recorded, the MDA doing the evaluation selects a date and time to use that is the latest possible point in time at which the mailbox could have been created or reassigned. For example, this might be the earliest of all recorded mailbox creation/reassignment timestamps, or the time when the host was first installed. If no reasonable substitute for the timestamp can be selected, the MDA rejects the message using an SMTP reply code, preferably with an enhanced mail system status code (see Section 15.3), that indicates the test cannot be completed. A message originator can then decide whether to reissue the message without RRVs protection, or find another way to reach the mailbox owner.

#### 5.1. SMTP Extension Used

For an MTA supporting the SMTP extension, the requirement is to continue enforcement of RRVs during the relaying process to the next MTA or the MDA.

A receiving MTA or MDA that implements the SMTP extension declared above and observes an RRVs parameter on a RCPT TO command checks whether the current owner of the destination mailbox has held it continuously, far enough back to include the given point in time, and delivers it unless that check returns in the negative. Specifically, an MDA will do the following before continuing with delivery:

1. Ignore the parameter if the named mailbox is known to be a role account as listed in Mailbox Names For Common Services, Roles And Functions [ROLES].
2. If the address is not known to be a role account, and if that address has not been under continuous ownership since the timestamp specified in the extension, return a 550 error to the RCPT command. (See also Section 15.3.)

#### 5.1.1.1. Relays

An MTA that does not make mailbox ownership checks, such as an MTA positioned to do SMTP ingress at an organizational boundary, SHOULD relay the RRVS extension parameter to the next MTA or MDA so that it can be processed there.

For the SMTP extension, the optional RRVS parameter defined in Section 5.1 indicates the action to be taken when relaying a message to another MTA that does not advertise support for this extension. When this is the case and the no-support action was not specified or is "R" (reject), the MTA handling the message MUST reject the message by:

1. returning a 550 error to the DATA command, if synchronous service is being provided to the SMTP client that introduced the message; or
2. generating a [DSN] to indicate to the originator of the message that the non-delivery occurred, and terminating further relay attempts.

An enhanced mail system status code is defined for such rejections in Section 15.3.

See Section 8.2 for additional discussion.

When relaying, an MTA MUST preserve the no-support action if it was used by the SMTP client.

#### 5.2. Header Field Used

A receiving system that implements this specification, upon receiving a message bearing a Require-Recipient-Valid-Since header field when no corresponding RRVS SMTP extension was used, checks whether the destination mailbox owner has held it continuously, far enough back to include the given date-time, and delivers it unless that check returns in the negative. Expressed as a sequence of steps:

1. Extract those Require-Recipient-Valid-Since fields from the message that contain a recipient for which no corresponding RRVS SMTP extension was used.
2. Discard any such fields that match any of these criteria:
  - \* are syntactically invalid;

- \* name a role account as listed in [ROLES];
  - \* the "addr-spec" portion does not match a current recipient, as listed in the RCPT TO commands in the SMTP session; or
  - \* the "addr-spec" portion does not refer to a mailbox handled for local delivery by this ADMD.
3. For each field remaining, determine if the named address has been under continuous ownership since the corresponding timestamp. If it has not, reject the message.
  4. RECOMMENDED: If local delivery is being performed, remove all instances of this field prior to delivery to a mailbox; if the message is being forwarded, remove those instances of this header field that were not discarded by step 2 above.

Handling proceeds normally upon completion of the above steps if rejection has not been performed.

The final step is not mandatory as not all mail handling agents are capable of stripping away header fields, and there are sometimes reasons to keep the field intact such as debugging or presence of digital signatures that might be invalidated by such a change. See Section 10 for additional discussion.

If a message is to be rejected within the SMTP protocol itself (versus generating a rejection message separately), servers implementing this protocol SHOULD also implement the SMTP extension described in Enhanced Mail System Status Codes [ESC] and use the enhanced status codes described in Section 15.3 as appropriate.

Implementation by this method is expected to be transparent to non-participants, since they would typically ignore this header field.

This header field is not normally added to a message that is addressed to multiple recipients. The intended use of this field involves an author seeking to protect transactional or otherwise sensitive data intended for a single recipient, and thus generating independent messages for each individual recipient is normal practice. See Section 7 for further discussion and restrictions.

#### 5.2.1. Design Choices

The presence of the address in the field content supports the case where a message bearing this header field is forwarded. The specific use case is as follows:

1. A user subscribes to a service "S" on date "D" and confirms an email address at the user's current location, "A";
2. At some later date, the user intends to leave the current location, and thus creates a new mailbox elsewhere, at "B";
3. The user configures address "A" to forward to "B";
4. "S" constructs a message to "A" claiming that address was valid at date "D" and sends it to "A";
5. The receiving MTA for "A" determines that the forwarding in effect was created by the same party that owned the mailbox there, and thus concludes the continuous ownership test has been satisfied;
6. If possible, the MTA for "A" removes this header field from the message, and in either case, forwards it to "B";
7. On receipt at "B", either the header field has been removed, or the header field does not refer to a current envelope recipient, and in either case delivers the message.

Section 8 discusses some interesting use cases, such as the case where "B" above results in further forwarding of the message.

SMTP has never required any correspondence between addresses in the RFC5321.MailFrom and RFC5321.RcptTo parameters and header fields of a message, which is why the header field defined here contains the recipient address to which the timestamp applies.

### 5.3. Clock Synchronization

The timestamp portion of this specification supports a precision at the seconds level. Although uncommon, it is not impossible for a clock at either a generator or a receiver to be incorrect, leading to an incorrect result in the RRVS evaluation.

To minimize the risk of such incorrect results, both generators and receivers implementing this specification MUST use a standard clock synchronization protocol such as [NTP] to synchronize to a common clock.

### 6. Relaying Without RRVS Support

When a message is received using the SMTP extension defined here but will not be delivered locally (that is, it needs to be relayed further), the MTA to which the relay will take place might not be



compliant with this specification. Where the MTA in possession of the message observes it is going to relay the message to an MTA that does not advertise this extension, it needs to choose one of the following actions:

1. Decline to relay the message further, preferably generating a Delivery Status Notification [DSN] to indicate failure (RECOMMENDED);
2. Downgrade the data thus provided in the SMTP extension to a header field, as described in Section 6.1 below (SHOULD NOT unless the conditions in that section are satisfied, and only when the previous option is not available); or
3. Silently continue with delivery, dropping the protection offered by this protocol.

Using other than the first option needs to be avoided unless there is specific knowledge that further relaying with the degraded protections thus provided does not introduce undue risk.

#### 6.1. Header Field Conversion

If an SMTP server ("B") receives a message bearing one or more Require-Recipient-Valid-Since from a client ("A"), presumably because "A" does not support the SMTP extension, and needs to relay the corresponding message on to another server ("C") (thereby becoming a client), and "C" advertises support for the SMTP extension, "B" SHOULD delete the header field(s) and instead relay this information by making use of the SMTP extension. Note that such modification of the header might affect later validation of the header upon delivery; for example, a hash of the modified header would produce a different result. This might be a valid cause for some operators to skip this delete operation.

Conversely, if "B" has received a mailbox timestamp from "A" using the SMTP extension for which it must now relay the message on to "C", but "C" does not advertise the SMTP extension, and "B" does not reject the message because rejection was specifically declined by the client (see Section 5.1.1), "B" SHOULD add a Require-Recipient-Valid-Since header field matching the mailbox to which relaying is being done, and the corresponding valid-since timestamp for it, if it has prior information that the eventual MDA or another intermediate MTA supports this mechanism and will be able to process the header field as described in this specification.

The admonitions about very cautious use of the header field described in Section 4 apply to this relaying mechanism as well. If multiple

mailbox timestamps are received from "A", the admonitions in Section 7 also apply.

## 7. Header Field with Multiple Recipients

Numerous issues arise when using the header field form of this extension, particularly when multiple recipients are specified for a single message resulting in multiple fields each with a distinct address and timestamp.

Because of the nature of SMTP, a message bearing a multiplicity of Require-Recipient-Valid-Since header fields could result in a single delivery attempt for multiple recipients (in particular, if two of the recipients are handled by the same server), and if any one of them fails the test, the delivery fails to all of them; it then becomes necessary to do one of the following:

- o reject the message on completion of the DATA phase of the SMTP session, which is a rejection of delivery to all recipients; or
- o accept the message on completion of DATA, and then generate a Delivery Status Notification [DSN] message for each of the failed recipients.

Additional complexity arises when a message is sent to two recipients, "A" and "B", presumably with different timestamps, both of which are then redirected to a common address "C". The author is not necessarily aware of the current or past ownership of mailbox "C", or indeed that "A" and/or "B" have been redirected. This might result in either or both of the two deliveries failing at "C", which is likely to confuse the message author, who (as far as the author is aware) never sent a message to "C" in the first place.

Finally, there is an obvious concern with the fan-out of a message bearing the timestamps of multiple users; tight control over the handling of the timestamp information is very difficult to assure as the number of handling agents increases.

## 8. Special Use Addresses

In [DSN-SMTP], an SMTP extension was defined to allow SMTP clients to request generation of DSNs, and related information to allow such reports to be maximally useful. Section 5.2.7 of that document explored the issue of the use of that extension where the recipient is a mailing list. This extension has similar concerns which are covered here following that document as a model.

For all cases described below, a receiving MTA SHOULD NOT introduce

RRVS in either form (SMTP extension or header field) if the message did not arrive with RRVS in use. This would amount to second-guessing of the message originator's intention and might lead to an undesirable outcome.

### 8.1. Mailing Lists

Delivery to a mailing list service is considered a final delivery. Where this protocol is in use, it is evaluated as per any normal delivery: If the same mailing list has been operating in place of the specified recipient mailbox since at least the timestamp given as the RRVS parameter, the message is delivered to the list service normally, and is otherwise not delivered.

It is important, however, that the participating MDA passing the message to the list service needs to omit the RRVS parameter in either form (SMTP extension or header field) when doing so. The emission of a message from the list service to its subscribers constitutes a new message not covered by the previous transaction.

### 8.2. Single-Recipient Aliases

Upon delivery of an RRVS-protected message to an alias (acting in place of a mailbox) that results in relaying of the message to a single other destination, the usual RRVS check is performed. The continuous ownership test here might succeed if, for example, a conventional user inbox was replaced with an alias on behalf of that same user, and the time when this was done is recorded in a way that can be queried by the relaying MTA.

If the relaying system also performs some kind of step where ownership of the new destination address is confirmed, it SHOULD apply RRVS using the later of that timestamp and the one that was used inbound. This also allows for changes to the alias without disrupting the protection offered by RRVS.

If the relaying system has no such time records related to the new destination address, the RRVS SMTP extension is not used on the relaying SMTP session, and the header field relative to the local alias is removed, in accordance with Section 5.

### 8.3. Multiple-Recipient Aliases

Upon delivery of an RRVS-protected message to an alias (acting in place of a mailbox) that results in relaying of the message to multiple other destinations, the usual RRVS check is performed as in Section 8.2. The MTA expanding such an alias then decides which of the options enumerated in that section is to be applied for each new

recipient.

#### 8.4. Confidential Forwarding Addresses

In the above cases, the original author could receive message rejections, such as DSNs, from the ultimate destination, where the RRVS check (or indeed, any other) fails and rejection is warranted. This can reveal the existence of a forwarding relationship between the original intended recipient and the actual final recipient.

Where this is a concern, the initial delivery attempt is to be treated like a mailing list delivery, with RRVS evaluation done and then all RRVS information removed from the message prior to relaying it to its true destination.

#### 8.5. Suggested Mailing List Enhancements

Mailing list services could store the timestamp at which a subscriber was added to a mailing list. This specification could then be used in conjunction with that information in order to restrict list traffic to the original subscriber, rather than a different person now in possession of an address under which the original subscriber was added to the list. Upon receiving a rejection caused by this specification, the list service can remove that address from further distribution.

A mailing list service that receives a message containing the header field defined here needs to remove it from the message prior to redistributing it, limiting exposure of information regarding the relationship between the message's author and the mailing list.

### 9. Continuous Ownership

For the purposes of this specification, an address is defined as having been under continuous ownership since a given date-time if a message sent to the address at any point since the given date would not go to anyone except the owner at that given date-time. That is, while an address may have been suspended or otherwise disabled for some period, any mail actually delivered would have been delivered exclusively to the same owner. It is presumed that some sort of relationship exists between the message sender and the intended recipient. Presumably there has been some confirmation process applied to establish this ownership of the receiver's mailbox; however, the method of making such determinations is a local matter and outside the scope of this document.

Evaluating the notion of continuous ownership is accomplished by doing any query that establishes whether the above condition holds

for a given mailbox.

Determining continuous ownership of a mailbox is a local matter at the receiving site. The only possible answers to the continuous-ownership-since question are "yes", "no", and "unknown"; the action to be taken in the "unknown" case is a matter of local policy.

For example, when control of a domain name is transferred, the new domain owner might be unable to determine whether the owner of the subject address has been under continuous ownership since the stated date if the mailbox history is not also transferred (or was not previously maintained). It will also be "unknown" if whatever database contains mailbox ownership data is temporarily unavailable at the time a message arrives for delivery. In this latter case, typical SMTP temporary failure handling is appropriate.

To avoid exposing account details unnecessarily, if the address specified has had one continuous owner since it was created, any confirmation date SHOULD be considered to pass the test, even if that date is earlier than the account creation date. This is further discussed in Section 13.

## 10. Digital Signatures

This protocol mandates removal of the header field (when used) upon delivery in all but exceptional circumstances. If a message with the header field were digitally signed in a way that included the header field, altering a message in this way would invalidate the signature. However, the header field is strictly for tunneling purposes and should be regarded by the rest of the transport system as purely trace information.

Accordingly, the header field MUST NOT be included in the content covered by digital signatures.

## 11. Authentication-Results Definitions

[AUTHRES] defines a mechanism for indicating, via a header field, the results of message authentication checks. Section 15 registers RRVS as a new method that can be reported in this way, and corresponding result names. The possible result names and their meanings are as follows:

none: The message had no recipient mailbox timestamp associated with it, either via the SMTP extension or header field method; this protocol was not in use.

unknown: At least one form of this protocol was in use, but continuous ownership of the recipient mailbox could not be determined.

temperror: At least one form of this protocol was in use, but some kind of error occurred during evaluation that was transient in nature; a later retry will likely produce a final result.

permerror: At least one form of this protocol was in use, but some kind of error occurred during evaluation that was not recoverable; a later retry will not likely produce a final result.

pass: At least one form of this protocol was in use, and the destination mailbox was confirmed to have been under continuous ownership since the timestamp thus provided.

fail: At least one form of this protocol was in use, and the destination mailbox was confirmed not to have been under continuous ownership since the timestamp thus provided.

Where multiple recipients are present on a message, multiple results can be reported using the mechanism described in [AUTHRES].

## 12. Examples

In the following examples, "C:" indicates data sent by an SMTP client, and "S:" indicates responses by the SMTP server. Message content is CRLF terminated, though these are omitted here for ease of reading.

### 12.1. SMTP Extension Example

```
C: [connection established]
S: 220 server.example.com ESMTP ready
C: EHLO client.example.net
S: 250-server.example.com
S: 250 RRVS
C: MAIL FROM:<sender@example.net>
S: 250 OK
C: RCPT TO:<receiver@example.com> RRVS=2014-04-03T23:01:00Z
S: 550 5.7.17 receiver@example.com is no longer valid
C: QUIT
S: 221 So long!
```

## 12.2. Header Field Example

```
C: [connection established]
S: 220 server.example.com ESMTP ready
C: HELO client.example.net
S: 250 server.example.com
C: MAIL FROM:<sender@example.net>
S: 250 OK
C: RCPT TO:<receiver@example.com>
S: 250 OK
C: DATA
S: 354 Ready for message content
C: From: Mister Sender <sender@example.net>
  To: Miss Receiver <receiver@example.com>
  Subject: Are you still there?
  Date: Fri, 28 Jun 2013 18:01:01 +0200
  Require-Recipient-Valid-Since: receiver@example.com;
    Sat, 1 Jun 2013 09:23:01 -0700

  Are you still there?
  .
S: 550 5.7.17 receiver@example.com is no longer valid
C: QUIT
S: 221 So long!
```

## 12.3. Authentication-Results Example

An example use of the Authentication-Results header field used to yield the results of an RRVS evaluation:

```
Authentication-Results: mx.example.com; rrvs=pass
                        smtp.rcptto=user@example.com
```

This indicates that the message arrived addressed to the mailbox user@example.com, the continuous ownership test was applied with the provided timestamp, and the check revealed that test was satisfied. The timestamp is not revealed.

## 13. Security Considerations

### 13.1. Abuse Countermeasures

The response of a server implementing this protocol can disclose information about the age of an existing email mailbox. Implementation of countermeasures against probing attacks is RECOMMENDED. For example, an operator could track appearance of this field with respect to a particular mailbox and observe the timestamps

being submitted for testing; if it appears a variety of timestamps is being tried against a single mailbox in short order, the field could be ignored and the message silently discarded. This concern is discussed further in Section 14.

#### 13.2. Suggested Use Restrictions

If the mailbox named in the field is known to have had only a single continuous owner since creation, or not to have existed at all (under any owner) prior to the date specified in the field, then the field SHOULD be silently ignored and normal message handling applied so that this information is not disclosed. Such fields are likely the product of either gross error or an attack.

