Network Working Group
Internet Draft
<draft-kucherawy-sender-auth-header-14>
Intended status: Standards Track
Expires: September 2008

M.S. Kucherawy
Sendmail, Inc.
March 19, 2008

Message Header Field for Indicating Message Authentication Status

draft-kucherawy-sender-auth-header-14

Status of this Memo

Abstract

This memo defines a new message header field for use with electronic mail messages to indicate the results of message authentication efforts. Mail user agents (MUAs) may use this message header field to relay that information in a convenient way to users or to make sorting and filtering decisions.

1. Introduction

This memo defines a new message header field for electronic mail messages which presents the results of a message authentication effort in a machine-readable format. The intent is to create a place to collect such data when message authentication mechanisms are in use so that a Mail User Agent (MUA) can provide a recommendation to the user as to the validity of the message's origin and possibly the integrity of its content.

This memo defines both the format of this new header field, and discusses the implications of its presence or absence.

[UPDATE PRIOR TO FINAL VERSION] At the time of publication of this draft, [AUTH], [DKIM], [DOMAINKEYS], [SENDERID] and [SPF] are the published e-mail authentication methods in common use. As various methods emerge, it is necessary to prepare for their appearance and encourage convergence in the area of interfacing these filters to MUAs.

Although [SPF] defined a header field called Received-SPF for this purpose, that header field is specific to the conveyance of SPF and similar results only and thus is insufficient to satisfy the requirements enumerated below.

1.1 Purpose

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

1. Convey to MUAs from filters and Mail Transfer Agents (MTAs) the results of various message authentication checks being applied;
2. Provide a common location for the presentation of this data;
3. Create an extensible framework for reporting new authentication methods as such emerge;
4. Convey the results of message authentication tests to later filtering agents within the same "trust domain", as such agents might apply more or less stringent checks based on message authentication results.

1.2 Requirements

This memo establishes no new requirements on existing protocols or servers, as there is currently no standard place for the described data to be collected or presented.

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

1.3 Definitions

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].

A "border MTA" is an MTA which acts as a gateway between the general Internet and the users within an organizational boundary.

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

An "intermediate MTA" is an MTA which handles messages after a border MTAs and before a delivery MTA.

```
                        +-----+   +-----+   +------------+
                        | MUA |-->| MSA |-->| Border MTA |
                        +-----+   +-----+   +------------+
                                                  |
                                                  |
                                                  V
                                              [Internet]
                                                  |
                                                  |
                                                  V
+-----+   +-----+   +------------------+   +------------+
| MUA |<--| MDA |<--| Intermediate MTA |<--| Border MTA |
+-----+   +-----+   +------------------+   +------------+
```

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" may well apply to the same machine or even the very same agent. It is also possible that message authentication could take place on an intermediate MTA. Although this document doesn't specifically describe such cases, they are not meant to be excluded from this specification.

See [I-D.DRAFT-CROCKER-EMAIL-ARCH] for further discussion on e-mail system architecture.

2. Definition and Format of the Header

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 new header field being defined here is called "Authentication-Results". It is a Structured Header Field as defined in [MAIL] and thus all of the related definitions in that document apply.

This new header field MUST be added at the top of the message as it transits MTAs which do authentication checks so some idea of how far away the checks were done can be inferred. It therefore should be treated as a Trace Header Field as defined in [MAIL] and thus all of the related definitions in that document apply.

The value of the header field (after removing [MAIL] comments) consists of an authentication identifier, an optional version and then a series of "method=result" statements indicating which authentication method(s) were applied and their respective results, and then, for each applied method, an optional "reason" string plus optional "property=value" statements indicating which message properties were evaluated to reach that conclusion.

The header field MAY appear more than once in a single message, or more than one result MAY be represented in a single header field, or a combination of these MAY be applied.

