Workgroup: lamps
Internet-Draft:
draft-dkg-lamps-expect-signed-mail-01
Published: 1 October 2023
Intended Status: Informational
Expires: 3 April 2024
Authors: D. K. Gillmor
American Civil Liberties Union
A. Wussler
Proton AG

Expect Signed Mail

Abstract

This draft proposes a mechanism for e-mail users to indicate to their peers that the peer should expect all future e-mail from them to be cryptographically signed.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://dkg.gitlab.io/expect-signed-mail/. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-dkg-lamps-expect-signed-mail/.

Discussion of this document takes place on the LAMPS Working Group mailing list (mailto:spasm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/spasm/. Subscribe at https://www.ietf.org/mailman/listinfo/spasm/.

Source for this draft and an issue tracker can be found at https://gitlab.com/dkg/expect-signed-mail.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://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 3 April 2024.

Copyright Notice

Table of Contents

1. Introduction

E-mail signature validation is hard. When an e-mail signature is absent (or invalid), a reasonable mail user agent will hide their cryptographic authenticity security indicator for the message. But an absent security indicator is hard to notice.

Some e-mail users create end-to-end signatures of all of their e-mails. The peers of those users may want to display a more significant warning message when a signature is absent or invalid.

This draft proposes a mechanism whereby an e-mail author can indicate to a peer that they should expect all their future messages to be cryptographically signed.

1.1. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

The terms Message Submission Agent (MSA), Message Transfer Agent (MTA), and Message User Agent (MUA) are to be interpreted as defined in [RFC6409].

2. Signalling Mechanism

A sender that intends to signal their intention to activate the Expected-Signed mechanism **MUST** add an "Expected-Signed" header to the outgoing email, specifying an expiration date in the value of the header.

The authenticity of the header **MUST** be validated. The header **SHOULD** be transmitted in the protected headers (see [I-D.ietf-lamps-header-protection-16]), and a recipient MUA **SHOULD** deem it valid if the signing key is already trusted. Alternatively, the recipient's MUA:

* **SHOULD** validate the DKIM headers [RFC6376] and **SHOULD** require them to be in the protected headers.

* **SHOULD** validate the SPF records [RFC7208] to verify it is coming from an authorized source and **SHOULD** require the message to be delivered over TLS before deeming it valid.

FIXME: Do we want to support them all?

A recipient MUA that receives a valid "Expected-Signed" header **SHOULD** safely store this information and its expiration date indexed per email address.

2.1. Expect-Signed syntax

The ABNF (Augmented Backus-Naur Form) syntax for the Expect-Signed header field is given below, as described by [RFC2616].

```
Expect-Signed = "Expect-Signed" ":"
                [ directive ]  *( ";" [ directive ] )

directive                 = directive-name [ "=" directive-value ]
directive-name            = token
directive-value           = token | quoted-string
```

Note that:

- The Expect-Signed header field name and directive names are not case sensitive.
- All directives **MUST** appear only once in an Expect-Signed header field. Directives are either optional or required, as stipulated in their definitions.
- The order of appearance of directives is not significant.
- MUAs **MUST** ignore any Expect-Signed header field containing directives, or other header field value data, that does not conform to the syntax defined in this specification.
- If an Expect-Signed header field contains directive(s) not recognized by the MUA, the MUA **MUST** ignore the unrecognised directives. If the Expect-Signed header field otherwise satisfies the above requirements, the MUA **MUST** process the recognized directives.

2.1.1. The expiry directive

The expiry directive defines an expiration date for the expectation of a signature. The policy **MUST** be enforced until the expiry date is in the past.

This value is expressed by the sender with a fixed-length and single-zone subset of the date and time specification used by the Internet Message Format [RFC5322].

```
expiry-name  = "expiry"
expiry-value = IMF-fixdate
```

This value **SHOULD** be safely stored by the recipient's MUA, and **SHOULD** be updated when a valid newer one is received.

