Workgroup: Network Working Group
Internet-Draft:
draft-levine-dkim-conditional-04
Published: 30 August 2020
Intended Status: Standards Track
Expires: 3 March 2021
Authors: J. Levine
Taughannock Networks

Mandatory Tags for DKIM Signatures

Abstract

The DKIM protocol applies a cryptographic signature to an e-mail message. This specification extends DKIM to allow new signature tags that validators are required to evaluate. The first such tag specifies a second signature that must be present for a signature to be valid.

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 March 2021.

Table of Contents

1. Introduction

DKIM [RFC6376] defines a cryptographic 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. 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.

This specification defines the syntax for new tags in a signature header field that specify additional conditions that must be satisfied for a signature to be valid. The first such condition requires the presence of an additional signature from a specified different domain. It also changes the DKIM version tag to a verification features tag to allow the new semantics of conditional signatures.

2. Definitions

The upper case 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 ABNF "ALPHA", "FWS", "tag-list" and "domain-name" are defined as in [RFC6376].

3. Mandatory DKIM header tags

The current DKIM specification defines a set of header tags, some of which are required to appear in every signature and some of which are optional. It also allows a signer to include private tags that don't conflict with the registered ones. Since verifiers ignore tags that they don't understand, new tags can only provide new information about the message, or enable new verification schemes for signatures that would otherwise be considered invalid.

A Mandatory Tag is a new kind of tag prefixed with an exclamation point. Its syntax is otherwise identical to an ordinary tag.

ABNF:

```
tag-spec  =/  [FWS] "!" tag-name [FWS] "=" [FWS] tag-value [FWS]
```

3.1. Signature verification features

The v= tag defined in section [RFC6376] section 3.5 is renamed to the Verification Features tag. Its value is a comma-separated sequence of alphanumeric feature names.

ABNF:

```
sig-v-tag = %x76 [FWS] "=" [FWS] 1*(ALPHA / DIGIT)
                              0*(, 1*(ALPHA / DIGIT))
```

Feature name "1" includes all of the features described in [RFC6376]. Feature name "man" includes the Mandatory Tag.

When a signer creates a signature, the v= tag MUST include feature names for all features used in the signature. The v= tag SHOULD NOT include feature names for features not used in the signature. For example, signatures that use only RFC 6376 features have a "v=1" tag.

When a verifier encounters a feature name in the v= tag that it does not support, it MUST return PERMFAIL for that signature.

3.2. Processing mandatory tags

When a verifier encounters a mandatory tag in a signature, it MUST process the tag according to the tag's definition. If the verifier is unable to process the tag the verifier MUST return PERMFAIL for that signature. If there are multiple signatures on a message, the

verifier continues to verify other signatures as usual. It is valid to have a signatures using different features on a single message.

3.3. Forward signature (!fs) tag

The "!fs" mandatory tag means that the signature is only valid if an additional signature is present in the message. The value of the !fs tag is a domain name that is the value of the d= tag of the additional signature. The condition is satisfied if the message includes at least one valid DKIM signature header field with responsible domain (the d= tag) being one specified by the !fs tag.

Chained !fs tags are valid and may be useful in scenarios with multiple levels of forwarders. DKIM verifiers SHOULD handle at least three levels of !fs chaining.

4. Typical application scenarios

A sender that expects a message to be forwarded might put both a conventional DKIM signature and a signature with a !fs tag that refers to the domain name of the expected forwarder, most likely the domain of the recipient in the To header. That signature would be a "weak" signature that covers the From, To, Date, and Message-ID headers but does not cover the Subject header or the message body, so that it would remain valid even if a forwarder made changes that forwarders such as mailing lists often make. Subsequent recipients observe both the forwarder's signature and the signature with the !fs tag that matches the other signature, and use either or both to assess the message.

4.1. Sender use

A small sender that doesn't know which of its mail recipients are likely to be forwarders might put a weak signature on all outgoing mail, in the expectation that few of its users correspondents are likely to be malicious. A sender that had some idea which recipients are forwarders could apply weak signnatures only to mail to those recipients. Or a sender might apply weak signatures to all mail except that sent to recipients with poor reputations.

For the second or third possibilities, the sender might keep its own reputation data, or might query shared reputation services.

4.2. Forwarder use

At the time the message is forwarded, the forwarder uses the conventional signature to assess the message, edits the message, and then signs the outgoing message with its own signature. This process is the same as what forwarders typically do now. The forwarder must not strip the weak signature from the outgoing message.

The forwarder's signature d= domain has to match the one in the original !fs= tag. The simplest way to arrange this is for that domain to be the one in the To header, normally one that the forwarder controls.

[[Possibly allow some flexibility about superdomain or subdomain matching?]]

4.3. Recipient use

A sample set of weak and forwarder signatures might be:

```
DKIM-Signature: v=man,1; a=rsa-sha256; d=example.net; s=abc;
   c=simple; t=1518456670; h=from:to:date:message-id; l=0;
   !fs=lists.example.com; bh=MT34908vdk3l24kedfkpI=;
   b=dzdfAKCdLXdJOc9G2q8LoXSlEniSbav+yuU4zGeeruD00lszZVoG4ZHRNiYzR;
DKIM-Signature: v=1; a=rsa-sha256; s=brisbane; d=lists.example.com;
   h=From : To : Subject : Date : Message-ID;
   bh=2jUSOH9NhtVGCQWNr9BrIAPreKQjO6Sn7XIkfJVOzv8=;
   b=AuUoFEfDxTDkHlLXSZEpZj79LICEps6eda7W3deTVFOk4yAUoqOB=;
```

A message with a weak signature and a forwarder's signature is signed by both, and the recipient would typically use either or both to assess the message. In particular, if the original sender asserts a DMARC policy, the weak signature would be adequate to satisfy that policy.

If a message arrives with signature containing a !fs but no forwarding signature, the recipient would ignore that signature. If the message contains other signatures, the recipient can use them to assess the message.

5. IANA Considerations

IANA is requested to add this entry to the "DKIM-Signature Tag Specifications" registry.

TYPE	REFERENCE	STATUS
!fs	(this document)	active

Table 1: DKIM-Signature Tag Specifications additions

IANA is requested to create the "DKIM-Signature Feature Name" registry, with the following initial contents.

NAME	REFERENCE	STATUS
1	(this document)	active
man	(this document)	active

Table 2: DKIM-Signature Feature
Name contents

6. Security Considerations

DKIM was designed to provide assurances that a message with a valid signature was received in essentially the same form that it was sent. The forwarding signature condition deliberately creates a loophole for messages intended to be forwarded by entities that edit the message. It opens up a variety of obvious replay attacks that may or may not be important depending on both the selection of target domains for messages to be forwarded, and the behavior of forwarders that receive messages with conditional signatures.

A sender can limit the conceptual size of the loophole by being selective about what other domains it allows in its !fs tags, and by using the x= tag to limit the time during which forwarded signatures are valid.

7. Change Log

Please remove this section before publication.

7.1. -03 to -04

Add hints to use To domain as the chain link

7.2. -02 to -03

Add feature names.

Expand usage scenarios.

7.3. -01 to -02

Change tag character from @ to ! per Murray.

Add suggestions about limiting the forwarding loophole.

8. Normative 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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

Author's Address

John Levine
Taughannock Networks
PO Box 727
Trumansburg

Email: standards@taugh.com
URI: http://jl.ly