Why does the ESA handle the DKIM authentication result "permfail" as "hardfail"?

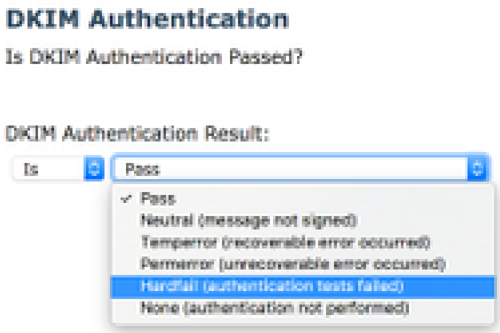
Contents

Introduction

This document describes how the Email Security Appliance (ESA) handles DomainKeys Identified Mail (DKIM) authentication results.

Why does the ESA handle the DKIM authentication result "permfail" as "hardfail"?

The ESA content filter condition DKIM Authentication has several options, as shown in this image:



DKIM Authentication

Is DKIM Authentication Passed?

DKIM Authentication Result:

Is Pass

✓ Pass
Neutral (message not signed)
Temperror (recoverable error occurred)
Permerror (unrecoverable error occurred)
Hardfail (authentication tests failed)
None (authentication not performed)

When the condition DKIM Authentication Result is set to **Hardfail**, permfail messages appear in the mail log file and tracked messages, as shown in this example:

```
Message 815204 DKIM: permfail body hash did not verify [final] (d=sub.example.com s=selector1-sub-com i=@sub.example.com)
```

The ESA considers permfail to be the same as hardfail and includes the result in the Authentication-Results header as dkim=hardfail. The ESA names for DKIM events are different than RFC6376 names. In Authentication-Results headers (and tracked messages), ESA must show proper RFC6376 strings, whereas the content filter uses different event names.

These events are mapped: RFC6376.PERMFAIL == ESA Content Filter Hardfail

Signature and message body hash verification failures constitute the majority of verification failures. Body hash verification errors indicate that the body of the message does not agree with the hash (digest) value in the signature. Signature verification errors indicate that the signature value does not correctly verify the signed header fields (which include the signature itself) on the message.

There are several possible causes for these two errors. The message might have been modified in transit (perhaps by a mailing list or forwarder); the signature or hash values might have been calculated or applied incorrectly by the signer; the wrong public key value might have been published in the Domain Name System (DNS); or the message might have been spoofed by an entity that does not possess the private key that is needed in order to calculate a correct signature.

It is very hard to distinguish these causes by analysis of the message, although the origin IP address can provide some helpful forensics in the case of a spoofed message. However, for privacy reasons we do not have access to the messages themselves, so any such analysis is not possible.

There are messages whose signatures are not verified for other reasons, often because of easily avoided configuration errors in the public key (selector) records that are published in DNS.