Individual submission
Internet-Draft
Obsoletes: 7001, 7410
(if approved)
Intended status: Standards Track
Expires: September 27, 2015

M. Kucherawy
March 26, 2015

Message Header Field for Indicating Message Authentication Status
draft-ietf-appsawg-rfc7001bis-05

Abstract

This document specifies a message header field called Authentication-Results for use with electronic mail messages to indicate the results of message authentication efforts. Any receiver-side software, such as mail filters or Mail User Agents (MUAs), can use this header field to relay that information in a convenient and meaningful way to users or to make sorting and filtering decisions.

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 September 27, 2015.

Table of Contents

1. Introduction

This document describes a header field called Authentication-Results for electronic mail messages that presents the results of a message authentication effort in a machine-readable format. The intent of the header field is to create a place to collect such data when message authentication mechanisms are in use so that a Mail User Agent (MUA) and downstream filters can make filtering decisions and/or provide a recommendation to the user as to the validity of the message's origin and possibly the safety and integrity of its content.

This document revises the original definition found in [RFC5451] based upon various authentication protocols in current use and incorporates errata logged since the publication of the original specification.

End users are not expected to be direct consumers of this header field. This header field is intended for consumption by programs that will then use such data or render it in a human-usable form.

This document specifies the format of this header field and discusses the implications of its presence or absence. However, it does not discuss how the data contained in the header field ought to be used, such as what filtering decisions are appropriate or how an MUA might render those results, as these are local policy and/or user interface design questions that are not appropriate for this document.

At the time of publication of this document, the following are published, domain-level email authentication methods in common use:

- Author Domain Signing Practices ([ADSP])
- SMTP Service Extension for Authentication ([AUTH])
- DomainKeys Identified Mail Signatures ([DKIM])
- Sender Policy Framework ([SPF])
- Vouch By Reference ([VBR])
- reverse IP address name validation ("iprev", defined in Section 3)

In addition, the following are non-standard methods recognized by this specification that are no longer common:

- DomainKeys ([DOMAINKEYS]) (Historic)

o Sender ID ([SENDERID]) (Experimental)

This specification is not intended to be restricted to domain-based authentication schemes, but the existing schemes in that family have proven to be a good starting point for implementations. The goal is to give current and future authentication schemes a common framework within which to deliver their results to downstream agents and discourage the creation of unique header fields for each.

Although SPF defined a header field called "Received-SPF" and the historic DomainKeys defined one called "DomainKey-Status" for this purpose, those header fields are specific to the conveyance of their respective results only and thus are insufficient to satisfy the requirements enumerated below. In addition, many SPF implementations have adopted the header field specified here at least as an option, and DomainKeys has been obsoleted by DKIM.

1.1. Purpose

The header field defined in this document is expected to serve several purposes:

1. Convey the results of various message authentication checks, which are applied by upstream filters and Mail Transfer Agents (MTAs) and then passed to MUAs and downstream filters within the same "trust domain". Such agents might wish to render those results to end users or to use those data to apply more or less stringent content checks based on authentication results;

2. Provide a common location within a message for this data;

3. Create an extensible framework for reporting new authentication methods as they emerge.

In particular, the mere presence of this header field does not mean its contents are valid. Rather, the header field is reporting assertions made by one or more authentication schemes (supposedly) applied somewhere upstream. For an MUA or downstream filter to treat the assertions as actually valid, there must be an assessment of the trust relationship among such agents, the validating MTA, and the mechanism for conveying the information.

1.2. Trust Boundary

This document makes several references to the "trust boundary" of an administrative management domain (ADMD). Given the diversity among existing mail environments, a precise definition of this term isn't possible.

Simply put, a transfer from the producer of the header field to the consumer must occur within a context that permits the consumer to treat assertions by the producer as being reliable and accurate (trustworthy). How this trust is obtained is outside the scope of this document. It is entirely a local matter.

Thus, this document defines a "trust boundary" as the delineation between "external" and "internal" entities. Services that are internal -- within the trust boundary -- are provided by the ADMD's infrastructure for its users. Those that are external are outside of the authority of the ADMD. By this definition, hosts that are within a trust boundary are subject to the ADMD's authority and policies, independent of their physical placement or their physical operation. For example, a host within a trust boundary might actually be operated by a remote service provider and reside physically within its data center.

It is possible for a message to be evaluated inside a trust boundary but then depart and re-enter the trust boundary. An example might be a forwarded message such as a message/rfc822 attachment (see Multipurpose Internet Mail Extensions [MIME]) or one that is part of a multipart/digest. The details reported by this field cannot be trusted in that case. Thus, this field found within one of those media types is typically ignored.

1.3. Processing Scope

The content of this header field is meant to convey to message consumers that authentication work on the message was already done within its trust boundary, and those results are being presented. It is not intended to provide message parameters to consumers so that they can perform authentication protocols on their own.

1.4. Requirements

This document establishes no new requirements on existing protocols or servers.

In particular, this document establishes no requirement on MTAs to reject or filter arriving messages that do not pass authentication checks. The data conveyed by the specified header field's contents are for the information of MUAs and filters and are to be used at their discretion.

1.5. Definitions

This section defines various terms used throughout this document.

1.5.1. Key Words

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].

1.5.2. Security

"Guidelines for Writing RFC Text on Security Considerations" ([SECURITY]) discusses authentication and authorization and the conflation of the two concepts. The use of those terms within the context of recent message security work has given rise to slightly different definitions, and this document reflects those current usages, as follows:

- "Authorization" is the establishment of permission to use a resource or represent an identity. In this context, authorization indicates that a message from a particular ADMD arrived via a route the ADMD has explicitly approved.

- "Authentication" is the assertion of validity of a piece of data about a message (such as the sender's identity) or the message in its entirety.

As examples: SPF and Sender ID are authorization mechanisms in that they express a result that shows whether or not the ADMD that apparently sent the message has explicitly authorized the connecting Simple Mail Transfer Protocol ([SMTP]) client to relay messages on its behalf, but they do not actually validate any other property of the message itself. By contrast, DKIM is agnostic as to the routing of a message but uses cryptographic signatures to authenticate agents, assign (some) responsibility for the message (which implies authorization), and ensure that the listed portions of the message were not modified in transit. Since the signatures are not tied to SMTP connections, they can be added by either the ADMD of origin, intermediate ADMDs (such as a mailing list server), other handling agents, or any combination.

Rather than create a separate header field for each class of solution, this proposal groups them both into a single header field.

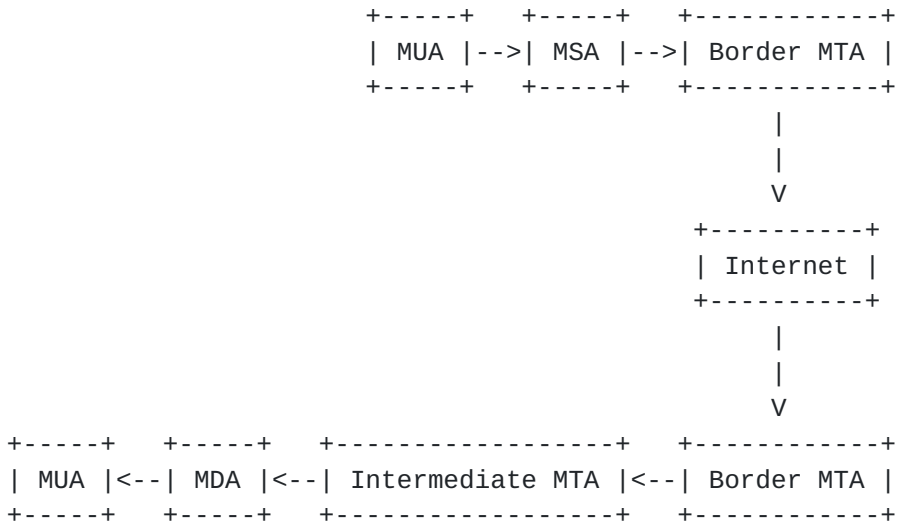
1.5.3. Email Architecture

- A "border MTA" is an MTA that acts as a gateway between the general Internet and the users within an organizational boundary. (See also Section 1.2.)

- A "delivery MTA" (or Mail Delivery Agent or MDA) is an MTA that actually enacts delivery of a message to a user's inbox or other final delivery.

- An "intermediate MTA" is any MTA that is not a delivery MTA and is also not the first MTA to handle the message.

The following diagram illustrates the flow of mail among these defined components. See Internet Mail Architecture [EMAIL-ARCH] for further discussion on general email system architecture, which includes detailed descriptions of these components, and Appendix D of this document for discussion about the common aspects of email authentication in current environments.



