Sieve Working Group
Internet-Draft
Expires: September 24, 2010

K. Murchison
Carnegie Mellon University
N. Freed
Oracle Corporation

March 23, 2010

Sieve Email Filtering: Regular Expression Extension
draft-ietf-sieve-regex-01.txt

Abstract

This document describes the "regex" extension to the Sieve email filtering language. In some cases, it is desirable to have a string matching mechanism which is more powerful than a simple exact match, a substring match or a glob-style wildcard match. The regular expression matching mechanism defined in this draft provides users with much more powerful string matching capabilities.

Change History (to be removed prior to publication as an RFC)

Changes from draft-murchison-sieve-regex-08:

o Updated to XML source.

o Documented interaction with variables.

Changes from draft-ietf-sieve-regex-00:

o Various cleanup and updates.

o Added trial text specifying comparator interactions.

Open Issues (to be removed prior to publication as an RFC)

o The major open issue with this draft is what to do, if anything, about localization/internationalization. Are [IEEE.1003-2.1992] collating sequences and character equivalents sufficient? Should we reference the Unicode technical specification? Should we punt and publish the document as experimental?

o Is the current approach to comparator integration the right one to use?

o Should we allow shorthands such as \\b (word boundary) and \\w (word character)?

o Should we allow backreferences (useful for matching double words, etc.)?

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 24, 2010.

Copyright Notice

Copyright (c) 2010 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 BSD License.

1. Introduction

Sieve [RFC5228] is a language for filtering email messages at or around the time of final delivery. It is designed to be implementable on either a mail client or mail server.

The Sieve base specification defines so-called match types for tests: is, contains, and matches. An "is" test requires an exact match, a "contains" test provides a substring match, and "matches" provides glob-style wildcards. This document describes an extension to the Sieve language that provides a new match type for regular expression comparisons.

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 [RFC2119].

The terms used to describe the various components of the Sieve language are taken from Section 1.1 of [RFC5228].

3. Capability Identifier

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

4. Regex Match Type

When the regex extension is available, commands that support matching may take the optional tagged argument ":regex" to specify that a regular expression match should be performed. The ":regex" match type is subject to the same rules and restrictions as the standard match types defined in [RFC5228].

The "MATCH-TYPE" syntax element defined in [RFC5228] is augmented here as follows:

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

5. Interaction with Sieve comparators

In order to provide for matches between character sets and case insensitivity, Sieve uses the comparators defined in the Internet Application Protocol Collation Registry [RFC5228]. The comparator used by a given test is specified by the :comparator argument.

The interaction between collators and the match types defined in the Sieve base specification is straightforward. Howeer, the nature of regular expressions does not lend itself to this usage for the :regex

match type.

A component of the definition of many collators is a normalization operation. For example, the "i;octet" comparator employs an identity normalization; whereas the "i;ascii-casema" normalizes all lower case ASCII characters to upper case.

The :regex match type only uses the normalization component of the associated comparator. This normalization operation is applied to the key-list argument to the test; the result of that normalization becomes the target of the regular expression comparison. The comparator has no effect on the regular expression pattern or the underlying comparison operation.

It is an error to specify a comparator that has no associated normalization operation in conjunction with a :regex match type.

6. Regular expression comparisions

Implementations MUST support extended regular expressions (EREs) as defined by [IEEE.1003-2.1992]. Any regular expression not defined by [IEEE.1003-2.1992], as well as [IEEE.1003-2.1992] basic regular expressions, word boundaries and backreferences are not supported by this extension. Implementations SHOULD reject regular expressions that are unsupported by this specification as a syntax error.

The following tables provide a brief summary of the regular expressions that MUST be supported. This table is presented here only as a guideline. [IEEE.1003-2.1992] should be used as the definitive reference.

Expression	Pattern
.	Match any single character except newline.
[ ]	Bracket expression. Match any one of the enclosed characters. A hypen (-) indicates a range of consecutive characters.
[^]	Negated bracket expression. Match any one character NOT in the enclosed list. A hypen (-) indicates a range of consecutive characters.
\\	Escape the following special character (match the literal character). Undefined for other characters. NOTE: Unlike [IEEE.1003-2.1992], a double-backslash is required as per section 2.4.2 of [RFC5228].

Table 1: Items to match a single character

Expression	Pattern
[: :]	Character class (alnum, alpha, blank, cntrl, digit, graph, lower, print, punct, space, upper, xdigit).
[= =]	Character equivalents.
[. .]	Collating sequence.

