NAME

header_checks - Postfix built-in content inspection

SYNOPSIS

header_checks = pcre:/etc/postfix/header_checks
mime_header_checks = pcre:/etc/postfix/mime_header_checks
nested_header_checks = pcre:/etc/postfix/nested_header_checks
body_checks = pcre:/etc/postfix/body_checks

milter_header_checks = pcre:/etc/postfix/milter_header_checks

smtp_header_checks = pcre:/etc/postfix/smtp_header_checks
smtp_mime_header_checks = pcre:/etc/postfix/smtp_mime_header_checks
smtp_nested_header_checks = pcre:/etc/postfix/smtp_nested_header_checks
smtp_body_checks = pcre:/etc/postfix/smtp_body_checks

postmap -q "*string***" pcre:/etc/postfix/***filename*
postmap -q - pcre:/etc/postfix/*filename* <*inputfile*

DESCRIPTION

This document describes access control on the content of message headers and message body lines; it is implemented by the Postfix **cleanup(8)** server before mail is queued. See **access(5)** for access control on remote SMTP client information.

Each message header or message body line is compared against a list of patterns. When a match is found the corresponding action is executed, and the matching process is repeated for the next message header or message body line.

Note: message headers are examined one logical header at a time, even when a message header spans multiple lines. Body lines are always examined one line at a time.

For examples, see the EXAMPLES section at the end of this manual page.

Postfix header or body_checks are designed to stop a flood of mail from worms or viruses; they do not decode attachments, and they do not unzip archives. See the documents referenced below in the README FILES section if you need more sophisticated content analysis.

FILTERS WHILE RECEIVING MAIL

Postfix implements the following four built-in content inspection classes while receiving mail:

header_checks (default: empty)
: These are applied to initial message headers (except for the headers that are processed with **mime_header_checks**).

mime_header_checks (default: **$header_checks**)
: These are applied to MIME related message headers only.

  This feature is available in Postfix 2.0 and later.

nested_header_checks (default: **$header_checks**)
: These are applied to message headers of attached email messages (except for the headers that are processed with **mime_header_checks**).

  This feature is available in Postfix 2.0 and later.

body_checks
: These are applied to all other content, including multi-part message boundaries.

  With Postfix versions before 2.0, all content after the initial message headers is treated as body content.

FILTERS AFTER RECEIVING MAIL

Postfix supports a subset of the built-in content inspection classes after the message is received:

milter_header_checks (default: empty)
These are applied to headers that are added with Milter applications.

This feature is available in Postfix 2.7 and later.

FILTERS WHILE DELIVERING MAIL

Postfix supports all four content inspection classes while delivering mail via SMTP.

smtp_header_checks (default: empty)

smtp_mime_header_checks (default: empty)

smtp_nested_header_checks (default: empty)

smtp_body_checks (default: empty)
These features are available in Postfix 2.5 and later.

COMPATIBILITY

With Postfix version 2.2 and earlier specify "**postmap -fq**" to query a table that contains case sensitive patterns. By default, regexp: and pcre: patterns are case insensitive.

TABLE FORMAT

This document assumes that header and body_checks rules are specified in the form of Postfix regular expression lookup tables. Usually the best performance is obtained with **pcre** (Perl Compatible Regular Expression) tables. The **regexp** (POSIX regular expressions) tables are usually slower, but more widely available. Use the command "**postconf -m**" to find out what lookup table types your Postfix system supports.

The general format of Postfix regular expression tables is given below. For a discussion of specific pattern or flags syntax, see **pcre_table(5)** or **regexp_table(5)**, respectively.

/pattern/flags action
When /*pattern*/ matches the input string, execute the corresponding *action*. See below for a list of possible actions.

!/*pattern*/*flags action*
When /*pattern*/ does **not** match the input string, execute the corresponding *action*.

if /*pattern*/*flags*

endif Match the input string against the patterns between **if** and **endif**, if and only if the same input string also matches /*pattern*/. The **if**..**endif** can nest.

Note: do not prepend whitespace to patterns inside **if**..**endif**.

if !/*pattern*/*flags*

endif Match the input string against the patterns between **if** and **endif**, if and only if the same input string does **not** match /*pattern*/. The **if**..**endif** can nest.

blank lines and comments
Empty lines and whitespace-only lines are ignored, as are lines whose first non-whitespace character is a '#'.

multi-line text
A pattern/action line starts with non-whitespace text. A line that starts with whitespace continues a logical line.

TABLE SEARCH ORDER

For each line of message input, the patterns are applied in the order as specified in the table. When a pattern is found that matches the input line, the corresponding action is executed and then the next input line is inspected.