A message author using this specification might restrict inclusion of the header field such that it is only done for recipients known also to implement this specification, in order to reduce the possibility of revealing information about the relationship between the author and the mailbox.

If ownership of an entire domain is transferred, the new owner may not know what addresses were assigned in the past by the prior owner. Hence, no address can be known not to have had a single owner, or to have existed (or not) at all. In this case, the "unknown" result is likely appropriate.

#### 13.3. False Sense of Security

Senders implementing this protocol likely believe their content is being protected by doing so. It has to be considered, however, that receiving systems might not implement this protocol correctly, or at all. Furthermore, use of RRVs by a sending system constitutes nothing more than a request to the receiving system; that system could choose not to prevent delivery for some local policy, legal or operational reason, which compromises the security the sending system believed was a benefit to using RRVs. This could mean the timestamp information involved in the protocol becomes inadvertently revealed.

This concern lends further support to the notion that senders would do well to avoid using this protocol other than when sending to known, trusted receivers.

#### 13.4. Reassignment of Mailboxes

This specification is a direct response to the risks involved with reassignment or recycling of email addresses, an inherently dangerous practice. It is typically expected that email addresses will not have a high rate of turnover or ownership change.



It is RECOMMENDED to have a substantial period of time between mailbox owners during which the mailbox accepts no mail, giving message generators an opportunity to detect that the previous owner is no longer at that address.

#### 14. Privacy Considerations

##### 14.1. The Trade-Off

That some MSPs allow for expiration of account names when they have been unused for a protracted period forces a choice between two potential types of privacy vulnerabilities, one of which presents significantly greater threats to users than the other. Automatically generated mail is often used to convey authentication credentials that can potentially provide access to extremely sensitive information. Supplying such credentials to the wrong party after a mailbox ownership change could allow the previous owner's data to be exposed without his or her authorization or knowledge. In contrast, the information that may be exposed to a third party via the proposal in this document is limited to information about the mailbox history. Given that MSPs have chosen to allow transfers of mailbox ownership without the prior owner's involvement, the information leakage from the extensions specified here creates far lower overall risk than the potential for delivering mail to the wrong party.

##### 14.2. Probing Attacks

As described above, use of this extension or header field in probing attacks can disclose information about the history of the mailbox. The harm that can be done by leaking any kind of private information is difficult to predict, so it is prudent to be sensitive to this sort of disclosure, either inadvertently or in response to probing by an attacker. It bears restating, then, that implementing countermeasures to abuse of this capability needs strong consideration.

##### 14.3. Envelope Recipients

The email To and Cc header fields are not required to be populated with addresses that match the envelope recipient set, and Cc may even be absent. However, the algorithm in Section 3 requires that this header field contain a match for an envelope recipient in order to be actionable. As such, use of this specification can reveal some or all of the original intended recipient set to any party that can see the message in transit or upon delivery.

For a message destined to a single recipient, this is unlikely to be a concern, which is one of the reasons use of this specification on

multi-recipient messages is discouraged.

#### 14.4. Risks with Use

MDAs might not implement the recommendation to remove the header field defined here when messages are delivered, either out of ignorance or due to error. Since user agents often do not render all of the header fields present, the message could be forwarded to another party that would then inadvertently have the content of this header field.

A bad actor may detect use of either form of the RRVS protocol and interpret it as an indication of high value content.

### 15. IANA Considerations

#### 15.1. SMTP Extension Registration

Section 2.2.2 of [MAIL] sets out the procedure for registering a new SMTP extension. IANA is requested to register the SMTP extension using the details provided in Section 3.1 of this document.

#### 15.2. Header Field Registration

IANA is requested to add the following entry to the Permanent Message Header Field Names registry, as per the procedure found in [IANA-HEADERS]:

Header field name: Require-Recipient-Valid-Since  
Applicable protocol: mail ([MAIL])  
Status: Standard  
Author/Change controller: IETF  
Specification document(s): [this document]  
Related information:  
    Requesting review of any proposed changes and additions to  
    this field is recommended.

#### 15.3. Enhanced Status Code Registration

IANA is requested to register the following in the Enumerated Status Codes table of the Simple Mail Transfer Protocol (SMTP) Enhanced Status Codes Registry:

Code: X.7.17  
Sample Text: Mailbox owner has changed  
Associated basic status code: 5XX  
Description: This status code is returned when a message is received with a Require-Recipient-Valid-Since field or RRVS extension and the receiving system is able to determine that the intended recipient mailbox has not been under continuous ownership since the specified date.  
Reference: [this document]  
Submitter: M. Kucherawy  
Change controller: IESG

Code: X.7.18  
Sample Text: Domain owner has changed  
Associated basic status code: 5XX  
Description: This status code is returned when a message is received with a Require-Recipient-Valid-Since field or RRVS extension and the receiving system wishes to disclose that the owner of the domain name of the recipient has changed since the specified date.  
Reference: [this document]  
Submitter: M. Kucherawy  
Change controller: IESG

Code: X.7.19  
Sample Text: RRVS test cannot be completed  
Associated basic status code: 5XX  
Description: This status code is returned when a message is received with a Require-Recipient-Valid-Since field or RRVS extension and the receiving system cannot complete the requested evaluation because the required timestamp was not recorded. The message originator needs to decide whether to reissue the message without RRVS protection.  
Reference: [this document]  
Submitter: M. Kucherawy  
Change controller: IESG

#### 15.4. Authentication Results Registration

IANA is requested to register the following in the "Email Authentication Methods" Registry:

Method: rrvs

Specifying Document: [this document]

ptype: smtp

Property: rcptto

Value: envelope recipient

Status: active

Version: 1

IANA is also requested to register the following in the "Email Authentication Result Names" Registry:

Codes: none, unknown, temperror, permerror, pass, fail

Defined: [this document]

Auth Method(s): rrvs

Meaning: Section 11 of [this document]

Status: active

## 16. References

### 16.1. Normative References

- |                |   |
|----------------|---|
| [ABNF]         | Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.                       |
| [DATETIME]     | Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.                                      |
| [IANA-HEADERS] | Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004. |
| [KEYWORDS]     | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.                          |
| [MAIL]         | Resnick, P., "Internet Message Format", RFC 5322, October 2008.   |
| [NTP]          | Mills, D., Martin, J., Ed., Burbank, J., and W.   |

Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[ROLES] Crocker, D., "Mailbox Names For Common Services, Roles And Functions", RFC 2142, May 1997.

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

## 16.2. Informative References

[AUTHRES] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 7001, September 2013.

[DSN] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[DSN-SMTP] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[EMAIL-ARCH] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.

[ESC] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

## Appendix A. Acknowledgments

Erling Ellingsen proposed the idea.

Reviews and comments were provided by Michael Adkins, Kurt Andersen, Eric Burger, Alissa Cooper, Dave Cridland, Dave Crocker, Ned Freed, John Levine, Alexey Melnikov, Jay Nancarrow, Hector Santos, Gregg Stefancik, Ed Zayas, (others)

## Authors' Addresses

William J. Mills  
Yahoo! Inc.

EMail: [wmills\\_92105@yahoo.com](mailto:wmills_92105@yahoo.com)

Internet-Draft

Require-Recipient-Valid-Since

April 2014

Murray S. Kucherawy  
Facebook, Inc.  
1 Hacker Way  
Menlo Park, CA 94025  
USA

EMail: msk@fb.com



appsawg  
Internet-Draft  
Updates: 3986 (if approved)  
Intended status: Best Current Practice  
Expires: November 22, 2014

M. Nottingham  
May 21, 2014

URI Design and Ownership  
draft-ietf-appsawg-uri-get-off-my-lawn-05

Abstract

RFC3986 Section 1.1.1 defines URI syntax as "a federated and extensible naming system wherein each scheme's specification may further restrict the syntax and semantics of identifiers using that scheme." In other words, the structure of a URI is defined by its scheme. While it is common for schemes to further delegate their substructure to the URI's owner, publishing independent standards that mandate particular forms of URI substructure is inappropriate, because that essentially usurps ownership. This document further describes this problematic practice and provides some acceptable alternatives for use in standards.

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 November 22, 2014.

Copyright Notice

Copyright (c) 2014 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. Introduction . . . . .	2
1.1. Who This Document Is For . . . . .	3
1.2. Notational Conventions . . . . .	4
2. Best Current Practices for Standardizing Structured URIs . .	4
2.1. URI Schemes . . . . .	4
2.2. URI Authorities . . . . .	5
2.3. URI Paths . . . . .	5
2.4. URI Queries . . . . .	5
2.5. URI Fragment Identifiers . . . . .	6
3. Alternatives to Specifying Structure in URIs . . . . .	6
4. Security Considerations . . . . .	7
5. IANA Considerations . . . . .	7
6. References . . . . .	7
6.1. Normative References . . . . .	7
6.2. Informative References . . . . .	8
Appendix A. Acknowledgments . . . . .	8
Author's Address . . . . .	8

## 1. Introduction

URIs [RFC3986] very often include structured application data. This might include artifacts from filesystems (often occurring in the path component), and user information (often in the query component). In some cases, there can even be application-specific data in the authority component (e.g., some applications are spread across several hostnames to enable a form of partitioning or dispatch).

Furthermore, constraints upon the structure of URIs can be imposed by an implementation; for example, many Web servers use the filename extension of the last path segment to determine the media type of the response. Likewise, pre-packaged applications often have highly structured URIs that can only be changed in limited ways (often, just the hostname and port they are deployed upon).

Because the owner of the URI (as defined in [webarch] Section 2.2.2.1) is choosing to use the server or the application, this can be seen as reasonable delegation of authority. When such conventions are mandated by a party other than the owner, however, it can have several potentially detrimental effects:

- o Collisions - As more ad hoc conventions for URI structure become standardized, it becomes more likely that there will be collisions between them (especially considering that servers, applications and individual deployments will have their own conventions).
- o Dilution - When the information added to a URI is ephemeral, this dilutes its utility by reducing its stability (see [webarch] Section 3.5.1), and can cause several alternate forms of the URI to exist (see [webarch] Section 2.3.1).
- o Rigidity - Fixed URI syntax often interferes with desired deployment patterns. For example, if an authority wishes to offer several applications on a single hostname, it becomes difficult to impossible to do if their URIs do not allow the required flexibility.
- o Operational Difficulty - Supporting some URI conventions can be difficult in some implementations. For example, specifying that a particular query parameter be used with "HTTP" URIs precludes the use of Web servers that serve the response from a filesystem. Likewise, an application that fixes a base path for its operation (e.g., "/v1") makes it impossible to deploy other applications with the same prefix on the same host.
- o Client Assumptions - When conventions are standardized, some clients will inevitably assume that the standards are in use when those conventions are seen. This can lead to interoperability problems; for example, if a specification documents that the "sig" URI query parameter indicates that its payload is a cryptographic signature for the URI, it can lead to undesirable behavior.

Publishing a standard that constrains an existing URI structure in ways which aren't explicitly allowed by [RFC3986] (usually, by updating the URI scheme definition) is inappropriate, because the structure of a URI needs to be firmly under the control of its owner, and the IETF (as well as other organizations) should not usurp it.

This document explains some best current practices for establishing URI structures, conventions and formats in standards. It also offers strategies for specifications to avoid violating these guidelines in Section 3.

### 1.1. Who This Document Is For

This document's requirements target the authors of specifications that constrain the syntax or structure of URIs or parts of them. Two classes of such specifications are called out specifically:

- o Protocol Extensions ("extensions") - specifications that offer new capabilities that could apply to any identifier, or to a large subset of possible identifiers; e.g., a new signature mechanism for 'http' URIs, or metadata for any URI.
- o Applications Using URIs ("applications") - specifications that use URIs to meet specific needs; e.g., a HTTP interface to particular information on a host.

Requirements that target the generic class "Specifications" apply to all specifications, including both those enumerated above and others.

Note that this specification ought not be interpreted as preventing the allocation of control of URIs by parties that legitimately own them, or have delegated that ownership; for example, a specification might legitimately define the semantics of a URI on the IANA.ORG Web site as part of the establishment of a registry.

There may be existing IETF specifications that already deviate from the guidance in this document. In these cases, it is up to the relevant communities (i.e. those of the URI scheme as well as that which produced the specification in question) to determine an appropriate outcome; e.g., updating the scheme definition, or changing the specification.

## 1.2. Notational Conventions

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].

## 2. Best Current Practices for Standardizing Structured URIs

This section updates [RFC3986] by setting limitations on how other specifications may define structure and semantics within URIs. Best practices differ depending on the URI component, as described below.

### 2.1. URI Schemes

Applications and extensions MAY require use of specific URI scheme(s); for example, it is perfectly acceptable to require that an application support 'http' and 'https' URIs. However, applications SHOULD NOT preclude the use of other URI schemes in the future, unless they are clearly only usable with the nominated schemes.

A specification that defines substructure within a specific URI scheme **MUST** do so in the defining document for that URI scheme. A specification that defines substructure for URI schemes overall **MUST** do so by modifying [BCP115] (an exceptional circumstance).

## 2.2. URI Authorities

Scheme definitions define the presence, format and semantics of an authority component in URIs; all other specifications **MUST NOT** constrain, or define the structure or the semantics for URI authorities, unless they update the scheme registration itself.

For example, an extension or application ought not say that the "foo" prefix in "foo\_app.example.com" is meaningful or triggers special handling in URIs.

However, applications **MAY** nominate or constrain the port they use, when applicable. For example, BarApp could run over port nnnn (provided that it is properly registered).

## 2.3. URI Paths

Scheme definitions define the presence, format, and semantics of a path component in URIs; all other specifications **MUST NOT** constrain, or define the structure or the semantics for any path component.

The only exception to this requirement is registered "well-known" URIs, as specified by [RFC5785]. See that document for a description of the applicability of that mechanism.

For example, an application ought not specify a fixed URI path "/"myapp", since this usurps the host's control of that space.

Specifying a fixed path relative to another (e.g., {whatever}/myapp) is also bad practice (even if "whatever" is discovered as suggested in Section 3); while doing so might prevent collisions, it does not avoid the potential for operational difficulties (for example, an implementation that prefers to use query processing instead, because of implementation constraints).

## 2.4. URI Queries

The presence, format and semantics of the query component of URIs is dependent upon many factors, and **MAY** be constrained by a scheme definition. Often, they are determined by the implementation of a resource itself.

Applications MUST NOT directly specify the syntax of queries, as this can cause operational difficulties for deployments that do not support a particular form of a query. For example, a site may wish to support an application using "static" files that do not support query parameters.

Extensions MUST NOT constrain the format or semantics of queries.

For example, an extension that indicates that all query parameters with the name "sig" indicate a cryptographic signature would collide with potentially pre-existing query parameters on sites, and lead clients to assume that any matching query parameter is a signature.

HTML [W3C.REC-html401-19991224] constrains the syntax of query strings used in form submission. New form languages SHOULD NOT emulate it, but instead allow creation of a broader variety of URIs (e.g., by allowing the form to create new path components, and so forth).

Note that "well-known" URIs (see [RFC5785]) MAY constrain their own query syntax, since these name spaces are effectively delegated to the registering party.

## 2.5. URI Fragment Identifiers

Media type definitions (as per [RFC6838]) SHOULD specify the fragment identifier syntax(es) to be used with them; other specifications MUST NOT define structure within the fragment identifier, unless they are explicitly defining one for reuse by media type definitions.

For example, an application that defines common fragment identifiers across media types not controlled by it would engender interoperability problems with handlers for those media types (because the new, non-standard syntax is not expected).

## 3. Alternatives to Specifying Structure in URIs

Given the issues described in Section 1, the most successful strategy for applications and extensions that wish to use URIs is to use them in the fashion they were designed: as links that are exchanged as part of the protocol, rather than statically specified syntax. Several existing specifications can aid in this.

[RFC5988] specifies relation types for Web links. By providing a framework for linking on the Web, where every link has a relation type, context and target, it allows applications to define a link's semantics and connectivity.

[RFC6570] provides a standard syntax for URI Templates that can be used to dynamically insert application-specific variables into a URI to enable such applications while avoiding impinging upon URI owners' control of them.

[RFC5785] allows specific paths to be 'reserved' for standard use on URI schemes that opt into that mechanism ('http' and 'https' by default). Note, however, that this is not a general "escape valve" for applications that need structured URIs; see that specification for more information.

Specifying more elaborate structures in an attempt to avoid collisions is not an acceptable solution, and does not address the issues in Section 1. For example, prefixing query parameters with "myapp\_" does not help, because the prefix itself is subject to the risk of collision (since it is not "reserved").

#### 4. Security Considerations

This document does not introduce new protocol artifacts with security considerations. It prohibits some practices that might lead to vulnerabilities; for example, if a security-sensitive mechanism is introduced by assuming that a URI path component or query string has a particular meaning, false positives might be encountered (due to sites that already use the chosen string). See also [RFC6943].

#### 5. IANA Considerations

There are no direct IANA actions specified in this document.

#### 6. References

##### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.
- [webarch] Jacobs, I. and N. Walsh, "Architecture of the World Wide Web, Volume One", December 2004, <<http://www.w3.org/TR/2004/REC-webarch-20041215>>.