Generally, it is assumed that the work of applying message authentication schemes takes place at a border MTA or a delivery MTA. This specification is written with that assumption in mind. However, there are some sites at which the entire mail infrastructure consists of a single host. In such cases, such terms as "border MTA" and "delivery MTA" might well apply to the same machine or even the very same agent. It is also possible that some message authentication tests could take place on an intermediate MTA. Although this document doesn't specifically describe such cases, they are not meant to be excluded.

1.5.4. Other Terms

In this document, the term "producer" refers to any component that adds this header field to messages it is handling, and "consumer" refers to any component that identifies, extracts, and parses the header field to use as part of a handling decision.

1.6. Trust Environment

This header field permits one or more message validation mechanisms to communicate output to one or more separate assessment mechanisms. These mechanisms operate within a unified trust boundary that defines an Administrative Management Domain (ADMD). An ADMD contains one or more entities that perform validation and generate the header field and one or more that consume it for some type of assessment. The field often contains no integrity or validation mechanism of its own, so its presence must be trusted implicitly. Hence, valid use of the header field requires removing any occurrences of it that are present when the message enters the ADMD. This ensures that later occurrences have been added within the trust boundary of the ADMD.

The authserv-id token defined in Section 2.2 can be used to reference an entire ADMD or a specific validation engine within an ADMD. Although the labeling scheme is left as an operational choice, some guidance for selecting a token is provided in later sections of this document.

2. Definition and Format of the Header Field

This section gives a general overview of the format of the header field being defined and then provides more formal specification.

2.1. General Description

The header field specified here is called Authentication-Results. It is a Structured Header Field as defined in Internet Message Format ([MAIL]), and thus all of the related definitions in that document apply.

This header field is added at the top of the message as it transits MTAs that do authentication checks, so some idea of how far away the checks were done can be inferred. It is therefore considered to be a trace field as defined in [MAIL], and thus all of the related definitions in that document apply.

The value of the header field (after removing comments) consists of an authentication identifier, an optional version, and then a series of statements and supporting data. The statements are of the form "method=result" and indicate which authentication method(s) were applied and their respective results. For each such statement, the supporting data can include a "reason" string and one or more "property=value" statements indicating which message properties were evaluated to reach that conclusion.

The header field can appear more than once in a single message, more

than one result can be represented in a single header field, or a combination of these can be applied.

2.2. Formal Definition

Formally, the header field is specified as follows using Augmented Backus-Naur Form ([ABNF]):

```
authres-header = "Authentication-Results:" [CFWS] authserv-id
         [ CFWS authres-version ]
         ( no-result / 1*resinfo ) [CFWS] CRLF

authserv-id = value
            ; see below for a description of this element

authres-version = 1*DIGIT [CFWS]
        ; indicates which version of this specification is in use;
        ; this specification is version "1", and the absence of a
        ; version implies this version of the specification

no-result = [CFWS] ";" [CFWS] "none"
          ; the special case of "none" is used to indicate that no
          ; message authentication was performed

resinfo = [CFWS] ";" methodspec [ CFWS reasonspec ]
          *( CFWS propspec )

methodspec = [CFWS] method [CFWS] "=" [CFWS] result
           ; indicates which authentication method was evaluated
           ; and what its output was

reasonspec = "reason" [CFWS] "=" [CFWS] value
           ; a free-form comment on the reason the given result
           ; was returned

propspec = ptype [CFWS] "." [CFWS] property [CFWS] "=" pvalue
         ; an indication of which properties of the message
         ; were evaluated by the authentication scheme being
         ; applied to yield the reported result

method = Keyword [ [CFWS] "/" [CFWS] method-version ]
       ; a method indicates which method's result is
       ; represented by "result", and is one of the methods
       ; explicitly defined as valid in this document
       ; or is an extension method as defined below

method-version = 1*DIGIT [CFWS]
       ; indicates which version of the method specification is
```

```
            ; in use, corresponding to the matching entry in the IANA
            ; "Email Authentication Methods" registry; a value of "1"
            ; is assumed if this version string is absent

     result = Keyword
            ; indicates the results of the attempt to authenticate
            ; the message; see below for details

     ptype = Keyword
           ; indicates whether the property being evaluated was
           ; a parameter to an [SMTP] command, was a value taken
           ; from a message header field, was some property of
           ; the message body, or was some other property evaluated by
           ; the receiving MTA; expected to be one of the "property
           ; types" explicitly defined as valid, or an extension
           ; ptype, as defined below

     property = special-smtp-verb / Keyword
             ; indicates more specifically than "ptype" what the
             ; source of the evaluated property is; the exact meaning
             ; is specific to the method whose result is being reported
             ; and is defined more clearly below

     special-smtp-verb = "mailfrom" / "rcptto"
             ; special cases of [SMTP] commands that are made up
             ; of multiple words

     pvalue = [CFWS] ( value / [ [ local-part ] "@" ] domain-name )
              [CFWS]
            ; the value extracted from the message property defined
            ; by the "ptype.property" construction
```

"local-part" is defined in Section 3.4.1 of [MAIL], and "CFWS" is defined in Section 3.2.2 of [MAIL].

"Keyword" is defined in Section 4.1.2 of [SMTP].

The "value" is as defined in Section 5.1 of [MIME].

The "domain-name" is as defined in Section 3.5 of [DKIM].

The "Keyword" used in "result" above is further constrained by the necessity of being enumerated in Section 2.7.

See Section 2.5 for a description of the authserv-id element.

If the value portion of a "pvalue" construction identifies something intended to be an e-mail identity, then it MUST use the right hand

portion of that ABNF definition.

The list of commands eligible for use with the "smtp" ptype can be found in Section 4.1 of [SMTP].

The "propspec" may be omitted if, for example, the method was unable to extract any properties to do its evaluation yet has a result to report.

Where an SMTP command name is being reported as a "property", the agent generating the header field represents that command by converting it to lowercase and dropping any spaces (e.g., "MAIL FROM" becomes "mailfrom", "RCPT TO" becomes "rcptto", etc.).

A "ptype" value of "policy" indicates a policy decision about the message not specific to a property of the message that could be extracted. See Section 2.4 for details.

Examples of complete messages using this header field can be found in Appendix C.

2.3. Property Types (ptypes) and Properties

The "ptype" in the ABNF above indicates the general type of property being described by the result being reported, upon which the reported result was based. Coupled with the "property", which is more specific, they indicate from which particular part of the message the reported data were extracted.

Combinations of ptypes and properties are registered and described in the "Email Authentication Methods" registry, coupled with the authentication methods with which they are used. This is further described in Section 6.

Legal values of "ptype" are as defined in the IANA "Email Authentication Property Types" registry, created by [PTYPES-REGISTRY]. The initial values and what they typically indicate are as follows, copied from [RFC7001]:

body: Information that was extracted from the body of the message. This might be an arbitrary string of bytes, a hash of a string of bytes, a Uniform Resource Identifier, or some other content of interest. The "property" is an indication of where within the message body the extracted content was found, and can indicate an offset, identify a MIME part, or

header: Indicates information that was extracted from the header of the message. This might be the value of a header field or some portion of a header field. The "property" gives a more precise indication of the place in the header from which the extraction took place.

policy: A local policy mechanism was applied that augments or overrides the result returned by the authentication mechanism. (See Section 2.4.)

smtp: Indicates information that was extracted from an SMTP command that was used to relay the message. The "property" indicates which SMTP command included the extracted content as a parameter.

Results reported using unknown ptypes MUST NOT be used in making handling decisions. They can be safely ignored by consumers.

Entries in the "Email Authentication Methods" registry can define properties that deviate from these definitions when appropriate. Such deviations need to be clear in the registry and/or in the defining document. See Section 2.7.1 for an example.

2.4. The "policy" ptype

A special ptype value of "policy" is also defined. This ptype is provided to indicate that some local policy mechanism was applied that augments or even replaces (i.e., overrides) the result returned by the authentication mechanism. The property and value in this case identify the local policy that was applied and the result it returned.

For example, a DKIM signature is not required to include the Subject header field in the set of fields that are signed. An ADMD receiving such a message might decide that such a signature is unacceptable, even if it passes, because the content of the Subject header field could be altered post-signing without invalidating the signature. Such an ADMD could replace the DKIM "pass" result with a "policy" result and then also include the following in the corresponding Authentication-Result field:

```
... dkim=fail policy.dkim-rules=unsigned-subject ...
```

In this case, the property is "dkim-rules", indicating some local check by that name took place and that check returned a result of "unsigned-subject". These are arbitrary names selected by (and presumably used within) the ADMD making use of them, so they are not normally registered with IANA or otherwise specified apart from setting syntax restrictions that allow for easy parsing within the

rest of the header field.

This ptype existed in the original specification for this header field, but without a complete description or example of intended use. As a result, it has not seen any practical use to date that matches its intended purpose. These added details are provided to guide implementers toward proper use.

2.5. Authentication Identifier Field

Every Authentication-Results header field has an authentication service identifier field (authserv-id above). Specifically, this is any string intended to identify the authentication service within the ADMD that conducted authentication checks on the message. This identifier is intended to be machine-readable and not necessarily meaningful to users.

