Including Recipients in DKIM Signatures
draft-kucherawy-dkim-rcpts-00

Abstract

The DomainKeys Identified Mail (DKIM) protocol applies a domain-level cryptographic signature to an e-mail message. DKIM only guarantees authenticity of the message content and does not consider the message envelope. This allows for replay attacks by recycling a signed message with an arbitrary new set of recipients.

This document presents a protocol extension that can include the original set of envelope recipients in the signed content, so that an altered set of recipients renders the signature invalid.

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 May 9, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

DKIM [RFC6376] defines a cryptographic signature, placed in a header field consisting of a series of tags and values. The values include signed hashes of some of the header fields and part or all of the body of a message. The signature contains a domain name that is responsible for the signature and thus takes some responsibility for the presence of the message in the email stream.

The signature is valid if the hashes in the signature match the corresponding hashes of the message at validation time, the signature is validated by a public key retrieved from that responsible domain’s DNS, and it is before the expiration time in the signature header field (if set).

There have been recent incidents of a replay attack, where a message of undesirable content (spam, malware, phishing, etc.) is sent by a bad actor to itself through an email service, which dutifully signs it. This message now bears the digital signature of the signing agent’s domain, which means in many cases that the signing agent’s reputation will be weighed by a receiver when assessing the likely safety of the message. The bad actor is then free to re-send that message to any number of other recipients with that same signature, any number of times, by altering the set of recipients on the message
Internet-Draft        DKIM Canonicalized Recipients        November 2016

(the "envelope" in terms of the Simple Mail Transfer Protocol (SMTP) [RFC5321]) and re-sending it. This was anticipated by [RFC6376] Section 8.6.

Obviously a signing agent would be well within its rights and own interests to decline to sign something that looks like it might be unwanted content, but such measures are not fool-proof. What is needed, then, is a way to thwart these sorts of replay attacks.

The proposal presented here is to include in the content presented for signing the original recipient set of the message. A verifier would thereby expect that the original recipient set coupled with the regular signed content in a DKIM signature would be needed to verify the signature, and any change to either the message or the recipient set would result in the signature being invalidated.

2. 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].

Syntax descriptions use Augmented BNF (ABNF) [RFC5234]. The definition of the "FWS" ABNF token is taken from [RFC6376] Section 2.8.

A full description of the email ecosystem can be found in [RFC5598].

3. 'nr' Tag Definition

The following DKIM tag (see [RFC6376] Section 3.5) is introduced:

nr= Number of canonicalized recipients (plain-text unsigned decimal integer; OPTIONAL).

ABNF:

    sig-nr-tag = %x6e.72 [FWS] "=" [FWS] 1*DIGIT

If present, this indicates that the canonicalized form of the header produced according to (see [RFC6376] Section 3.4) is prefixed with this number of SMTP [RFC5321] recipients.

4. Implementation

This section describes implementation of this extension in detail.
4.1.  Signers

When producing the canonicalized header using this tag, the signer takes the following steps:

1. Collect the SMTP recipients that are to be used for sending the message being signed.
2. Sort these using a typical string sort operation.
3. Append a carriage return and line feed to each (ASCII 0x0D and 0x0A respectively).
4. Prepend this construction to the canonicalized header before hashing.
5. Include in the DKIM-Signature header field an "nr" tag as described in Section 3 indicating the number of recipients in the list above.
6. Continue with header canonicalization and hashing as defined in [RFC6376].

4.2.  Verifiers

When analyzing the DKIM-Signature field on an arriving message that includes the tag defined in Section 3, the verifier takes the following steps:

1. Collect the SMTP recipients that were present in the transaction that delivered the message under evaluation.
2. If the size of this list does not match the value of the "nr" tag defined in Section 3, the DKIM signature is invalid; stop and report PERMFAIL ([RFC6376] Section 3.9).
3. Sort these using a typical string sort operation.
4. Append a carriage return and line feed to each (ASCII 0x0D and 0x0A respectively).
5. Prepend this construction to the canonicalized header before hashing.
6. Continue with header canonicalization and hashing and signature verification as defined in [RFC6376].
This has the effect of requiring the same recipient set on the message at time of receipt (more precisely, at time of verification) as was there at the time of signing of the message. If that is not the case, different content will be produced for the canonicalized header, ultimately leading to an invalidated signature. This effectively prevents the sort of attack described in Section 1.

5. Compatibility with Current Infrastructure

[RFC6376] Section 3.5 requires verifiers to ignore tags they do not understand. Accordingly, the introduction of this tag by signers should have no negative impact on existing implementations except that any signatures using them are guaranteed to be considered invalid by verifiers that don’t implement this extension, since the signer and verifier will disagree on what the canonicalized header ought to contain.

Further, [RFC6376] Section 6.1 advises against taking any sort of punitive action against a message with an invalid signature, so again, no negative impact is anticipated.

This will impact the validity of DKIM signatures transiting any service that legitimately alters the envelope recipient set. Examples of this are email forwarding services and mailing lists, or any Mail Transfer Agent (MTA) that needs to send a message to different recipients via different connections such as when they exist in different Administrative Management Domains (ADMDs).

Some DKIM verifiers are implemented as modules downstream of the SMTP receiver. As such, they may not get the complete list of recipients that was presented by the sender if, for example, the SMTP receiver rejected one or more of those recipients due to invalidity or a limit on the number of acceptable recipients per message or per session. They may also get a rewritten (e.g., canonicalized) form of certain recipients. This would lead to invalidation of signatures generated via the extension presented here.

Taken together, the issues raised above suggest that the optimal use case for this extension would be messages that have only a single recipient at the time of signing. However, the details of the proposal do not actually proscribe other use cases.

6. IANA Considerations

IANA is requested to register the following in the "DKIM-Signature Tag Specifications" registry:

Type: nr
7. Privacy Considerations

The list of recipients of a message is not typically recorded anywhere in the message content itself and is instead a property of the SMTP "envelope" used to transport it that is discarded on delivery. This results in the ability to, among other things, do a "blind carbon copy" of a message that does not reveal one recipient to the others.

This proposal adds the full recipient set to the content presented for hashing and ultimate transmission of the message. It does not expose that content to receivers, so there is not a direct leak of potentially private information.

However, this proposal obviously leaks the number of recipients the verifier included in the hashed header content. This by itself might be of interest to an attacker.

Moreover, if a signed message using this tag is received by an attacker with some idea of what recipients might have been on the envelope, such an attacker could simply attempt to repeat signature verification using the suspected set of recipients as described in Section 4.2 and, if successful, thereby recover the original recipient set.

8. Security Considerations

Section 8 of [RFC6376] enumerates known security issues with DKIM. In particular, Section 8.6 of [RFC6376] anticipated this attack.

The issues of compatibility discussed in [RFC6376] are unfortunately the ideal. It is possible or even likely that introducing a new DKIM tag that requires verifier participation for success will result in rejection of otherwise legitimate messages, the impact of which depends almost entirely on the sensitivity of the content thus rejected.

Apart from the privacy-specific discussion in Section 7, and the potential impact on current infrastructure discussed in Section 5, no new security issues are introduced here.
9. Implementation Status

The next release of OpenDKIM will implement this proposal. OpenDKIM is in widespread use, including at very large installations, so use and utility of this extension can be easily observed.

10. Change Log

10.1. 00

- Initial version.

11. Normative References


Appendix A. Acknowledgments

Valuable input to this proposal was provided by Michael Adkins, Peter Blair, Dave Crocker, and Alexey Toptygin.

Author’s Address