Sieve Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 1, 2010

A. Melnikov
Isode Limited
B. Leiba
Huawei Technologies

August 28, 2009

Sieve Extension: Externally Stored Lists
draft-ietf-sieve-external-lists-01

Status of this Memo

Copyright Notice

Abstract

The Sieve scripting language can be used to implement whitelisting, blacklisting, and personal distribution lists. Currently, this

requires that all members of such lists be hardcoded in the script itself. Whenever a member of a list is added or deleted, the script needs to be updated and possibly uploaded to a mail server.

This document defines a Sieve extension for accessing externally stored lists -- lists whose members are stored externally to the script, such as using LDAP (RFC 4510), ACAP (RFC 2244), or relational databases.

ToDo

- Need a way to advertise supported URI schemas in ManageSieve and ihave.

Table of Contents

1. Introduction

This document specifies an extension to the Sieve language [Sieve] for checking membership in an external list or for redirecting messages to an external list of recipients. An "external list" is a list whose members are stored externally to the Sieve script, such as using LDAP [LDAP], ACAP [ACAP], or relational databases.

This extension adds a new match type to the "address", "envelope", and "header" tests, and a new tagged argument to the "redirect" action.

1.1. Conventions used in this document

Conventions for notations are as in [Sieve] section 1.1, including the use of [ABNF].

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 [Kwds].

2. Extlists extension

2.1. Capability Identifier

The capability string associated with the extension defined in this document is "extlists".

2.2. :list match type for "address", "envelope", and "header" tests

ABNF:

```
MATCH-TYPE  =/ ":list"
```

The new ":list" match type changes the interpretation of the "key-

list" parameter (the second parameter) to the "address"/"envelope"/"header" test. [[anchor4: This is not quite right. Ned convinced me that this new match type can apply to other tests as well.]] When the match type is ":list", the key-list becomes a list of names of externally stored lists. The external lists are queried, perhaps through a list-specific mechanism, and the test evaluates to "true" if any of the specified values matches any member of one or more of the lists.

For example, testing ':header ["to", "cc"]' against a list would cause each "to" and "cc" value, ignoring leading and trailing whitespace, to be queried. When any value is found to belong to the

list, the queries may stop and the test returns "true". If no value belongs to the list, the test returns "false".

For some lists, the Sieve engine might directly retrieve the list and make its own comparison. Other lists might not work that way -- they might provide a way to ask if a value is in the list, but not permit retrieval of the list itself. It is up to the Sieve implementation to understand how to interact with any supported list. If the Sieve engine is permanently unable to query the list (perhaps because the list doesn't support the required operation), the test MUST result in a runtime error in the Sieve script.

See Section 2.4 for the detailed description of syntax used for naming externally stored lists.

2.3. :list tagged argument to the "redirect" action

Usage: redirect :list <ext-list-name: string>

The "redirect" action with the ":list" argument is used to send the message to one or more email addresses stored in the externally stored list 'ext-list-name'. This variant of the redirect command can be used to implement a personal distribution list.

Use of this feature requires that the list resolve to a list of email addresses, and that the Sieve engine be able to enumerate those addresses. [[anchor6: Alexey would like the option of allowing the list handler to enumerate the addresses and do the redirect there. (Ned seems to agree that this should be allowed.) Barry thinks

that's contrary to Sieve, which expects to queue the redirect action for processing at a later stage, and that it would be a bad idea to have the redirect happen in the list handler. The WG needs to resolve this issue.]] In cases where, for example, a list contains hashed email address values or an email address pattern ("sz*@example.com", "*+ietf@example.net"), it will not be possible to redirect to that list.

If the Sieve engine [[anchor7: or list handler?]] is permanently unable to enumerate the list or the list does not resolve to email addresses, the situation MUST result in a runtime error in the Sieve script.

See Section 2.4 for the detailed description of syntax used for naming externally stored lists.

2.4. Syntax of an externally stored list name

A name of an externally stored list is always an absolute URI [URI]. Implementations might find URLs such as [LDAP], [CardDAV], or [TAG-URI] to be useful for naming external lists.

The "tag" URI scheme [TAG-URI] can be used to represent opaque, but user friendlier identifiers. Resolution of such identifiers is going to be implementation specific and it can help in hiding the complexity of an implementation from end users. For example, an implementation can provide a web interface for managing lists of users stored in LDAP. Requiring users to know generic LDAP URL syntax might not be very practical, due to its complexity. An implementation can instead use a fixed tag URI prefix such as "tag:example.com,<date>:" (where <date> can be, for example, a date generated once on installation of the web interface and left untouched upon upgrades) and the prefix doesn't even need to be shown to end users.

