Network Working Group
Request for Comments: 5293
Category: Standards Track

J. Degener
P. Guenther
Sendmail, Inc.

August 2008

Sieve Email Filtering: Editheader Extension

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document defines two new actions for the "Sieve" email filtering language that add and delete email header fields.

1. Introduction

Email header fields are a flexible and easy-to-understand means of communication between email processors. This extension enables sieve scripts to interact with other components that consume or produce header fields by allowing the script to delete and add header fields.

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 [KEYWORDS].

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

The term "header field" is used here as in [IMAIL] to mean a logical line of an email message header.

3. Capability Identifier

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

4. Action addheader

```
Usage: "addheader" [":last"] <field-name: string> <value: string>
```

The addheader action adds a header field to the existing message header. If the field-name is not a valid 7-bit US-ASCII header field name, as described by the [IMAIL] "field-name" nonterminal syntax element, the implementation MUST flag an error. The addheader action does not affect Sieve's implicit keep.

If the specified field value does not match the [IMAIL] "unstructured" nonterminal syntax element or exceeds a length limit set by the implementation, the implementation MUST either flag an error or encode the field using folding white space and the encodings described in [MIME3] or [MIMEPARAM] to be compliant with [IMAIL].

An implementation MAY impose a length limit onto the size of the encoded header field; such a limit MUST NOT be less than 998 characters, not including the terminating CRLF supplied by the implementation.

By default, the header field is inserted at the beginning of the existing message header. If the optional flag ":last" is specified, it is appended at the end.

Example:

```
/* Don't redirect if we already redirected */
if not header :contains "X-Sieve-Filtered"
        ["<kim@job.example.com>", "<kim@home.example.com>"]
{
        addheader "X-Sieve-Filtered" "<kim@job.example.com>";
        redirect "kim@home.example.com";
}
```

5. Action deleteheader

```
Usage: "deleteheader" [":index" <fieldno: number> [":last"]]
             [COMPARATOR] [MATCH-TYPE]
             <field-name: string>
             [<value-patterns: string-list>]
```

By default, the deleteheader action deletes all occurrences of the named header field. The deleteheader action does not affect Sieve's implicit keep.

The field-name is mandatory and always matched as a case-insensitive US-ASCII string. If the field-name is not a valid 7-bit header field name as described by the [IMAIL] "field-name" nonterminal syntax element, the implementation MUST flag an error.

The value-patterns, if specified, restrict which occurrences of the header field are deleted to those whose values match any of the specified value-patterns, the matching being according to the match-type and comparator and performed as if by the "header" test. In particular, leading and trailing whitespace in the field values is ignored. If no value-patterns are specified, then the comparator and match-type options are silently ignored.

If :index <fieldno> is specified, the attempts to match a value are limited to the <fieldno> occurrence of the named header field, beginning at 1, the first named header field. If :last is specified, the count is backwards; 1 denotes the last named header field, 2 the second to last, and so on. The counting happens before the <value-patterns> match, if any. For example:

```
   deleteheader :index 1 :contains "Delivered-To"
                           "bob@example.com";
```

deletes the first "Delivered-To" header field if it contains the string "bob@example.com" (not the first "Delivered-To" field that contains "bob@example.com").

It is not an error if no header fields match the conditions in the deleteheader action or if the :index argument is greater than the number of named header fields.

The implementation MUST flag an error if :last is specified without also specifying :index.

6. Implementation Limitations on Changes

As a matter of local policy, implementations MAY limit which header fields may be deleted and which header fields may be added. However, implementations MUST NOT permit attempts to delete "Received" and "Auto-Submitted" header fields and MUST permit both addition and deletion of the "Subject" header field.

If a script tries to make a change that isn’t permitted, the attempt MUST be silently ignored.

7. Interaction with Other Sieve Extensions

Actions that generate [MDN], [DSN], or similar disposition messages MUST do so using the original, unmodified message header. Similarly, if an error terminates processing of the script, the original message header MUST be used when doing the implicit keep required by Section 2.10.6 of [SIEVE].

All other actions that store, send, or alter the message MUST do so with the current set of header fields. This includes the addheader and deleteheader actions themselves. For example, the following leaves the message unchanged:

```
addheader "X-Hello" "World";
deleteheader :index 1 "X-Hello";
```

Similarly, given a message with three or more "X-Hello" header fields, the following example deletes the first and third of them, not the first and second:

```
deleteheader :index 1 "X-Hello";
deleteheader :index 2 "X-Hello";
```

Tests and actions such as "exists", "header", or "vacation" [VACATION] that examine header fields MUST examine the current state of a header as modified by any actions that have taken place so far.