Since agents consuming this field will use this identifier to determine whether its contents are of interest (and are safe to use), the uniqueness of the identifier MUST be guaranteed by the ADMD that generates it and MUST pertain to that ADMD. MUAs or downstream filters SHOULD use this identifier to determine whether or not the data contained in an Authentication-Results header field ought to be used or ignored.

For simplicity and scalability, the authentication service identifier SHOULD be a common token used throughout the ADMD. Common practice is to use the DNS domain name used by or within that ADMD, sometimes called the "organizational domain", but this is not strictly necessary.

For tracing and debugging purposes, the authentication identifier can instead be the specific hostname of the MTA performing the authentication check whose result is being reported. Moreover, some implementations define a substructure to the identifier; these are outside of the scope of this specification.

Note, however, that using a local, relative identifier like a flat hostname, rather than a hierarchical and globally unique ADMD identifier like a DNS domain name, makes configuration more difficult for large sites. The hierarchical identifier permits aggregating related, trusted systems together under a single, parent identifier, which in turn permits assessing the trust relationship with a single reference. The alternative is a flat namespace requiring individually listing each trusted system. Since consumers will use the identifier to determine whether to use the contents of the header field:

- Changes to the identifier impose a large, centralized administrative burden.

- Ongoing administrative changes require constantly updating this centralized table, making it difficult to ensure that an MUA or downstream filter will have access to accurate information for assessing the usability of the header field's content. In particular, consumers of the header field will need to know not only the current identifier(s) in use but previous ones as well to account for delivery latency or later re-assessment of the header field's contents.

Examples of valid authentication identifiers are "example.com", "mail.example.org", "ms1.newyork.example.com", and "example-auth".

2.6. Version Tokens

The grammar above provides for the optional inclusion of versions on both the header field itself (attached to the authserv-id token) and on each of the methods being reported. The method version refers to the method itself, which is specified in the documents describing those methods, while the authserv-id version refers to this document and thus the syntax of this header field.

The purpose of including these is to avoid misinterpretation of the results. That is, if a parser finds a version after an authserv-id that it does not explicitly know, it can immediately discontinue trying to parse since what follows might not be in an expected format. For a method version, the parser SHOULD ignore a method result if the version is not supported in case the semantics of the result have a different meaning than what is expected. For example, if a hypothetical DKIM version 2 yielded a "pass" result for different reasons than version 1 does, a consumer of this field might not want to use the altered semantics. Allowing versions in the syntax is a way to indicate this and let the consumer of the header field decide.

2.7. Defined Methods and Result Values

Each individual authentication method returns one of a set of specific result values. The subsections below provide references to the documents defining the authentication methods specifically supported by this document, and their corresponding result values. Verifiers SHOULD use these values as described below. New methods not specified in this document, but intended to be supported by the header field defined here, MUST include a similar result table either in their defining documents or in supplementary ones.

2.7.1. DKIM and DomainKeys

DKIM is represented by the "dkim" method and is defined in [DKIM]. DomainKeys is defined in [DOMAINKEYS] and is represented by the "domainkeys" method.

Section 3.8 of [DOMAINKEYS] enumerates some possible results of a DomainKeys evaluation. Those results are not used when generating this header field; rather, the results returned are listed below.

A signature is "acceptable to the ADMD" if it passes local policy checks (or there are no specific local policy checks). For example, an ADMD policy might require that the signature(s) on the message be added using the DNS domain present in the From header field of the message, thus making third-party signatures unacceptable even if they verify.

Both DKIM and DomainKeys use the same result set, as follows:

none: The message was not signed.

pass: The message was signed, the signature or signatures were acceptable to the ADMD, and the signature(s) passed verification tests.

fail: The message was signed and the signature or signatures were acceptable to the ADMD, but they failed the verification test(s).

policy: The message was signed, but some aspect of the signature or signatures was not acceptable to the ADMD.

neutral: The message was signed, but the signature or signatures contained syntax errors or were not otherwise able to be processed. This result is also used for other failures not covered elsewhere in this list.

temperror: The message could not be verified due to some error that is likely transient in nature, such as a temporary inability to retrieve a public key. A later attempt may produce a final result.

permerror: The message could not be verified due to some error that is unrecoverable, such as a required header field being absent. A later attempt is unlikely to produce a final result.

DKIM results are reported using a ptype of "header". The property, however, represents one of the tags found in the DKIM-Signature header field rather than a distinct header field. For example, the

ptype-property combination "header.d" refers to the content of the "d" (signing domain) tag from within the signature header field, and not a distinct header field called "d".

The ability to report different DKIM results for a multiply-signed message is described in [RFC6008].

[DKIM] advises that if a message fails verification, it is to be treated as an unsigned message. A report of "fail" here permits the receiver of the report to decide how to handle the failure. A report of "neutral" or "none" preempts that choice, ensuring the message will be treated as if it had not been signed.

Section 3.1 of [DOMAINKEYS] describes a process by which the sending address of the message is determined. DomainKeys results are thus reported along with the signing domain name, the sending address of the message, and the name of the header field from which the latter was extracted. This means that a DomainKeys result includes a ptype-property combination of "header.d", plus one of "header.from" and "header.sender". The sending address extracted from the header is included with any [MAIL]-style comments removed; moreover, the local-part of the address is removed if it has not been authenticated in some way.

2.7.2. SPF and Sender ID

SPF and Sender ID use the "spf" and "sender-id" method names, respectively. The result values for SPF are defined in Section 2.6 of [SPF], and those definitions are included here by reference:

Code	Meaning
none	[RFC7208], Section 2.6.1
pass	[RFC7208], Section 2.6.3
fail	[RFC7208], Section 2.6.4
softfail	[RFC7208], Section 2.6.5
policy	[this RFC], Section 2.4
neutral	[RFC7208], Section 2.6.2
temperror	[RFC7208], Section 2.6.6
permerror	[RFC7208], Section 2.6.7

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

These result codes are used in the context of this specification to reflect the result returned by the component conducting SPF evaluation.

For SPF, the ptype used is "smtp", and the property is either "mailfrom" or "helo", since those values are the ones SPF can evaluate. (If the SMTP client issued the EHLO command instead of HELO, the property used is "helo".)

The "sender-id" method is described in [SENDERID]. For this method, the ptype used is "header" and the property will be the name of the header field from which the Purported Responsible address (see [PRA]) was extracted, namely one of "Resent-Sender", "Resent-From", "Sender", or "From".

The results for Sender ID are listed and described in Section 4.2 of [SENDERID], but for the purposes of this specification, the SPF definitions enumerated above are used instead. Also, [SENDERID] specifies result codes that use mixed case, but they are typically used all lowercase in this context.

For both methods, an additional result of "policy" is defined, which means the client was authorized to inject or relay mail on behalf of the sender's DNS domain according to the authentication method's algorithm, but local policy dictates that the result is unacceptable. For example, "policy" might be used if SPF returns a "pass" result, but a local policy check matches the sending DNS domain to one found in an explicit list of unacceptable DNS domains (e.g., spammers).

If the retrieved sender policies used to evaluate SPF and Sender ID do not contain explicit provisions for authenticating the local-part (see Section 3.4.1 of [MAIL]) of an address, the "pvalue" reported along with results for these mechanisms SHOULD NOT include the local-part.

2.7.3. "iprev"

The result values used by the "iprev" method, defined in Section 3, are as follows:

pass: The DNS evaluation succeeded, i.e., the "reverse" and "forward" lookup results were returned and were in agreement.

fail: The DNS evaluation failed. In particular, the "reverse" and "forward" lookups each produced results, but they were not in agreement, or the "forward" query completed but produced no result, e.g., a DNS RCODE of 3, commonly known as NXDOMAIN, or an RCODE of 0 (NOERROR) in a reply containing no answers, was returned.

temperror: The DNS evaluation could not be completed due to some error that is likely transient in nature, such as a temporary DNS error, e.g., a DNS RCODE of 2, commonly known as SERVFAIL, or other error condition resulted. A later attempt may produce a final result.

permerror: The DNS evaluation could not be completed because no PTR data are published for the connecting IP address, e.g., a DNS RCODE of 3, commonly known as NXDOMAIN, or an RCODE of 0 (NOERROR) in a reply containing no answers, was returned. This prevented completion of the evaluation. A later attempt is unlikely to produce a final result.

There is no "none" for this method since any TCP connection delivering email has an IP address associated with it, so some kind of evaluation will always be possible.

The result is reported using a ptype of "policy" (as this is not part of any established protocol) and a property of "iprev".

For discussion of the format of DNS replies, see "Domain Names - Implementation and Specification" ([DNS]).

