Workgroup: Independent Stream
Internet-Draft:
draft-bradshaw-envelope-validation-extension-
dkim-00
Published: 4 October 2022
Intended Status: Experimental
Expires: 7 April 2023
Authors: M. Bradshaw
Fastmail, PTY LTD

DKIM Envelope Validation Extension (eve)

Abstract

DKIM as defined in RFC6376 is an IETF standard of cryptographically signing email with a domain key. DKIM is widely used to build a reputation based on the signing domain and assign that reputation to message filtering. Section 8.6 defines a vulnerability called DKIM replay, in which a single message can be replayed to a large group of unrelated recipients, thereby hijacking the reputation of the original sender. This proposal defines a method of declaring the original envelope sender and recipient(s) within a DKIM signature such that compliant DKIM validators can detect a DKIM signature which may have been replayed and modify their use of domain reputation accordingly. This technique remains fully backwards compatible with DKIM validators which do not support the new methods, while allowing compliant forwarders to declare their ingress authentication state in Authentication Results headers for consumption by subsequent validators.

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 7 April 2023.

Copyright Notice

Table of Contents

1. Terminology and 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 RFC2119 [RFC2119].

2. Goals

Allow a DKIM ([RFC6376]) validator to detect DKIM replay by validating a DKIM signature against the envelope sender and recipient addresses of the SMTP transaction.

*Allow a DKIM signature to assert the envelope addresses for which it is valid.

*Does not require large scale modification of DKIM.

*Does not break existing DKIM mail flows.

*Does not reveal original envelope details to third party viewers when mail is forwarded via an intermediary.

*Does not require non participating intermediaries to modify their handling.

*Builds on existing DKIM mechanism to provide a method of digitally signing the envelope details.

*Does not require modifications to the SMTP protocol.

*Allows messages to be signed by multiple DKIM domains, and assigns envelope assertion only to the domains who signed that portion of the mail flow.

*Allows receivers to determine that a given DKIM signature was added before intermediary redirection of mail, and exclude that DKIM domain from reputational misuse.

3. Introduction

Allow a DKIM signer to assert the envelope addresses their signature should be considered valid for, without breaking DKIM signatures when sending mail via intermediaries.

DKIM signers add and sign additional headers which contain a hashed list of the valid envelope addresses. This is cross linked back to the DKIM signer(s) by including a unique identifier in the hash generation, and to the message by including the "unique" message-id of that message.

DKIM signers add and sign the eve headers, including the eve headers added by previous hops. A specially formatted eve header is included to indicate the trust boundary.

Eve aware validators check the envelope addresses of received mail against all DKIM signatures present, and add the eve status of each signature in the authentication results entry for that signature. This is then used to determine which signatures have been verified against the envelope details and which have not, and are therefore vulnerable to replay, or may have been forwarded by an intermediary. The receiver can then apply domain reputation based on a validated path, and can avoid applying domain reputation of a domain which is the victim of a DKIM replay attack.

4. Generating EVE headers.

DKIM eve headers are added in sets, a set of eve headers is referred to as an eve assertion set. An eve assertion set consists of exactly 1 eve assertion set marker header, and 1 or more eve assertion hash headers. Both of these headers are added as DKIM-EVE headers, this allows us to use the header ordering present in DKIM to treat each set as a single signed unit.

DKIM-EVE headers MUST be signed by 1 or more DKIM signatures, they MUST NOT be oversigned, this allows for intermediaries to continue the EVE chain without breaking the original DKIM signature. The breaks provided by the EVE assertion header prevents the new signer asserting validation of the original envelope details

The eve assertion set marker header is used to define the boundary of eve assertion sets claimed by a given DKIM signature.

The EVE assertion set marker consists of a version number, a semicolon, with the remainder of the header being defined by the version declared. For version 1 the remainder of the header contains the unique eve identifier for this eve assertion set. This is a unique alphanumeric string.