TEXT SUBSTITUTION

Substitution of substrings from the matched expression into the *action* string is possible using the conventional Perl syntax (**$1**, **$2**, etc.). The macros in the result string may need to be written as **${n}** or **$(n)** if

they aren't followed by whitespace.

Note: since negated patterns (those preceded by **!**) return a result when the expression does not match, substitutions are not available for negated patterns.

ACTIONS

Action names are case insensitive. They are shown in upper case for consistency with other Postfix documentation.

DISCARD *optional text...*

Claim successful delivery and silently discard the message. Log the optional text if specified, otherwise log a generic message.

Note: this action disables further header or body_checks inspection of the current message and affects all recipients. To discard only one recipient without discarding the entire message, use the transport(5) table to direct mail to the discard(8) service.

This feature is available in Postfix 2.0 and later.

This feature is not supported with smtp header/body checks.

DUNNO

Pretend that the input line did not match any pattern, and inspect the next input line. This action can be used to shorten the table search.

For backwards compatibility reasons, Postfix also accepts **OK** but it is (and always has been) treated as **DUNNO**.

This feature is available in Postfix 2.1 and later.

FILTER *transport:destination*

After the message is queued, send the entire message through the specified external content filter. The *transport* name specifies the first field of a mail delivery agent definition in master.cf; the syntax of the next-hop *destination* is described in the manual page of the corresponding delivery agent. More information about external content filters is in the Postfix FILTER_README file.

Note 1: do not use $*number* regular expression substitutions for *transport* or *destination* unless you know that the information has a trusted origin.

Note 2: this action overrides the main.cf **content_filter** setting, and affects all recipients of the message. In the case that multiple **FILTER** actions fire, only the last one is executed.

Note 3: the purpose of the FILTER command is to override message routing. To override the recipient's *transport* but not the next-hop *destination*, specify an empty filter *destination* (Postfix 2.7 and later), or specify a *transport:destination* that delivers through a different Postfix instance (Postfix 2.6 and earlier). Other options are using the recipient-dependent **transport_maps** or the sender-dependent **sender_dependent_default_transport_maps** features.

This feature is available in Postfix 2.0 and later.

This feature is not supported with smtp header/body checks.

HOLD *optional text...*

Arrange for the message to be placed on the **hold** queue, and inspect the next input line. The message remains on **hold** until someone either deletes it or releases it for delivery. Log the optional text if specified, otherwise log a generic message.

Mail that is placed on hold can be examined with the **postcat(1)** command, and can be destroyed or released with the **postsuper(1)** command.

Note: use "**postsuper -r**" to release mail that was kept on hold for a significant fraction of **$maximal_queue_lifetime** or **$bounce_queue_lifetime**, or longer. Use "**postsuper -H**" only for mail that will not expire within a few delivery attempts.

Note: this action affects all recipients of the message.

This feature is available in Postfix 2.0 and later.

This feature is not supported with smtp header/body checks.

IGNORE
Delete the current line from the input, and inspect the next input line.

INFO *optional text...*
Log an "info:" record with the *optional text...* (or log a generic text), and inspect the next input line. This action is useful for routine logging or for debugging.

This feature is available in Postfix 2.8 and later.

PREPEND *text...*
Prepend one line with the specified text, and inspect the next input line.

Notes:

- The prepended text is output on a separate line, immediately before the input that triggered the **PREPEND** action.
- The prepended text is not considered part of the input stream: it is not subject to header/body checks or address rewriting, and it does not affect the way that Postfix adds missing message headers.
- When prepending text before a message header line, the prepended text must begin with a valid message header label.
- This action cannot be used to prepend multi-line text.

This feature is available in Postfix 2.1 and later.

This feature is not supported with milter_header_checks.

REDIRECT *user@domain*
Write a message redirection request to the queue file, and inspect the next input line. After the message is queued, it will be sent to the specified address instead of the intended recipient(s).

Note: this action overrides the **FILTER** action, and affects all recipients of the message. If multiple **REDIRECT** actions fire, only the last one is executed.

This feature is available in Postfix 2.1 and later.

This feature is not supported with smtp header/body checks.

REPLACE *text...*
Replace the current line with the specified text, and inspect the next input line.

This feature is available in Postfix 2.2 and later. The description below applies to Postfix 2.2.2 and later.

Notes:

- When replacing a message header line, the replacement text must begin with a valid header label.
- The replaced text remains part of the input stream. Unlike the result from the **PREPEND** action, a replaced message header may be subject to address rewriting and may affect the way that Postfix adds missing message headers.

REJECT *optional text...*
Reject the entire message. Reply with *optional text...* when the optional text is specified, otherwise reply with a generic error message.