2.7.4. SMTP AUTH

SMTP AUTH (defined in [AUTH]) is represented by the "auth" method, and its result values are as follows:

none: SMTP authentication was not attempted.

pass: The SMTP client authenticated to the server reporting the result using the protocol described in [AUTH].

fail: The SMTP client attempted to authenticate to the server using the protocol described in [AUTH] but was not successful, yet continued to send the message about which a result is being reported.

temperror: The SMTP client attempted to authenticate using the protocol described in [AUTH] but was not able to complete the attempt due to some error that is likely transient in nature, such as a temporary directory service lookup error. A later attempt may produce a final result.

permerror: The SMTP client attempted to authenticate using the protocol described in [AUTH] but was not able to complete the attempt due to some error that is likely not transient in nature, such as a permanent directory service lookup error. A later attempt is not likely to produce a final result.

The result of AUTH is reported using a ptype of "smtp" and a property of either:

- "auth", in which case the value is the authorization identity generated by the exchange initiated by the AUTH command; or

- "mailfrom", in which case the value is the mailbox identified by the AUTH parameter used with the MAIL FROM command.

If both identities are available, both can be reported. For example, consider this command issued by a client that has completed session authentication with the AUTH command resulting in an authorized identity of "client@c.example":

```
MAIL FROM:<alice@a.example> AUTH=<bob@b.example>
```

This could result in a resinfo construction like so:

```
; auth=pass smtp.auth=client@c.example smtp.mailfrom=bob@b.example
```

An agent making use of the data provided by this header field SHOULD consider "fail" and "temperror" to be synonymous in terms of message authentication, i.e., the client did not authenticate in either case.

2.7.5. Other Registered Codes

Result codes were also registered in other RFCs as follows:

- Vouch By Reference (in [AR-VBR], represented by "vbr");

- Authorized Third-Party Signatures (in [ATPS], represented by "dkim-atps");

- Author Domain Signing Practices (in [ADSP], represented by "dkim-adsp");

- Require-Recipient-Valid-Since (in [RRVS], represented by "rrvs");

- S/MIME (in [SMIME-REG], represented by "smime").

2.7.6. Extension Methods

Additional authentication method identifiers (extension methods) may be defined in the future by later revisions or extensions to this specification. These method identifiers are registered with the Internet Assigned Numbers Authority (IANA) and, preferably, published in an RFC. See Section 6 for further details.

Extension methods can be defined for the following reasons:

1. To allow additional information from new authentication systems to be communicated to MUAs or downstream filters. The names of such identifiers ought to reflect the name of the method being defined but ought not be needlessly long.

2. To allow the creation of "sub-identifiers" that indicate different levels of authentication and differentiate between their relative strengths, e.g., "auth1-weak" and "auth1-strong".

Authentication method implementers are encouraged to provide adequate information, via message header field comments if necessary, to allow an MUA developer to understand or relay ancillary details of authentication results. For example, if it might be of interest to relay what data was used to perform an evaluation, such information could be relayed as a comment in the header field, such as:

```
Authentication-Results: example.com;
          foo=pass bar.baz=blob (2 of 3 tests OK)
```

Experimental method identifiers MUST only be used within ADMDs that have explicitly consented to use them. These method identifiers and the parameters associated with them are not documented in RFCs. Therefore, they are subject to change at any time and not suitable for production use. Any MTA, MUA, or downstream filter intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an experimental (unknown) method identifier.

2.7.7. Extension Result Codes

Additional result codes (extension results) might be defined in the future by later revisions or extensions to this specification. Result codes MUST be registered with the Internet Assigned Numbers Authority (IANA) and preferably published in an RFC. See Section 6

for further details.

Experimental results MUST only be used within ADMDs that have explicitly consented to use them. These results and the parameters associated with them are not formally documented. Therefore, they are subject to change at any time and not suitable for production use. Any MTA, MUA, or downstream filter intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an extension result.

3. The "iprev" Authentication Method

This section defines an additional authentication method called "iprev".

"iprev" is an attempt to verify that a client appears to be valid based on some DNS queries, which is to say that the IP address is explicitly associated with a domain name. Upon receiving a session initiation of some kind from a client, the IP address of the client peer is queried for matching names (i.e., a number-to-name translation, also known as a "reverse lookup" or a "PTR" record query). Once that result is acquired, a lookup of each of the names (i.e., a name-to-number translation, or an "A" or "AAAA" record query) thus retrieved is done. The response to this second check will typically result in at least one mapping back to the client's IP address.

Expressed as an algorithm: If the client peer's IP address is I, the list of names to which I maps (after a "PTR" query) is the set N, and the union of IP addresses to which each member of N maps (after corresponding "A" and "AAAA" queries) is L, then this test is successful if I is an element of L.

The response to a PTR query could contain multiple names. To prevent heavy DNS loads, agents performing these queries MUST be implemented such that the number of names evaluated by generation of corresponding A or AAAA queries is limited so as not to be unduly taxing to the DNS infrastructure, though it MAY be configurable by an administrator. As an example, Section 4.6.4 of [SPF] chose a limit of 10 for its implementation of this algorithm.

"DNS Extensions to Support IP Version 6" ([DNS-IP6]) discusses the query formats for the IPv6 case.

There is some contention regarding the wisdom and reliability of this test. For example, in some regions, it can be difficult for this test ever to pass because the practice of arranging to match the forward and reverse DNS is infrequently observed. Therefore, the

precise implementation details of how a verifier performs an "iprev" test are not specified here. The verifier MAY report a successful or failed "iprev" test at its discretion having done some kind of check of the validity of the connection's identity using DNS. It is incumbent upon an agent making use of the reported "iprev" result to understand what exactly that particular verifier is attempting to report.

Extensive discussion of reverse DNS mapping and its implications can be found in "Considerations for the use of DNS Reverse Mapping" ([DNSOP-REVERSE]). In particular, it recommends that applications avoid using this test as a means of authentication or security. Its presence in this document is not an endorsement but is merely acknowledgment that the method remains common and provides the means to relay the results of that test.

4. Adding the Header Field to a Message

This specification makes no attempt to evaluate the relative strengths of various message authentication methods that may become available. The methods listed are an order-independent set; their sequence does not indicate relative strength or importance of one method over another. Instead, the MUA or downstream filter consuming this header field is to interpret the result of each method based on its own knowledge of what that method evaluates.

Each "method" MUST refer to an authentication method declared in the IANA registry or an extension method as described in Section 2.7.6, and each "result" MUST refer to a result code declared in the IANA registry or an extension result code as defined in Section 2.7.7. See Section 6 for further information about the registered methods and result codes.

An MTA compliant with this specification adds this header field (after performing one or more message authentication tests) to indicate which MTA or ADMD performed the test, which test got applied, and what the result was. If an MTA applies more than one such test, it adds this header field either once per test or once indicating all of the results. An MTA MUST NOT add a result to an existing header field.

An MTA MAY add this header field containing only the authentication identifier portion and the "none" token (see Section 2.2) to indicate explicitly that no message authentication schemes were applied prior to delivery of this message.

An MTA adding this header field has to take steps to identify it as legitimate to the MUAs or downstream filters that will ultimately

consume its content. One process to do so is described in Section 5. Further measures may be necessary in some environments. Some possible solutions are enumerated in Section 7.1. This document does not mandate any specific solution to this issue as each environment has its own facilities and limitations.

Most known message authentication methods focus on a particular identifier to evaluate. SPF and Sender ID differ in that they can yield a result based on more than one identifier; specifically, SPF can evaluate the RFC5321.HELO parameter or the RFC5321.MailFrom parameter, and Sender ID can evaluate the RFC5321.MailFrom parameter or the Purported Responsible Address (PRA) identity. When generating this field to report those results, only the parameter that yielded the result is included.

For MTAs that add this header field, adding header fields in order (at the top), per Section 3.6 of [MAIL], is particularly important. Moreover, this header field SHOULD be inserted above any other trace header fields such MTAs might prepend. This placement allows easy detection of header fields that can be trusted.

End users making direct use of this header field might inadvertently trust information that has not been properly vetted. If, for example, a basic SPF result were to be relayed that claims an authenticated addr-spec, the local-part of that addr-spec has actually not been authenticated. Thus, an MTA adding this header field SHOULD NOT include any data that has not been authenticated by the method(s) being applied. Moreover, MUAs SHOULD NOT render to users such information if it is presented by a method known not to authenticate it.

4.1. Header Field Position and Interpretation

In order to ensure non-ambiguous results and avoid the impact of false header fields, MUAs and downstream filters SHOULD NOT interpret this header field unless specifically configured to do so by the user or administrator. That is, this interpretation should not be "on by default". Naturally then, users or administrators ought not activate such a feature unless they are certain the header field will be validly added by an agent within the ADMD that accepts the mail that is ultimately read by the MUA, and instances of the header field appearing to originate within the ADMD but are actually added by foreign MTAs will be removed before delivery.

