Network Working Group
Request for Comments: 5231
Obsoletes: 3431
Category: Standards Track

W. Segmuller
B. Leiba
IBM T.J. Watson Research Center

January 2008

Sieve Email Filtering: Relational Extension

Status of This Memo

Abstract

This document describes the RELATIONAL extension to the Sieve mail filtering language defined in RFC 3028. This extension extends existing conditional tests in Sieve to allow relational operators. In addition to testing their content, it also allows for testing of the number of entities in header and envelope fields.

This document obsoletes RFC 3431.

Table of Contents

1. Introduction 2
2. Conventions Used in This Document 2
3. Comparators 2
4. Match Types 3
4.1. Match Type VALUE 3
4.2. Match Type COUNT 3
5. Interaction with Other Sieve Actions 4
6. Example 4
7. Extended Example 6
8. Changes since RFC 3431 6
9. IANA Considerations 7
10. Security Considerations 7
11. Normative References 7

1. Introduction

The RELATIONAL extension to the Sieve mail filtering language [Sieve] provides relational operators on the address, envelope, and header tests. This extension also provides a way of counting the entities in a message header or address field.

With this extension, the Sieve script may now determine if a field is greater than or less than a value instead of just equivalent. One use is for the x-priority field: move messages with a priority greater than 3 to the "work on later" folder. Mail could also be sorted by the from address. Those userids that start with 'a'-'m' go to one folder, and the rest go to another folder.

The Sieve script can also determine the number of fields in the header, or the number of addresses in a recipient field, for example, whether there are more than 5 addresses in the to and cc fields.

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

2. 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 BCP 14, RFC 2119.

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

3. Comparators

This document does not define any comparators or exempt any comparators from the require clause. Any comparator used must be treated as defined in [Sieve].

The "i;ascii-numeric" comparator, as defined in [RFC4790], MUST be supported for any implementation of this extension. The comparator "i;ascii-numeric" MUST support at least 32-bit unsigned integers.

Larger integers MAY be supported. Note: the "i;ascii-numeric" comparator does not support negative numbers.

4. Match Types

This document defines two new match types. They are the VALUE match type and the COUNT match type.

The syntax is:

```
MATCH-TYPE =/ COUNT / VALUE

COUNT = ":count" relational-match

VALUE = ":value" relational-match

relational-match = DQUOTE
        ("gt" / "ge" / "lt" / "le" / "eq" / "ne") DQUOTE
        ; "gt" means "greater than", the C operator ">".
        ; "ge" means "greater than or equal", the C operator ">=".
        ; "lt" means "less than", the C operator "<".
        ; "le" means "less than or equal", the C operator "<=".
        ; "eq" means "equal to", the C operator "==".
        ; "ne" means "not equal to", the C operator "!=".
```

4.1. Match Type VALUE

The VALUE match type does a relational comparison between strings.

The VALUE match type may be used with any comparator that returns sort information.

A value from the message is considered the left side of the relation. A value from the test expression, the key-list for address, envelope, and header tests, is the right side of the relation.

If there are multiple values on either side or both sides, the test is considered true if any pair is true.

4.2. Match Type COUNT

The COUNT match type first determines the number of the specified entities in the message and does a relational comparison of the number of entities, as defined below, to the values specified in the test expression.

The COUNT match type SHOULD only be used with numeric comparators.

The Address Test counts the number of addresses (the number of "mailbox" elements, as defined in [RFC2822]) in the specified fields. Group names are ignored, but the contained mailboxes are counted.

The Envelope Test counts the number of addresses in the specified envelope parts. The envelope "to" will always have only one entry, which is the address of the user for whom the Sieve script is running. Using this test, there is no way a Sieve script can determine if the message was actually sent to someone else. The envelope "from" will be 0 if the MAIL FROM is empty, or 1 if MAIL FROM is not empty.

The Header Test counts the total number of instances of the specified fields. This does not count individual addresses in the "to", "cc", and other recipient fields.