As an example, the "header" test in the following fragment will always evaluate to true, regardless of whether or not the incoming message contained an "X-Hello" header field:

```
addheader "X-Hello" "World";
if header :contains "X-Hello" "World"
{
        fileinto "international";
}
```

However, if the presence or value of a header field affects how the implementation parses or decodes other parts of the message, then, for the purposes of that parsing or decoding, the implementation MAY ignore some or all changes made to those header fields. For example, in an implementation that supports the [BODY] extension, "body" tests may be unaffected by deleting or adding "Content-Type" or "Content-Transfer-Encoding" header fields. This does not rescind the requirement that changes to those header fields affect direct tests; only the semantic side effects of changes to the fields may be ignored.

For the purpose of weeding out duplicates, a message modified by addheader or deleteheader MUST be considered the same as the original message. For example, in an implementation that obeys the constraint in Section 2.10.3 of [SIEVE] and does not deliver the same message to a folder more than once, the following code fragment

```
keep;
addheader "X-Flavor" "vanilla";
keep;
```

MUST only file one message. It is up to the implementation to pick which of the redundant "fileinto" or "keep" actions is executed, and which ones are ignored.

The "implicit keep" is thought to be executed at the end of the script, after the headers have been modified. (However, a canceled "implicit keep" remains canceled.)

8. 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: editheader
Description:     Adds actions ’addheader’ and ’deleteheader’ that
                 modify the header of the message being processed
RFC number:      RFC 5293
Contact Address: The Sieve discussion list <ietf-mta-filters&imc.org>
```

9. Security Considerations

Someone with write access to a user’s script storage may use this extension to generate headers that a user would otherwise be shielded from (e.g., by a gateway Mail Transport Agent (MTA) that removes them).

This is the first Sieve extension to be standardized that allows alteration of messages being processed by Sieve engines. A Sieve script that uses Sieve tests defined in [SIEVE], the editheader extension, and the redirect action back to the same user can keep some state between different invocations of the same script for the same message. But note that it would not be possible to introduce an infinite loop using any such script, because each iteration adds a new Received header field, so email loop prevention described in [SMTP] will eventually non deliver the message, and because the editheader extension is explicitly prohibited to alter or delete Received header fields (i.e., it can’t interfere with loop prevention).

A sieve filter that removes header fields may unwisely destroy evidence about the path a message has taken.

Any change in message content may interfere with digital signature mechanisms that include the header in the signed material. For example, changes to (or deletion/addition of) header fields included in the "SHOULD be included in the signature" list in Section 5.5 of [DKIM] can invalidate DKIM signatures. This also includes DKIM signatures that guarantee a header field absence.

The editheader extension doesn’t directly affect [IMAIL] header field signatures generated using [SMIME] or [OPENPGP], because these signature schemes include a separate copy of the header fields inside the signed message/rfc822 body part. However, software written to detect differences between the inner (signed) copy of header fields and the outer (modified by editheader) header fields might be affected by changes made by editheader.

Since normal message delivery adds "Received" header fields and other trace fields to the beginning of a message, many such digital signature mechanisms are impervious to headers prefixed to a message, and will work with "addheader" unless :last is used.

Any decision mechanism in a user’s filter that is based on headers is vulnerable to header spoofing. For example, if the user adds an APPROVED header or tag, a malicious sender may add that tag or header themselves. One way to guard against this is to delete or rename any such headers or stamps prior to processing the message.

10. Acknowledgments

Thanks to Eric Allman, Cyrus Daboo, Matthew Elvey, Ned Freed, Arnt Gulbrandsen, Kjetil Torgrim Homme, Simon Josefsson, Will Lee, William Leibzon, Mark E. Mallett, Chris Markle, Alexey Melnikov, Randall Schwartz, Aaron Stone, Nigel Swinson, and Rand Wacker for extensive corrections and suggestions.

11. References

11.1. Normative References

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

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

[MIME3] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[MIMEPARAM] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

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

11.2. Informative References

[BODY] Degener, J. and P. Guenther, "Sieve Email Filtering: Body Extension", RFC 5173, April 2008.

[DKIM] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and M. Thomas, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

[DSN] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[MDN] Hansen, T., Ed., and G. Vaudreuil, Ed., "Message Disposition Notification", RFC 3798, May 2004.

[OPENPGP] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

[SMIME] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[SMTP] Klensin, J., Ed., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[VACATION]   Showalter, T. and N. Freed, Ed., "Sieve Email Filtering: Vacation Extension", RFC 5230, January 2008.

Authors' Addresses

Jutta Degener
5245 College Ave, Suite #127
Oakland, CA 94618

EMail: jutta@pobox.com

Philip Guenther
Sendmail, Inc.
6475 Christie Ave., Ste 350
Emeryville, CA 94608

EMail: guenther@sendmail.com

Full Copyright Statement

Intellectual Property