## 6.2. Informative References

- [BCP115] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", RFC 4395, BCP 115, February 2006, <<https://tools.ietf.org/html/bcp115>>.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, April 2010.
- [RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, March 2012.
- [RFC6943] Thaler, D., "Issues in Identifier Comparison for Security Purposes", RFC 6943, May 2013.
- [W3C.REC-html401-19991224] Raggett, D., Hors, A., and I. Jacobs, "HTML 4.01 Specification", World Wide Web Consortium Recommendation REC-html401-19991224, December 1999, <<http://www.w3.org/TR/1999/REC-html401-19991224>>.

## Appendix A. Acknowledgments

Thanks to David Booth, Dave Crocker, Tim Bray, Anne van Kesteren, Martin Thomson, Erik Wilde, Dave Thaler and Barry Leiba for their suggestions and feedback.

## Author's Address

Mark Nottingham

Email: [mnot@mnot.net](mailto:mnot@mnot.net)

URI: <http://www.mnot.net/>

DANE  
Internet-Draft  
Intended status: Standards Track  
Expires: November 30, 2015

V. Dukhovni  
Two Sigma  
W. Hardaker  
Parsons  
May 29, 2015

SMTP security via opportunistic DANE TLS  
draft-ietf-dane-smtp-with-dane-19

Abstract

This memo describes a downgrade-resistant protocol for SMTP transport security between Mail Transfer Agents (MTAs) based on the DNS-Based Authentication of Named Entities (DANE) TLSA DNS record. Adoption of this protocol enables an incremental transition of the Internet email backbone to one using encrypted and authenticated Transport Layer Security (TLS).

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 November 30, 2015.

Copyright Notice

Copyright (c) 2015 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. Introduction . . . . .	3
1.1. Terminology . . . . .	3
1.2. Background . . . . .	5
1.3. SMTP channel security . . . . .	6
1.3.1. STARTTLS downgrade attack . . . . .	6
1.3.2. Insecure server name without DNSSEC . . . . .	7
1.3.3. Sender policy does not scale . . . . .	8
1.3.4. Too many certification authorities . . . . .	8
2. Identifying applicable TLSA records . . . . .	8
2.1. DNS considerations . . . . .	9
2.1.1. DNS errors, bogus and indeterminate responses . . . . .	9
2.1.2. DNS error handling . . . . .	11
2.1.3. Stub resolver considerations . . . . .	11
2.2. TLS discovery . . . . .	12
2.2.1. MX resolution . . . . .	14
2.2.2. Non-MX destinations . . . . .	15
2.2.3. TLSA record lookup . . . . .	17
3. DANE authentication . . . . .	19
3.1. TLSA certificate usages . . . . .	19
3.1.1. Certificate usage DANE-EE(3) . . . . .	20
3.1.2. Certificate usage DANE-TA(2) . . . . .	21
3.1.3. Certificate usages PKIX-TA(0) and PKIX-EE(1) . . . . .	22
3.2. Certificate matching . . . . .	23
3.2.1. DANE-EE(3) name checks . . . . .	23
3.2.2. DANE-TA(2) name checks . . . . .	23
3.2.3. Reference identifier matching . . . . .	24
4. Server key management . . . . .	25
5. Digest algorithm agility . . . . .	26
6. Mandatory TLS Security . . . . .	26
7. Note on DANE for Message User Agents . . . . .	27
8. Interoperability considerations . . . . .	27
8.1. SNI support . . . . .	27
8.2. Anonymous TLS cipher suites . . . . .	28
9. Operational Considerations . . . . .	28
9.1. Client Operational Considerations . . . . .	28
9.2. Publisher Operational Considerations . . . . .	29
10. Security Considerations . . . . .	29
11. IANA considerations . . . . .	30
12. Acknowledgements . . . . .	30
13. References . . . . .	30
13.1. Normative References . . . . .	30
13.2. Informative References . . . . .	32
Authors' Addresses . . . . .	32

## 1. Introduction

This memo specifies a new connection security model for Message Transfer Agents (MTAs). This model is motivated by key features of inter-domain SMTP delivery, in particular the fact that the destination server is selected indirectly via DNS Mail Exchange (MX) records and that neither email addresses nor MX hostnames signal a requirement for either secure or cleartext transport. Therefore, aside from a few manually configured exceptions, SMTP transport security is of necessity opportunistic (for a definition of "Opportunistic Security" see [RFC7435]).

This specification uses the presence of DANE TLSA records to securely signal TLS support and to publish the means by which SMTP clients can successfully authenticate legitimate SMTP servers. This becomes "opportunistic DANE TLS" and is resistant to downgrade and man-in-the-middle (MITM) attacks. It enables an incremental transition of the email backbone to authenticated TLS delivery, with increased global protection as adoption increases.

With opportunistic DANE TLS, traffic from SMTP clients to domains that publish "usable" DANE TLSA records in accordance with this memo is authenticated and encrypted. Traffic from legacy clients or to domains that do not publish TLSA records will continue to be sent in the same manner as before, via manually configured security, (pre-DANE) opportunistic TLS or just cleartext SMTP.

Problems with existing use of TLS in MTA to MTA SMTP that motivate this specification are described in Section 1.3. The specification itself follows in Section 2 and Section 3 which describe respectively how to locate and use DANE TLSA records with SMTP. In Section 6, we discuss application of DANE TLS to destinations for which channel integrity and confidentiality are mandatory. In Section 7 we briefly comment on potential applicability of this specification to Message User Agents.

### 1.1. Terminology

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 following terms or concepts are used through the document:

Man-in-the-middle or MITM attack: Active modification of network traffic by an adversary able to thereby compromise the confidentiality or integrity of the data.

Downgrade attack: (From [RFC4949]). A type of man-in-the-middle attack in which the attacker can cause two parties, at the time they negotiate a security association, to agree on a lower level of protection than the highest level that could have been supported by both of them.

Downgrade-resistant: A protocol is "downgrade-resistant" if it employs effective counter-measures against downgrade attacks.

secure, bogus, insecure, indeterminate: DNSSEC validation results, as defined in Section 4.3 of [RFC4035].

Validating Security-Aware Stub Resolver and Non-Validating Security-Aware Stub Resolver:

Capabilities of the stub resolver in use as defined in [RFC4033]; note that this specification requires the use of a Security-Aware Stub Resolver.

(pre-DANE) opportunistic TLS: Best-effort use of TLS that is generally vulnerable to DNS forgery and STARTTLS downgrade attacks. When a TLS-encrypted communication channel is not available, message transmission takes place in the clear. MX record indirection generally precludes authentication even when TLS is available.

opportunistic DANE TLS: Best-effort use of TLS, resistant to downgrade attacks for destinations with DNSSEC-validated TLSA records. When opportunistic DANE TLS is determined to be unavailable, clients should fall back to opportunistic TLS. Opportunistic DANE TLS requires support for DNSSEC, DANE and STARTTLS on the client side and STARTTLS plus a DNSSEC published TLSA record on the server side.

reference identifier: (Special case of [RFC6125] definition). One of the domain names associated by the SMTP client with the destination SMTP server for performing name checks on the server certificate. When name checks are applicable, at least one of the reference identifiers MUST match an [RFC6125] DNS-ID (or if none are present the [RFC6125] CN-ID) of the server certificate (see Section 3.2.3).

MX hostname: The RRDATA of an MX record consists of a 16 bit preference followed by a Mail Exchange domain name (see [RFC1035], Section 3.3.9). We will use the term "MX hostname" to refer to the latter, that is, the DNS domain name found after the preference value in an MX record. Thus an "MX hostname" is specifically a reference to a DNS domain name, rather than any host that bears that name.

delayed delivery: Email delivery is a multi-hop store & forward process. When an MTA is unable to forward a message that may become deliverable later the message is queued and delivery is retried periodically. Some MTAs may be configured with a fallback next-hop destination that handles messages that the MTA would otherwise queue and retry. When a fallback next-hop is configured, messages that would otherwise have to be delayed may be sent to the fallback next-hop destination instead. The fallback destination may itself be subject to opportunistic or mandatory DANE TLS (Section 6) as though it were the original message destination.

original next hop destination: The logical destination for mail delivery. By default this is the domain portion of the recipient address, but MTAs may be configured to forward mail for some or all recipients via designated relays. The original next hop destination is, respectively, either the recipient domain or the associated configured relay.

MTA: Message Transfer Agent ([RFC5598], Section 4.3.2).

MSA: Message Submission Agent ([RFC5598], Section 4.3.1).

MUA: Message User Agent ([RFC5598], Section 4.2.1).

RR: A DNS Resource Record as defined in [RFC1034], Section 3.6.

RRSet: An RRSet ([RFC2181], Section 5) is a group of DNS resource records that share the same label, class and type.

## 1.2. Background

The Domain Name System Security Extensions (DNSSEC) add data origin authentication, data integrity and data non-existence proofs to the Domain Name System (DNS). DNSSEC is defined in [RFC4033], [RFC4034] and [RFC4035].

As described in the introduction of [RFC6698], TLS authentication via the existing public Certification Authority (CA) PKI suffers from an over-abundance of trusted parties capable of issuing certificates for any domain of their choice. DANE leverages the DNSSEC infrastructure to publish public keys and certificates for use with the Transport Layer Security (TLS) [RFC5246] protocol via the "TLSA" DNS record type. With DNSSEC each domain can only vouch for the keys of its delegated sub-domains.

The TLS protocol enables secure TCP communication. In the context of this memo, channel security is assumed to be provided by TLS. Used

without authentication, TLS provides only privacy protection against eavesdropping attacks. Otherwise, TLS also provides data origin authentication to guard against MITM attacks.

### 1.3. SMTP channel security

With HTTPS, Transport Layer Security (TLS) employs X.509 certificates [RFC5280] issued by one of the many Certification Authorities (CAs) bundled with popular web browsers to allow users to authenticate their "secure" websites. Before we specify a new DANE TLS security model for SMTP, we will explain why a new security model is needed. In the process, we will explain why the familiar HTTPS security model is inadequate to protect inter-domain SMTP traffic.

The subsections below outline four key problems with applying traditional Web PKI to SMTP that are addressed by this specification. Since SMTP channel security policy is not explicitly specified in either the recipient address or the MX record, a new signaling mechanism is required to indicate when channel security is possible and should be used. The publication of TLSA records allows server operators to securely signal to SMTP clients that TLS is available and should be used. DANE TLSA makes it possible to simultaneously discover which destination domains support secure delivery via TLS and how to verify the authenticity of the associated SMTP services, providing a path forward to ubiquitous SMTP channel security.

#### 1.3.1. STARTTLS downgrade attack

The Simple Mail Transfer Protocol (SMTP) [RFC5321] is a single-hop protocol in a multi-hop store & forward email delivery process. An SMTP envelope recipient address does not correspond to a specific transport-layer endpoint address, rather at each relay hop the transport-layer endpoint is the next-hop relay, while the envelope recipient address typically remains the same. Unlike the Hypertext Transfer Protocol (HTTP) and its corresponding secured version, HTTPS, where the use of TLS is signaled via the URI scheme, email recipient addresses do not directly signal transport security policy. Indeed, no such signaling could work well with SMTP since TLS encryption of SMTP protects email traffic on a hop-by-hop basis while email addresses could only express end-to-end policy.

With no mechanism available to signal transport security policy, SMTP relays employ a best-effort "opportunistic" security model for TLS. A single SMTP server TCP listening endpoint can serve both TLS and non-TLS clients; the use of TLS is negotiated via the SMTP STARTTLS command ([RFC3207]). The server signals TLS support to the client over a cleartext SMTP connection, and, if the client also supports TLS, it may negotiate a TLS encrypted channel to use for email

transmission. The server's indication of TLS support can be easily suppressed by an MITM attacker. Thus pre-DANE SMTP TLS security can be subverted by simply downgrading a connection to cleartext. No TLS security feature can prevent this. The attacker can simply disable TLS.

### 1.3.2. Insecure server name without DNSSEC

With SMTP, DNS Mail Exchange (MX) records abstract the next-hop transport endpoint and allow administrators to specify a set of target servers to which SMTP traffic should be directed for a given domain.

A TLS client is vulnerable to MITM attacks unless it verifies that the server's certificate binds the public key to a name that matches one of the client's reference identifiers. A natural choice of reference identifier is the server's domain name. However, with SMTP, server names are not directly encoded in the recipient address, instead they are obtained indirectly via MX records. Without DNSSEC, the MX lookup is vulnerable to MITM and DNS cache poisoning attacks. Active attackers can forge DNS replies with fake MX records and can redirect email to servers with names of their choice. Therefore, secure verification of SMTP TLS certificates matching the server name is not possible without DNSSEC.

One might try to harden TLS for SMTP against DNS attacks by using the envelope recipient domain as a reference identifier and by requiring each SMTP server to possess a trusted certificate for the envelope recipient domain rather than the MX hostname. Unfortunately, this is impractical as email for many domains is handled by third parties that are not in a position to obtain certificates for all the domains they serve. Deployment of the Server Name Indication (SNI) extension to TLS (see [RFC6066] Section 3) is no panacea, since SNI key management is operationally challenging except when the email service provider is also the domain's registrar and its certificate issuer; this is rarely the case for email.

Since the recipient domain name cannot be used as the SMTP server reference identifier, and neither can the MX hostname without DNSSEC, large-scale deployment of authenticated TLS for SMTP requires that the DNS be secure.

Since SMTP security depends critically on DNSSEC, it is important to point out that consequently SMTP with DANE is the most conservative possible trust model. It trusts only what must be trusted and no more. Adding any other trusted actors to the mix can only reduce SMTP security. A sender may choose to further harden DNSSEC for selected high-value receiving domains by configuring explicit trust

anchors for those domains instead of relying on the chain of trust from the root domain. However, detailed discussion of DNSSEC security practices is out of scope for this document.

#### 1.3.3. Sender policy does not scale

Sending systems are in some cases explicitly configured to use TLS for mail sent to selected peer domains, but this requires configuring sending MTAs with appropriate subject names or certificate content digests from their peer domains. Due to the resulting administrative burden, such statically configured SMTP secure channels are used rarely (generally only between domains that make bilateral arrangements with their business partners). Internet email, on the other hand, requires regularly contacting new domains for which security configurations cannot be established in advance.

The abstraction of the SMTP transport endpoint via DNS MX records, often across organization boundaries, limits the use of public CA PKI with SMTP to a small set of sender-configured peer domains. With little opportunity to use TLS authentication, sending MTAs are rarely configured with a comprehensive list of trusted CAs. SMTP services that support STARTTLS often deploy X.509 certificates that are self-signed or issued by a private CA.

#### 1.3.4. Too many certification authorities

Even if it were generally possible to determine a secure server name, the SMTP client would still need to verify that the server's certificate chain is issued by a trusted Certification Authority (a trust anchor). MTAs are not interactive applications where a human operator can make a decision (wisely or otherwise) to selectively disable TLS security policy when certificate chain verification fails. With no user to "click OK", the MTA's list of public CA trust anchors would need to be comprehensive in order to avoid bouncing mail addressed to sites that employ unknown Certification Authorities.

On the other hand, each trusted CA can issue certificates for any domain. If even one of the configured CAs is compromised or operated by an adversary, it can subvert TLS security for all destinations. Any set of CAs is simultaneously both overly inclusive and not inclusive enough.

## 2. Identifying applicable TLSA records

## 2.1. DNS considerations

### 2.1.1. DNS errors, bogus and indeterminate responses

An SMTP client that implements opportunistic DANE TLS per this specification depends critically on the integrity of DNSSEC lookups, as discussed in Section 1.3.2. This section lists the DNS resolver requirements needed to avoid downgrade attacks when using opportunistic DANE TLS.

A DNS lookup may signal an error or return a definitive answer. A security-aware resolver MUST be used for this specification. Security-aware resolvers will indicate the security status of a DNS RRSset with one of four possible values defined in Section 4.3 of [RFC4035]: "secure", "insecure", "bogus" and "indeterminate". In [RFC4035] the meaning of the "indeterminate" security status is:

An RRSset for which the resolver is not able to determine whether the RRSset should be signed, as the resolver is not able to obtain the necessary DNSSEC RRs. This can occur when the security-aware resolver is not able to contact security-aware name servers for the relevant zones.

Note, the "indeterminate" security status has a conflicting definition in section 5 of [RFC4033].

There is no trust anchor that would indicate that a specific portion of the tree is secure.

In this document the term "indeterminate" will be used exclusively in the [RFC4035] sense. Therefore, obtaining "indeterminate" lookup results is a (transient) failure condition, namely, the inability to locate the relevant DNS records. DNS records that would be classified "indeterminate" in the sense of [RFC4035] are simply classified as "insecure".

We do not need to distinguish between zones that lack a suitable ancestor trust anchor, and delegations (ultimately) from a trust-anchor that designate a child zone as being "insecure". All "insecure" RRSets MUST be handled identically: in either case unvalidated data for the query domain is all that is and can be available, and authentication using the data is impossible. As the DNS root zone has been signed, we expect that validating resolvers used by Internet-facing MTAs will be configured with trust anchor data for the root zone, and that therefore domains with no ancestor trust anchor will not be possible in most deployments.