An eve hash is a secure hash of the envelope addresses. A version 1 eve hash is generated using the following algorithm

First, create the following string(s)

<unique eve id> ; <message-id> ; <envelope from address> ; <envelope to address>

A message should have only a single envelope from address, and 1 unique eve hash must be created for each envelope to address. Just in time signers may choose to add only a single eve hash, whereas before queue signers would add 1 eve hash for each intended recipient. Privacy conscious senders who wish to mask the number of recipients a message is intended for may add additional eve-headers, however this may not be recommended.

Generate the SHA256 hash of the eve string, this is the eve hash. Add an eve hash for each set of envelope sender/recipient tuples we are claiming validation for. These may be added in a single DKIM-EVE header enclosed in <>, or one per header, again enclosed in <>

DKIM-EVE headers are added as trace headers, starting with the assertion header (at the bottom), followed by 1 or more data headers (above the assertion header). Headers are added in this way to maintain the DKIM validity of previous signatures.

5. Parsing and validating EVE headers.

For each DKIM signature in the message.

Validate DKIM as usual. To participate in EVE the DKIM signature MUST pass.

Check the headers signed by this signature, if they include at least 1 DKIM-EVE header then this signature is making an EVE assertion.

Extract (bottom up) the DKIM-EVE headers from the message, taking the signed last x, these are the headers which were signed by this DKIM signature, which includes both the EVE assertion set added by this hop, and all previous EVE assertion sets.

Starting at the top of this set, consider each eve data header stopping when we reach an eve assertion header. This is the set of eve headers asserted by this DKIM signature.

Using the eve version and unique id from the eve assertion header, generate a new eve hash for each envelope set for the current SMTP transaction. ie one hash per envelope to address.

Check the set of eve hashes in the current assertion set. If there is a valid eve hash present for every envelope to address in the SMTP transaction then the eve result is pass. If there are no matches, or matches for only a subset of addresses, then the eve result for this signature is FAIL. This result is recorded in the Authentication-Results ([RFC8601]) entry for this DKIM signature using the smtp.eve key.

If this signature makes no eve assertion then the eve result is none. Authenticators may choose to record this in the Authentication-Results DKIM results as smtp.eve=none, or may choose to omit this entirely.

if a unique id is repeated within a group of eve assertion sets then a receiver MAY consider the most recently duplicated set invalid.

6. Example

Message originates at sender.com, is sent by user@sender.com to bob@example.com and alice@intermediate.com

Intermediate.com forwards this message on to sue@recipient.com using the envelope sender address bounces@intermediate.com

recipient.com is able to determine that the DKIM signature for sender.com is valid, but does not assert the envelope addresses as received, an smtp.eve=fail status is applied to this signature.

recipient.com is able to validate that the DKIM signature for intermediate.com is valid, and does assert the envelope addresses received, an smtp.eve=pass status is applied to this signature.

The reputation engine for recipient.com considers the eve status of DKIM signatures when applying domain reputation to the message. It notes that the signature for sender.com does not assert validity of the envelope addresses, and checks for other relevant signatures.

The signature for intermediate.com is found with a valid assertion, and the reputation of intermediate.com (both as an intermediate and as a general signer) is considered. The details of how this reputation is applied is beyond the scope of this document, however it is expected that eve may be used to build forwarding reputation for intermediates forwarding mail send my eve compliant senders allowing these intermediaries to build their own reputation as a forwarder. When mail is forwarded by non eve aware intermediaries the reputation of the sender is still protected, as the recpient is aware that the signature is vulnerable to replay.

```
Authentication-Results: receiver.net; dkim=pass header.d=example.com
  header.a=rsa-sha256 header.s=foo smtp.eve=fail; dkim=pass
  header.d=intermediate.com header.a=rsa-sha256 header.s=bar
  smtp.eve=pass
DKIM-Signature: v=1; a=rsa-sha256; d=intermediate.com; s=foo;
  h=dkim-eve,dkim-eve,dkim-eve,dkim-eve, etc
DKIM-EVE:
  <92570938870d5d03c444616ed79e4a2fe782d032bd23dc93aad4c3afe3a8add8>
DKIM-EVE: v=1;iddhwhnqwe
Authentication-Results: intermediate.com; dkim=pass
  header.d=sender.com header.a=rsa-sha256 header.s=foo
  smtp.eve=pass
DKIM-Signature: v=1; a=rsa-sha256; d=sender.com; s=foo;
  h=dkim-eve,dkim-eve, etc
DKIM-EVE:
  <fa72687e3485f66825349e19c49f0ef47505adb557dbe71d9c19d143326a37f2>
  <f11ea3da3b7f96c9debed19039352d0372fd7656e696e71de528af11f34b029e>
DKIM-EVE: v=1;sjpoerwejbn
Message-id: message@sender.com
```