2.5. Examples

[[anchor8: Barry: This example looks wrong: the "envelope" test is

probably not right. Should it really be using the :list test? It's testing :detail, so I think it should just be a simple test, maybe ' :is "mylist" ' or some such. No? Alexey: I think this test is correct. This is checking for known plus addresses (parts).]]

Example 1 uses the "envelope" option [Sieve] and the "subaddress" extension [Subaddress]:

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

# Submission from list members is sent to all members
if allof (envelope :detail :list "to"
                   "tag:example.com,2009-05-28:mylist",
          header :list "from"
                 "tag:example.com,2009-05-28:mylist") {
    redirect :list "tag:example.com,2009-05-28:mylist";
}
```

3. Security Considerations

Security considerations related to the "address"/"envelope"/"header" tests and "redirect" action discussed in [Sieve] also apply to this document.

A failure to retrieve data due to the server storing the external

list membership being down or otherwise inaccessible may alter the result of Sieve processing. Implementations SHOULD treat a temporary failure to retrieve or verify external list membership in the same manner as a temporary failure to retrieve a Sieve script. For example, if the Sieve script is stored in the Lightweight Directory Access Protocol [LDAP] and the script can't be retrieved when a message is processed, then the agent performing Sieve processing can either assume that the script doesn't exist or delay message delivery until the script can be retrieved successfully. External list memberships should be treated as if they are a part of the script itself, so a temporary failure to retrieve or query them should be handled in the same way as a temporary failure to retrieve the Sieve script itself.

Protocols/APIs used to retrieve/verify external list membership MUST

provide an appropriate level of confidentiality and authentication. Usually, that will be at least the same level of confidentiality as protocols/APIs used to retrieve Sieve scripts, but only the implementation (or deployment) will know what is appropriate. There's a difference, for example, between making an LDAP request on a closed LAN that's only used for trusted servers (it may be that neither encryption nor authentication is needed), on a firewalled LAN internal to a company (it might be OK to skip encryption, depending upon policy), and on the open Internet (encryption and authentication are probably both required). It also matters whether the list being accessed is private or public (no encryption or authentication may be needed for public data, even on the Internet).

Implementations of this extensions should keep in mind that matching values against an externally stored list can be IO and/or CPU intensive. This can be used to deny service to the mailserver and/or to servers providing access to externally stored mailing lists. A naive implementation, such as the one that tries to retrieve content of the whole list to perform matching can make this worse. But note that many protocols that can be used for accessing externally stored lists support flexible searching features that can be used to minimize network traffic and load on the directory service. For example, LDAP allows for search filters. Implementations SHOULD use such features whenever they can.

Many organizations support external lists with thousands of recipients. In order to avoid mailbombs when redirecting a message to an externally stored list, implementations SHOULD enforce limits on the number of recipients and/or on domains to which such recipients belong.

4. IANA Considerations

The following template specifies the IANA registration of the notify Sieve extension specified in this document:

To: iana@iana.org
Subject: Registration of new Sieve extension
Capability name: extlists

Description: adds the ':list' tagged argument to 'address', 'header' and 'envelope' tests, and to the 'redirect' action. The ':list' argument changes address/header/envelope test to match values against values stored in one or more externally stored list. The ':list' argument to the redirect action changes the redirect action to forward the message to email addresses stored in the externally stored list.
RFC number: this RFC
Contact address:
The Sieve discussion list <ietf-mta-filters@imc.org>

This information should be added to the list of sieve extensions given on http://www.iana.org/assignments/sieve-extensions.

5. Acknowledgements

Thanks to Alexandros Vellis, Barry Leiba, Nigel Swinson, Kjetil Torgrim Homme, Dave Cridland, Cyrus Daboo, Pete Resnick for ideas, comments and suggestions.

6. References

6.1. Normative References

[ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

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

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

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

6.2. Informative References

[ACAP] Newman, C. and J. Myers, "ACAP -- Application Configuration Access Protocol", RFC 2244, November 1997.

[CardDAV] Daboo, C., "vCard Extensions to WebDAV (CardDAV)", work in progress, draft-ietf-vcarddav-carddav, July 2009.

[LDAP] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

[Subaddress] Murchison, K., "Sieve Email Filtering: Subaddress Extension", RFC 5233, January 2008.

[TAG-URI] Kindberg, T. and S. Hawke, "The 'tag' URI Scheme", RFC 4151, October 2005.

Authors' Addresses

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

Email: Alexey.Melnikov@isode.com

Barry Leiba
Huawei Technologies
USA

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