NOTE: any date in the past will effectively cease the policy enforcement.

2.2. Header Examples

The Expect-Signed header stipulates an Expect-Signed policy to remain in effect until the specified date:

```
Expect-Signed: expiry="Sun, 20 Oct 2019 14:19:20 GMT";
```

NOTE: the expiry-value must be quoted since it is not a token.

FIXME: should the expiry use a more sane date format, like ISO-8601?

3. Validating Policy

All e-mails coming from addresses that are stored and valid as Expect-Signed **MUST** be validated. To validate a message's signature the recipient's client **MUST** follow OpenPGP's specification Section 5.2.4 of [RFC4880].

FIXME: This is not only OpenPGP. Should include a reference to PGP/MIME [RFC3156], S/MIME [RFC8551], and [I-D.ietf-lamps-e2e-mail-guidance-12].

The sender's key can be retrieved from any trusted storage or repository, and if none is found it **SHOULD** be indicated in the feedback. This mechanism will allow the sender to automatically reply with their key.

4. On Policy Violation

In the scenario where a sender has enabled the Expect-Signature it is expected that all the outgoing messages are provided with a a valid signature, and both the sender and recipient should be notified when a signature is missing or invalid.

An MSA, MTA, or MUA **SHOULD NOT** prevent a message from being received due to a missing signature. The MUA **MUST** warn the user if an expected signature is missing or invalid, and **SHOULD** provide feedback as specified in the following sections.

4.1. Warn

The recipient's MUA **MUST** warn the user that the signature is missing or invalid on every instance where the signature is expected but not verified. The two cases **SHOULD** be treated as equal, because a missing signature is not any more suspicious than a broken signature: a malicious attacker that alters a message can easily remove the signature too.

4.2. Explicit Feedback

FIXME: describe TLSRPT-style feedback

The MUA **MAY** avoid automatic explicit feedback, as it introduces a vector for attackers to know if an email is reachable or if a user read the message.

4.2.1. Sender behaviour

FIXME: Define what to do with the explicit feedback. FIXME: Key points: Should it be machine or human readable? Localization?

4.3. Inline Feedback

When replying to a message whose expectation of signature is failed the MUA **SHOULD** introduce an Expect-Signed-Failure header to signal to the original sender that the message signature was missing or invalid.

The syntax of the Expect-Signed-Failure field, described by the ABNF [RFC2616] is as follows:

```
"Expect-Signed-Failure" ":" failure-reason CRLF
```

Note that the Expect-Signed-Failure header field name and failure-reason value are not case sensitive.

4.3.1. Failure-reason

There are four categories of failure for signature verification:

* **no-signature**: The message is not provided with a signature packet or part.

* **signature-invalid**: The message is provided with a not matching signature, and the key ID matches the signature key ID, i.e., the content is different from the signed data.

* **signature-not-verified**: The message is provided with a signature, but the MUA is unable to verify it because it does not have or can not retrieve the matching key.

* **signature-expired**: The message has a corresponding signature, that is invalid because either the key or signature expiration are in the past. In this case an MUA **SHOULD NOT** give any feedback to the sender.

The failure-reason value can then assume the following values:

```
failure-reason =  (   "no-signature"
                    / "signature-invalid"
                    / "signature-not-verified"
                    / "signature-expired"
                  )
```

4.3.2. Sender behaviour

The sender's MUA receiving an inline feedback **MUST** display a warning to the user if the reason is "no-signature" or "signature-invalid", and **MAY** display a warning if the reason is "signature-not-verified" or "signature-expired".

The purpose of this warning is to warn the sender that there might be some misconfigured option in the mail client, that result in messages being unsigned or malformed, or that he is victim of an impersonation.

Furthermore, when receiving an inline feedback with reason "signature-not-verified" the sender's MUA **MAY** automatically attach a copy of their public key to a successive reply.

5. Common UX for Absent and Invalid Signatures

FIXME: explain why receiving MUAs should display the same thing when a signature is missing as when it is absent.