Table 2: Items to be used within a bracket expression (localization)

Expression	Pattern
?	Match zero or one instances.
*	Match zero or more instances.
+	Match one or more instances.
{n,m}	Match any number of instances between n and m (inclusive). {n} matches exactly n instances. {n,} matches n or more instances.

Table 3: Quantifiers - Items to count the preceding regular expression

Expression	Pattern
^	Match the beginning of the line or string.
$	Match the end of the line or string.

Table 4: Anchoring - Items to match positions

Expression	Pattern
|	Alternation. Match either of the separated regular expressions.
( )	Group the enclosed regular expression(s).

Table 5: Other constructs

7. Interaction with Sieve Variables

This extension is compatible with, and may be used in conjunction with the Sieve Variables extension [RFC5229].

7.1. Match variables

A sieve interpreter which supports both "regex" and "variables", MUST set "match variables" (as defined by [RFC5229] section 3.2) whenever the ":regex" match type is used. The list of match variables will contain the strings corresponding to the group operators in the regular expression. The groups are ordered by the position of the opening parenthesis, from left to right. Note that in regular expressions, expansions match as much as possible (greedy matching).

Example:

```
   require ["fileinto", "regex", "variables"];

   if header :regex "List-ID" "<(.*)@" {
       fileinto "lists.${1}"; stop;
   }

   # Imagine the header
   # Subject: [acme-users] [fwd] version 1.0 is out
   if header :regex "Subject" "^[(.*)] (.*)$" {
       # ${1} will hold "acme-users] [fwd"
       stop;
   }
```

7.2. Set modifier :quoteregex

A sieve interpreter which supports both "regex" and "variables", MUST support the optional tagged argument ":quoteregex" for use with the "set" action. The ":quoteregex" modifier is subject to the same rules and restrictions as the standard modifiers defined in [RFC5229] section 4.

For convenience, the "MODIFIER" syntax element defined in [RFC5229] is augmented here as follows:

```
   MODIFIER  =/  ":quoteregex"
```

This modifier adds the necessary quoting to ensure that the expanded text will only match a literal occurrence if used as a parameter to :regex. Every character with special meaning (".", "*", "?", etc.) is prefixed with "\" in the expansion. This modifier has a precedence value of 20 when used with other modifiers.

8. Examples

Example:

```
   require "regex";

   # Try to catch unsolicited email.
   if anyof (
     # if a message is not to me (with optional +detail),
     not address :regex ["to", "cc", "bcc"]
       "me(\\\\+.*)?@company\\\\.com",

     # or the subject is all uppercase (no lowercase)
     header :regex :comparator "i;octet" "subject"
       "^[^[:lower:]]+$" ) {

     discard;    # junk it
   }
```

9. IANA Considerations

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

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


   Capability name: regex
   Capability keyword: regex
   Capability arguments: N/A
   Standards Track/IESG-approved experimental RFC number: this RFC
   Person and email address to contact for further information:
       Kenneth Murchison
       E-Mail: murch@andrew.cmu.edu
```

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

10. Security Considerations

General Sieve security considerations are discussed in [RFC5228]. All of the issues described there also apply to regular expression matching.

It is easy to construct problematic regular expressions that are

computationally infeasible to evaluate. Execution of a Sieve that employs a potentially problematic regular expression, such as "(.*)*", may cause problems ranging from degradation of performance to and outright denial of service. Moreover, determining the computationl complexity associated with evaluating a given regular expression is in general an intractable problem.

For this reason, all implementations MUST take appropriate steps to limit the impact of runaway regular expression evaluation. Implementations MAY restrict the regular expressions users are allowed to specify. Implementations that do not impose such restrictions SHOULD provide a means to abort evaluation of tests using the :regex match type if the operation is taking too long.

11. Normative References

[IEEE.1003-2.1992]
Institute of Electrical and Electronics Engineers, "Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Standard 1003.2, 1992.

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

[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.

Appendix A. Acknowledgments

Most of the text documenting the interaction with Sieve variables was taken from an early draft of Kjetil Homme's Sieve variables specification.

Thanks to Tim Showalter, Alexey Melnikov, Tony Hansen, Phil Pennock, and Jutta Degener for their help with this document.

Authors' Addresses

Kenneth Murchison
Carnegie Mellon University
5000 Forbes Avenue
Cyert Hall 285
Pittsburgh, PA 15213
US

Phone: +1 412 268 2638
Email: murch@andrew.cmu.edu

Ned Freed
Oracle Corporation
800 Royal Oaks
Monrovia, CA 91016-6347
USA

Phone: +1 909 457 4293
Email: ned.freed@mrochek.com