2.2 Formal Definition

Formally, the header field is specified as follows using [ABNF]:

```
 header = "Authentication-Results:" [CFWS] authserv-id
          [ CFWS version ]
          ( [CFWS] ";" [CFWS] "none" / 1*resinfo ) [CFWS] CRLF
        ; the special case of "none" is used to indicate that no
        ; message authentication is performed
```

```
 authserv-id = dot-atom-text
             ; see below for a description of this element;
             ; "dot-atom-text" is defined in section 3.2.4 of
[MAIL]
```

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

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

```
methodspec = [CFWS] method [CFWS] "=" [CFWS] result
           ; indicates which authentication method was evaluated
```

```
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 = token [ [CFWS] "/" [CFWS] 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
```

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

```
ptype = "smtp" / "header" / "body" / "policy"
      ; indicates whether the property being evaluated was
      ; a parameter to an [SMTP] command, or was a value taken
      ; from a message header field, or was some property of
      ; the message body, or some other property evaluated by
      ; the receiving MTA
```

```
 property = token
         ; if "ptype" is "smtp", this indicates which [SMTP]
         ; command provided the value which was evaluated by the
         ; authentication scheme being applied; if "ptype" is
         ; "header", this indicates from which header field the
         ; value being evaluated was extracted; if "ptype" is
         ; "body", this indicates the offset into the body at
which
         ; content of interest was detected; if "ptype" is
"policy"
         ; then this indicates the name of the policy which caused
         ; this header field to be added (see below)
```

```
 pvalue = [CFWS] ( token / addr-spec ) [CFWS]
        ; the value extracted from the message property defined
        ; by the "ptype.property" construction; if the value
        ; identifies an address, then it is an "addr-spec"
        ; as defined in section 3.4 of [MAIL]
```

The "token" and "value" are as defined in section 5.1 of [MIME].

The "token" used in a "result" above is further constrained by the necessity of being enumerated in Section 2.4 or an amendment to it.

See Section 2.3 for a description of the "authserv-id" element.

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

"CFWS" is as defined in section 3.2.3 of [MAIL].

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.

The "ptype" and "property" values used by each authentication method should be defined in the specification for that method (or its amendments).

The "ptype" and "property" are case-insensitive.

A "ptype" value of "policy" indicates a policy decision about the message not specific to a property of the message that could be extracted. For example, if a method would normally report a "ptype.property" of "header.From" and no From: header field was present, the method can use "policy" to indicate that no conclusion about the authenticity of the message could be reached.

If the parsed "ptype.property" construction clearly identifies a mailbox (in particular, smtp.mailfrom, smtp.rcpt, header.from, header.sender), then the "pvalue" MUST be an "addr-spec". Other properties (e.g. smtp.helo) may be evaluated, but the property MUST still be expressed as a "token" for simplified parsing.

2.3 Authentication Identifier Fields

Every Authentication-Results header field has an authentication identifier field ("authserv-id" above). This is similar in syntax to a fully-qualified domain name.

The authentication identifier field provides a unique identifier that refers to the authenticating service within a given mail administrative domain. The uniqueness of the identifier is guaranteed by the mail administrative domain that generates it and must pertain to exactly that one mail administrative domain. This identifier is intended to be machine-readable and not necessarily meaningful to users. MUAs may use this identifier to determine whether or not the data contained in an Authentication-Results header field can be trusted.

For tracing and debugging purposes, the authentication identifier SHOULD be the domain name of the MTA performing the authentication check whose result is being reported.

Examples of valid authentication identifiers are mail.example.org, engineering.example.net and ms1.newyork.example.com.

2.4 Result Values

Each individual authentication method returns one of a set of specific result values. The subsections below define these results for the authentication methods specifically supported by this memo. New methods not specified in this document intended to be supported by the header field defined in this memo MUST include a similar result table either in its defining memo or in a supplementary one.

2.4.1 DKIM and DomainKeys Results

The result values used by [DKIM] and [DOMAINKEYS] are as follows:

none: The message was not signed.

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

fail: The message was signed and the signature(s) is (were) acceptable to the verifier, but it (they) failed the verification test(s).

policy: The message was signed but the signature(s) is (were) not acceptable to the verifier.

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

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

A signature is "acceptable to the verifier" if it passes local policy checks (or there are no specific local policy checks). For example, a verifier might require that the signature(s) on the message be added

by the domain identified in the From: header field of the message, thus making third-party signatures unacceptable.

