Network Working Group
Request for Comments: 5429
Obsoletes: 3028
Updates: 5228
Category: Standards Track

A. Stone, Ed.
Serendipity

March 2009

Sieve Email Filtering: Reject and Extended Reject Extensions

Status of This Memo

Copyright Notice

Abstract

This memo updates the definition of the Sieve mail filtering language "reject" extension, originally defined in RFC 3028.

A "Joe-job" is a spam run forged to appear as though it came from an innocent party, who is then generally flooded by automated bounces, Message Disposition Notifications (MDNs), and personal messages with

complaints. The original Sieve "reject" action defined in RFC 3028 required use of MDNs for rejecting messages, thus contributing to the flood of Joe-job spam to victims of Joe-jobs.

This memo updates the definition of the "reject" action to allow messages to be refused during the SMTP transaction, and defines the "ereject" action to require messages to be refused during the SMTP transaction, if possible.

The "ereject" action is intended to replace the "reject" action wherever possible. The "ereject" action is similar to "reject", but will always favor protocol-level message rejection.

Table of Contents

1. Introduction

The Sieve mail filtering language, as originally defined in RFC 3028 [RFC3028], specified that the "reject" action shall discard a message and send a Message Disposition Notification [MDN] to the envelope sender along with an explanatory message. The Sieve mail filtering language, as updated in RFC 5228 [SIEVE], does not define any "reject" action, hence that is the purpose of this document.

This document updates the definition of the "reject" action to permit refusal of the message during the SMTP transaction, if possible, and defines a new "ereject" action to require refusal of the message during the SMTP transaction, if possible.

An important goal of this document is to reduce the risk of Sieve scripts being used to perpetrate "Joe-job" spam runs, where the MDN is sent notifying the sender of a message of its non-delivery is in fact sent to an innocent third-party. The original Sieve "reject" action defined in RFC 3028 required use of MDNs for rejecting messages, thus contributing to the flood of Joe-job spam to victims of Joe-jobs. By rejecting the message at the protocol level, it is less likely that an MDN will be needed, and thus less likely that an MDN will be misdirected at an innocent third-party.

Implementations are further encouraged to use spam-detection systems to determine the level of risk associated with sending an MDN, and this document allows implementations to silently drop the MDN if the rejected message is deemed likely to be spam.

This document also describes how to use "reject"/"ereject" at varying points in the email stack: Mail Transfer Agent (MTA), Mail Delivery Agent (MDA), and Mail User Agent (MUA). See [EMAIL-ARCH] for a comprehensive discussion of these environments.

In general, an MDN is generated by an MUA, and can be used to indicate the status of a message with respect to its recipient, while a Delivery Status Notification (DSN) [DSN] is generated by an MTA, and can be used to indicate whether or not a message was received and delivered by the mail system.

Further discussion highlighting the risks of generating MDNs and the benefits of protocol-level refusal can be found in [Joe-DoS].

1.1. Conventions Used in This Document

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 [KEYWORDS].

Conventions for notations are as in Section 1.1 of RFC 5228 [SIEVE].

This document does not attempt to define spam or how it should be identified, nor does it attempt to define an email virus or how it should be detected. Implementors are advised to follow best practices and keep abreast of current research in these fields.

2. Sieve "reject" and "ereject" Extensions

2.1. Action ereject

Usage: ereject <reason: string>

Sieve implementations that implement the "ereject" action must use the "ereject" capability string.

The "ereject" action cancels the implicit keep and refuses delivery of a message. The "reason" string is a UTF-8 [UTF-8] string specifying the reason for refusal. How a message is refused depends on the capabilities of the mail component (MDA or MTA) executing the Sieve script. The Sieve interpreter MUST carry out one of the following actions (listed in order from most to least preferred), MUST carry out the most preferable action possible, and MUST fall back to lesser actions if a preferred action fails.

1. Refuse message delivery by sending a 5XX response code over SMTP [SMTP] or Local Mail Transfer Protocol (LMTP) [LMTP]. See Section 2.1.1 for more details.

2. Send a non-delivery report to the envelope sender ([REPORT] [DSN]), unless the envelope sender address is determined to be a forged or otherwise invalid address.

Note that the determination of whether or not an envelope sender is a forgery may be performed by site-specific and implementation-specific heuristic techniques, such as "return-path verification", details of which are outside the scope of this document. Implementations SHOULD log instances when a non-delivery report is not sent and the reason for not sending the report (e.g., content was spam, return-path invalid, etc.).

