Internet Engineering Task Force (IETF)
Request for Comments: 6468
Category: Standards Track
ISSN: 2070-1721

A. Melnikov
Isode Limited
B. Leiba
K. Li
Huawei Technologies

February 2012

Sieve Notification Mechanism: SIP MESSAGE

Abstract

This document describes a profile of the Sieve extension for notifications, to allow notifications to be sent over SIP MESSAGE.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6468.

Table of Contents

1. Introduction

1.1. Overview

The Notify extension [RFC5435] to the Sieve mail filtering language [RFC5228] is a framework for providing notifications by employing URIs that specify the notification mechanism. (See RFC 5435 for details about the motivation and use cases.) This document defines how Session Initiation Protocol (SIP) URIs [RFC3261] are used to generate notifications via SIP MESSAGE [RFC3428].

1.2. Terminology

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 RFC 2119 [RFC2119].

This document inherits terminology from the Sieve email filtering language [RFC5228], the Sieve Notify extension [RFC5435], and RFC 3261 [RFC3261].

2. Definition

The SIP MESSAGE mechanism defined in this document results in the sending of a SIP MESSAGE request to notify a recipient about an email message.

2.1. Notify Parameter "method"

The "method" parameter MUST be a URI that conforms to the SIP or SIPS URI scheme (as specified in RFC 3261 [RFC3261]) and that identifies a SIP or SIPS recipient of the notification. The URI MAY include the resource identifier portion of a SIP address and URI parameters. The URI MUST include the URI parameter "method", with the value "MESSAGE". Example:

```
      notify "sip:romeo@example.com;method=MESSAGE"
                                     --------------
```

Note that future specifications might extend this document and define Sieve notifications that use SIP methods other than "MESSAGE".

The processing application MUST form a request according to the rules specified in RFC 3261 [RFC3261].

Note that other URI schemes can also trigger SIP processing, but only SIP and SIPS are defined here. Future extensions might define other Sieve notification methods that use SIP through other URI schemes.

2.2. Notify Tag ":from"

The value of the ":from" tag MUST use the SIP "From" header field syntax; if the ":from" value is specified, has valid syntax, and is valid according to the implementation-specific security checks (see Section 3.3 of Sieve Notify [RFC5435]), then the notification SHOULD include the "From" SIP header field containing the value of the ":from" notify tag. If the specified value is not valid, then it is ignored.

All SIP authentication, including challenges and client certificates, SHOULD be done in the context of the Sieve engine -- the Sieve engine is the identity being authenticated. This avoids security issues associated with the Sieve engine’s having access to the end user’s SIP authentication credentials. The Sieve engine MAY use server-wide credentials (including applicable certificates) that are the same for all scripts. Alternatively, it MAY, for auditing purposes, use different sets of Sieve-engine credentials when operating on behalf of different users.

See Section 22 of RFC 3261 [RFC3261] for more information about SIP authentication.

2.3. Notify Tag ":options"

Handling of the ":options" tag is implementation specific. This document doesn't require presence of any option and doesn't define how options are processed.

2.4. Notify Tag ":importance"

The ":importance" tag is intended to convey the importance of the SIP MESSAGE notification, not the importance of the email message that generated the notification. The value of the ":importance" tag MAY, therefore, be transformed into SIP "Priority" header field (in addition to or instead of including it in the body of the message). Note that because the Sieve ":importance" tag only has three values, not all SIP "Priority" values can be represented in the transformation. If this transformation is done, the value of the "Priority" header field MUST be "urgent" if the value of the ":importance" tag is "1", "normal" if the value of the ":importance" tag is "2", and "non-urgent" if the value of the ":importance" tag is "3". There is no mapping to the SIP value "emergency", nor to any additional values that might be defined.

2.5. Notify tag ":message"

If the ":message" tag is included, it MUST be transformed into the message body of a SIP MESSAGE, which MUST have Content-Type value of "text/plain" with CHARSET="UTF-8". If the ":message" tag is not included, a default message will be used. The values of the "From" and "Subject" header fields of the triggering email message are particularly useful to users receiving notifications, and including them in the default message is generally a good idea, as shown in Section 3.2 below. (However, see the Security Considerations, Section 5.) The default body might also include the value of the ":importance" tag, if one is specified.