Furthermore, MUAs and downstream filters SHOULD NOT interpret this header field unless the authentication service identifier it bears appears to be one used within its own ADMD as configured by the user or administrator.

MUAs and downstream filters MUST ignore any result reported using a "result" not specified in the IANA "Result Code" registry or a "ptype" not listed in the corresponding registry for such values as defined in Section 6. Moreover, such agents MUST ignore a result indicated for any "method" they do not specifically support.

An MUA SHOULD NOT reveal these results to end users, absent careful human factors design considerations and testing, for the presentation of trust-related materials. For example, an attacker could register examp1e.com (note the digit "one") and send signed mail to intended victims; a verifier would detect that the signature was valid and report a "pass" even though it's clear the DNS domain name was intended to mislead. See Section 7.2 for further discussion.

As stated in Section 2.1, this header field MUST be treated as though it were a trace header field as defined in Section 3.6.7 of [MAIL] and hence MUST NOT be reordered and MUST be prepended to the message, so that there is generally some indication upon delivery of where in the chain of handling MTAs the message authentication was done.

Note that there are a few message handlers that are only capable of appending new header fields to a message. Strictly speaking, these handlers are not compliant with this specification. They can still add the header field to carry authentication details, but any signal about where in the handling chain the work was done may be lost. Consumers SHOULD be designed such that this can be tolerated, especially from a producer known to have this limitation.

MUAs SHOULD ignore instances of this header field discovered within message/rfc822 MIME attachments.

Further discussion of these topics can be found in Section 7 below.

4.2. Local Policy Enforcement

Some sites have a local policy that considers any particular authentication policy's non-recoverable failure results (typically "fail" or similar) as justification for rejecting the message. In such cases, the border MTA SHOULD issue an SMTP rejection response to the message, rather than adding this header field and allowing the message to proceed toward delivery. This is more desirable than allowing the message to reach an internal host's MTA or spam filter, thus possibly generating a local rejection such as a Delivery Status Notification (DSN) [DSN] to a forged originator. Such generated rejections are colloquially known as "backscatter".

The same MAY also be done for local policy decisions overriding the results of the authentication methods (e.g., the "policy" result

codes described in Section 2.7).

Such rejections at the SMTP protocol level are not possible if local policy is enforced at the MUA and not the MTA.

5. Removing Existing Header Fields

For security reasons, any MTA conforming to this specification MUST delete any discovered instance of this header field that claims, by virtue of its authentication service identifier, to have been added within its trust boundary but that did not come directly from another trusted MTA. For example, an MTA for example.com receiving a message MUST delete or otherwise obscure any instance of this header field bearing an authentication service identifier indicating that the header field was added within example.com prior to adding its own header fields. This could mean each MTA will have to be equipped with a list of internal MTAs known to be compliant (and hence trustworthy).

For simplicity and maximum security, a border MTA could remove all instances of this header field on mail crossing into its trust boundary. However, this may conflict with the desire to access authentication results performed by trusted external service providers. It may also invalidate signed messages whose signatures cover external instances of this header field. A more robust border MTA could allow a specific list of authenticating MTAs whose information is to be admitted, removing the header field originating from all others.

As stated in Section 1.2, a formal definition of "trust boundary" is deliberately not made here. It is entirely possible that a border MTA for example.com will explicitly trust authentication results asserted by upstream host example.net even though they exist in completely disjoint administrative boundaries. In that case, the border MTA MAY elect not to delete those results; moreover, the upstream host doing some authentication work could apply a signing technology such as [DKIM] on its own results to assure downstream hosts of their authenticity. An example of this is provided in Appendix C.

Similarly, in the case of messages signed using [DKIM] or other message-signing methods that sign header fields, this removal action could invalidate one or more signatures on the message if they covered the header field to be removed. This behavior can be desirable since there's little value in validating the signature on a message with forged header fields. However, signing agents MAY therefore elect to omit these header fields from signing to avoid this situation.

An MTA SHOULD remove any instance of this header field bearing a version (express or implied) that it does not support. However, an MTA MUST remove such a header field if the [SMTP] connection relaying the message is not from a trusted internal MTA. This means the MTA needs to be able to understand versions of this header field at least as late as the ones understood by the MUAs or other consumers within its ADMD.

6. IANA Considerations

IANA has registered the defined header field and created tables as described below. These registry actions were originally defined by [RFC5451] and updated by [RFC6577] and [RFC7001]. The created registries are being further updated here to increase their completeness.

6.1. The Authentication-Results Header Field

[RFC5451] added the Authentication-Results header field to the IANA "Permanent Message Header Field Names" registry, per the procedure found in [IANA-HEADERS]. That entry is to be updated to reference this document. The following is the registration template:

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

6.2. "Email Authentication Methods" Registry Description

Names of message authentication methods supported by this specification are to be registered with IANA, with the exception of experimental names as described in Section 2.7.6. Along with each method is recorded the properties that accompany the method's result.

The "Email Authentication Parameters" group, and within it the "Email Authentication Methods" registry, were created by [RFC5451] for this purpose. [RFC6577] added a "status" field for each entry. [RFC7001] amended the rules governing that registry, and also added a "version" field to the registry.

The reference for that registry shall be updated to reference this document.

New entries are assigned only for values that have received Expert Review, per [IANA-CONSIDERATIONS]. The designated expert shall be appointed by the IESG. The designated expert has discretion to request that a publication be referenced if a clear, concise definition of the authentication method cannot be provided such that interoperability is assured. Registrations should otherwise be permitted. The designated expert can also handle requests to mark any current registration as "deprecated".

No two entries can have the same combination of method, ptype, and property.

An entry in this registry contains the following:

Method: the name of the method;

Defined: a reference to the document that created this entry, if any (see below);

ptype: a "ptype" value appropriate for use with that method;

property: a "property" value matching that "ptype" also appropriate for use with that method;

Value: a brief description of the value to be supplied with that method/ptype/property tuple;

Status: the status of this entry, which is either:

active: The entry is in current use.

deprecated: The entry is no longer in current use.

Version: a version number associated with the method (preferably starting at "1").

The "Defined" field will typically refer to a permanent document, or at least some descriptive text, where additional information about the entry being added can be found. This in turn would reference the document where the method is defined so that all of the semantics around creating or interpreting an Authentication-Results header field using this method, ptype, and property can be understood.

6.3. "Email Authentication Methods" Registry Update

The following changes are to be made to this registry upon approval of this document:

1. The current entry for the "auth" method shall have its "property" field changed to "mailfrom", and its "Defined" field changed to this document.

2. The entry for the "dkim" method, "header" ptype and "b" property shall now reference [RFC6008] as its defining document, and the reference shall be removed from the description.

3. All other "dkim", "domainkeys", "iprev", "sender-id", and "spf" method entries shall have their "Defined" fields changed to this document.

4. All "smime" entries have their "Defined" fields changed to [SMIME-REG].

5. The "value" field of the "smime" entry using property "smime-part" shall be changed to read "A reference to the MIME body part that contains the signature." The redundant reference is thus removed.

6. The following entry is to be added:

   Method: auth

   Defined: [this document]

   ptype: smtp

   property: auth

   Value: identity confirmed by the AUTH command

   Status: active

   Version: 1

7. The values of the "domainkeys" entries for ptype "header" are updated as follows:

   from: contents of the [MAIL] From: header field, after removing comments, and removing the local-part if not authenticated

   sender: contents of the [MAIL] Sender: header field, after removing comments, and removing the local-part if not authenticated

8. All entries for "dkim-adsp" and "domainkeys" shall have their Status values changed to "deprecated", reflecting the fact that the corresponding specifications now have Historical status.

6.4. "Email Authentication Property Types" Registry

[PTYPES-REGISTRY] created the Email Authentication Property Types registry.

Entries in this registry are subject to the Expert Review rules as described in [IANA-CONSIDERATIONS]. Each entry in the registry requires the following values:

ptype: The name of the ptype being registered, which must fit within the ABNF described in Section 2.2.

Definition: An optional reference to a defining specification.

Description: A brief description of what sort of information this "ptype" is meant to cover.

For new entries, the Designated Expert needs to assure that the description provided for the new entry adequately describes the intended use. An example would be helpful to include in the entry's defining document, if any, although entries in the "Email Authentication Methods" registry or the "Email Authentication Result Names" registry might also serve as examples of intended use.

IANA shall update this registry to show Section 2.3 of this document as the current definitions for the "body", "header", "policy" and "smtp" entries of that registry.

6.5. "Email Authentication Result Names" Description

Names of message authentication result codes supported by this specification must be registered with IANA, with the exception of experimental codes as described in Section 2.7.7. A registry was created by [RFC5451] for this purpose. [RFC6577] added the "status" column, and [RFC7001] updated the rules governing that registry.