The "ereject" action MUST NOT be available in environments that do not support protocol-level rejection, e.g., an MUA, and MUST be available in all other environments that support the "reject" action.

Example:

```
require ["ereject"];

if address "from" "someone@example.com" {
    ereject "I no longer accept mail from this address";
}
```

2.1.1. Rejecting a Message at the SMTP/LMTP Protocol Level

Sieve implementations that are able to reject messages at the SMTP/LMTP level MUST do so and SHOULD use the 550 response code. Note that if a message is arriving over SMTP and has multiple recipients, some of whom have accepted the message, Section 2.1.2 defines how to reject such a message.

The risk that these actions will generate blowback spam are minimized but cannot be eliminated completely even in the case of "ereject", so caution is advised when using these actions to deal with messages determined to be spam.

Note that SMTP [SMTP] does not allow non-US-ACSII characters in the SMTP response text. If non-US-ACSII characters appear in the "reason" string, they can be sent at the protocol level if and only if the client and the server use an SMTP extension that allows for transmission of non-US-ACSII reply text. (One example of such an SMTP extension is described in [UTF8-RESP].) In the absence of such an SMTP extension, the Sieve engine MUST replace any "reason" string being sent at the protocol level and containing non-US-ACSII characters with an implementation-defined US-ACSII-only string.

Users who don't like this behavior should consider using the "reject" action described in Section 2.2, if available.

See Section 2.5 for the detailed instructions about performing protocol-level rejection.

2.1.2. Rejecting a Message by Sending a DSN

An implementation may receive a message via SMTP that has more than one RCPT TO that has been accepted by the server, and at least one but not all of them are refusing delivery (whether the refusal is caused by a Sieve "ereject" action or for some other reason). In this case, the server MUST accept the message and generate DSNs for all recipients that are refusing it. Note that this exception does not apply to LMTP, as LMTP is able to reject messages on a per-recipient basis. (However, the LMTP client may then have no choice but to generate a DSN to report the error, which may result in blowback spam.)

Note that according to [DSN], Delivery Status Notifications MUST NOT be generated if the MAIL FROM (or Return-Path) is empty.

The DSN message MUST follow the requirements of [DSN] and [REPORT]. The action-value field defined in [DSN], Section 2.3.3, MUST contain the value "failed". The human-readable portion of the non-delivery report MUST contain the "reason" string from the "ereject" action and SHOULD contain additional text alerting the apparent original sender that the message was refused by an email filter. This part of the report might appear as follows:

```
---------------------------------------------------------
Your message was refused by the recipient's mail filtering program.
The reason given is as follows:

I am not taking mail from you, and I don't want your birdseed,
either!
---------------------------------------------------------
```

2.2. Action reject

This section updates the definition of the "reject" action in Section 4.1 of RFC 3028 [RFC3028] and is an optional extension to [SIEVE].

```
       Usage:   reject <reason: string>
```

Sieve implementations that implement the "reject" action must use the "reject" capability string.

The "reject" action cancels the implicit keep and refuses delivery of a message. The "reason" string is a UTF-8 [UTF-8] string specifying the reason for refusal. Unlike the "ereject" action described above, this action would always favor preserving the exact text of the refusal reason. Typically, the "reject" action refuses delivery of a message by sending back an MDN to the sender (see Section 2.2.1). However, implementations MAY refuse delivery over SMTP/LMTP protocol (as detailed in Section 2.5), if and only if all of the following conditions are true:

1. The "reason" string consists of only US-ASCII characters
   or
   The "reason" string contains non-US-ASCII and both the client and server support and negotiate use of an SMTP/LMTP extension for sending UTF-8 responses.

2. LMTP protocol is used
   or
   SMTP protocol is used and the message has a single recipient
   or
   SMTP protocol is used, the message has multiple recipients, and all of them refused message delivery (whether or not Sieve is being used).

Example:

```
        require ["reject"];

        if size :over 100K {
            reject text:
Your message is too big.  If you want to send me a big attachment,
put it on a public web site and send me a URL.
.
            ;
        }
```

(Pretend that the "reason" string above contains some non-US-ACSII text.)

Implementations may use techniques as described in Section 2.1 to determine if a non-delivery report should not be sent to a forged sender. Implementations SHOULD log instances when a non-delivery report is not sent and the reason for not sending the report.

2.2.1. Rejecting a Message by Sending an MDN

The "reject" action resends the received message to the envelope sender specified by the MAIL FROM (or Return-Path) address, wrapping it in a "reject" form, explaining that it was rejected by the recipient.

Note that according to [MDN], Message Disposition Notifications MUST NOT be generated if the MAIL FROM (or Return-Path) is empty.