Note: this action disables further header or body_checks inspection of the current message and affects all recipients.

Postfix version 2.3 and later support enhanced status codes. When no code is specified at the beginning of *optional text...*, Postfix inserts a default enhanced status code of "5.7.1".

This feature is not supported with smtp header/body checks.

WARN *optional text...*

Log a "warning:" record with the *optional text...* (or log a generic text), and inspect the next input line. This action is useful for debugging and for testing a pattern before applying more drastic actions.

BUGS

Empty lines never match, because some map types mis-behave when given a zero-length search string. This limitation may be removed for regular expression tables in a future release.

Many people overlook the main limitations of header and body_checks rules.

- These rules operate on one logical message header or one body line at a time. A decision made for one line is not carried over to the next line.
- If text in the message body is encoded (RFC 2045) then the rules need to be specified for the encoded form.
- Likewise, when message headers are encoded (RFC 2047) then the rules need to be specified for the encoded form.

Message headers added by the **cleanup(8)** daemon itself are excluded from inspection. Examples of such message headers are **From:**, **To:**, **Message-ID:**, **Date:**.

Message headers deleted by the **cleanup(8)** daemon will be examined before they are deleted. Examples are: **Bcc:**, **Content-Length:**, **Return-Path:**.

CONFIGURATION PARAMETERS

body_checks

Lookup tables with content filter rules for message body lines. These filters see one physical line at a time, in chunks of at most **$line_length_limit** bytes.

body_checks_size_limit

The amount of content per message body segment (attachment) that is subjected to **$body_checks** filtering.

header_checks

mime_header_checks (default: **$header_checks**)

nested_header_checks (default: **$header_checks**)

Lookup tables with content filter rules for message header lines: respectively, these are applied to the initial message headers (not including MIME headers), to the MIME headers anywhere in the message, and to the initial headers of attached messages.

Note: these filters see one logical message header at a time, even when a message header spans multiple lines. Message headers that are longer than **$header_size_limit** characters are truncated.

disable_mime_input_processing

While receiving mail, give no special treatment to MIME related message headers; all text after the initial message headers is considered to be part of the message body. This means that **header_checks** is applied to all the initial message headers, and that **body_checks** is applied to the remainder of the message.

Note: when used in this manner, **body_checks** will process a multi-line message header one line at a time.

EXAMPLES

Header pattern to block attachments with bad file name extensions. For convenience, the PCRE /x flag is specified, so that there is no need to collapse the pattern into a single line of text. The purpose of the [[:xdigit:]] sub-expressions is to recognize Windows CLSID strings.

/etc/postfix/main.cf:
header_checks = pcre:/etc/postfix/header_checks.pcre

/etc/postfix/header_checks.pcre:
/^Content-(Disposition|Type).*name\s*=\s*"?(.*(\.|=2E)(
ade|adp|asp|bas|bat|chm|cmd|com|cpl|crt|dll|exe|
hlp|ht[at]|
inf|ins|isp|jse?|lnk|md[betw]|ms[cipt]|nws|
\{[[:xdigit:]]{8}(?:-[[:xdigit:]]{4}){3}-[[:xdigit:]]{12}\}|
ops|pcd|pif|prf|reg|sc[frt]|sh[bsm]|swf|
vb[esx]?|vxd|ws[cfh]))(\?=)?"?\s*(;|$)/x
REJECT Attachment name "$2" may not end with ".$4"

Body pattern to stop a specific HTML browser vulnerability exploit.

/etc/postfix/main.cf:
body_checks = regexp:/etc/postfix/body_checks

/etc/postfix/body_checks:
/^<iframe src=(3D)?cid:.* height=(3D)?0 width=(3D)?0>$/
REJECT IFRAME vulnerability exploit

SEE ALSO

cleanup(8),
canonicalize and enqueue Postfix message
pcre_table(5),
format of PCRE lookup tables
regexp_table(5),
format of POSIX regular expression tables
postconf(1),
Postfix configuration utility
postmap(1),
Postfix lookup table management
postsuper(1),
Postfix janitor
postcat(1),
show Postfix queue file contents
RFC 2045, base64 and quoted-printable encoding rules
RFC 2047, message header encoding for non-ASCII text

README FILES

Use "**postconf readme_directory**" or "**postconf html_directory**" to locate this information.
DATABASE_README, Postfix lookup table overview
CONTENT_INSPECTION_README, Postfix content inspection overview
BUILTIN_FILTER_README, Postfix built-in content inspection
BACKSCATTER_README, blocking returned forged mail

LICENSE

AUTHOR(S)

Wietse Venema
IBM T.J. Watson Research
P.O. Box 704
Yorktown Heights, NY 10598, USA