Sieve -- "body" extension
<**draft-degener-sieve-body-00.txt**>

Status of this memo

   This document is an Internet-Draft and is subject to all
   provisions of Section 10 of RFC2026.

   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/1id-abstracts.html

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

Abstract

   This document defines a new primitive for the "sieve" language
   that tests for the occurrence of one or more strings in the body
   of an e-mail message.

1. **Introduction**

   The proposed "body" test checks for the occurrence of one
   or more strings in the body of an e-mail message.
   Such a test was initially discussed for the [SIEVE] base
   document, but was subsequently removed because it was
   thought to be too costly to implement.

   Nevertheless, several server vendors have implemented
   some form of the "body" test.

   This document reintroduces the "body" test as an extension,
   and specifies it syntax and semantics.

2. Conventions used.

Conventions for notations are as in [SIEVE] section 1.1, including use of [KEYWORDS] and "Syntax:" label for the definition of action and tagged arguments syntax.

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

3. Test body

Syntax:
        "body" [COMPARATOR] [MATCH-TYPE] [BODY-TRANSFORM]
                <key-list: string-list>

The body test matches text in the body of an e-mail message, that is, anything following the first empty line after the header. (The empty line itself, if present, is not considered to be part of the body.)

The COMPARATOR and MATCH-TYPE keyword parameters are defined in [SIEVE]. The BODY-TRANSFORM is a keyword parameter discussed in section 4, below.

If a message consists of a header only, not followed by an empty line, all "body" tests fail, including that for an empty string.

If a message consists of a header followed only by an empty line with no body lines following it, the message is considered to have an empty string as a body.

4. Body Transform

Prior to matching text in a message body, "transformations" can be applied that filter and decode certain parts of the body. These transformations are selected by a "BODY-TRANSFORM" keyword parameter.

Syntax:
          ":raw"
      / ":content" <content-types: string-list>
      / ":text"

4.1 Body Transform ":raw"

The ":raw" transform is intended to match against the undecoded body of a message.

If the specified body-transform is ":raw", the MIME structure of
the body is irrelevant. The implementation MUST NOT remove any
transfer encoding from the message, MUST NOT refuse to filter
messages with syntactic errors (unless the environment it is part
of rejects them outright), and MUST NOT interpret or skip MIME
headers of enclosed body parts.

Example:

```
require "body";

# This will match a message containing the words "MAKE MONEY FAST"
# in body or MIME headers other than the outermost RFC 822 header,
# but will not match a message containing the words in a
# content-transfer-encoded body.

if body :raw :contains "MAKE MONEY FAST" {
        reject;
}
```

4.2 Body Transform ":content_type"

If the body transform is ":content_type", only MIME parts that have
the specified content-types are selected for matching.

If an individual content type contains a '/' (slash), it
specifies a full <type>/<subtype> pair, and matches only
that specific content type. Otherwise, it specifies a
<type> only, and any subtype matches it.

The search for MIME parts is recursive and automatically
descends into multipart and message MIME parts.

MIME parts encoded in a content transfer encoding must be decoded,
and text MIME parts in charsets other than UTF-8 MUST be converted
to UTF-8 prior to the match.

Search expressions MUST NOT match across MIME part boundaries.
MIME headers of the containing text MUST NOT be included in the
data.

Example:
```
require ["body", "fileinto"];

# Save any message with any text MIME part that contains the
# worlds "missile" or "coordinates" in the "secrets" folder.

if body :content_type "text" :contains ["missile", "coordinates"] {
        fileinto "secrets";
```

```
        }

        # Save any message with an audio/mp3 MIME part in
        # the "jukebox" folder.

        if body :content_type "audio/mp3" :contains "" {
                fileinto "jukebox";
        }
```

4.3 Body Transform ":text"

The ":text" body transform matches against the results of
an implementation's best effort at extracting text from a message.

In simple implementations, :text MAY be a macro that stands
for :content_type "text".

Sophisticated implementations MAY strip mark-up from the text
prior to matching, and MAY convert media types other than text
to text prior to matching.

(For example, they may be able to convert proprietary text
editor formats to text or apply optical character recognition
algorithms to image data.)

5. Interaction with Other Sieve Extensions

Any extension that extends the grammar for the COMPARATOR or
MATCH-TYPE nonterminals will also affect the implementation of
"body".

The [REGEX] extension can place a considerable load on a system
when applied to whole bodies of messages, especially when
implemented naively or used maliciously.

6. Security Considerations

The system MUST be sized and restricted in such a manner that
even malicious use of body matching does not deny service to
other users of the host system.

Filters relying on string matches in the raw body of an e-mail
message may be more general than intended. Text matches are no
replacement for a virus or spam filtering system.

7. Acknowledgments

This document has been revised in part based on comments and

discussions that took place on and off the SIEVE mailing list.
Thanks to Cyrus Daboo, Simon Josefsson, Chris Markle, Greg Shapiro,
Tim Showalter, Nigel Swinson, and Dowson Tong for taking the time
to review this draft and make suggestions.

8. Author's Address

Jutta Degener
Sendmail, Inc.
6425 Christie Ave, 4th Floor
Emeryville, CA 94608

Email: jutta@sendmail.com

9. Discussion

This section will be removed when this document leaves the
Internet-Draft stage.

This draft is intended as an extension to the Sieve mail filtering
language. Sieve extensions are 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/>.

9.1 Consensus

A "body" operation is being used for mail filtering.
Some systems implement it in something similar to sieve,
some systems implement a body or x_body operations within sieve.

The implementations do not process the body contents.
They do not strip content-transfer encodings and do not convert
text to UTF-8 prior to comparison.

There is a strong feeling that this behavior is unworthy of
standardization, that body is a "valuable piece of real-estate"
that one should get right, and that users will expect a "body"
test to at least be capable of reaching inside encodings.

9.2 Open Issues

9.2.1 Body Transformations

This document names three possible transformations

(one of which, ":raw", does nothing) to apply to the
body before matching. Are there important others that
are missing? Are three too many? Which ones should
be dropped?

9.2.2 **Default Transformation**

If none of the transformations is specified, what should
be the default? Should there be a default, or should it
be an error?

I'd like there to be a default, so that existing scripts
that use an undocumented "body" extensions just work.

The default should probably be :raw or :text.
It should be ":raw", because that's what current implementations do.
It should be ":text", because that's what we believe users expect.

I'd like this to be implementation-defined, allowing for a
gradual migration from the current behavior to something closer
to user expectations.

Appendices

Appendix A. **References**

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

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

Appendix B. **Full Copyright Statement**