New entries are assigned only for values that have received Expert Review, per [IANA-CONSIDERATIONS]. The designated expert shall be appointed by the IESG. The designated expert has discretion to request that a publication be referenced if a clear, concise definition of the authentication result cannot be provided such that interoperability is assured. Registrations should otherwise be permitted. The designated expert can also handle requests to mark any current registration as "deprecated".

No two entries can have the same combination of method and code.

An entry in this registry contains the following:

Auth Method: an authentication method for which results are being returned using the header field defined in this document;

Code: a result code that can be returned for this authentication method;

Specification: either free form text explaining the meaning of this method-code combination, or a reference to such a definition.

6.6. "Email Authentication Result Names" Update

The following changes are to be made to this registry on publication of this document:

- The "Defined" field shall be removed.
- The "Meaning" field shall be renamed to "Specification", as described above.
- The "Auth Method" field shall appear before the "Code" field.
- For easier searching, the table shall be arranged such that it is sorted first by Auth Method, then by Code within each Auth Method grouping.
- All entries for the "dkim", "domainkeys", "spf", "sender-id", "auth", and "iprev" methods shall have their "Specification" fields changed to refer to this document, as follows:

  dkim: [this document] Section 2.7.1

  domainkeys: [this document] Section 2.7.1

  spf: [this document] Section 2.7.2

  sender-id: [this document] Section 2.7.2

  auth: [this document] Section 2.7.4

  iprev: [this document] Section 2.7.3

- All entries for "dkim-adsp" that are missing an explicit reference to a defining document shall have [ADSP] added to their Specification fields.

- o All entries for "dkim-adsp" and "domainkeys" shall have their Status values changed to "deprecated", reflecting the fact that the corresponding specifications now have Historical status.

7. Security Considerations

The following security considerations apply when adding or processing the Authentication-Results header field:

7.1. Forged Header Fields

An MUA or filter that accesses a mailbox whose messages are handled by a non-conformant MTA, and understands Authentication-Results header fields, could potentially make false conclusions based on forged header fields. A malicious user or agent could forge a header field using the DNS domain of a receiving ADMD as the authserv-id token in the value of the header field and, with the rest of the value, claim that the message was properly authenticated. The non-conformant MTA would fail to strip the forged header field, and the MUA could inappropriately trust it.

For this reason, it is best not to have processing of the Authentication-Results header field enabled by default; instead, it should be ignored, at least for the purposes of enacting filtering decisions, unless specifically enabled by the user or administrator after verifying that the border MTA is compliant. It is acceptable to have an MUA aware of this specification but have an explicit list of hostnames whose Authentication-Results header fields are trustworthy; however, this list should initially be empty.

Proposed alternative solutions to this problem were made some time ago and are listed below. To date, they have not been developed due to lack of demand but are documented here should the information be useful at some point in the future:

1. Possibly the simplest is a digital signature protecting the header field, such as using [DKIM], that can be verified by an MUA by using a posted public key. Although one of the main purposes of this document is to relieve the burden of doing message authentication work at the MUA, this only requires that the MUA learn a single authentication scheme even if a number of them are in use at the border MTA. Note that [DKIM] requires that the From header field be signed, although in this application, the signing agent (a trusted MTA) likely cannot authenticate that value, so the fact that it is signed should be ignored. Where the authserv-id is the ADMD's domain name, the authserv-id matching this valid internal signature's "d=" DKIM value is sufficient.

2. Another would be a means to interrogate the MTA that added the header field to see if it is actually providing any message authentication services and saw the message in question, but this isn't especially palatable given the work required to craft and implement such a scheme.

3. Yet another might be a method to interrogate the internal MTAs that apparently handled the message (based on Received header fields) to determine whether any of them conform to Section 5 of this memo. This, too, has potentially high barriers to entry.

4. Extensions to [IMAP], [SMTP], and [POP3] could be defined to allow an MUA or filtering agent to acquire the authserv-id in use within an ADMD, thus allowing it to identify which Authentication-Results header fields it can trust.

5. On the presumption that internal MTAs are fully compliant with Section 3.6 of [MAIL] and the compliant internal MTAs are using their own hostnames or the ADMD's DNS domain name as the authserv-id token, the header field proposed here should always appear above a Received header added by a trusted MTA. This can be used as a test for header field validity.

Support for some of these is being considered for future work.

In any case, a mechanism needs to exist for an MUA or filter to verify that the host that appears to have added the header field (a) actually did so and (b) is legitimately adding that header field for this delivery. Given the variety of messaging environments deployed today, consensus appears to be that specifying a particular mechanism for doing so is not appropriate for this document.

Mitigation of the forged header field attack can also be accomplished by moving the authentication results data into meta-data associated with the message. In particular, an [SMTP] extension could be established to communicate authentication results from the border MTA to intermediate and delivery MTAs; the latter of these could arrange to store the authentication results as meta-data retrieved and rendered along with the message by an [IMAP] client aware of a similar extension in that protocol. The delivery MTA would be told to trust data via this extension only from MTAs it trusts, and border MTAs would not accept data via this extension from any source. There is no vector in such an arrangement for forgery of authentication data by an outside agent.

7.2. Misleading Results

Until some form of service for querying the reputation of a sending agent is widely deployed, the existence of this header field indicating a "pass" does not render the message trustworthy. It is possible for an arriving piece of spam or other undesirable mail to pass checks by several of the methods enumerated above (e.g., a piece of spam signed using [DKIM] by the originator of the spam, which might be a spammer or a compromised system). In particular, this issue is not resolved by forged header field removal discussed above.

Hence, MUAs and downstream filters must take some care with use of this header even after possibly malicious headers are scrubbed.

7.3. Header Field Position

Despite the requirements of [MAIL], header fields can sometimes be reordered en route by intermediate MTAs. The goal of requiring header field addition only at the top of a message is an acknowledgment that some MTAs do reorder header fields, but most do not. Thus, in the general case, there will be some indication of which MTAs (if any) handled the message after the addition of the header field defined here.

7.4. Reverse IP Query Denial-of-Service Attacks

Section 4.6.4 of [SPF] describes a DNS-based denial-of-service attack for verifiers that attempt DNS-based identity verification of arriving client connections. A verifier wishing to do this check and report this information needs to take care not to go to unbounded lengths to resolve "A" and "PTR" queries. MUAs or other filters making use of an "iprev" result specified by this document need to be aware of the algorithm used by the verifier reporting the result and, especially, its limitations.

7.5. Mitigation of Backscatter

Failing to follow the instructions of Section 4.2 can result in a denial-of-service attack caused by the generation of [DSN] messages (or equivalent) to addresses that did not send the messages being rejected.

7.6. Internal MTA Lists

Section 5 describes a procedure for scrubbing header fields that may contain forged authentication results about a message. A compliant installation will have to include, at each MTA, a list of other MTAs known to be compliant and trustworthy. Failing to keep this list

current as internal infrastructure changes may expose an ADMD to attack.

7.7. Attacks against Authentication Methods

If an attack becomes known against an authentication method, clearly then the agent verifying that method can be fooled into thinking an inauthentic message is authentic, and thus the value of this header field can be misleading. It follows that any attack against the authentication methods supported by this document is also a security consideration here.

7.8. Intentionally Malformed Header Fields

It is possible for an attacker to add an Authentication-Results header field that is extraordinarily large or otherwise malformed in an attempt to discover or exploit weaknesses in header field parsing code. Implementers must thoroughly verify all such header fields received from MTAs and be robust against intentionally as well as unintentionally malformed header fields.

7.9. Compromised Internal Hosts

An internal MUA or MTA that has been compromised could generate mail with a forged From header field and a forged Authentication-Results header field that endorses it. Although it is clearly a larger concern to have compromised internal machines than it is to prove the value of this header field, this risk can be mitigated by arranging that internal MTAs will remove this header field if it claims to have been added by a trusted border MTA (as described above), yet the [SMTP] connection is not coming from an internal machine known to be running an authorized MTA. However, in such a configuration, legitimate MTAs will have to add this header field when legitimate internal-only messages are generated. This is also covered in Section 5.

7.10. Encapsulated Instances

MIME messages can contain attachments of type "message/rfc822", which contain other messages. Such an encapsulated message can also contain an Authentication-Results header field. Although the processing of these is outside of the intended scope of this document (see Section 1.3), some early guidance to MUA developers is appropriate here.

Since MTAs are unlikely to strip Authentication-Results header fields after mailbox delivery, MUAs are advised in Section 4.1 to ignore such instances within MIME attachments. Moreover, when extracting a

message digest to separate mail store messages or other media, such header fields should be removed so that they will never be interpreted improperly by MUAs that might later consume them.

7.11. Reverse Mapping

Although Section 3 of this memo includes explicit support for the "iprev" method, its value as an authentication mechanism is limited. Implementers of both this proposal and agents that use the data it relays are encouraged to become familiar with the issues raised by [DNSOP-REVERSE] when deciding whether or not to include support for "iprev".