As noted in section 4.3 of [RFC4035], a security-aware DNS resolver MUST be able to determine whether a given non-error DNS response is "secure", "insecure", "bogus" or "indeterminate". It is expected that most security-aware stub resolvers will not signal an "indeterminate" security status (in the sense of RFC4035) to the application, and will signal a "bogus" or error result instead. If a resolver does signal an RFC4035 "indeterminate" security status, this MUST be treated by the SMTP client as though a "bogus" or error result had been returned.

An MTA making use of a non-validating security-aware stub resolver MAY use the stub resolver's ability, if available, to signal DNSSEC validation status based on information the stub resolver has learned from an upstream validating recursive resolver. Security-Oblivious stub-resolvers ([RFC4033]) MUST NOT be used. In accordance with section 4.9.3 of [RFC4035]:

... a security-aware stub resolver MUST NOT place any reliance on signature validation allegedly performed on its behalf, except when the security-aware stub resolver obtained the data in question from a trusted security-aware recursive name server via a secure channel.

To avoid much repetition in the text below, we will pause to explain the handling of "bogus" or "indeterminate" DNSSEC query responses. These are not necessarily the result of a malicious actor; they can, for example, occur when network packets are corrupted or lost in transit. Therefore, "bogus" or "indeterminate" replies are equated in this memo with lookup failure.

There is an important non-failure condition we need to highlight in addition to the obvious case of the DNS client obtaining a non-empty "secure" or "insecure" RRSset of the requested type. Namely, it is not an error when either "secure" or "insecure" non-existence is determined for the requested data. When a DNSSEC response with a validation status that is either "secure" or "insecure" reports either no records of the requested type or non-existence of the query domain, the response is not a DNS error condition. The DNS client has not been left without an answer; it has learned that records of the requested type do not exist.

Security-aware stub resolvers will, of course, also signal DNS lookup errors in other cases, for example when processing a "ServFail" RCODE, which will not have an associated DNSSEC status. All lookup errors are treated the same way by this specification, regardless of whether they are from a "bogus" or "indeterminate" DNSSEC status or from a more generic DNS error: the information that was requested cannot be obtained by the security-aware resolver at this time. A

lookup error is thus a failure to obtain the relevant RRSset if it exists, or to determine that no such RRsset exists when it does not.

In contrast to a "bogus" or an "indeterminate" response, an "insecure" DNSSEC response is not an error, rather, as explained above, it indicates that the target DNS zone is either delegated as an "insecure" child of a "secure" parent zone, or is not a descendant of any of the configured DNSSEC trust anchors in use by the SMTP client. "Insecure" results will leave the SMTP client with degraded channel security, but do not stand in the way of message delivery. See section Section 2.2 for further details.

#### 2.1.2. DNS error handling

When a DNS lookup failure (error or "bogus" or "indeterminate" as defined above) prevents an SMTP client from determining which SMTP server or servers it should connect to, message delivery MUST be delayed. This naturally includes, for example, the case when a "bogus" or "indeterminate" response is encountered during MX resolution. When multiple MX hostnames are obtained from a successful MX lookup, but a later DNS lookup failure prevents network address resolution for a given MX hostname, delivery may proceed via any remaining MX hosts.

When a particular SMTP server is securely identified as the delivery destination, a set of DNS lookups (Section 2.2) MUST be performed to locate any related TLSA records. If any DNS queries used to locate TLSA records fail (be it due to "bogus" or "indeterminate" records, timeouts, malformed replies, ServFails, etc.), then the SMTP client MUST treat that server as unreachable and MUST NOT deliver the message via that server. If no servers are reachable, delivery is delayed.

In what follows, we will only describe what happens when all relevant DNS queries succeed. If any DNS failure occurs, the SMTP client MUST behave as described in this section, by skipping the problem SMTP server, or the problem destination. Queries for candidate TLSA records are explicitly part of "all relevant DNS queries" and SMTP clients MUST NOT continue to connect to an SMTP server or destination whose TLSA record lookup fails.

#### 2.1.3. Stub resolver considerations

A note about DNAME aliases: a query for a domain name whose ancestor domain is a DNAME alias returns the DNAME RR for the ancestor domain along with a CNAME that maps the query domain to the corresponding sub-domain of the target domain of the DNAME alias [RFC6672]. Therefore, whenever we speak of CNAME aliases, we implicitly allow

for the possibility that the alias in question is the result of an ancestor domain DNAME record. Consequently, no explicit support for DNAME records is needed in SMTP software; it is sufficient to process the resulting CNAME aliases. DNAME records only require special processing in the validating stub-resolver library that checks the integrity of the combined DNAME + CNAME reply. When DNSSEC validation is handled by a local caching resolver, rather than the MTA itself, even that part of the DNAME support logic is outside the MTA.

When a stub resolver returns a response containing a CNAME alias that does not also contain the corresponding query results for the target of the alias, the SMTP client will need to repeat the query at the target of the alias, and should do so recursively up to some configured or implementation-dependent recursion limit. If at any stage of CNAME expansion an error is detected, the lookup of the original requested records MUST be considered to have failed.

Whether a chain of CNAME records was returned in a single stub resolver response or via explicit recursion by the SMTP client, if at any stage of recursive expansion an "insecure" CNAME record is encountered, then it and all subsequent results (in particular, the final result) MUST be considered "insecure" regardless of whether any earlier CNAME records leading to the "insecure" record were "secure".

Note that a security-aware non-validating stub resolver may return to the SMTP client an "insecure" reply received from a validating recursive resolver that contains a CNAME record along with additional answers recursively obtained starting at the target of the CNAME. In this case, the only possible conclusion is that some record in the set of records returned is "insecure", and it is in fact possible that the initial CNAME record and a subset of the subsequent records are "secure".

If the SMTP client needs to determine the security status of the DNS zone containing the initial CNAME record, it will need to issue a separate query of type "CNAME" that returns only the initial CNAME record. Specifically, in Section 2.2.2 when insecure A or AAAA records are found for an SMTP server via a CNAME alias, the SMTP client will need to perform an additional CNAME query in order to determine whether the DNS zone in which the alias is published is DNSSEC signed.

## 2.2. TLS discovery

As noted previously (in Section 1.3.1), opportunistic TLS with SMTP servers that advertise TLS support via STARTTLS is subject to an MITM downgrade attack. Also some SMTP servers that are not, in fact, TLS

capable erroneously advertise STARTTLS by default and clients need to be prepared to retry cleartext delivery after STARTTLS fails. In contrast, DNSSEC validated TLSA records MUST NOT be published for servers that do not support TLS. Clients can safely interpret their presence as a commitment by the server operator to implement TLS and STARTTLS.

This memo defines four actions to be taken after the search for a TLSA record returns secure usable results, secure unusable results, insecure or no results or an error signal. The term "usable" in this context is in the sense of Section 4.1 of [RFC6698]. Specifically, if the DNS lookup for a TLSA record returns:

A secure TLSA RRSset with at least one usable record: Any connection to the MTA MUST employ TLS encryption and MUST authenticate the SMTP server using the techniques discussed in the rest of this document. Failure to establish an authenticated TLS connection MUST result in falling back to the next SMTP server or delayed delivery.

A secure non-empty TLSA RRsset where all the records are unusable: Any connection to the MTA MUST be made via TLS, but authentication is not required. Failure to establish an encrypted TLS connection MUST result in falling back to the next SMTP server or delayed delivery.

An insecure TLSA RRsset or DNSSEC validated proof-of-non-existent TLSA records:

A connection to the MTA SHOULD be made using (pre-DANE) opportunistic TLS, this includes using cleartext delivery when the remote SMTP server does not appear to support TLS. The MTA MAY retry in cleartext when delivery via TLS fails either during the handshake or even during data transfer.

Any lookup error: Lookup errors, including "bogus" and "indeterminate", as explained in Section 2.1.1 MUST result in falling back to the next SMTP server or delayed delivery.

An SMTP client MAY be configured to mandate DANE verified delivery for some destinations. With mandatory DANE TLS (Section 6), delivery proceeds only when "secure" TLSA records are used to establish an encrypted and authenticated TLS channel with the SMTP server.

When the original next-hop destination is an address literal, rather than a DNS domain, DANE TLS does not apply. Delivery proceeds using any relevant security policy configured by the MTA administrator. Similarly, when an MX RRsset incorrectly lists a network address in lieu of an MX hostname, if an MTA chooses to connect to the network

address in the non-conformant MX record, DANE TLSA does not apply for such a connection.

In the subsections that follow we explain how to locate the SMTP servers and the associated TLSA records for a given next-hop destination domain. We also explain which name or names are to be used in identity checks of the SMTP server certificate.

#### 2.2.1. MX resolution

In this section we consider next-hop domains that are subject to MX resolution and have MX records. The TLSA records and the associated base domain are derived separately for each MX hostname that is used to attempt message delivery. DANE TLS can authenticate message delivery to the intended next-hop domain only when the MX records are obtained securely via a DNSSEC validated lookup.

MX records MUST be sorted by preference; an MX hostname with a worse (numerically higher) MX preference that has TLSA records MUST NOT preempt an MX hostname with a better (numerically lower) preference that has no TLSA records. In other words, prevention of delivery loops by obeying MX preferences MUST take precedence over channel security considerations. Even with two equal-preference MX records, an MTA is not obligated to choose the MX hostname that offers more security. Domains that want secure inbound mail delivery need to ensure that all their SMTP servers and MX records are configured accordingly.

In the language of [RFC5321] Section 5.1, the original next-hop domain is the "initial name". If the MX lookup of the initial name results in a CNAME alias, the MTA replaces the initial name with the resulting name and performs a new lookup with the new name. MTAs typically support recursion in CNAME expansion, so this replacement is performed repeatedly (up to the MTA's recursion limit) until the ultimate non-CNAME domain is found.

If the MX RRSet (or any CNAME leading to it) is "insecure" (see Section 2.1.1) and DANE TLS for the given destination is mandatory (Section 6), delivery MUST be delayed. If the MX RRSet is "insecure" and DANE TLS is not mandatory, the SMTP client is free to use pre-DANE opportunistic TLS (possibly even cleartext).

Since the protocol in this memo is an "opportunistic security" protocol ([RFC7435]) the SMTP client MAY elect to use DANE TLS (as described in Section 2.2.2 below) even with MX hosts obtained via an "insecure" MX RRSet. For example, when a hosting provider has a signed DNS zone and publishes TLSA records for its SMTP servers, hosted domains that are not signed may still benefit from the

provider's TLSA records. Deliveries via the provider's SMTP servers will not be subject to active attacks when sending SMTP clients elect to make use of the provider's TLSA records (active attacks that tamper with the "insecure" MX RRSet are of course still possible in this case).

When the MX records are not (DNSSEC) signed, an active attacker can redirect SMTP clients to MX hosts of his choice. Such redirection is tamper-evident when SMTP servers found via "insecure" MX records are recorded as the next-hop relay in the MTA delivery logs in their original (rather than CNAME expanded) form. Sending MTAs SHOULD log unexpanded MX hostnames when these result from insecure MX lookups. Any successful authentication via an insecurely determined MX host MUST NOT be misrepresented in the mail logs as secure delivery to the intended next-hop domain.

In the absence of DNS lookup errors (Section 2.1.1), if the MX RRSet is not "insecure" then it is "secure", and the SMTP client MUST treat each MX hostname as described in Section 2.2.2). When, for a given MX hostname, no TLSA records are found, or only "insecure" TLSA records are found, DANE TLSA is not applicable with the SMTP server in question and delivery proceeds to that host as with pre-DANE opportunistic TLS. To avoid downgrade attacks, any errors during TLSA lookups MUST, as explained in Section 2.1.1, cause the SMTP server in question to be treated as unreachable.

#### 2.2.2. Non-MX destinations

This section describes the algorithm used to locate the TLSA records and associated TLSA base domain for an input domain that is not subject to MX resolution, that represents a hostname from a secure MX RRSet, or that lacks MX records. Such domains include:

- o Any host configured by the sending MTA administrator as the next-hop relay for some or all domains, that is not subject to MX resolution.
- o When a domain has MX records, we treat each MX host listed in those MX records as though it were a non-MX destination. That is, in the same way as we would treat an administrator-configured relay that handles mail for that domain. (Unlike administrator-specified relays, MTAs are not required to support CNAME expansion of next-hop names found via MX lookups).
- o A next-hop destination domain subject to MX resolution that has no MX records. In this case the domain's name is implicitly also its sole SMTP server name.

Note that DNS queries with type TLSA are mishandled by load balancing nameservers that serve the MX hostnames of some large email providers. The DNS zones served by these nameservers are not signed and contain no TLSA records, but queries for TLSA records fail, rather than returning the non-existence of the requested TLSA records.

To avoid problems delivering mail to domains whose SMTP servers are served by the problem nameservers the SMTP client MUST perform any A and/or AAAA queries for the destination before attempting to locate the associated TLSA records. This lookup is needed in any case to determine whether the destination domain is reachable and the DNSSEC validation status of the chain of CNAME queries required to reach the ultimate address records.

If no address records are found, the destination is unreachable. If address records are found, but the DNSSEC validation status of the first query response is "insecure" (see Section 2.1.3), the SMTP client SHOULD NOT proceed to search for any associated TLSA records. With the problem domains, TLSA queries will lead to DNS lookup errors and cause messages to be consistently delayed and ultimately returned to the sender. We don't expect to find any "secure" TLSA records associated with a TLSA base domain that lies in an unsigned DNS zone. Therefore, skipping TLSA lookups in this case will also reduce latency with no detrimental impact on security.

If the A and/or AAAA lookup of the "initial name" yields a CNAME, we replace it with the resulting name as if it were the initial name and perform a lookup again using the new name. This replacement is performed recursively (up to the MTA's recursion limit).

We consider the following cases for handling a DNS response for an A or AAAA DNS lookup:

Not found: When the DNS queries for A and/or AAAA records yield neither a list of addresses nor a CNAME (or CNAME expansion is not supported) the destination is unreachable.

Non-CNAME: The answer is not a CNAME alias. If the address RRSset is "secure", TLSA lookups are performed as described in Section 2.2.3 with the initial name as the candidate TLSA base domain. If no "secure" TLSA records are found, DANE TLS is not applicable and mail delivery proceeds with pre-DANE opportunistic TLS (which, being best-effort, degrades to cleartext delivery when STARTTLS is not available or the TLS handshake fails).

Insecure CNAME: The input domain is a CNAME alias, but the ultimate network address RRSset is "insecure" (see Section 2.1.1). If the

initial CNAME response is also "insecure", DANE TLS does not apply. Otherwise, this case is treated just like the non-CNAME case above, where a search is performed for a TLSA record with the original input domain as the candidate TLSA base domain.

**Secure CNAME:** The input domain is a CNAME alias, and the ultimate network address RRSset is "secure" (see Section 2.1.1). Two candidate TLSA base domains are tried: the fully CNAME-expanded initial name and, failing that, then the initial name itself.

In summary, if it is possible to securely obtain the full, CNAME-expanded, DNSSEC-validated address records for the input domain, then that name is the preferred TLSA base domain. Otherwise, the unexpanded input-MX domain is the candidate TLSA base domain. When no "secure" TLSA records are found at either the CNAME-expanded or unexpanded domain, then DANE TLS does not apply for mail delivery via the input domain in question. And, as always, errors, bogus or indeterminate results for any query in the process MUST result in delaying or abandoning delivery.

### 2.2.3. TLSA record lookup

When the SMTP server's hostname is not a CNAME or DNAME alias, the list of associated candidate TLSA base domains (see below) consists of just the server hostname.

When hostname is an alias with a "secure" (at every stage) full expansion, the list of candidate TLSA base domains (see below) is a pair of domains: the fully expanded server hostname first and the unexpanded server hostname second.

Each candidate TLSA base domain (alias-expanded or original) is in turn prefixed with service labels of the form "\_<port>.\_tcp". The resulting domain name is used to issue a DNSSEC query with the query type set to TLSA ([RFC6698] Section 7.1).

The first of these candidate domains to yield a "secure" TLSA RRSset becomes the actual TLSA base domain.

For SMTP, the destination TCP port is typically 25, but this may be different with custom routes specified by the MTA administrator in which case the SMTP client MUST use the appropriate number in the "\_<port>" prefix in place of "\_25". If, for example, the candidate base domain is "mx.example.com", and the SMTP connection is to port 25, the TLSA RRsset is obtained via a DNSSEC query of the form:

\_25.\_tcp.mx.example.com. IN TLSA ?



The query response may be a CNAME, or the actual TLSA RRSset. If the response is a CNAME, the SMTP client (through the use of its security-aware stub resolver) restarts the TLSA query at the target domain, following CNAMEs as appropriate and keeps track of whether the entire chain is "secure". If any "insecure" records are encountered, or the TLSA records don't exist, the next candidate TLSA base domain is tried instead.

If the ultimate response is a "secure" TLSA RRSset, then the candidate TLSA base domain will be the actual TLSA base domain and the TLSA RRSset will constitute the TLSA records for the destination. If none of the candidate TLSA base domains yield "secure" TLSA records then the SMTP client is free to use pre-DANE opportunistic TLS (possibly even cleartext).