Note that in no case is the actual triggering message body included in the notification.

Implementations MUST comply with the SIP MESSAGE size limits, as discussed in Section 8 of RFC 3428 [RFC3428].

2.6. Other Definitions

An implementation MUST ignore any URI parameter it does not understand (the URI MUST be processed as if the parameter were not present). The URI "body" parameter can serve the same purpose as the Sieve ":message" tag, providing the message body of the SIP MESSAGE request. If both are present at the same time, the Sieve processing MUST ignore the "body" parameter.

Using the ":message" tag has advantages over using the "body" parameter. Because the ":message" tag is part of the "notify" statement syntax, it can be easier to include it in a script, and to do things such as variable substitutions [RFC5229] with it. It is also easier to include non-ASCII characters in the ":message" tag because such characters have to be encoded if they are within URI parameters, but can be included directly in UTF-8 in Sieve tag values.

The policy for retrying delivery of failed notifications is specified in RFC 3261 [RFC3261], according to the SIP error code returned during an attempt to deliver a SIP notification. In other words, unlike the situation with some other Sieve notification methods, retries for SIP MESSAGE notifications are controlled by the notification protocol itself (SIP).

2.7. Test notify_method_capability

Absent use of SIP extensions such as [RFC3856], it is impossible to tell in advance whether the notification recipient is online and able to receive a SIP MESSAGE. Expect the notify_method_capability test for "online" to frequently return "maybe" for this notification method.

3. Examples

In the following examples, the sender of the email has an address of juliet@example.org, the entity to be notified has a SIP address of <sip:romeo@example.com>, and the notification service has a SIP address <sip:notifier@example.com>.

3.1. Example 1

The following is a basic Sieve notify action with only a method:

```
notify "sip:romeo@example.com;method=MESSAGE"
```

The resulting SIP MESSAGE request might be as follows:

```
MESSAGE sip:romeo@example.com SIP/2.0
Via: SIP/2.0/TCP notifier.example.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:notifier@example.com;tag=32328
To: sip:romeo@example.com
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Date: Sat, 13 Nov 2010 23:29:00 GMT
Content-Type: text/plain
Content-Length: 53

<juliet@example.com> wrote: Contact me immediately!
```

In the example above, the email message was received from juliet@example.com and had "Subject: Contact me immediately!"

3.2. Example 2

The following is a more advanced Sieve notify action with a method, importance, subject, and message:

```
notify :importance "1"
    :message "You got new mail!"
    "sip:romeo@example.com;method=MESSAGE?subject=SIEVE"

MESSAGE sip:romeo@example.com SIP/2.0
Via: SIP/2.0/TCP notifier.example.com;branch=z9hG4bK776sgdkse
Max-Forwards: 70
From: sip:notifier@example.com;tag=32328
To: sip:romeo@example.com
Subject: SIEVE
Priority: urgent
Call-ID: asd88asd77a@1.2.3.4
CSeq: 1 MESSAGE
Date: Fri, 08 Apr 2011 06:54:00 GMT
Content-Type: text/plain
Content-Length: 19

You got new mail!
```

4. Requirements Conformance Checklist

Section 3.8 of Sieve Notify [RFC5435] specifies a set of requirements for Sieve notification methods. A checklist is provided here to show conformance of the SIP MESSAGE notification method.

1. No new Sieve tags have been added to the "notify" action.

2. An implementation of the SIP MESSAGE notification method SHOULD NOT modify the final notification text, except to comply with SIP MESSAGE length limits. Deployments MAY make operational decisions about notification text, for reasons such as privacy and confidentiality. Modification of characters themselves should not be necessary, since the SIP MESSAGE body is encoded in UTF-8 [RFC3629].

3. An implementation MAY ignore parameters specified in the ":importance" and ":options" tags.

4. A default message is suggested in Section 2.5.

5. A notification sent via the SIP MESSAGE notification method MAY include the Date header field containing the date-time of the moment when the SIP MESSAGE notification was generated.

6. The notification source is identified through the SIP "From:" header field, via the Sieve Notify ":from" tag (see Section 2.2).

7. An implementation MUST NOT include any extraneous information not specified in parameters to the notify action.

8. An implementation MUST ignore any URI parameters it does not understand (i.e., the URI MUST be processed as if the action or parameter were not present). See Section 2.6 for more details.