Sender.com used the following strings to generate the eve hashes

<sjpoerwejbn> : <message@sender.com> : <user@sender.com> : <bob@example.com>

<sjpoerwejbn> : <message@sender.com> : <user@sender.com> : <alice@intermediate.com>

Resulting in the following 2 hash strings

fa72687e3485f66825349e19c49f0ef47505adb557dbe71d9c19d143326a37f2

f11ea3da3b7f96c9debed19039352d0372fd7656e696e71de528af11f34b029e

Intermediate.com used the following strings to generate the eve hashes

<iddhwhnqwe> : <message@sender.com> : <bounces@intermediate.com> : <sue@recipient.com>

Resulting in the following hash string

92570938870d5d03c444616ed79e4a2fe782d032bd23dc93aad4c3afe3a8add8

Recipient.com uses the eve-id and envelope set set by Intermediate.com to generate the eve hash used to check. As this matches the data used by Intermediate.com the hash is identical, and eve passes for this hop. # Declination to participate.

A sender who intends to decline to participate may do so be adding a DKIM-EVE: header with a version of 0 and an ID of decline. These messages should be treated as though no EVE headers are present. Intermediaries may add further EVE headers as they process mail, however the intent of the original sender SHOULD be considered by the final recipient system.

An intermediary who intends to decline to participate may do so by adding a similar header, however this MUST NOT override an assertion made by the sender of a message. In this case the receiver SHOULD apply neither the domain reputation of the sender, nor the domain reputation of the intermediary.

DKIM-EVE: v=0;decline

7. Use of EVE results in reputation systems.

eve is intended to tie the DKIM signatures, and so the reputation of DKIM domains to envelope details.

When eve is asserted for a message, a receiver using DKIM domains as a reputation anchor may choose to modify which DKIM domain is used as a reputation anchor when eve indicates that the message has been forwarded, as this may indicate a possible replay attack.

Intermediaries participating in eve may re-sign messages with their own DKIM signatures, and so indicate that they take responsibility for the content they are forwarding. Receivers may choose to use the DKIM domain of the forwarding system as a reputation anchor in this case.

8. Privacy Considerations

Envelope sender and recipient addresses are hashed, reducing the likelihood of those addresses being leaked to downstream participants in the chain. It is expected that many of these addresses are already present in other headers such as From, To, and Received, however this is considered out of scope for this draft.

Each participant sets a unique eve id, thus hashes for like addresses will not produce identical trackable eve hashes over time.

9. Security Considerations

The security considerations of ([RFC6376]) also apply to this extension.

Implementors MAY consider implementing limits to the number of hashes generated for emails sent or received with a large number of envelope addresses in order to avoid resource issues.

Senders MAY choose to explode messages to multiple recipients before DKIM signing, such that they are adding only a single eve hash per message.

10. IANA Considerations

IANA is requested to add the following item to the "Email Authentication Results Method Name Registry"

Method: dkim

Defined: TBC

ptype: smtp

property: eve

value: "pass", "fail", or "none"

11. Notes

11.1. ARC

Receivers using ARC may use verified and trusted ARC-Authentication-Results to determine the EVE authentication status of a message received by a point in their trusted ARC chain. Details of this, and the reputation systems required to support the ARC trust model are beyond the scope of this document.

12. Informative References

[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/info/rfc2119>.

[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/info/rfc6376>.

[RFC8601] Kucherawy, M., "Message Header Field for Indicating Message Authentication Status", RFC 8601, DOI 10.17487/RFC8601, May 2019, <https://www.rfc-editor.org/info/rfc8601>.

Appendix A. Acknowledgments

Author's Address

Marc Bradshaw
Fastmail, PTY LTD

Email: marc@fastmailteam.com