TLSA record publishers may leverage CNAMEs to reference a single authoritative TLSA RRSset specifying a common Certification Authority or a common end-entity certificate to be used with multiple TLS services. Such CNAME expansion does not change the SMTP client's notion of the TLSA base domain; thus, when `_25._tcp.mx.example.com` is a CNAME, the base domain remains `mx.example.com` and this is still the reference identifier used together with the next-hop domain in peer certificate name checks.

Note that shared end-entity certificate associations expose the publishing domain to substitution attacks, where an MITM attacker can reroute traffic to a different server that shares the same end-entity certificate. Such shared end-entity TLSA records SHOULD be avoided unless the servers in question are functionally equivalent or employ mutually incompatible protocols (an active attacker gains nothing by diverting client traffic from one such server to another).

A better example, employing a shared trust anchor rather than shared end-entity certificates, is illustrated by the DNSSEC validated records below:

```
example.com.           IN MX 0 mx1.example.com.
example.com.           IN MX 0 mx2.example.com.
_25._tcp.mx1.example.com. IN CNAME tlsa201._dane.example.com.
_25._tcp.mx2.example.com. IN CNAME tlsa201._dane.example.com.
tlsa201._dane.example.com. IN TLSA 2 0 1 e3b0c44298fc1c149a...
```

The SMTP servers `mx1.example.com` and `mx2.example.com` will be expected to have certificates issued under a common trust anchor, but each MX hostname's TLSA base domain remains unchanged despite the above CNAME records. Correspondingly, each SMTP server will be associated with a pair of reference identifiers consisting of its hostname plus the next-hop domain "example.com".

If, during TLSA resolution (including possible CNAME indirection), at least one "secure" TLSA record is found (even if not usable because it is unsupported by the implementation or support is administratively disabled), then the corresponding host has signaled its commitment to implement TLS. The SMTP client MUST NOT deliver mail via the corresponding host unless a TLS session is negotiated via STARTTLS. This is required to avoid MITM STARTTLS downgrade attacks.

As noted previously (in Section 2.2.2), when no "secure" TLSA records are found at the fully CNAME-expanded name, the original unexpanded name MUST be tried instead. This supports customers of hosting providers where the provider's zone cannot be validated with DNSSEC, but the customer has shared appropriate key material with the hosting provider to enable TLS via SNI. Intermediate names that arise during CNAME expansion that are neither the original, nor the final name, are never candidate TLSA base domains, even if "secure".

### 3. DANE authentication

This section describes which TLSA records are applicable to SMTP opportunistic DANE TLS and how to apply such records to authenticate the SMTP server. With opportunistic DANE TLS, both the TLS support implied by the presence of DANE TLSA records and the verification parameters necessary to authenticate the TLS peer are obtained together. In contrast to protocols where channel security policy is set exclusively by the client, authentication via this protocol is expected to be less prone to connection failure caused by incompatible configuration of the client and server.

#### 3.1. TLSA certificate usages

The DANE TLSA specification [RFC6698] defines multiple TLSA RR types via combinations of 3 numeric parameters. The numeric values of these parameters were later given symbolic names in [RFC7218]. The rest of the TLSA record is the "certificate association data field", which specifies the full or digest value of a certificate or public key.

Since opportunistic DANE TLS will be used by non-interactive MTAs, with no user to "press OK" when authentication fails, reliability of peer authentication is paramount. Server operators are advised to publish TLSA records that are least likely to fail authentication due to interoperability or operational problems. Because DANE TLS relies on coordinated changes to DNS and SMTP server settings, the best choice of records to publish will depend on site-specific practices.

The certificate usage element of a TLSA record plays a critical role in determining how the corresponding certificate association data field is used to authenticate server's certificate chain. The next two subsections explain the process for certificate usages DANE-EE(3) and DANE-TA(2). The third subsection briefly explains why certificate usages PKIX-TA(0) and PKIX-EE(1) are not applicable with opportunistic DANE TLS.

In summary, we RECOMMEND the use of "DANE-EE(3) SPKI(1) SHA2-256(1)", with "DANE-TA(2) Cert(0) SHA2-256(1)" TLSA records as a second choice, depending on site needs. See the following two subsections for more details. Other combinations of TLSA parameters are either explicitly unsupported, or offer little to recommend them over these two.

### 3.1.1. Certificate usage DANE-EE(3)

Authentication via certificate usage DANE-EE(3) TLSA records involves simply checking that the server's leaf certificate matches the TLSA record. In particular the binding of the server public key to its name is based entirely on the TLSA record association. The server MUST be considered authenticated even if none of the names in the certificate match the client's reference identity for the server.

The expiration date of the server certificate MUST be ignored: the validity period of the TLSA record key binding is determined by the validity interval of the TLSA record DNSSEC signature.

With DANE-EE(3), servers need not employ SNI (they may ignore the client's SNI message) even when the server is known under independent names that would otherwise require separate certificates. It is instead sufficient for the TLSA RRsets for all the domains in question to match the server's default certificate. Of course with SMTP servers it is simpler still to publish the same MX hostname for all the hosted domains.

For domains where it is practical to make coordinated changes in DNS TLSA records during SMTP server key rotation, it is often best to publish end-entity DANE-EE(3) certificate associations. DANE-EE(3) certificates don't suddenly stop working when leaf or intermediate certificates expire, and don't fail when the server operator neglects to configure all the required issuer certificates in the server certificate chain.

TLSA records published for SMTP servers SHOULD, in most cases, be "DANE-EE(3) SPKI(1) SHA2-256(1)" records. Since all DANE implementations are required to support SHA2-256, this record type

works for all clients and need not change across certificate renewals with the same key.

### 3.1.2. Certificate usage DANE-TA(2)

Some domains may prefer to avoid the operational complexity of publishing unique TLSA RRs for each TLS service. If the domain employs a common issuing Certification Authority to create certificates for multiple TLS services, it may be simpler to publish the issuing authority as a trust anchor (TA) for the certificate chains of all relevant services. The TLSA query domain (TLSA base domain with port and protocol prefix labels) for each service issued by the same TA may then be set to a CNAME alias that points to a common TLSA RRSet that matches the TA. For example:

```
example.com.           IN MX 0 mx1.example.com.
example.com.           IN MX 0 mx2.example.com.
_25._tcp.mx1.example.com. IN CNAME tlsa201._dane.example.com.
_25._tcp.mx2.example.com. IN CNAME tlsa201._dane.example.com.
tlsa201._dane.example.com. IN TLSA 2 0 1 e3b0c44298fc1c14....
```

With usage DANE-TA(2) the server certificates will need to have names that match one of the client's reference identifiers (see [RFC6125]). The server MAY employ SNI to select the appropriate certificate to present to the client.

SMTP servers that rely on certificate usage DANE-TA(2) TLSA records for TLS authentication MUST include the TA certificate as part of the certificate chain presented in the TLS handshake server certificate message even when it is a self-signed root certificate. Many SMTP servers are not configured with a comprehensive list of trust anchors, nor are they expected to at any point in the future. Some MTAs will ignore all locally trusted certificates when processing usage DANE-TA(2) TLSA records. Thus even when the TA happens to be a public Certification Authority known to the SMTP client, authentication is likely to fail unless the TA certificate is included in the TLS server certificate message.

With some SMTP server software it is not possible to configure the server to include self-signed (root) CA certificates in the server certificate chain. Such servers either MUST publish DANE-TA(2) records for an intermediate certificate or MUST instead use DANE-EE(3) TLSA records.

TLSA records with matching type Full(0) are discouraged. While these potentially obviate the need to transmit the TA certificate in the TLS server certificate message, client implementations may not be able to augment the server certificate chain with the data obtained

from DNS, especially when the TLSA record supplies a bare key (selector SPKI(1)). Since the server will need to transmit the TA certificate in any case, server operators SHOULD publish TLSA records with a matching type other than Full(0) and avoid potential interoperability issues with large TLSA records containing full certificates or keys.

TLSA Publishers employing DANE-TA(2) records SHOULD publish records with a selector of Cert(0). Such TLSA records are associated with the whole trust anchor certificate, not just with the trust anchor public key. In particular, the SMTP client SHOULD then apply any relevant constraints from the trust anchor certificate, such as, for example, path length constraints.

While a selector of SPKI(1) may also be employed, the resulting TLSA record will not specify the full trust anchor certificate content, and elements of the trust anchor certificate other than the public key become mutable. This may, for example, allow a subsidiary CA to issue a chain that violates the trust anchor's path length or name constraints.

### 3.1.3. Certificate usages PKIX-TA(0) and PKIX-EE(1)

Note, this section applies to MTA-to-MTA SMTP, which is normally on port 25. That is, to servers that are the SMTP servers for one or more destination domains. Other uses of SMTP, such as in MUA-to-MSA submission on ports 587 or 465 are out of scope for this document. Where those other uses also employ TLS opportunistically and/or depend on DNSSEC as a result of DNS-based discovery of service location, the relevant specifications should, as appropriate, arrive at similar conclusions.

As noted in Section 1.3.1 and Section 1.3.2, sending MTAs cannot, without relying on DNSSEC for secure MX records and DANE for STARTTLS support signaling, perform server identity verification or prevent STARTTLS downgrade attacks. The use of PKIX CAs offers no added security since an attacker capable of compromising DNSSEC is free to replace any PKIX-TA(0) or PKIX-EE(1) TLSA records with records bearing any convenient non-PKIX certificate usage. Finally, as explained in Section 1.3.4, there is no list of trusted CAs agreed by all MTAs, and no user to "click OK" when a server's CA is not trusted by a client.

Therefore, TLSA records for the port 25 SMTP service used by client MTAs SHOULD NOT include TLSA RRs with certificate usage PKIX-TA(0) or PKIX-EE(1). SMTP client MTAs cannot be expected to be configured with a suitably complete set of trusted public CAs. Lacking a complete set of public CAs, MTA clients would not be able to verify

the certificates of SMTP servers whose issuing root CAs are not trusted by the client.

Opportunistic DANE TLS needs to interoperate without bilateral coordination of security settings between client and server systems. Therefore, parameter choices that are fragile in the absence of bilateral coordination are unsupported. Nothing is lost since the PKIX certificate usages cannot aid SMTP TLS security, they can only impede SMTP TLS interoperability.

SMTP client treatment of TLSA RRs with certificate usages PKIX-TA(0) or PKIX-EE(1) is undefined. As with any other unsupported certificate usage, SMTP clients MAY treat such records as "unusable".

### 3.2. Certificate matching

When at least one usable "secure" TLSA record is found, the SMTP client MUST use TLSA records to authenticate the SMTP server. Messages MUST NOT be delivered via the SMTP server if authentication fails, otherwise the SMTP client is vulnerable to MITM attacks.

#### 3.2.1. DANE-EE(3) name checks

The SMTP client MUST NOT perform certificate name checks with certificate usage DANE-EE(3); see Section 3.1.1 above.

#### 3.2.2. DANE-TA(2) name checks

To match a server via a TLSA record with certificate usage DANE-TA(2), the client MUST perform name checks to ensure that it has reached the correct server. In all DANE-TA(2) cases the SMTP client MUST employ the TLSA base domain as the primary reference identifier for matching the server certificate.

TLSA records for MX hostnames: If the TLSA base domain was obtained indirectly via a "secure" MX lookup (including any CNAME-expanded name of an MX hostname), then the original next-hop domain used in the MX lookup MUST be included as a second reference identifier. The CNAME-expanded original next-hop domain MUST be included as a third reference identifier if different from the original next-hop domain. When the client MTA is employing DANE TLS security despite "insecure" MX redirection the MX hostname is the only reference identifier.

TLSA records for Non-MX hostnames: If MX records were not used (e.g., if none exist) and the TLSA base domain is the CNAME-expanded original next-hop domain, then the original next-hop domain MUST be included as a second reference identifier.

Accepting certificates with the original next-hop domain in addition to the MX hostname allows a domain with multiple MX hostnames to field a single certificate bearing a single domain name (i.e., the email domain) across all the SMTP servers. This also aids interoperability with pre-DANE SMTP clients that are configured to look for the email domain name in server certificates. For example, with "secure" DNS records as below:

```
exchange.example.org.      IN CNAME mail.example.org.
mail.example.org.          IN CNAME example.com.
example.com.               IN MX      10 mx10.example.com.
example.com.               IN MX      15 mx15.example.com.
example.com.               IN MX      20 mx20.example.com.
;
mx10.example.com.          IN A        192.0.2.10
_25._tcp.mx10.example.com. IN TLSA    2 0 1 ...
;
mx15.example.com.          IN CNAME mxbackup.example.com.
mxbackup.example.com.      IN A        192.0.2.15
; _25._tcp.mxbackup.example.com. IN TLSA    ? (NXDOMAIN)
_25._tcp.mx15.example.com. IN TLSA    2 0 1 ...
;
mx20.example.com.          IN CNAME mxbackup.example.net.
mxbackup.example.net.      IN A        198.51.100.20
_25._tcp.mxbackup.example.net. IN TLSA    2 0 1 ...
```

Certificate name checks for delivery of mail to exchange.example.org via any of the associated SMTP servers MUST accept at least the names "exchange.example.org" and "example.com", which are respectively the original and fully expanded next-hop domain. When the SMTP server is mx10.example.com, name checks MUST accept the TLSA base domain "mx10.example.com". If, despite the fact that MX hostnames are required to not be aliases, the MTA supports delivery via "mx15.example.com" or "mx20.example.com" then name checks MUST accept the respective TLSA base domains "mx15.example.com" and "mxbackup.example.net".

### 3.2.3. Reference identifier matching

When name checks are applicable (certificate usage DANE-TA(2)), if the server certificate contains a Subject Alternative Name extension ([RFC5280]), with at least one DNS-ID ([RFC6125]) then only the DNS-IDs are matched against the client's reference identifiers. The CN-ID ([RFC6125]) is only considered when no DNS-IDs are present. The server certificate is considered matched when one of its presented identifiers ([RFC5280]) matches any of the client's reference identifiers.

Wildcards are valid in either DNS-IDs or the CN-ID when applicable. The wildcard character must be the entire first label of the DNS-ID or CN-ID. Thus, "\*.example.com" is valid, while "smtp\*.example.com" and "\*smtp.example.com" are not. SMTP clients MUST support wildcards that match the first label of the reference identifier, with the remaining labels matching verbatim. For example, the DNS-ID "\*.example.com" matches the reference identifier "mx1.example.com". SMTP clients MAY, subject to local policy allow wildcards to match multiple reference identifier labels, but servers cannot expect broad support for such a policy. Therefore any wildcards in server certificates SHOULD match exactly one label in either the TLSA base domain or the next-hop domain.

#### 4. Server key management

Two TLSA records MUST be published before employing a new EE or TA public key or certificate, one matching the currently deployed key and the other matching the new key scheduled to replace it. Once sufficient time has elapsed for all DNS caches to expire the previous TLSA RRSset and related signature RRSets, servers may be configured to use the new EE private key and associated public key certificate or may employ certificates signed by the new trust anchor.

Once the new public key or certificate is in use, the TLSA RR that matches the retired key can be removed from DNS, leaving only RRs that match keys or certificates in active use.

As described in Section 3.1.2, when server certificates are validated via a DANE-TA(2) trust anchor, and CNAME records are employed to store the TA association data at a single location, the responsibility of updating the TLSA RRSset shifts to the operator of the trust anchor. Before a new trust anchor is used to sign any new server certificates, its certificate (digest) is added to the relevant TLSA RRSset. After enough time elapses for the original TLSA RRSset to age out of DNS caches, the new trust anchor can start issuing new server certificates. Once all certificates issued under the previous trust anchor have expired, its associated RRs can be removed from the TLSA RRSset.

In the DANE-TA(2) key management model server operators do not generally need to update DNS TLSA records after initially creating a CNAME record that references the centrally operated DANE-TA(2) RRSset. If a particular server's key is compromised, its TLSA CNAME SHOULD be replaced with a DANE-EE(3) association until the certificate for the compromised key expires, at which point it can return to using a CNAME record. If the central trust anchor is compromised, all servers need to be issued new keys by a new TA, and an updated DANE-TA(2) TLSA RRSset needs to be published containing just the new TA.



SMTP servers cannot expect broad CRL or OCSP support from SMTP clients. As outlined above, with DANE, compromised server or trust anchor keys can be "revoked" by removing them from the DNS without the need for client-side support for OCSP or CRLs.

#### 5. Digest algorithm agility

While [RFC6698] specifies multiple digest algorithms, it does not specify a protocol by which the SMTP client and TLSA record publisher can agree on the strongest shared algorithm. Such a protocol would allow the client and server to avoid exposure to any deprecated weaker algorithms that are published for compatibility with less capable clients, but should be ignored when possible. Such a protocol is specified in [I-D.ietf-dane-ops]. SMTP clients and servers that implement this specification MUST comply with the requirements outlined under "Digest Algorithm Agility" in [I-D.ietf-dane-ops].

#### 6. Mandatory TLS Security

An MTA implementing this protocol may require a stronger security assurance when sending email to selected destinations. The sending organization may need to send sensitive email and/or may have regulatory obligations to protect its content. This protocol is not in conflict with such a requirement, and in fact can often simplify authenticated delivery to such destinations.