2.4.2 DKIM ASP Results

The result values are used by [I-D.DRAFT-IETF-DKIM-SSP] as follows:

none: No DKIM policy author signing practises (ASP) record was published.

pass: A DKIM ASP policy was published which indicated the mail should be signed with an author signature, and this message had such a signature that validated.

unknown: No valid author signature was found on the message and either the published ASP policy was "unknown", or no policy was published.

signed: A valid author signature was found on the message and the published ASP policy was "unknown".

fail: No valid author signature was found on the message and the published ASP record indicated an "all" policy.

discard: No valid author signature was found on the message and the published ASP record indicated a "discardable" policy.

nxdomain: Evaluating the ASP for the author's domain indicated that the author's domain does not exist.

temperror: A DKIM policy could not be retrieved due to some error which is likely transient in nature, such as a temporary DNS error. A later attempt may produce a final result.

permerror: A DKIM policy could not be retrieved due to some error which is likely not transient in nature, such as a permanent DNS error. A later attempt is unlikely to produce a final result.

2.4.3 SPF and Sender-ID Results

The result values are used by [SPF] and [SENDERID] as follows:

none: No policy records were published by the sender's domain.

neutral: The sender's domain has asserted that it cannot or does not want to assert whether or not the sending IP address is authorized to send mail on behalf of the sender's domain.

pass: The client is authorized to inject or relay mail on behalf of the sender's domain.

policy: The client is authorized to inject or relay mail on behalf of the sender's domain according to the authentication method's algorithm, but local policy dictates that the result is unacceptable.

hardfail: This client is explicitly not authorized to inject or relay mail on behalf of the sender's domain.

softfail: The sender's domain believes the client was not authorized to inject or relay mail on its behalf but is unwilling to make a strong assertion to that effect.

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

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

The distinction between and interpretation of "none" and "neutral" under these methods is discussed further in [SPF].

The "policy" result would be returned if, for example, [SPF] returned as "pass" result, but a local policy check matches the sending domain to one found in an explicit list of unacceptable domains (e.g. spammers).

2.4.4 iprev Results

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

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

hardfail: The reverse DNS evaluation failed. In particular, the "reverse" and "forward" lookups each produced results but they were not in agreement.

softfail: The reverse DNS evaluation failed. In particular, one or both of the "reverse" and forward lookups returned no data (i.e. a DNS reply code of NODATA).

temperror: The reverse DNS evaluation could not be completed due to some error which is likely transient in nature, such as a temporary DNS error. A later attempt may produce a final result.

permerror: The reverse DNS evaluation could not be completed due to some error which is unrecoverable (e.g. a DNS reply code of NODATA or NXDOMAIN). A later attempt is unlikely to produce a final result.

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

2.4.5 SMTP AUTH Results

The result values are used by the [AUTH] method are as follows:

none: SMTP authentication was not attempted.

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

fail: The SMTP client had 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 which is likely transient in nature, such as a temporary LDAP 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 which is likely not transient in nature, such as a permanent LDAP lookup error. A later attempt is not likely produce a final result.

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

2.4.6 Extension Result Codes

Additional result codes (extension results) may be defined in the future by later revisions or extensions to this specification. Extension results beginning with "x-" will never be defined as standard fields; such names are reserved for experimental use. Result codes not beginning with "x-" MUST be registered with the Internet Assigned Numbers Authority (IANA) and published in an RFC. See Section 7 for further details.

Implementations reporting new result codes MUST use the "x-" prefix until such time as the new method is registered by IANA.

Extension results MUST only be used within trust domains that have explicitly consented to use them. These results 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 or MUA intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an extension result.

2.5 Authentication Methods

This section defines the supported authentication methods and discusses the proper means for applying experimental and other extension methods.

2.5.1 Definition Of Initial Methods

As they are currently existing specifications for message authentication, it is appropriate to define an authentication method identifier for each of [AUTH], [DKIM], [DOMAINKEYS], [SENDERID] and [SPF]. Therefore, the authentication method identifiers "auth", "dkim", "domainkeys", "senderid" and "spf" respectively are hereby defined for MTAs applying those specifications for e-mail message authentication.