In all cases, if more than one field name is specified, the counts for all specified fields are added together to obtain the number for comparison. Thus, specifying ["to", "cc"] in an address COUNT test compares the total number of "to" and "cc" addresses; if separate counts are desired, they must be done in two comparisons, perhaps joined by "allof" or "anyof".

5. Interaction with Other Sieve Actions

This specification adds two match types. The VALUE match type only works with comparators that return sort information. The COUNT match type only makes sense with numeric comparators.

There is no interaction with any other Sieve operations, nor with any known extensions. In particular, this specification has no effect on implicit KEEP, nor on any explicit message actions.

6. Example

Using the message:

```
received: ...
received: ...
subject: example
to: foo@example.com, baz@example.com
cc: qux@example.com
```

The test:

```
address :count "ge" :comparator "i;ascii-numeric"
                ["to", "cc"] ["3"]
```

would evaluate to true, and the test

```
anyof ( address :count "ge" :comparator "i;ascii-numeric"
                ["to"] ["3"],
        address :count "ge" :comparator "i;ascii-numeric"
                ["cc"] ["3"] )
```

would evaluate to false.

To check the number of received fields in the header, the following test may be used:

```
header :count "ge" :comparator "i;ascii-numeric"
               ["received"] ["3"]
```

This would evaluate to false. But

```
header :count "ge" :comparator "i;ascii-numeric"
               ["received", "subject"] ["3"]
```

would evaluate to true.

The test:

```
header :count "ge" :comparator "i;ascii-numeric"
               ["to", "cc"] ["3"]
```

will always evaluate to false on an RFC 2822 compliant message [RFC2822], since a message can have at most one "to" field and at most one "cc" field. This test counts the number of fields, not the number of addresses.

7. Extended Example

```
require ["relational", "comparator-i;ascii-numeric", "fileinto"];

if header :value "lt" :comparator "i;ascii-numeric"
          ["x-priority"] ["3"]
{
   fileinto "Priority";
}

elsif address :count "gt" :comparator "i;ascii-numeric"
           ["to"] ["5"]
{
   # everything with more than 5 recipients in the "to" field
   # is considered SPAM
   fileinto "SPAM";
}

elsif address :value "gt" :all :comparator "i;ascii-casemap"
           ["from"] ["M"]
{
   fileinto "From N-Z";
} else {
   fileinto "From A-M";
}

if allof ( address :count "eq" :comparator "i;ascii-numeric"
                   ["to", "cc"] ["1"] ,
           address :all :comparator "i;ascii-casemap"
                   ["to", "cc"] ["me@foo.example.com"] )
{
   fileinto "Only me";
}
```

8. Changes since RFC 3431

Apart from several minor editorial/wording changes, the following list describes the notable changes to this specification since RFC 3431.

- Updated references, including changing the comparator reference from the Application Configuration Access Protocol (ACAP) to the "Internet Application Protocol Collation Registry" document [RFC4790].

- Updated and corrected the examples.

- Added definition comments to ABNF for "gt", "lt", etc.
- Clarified what RFC 2822 elements are counted in the COUNT test.
- Removed the requirement to strip white space from header fields before comparing; a more general version of this requirement has been added to the Sieve base spec.

9. IANA Considerations

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

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

Capability name: relational
Description: Extends existing conditional tests in Sieve language to allow relational operators
RFC number: RFC 5231
Contact address: The Sieve discussion list <ietf-mta-filters@imc.org>

10. Security Considerations

An implementation MUST ensure that the test for envelope "to" only reflects the delivery to the current user. Using this test, it MUST not be possible for a user to determine if this message was delivered to someone else.

Additional security considerations are discussed in [Sieve].

11. Normative References

[ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

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

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC4790] Newman, C., Duerst, M., and A. Gulbrandsen, "Internet Application Protocol Collation Registry", RFC 4790, March 2007.

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

Authors' Addresses

Wolfgang Segmuller
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
US

Phone: +1 914 784 7408
EMail: werewolf@us.ibm.com

Barry Leiba
IBM T.J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
US

Phone: +1 914 784 7941
EMail: leiba@watson.ibm.com

Full Copyright Statement

Intellectual Property