Specifically, with domains that publish DANE TLSA records for their MX hostnames, a sending MTA can be configured to use the receiving domains's DANE TLSA records to authenticate the corresponding SMTP server. Authentication via DANE TLSA records is easier to manage, as changes in the receiver's expected certificate properties are made on the receiver end and don't require manually communicated configuration changes. With mandatory DANE TLS, when no usable TLSA records are found, message delivery is delayed. Thus, mail is only sent when an authenticated TLS channel is established to the remote SMTP server.

Administrators of mail servers that employ mandatory DANE TLS need to carefully monitor their mail logs and queues. If a partner domain unwittingly misconfigures their TLSA records, disables DNSSEC, or misconfigures SMTP server certificate chains, mail will be delayed and may bounce if the issue is not resolved in a timely manner.

## 7. Note on DANE for Message User Agents

We note that the SMTP protocol is also used between Message User Agents (MUAs) and Message Submission Agents (MSAs) [RFC6409]. In [RFC6186] a protocol is specified that enables an MUA to dynamically locate the MSA based on the user's email address. SMTP connection security considerations for MUAs implementing [RFC6186] are largely analogous to connection security requirements for MTAs, and this specification could be applied largely verbatim with DNS MX records replaced by corresponding DNS Service (SRV) records [I-D.ietf-dane-srv].

However, until MUAs begin to adopt the dynamic configuration mechanisms of [RFC6186] they are adequately served by more traditional static TLS security policies. Specification of DANE TLS for Message User Agent (MUA) to Message Submission Agent (MSA) SMTP is left to future documents that focus specifically on SMTP security between MUAs and MSAs.

## 8. Interoperability considerations

### 8.1. SNI support

To ensure that the server sends the right certificate chain, the SMTP client MUST send the TLS SNI extension containing the TLSA base domain. This precludes the use of the SSL 2.0 compatible SSL HELLO by the SMTP client.

Each SMTP server MUST present a certificate chain (see [RFC5246] Section 7.4.2) that matches at least one of the TLSA records. The server MAY rely on SNI to determine which certificate chain to present to the client. Clients that don't send SNI information may not see the expected certificate chain.

If the server's TLSA records match the server's default certificate chain, the server need not support SNI. In either case, the server need not include the SNI extension in its TLS HELLO as simply returning a matching certificate chain is sufficient. Servers MUST NOT enforce the use of SNI by clients, as the client may be using unauthenticated opportunistic TLS and may not expect any particular certificate from the server. If the client sends no SNI extension, or sends an SNI extension for an unsupported domain, the server MUST simply send some fallback certificate chain of its choice. The reason for not enforcing strict matching of the requested SNI hostname is that DANE TLS clients are typically willing to accept multiple server names, but can only send one name in the SNI extension. The server's fallback certificate may match a different name acceptable to the client, e.g., the original next-hop domain.

## 8.2. Anonymous TLS cipher suites

Since many SMTP servers either do not support or do not enable any anonymous TLS cipher suites, SMTP client TLS HELLO messages SHOULD offer to negotiate a typical set of non-anonymous cipher suites required for interoperability with such servers. An SMTP client employing pre-DANE opportunistic TLS MAY in addition include one or more anonymous TLS cipher suites in its TLS HELLO. SMTP servers, that need to interoperate with opportunistic TLS clients SHOULD be prepared to interoperate with such clients by either always selecting a mutually supported non-anonymous cipher suite or by correctly handling client connections that negotiate anonymous cipher suites.

Note that while SMTP server operators are under no obligation to enable anonymous cipher suites, no security is gained by sending certificates to clients that will ignore them. Indeed support for anonymous cipher suites in the server makes audit trails more informative. Log entries that record connections that employed an anonymous cipher suite record the fact that the clients did not care to authenticate the server.

## 9. Operational Considerations

### 9.1. Client Operational Considerations

An operational error on the sending or receiving side that cannot be corrected in a timely manner may, at times, lead to consistent failure to deliver time-sensitive email. The sending MTA administrator may have to choose between letting email queue until the error is resolved and disabling opportunistic or mandatory DANE TLS (Section 6) for one or more destinations. The choice to disable DANE TLS security should not be made lightly. Every reasonable effort should be made to determine that problems with mail delivery are the result of an operational error, and not an attack. A fallback strategy may be to configure explicit out-of-band TLS security settings if supported by the sending MTA.

SMTP clients may deploy opportunistic DANE TLS incrementally by enabling it only for selected sites, or may occasionally need to disable opportunistic DANE TLS for peers that fail to interoperate due to misconfiguration or software defects on either end. Some implementations MAY support DANE TLS in an "audit only" mode in which failure to achieve the requisite security level is logged as a warning and delivery proceeds at a reduced security level. Unless local policy specifies "audit only" or that opportunistic DANE TLS is not to be used for a particular destination, an SMTP client MUST NOT deliver mail via a server whose certificate chain fails to match at

least one TLSA record when usable TLSA records are found for that server.

## 9.2. Publisher Operational Considerations

Some MTAs enable STARTTLS selectively. For example they might only support STARTTLS with clients that have previously demonstrated "proper MTA behavior", for example by retrying the delivery of deferred messages (greylisting). If such an MTA publishes DANE TLSA records, sending MTAs that implement this specification will not attempt the initial cleartext SMTP transaction needed to establish the "proper MTA behavior", because they cannot establish the required channel security. Server operators MUST NOT implement selective STARTTLS if they also want to support DANE TLSA.

TLSA Publishers MUST follow the guidelines in the "TLSA Publisher Requirements" section of [I-D.ietf-dane-ops].

TLSA Publishers SHOULD follow the TLSA publication size guidance found in [I-D.ietf-dane-ops] under "DANE DNS Record Size Guidelines".

TLSA Publishers SHOULD follow the TLSA record TTL and signature lifetime recommendations found in [I-D.ietf-dane-ops] under "Operational Considerations".

## 10. Security Considerations

This protocol leverages DANE TLSA records to implement MITM resistant opportunistic security ([RFC7435]) for SMTP. For destination domains that sign their MX records and publish signed TLSA records for their MX hostnames, this protocol allows sending MTAs to securely discover both the availability of TLS and how to authenticate the destination.

This protocol does not aim to secure all SMTP traffic, as that is not practical until DNSSEC and DANE adoption are universal. The incremental deployment provided by following this specification is a best possible path for securing SMTP. This protocol coexists and interoperates with the existing insecure Internet email backbone.

The protocol does not preclude existing non-opportunistic SMTP TLS security arrangements, which can continue to be used as before via manual configuration with negotiated out-of-band key and TLS configuration exchanges.

Opportunistic SMTP TLS depends critically on DNSSEC for downgrade resistance and secure resolution of the destination name. If DNSSEC is compromised, it is not possible to fall back on the public CA PKI to prevent MITM attacks. A successful breach of DNSSEC enables the

attacker to publish TLSA usage 3 certificate associations, and thereby bypass any security benefit the legitimate domain owner might hope to gain by publishing usage 0 or 1 TLSA RRs. Given the lack of public CA PKI support in existing MTA deployments, avoiding certificate usages 0 and 1 simplifies implementation and deployment with no adverse security consequences.

Implementations must strictly follow the portions of this specification that indicate when it is appropriate to initiate a non-authenticated connection or cleartext connection to a SMTP server. Specifically, in order to prevent downgrade attacks on this protocol, implementation must not initiate a connection when this specification indicates a particular SMTP server must be considered unreachable.

## 11. IANA considerations

This specification requires no support from IANA.

## 12. Acknowledgements

The authors would like to extend great thanks to Tony Finch, who started the original version of a DANE SMTP document. His work is greatly appreciated and has been incorporated into this document. The authors would like to additionally thank Phil Pennock for his comments and advice on this document.

Acknowledgments from Viktor: Thanks to Paul Hoffman who motivated me to begin work on this memo and provided feedback on early drafts. Thanks to Patrick Koetter, Perry Metzger and Nico Williams for valuable review comments. Thanks also to Wietse Venema who created Postfix, and whose advice and feedback were essential to the development of the Postfix DANE implementation.

## 13. References

### 13.1. Normative References

- [I-D.ietf-dane-ops]  
Dukhovni, V. and W. Hardaker, "Updates to and Operational Guidance for the DANE Protocol", draft-ietf-dane-ops-09 (work in progress), May 2015.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [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.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.
- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 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.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, July 2009.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.
- [RFC6672] Rose, S. and W. Wijngaards, "DNAME Redirection in the DNS", RFC 6672, June 2012.
- [RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

- [RFC7218] Gudmundsson, O., "Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE)", RFC 7218, April 2014.

### 13.2. Informative References

- [I-D.ietf-dane-srv]  
Finch, T., Miller, M., and P. Saint-Andre, "Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records", draft-ietf-dane-srv-14 (work in progress), April 2015.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.
- [RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, December 2014.

### Authors' Addresses

Viktor Dukhovni  
Two Sigma

Email: [ietf-dane@dukhovni.org](mailto:ietf-dane@dukhovni.org)

Wes Hardaker  
Parsons  
P.O. Box 382  
Davis, CA 95617  
US

Email: [ietf@hardakers.net](mailto:ietf@hardakers.net)

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: April 24, 2014

K. Moore  
Network Heretics  
October 21, 2013

Recommendations for use of TLS by Electronic Mail Access Protocols  
draft-moore-email-tls-00

## Abstract

This memo requires support for Transport Layer Security (TLS) in all electronic mail user agents (MUAs) and the servers with which they communicate when using standard protocols, including Interactive Message Access Protocol (IMAP), Post Office Protocol (POP) and the variant of the Simple Message Transfer Protocol (SMTP) used in message submission. It also requires support for TLS in mail protocol servers provided by electronic mail service providers, and encourages mail service providers to migrate to requiring TLS for all interaction with their servers. In addition, this memo details specific recommendations for implementation and use of TLS with electronic mail protocols used in interactions between MUAs and mail service providers.

Use of TLS with SMTP for message relaying is described in a separate document, and not in scope for this document.

The recommendations in this memo do not replace the functionality of, and are not intended as a substitute for, end-to-end encryption of electronic mail.

## 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 April 24, 2014.



## Copyright Notice

Copyright (c) 2013 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. Introduction . . . . .	3
1.1. Definitions . . . . .	4
1.2. Goals and Rationale . . . . .	5
1.3. Approach . . . . .	7
2. Implementation Requirements . . . . .	8
2.1. Mail Server Requirements . . . . .	8
2.2. Mail User Agent Requirements . . . . .	9
2.2.1. MUAs Configurable to Require TLS . . . . .	9
2.2.2. Non-configurable MUAs and nonstandard access protocols . . . . .	9
2.2.3. Implicit TLS vs. STARTTLS . . . . .	10
2.2.4. Use of SRV records in Establishing Configuration . . . . .	10
2.2.5. Manual configuration of MUA connection to servers . . . . .	11
2.2.6. Verification of new or edited server configurations . . . . .	11
2.2.7. Downgrading of TLS-required Configurations . . . . .	12
2.2.8. Requirements for MUA use of TLS . . . . .	12
2.2.9. Use of SMTP by MUAs for other than mail submission . . . . .	13
2.2.10. Other network-accessible services used by MUAs . . . . .	13
2.2.11. Additional Considerations for Webmail and other Split-MUA Clients . . . . .	14
2.2.12. Use of DANE by MUAs . . . . .	14
2.2.13. Use of DNSSEC . . . . .	15
2.3. Requirements Common To Both Servers and MUAs . . . . .	16
3. Mail Service Provider Requirements . . . . .	16
3.1. Server Requirements . . . . .	16
3.2. MSPs MUST provide Submission Servers . . . . .	16
3.3. TLS Server Certificate Requirements . . . . .	16
3.4. Recommended DNS records for mail protocol servers . . . . .	17
3.4.1. MX records . . . . .	17
3.4.2. SRV records . . . . .	17
3.4.3. TLSA records . . . . .	17

3.4.4. DNSSEC . . . . .	18
3.5. MSP Server Monitoring . . . . .	18
3.6. Encourage Transition to TLS Required Configurations . . . .	18
4. Security Considerations . . . . .	18
5. IANA Considerations . . . . .	19
6. References . . . . .	20
6.1. Normative References . . . . .	20
6.2. Informative References . . . . .	21

## 1. Introduction

Most Internet electronic mail protocols, including SMTP Submission Protocol [RFC4409], Interactive Message Access Protocol (IMAP) [RFC3501], and Post Office Protocol (POP) [RFC1939], were originally designed to transmit all authentication credentials, commands, and application data in cleartext only. At the time that those protocols were originally designed, encryption was computationally expensive, and/or not widely available due to export limitations and other constraints. In the earliest days of these protocols, it was also typical that Internet service was provided through hardwired hosts and networks, which provided some degree of security against eavesdropping by limiting physical access to the server hosts and network media.

Recently it has become apparent that the potential for eavesdropping of electronic mail traffic has increased for a variety of reasons, including: "rogue" wireless LAN access points that monitor traffic, industrial espionage, and government-supported espionage by a variety of governments. For these reasons it now seems prudent to recommend a much wider use of TLS encryption than has been conventional in the past.

In brief, this memo now recommends that:

- o TLS on a well-known port ("Implicit TLS") be supported for Interactive Message Access Protocol (IMAP), Post Office Protocol (POP), and SMTP Submission protocol for all electronic mail user agents and servers;
- o Electronic mail user agents (MUAs) require TLS for all newly configured connections to servers, unless explicitly configured by their users to not require TLS;
- o When explicitly configuring an MUA to not require TLS, the MUA warn users that their mail traffic is insecure;
- o Electronic mail service providers (MSPs) support use of Implicit TLS for IMAP, POP, and SMTP Submission; and

- o MSPs encourage new users to configure their MUAs to require TLS when connecting to their servers, and encourage existing users to transition to MUA configurations that require TLS, using mechanisms appropriate for their user communities.

This document therefore defines profiles of the above protocols which impose additional requirements beyond those in the base protocol specifications. Specific details of these requirements, and additional requirements, are outlined below.

### 1.1. Definitions

**Implicit TLS** - The practice of automatically negotiating a TLS layer as soon as a TCP connection is established between client and server, on a TCP port configured on that server to perform such negotiation. This port may be assigned by IANA for that purpose, advertised by DNS SRV record, or used by private agreement between client and server. (See also STARTTLS mechanism)

**Interactive Message Access Protocol (IMAP)** - The protocol defined in [RFC3501] which is used for accessing and managing received electronic mail. This memo will also refer to "IMAP client" and "IMAP server" when appropriate.

**mail account** - A set of services provided by a Mail Service Provider for a particular sender and/or recipient, which may include (among others): mail submission, access to delivered mail, management of delivered mail, configuration of incoming mail filters, management of authentication credentials. A mail account will generally be implemented with a variety of protocol servers, for example IMAP, POP, Submission, and/or a webmail service, but will usually share a common set of authentication credentials across all of those servers.

**Mail User Agent (MUA)** - A client that performs one or more of the following: (a) submits electronic mail for delivery, (b) accesses mail delivered to one or more mailboxes, and/or (c) manages mail delivered to one or more mailboxes, on behalf of one or more (human or nonhuman) users. An MUA may function as any of an IMAP client, POP client, Submission client, or SMTP client, among other roles.

**Mail Service Provider (MSP)** - A provider of electronic mail services including (a) submission of outgoing mail and/or (b) acceptance of incoming mail and providing recipients with the ability to access that mail. In this memo, the term Mail Service Provider applies not only to providers that offer such services to the public (whether for "free" or in exchange for monetary remuneration), but also to providers of mail services to private communities, including business enterprises.

Opportunistic TLS - The practice of negotiating TLS when it appears to a TLS-capable client that the server also supports TLS, but continuing the intended operation in cleartext when it appears to the client that the server does not support TLS.

pinning - The act of establishing a cached name association between the application service's certificate and one of the client's reference identifiers, whether or not any of the certificate's presented identifiers matches one of the client's reference identifiers. (See also section 1.8 of [RFC6125].)

Post Office Protocol (POP) - The protocol defined in [RFC1939] which is used for accessing and managing received electronic mail. Since POP is a client-server protocol, this memo will refer to POP client and POP server when appropriate.

presented identifier - Any of the identifiers presented to a client in a validated TLS server certificate. (See also section 1.8 of [RFC6125].)

reference identifier - Any of a set of identifiers pre-determined by a TLS client to be acceptable identifiers for a particular service, to be matched against the presented identifiers from the server's certificate. (See also section 1.8 of [RFC6125].)

STARTTLS mechanism - One of the protocol extensions defined in [RFC2595] or [RFC3207] for negotiating TLS after a cleartext application layer connection between client and server have already been established. (See also Implicit TLS.)

Submission protocol - the variant of SMTP defined in [RFC6409] and used exclusively for submission of outgoing messages by MUAs.

Transport Layer Security (TLS) - The protocol defined in [RFC5246] and its revisions for providing security services over a TCP stream.

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].

## 1.2. Goals and Rationale

This memo is one of several with the shared goal of encouraging use of strong encryption for all uses of Internet electronic mail protocols, and thus reducing the effectiveness of mass surveillance which is known to be conducted on a large scale by several parties. Other memos address other aspects of this problem, including opportunistic encryption for relayed mail using SMTP

[I-D.ietf-dane-smtp-with-dane], and improving TLS server identity checks [I-D.melnikov-email-tls-certs].