Furthermore, method "iprev" is defined in Section 3.

Finally, as its publication is imminent, this document also defines "dkim-asp" per [I-D.DRAFT-IETF-DKIM-SSP].

See Section 7 for details.

2.5.2 Extension Methods

Additional authentication method identifiers (extension methods) may be defined in the future by later revisions or extensions to this specification. Extension methods beginning with "x-" will never be defined as standard fields; such names are reserved for experimental use. Method identifiers not beginning with "x-" MUST be registered with the Internet Assigned Numbers Authority (IANA) and published in an RFC. See Section 7 for further details.

Extension methods may be defined for the following reasons:

1. To allow additional information from new authentication systems to be communicated to MUAs. The names of such identifiers should reflect the name of the method being defined, but should not be needlessly long.
2. To allow the creation of "sub-identifiers" which indicate different levels of authentication and differentiate between their relative strengths, e.g. "auth1-weak" and "auth1-strong".

Implementations of new methods MUST use the "x-" prefix until such time as the new method is registered by IANA.

Authentication method implementors are encouraged to provide adequate information, via [MAIL] comments if necessary, to allow an MUA developer to understand or relay ancilliary 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: mx.example.com;
          foo=pass bar.baz=blob (2 of 3 tests OK)
```

Experimental method identifiers MUST only be used within trust domains 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 or MUA intended for production use SHOULD ignore or delete any Authentication-Results header field that includes an experimental method identifier.

3. The 'iprev' Authentication Method

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

In general, "iprev" is an attempt to verify that a client appears to be valid based on some DNS queries. 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" record query) thus retrieved is done. The response to this second check should result in at least one mapping back to the client's IP address.

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

Section 5.5 of [SPF] contains more detail about this process as well as some discussion of possible denial-of-service attacks. [DNS-IP6] discusses the format of this query for the IPv6 case.

A successful test using this algorithm constitutes a result of "pass" since the domain in which the client's PTR claims it belongs has confirmed that claim. A failure to match constitutes a "hardfail". There is no case in which "softfail" or "neutral" can be returned. The remaining "temperror" and "permerror" cases refer respectively to temporary and permanent DNS query errors.

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 practise of arranging to match the forward and reverse DNS is infrequently observed. Therefore, the actual 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.

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. As such, the order of the presented authentication methods and results MUST NOT be used to determine the importance or strength of any given method over another. Instead, the MUA must interpret the result of each method based on its knowledge of what that method evaluates.

Each "method" MUST refer to an authentication method declared in the IANA registry, or an extension method as defined in Section 2.5.2, and each "result" MUST refer to a result code declared in the IANA registry, or an extension result code as defined in Section 2.4.6. See Section 7 for further information about the registered methods and result codes.

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

An MTA MAY add this header field containing only the authentication identifier portion to indicate explicitly that no message authentication schemes were applied prior to delivery of this message.

4.1 Header Position and Interpretation

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

Furthermore, an MUA SHOULD NOT interpret this header field unless the authentication identifier it bears appears to be one within its own trust domain as configured by the user or administrator.

An MUA MUST ignore any result reported using a "result" not specified in the result code registry, or a "ptype" not listed in the corresponding registry for such values as defined in Section 7. Moreover, an MUA MUST ignore a result indicated for any "method" it does not support.

An MUA SHOULD NOT reveal these results to end users unless the results are accompanied by, at a minimum, some associated reputation data about the message that was authenticated. 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 domain name was intended to mislead. See Section 8.2 for further discussion.

As stated in Section 2.1, this header field SHOULD be treated as though it were a trace header field as defined in section 3.6 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.

Further discussion of this can be found in Section 8 below.

4.2 Local Policy Enforcement

If a site's local policy is to consider a non-recoverable failure result (e.g. "fail" for DKIM, "hardfail" for SPF or "discard" for DKIM-ASP) for any particular authentication method as justification to reject the message completely, the border MTA SHOULD issue an [SMTP] rejection response to the message rather than adding this header with the failure result and allowing it 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 [DSN] to a forged originator.

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 4.2.

Such rejections at the SMTP protocol level are not possible if local policy is enforced at the MUA and not the MTA. Unfortunately, this may be a common scenario.

5. Removing The Header Field

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

Furthermore, a border MTA MAY elect simply to remove all instances of this header field on mail crossing into its trust boundary.

A formal definition of "trust boundary" is deliberately not made here. It is entirely possible that a border MTA for example.com might 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 headers, this may invalidate one or more signature on the message if they included the header field to be removed at the time of signing. This behaviour can be desirable since there's little value in validating the signature on a message with forged headers. 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 bearing a version (express or implied) that it does not support. However, an MTA MUST remove such a header if the [SMTP] connection relaying the message is from not a trusted internal MTA.

6. Conformance and Usage Requirements

An MTA or gateway conforms to this specification if it applies one or more message authentication mechanisms and inserts a header field corresponding to this specification after doing so and prior to delivery (per Section 4) and removes apparently improper headers (per Section 5).

MTAs that are relaying mail rather than delivering it, i.e. are not part of an intended recipient's trust boundary, MAY perform message authentication or even take actions based on the results found, but SHOULD NOT add an "Authentication-Results" header field if relaying (rather than rejecting or discarding at the gateway). Conversely, an MTA doing local delivery and some form of message authentication MUST add this header field prior to delivering the message in order to be compliant. An exception to the former case is described in Section 5.

A minimal implementation that does at least one message authentication check will add the header field defined by this memo prior to invoking local delivery procedures.

7. IANA Considerations

IANA is requested to register a new header field and to create a new table as described below.

7.1 The Authentication-Results: header

Per [IANA-HEADERS], the "Authentication-Results:" header field is added to the IANA Permanent Message Header Field Registry. The following is the registration template:

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

7.2 Email Authentication Method Name Registry

Names of message authentication methods supported by this specification must be registered with IANA, with the exception of experimental names as described in Section 2.5.2.

New entries are assigned only for values that have been documented in a published RFC that has IETF Consensus, per [IANA-CONSIDERATIONS]. Each method must register a name, the specification that defines it, one or more "ptype" values appropriate for use with that method, and which "property" value(s) should be reported by that method.

The initial set of entries in this registry is as follows:

```
+------------+----------+--------+----------------
+--------------------+
|   Method   | Defined  | ptype  | property       | value
   |
+------------+----------+--------+----------------
+--------------------+
|    auth    | RFC4954  | smtp   | auth           | AUTH parameter
 of  |
|            |          |        |                | the SMTP MAIL
   |
|            |          |        |                | command
   |
+------------+----------+--------+----------------
+--------------------+
|    dkim    | RFC4871  | header | d              | value of
   |
|            |          |        |                | signature "d"
 tag  |
|            |          |        +----------------
+--------------------+
|            |          |        | i              | value of
   |
|            |          |        |                | signature "i"
 tag  |
+------------+----------+--------+----------------
+--------------------+
|  dkim-asp  |  [TBD]   | header | from           | value of From
   |
|            |          |        |                | header field
   |
|            |          |        |                | w/comments
 removed |
+------------+----------+--------+----------------
+--------------------+
| domainkeys | RFC4870  | header | from           | value of From
   |
|            |          |        |                | header field
   |
|            |          |        |                | w/comments
 removed |
|            |          |        +----------------
+--------------------+
|            |          |        | sender         | value of Sender
   |
|            |          |        |                | header field
   |
|            |          |        |                | w/comments
 removed |
+------------+----------+--------+----------------
+--------------------+
|    iprev   | this     | policy | iprev          | client IP
 address  |
|            | document |        |                |
   |
```

7.3 Email Authentication Result Name Registry

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 2.4.6.

New entries are assigned only for result codes that have been documented in a published RFC that has IETF Consensus, per [IANA-CONSIDERATIONS]. Each code must register a name, the document which establishes the registration, the authentication method(s) which uses it, and either a definition of the semantics of its use or a reference to the place where those semantics are defined.

The initial set of entries in this registry is as follows:

```
+-----------+----------+----------------
+------------------------------+
|   Code    | Defined  | Auth Method(s) | Meaning
   |
+-----------+----------+----------------
+------------------------------+
| none      | this     | dkim           | section 2.4.1
   |
|           | document | domainkeys     |
   |
|           |          +----------------
+------------------------------+
|           |          | dkim-asp       | section 2.4.2
   |
|           |          +----------------
+------------------------------+
|           |          | spf            | section 2.4.3
   |
|           |          | sender-id      |
   |
|           |          +----------------
+------------------------------+
|           |          | auth           | section 2.4.5
   |
+-----------+----------+----------------
+------------------------------+
| pass      | this     | dkim           | section 2.4.1
   |
|           | document | domainkeys     |
   |
|           |          +----------------
+------------------------------+
|           |          | dkim-asp       | section 2.4.2
   |
|           |          +----------------
+------------------------------+
|           |          | spf            | section 2.4.3
   |
|           |          | sender-id      |
   |
|           |          +----------------
+------------------------------+
|           |          | iprev          | section 2.4.4
   |
|           |          +----------------
+------------------------------+
|           |          | auth           | section 2.4.5
   |
+-----------+----------+----------------
+------------------------------+
| fail      | this     | dkim           | section 2.4.1
   |
|           | document | domainkeys     |
   |
```

8. Security Considerations

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

8.1 Forged Headers

An MUA that accesses a mailbox whose mail is 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 destination MX for a receiving domain as the hostname token in the value of the header, 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.

It is for this reason an MUA should not 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 alternate solutions to this problem are nascent. Possibly the simplest is a digital signature on the header field that can be verified by a posted public key. 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. In either case, a mechanism needs to exist 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.

8.2 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 removal discussed above.

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

8.3 Other Protocols

Mitigation of the forged header 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 which is used 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.

8.4 Header Position

Despite the requirements of [MAIL], header fields can sometimes be reordered enroute by intermediate MTAs. The goal of requiring header field addition only at the top of a message is an acknowledgement 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.

8.5 Reverse IP Query Denial-Of-Service Attacks

Section 5.5 of [SPF] describes a DNS-based denial-of-service attack for verifiers that attempt to DNS-based identity verification of arriving client connections. A verifier wishing to do this check and report this information SHOULD 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 memo SHOULD be aware of the algorithm used by the verifier reporting the result and thus be aware of its limitations.

8.6 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 which did not send the messages being rejected.

8.7 Internal MTA Lists

Section 5 describes a procedure for scrubbing headers which 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 a domain to attack.

8.8 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 (and later amendments to it) is also a security consideration here.

8.9 Intentionally Malformed Header Fields

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

8.10 Compromised Internal Hosts

An internal MUA or MTA which has been compromised could generate mail with a forged From: header and a forged Authentication-Results: header which endorses it. Although it is clearly a larger concern to have compromised internal machines than it is to prove the value of this header, 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.

9. References

9.1 Normative References

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

[IANA-HEADERS] Klyne, G., Nottingham, M., and J. Mogul, " Registration Procedures for Message Header Fields ", RFC 3864, September 2004.

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

[MAIL] Resnick, P., " Internet Message Format ", RFC 2822, April 2001.

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

9.2 Informative References

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

[DKIM] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and M. Thomas, " DomainKeys Identified Mail (DKIM) Signatures ", RFC 4871, May 2007.

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

[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.

[I-D.DRAFT-CROCKER-EMAIL-ARCH] Crocker, D., " Internet Mail Architecture ", I-D draft-crocker-email-arch, May 2007.

[I-D.DRAFT-IETF-DKIM-SSP] Allman, E., Delany, M., and J. Fenton, " DKIM Author Signing Practices ", I-D draft-ietf-dkim-ssp, January 2008.

[IANA-CONSIDERATIONS] Alvestrand, H. and T. Narten, " Guidelines for Writing an IANA Considerations Section in RFCs ", RFC 2434, October 1998.

[IMAP] Crispin, M., " Internet Message Access Protocol - Version 4rev1 ", RFC 3501, March 2003.

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

[SMTP] Klensin, J., " Simple Mail Transfer Protocol ", RFC 2821, April 2001.

[SPF] Wong, M. and W. Schlitt, " Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1 ", RFC 4408, April 2006.

Author's Address

Murray S. Kucherawy
Sendmail, Inc.
6475 Christie Ave., Suite 350
Emeryville, CA 94608
US
Phone: +1 510 594 5400
EMail: msk+ietf@sendmail.com

A. Acknowledgements

The author wishes to acknowledge the following for their review and constructive criticism of this proposal: Eric Allman, Dave Crocker, Mark Delany, Frank Ellermann, Jim Fenton, Philip Guenther, Tony Hansen, Paul Hoffman, Eliot Lear, John Levine, Miles Libbey, Charles Lindsey, S. Moonesamy, Juan Altmayer Pizzorno, Michael Thomas.

B. Legacy MUAs

Implementors of this proposal 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 adaptation, 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.

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.sendmail.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@sendmail.com
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: mail-router.example.com; none
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by server.sendmail.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@sendmail.com
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, indicating that the delivering MTA (which is named in the value of the header field) conforms to this specification. The presence of "none" (and the absence of any method and result tokens) indicates that no message authentication was done.

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: mail-router.example.com;
          spf=pass smtp.mailfrom=sender@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 (which is identified in the value of the header field) conforms to this specification. Furthermore, the message

was authenticated by that MTA via the method specified in [SPF]. 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: mail-router.example.com;
          auth=pass (cram-md5) smtp.auth=sender@example.com;
          spf=pass smtp.mailfrom=sender@example.com
Authentication-Results: mail-router.example.com;
          sender-id=pass header.from=sender@example.com
Received: from mail-router.example.com
              (mail-router.example.com [192.0.2.1])
          by dialup-1-2-3-4.example.net (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.net
From: sender@example.com
Message-Id: <12345.abc@example.com>
Subject: here's a sample

Hello!  Goodbye!
```

