Internet Engineering Task Force (IETF)
Request for Comments: 6133
Category: Informational
ISSN: 2070-1721

R. George
B. Leiba
Huawei Technologies
A. Melnikov
Isode Limited

July 2011

Sieve Email Filtering: Use of Presence Information with Auto-Responder Functionality

Abstract

This document describes how the Sieve email filtering language, along with some extensions, can be used to create automatic replies to incoming electronic mail messages based on the address book and presence information of the recipient.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see 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/rfc6133.

Copyright Notice

Copyright (c) 2011 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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. How To Create Auto-Replies 3
3. Example Use Cases for Auto-Replies 3
4. Security Considerations 6
5. Normative References 8

1. Introduction

This document describes how the Sieve email filtering language [RFC5228], along with some extensions [RFC5230] [RFC5435] [RFC6134] [RFC6132] [RFC6131], can be used to generate automatic replies to incoming electronic mail messages based on the presence information of the recipient. This can be used, for example, to inform the sender that messages will not be answered immediately because the recipient is busy or away.

The auto-reply message can additionally be based on information about the sender from the recipient's address book, sub-lists therefrom, or other lists available to the recipient, so that different senders might get different responses. The recipient can create separate rules for friends, family members, colleagues, and so on.

This can be used in mail filtering software, email-based information services, and other automatic responder situations. There are many programs currently in use that automatically respond to email. Some of them send many useless or unwanted responses, or send responses to inappropriate addresses. The mechanism described herein will help avoid those problems (but see the discussion in Section 4). Implementations need to take care of tracking previous messages received from the same sender, and they will start or stop sending responses as the presence status of the recipient changes.

An important note, though: users of any auto-reply mechanism should really think about whether automatic replies are necessary, and at what interval they make sense when they are. Email is not Instant Messaging, and senders generally expect that replies might take a while. Consider whether it's truly important to tell people that you'll read their mail in an hour or so, or whether that can just be taken as how email works. There are times when this makes sense, but let's not use it to exacerbate information overload. Judicious use of appropriate presence information might serve to mitigate these issues.

Implementors, therefore, need to consider this with respect to the features they expose to users, and the potential for inappropriate use those features represent. The ability to create auto responders

might be hidden behind an "advanced" button, and users might be warned of the consequences and advised of the considerations in the previous paragraph.

2. How To Create Auto-Replies

When an email message arrives, the Sieve script can use the notify_method_capability of the Notify extension [RFC5435] to check the recipient's presence information. The Notify-presence extension [RFC6132] makes additional presence, such as "away" and "do not disturb" status, available. The script can use the External-lists extension [RFC6134] to look the sender up in the recipient's address book or other list. If the information retrieved warrants an auto-reply message, the message can then be composed based on that information.

The Vacation extension [RFC5230] provides an easy way to send the auto-reply message to the sender, as it automatically keeps track of the automatic replies and attempts to avoid excessive messages and mail loops. The Vacation-seconds extension [RFC6131] allows auto-replies to be sent this way more frequently than once per day, when that's appropriate. (Alternatively, the script can use the Notify extension [RFC5435] to send a notification by a means other than email.)

Personal and Group Responders can refuse to generate responses except to known correspondents or addresses otherwise known to the recipient. Such responders can also generate different kinds of responses for "trusted" vs. "untrusted" addresses. This might be useful, for instance, to avoid inappropriate disclosure of personal or confidential information to arbitrary addresses.

3. Example Use Cases for Auto-Replies

1. In this example, we check that the envelope "from" is in the recipient's address book [RFC6134] and that the recipient's presence shows "extended away" [RFC6132]. If both of those are true, the "vacation" action [RFC5230] is used to send an auto-reply, making sure we don't reply to the same sender more than once every half hour [RFC6131]. The variables extension [RFC5229] is used to extract the value of the recipient's natural-language presence status message, which will be used as the response to the sender.

```
require ["envelope", "extlists", "enotify", "variables",
         "vacation-seconds"];
if allof (
    envelope :list "from" ":addrbook:default",
    notify_method_capability "xmpp:me@example.com" "show" "xa"
  ) {
    # :matches "*" is used here to extract the value
    if notify_method_capability :matches
        "xmpp:myjid@example.com" "status" "*" {
      set "resp_msg" "${1}";
    } else {
      set "resp_msg" "Away for a while, without access to email.";
    }
    vacation :handle "ext-away" :seconds 1800 "${resp_msg}";
  }
```

2. In the next example, we'll check for the recipient's personal assistant, and give very detailed information about the recipient's status to that sender. For other senders in the "family" and "friends" lists, we'll also send an auto-reply. Other senders will be considered less important, and don't need auto-replies.

```
require ["envelope", "extlists", "enotify", "vacation-seconds"];

if envelope :is "from" "assistant@example.com"
  {
    if notify_method_capability "xmpp:me@example.com" "show" "away"
      {
        vacation :handle "away" :seconds 600
            "I'm away for now, but I'll be back soon.";
      }
    elsif notify_method_capability "xmpp:me@example.com" "show" "dnd"
      {
        vacation :handle "dnd" :seconds 1800
            "I'm not to be disturbed.  I'll check mail later.";
      }
    elsif notify_method_capability "xmpp:me@example.com" "show" "xa"
      {
        vacation :handle "ext-away" :seconds 3600
            "I'm away for a while, without access to email.";
      }
    elsif notify_method_capability "xmpp:me@example.com" "busy" "yes"
      {
        vacation :handle "busy" :seconds 1800
            "I'm very busy, but might check email now and then.";
      }
```

```
  }
elsif envelope :list "from" [":addrbook:family",
                             ":addrbook:friends"]
  {
    if notify_method_capability "xmpp:me@example.com" "show"
           ["away", "dnd", "xa"]
      {
        vacation :handle "away" :seconds 3600
            "I'm not available to respond to email.";
      }
  }
else
  { # We could respond as below, making it only once a day
    # for less important senders.  Better to just omit
    # that, though (see the end of the Introduction section).
    #
    # vacation :handle "catchall" :days 1
    #     "I got your message, and might read it eventually.";
  }
```