The primary goal for this memo is to encourage a much wider adoption of reliable encryption for email protocol traffic between Mail User Agents (MUAs) and mail servers. "Reliable encryption" means that a user can have confidence that his mail traffic is securely encrypted when it travels over the network. By contrast, if the traffic is not encrypted, the user should be made explicitly aware of this. Since TLS is the Internet standards-track encryption mechanism which is most widely implemented in email clients and servers, is well-maintained, and believed to be sufficiently extensible to accommodate newly identified threats and use cases, TLS is the mechanism specified for providing such a reliable encryption service.

Note: The goal of "reliable encryption" is a distinct goal from, and in contrast with, a goal to encrypt as much traffic as possible. Encrypting as much traffic as possible could be accomplished using Opportunistic TLS. However, this would not be the same as "reliable encryption", as it would not provide the user with assurance that his traffic is encrypted. It also appears that there are several ways in which Opportunistic TLS can easily be defeated by an attacker. So while in some sense encrypting as much traffic as possible is also a worthy goal, reliable encryption appears to be more important. Only reliable encryption provides protection in the case of an active attack.

In furtherance of the goal of reliable encryption, a number of new requirements are imposed on mail protocol engines. However, an additional goal of this memo is to facilitate continued operation between legacy clients and servers that meet the requirements in this memo, and between legacy servers and clients that meet the requirements in this memo. Another part of that goal is to facilitate such continued operation while providing an "upgrade path" such that the vast majority of clients and servers should be able to be modified to meet these requirements within a short time, without disruption of service or significant support costs.

An additional goal of this memo is to discourage exposure of reusable authentication credentials (such as passwords) over an unencrypted channel when using IMAP, POP, or SMTP Submission, or any other protocol for which the same credentials are used as with one of the above protocols.

It is explicitly not a goal of this memo to provide any assurance of either end-to-end encryption (from submission to delivery), or encryption of delivered email that has been stored in a mailbox. Unless additionally encrypted by other means such as S/MIME

[RFC33369], email messages will still be available in cleartext on each client and server that processes or stores those messages.

### 1.3. Approach

The basic approach is to recommend that TLS and the Implicit TLS mechanism be used for all interactions between MUAs and servers: that all MUAs and servers support TLS and Implicit TLS, that MUAs use TLS by default for all newly configured server connections unless explicitly configured otherwise by their users, and that mail service providers encourage existing clients to upgrade to MUAs that support TLS and upgrade existing MUA configurations to require TLS.

After much consideration, TLS over a well-known port ("Implicit TLS") is recommended instead of the STARTTLS mechanism, for the following reasons:

- o It appears to be the desired end-state. In a future world where TLS were always used, there would no longer be a need for the STARTTLS mechanism. Even if it were still necessary for some MSPs to continue to support cleartext operation for legacy or very lightweight clients, all MUAs capable of using TLS could eventually be expected to migrate to configurations using Implicit TLS.
- o Implicit TLS capability is discoverable using SRV records as described in [RFC6186], whereas discovering STARTTLS capability requires opening a connection to the server.
- o Use of Implicit TLS appears to be less susceptible to both MUA misconfiguration and to unintended downgrading to cleartext operation, even for legacy MUAs. If an MUA's configuration explicitly specifies either use of TLS or use of the well-known port assigned by IANA for use with Implicit TLS (often termed the "SSL port"), it seems unlikely that the MUA will downgrade to the "non-SSL port" under any circumstances, even if the server is unreachable or TLS negotiation fails. In addition, if a mail service provider advertises Implicit TLS as its preferred mechanism to connect to servers (via SRV records and/or human-readable documentation), the mail service provider can defeat automatic downgrading to cleartext operation by MUAs (even with legacy MUAs) simply by not providing a working server that supports cleartext operation on the same IP address recommended for use with new configurations. (Cleartext access for existing users and configurations can still be maintained on the existing IP address.)

- o In earlier unpublished drafts of this memo, the author attempted to recommend STARTTLS in preference to Implicit TLS. The ability of the same server to support both TLS and cleartext operation seemed to conflict with the desire for a server to be able to disable cleartext operation for new users or users who had migrated to require TLS. It was found difficult to describe how servers requiring TLS for some users and permitting cleartext access for others, could do so without introducing the possibility for MUAs to expose the user's username and password in cleartext even when that user was required to use TLS - because with most of the password-based authentication mechanisms defined for these protocols, the server does not have the opportunity to refuse an authentication attempt until the user's password has been transmitted. Rather than recommend STARTTLS or allow either mechanism, it seemed simpler and less error-prone to just specify Implicit TLS as the required and recommended TLS negotiation mechanism for new MUA-to-server configurations.

## 2. Implementation Requirements

This section details requirements for implementations of electronic mail protocol clients and servers. Note that a requirement for a client or server implementation to support a particular feature is not the same thing as a requirement that a client or server running a conforming implementation be configured to use that feature. Requirements for MSPs are distinct from requirements for protocol implementations, and are listed in a separate section.

### 2.1. Mail Server Requirements

The following requirements apply to IMAP, POP, and Submission server implementations:

All IMAP, POP, and Submission servers **MUST** be configurable to support the use of TLS and the Implicit TLS mechanism when communicating with MUAs.

IMAP, POP, and Submission servers **SHOULD** also support the STARTTLS mechanism for the sake of backward compatibility with existing MUAs and configurations that use it.

Servers which support STARTTLS **SHOULD** be capable of requiring TLS before performing any operation other than capability discovery and STARTTLS.

IMAP, POP, and Submission servers which support STARTTLS SHOULD be capable of disabling STARTTLS operation and/or disabling operation on any port that isn't configured to use Implicit TLS, so that the service provider may force all users to use Implicit TLS.

## 2.2. Mail User Agent Requirements

This section describes requirements on Mail User Agents (MUAs) using IMAP, POP, and/or Submission protocols.

Note: Requirements pertaining to use of Submission servers are also applicable to use of SMTP servers (whether on port 25 or on another port as advertised by a SRV record with `_smtp._tcp` or `_smtps._tcp` label) for mail submission.

### 2.2.1. MUAs Configurable to Require TLS

MUAs which are configurable to communicate with user-specified IMAP, POP, and/or Submission servers MUST be configurable (on a per-server or per-account basis) to require the use of TLS when communicating with those servers.

MUAs MAY also be configurable (on a per-server or per-account basis) to use Opportunistic TLS when connecting to IMAP, POP, and Submission servers. Such a configuration MUST NOT be the default. Note that support for an Opportunistic TLS configuration option does not satisfy the requirement that MUAs be able to require use of TLS when communicating with a particular server.

In addition, MUAs MAY be configurable (on a per-server or per-account basis) to not use TLS, to permit it to interoperate with legacy servers that do not support TLS.

Whenever requested to establish any configuration that does not require TLS to talk to a server or account (including a configuration using Opportunistic TLS), an MUA SHOULD warn its user that his or her mail traffic (including password, if applicable) will be exposed to attackers.

### 2.2.2. Non-configurable MUAs and nonstandard access protocols

MUAs which are not configurable to use user-specified servers MUST use TLS or similarly other strong encryption mechanism when communicating with their mail servers. This generally applies to MUAs that are pre-configured to operate with one or more specific services, whether or not supplied by the vendor of those services.



MUAs using protocols other than IMAP, POP, and Submission to communicate with mail servers, MUST use TLS or other similarly robust encryption mechanism in conjunction with those protocols.

#### 2.2.3. Implicit TLS vs. STARTTLS

User-configurable MUAs MUST support the ability to use the Implicit TLS mechanism when communicating with servers that support it.

User-configurable MUAs SHOULD also support the STARTTLS mechanism for the sake of backward compatibility with IMAP, POP, and Submission servers that do not support Implicit TLS with these services.

#### 2.2.4. Use of SRV records in Establishing Configuration

User-configurable MUAs SHOULD support use of [RFC6186] to determine (for mail service providers that advertise such information) which options are available for configuration of connections to IMAP, POP, and Submission servers. However, when using configuration information obtained by this method, MUAs SHOULD behave as if the user had explicitly required TLS, unless the user has explicitly requested to disable it. (Compare with section 6 of [RFC6186]). This will have the effect of causing the MUA to ignore advertised configurations which do not support TLS, even when those advertised configurations have a higher priority than other advertised configurations. (The specific user interface by which a user requests to disable encryption is an implementation detail, but the user interface should make it clear to users that disabling encryption will likely result in their email being spied upon.) Note: [RFC6186] does not define a label for use with SRV records to indicate that a Submission server supports Implicit TLS on a particular port. This memo defines the `_submissions._tcp` label for that purpose.

When using [RFC6186] configuration information, Mail User Agents SHOULD NOT automatically establish new configurations which do not require TLS for all servers, unless there are no advertised configurations using TLS. If such a configuration is chosen, prior to attempting to authenticate to the server or use the server for message submission, the MUA SHOULD warn the user that traffic to that server will not be encrypted and that it will therefore likely be intercepted by unauthorized parties. (The specific wording is to be determined by the implementation, but it should adequately capture the sense of risk given the widespread incidence of mass surveillance of email traffic.)

When establishing a new configuration for connecting to an IMAP, POP, or Submission server, an MUA MUST NOT blindly trust SRV records

unless they are signed by DNSSEC and have a valid signature. Instead, the MUA SHOULD warn the user that the DNS-advertised mechanism for connecting to the server is not authenticated, and request the user to manually verify the connection details by reference to his or her mail service provider's documentation.

Similarly, an MUA MUST NOT consult SRV records to determine which servers to use on every connection attempt, unless those SRV records are signed by DNSSEC and have a valid signature. However, an MUA MAY consult SRV records from time to time to determine if an MSP's server configuration has changed, and alert the user if it appears that this has happened. This can also serve as a means to encourage users to upgrade their configurations to require TLS if and when their MSPs support it. However, MUAs SHOULD NOT automatically upgrade configurations to require TLS without explicit user approval.

#### 2.2.5. Manual configuration of MUA connection to servers

Configurable MUAs SHOULD permit manual user configuration and re-configuration of server name or address, port number, and whether to use STARTTLS and/or Implicit TLS, for IMAP, POP, and Submission servers, regardless of any information obtained using [RFC6186] procedures or other means.

Note: While many users will always use the IMAP or POP and Submission servers provided by the same MSP to which their incoming mail is delivered, there are many valid use cases for having these servers provided by multiple parties. It is therefore useful for an MUA to permit users to configure each of those services separately.

If a user explicitly selects a configuration for a server that does not use TLS, the MUA SHOULD, prior to authenticating to the server as that user, warn the user that traffic to the server will not be encrypted and thus will likely be intercepted by unauthorized parties. (The specific wording is to be determined by the implementation, but it should adequately capture the sense of risk given the widespread use of mass surveillance).

Whenever a MUA is explicitly configured to connect to a specific IP address rather than a DNS name, the MUA MUST also either be configured to explicitly compare the server certificate against a known certificate ("pinning"), or be explicitly configured as to which reference identifier(s) will be matched with the TLS server certificate's presented identifiers.

#### 2.2.6. Verification of new or edited server configurations

Any time the configuration of an MUA is altered to change the servers with which the MUA communicates, the MUA SHOULD verify that it can connect to the servers, validate the TLS certificates, compare them with TLSA records if those are present and have valid DNSSEC signatures, and authenticate to the servers on behalf of the user.

If TLSA verification of the server's public key fails the MUA should not attempt to authenticate to the server.

If the server's TLS certificate does not present any identifiers that match any of the appropriate reference identifiers for the server name, the MUA MAY offer to "pin" the server certificate for use in future comparisons. In such cases the MUA SHOULD instruct the user to check with the MSP to determine whether the MSP thinks that it has a valid certificate that is issued by a trusted certificate authority, before the user approves the configuration that "pins" the certificate.

#### 2.2.7. Downgrading of TLS-required Configurations

Once a configuration that requires TLS to connect to a server has been established, Mail User Agents MUST NOT attempt to authenticate to that server, or use that server for mail submission, without successfully negotiating TLS (including server certificate validity checks and reference identifier matching checks), unless the user has explicitly reconfigured the MUA to do so.

An MUAs configured to use STARTTLS for a particular server SHOULD warn its user when a server which previously advertised STARTTLS capability is apparently no longer doing so, but MUST NOT downgrade the connection to cleartext unless explicitly (re)configured by the user to do so.

#### 2.2.8. Requirements for MUA use of TLS

An MUA configured to require TLS when connecting to a particular server MUST successfully negotiate TLS (including successful certificate validity and reference identifier checks) before attempting to use that server. The TLS layer MAY use either Implicit TLS or STARTTLS, according to the client's configuration for that server.

An MUA that is configured to require TLS for a particular server **MUST** negotiate TLS (including successful certificate validity and reference identifier checks) before attempting to authenticate to that server. This TLS layer **MAY** be negotiated using either Implicit TLS or the STARTTLS mechanism, according to the client's configuration for that server. Note: This requirement applies even if the authentication mechanism doesn't use cleartext credentials.

MUAs **MUST** abort the connection and refuse to interact with any server for which TLS negotiation signals any of the alert messages specified in section 7.2 of [RFC5246], or any other indication that the connection may be insecure (whether due to man-in-the-middle attack or other reason). Exception: Connections to a server with a self-signed certificate **MAY** be accepted if the Mail User Agent is explicitly configured ("pinned") to accept a self-signed certificate for that server.

MUAs **MUST** use the procedure defined in [RFC6125] to determine whether a server's TLS certificate contains an identifier which matches the DNS name to which the MUA is attempting to connect, and **MUST** abort the TLS session if the server's certificate does not present an identifier that matches one of the MUA's predetermined reference identifiers for that server.

It is important to avoid using DNS names obtained from SRV records (rather than from explicit user configuration) as reference identifiers when comparing with presented identifiers in TLS server certificates, unless those SRV records were signed with DNSSEC and the signatures were verified by the MUA.

Note in Draft: [I-D.melnikov-email-tls-certs] describes a profile of [RFC6125] for use in MUA checking of presented identifiers in TLS server certificates.

#### 2.2.9. Use of SMTP by MUAs for other than mail submission

Some Mail User Agents use SMTP for purposes other than submitting mail, e.g. to determine whether a particular recipient can receive a message of a particular size. Such uses **SHOULD** use TLS if the server advertises STARTTLS in response to EHLO.

To avoid exposing message metadata which could be used for traffic analysis, MUAs **SHOULD NOT** send MAIL or RCPT to an SMTP server without negotiating TLS.

#### 2.2.10. Other network-accessible services used by MUAs

MUAs which are configured to access other services requiring authentication, and using the same reusable credentials (e.g. passwords) with those servers as are used to authenticate to servers using TLS, MUST NOT expose those credentials over an unencrypted connection.

#### 2.2.11. Additional Considerations for Webmail and other Split-MUA Clients

A webmail MUA is any MUA that is designed to be used via a web browser. Typically a webmail MUA has two portions - a "front-end" portion which runs in the user's web browser, and a "back-end" which runs on a web server. The webmail MUA typically uses HTTP to communicate between the front-end and back-end, and the back-end is responsible for communicating with message stores and mail submission services. Other "split MUA" arrangements also exist, notably to support mobile and other devices with modest local compute capability and/or bandwidth limitations.

The above requirements are also applicable to Webmail and other split MUA arrangements. For example, the requirements listed above for use of TLS between IMAP, POP, and Submission clients and servers also apply to communications between the back-ends of split MUAs and servers for those protocols. If the communications between the back-end of a split MUA and those servers doesn't use TLS, it MUST use a similarly-secure encryption mechanism.

In addition, split MUAs MUST use TLS or a similarly-secure encryption mechanism, to communicate between the front-end (web browser in the case of a webmail MUA) and the back-end.

#### 2.2.12. Use of DANE by MUAs

MUAs SHOULD be able to use the DANE TLSA records in DNS [RFC6698] to verify that the public key presented in a certificate ostensibly received from a server, is actually a key authorized for use by that domain name. Use of TLSA records can provide a trust anchor in addition to that provided by the TLS server certificate, and help protect against rogue certificate authorities and compromised certificate authority private keys. There are multiple cases which must be considered:

- o No TLSA record for the target domain exists. In this case verification of the server's certificate SHOULD rely entirely on whether the signing certificate authority is trusted by the client or whether the client has been explicitly configured ("pinned") to trust that particular certificate. However a MUA MAY be configurable to require both a signed TLSA record and a TLS server certificate signed by a trusted certificate authority.
- o One or more TLSA records exist for the target domain but are either unsigned, or the DNSSEC signature is invalid, or DNSSEC signature cannot be verified. In this case the client SHOULD refuse to connect to the server until the signature on the TLSA records can be verified, unless the client has been explicitly configured ("pinned") to trust a particular server certificate. This might either be an indication of an attack or a configuration error, but seems better to detect the configuration error and cause it to be fixed, than ignore it.
- o One or more TLSA records exist and have a valid DNSSEC signature but no TLSA records match the X.509 certificate presented by the server. In this cases the client MUST gracefully terminate the session with the server without attempting to authenticate or request services, as this may indicate a man-in-the-middle attack.
- o TLSA record exists and has a valid DNSSEC signature, and the public key specified in a TLSA record matches the public key in the X.509 certificate presented by the server. However, the server certificate is not signed by a trusted certificate authority, nor has the MUA been explicitly configured ("pinned") to accept that particular certificate. In this case the connection MUST gracefully terminate the session with the server without attempting to authenticate or request services.
- o The TLSA record has a valid DNSSEC signature, TLS has been successfully negotiated with no errors or alerts, and the server's certificate is valid and signed by a trusted certificate authority. In this case the session MAY proceed.