A reject message MUST take the form of a failure MDN as specified by [MDN]. The human-readable portion of the message, the first component of the MDN, contains the human-readable message describing the error, and it SHOULD contain additional text alerting the apparent original sender that mail was refused by an email filter.

The MDN disposition-field as defined in the MDN specification MUST be "deleted" and MUST have the "MDN-sent-automatically" and "automatic-action" modes set (see Section 3.2.6 of [MDN]).

In the following script, a message is rejected and returned to the sender.

```
    Example:
            require ["reject"];

            if header :contains "from" "coyote@desert.example.org" {
                reject text:
    I am not taking mail from you, and I don't
    want your birdseed, either!
    .
                ;
            }
```

For this script, the first part of the MDN might appear as follows:

```
---------------------------------------------------------
The message was refused by the recipient's mail filtering program.
The reason given was as follows:

I am not taking mail from you, and I don't want your birdseed,
either!
---------------------------------------------------------
```

2.3. Silent Upgrade from "reject" to "ereject"

Implementations MUST NOT silently upgrade "reject" actions to "ereject" actions in a Sieve script because this might lead to unpleasant changes of behavior not expected by the script owner.

User interfaces that present a generic rejection option, and generate Sieve script output, MAY switch from generating "reject" to "ereject" actions, so long as doing so does not create a confusing change for the script owner.

Script generators SHOULD ensure that a rejection action being executed as a result of an anti-spam/anti-virus positive test be done using the "ereject" action, as it is more suitable for such rejections.

Script generators MAY automatically upgrade scripts that previously used the "reject" action for anti-spam/anti-virus related rejections. Note that such generators MUST make sure that the target environment can support the "ereject" action.

2.4. Compatibility with Other Actions

This section applies equally to "reject" and "ereject" actions. All references to the "reject" action in this section can be replaced with the "ereject" action.

A "reject" action cancels the implicit keep.

Implementations MUST prohibit the execution of more than one "reject" in a Sieve script.

"reject" MUST be incompatible with the "vacation" [VACATION] action. It is NOT RECOMMENDED that implementations permit the use of "reject" with actions that cause mail delivery, such as "keep", "fileinto", and "redirect".

Making "reject" compatible with actions that cause mail delivery violates the RFC 5321 [SMTP] principle that a message is either delivered or bounced back to the sender. So bouncing a message back (rejecting) and delivering it will make the sender believe that the message was not delivered.

However, there are existing laws requiring certain organizations to archive all received messages, even the rejected ones. Also, it can be quite useful to save copies of rejected messages for later analysis.

Any action that would modify the message body will not have an effect on the body of any message refused by "reject" using an SMTP response code and MUST NOT have any effect on the content of generated DSN/MDNs.

2.5. Details of Protocol-Level Refusal

If the "reason" string consists of multiple CRLF separated lines, then the reason text MUST be returned as a multiline SMTP/LMTP response, per Section 4.2.1 of [SMTP]. Any line MUST NOT exceed the SMTP limit on the maximal line length. To make the "reason" string conform to any such limits, the server MAY insert CRLFs and turn the response into a multiline response.

In the following script (which assumes support for the "spamtest" [SPAMTEST] and "fileinto" extensions), messages that test highly positive for spam are refused.

```
Example:
        require ["ereject", "spamtest", "fileinto",
                 "comparator-i;ascii-numeric"];

        if spamtest :value "ge"
                    :comparator "i;ascii-numeric" "6" {
            ereject text:
AntiSpam engine thinks your message is spam.
It is therefore being refused.
Please call 1-900-PAY-US if you want to reach us.
.
            ;
        } elsif spamtest :value "ge"
                         :comparator "i;ascii-numeric" "4" {
            fileinto "Suspect";
        }
```

The following excerpt from an SMTP session shows it in action.

```
  ...
  C: DATA
  S: 354 Send message, ending in CRLF.CRLF.
   ...
  C: .
  S: 550-AntiSpam engine thinks your message is spam.
  S: 550-It is therefore being refused.
  S: 550 Please call 1-900-PAY-US if you want to reach us.
```

If the SMTP/LMTP server supports RFC 2034 [ENHANCED-CODES], it MUST prepend an appropriate Enhanced Error Code to the "reason" text. Enhanced Error code 5.7.1 or a more generic 5.7.0 are RECOMMENDED. With an Enhanced Error Code, the response to a DATA command in the SMTP example below will look like:

```
  S: 550-5.7.1 AntiSpam engine thinks your message is spam.
  S: 550-5.7.1 It is therefore being refused.
  S: 550 5.7.1 Please call 1-900-PAY-US if you want to reach us.
```

if the server selected "5.7.1" as appropriate.

