Sieve Email Filtering -- Regular Expression Extension

<draft-murchison-sieve-regex-08.txt>

Status of this Memo

Copyright Notice

Abstract

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 should allow users to isolate just
about any string or address in a message header or envelope.

Table of Contents

0.   Meta-information on this draft

This information is intended to facilitate discussion. It will be
removed when this document leaves the Internet-Draft stage.

0.1.   Discussion

This draft is intended to be an extension to the Sieve mail filtering
language, available from the RFC repository as
<ftp://ftp.ietf.org/rfc/rfc3028.txt>.

This draft and the Sieve language itself are being discussed on the
MTA Filters mailing list at <ietf-mta-filters@imc.org>. Subscription
requests can be sent to <ietf-mta-filters-request@imc.org> (send an
email message with the word "subscribe" in the body). More
information on the mailing list along with a WWW archive of back
messages is available at <http://www.imc.org/ietf-mta-filters/>.

0.2.   Noted Changes Since -07

Updated boilerplate.

0.3.   Open Issues

The major open issue with this draft is what to do, if anything,
about localization/internationalization. Are [POSIX.2] collating
sequences and character equivalents sufficient? Should we reference
the unicode technical specification? Should we punt and publish the
document as experimental?

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

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

Should we integrate with variables, so that $1, $2, ... correspond
to the first, second, ... groups within the regex?

1.   Introduction

This is an extension to the Sieve language defined by [SIEVE] for
comparing strings to regular expressions.

Conventions for notations are as in [SIEVE] section 1.1, including use of [KEYWORDS].

2. **Capability Identifier**

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

3. **Regex Match Type**

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 [SIEVE]. For convenience, the "MATCH-TYPE" syntax element defined in [SIEVE] is augmented here as follows:

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

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
      }
```

The ":regex" match type is compatible with both the "i;octet" and "i;ascii-casemap" comparators and may be used with them.

Implementations MUST support extended regular expressions (EREs) as defined by [POSIX.2]. Any regular expression not defined by [POSIX.2], as well as [POSIX.2] 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 table provides a brief summary of the regular
expressions that MUST be supported. This table is presented here
only as a guideline. [POSIX.2] should be used as the definitive
reference.

```
+------------+------------------------------------------------------+
| Expression | Pattern                                              |
+------------+------------------------------------------------------+
|               Items to match a single character              |
+------------+------------------------------------------------------+
|     .      | 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 [POSIX.2], a double-backslash is        |
|            | required as per section 2.4.2 of [SIEVE].            |
+------------+------------------------------------------------------+
|    Items to be used within a bracket expression (localization)   |
+------------+------------------------------------------------------+
|   [: :]    | Character class (alnum, alpha, blank, cntrl,         |
|            | digit, graph, lower, print, punct, space,            |
|            | upper, xdigit).                                      |
|   [= =]    | Character equivalents.                               |
|   [. .]    | Collating sequence.                                  |
+------------+------------------------------------------------------+
|  Quantifiers - Items to count the preceding regular expression   |
+------------+------------------------------------------------------+
|     ?      | 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.        |
+------------+------------------------------------------------------+
|                 Anchoring - Items to match positions             |
+------------+------------------------------------------------------+
|     ^      | Match the beginning of the line or string.           |
|     $      | Match the end of the line or string.                 |
+------------+------------------------------------------------------+
```

```
+------------+----------------------------------------------------+
| Expression | Pattern                                            |
+------------+----------------------------------------------------+
|                          Other constructs                       |
+------------+----------------------------------------------------+
|    |       | Alternation.  Match either of the separated        |
|            | regular expressions.                               |
|    ( )     | Group the enclosed regular expression(s).          |
+------------+----------------------------------------------------+
```

[4](#). Security Considerations

Security considerations are discussed in [[SIEVE](#)]. It is believed
that this extension doesn't introduce any additional security
concerns.

However, a poor implementation COULD introduce security problems
ranging from degradation of performance to denial of service. If an
implementation uses a third-party regular expression library, that
library should be checked for potentially problematic regular
expressions, such as "(.*)*".

[5](#). IANA Considerations

The following template specifies the IANA registration of the 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
ken@oceana.com

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

6.

   Normative References

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

     [SIEVE] Showalter, T., "Sieve: A Mail Filtering Language", RFC
          3028, January 2001.

     [POSIX.2], "Portable Operating System Interface (POSIX). Part 2,
          Shell and utilities", National Institute of Standards and Tech-
          nology (U.S.).

7.   **Acknowledgments**

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

8.   **Author's Address**

   Kenneth Murchison
   Oceana Matrix Ltd.
   21 Princeton Place
   Orchard Park, NY 14127

   Phone: (716) 662-8973

   EMail: ken@oceana.com

9.

   Intellectual Property Rights

   The IETF takes no position regarding the validity or scope of any
   intellectual property or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; neither does it represent that it has
   made any effort to identify any such rights. Information on the
   IETF's procedures with respect to rights in standards-track and
   standards-related documentation can be found in BCP-11. Copies of
   claims of rights made available for publication and any assurances of
   licenses to be made available, or the result of an attempt made to

obtain a general license or permission for the use of such proprietary
rights by implementers or users of this specification can be obtained
from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary
rights which may cover technology that may be required to practice
this standard. Please address the information to the IETF Executive
Director.

10.

Copyright

y