#### 2.2.13. Use of DNSSEC

All uses of DNSSEC by MUAs (including use of SRV and TLSA records) SHOULD explicitly verify the chain of DNSSEC signatures from the root, rather than trusting a recursive caching DNS name server to do so. It is acceptable to obtain RRSIG, DNSKEY, DS, etc., resource records from a recursive caching name server. But a recursive caching name server SHOULD NOT be assumed to be trustworthy enough to validate signatures.

### 2.3. Requirements Common To Both Servers and MUAs

TLS version 1.2 [RFC5246] SHOULD be supported.

Per [RFC6176], SSL version 2.0 MUST NOT be supported. MUAs MUST either disable SSL 2.0 support in their TLS implementations or immediately close a connection with a server if SSL 2.0 is negotiated. Servers MUST NOT advertise support for version 2.0 of SSL.

The renegotiation indication extension described in [RFC5746] SHOULD be supported.

The Server Name Indication extension [RFC6066] SHOULD be supported.

## 3. Mail Service Provider Requirements

### 3.1. Server Requirements

Mail Service Providers MUST use server implementations that conform to this specification.

### 3.2. MSPs MUST provide Submission Servers

Mail Service Providers which accept incoming mail for delivery using the Internet Protocol MUST provide one or more Submission servers for this purpose, separate from the SMTP servers used to process incoming mail. Those submission servers MUST be configured to support Implicit TLS and MAY be configured to support STARTTLS also.

MSPs MAY also support submission of messages via one or more designated SMTP servers to facilitate compatibility with existing MUA configurations and legacy MUAs.

Discussion: SMTP servers used to accept incoming mail or to relay mail are expected to accept mail in cleartext. This is incompatible with the purpose of this memo which is to encourage encryption of traffic between mail servers. There is no such requirement for Submission servers to accept mail in cleartext or without authentication. For other reasons, use of separate Submission servers has been best practice for many years.

Submission servers SHOULD require authentication as a condition of accepting mail.

### 3.3. TLS Server Certificate Requirements

MSPs MUST maintain valid server certificates for all servers. Those server certificates MUST present DNS-IDs and SRV-IDs conforming to [RFC6125] and which will be recognized by MUAs meeting the requirements of this memo. In addition, those server certificates MAY provide other DNS-IDs, SRV-IDs, or CN-IDs needed for compatibility with legacy MUAs.

A single certificate MAY be used for multiple electronic mail protocol servers (including webmail) which all providing service for a particular mail domain, but use of the same certificate for services other than electronic mail is discouraged.

If a protocol server provides service for more than one mail domain, its server certificates MAY advertise multiple domains. This will generally be necessary unless and until it is acceptable to impose the constraint that the server and all clients support the Server Name Indication extension to TLS.

#### 3.4. Recommended DNS records for mail protocol servers

This section discusses not only the DNS records that are recommended, but also implications of DNS records for server configuration and TLS server certificates.

##### 3.4.1. MX records

It is recommended that MSPs advertise MX records for handling of inbound mail (instead of relying entirely on A or AAAA records), and that those MX records be signed using DNSSEC. This is mentioned here only for completeness, as handling of inbound mail is out of scope for this document.

##### 3.4.2. SRV records

MSPs SHOULD advertise SRV records to aid MUAs in determination of proper configuration of servers, per the instructions in [RFC6186].

MSPs SHOULD advertise servers that support Implicit TLS in preference to those which support cleartext and/or STARTTLS operation.

##### 3.4.3. TLSA records

MSPs SHOULD advertise TLSA records to provide an additional trust anchor for public keys used in TLS server certificates. However, TLSA records MUST NOT be advertised unless they are signed using DNSSEC.



#### 3.4.4. DNSSEC

All DNS records advertised by an MSP as a means of aiding clients in communicating with the MSP's servers, SHOULD be signed using DNSSEC.

#### 3.5. MSP Server Monitoring

MSPs SHOULD regularly and frequently monitor their various servers to make sure that: TLS server certificates remain valid and are not about to expire, TLSA records match the public keys advertised in server certificates and are signed using DNSSEC, server configurations are consistent with SRV advertisements, and DNSSEC signatures are valid and verifiable. Failure to detect expired certificates and DNS configuration errors in a timely fashion can result in significant loss of service for an MSP's users.

#### 3.6. Encourage Transition to TLS Required Configurations

Mail Service Providers SHOULD encourage their users to transition to requiring TLS for communications with their servers.

Each MSP must determine which transition measures are most appropriate for its own user community. Possible mechanisms include, but are not limited to: using [RFC6186] to advertise servers which implement Implicit TLS; allowing individual users to configure their accounts so that the servers will refuse their authentication unless using TLS; requiring new users to always use TLS; providing or recommending MUA implementations that implement TLS and the ability to require TLS.

Note: there is a tradeoff here between encouraging use of TLS and not breaking access for existing users or users with legacy mail clients. Whether to enable "TLS required" for all users, new users only, or particular users that have expressed a preference to always use TLS, is a policy decision which should be re-evaluated periodically as conditions change - e.g. as more clients are upgraded to support TLS and [RFC6186]. Similarly, whether and when to require existing users to use TLS (and perhaps to upgrade their mail clients) is a policy decision that will differ from one service provider to the next depending on conditions and business needs.

#### 4. Security Considerations

This entire memo is about security considerations.

The mechanisms in this memo are intended to address certain specific identified threats, including:

- o A downgrading attack by thwarting connection to or TLS negotiation on the "SSL port", by a MUA implementing Opportunistic TLS. This is addressed by encouraging MUAs to implement "TLS required" operation so that the MUA will not downgrade.
- o Compromised certificate authority private keys, and rogue certificate authority issuing certificates to impersonators, to generate fake certificates that can be used with man-in-the-middle attacks. This is addressed by encouraging support for DNSSEC-signed TLSA records in both clients and servers, thus providing an additional trust anchor beyond the TLS server certificate.
- o An interception proxy, firewall, or other middlebox hiding STARTTLS capability advertisement or blocking the STARTTLS command, thus forcing a downgrade. This is addressed by encouraging MUAs to support "TLS required" configurations and users to migrate to them, as well as by encouraging Implicit TLS in preference to STARTTLS.
- o Attacks on DNS queries, including cache poisoning, man-in-the-middle, and forged responses. These are addressed by encouraging use of DNSSEC and by insisting on strict verification of presented identifiers obtained from TLS server certificates against a predetermined set of reference identifiers that are based either on explicit user input or DNSSEC-signed DNS responses.

In exchange for the perceived benefits listed above, the mechanisms described in this memo may increase the vulnerability of mail services to denial-of-service attacks. This appears to be a necessary and appropriate compromise.

Use of TLS is not a substitute for end-to-end encryption such as S/MIME. In particular, TLS does not and cannot protect against compromise of the message servers that see the messages in cleartext. Users are encouraged to use end-to-end encryption whenever available.

## 5. IANA Considerations

IANA is requested to allocate a well-known port for use with a Submission protocol server configured to use Implicit TLS. The recommended service identifier for this port is "submissions", for consistency with identifiers for other "SSL ports", even though this looks like a plural.

If there is a registry of labels for SRV records, IANA is requested to define a label of `_submissions._tcp` for use in advertising Submission servers using Implicit TLS.

## 6. References

### 6.1. Normative References

- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.
- [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.
- [RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", RFC 5746, February 2010.
- [RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.
- [RFC6176] Turner, S. and T. Polk, "Prohibiting Secure Sockets Layer (SSL) Version 2.0", RFC 6176, March 2011.
- [RFC6186] Daboo, C., "Use of SRV Records for Locating Email Submission/Access Services", RFC 6186, March 2011.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, November 2011.

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

## 6.2. Informative References

- [RFC3369] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3369, August 2002.
- [I-D.melnikov-email-tls-certs]  
Melnikov, A., "Updated TLS Server Identity Check Procedure for Email Related Protocols", draft-melnikov-email-tls-certs-01 (work in progress), October 2013.
- [I-D.ietf-dane-smtp-with-dane]  
Dukhovni, V. and W. Hardaker, "SMTP security via opportunistic DANE TLS", draft-ietf-dane-smtp-with-dane-00 (work in progress), October 2013.

## Author's Address

Keith Moore  
Network Heretics  
PO Box 1934  
Knoxville, TN 37901  
United States

EMail: moore@network-heretics.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 20, 2014

D. Thaler  
Microsoft  
October 17, 2013

Guidelines and Registration Procedures for New URI Schemes: Problem  
Statement  
draft-thaler-uri-scheme-reg-ps-01.txt

Abstract

This document describes some problems with the existing guidelines and procedures, as documented in RFC 4395, for new URI schemes.

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 April 20, 2014.

Copyright Notice

Copyright (c) 2013 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. Introduction . . . . .	2
2. Problems . . . . .	4
2.1. Current registration process doesn't scale well . . . . .	4
2.2. Lack of incentive to register . . . . .	5
2.3. Current private scheme guidance causes conflicts . . . . .	5
3. Security Considerations . . . . .	6
4. IANA Considerations . . . . .	6
5. Informative References . . . . .	6
Author's Address . . . . .	7

## 1. Introduction

RFC 4395 [RFC4395] provides guidelines and recommendations for the definition of Uniform Resource Identifier (URI) schemes. It defines procedures and guidelines for four types of URI schemes:

- a. Permanent, which [RFC4395] requires for all IETF Standards-Track schemes, and which has strict requirements.
- b. Provisional, which has a lower barrier.
- c. Historical, which is for schemes no longer in use and hence generally does not apply to "new" URI schemes.
- d. Private, meaning not registered with IANA.

As explained in Section 1 of [RFC4395], the purpose of an IANA-maintained registry is to:

1. provide a central point of discovery for established URI scheme names, and easy location of their defining documents;
2. discourage use of the same URI scheme name for different purposes;
3. help those proposing new URI scheme names to discern established trends and conventions, and avoid names that might be confused with existing ones;
4. encourage registration by setting a low barrier for provisional registrations.

However, the guidance in [RFC4395] is, in many cases that are now common, ambiguous or insufficient to accomplish the stated purposes. This document discusses a number of such problems. In doing so, we note that an effort was started to update the guidance, in

[I-D.ietf-iri-4395bis-irireg]. It does not, however, address the problems we discuss in this document, although it may be the logical place to do so.

It is first important to understand the scale of the problem. It is already common on many widely deployed platforms (including Windows, iOS, and Android) and form factors (PCs, phones, etc.) today to allow applications to be associated with specific URI schemes, such that when the URI is accessed (e.g., clicking on a link in a browser, or calling an equivalent API from an application), the associated application is launched to handle the URI. That is, the application is given the URI and determines what action to take (as opposed to being given content that the URI points to). Indeed, some such URIs are simply Uniform Resource Names that contain the data themselves, rather than Uniform Resource Locators that can be resolved to content. As such, URIs are increasingly becoming a form of inter-process communication as a way to invoke another application, with arguments placed in the scheme-specific part of the URI. Thus, in the extreme case, every application might define its own URI scheme, and the number of applications available on mainstream platforms today is easily numbered in the hundreds of thousands.

This use of URIs can be viewed as different from the web. That is, an increasingly larger portion of URI schemes are intended for "local" use, rather than for use with the web. The "URI Generic Syntax" [RFC3986] explicitly allows for such a wide scope of use of URIs. It states, in section 1.1:

This specification does not limit the scope of what might be a resource; rather, the term "resource" is used in a general sense for whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose (e.g., "today's weather report for Los Angeles"), a service (e.g., an HTTP-to-SMS gateway), and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity).

and

This specification does not place any limits on the nature of a resource, the reasons why an application might seek to refer to a resource, or the kinds of systems that might use URIs for the sake of identifying resources.

The current process was designed based in part on joint recommendations from the W3C and IETF in 2002 [RFC3305], when the known uses of schemes were such that there were 34 registered schemes, 51 known publically documented but unregistered schemes, and 50 or so private schemes with 2-3 being added every day, as noted (see Section 3.1 of [RFC3305]). Such private growth has continued and expanded to more platforms since then, such that the public schemes are now probably a small minority.

## 2. Problems

### 2.1. Current registration process doesn't scale well

Section 5.2 of [RFC4395] requires a four-week mailing list review for all Permanent registrations. It is, however, ambiguous as to whether a mailing list review is required for Provisional registrations and if so, for how long. The longer the process, the less of an incentive there is to register Provisional schemes. This problem was discussed in 2010 by the IRI WG, which concluded that a mailing list review should not be required for Provisional schemes, only expert review which may take up to two weeks, but this conclusion has not yet been documented.

The manual step of expert review still introduces a scalability bottleneck. What if all new applications being submitted to an app store started sending requests for Provisional URI schemes? The expert review process would be overwhelmed, especially if no one is paid to do the expert review. As such, the goals stated in Section 1 become far less effective when registered schemes are only a tiny fraction of the URI schemes in use in practice.

The author ran an experiment in 2012, which was reported to the IRI WG at its final meeting at IETF 85, where over 75 schemes that were listed on Wikipedia as being unregistered but in use were submitted as third-party registrations. All of them were registered after two weeks had passed and it was pointed out that the deadline had expired and per the process in [RFC4395], must be automatically listed. The only noticeable outcome of the expert review, other than to introduce a two week delay and manual effort, was to add a warning about the unknown security impact of one scheme. This is not intended to imply that the expert review was not valuable, only that the value provided could not scale effectively if the process were stressed with the current potential demand.

In summary, [RFC4395] defines a set of goals, which we listed above in Section 1. The current mechanism does not meet those goals. To meet the stated goals would require the majority of schemes to be registered. The current process cannot scale to do so, given current



practice. Hence, we either need to change the goals, or change the process, or both.

## 2.2. Lack of incentive to register

Currently there is little incentive for an organization outside the IETF to register schemes (whether as Permanent, Provisional, or Historical). Registering introduces a cost, both in terms of manual effort needed to apply, but also in the time delay introduced. This cost must be weighed against the benefit, which is primarily to simply lower the risk of collision. (Another benefit is to provide ease of access to relevant documentation via the IANA registry, although this benefit is often seen as unimportant or even undesirable in some cases.)

As long as the risk of collision is perceived to be low, or the effect of collision considered to be acceptable (e.g., asking the user which app to launch), registration is bypassed in favor of a "Private" scheme. The effect of collision can of course be problematic (though the scheme-defining organization may not realize the danger) when the syntax of the scheme-specific part differs. Launching an application with a URI that is invalid according to that application's syntax for the custom URI scheme is not useful.

An app store certification process could in theory require or encourage Provisional application. However, there is little incentive for them to do so either, since an app store itself has a process which would be delayed and disincent application developers to submit applications.

## 2.3. Current private scheme guidance causes conflicts

Section 2.8 of [RFC4395] states:

Organizations that desire a private name space for URI scheme names are encouraged to use a prefix based on their domain name, expressed in reverse order. For example, a URI scheme name of com-example-info might be registered by the vendor that owns the example.com domain name.

There are multiple problems with the above guidance:

1. No guidance is given for when it might or might not be appropriate to use a private name space. For example, is this guidance appropriate for application vendors defining a custom scheme that they want to associate the application with? As such, the current assumption is that it is appropriate for anyone who can live with some potential risk of collision.

2. Hyphens occur in actual domain names. Consider one organization that owns the domain name "foo.bar.example", and another organization that owns "foo-bar.example". Using the mechanism implied in the example can result in both colliding with "example-bar-foo-info".
3. The guidance is only an encouragement, and no precise algorithm is given. For example, whether "." should be converted to "-" as in the example is unclear. If an organization is actually trying to follow the recommended guidelines, they will likely use a "-" as directed and risk conflicts as noted above. More commonly, an organization today will simply use a string that identifies (say) their application, and not be based on a domain name.
4. No protection is suggested against IANA later granting registration to a scheme that follows the recommended convention that is in use by someone else. For example, as can be seen at [IANAURI], there are already registered schemes that use "." (e.g., "iris.beep") and "-" (e.g., "xcon-userid") in them, and there could be similar new schemes registered at any time. If an organization had previously acquired the TLD "iris" or "xcon", those values could already be in use in applications from those organizations. Especially now that ICANN is allowing gTLD applications, this is a very real possibility.

### 3. Security Considerations

The security considerations in [RFC4395] still apply.

### 4. IANA Considerations

This document requires no actions by the IANA.

### 5. Informative References

- [I-D.ietf-iri-4395bis-irireg]  
Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI/IRI Schemes", draft-ietf-iri-4395bis-irireg-04 (work in progress), December 2011.
- [IANAURI] IANA, ., "Uniform Resource Identifier (URI) Schemes", 2013, <<http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml>>.
- [RFC3305] Mealling, M. and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names

(URNs): Clarifications and Recommendations", RFC 3305, August 2002.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 35, RFC 4395, February 2006.

Author's Address

Dave Thaler  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052  
USA

Phone: +1 425 703 8835  
Email: dthaler@microsoft.com