If a Sieve implementation that supports "ereject" does not wish to immediately disclose the reason for rejection (for example, that it detected spam), it may delay immediately sending of the 550 error code by sending a 4XX error code on the first attempt to receive the message.

3. Changes from RFC 3028

Clarified that the "reject" action cancels the implicit keep. Extended the list of allowable actions on "reject" to include protocol-level message rejection.

Added the "ereject" action that is similar to "reject", but will always favor protocol-level message rejection.

4. Security Considerations

The introduction to this document discusses why rejecting messages before delivery is better than accepting and bouncing them.

While the details of techniques that can be used to determine when to silently drop a non-delivery report are outside the scope of this document, the explicit permission this document gives to take such action may enable denial-of-service situations. Techniques such as spam-checking, return-path verification, and others, can and do have false-positives. Care should be exercised to prevent the loss of legitimate messages by failing to notify the sender of non-delivery.

Security issues associated with email auto-responders are fully discussed in the Security Considerations section of [RFC3834]. This document is not believed to introduce any additional security considerations in this general area.

The "ereject" extension does not raise any other security considerations that are not already present in the base [SIEVE] specification, and these issues are discussed in [SIEVE].

5. IANA Considerations

The following section provides the IANA registrations for the Sieve extensions specified in this document.

5.1. "reject" Extension Registration

IANA is requested to update the registration for the Sieve "reject" extension as detailed below:

```
Capability name: reject
Description:     adds the "reject" action for refusing delivery
                 of a message.  The exact reason for refusal is
                 conveyed back to the client.
RFC number:      RFC 5429
Contact address: the Sieve discussion list <ietf-mta-filters@imc.org>
```

5.2. "ereject" Extension Registration

IANA is requested to replace the preliminary registration of the Sieve refuse extension with the following registration:

Capability name: ereject
Description: adds the "ereject" action for refusing delivery of a message. The refusal should happen as early as possible (e.g., at the protocol level) and might not preserve the exact reason for refusal if it contains non-US-ASCII text.
RFC number: RFC 5429
Contact address: the Sieve discussion list <ietf-mta-filters@imc.org>

6. References

6.1. Normative References

[DSN] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[ENHANCED-CODES] Freed, N., "SMTP Service Extension for Returning Enhanced Error Codes", RFC 2034, October 1996.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[LMTP] Myers, J., "Local Mail Transfer Protocol", RFC 2033, October 1996.

[MDN] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[REPORT] Vaudreuil, G., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 3462, January 2003.

[SIEVE] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", RFC 5228, January 2008.

[SMTP] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[UTF-8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[VACATION] Showalter, T. and N. Freed, "Sieve Email Filtering: Vacation Extension", RFC 5230, January 2008.

6.2. Informative References

[EMAIL-ARCH] Crocker, D., "Internet Mail Architecture", Work in Progress, October 2008.

[Joe-DoS] Frei, S., Silvestri, I., and G. Ollman, "Mail Non-Delivery Notice Attacks", April 2004, <http://www.techzoom.net/papers/mail_non_delivery_notice_attacks_2004.pdf>.

[RFC3028] Showalter, T., "Sieve: A Mail Filtering Language", RFC 3028, January 2001.

[RFC3834] Moore, K., "Recommendations for Automatic Responses to Electronic Mail", RFC 3834, August 2004.

[SPAMTEST] Daboo, C., "Sieve Email Filtering: Spamtest and Virustest Extensions", RFC 5235, January 2008.

[UTF8-RESP] Melnikov, A., "SMTP Language Extension", Work in Progress, June 2007.

Appendix A. Acknowledgements

Thanks to Ned Freed, Cyrus Daboo, Arnt Gulbrandsen, Kristin Hubner, Mark E. Mallett, Philip Guenther, Michael Haardt, and Randy Gellens for comments and corrections.

The authors gratefully acknowledge the extensive work of Tim Showalter as the author of the RFC 3028, which originally defined the "reject" action.

Appendix B. Contributors

Matthew Elvey
The Elvey Partnership, LLC
1819 Polk Street, Suite 133
San Francisco, CA 94109
USA

EMail: matthew@elvey.com

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Alexey.Melnikov@isode.com

Author's Address

Aaron Stone (editor)
Serendipity
260 El Verano Ave
Palo Alto, CA 94306
USA

EMail: aaron@serendipity.palo-alto.ca.us