3. For this example, if the sender is a work colleague and the recipient is on extended away status, then reply with a message giving alternative contact information. The message might also include details about the reason for the absence, or other personal or confidential information that shouldn't be shared with senders who aren't associated with the recipient's company.

```
require ["envelope", "extlists", "enotify", "vacation"];

if envelope :list "from" ":addrbook:co-workers"
  {
    if notify_method_capability "xmpp:me@example.com" "show" "xa"
      {
        vacation :handle "bigtrip" :days 3
            "I'm on an extended business trip to Texas for the Foo
             project.  Contact my backup, Susan <susan@example.com>,
             or call my assistant on +1 666 555 1234 if you urgently
             need to contact me.";
      }
  }
```

4. This example is used to send an acknowledgment to every message received. A :seconds value of zero is used to reply to every message, with no removal of duplicates to the same sender. This requires that the Sieve engine allow an interval of zero; if it

does not, and it imposes a minimum value, not every message will receive an auto-reply.

```
require ["envelope", "extlists", "vacation-seconds"];

if not envelope :list "from" ":addrbook:staff"
  {
    vacation :handle "auto-resp" :seconds 0
        "Your request has been received.  A service
         representative will contact you as soon as
         possible, usually within one business day.";
  }
```

5. This example uses the same structure to automatically send a copy of each incoming message to the recipient's backup, if the sender is a customer contact or co-worker, or if the message's subject includes the word "urgent".

```
require ["envelope", "extlists", "enotify"];

if anyof (
    envelope :list "from" [":addrbook:customers",
                           ":addrbook:co-workers"],
    header :contains "subject" "urgent"
  ) {
    if notify_method_capability "xmpp:me@example.com" "show" "xa"
      {
        redirect "susan@example.com"; # send a copy to my backup
        keep; # also keep a copy for myself
      }
  }
}
```

4. Security Considerations

See the Security Considerations sections of the following specifications for discussion of security considerations not covered here:

- Sieve base specification [RFC5228]
- Sieve Vacation extension [RFC5230]
- Vacation "Seconds" parameter [RFC6131]

- Sieve Externally Stored Lists extension [RFC6134]

- Sieve Notify extension [RFC5435] (and any applicable notification methods)

This document describes how to set up a system that creates automatic replies in an intelligent way. Despite the "intelligence", errors in scripts can result in too many auto-reply messages, especially when the reply interval is minimal (using the "notify" action, or the "vacation" action with a small value for ":seconds").

Despite the "intelligence", too, errors in scripts can result in private information getting to senders inappropriately. In example 3 in Section 3, for instance, if the :list test checks the wrong list, or none at all, information about the recipient's business trip might be sent to someone who has no need to know about it, and that information should not have been sent.

Even without errors in scripts, a sender who recognizes that auto-replies are dependent upon the recipient's presence can use that fact to probe the presence information. One result of that can be that the sender discerns changes in the recipient's presence that the sender would normally not be allowed to see, making this an unintentional back door into the user's presence information. Another result is that this can create a "covert channel", allowing the recipient to send information to a sender by changing his presence information, his address book, and/or his Sieve script (though in this regard, the exposure is comparable to any other case of shared presence information).

An auto responder can cause leaks of other pieces of information, including potentially providing the ability to attack cryptographic keying material. For example, using the time it takes to perform a cryptographic operation, an attacker may obtain information about the secret key. An auto responder that doesn't take timing into account could accidentally leak this kind of information.

Moreover, if an auto responder script directly returns the results of a cryptographic operation, that could also provide an attack vector. For example, if a script returns the results of a decryption operation, an attacker can send an arbitrarily encrypted message and use the results as a chosen cyphertext attack to decode the encryption key. Authors of scripts should be careful about what information they return to senders.

5. Normative References

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

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

[RFC5230] Showalter, T. and N. Freed, "Sieve Email Filtering: Vacation Extension", RFC 5230, January 2008.

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

[RFC6131] George, R. and B. Leiba, "Sieve Vacation Extension: "Seconds" Parameter", RFC 6131, July 2011.

[RFC6132] George, R. and B. Leiba, "Sieve Notification Using Presence Information", RFC 6132, July 2011.

[RFC6134] Melnikov, A. and B. Leiba, "Sieve Extension: Externally Stored Lists", RFC 6134, July 2011.

Authors' Addresses

Robins George
Huawei Technologies
Bangalore, Karnataka 560071
India

Phone: +91-080-41117676
EMail: robinsgv@gmail.com

Barry Leiba
Huawei Technologies

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

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/