9. The notify_method_capability test for the "online" notification-capability behaves as described in Section 2.7.

10. The policy for retrying delivery of failed notifications is specified in RFC 3261 [RFC3261], as noted in Section 2.6.

5. Security Considerations

Depending on the information included, sending a notification can be, from a confidentiality point of view, comparable to forwarding mail to the notification recipient. Care must be taken when automatically forwarding information, such as the sender and the subject of a

message, to ensure that confidential information is not sent into an insecure environment or over an insecure channel. Depending upon the environment, this might entail using SIPS URIs, not sending information about the subject and/or the sender, or applying heuristics to the message to determine what may be sent.

As required by RFC 3428, user agents that support the SIP MESSAGE request MUST implement end-to-end authentication, body integrity, and body confidentiality mechanisms. At the time of this writing, there is not widespread deployment of SIP end-to-end security, so there can be cases where it is not possible to use it, even though it is implemented on one end. It's important to note that such situations are open to exposure of user credentials, message content, and other private information via man-in-the-middle and other passive attacks.

The Sieve Notify extension specifies that notification methods MUST provide mechanisms for avoiding notification loops. In this case, the SIP protocol itself prevents loops, and no explicit work is needed within the notification mechanism. In situations where a SIP MESSAGE notification can result in an email message that could generate another SIP MESSAGE notification, loop prevention through rate detection and limiting might be necessary. An implementation might detect too many notifications within a given time period, too many triggered by a particular sender, too many with the same subject, or the like, and shut off the affected notifications for a period of time or until manual intervention turns them back on.

If SIP MESSAGE requests might be billed by the message, or the use of them might deplete a user's quota of messages, notification by this mechanism can present a situation where someone using a large number of messages to generate a large number of notifications will cause a significant expense to the recipient. Because there is no external way an attacker can tell that this is the case, such an attack would likely be a random or nuisance attack. Nevertheless, users might be warned of potential costs when they set up SIP MESSAGE notifications.

Other security considerations given in the Sieve base specification [RFC5228], the Sieve Notify extension [RFC5435], and RFC 3261 [RFC3261] are also relevant to this document.

6. IANA Considerations

The following template provides the IANA registration of the Sieve notification mechanism specified in this document. This information has been added to the list of Sieve notification mechanisms maintained at <http://www.iana.org/assignments/sieve-notification>.

```
To: iana@iana.org
Subject: Registration of new Sieve notification mechanism
Mechanism name: sip-message
Mechanism URI: SIP/SIPS as specified in RFC 3261 [RFC3261]
Mechanism-specific options: none
Standards Track/IESG-approved experimental RFC number: [RFC6468]
Person and email address to contact for further information:
    See authors of [RFC6468]
```

7. Acknowledgements

This document borrows some text from RFC 5437 [RFC5437].

Henning Schulzrinne (hgs@cs.columbia.edu) was a special contributor to this document, with early work and reviews.

The authors would like to thank Adam Roach and Eric Burger for their helpful comments. Ben Campbell did a very thorough RAI-team review, as well as a follow-up review to make sure we resolved all of his issues satisfactorily. This document was greatly improved by their input.

Qian Sun contributed text to this document.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5228] Guenther, P. and T. Showalter, "Sieve: An Email Filtering Language", RFC 5228, January 2008.

[RFC5435] Melnikov, A., Leiba, B., Segmuller, W., and T. Martin, "Sieve Email Filtering: Extension for Notifications", RFC 5435, January 2009.

8.2. Informative References

[RFC3856] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, August 2004.

[RFC5229] Homme, K., "Sieve Email Filtering: Variables Extension", RFC 5229, January 2008.

[RFC5437] Saint-Andre, P. and A. Melnikov, "Sieve Notification Mechanism: Extensible Messaging and Presence Protocol (XMPP)", RFC 5437, January 2009.

Authors' Addresses

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

EMail: Alexey.Melnikov@isode.com
URI: http://www.melnikov.ca/

Barry Leiba
Huawei Technologies

Phone: +1 646 827 0648
EMail: barryleiba@computer.org
URI: http://internetmessagingtechnology.org/

Kepeng Li
Huawei Technologies
Huawei Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R. China

Phone: +86-755-28974289
EMail: likepeng@huawei.com