Example 4: Headers reporting results from one MTA

The "Authentication-Results" header field is present, indicating the delivering MTA (which is identified in the value of the header field) conforms to this specification. Furthermore, the sender authenticated herself/himself to the MTA via a method specified in [AUTH], and both [SPF] and [SENDERID] 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 SenderID 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: auth-checker.example.com;
              sender-id=hardfail header.from=sender@example.com;
              dkim=pass (good signature)
 header.i=sender@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
    Authentication-Results: mail-router.example.com;
              auth=pass (cram-md5) smtp.auth=sender@example.com;
              spf=hardfail smtp.mailfrom=sender@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
    DKIM-Signature:  v=1; a=rsa-sha256; s=gatsby; d=example.com;
              t=1188964191; c=simple/simple;
              h=From:Date:To:Message-Id:Subject;
              bh=sEuZGD/pSr7ANysbY3jtdaQ3Xv9xPQtS0m70;
              b=EToRSuvUfQVP3Bkz ... rTB0t0gYnBVCM=
    From: sender@example.com
    Date: Fri, Feb 15 2002 16:54:30 -0800
    To: receiver@sendmail.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. It is present twice because two different MTAs in the chain of delivery did authentication tests. The first, "mail-router.example.com" reports that [AUTH] and [SPF] were both used, and [AUTH] passed but [SPF] failed. In the [AUTH] case, additional data is provided in the comment field, which the MUA can choose to render if desired.

The second MTA, identifying itself as "auth-checker.example.com", reports that it did a [SENDERID] 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.

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 [AUTH]. The [SPF] and [SENDERID] 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 domain:

```
     Authentication-Results: chicago.example.com;
           dkim=pass (good signature) header.i=@mail-
router.example.net;
           dkim=fail (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: mail-router.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 domains

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 which 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. 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 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 "hardfail". However, the first signature included the Authentication-Results: header added at mail-router.example.net which validated the second signature. Thus, indirectly, it can be determined that the authentication claimed by both signatures are indeed valid.

D. Public Discussion

[REMOVE BEFORE PUBLICATION]

Public discussion of this proposed specification is handled via the mail-vet-discuss@mipassoc.org mailing list. The list is open. Access to subscription forms and to list archives can be found at http://mipassoc.org/mailman/listinfo/mail-vet-discuss.

Full Copyright Statement

Intellectual Property

[1] mailto:ietf-ipr@ietf.org