6. Related Work

This draft is inspired by (and similar to) HSTS, MTA-STS, TLSRPT, etc.

FIXME: include references

7. IANA Considerations

IANA might need to register the e-mail headers Expect-Signed and Expect-Signed-Failure.

8. Normative References

[I-D.ietf-lamps-e2e-mail-guidance-12] Gillmor, D. K., Hoeneisen, B., and A. Melnikov, "Guidance on End-to-End E-mail Security", Work in Progress, Internet-Draft, draft-ietf-lamps-e2e-mail-guidance-12, 13 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-e2e-mail-guidance-12>.

[I-D.ietf-lamps-header-protection-16] Gillmor, D. K., Hoeneisen, B., and A. Melnikov, "Header Protection for Cryptographically Protected E-mail", Work in Progress, Internet-Draft, draft-ietf-lamps-header-protection-16, 13 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-header-protection-16>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <https://www.rfc-editor.org/rfc/rfc2616>.

[RFC3156] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, DOI 10.17487/RFC3156, August 2001, <https://www.rfc-editor.org/rfc/rfc3156>.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <https://www.rfc-editor.org/rfc/rfc4880>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/rfc/rfc5322>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/rfc/rfc6376>.

[RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011, <https://www.rfc-editor.org/rfc/rfc6409>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", RFC 7208, DOI 10.17487/RFC7208, April 2014, <https://www.rfc-editor.org/rfc/rfc7208>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/rfc/rfc8551>.

Appendix A. Acknowledgements

FIXME

Appendix B. Mapping the Solution Space

[RFC Editor: please remove this section before publication]

The range of possible solutions in this problem space is potentially quite wide.

The draft attempts to make some decisions, but they can be revisted. This appendix tries to document some distinct axes along which the problem can be resolved.

The completed draft should provide a clear choice along each axis, or a mechanism for some active participant in the protocol to select a choice.

B.1. Signal Location

Where should the signal be emitted? Is it a per-message signal? Is it in the sender's certificate? Is it in the DNS?

B.2. Signal Scope

What is the scope of the signal? For example, does it cover a particular e-mail address in the "From" field? Could it cover all e-mail addresses in a given domain? Or does it only cover a specific pair-wise promise (e.g., "alice@example.com will sign all mail that is only addressed to bob@example.net")? Does it apply to all mail, or could it be limited to end-to-end-encrypted mail?

B.3. Intervening Mail User Agents

How does this signal interact with messages that arrive through intervening MUAs, like mailing lists or bug-tracking systems that may (deliberately or not) break signatures while forwarding mail?

B.4. How to Signal?

How does the sender opt into emitting this signal such that all of their MUAs are aware of it? Clearly, you'd want each MUA controlled by the sender to know that the signal has been published so that they can all adjust their signing policy.

B.5. Retraction

How does the sender change their mind once such a signal has been emitted? Does the signal expire? What happens to messages during the period where the signal is in some sort of indeterminate state?

B.6. Consequences

What should the available consequences be when an unsigned (or broken-signature) message arrives from a sender who has emitted that signal? Should the recieving MUA show the message with a warning? Should the receiving MUA report the failure to the sender (e.g., like MTA-STS)? Should they reject the message entirely? How much control should the signaller be able to exercise?

B.7. What Kind of Cryptographic Signature?

Does the signal commit the sender to any particular kind of cryptographic signature? For example, PGP/MIME, or S/MIME? To signatures verifiable by any particular certificate?

Appendix C. Document Considerations

[RFC Editor: please remove this section before publication]

C.1. Document History

C.1.1. Changes from draft-dkg-lamps-expect-signed-mail-00 to draft-dkg-lamps-expect-signed-mail-01

*add Aron Wussler to authors

Authors' Addresses

Daniel Kahn Gillmor
American Civil Liberties Union

Email: dkg@fifthhorseman.net

Aron Wussler
Proton AG
Switzerland

Email: aron@wussler.it