8. References

8.1. Normative References

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

[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.

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

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

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

8.2. Informative References

[ADSP] Allman, E., Fenton, J., Delany, M., and J. Levine, "DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP)", RFC 5617, August 2009.

[AR-VBR] Kucherawy, M., "Authentication-Results

Registration for Vouch by Reference Results", RFC 6212, April 2011.

[ATPS] Kucherawy, M., "DomainKeys Identified Mail (DKIM) Authorized Third-Party Signatures", RFC 6541, February 2012.

[AUTH] Siemborski, R. and A. Melnikov, "SMTP Service Extension for Authentication", RFC 4954, July 2007.

[DKIM] Crocker, D., Hansen, T., and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, September 2011.

[DMARC] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting and Conformance (DMARC)", RFC 7489, March 2015.

[DNS] Mockapetris, P., "Domain names - Implementation and Specification", STD 13, RFC 1035, November 1987.

[DNS-IP6] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[DNSOP-REVERSE] Senie, D. and A. Sullivan, "Considerations for the use of DNS Reverse Mapping", Work in Progress, March 2008.

[DOMAINKEYS] Delany, M., "Domain-Based Email Authentication Using Public Keys Advertised in the DNS (DomainKeys)", RFC 4870, May 2007.

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

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

[IANA-CONSIDERATIONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[IMAP] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL

- VERSION 4rev1", RFC 3501, March 2003.

[POP3] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[PRA] Lyon, J., "Purported Responsible Address in E-Mail Messages", RFC 4407, April 2006.

[PTYPES-REGISTRY] Kucherawy, M., "A Property Types Registry for the Authentication-Results Header Field", RFC 7410, December 2014.

[RFC5451] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 5451, April 2009.

[RFC6008] Kucherawy, M., "Authentication-Results Registration for Differentiating among Cryptographic Results", RFC 6008, September 2010.

[RFC6577] Kucherawy, M., "Authentication-Results Registration Update for Sender Policy Framework (SPF) Results", RFC 6577, March 2012.

[RFC7001] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 7001, September 2013.

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

[SECURITY] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[SENDERID] Lyon, J. and M. Wong, "Sender ID: Authenticating E-Mail", RFC 4406, April 2006.

[SMIME-REG] Melnikov, A., "Authentication-Results Registration for S/MIME Signature Verification", RFC 7281, June 2014.

[SPF] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1", RFC 7208, April 2014.

[VBR] Hoffman, P., Levine, J., and A. Hathcock, "Vouch By Reference", RFC 5518, April 2009.

Appendix A. Acknowledgments

The author wishes to acknowledge the following individuals for their review and constructive criticism of this document: Stephane Bortzmeyer, Scott Kitterman, John Levine, Tom Petch, and Pete Resnick.

Appendix B. Legacy MUAs

Implementers of this protocol should be aware that many MUAs are unlikely to be retrofitted to support the new header field and its semantics. In the interests of convenience and quicker adoption, a delivery MTA might want to consider adding things that are processed by existing MUAs in addition to the Authentication-Results header field. One suggestion is to include a Priority header field, on messages that don't already have such a header field, containing a value that reflects the strength of the authentication that was accomplished, e.g., "low" for weak or no authentication, "normal" or "high" for good or strong authentication.

Some modern MUAs can already filter based on the content of this header field. However, there is keen interest in having MUAs make some kind of graphical representation of this header field's meaning to end users. Until this capability is added, other interim means of conveying authentication results may be necessary while this proposal and its successors are adopted.

Appendix C. Authentication-Results Examples

This section presents some examples of the use of this header field to indicate authentication results.

C.1. Trivial Case; Header Field Not Present

The trivial case:

```
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by server.example.org (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.org
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 1: Trivial Case

The Authentication-Results header field is completely absent. The MUA may make no conclusion about the validity of the message. This could be the case because the message authentication services were not available at the time of delivery, or no service is provided, or the MTA is not in compliance with this specification.

C.2. Nearly Trivial Case; Service Provided, but No Authentication Done

A message that was delivered by an MTA that conforms to this specification but provides no actual message authentication service:

```
Authentication-Results: example.org 1; none
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by server.example.org (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.org
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 2: Header Present but No Authentication Done

The Authentication-Results header field is present, showing that the delivering MTA conforms to this specification. It used its DNS

domain name as the authserv-id. The presence of "none" (and the absence of any method and result tokens) indicates that no message authentication was done. The version number of the specification to which the field's content conforms is explicitly provided.

C.3. Service Provided, Authentication Done

A message that was delivered by an MTA that conforms to this specification and applied some message authentication:

```
Authentication-Results: example.com;
          spf=pass smtp.mailfrom=example.net
Received: from dialup-1-2-3-4.example.net
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.net
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
Message-Id: <12345.abc@example.net>
Subject: here's a sample

Hello!  Goodbye!
```

Example 3: Header Reporting Results

The Authentication-Results header field is present, indicating that the border MTA conforms to this specification. The authserv-id is once again the DNS domain name. Furthermore, the message was authenticated by that MTA via the method specified in [SPF]. Note that since that method cannot authenticate the local-part, it has been omitted from the result's value. The MUA could extract and relay this extra information if desired.

C.4. Service Provided, Several Authentications Done, Single MTA

A message that was relayed inbound via a single MTA that conforms to this specification and applied three different message authentication checks:

```
Authentication-Results: example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.net;
          spf=pass smtp.mailfrom=example.net
Authentication-Results: example.com;
          sender-id=pass header.from=example.net
Received: from dialup-1-2-3-4.example.net (8.11.6/8.11.6)
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
From: sender@example.net
Message-Id: <12345.abc@example.net>
Subject: here's a sample

Hello!  Goodbye!
```

Example 4: Headers Reporting Results from One MTA

The Authentication-Results header field is present, indicating that the delivering MTA conforms to this specification. Once again, the receiving DNS domain name is used as the authserv-id. Furthermore, the sender authenticated herself/himself to the MTA via a method specified in [AUTH], and both SPF and Sender ID checks were done and passed. The MUA could extract and relay this extra information if desired.

Two Authentication-Results header fields are not required since the same host did all of the checking. The authenticating agent could have consolidated all the results into one header field.

This example illustrates a scenario in which a remote user on a dialup connection (example.net) sends mail to a border MTA (example.com) using SMTP authentication to prove identity. The dialup provider has been explicitly authorized to relay mail as example.com resulting in passes by the SPF and Sender ID checks.

C.5. Service Provided, Several Authentications Done, Different MTAs

A message that was relayed inbound by two different MTAs that conform to this specification and applied multiple message authentication checks:

```
Authentication-Results: example.com;
          sender-id=fail header.from=example.com;
          dkim=pass (good signature) header.d=example.com
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by auth-checker.example.com (8.11.6/8.11.6)
              with ESMTP id i7PK0sH7021929;
          Fri, Feb 15 2002 17:19:22 -0800
DKIM-Signature:  v=1; a=rsa-sha256; s=gatsby; d=example.com;
          t=1188964191; c=simple/simple; h=From:Date:To:Subject:
          Message-Id:Authentication-Results;
          bh=sEuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m70;
          b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
Authentication-Results: example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.com;
          spf=fail smtp.mailfrom=example.com
Received: from dialup-1-2-3-4.example.net
              (dialup-1-2-3-4.example.net [192.0.2.200])
          by mail-router.example.com (8.11.6/8.11.6)
              with ESMTP id g1G0r1kA003489;
          Fri, Feb 15 2002 17:19:07 -0800
From: sender@example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: receiver@example.com
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 5: Headers Reporting Results from Multiple MTAs

The Authentication-Results header field is present, indicating conformance to this specification. Once again, the authserv-id used is the recipient's DNS domain name. The header field is present twice because two different MTAs in the chain of delivery did authentication tests. The first MTA, mail-router.example.com, reports that SMTP AUTH and SPF were both used and that the former passed while the latter failed. In the SMTP AUTH case, additional information is provided in the comment field, which the MUA can choose to render if desired.

The second MTA, auth-checker.example.com, reports that it did a

Sender ID test (which failed) and a DKIM test (which passed). Again, additional data about one of the tests is provided as a comment, which the MUA may choose to render. Also noteworthy here is the fact that there is a DKIM signature added by example.com that assured the integrity of the lower Authentication-Results field.

Since different hosts did the two sets of authentication checks, the header fields cannot be consolidated in this example.

This example illustrates more typical transmission of mail into example.com from a user on a dialup connection example.net. The user appears to be legitimate as he/she had a valid password allowing authentication at the border MTA using SMTP AUTH. The SPF and Sender ID tests failed since example.com has not granted example.net authority to relay mail on its behalf. However, the DKIM test passed because the sending user had a private key matching one of example.com's published public keys and used it to sign the message.

C.6. Service Provided, Multi-Tiered Authentication Done

A message that had authentication done at various stages, one of which was outside the receiving ADMD:

```
Authentication-Results: example.com;
      dkim=pass reason="good signature"
        header.i=@mail-router.example.net;
      dkim=fail reason="bad signature"
        header.i=@newyork.example.com
Received: from mail-router.example.net
          (mail-router.example.net [192.0.2.250])
      by chicago.example.com (8.11.6/8.11.6)
          for <recipient@chicago.example.com>
          with ESMTP id i7PK0sH7021929;
      Fri, Feb 15 2002 17:19:22 -0800
DKIM-Signature: v=1; a=rsa-sha256; s=furble;
      d=mail-router.example.net; t=1188964198; c=relaxed/simple;
      h=From:Date:To:Message-Id:Subject:Authentication-Results;
      bh=ftA9J6GtX8OpwUECzHnCkRzKw1uk6FNiLfJl5Nmv49E=;
      b=oINEO8hgn/gnunsg ... 9n9ODSNFSDij3=
Authentication-Results: example.net;
      dkim=pass (good signature) header.i=@newyork.example.com
Received: from smtp.newyork.example.com
          (smtp.newyork.example.com [192.0.2.220])
      by mail-router.example.net (8.11.6/8.11.6)
          with ESMTP id g1G0r1kA003489;
      Fri, Feb 15 2002 17:19:07 -0800
DKIM-Signature: v=1; a=rsa-sha256; s=gatsby;
      d=newyork.example.com;
      t=1188964191; c=simple/simple;
      h=From:Date:To:Message-Id:Subject;
      bh=sEu28nfs9fuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m7=;
      b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
From: sender@newyork.example.com
Date: Fri, Feb 15 2002 16:54:30 -0800
To: meetings@example.net
Message-Id: <12345.abc@newyork.example.com>
Subject: here's a sample
```

Example 6: Headers Reporting Results from Multiple MTAs in Different ADMDs

In this example, we see multi-tiered authentication with an extended trust boundary.

The message was sent from someone at example.com's New York office (newyork.example.com) to a mailing list managed at an intermediary.

The message was signed at the origin using DKIM.

The message was sent to a mailing list service provider called example.net, which is used by example.com. There, meetings@example.net is expanded to a long list of recipients, one of whom is at the Chicago office. In this example, we will assume that the trust boundary for chicago.example.com includes the mailing list server at example.net.

The mailing list server there first authenticated the message and affixed an Authentication-Results header field indicating such using its DNS domain name for the authserv-id. It then altered the message by affixing some footer text to the body, including some administrivia such as unsubscription instructions. Finally, the mailing list server affixes a second DKIM signature and begins distribution of the message.

The border MTA for chicago.example.com explicitly trusts results from mail-router.example.net, so that header field is not removed. It performs evaluation of both signatures and determines that the first (most recent) is a "pass" but, because of the aforementioned modifications, the second is a "fail". However, the first signature included the Authentication-Results header added at mail-router.example.net that validated the second signature. Thus, indirectly, it can be determined that the authentications claimed by both signatures are indeed valid.

Note that two styles of presenting meta-data about the result are in use here. In one case, the "reason=" clause is present, which is intended for easy extraction by parsers; in the other case, the CFWS production of the ABNF is used to include such data as a header field comment. The latter can be harder for parsers to extract given the varied supported syntaxes of mail header fields.

C.7. Comment-Heavy Example

The formal syntax permits comments within the content in a number of places. For the sake of illustration, this example is also legal:

```
Authentication-Results: foo.example.net (foobar) 1 (baz);
    dkim (Because I like it) / 1 (One yay) = (wait for it) fail
      policy (A dot can go here) . (like that) expired
      (this surprised me) = (as I wasn't expecting it) 1362471462
```

Example 7: A Very Comment-Heavy but Perfectly Legal Example

Appendix D. Operational Considerations about Message Authentication

This protocol is predicated on the idea that authentication (and presumably in the future, reputation) work is typically done by border MTAs rather than MUAs or intermediate MTAs; the latter merely make use of the results determined by the former. Certainly this is not mandatory for participation in electronic mail or message authentication, but this protocol and its deployment to date are based on that model. The assumption satisfies several common ADMD requirements:

1. Service operators prefer to resolve the handling of problem messages as close to the border of the ADMD as possible. This enables, for example, rejection of messages at the SMTP level rather than generating a DSN internally. Thus, doing any of the authentication or reputation work exclusively at the MUA or intermediate MTA renders this desire unattainable.

2. Border MTAs are more likely to have direct access to external sources of authentication or reputation information since modern MUAs are more likely to be heavily firewalled. Thus, some MUAs might not even be able to complete the task of performing authentication or reputation evaluations without complex proxy configurations or similar burdens.

3. MUAs rely upon the upstream MTAs within their trust boundaries to make correct (as much as is possible) evaluations about the message's envelope, header, and content. Thus, MUAs don't need to know how to do the work that upstream MTAs do; they only need the results of that work.

4. Evaluations about the quality of a message, from simple token matching (e.g., a list of preferred DNS domains) to cryptanalysis (e.g., public/private key work), are at least a little bit expensive and thus need to be minimized. To that end, performing those tests at the border MTA is far preferred to doing that work at each MUA that handles a message. If an ADMD's environment adheres to common messaging protocols, a reputation query or an authentication check performed by a border MTA would return the same result as the same query performed by an MUA. By contrast, in an environment where the MUA does the work, a message arriving for multiple recipients would thus cause authentication or reputation evaluation to be done more than once for the same message (i.e., at each MUA), causing needless amplification of resource use and creating a possible denial-of-service attack vector.

5. Minimizing change is good. As new authentication and reputation methods emerge, the list of methods supported by this header field would presumably be extended. If MUAs simply consume the contents of this header field rather than actually attempt to do authentication and/or reputation work, then MUAs only need to learn to parse this header field once; emergence of new methods requires only a configuration change at the MUAs and software changes at the MTAs (which are presumably fewer in number). When choosing to implement these functions in MTAs vs. MUAs, the issues of individual flexibility, infrastructure inertia, and scale of effort must be considered. It is typically easier to change a single MUA than an MTA because the modification affects fewer users and can be pursued with less care. However, changing many MUAs is more effort than changing a smaller number of MTAs.

6. For decisions affecting message delivery and display, assessment based on authentication and reputation is best performed close to the time of message transit, as a message makes its journey toward a user's inbox, not afterwards. DKIM keys and IP address reputations, etc., can change over time or even become invalid, and users can take a long time to read a message once delivered. The value of this work thus degrades, perhaps quickly, once the delivery process has completed. This seriously diminishes the value of this work when done other than at MTAs.

Many operational choices are possible within an ADMD, including the venue for performing authentication and/or reputation assessment. The current specification does not dictate any of those choices. Rather, it facilitates those cases in which information produced by one stage of analysis needs to be transported with the message to the next stage.

Appendix E. Change History

E.1. RFC7001 to -00

- Remove "Changes since RFC5451" section; add this "Change History" section.
- Restore XML to previous format. (No visible changes).
- Reset "Acknowledgments".
- Add "To-Do" section.

E.2. -00 to -01

- Apply RFC7410.
- Update all the RFC4408 references to RFC7208.
- Add section explaining "property" values. (Errata #4201)
- Remove "To-Do" section.

E.3. -01 to -02

- Consolidate new sections.

E.4. -02 to -03

- Move the DKIM exception text down to where the DKIM results are defined, and add a forward reference to them.
- More detail about registry creation and previous augmentations.
- Add text explaining each of the method-ptype-property tuples registered by this document.
- Change the meaning of the "Defined" column of the methods registry to be the place where each entry was created and described; it is expected that this will then refer to the method's defining document. Provide IANA with corresponding update instructions.
- Add references: [DMARC], [PRA], [RFC6577], [RRVS], [SMIME-REG].

E.5. -03 to -04

- Add specific update instructions for the "dkim"/"header"/"b" entry in IANA Considerations.
- Add description of values that can be extracted from SMTP AUTH sessions and an example.
- Much more complete descriptions of reporting DomainKeys results.
- Minor editorial adjustments.
- Fix up "smime" entries.
- Update current definitions for the Email Authentication Property Types registry to point to this document.

- Rework the Email Authentication Result Names registry.
- Add more detail about Sender ID.
- Mark all ADSP and DomainKeys entries as deprecated since their defining documents are as well.
- Add references: [RFC6008].

E.6. -04 to -05

- Fix typos.
- Rework some text around ignoring unknown ptypes.
- Completely describe the ptypes registry.
- EHLO is mapped to HELO for SPF.
- RFC7208 uses all-lowercase result strings now, so adjust prose accordingly.
- Move the RFC6008 reference up to where the DKIM reporting is described.

Author's Address

Murray S. Kucherawy
270 Upland Drive
San Francisco, CA 94127
US

EMail: superuser@gmail.com