

filtering mail with php-milter



Felix Schwenk

Harald Radi

Content



- Introduction
- Motivation
- Components
- Setup and Configuration
- php-milter SAPI
- Example
- Outlook

Introduction – What is php-milter



- php-milter
 - SAPI
 - runs as a standalone daemon
 - talks to Sendmail at runtime via UNIX sockets or TCP connections
 - delegates Milter callback functions to a PHP script
- Milter
 - mail filter API provided by Sendmail mailer daemon

Milter – Mail Filter



- callback for every stage of SMTP session
 - HELO, MAIL, RCPT, ...
- check on message or connection
- perform actions
 - continue, reject, accept, discard

Motivation



- increasing demand for filtering mail
 - spam (Unsolicited Commercial Email)
 - content checking
 - sender-address checking
 - virus filtering
 - content control
 - adding disclaimer to outbound mail

Components



- recent version of Sendmail mailer daemon
 - Mail Filter API → Milter
- PHP SAPI
 - php-milter
 - currently available via cvs
 - part of next PHP release
 - runs as standalone daemon

Setup and Configuration



- Sendmail
- PHP4
- GNU/Linux

Setup - Sendmail

- get recent version of Sendmail
- modify devtools/Site/site.config.m4

```
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')dnl
APPENDDEF(`conf_libmilter_ENVDEF', `-
D_FFR_MILTER_ROOT_UNSAFE')dnl
```

- start build process

```
./Build
make install
```

Setup – PHP4

- php-milter only available via cvs

```
cv$ -d pserver:cvsread@cvs.php.net:/repository login
cv$ -d pserver:cvsread@cvs.php.net:/repository co -r
    PHP_4_3 php4
cv$ update -d php4/sapi/milter
```

- more detailed info
[http://www.php.net/anoncv\\$.php](http://www.php.net/anoncv$.php)

Setup – PHP4

- build and configure php4
- detailed info:
<http://www.php.net/install.configure>

```
./buildconf  
./configure --with-milter  
make  
make install
```

Configuration - Filters



- INPUT_MAIL_FILTER macro in sendmail.mc
- usually in /etc/mail folder
- detailed info: <http://www.sendmail.org>
- Filters communicate using
 - unix-domain sockets
 - IP sockets
- timeout
- order in which filters are applied

Configuration - Filters

```
define( `MILTER' , `1' )
INPUT_MAIL_FILTER( `php-milter1' ,
    `S=local:/var/milter/php-milter1.sock,
    F=R' )
INPUT_MAIL_FILTER( `php-milter2' ,
    `S=local:/var/milter/php-milter2.sock,
    F=T, T=S:10m;R:10m;E:10m' )
INPUT_MAIL_FILTER( `php-milter3' ,
    `S=inet:999@localhost' )
define( `confINPUT_MAIL_FILTER' , `php-
    milter2, php-milter1, php-milter3' )
```

Configuration - Filters

- flags F=
- fallback action when filter unavailable

Flag	Meaning
R	Reject connection if filter unavailable
T	Temporary fail connection if filter unavailable

Configuration - Filters

- override default timeouts using T=

Flag	Meaning
S	Timeout for sending information from the MTA to a filter. (default: 10s)
R	Timeout for reading reply from the filter. (default: 10s)
E	Overall timeout between sending end-of-message to filter and waiting for the final acknowledgement. (default: 5m)

Configuration - Startup



- `php-milter -D -p /var/milter/php-milter1.sock milter.php`
- more info: `php-milter -h`
- can filters be changed without rerunning `php-milter`?
 - yes if `milter_init` needs not to be called again
 - no otherwise

milter sapi



- Init Flags
- Callbacks
- Message Modification Functions
- Data Modification Functions
- Return Values

miter sapi - callbacks

Function	Description
miter_init	Initialize filter.
miter_connect	connection info
miter_helo	SMTP HELO/EHLO command
miter_envrcpt	envelope recipient
miter_header	header
miter_eoh	end of header
miter_body	body
miter_eom	end of message
miter_abort	message aborted
miter_close	connection cleanup

milter sapi – smfi flags

Flag	Description
SMFIF_ADDHDRS	This filter may add headers.
SMFIF_CHGHDRS	This filter may change and/or delete headers.
SMFIF_CHGBODY	This filter may replace the body during filtering. (significant performance impact)
SMFIF_ADDRCPT	This filter may add recipients to the message.
SMFIF_DELRCPT	This filter may remove recipients from the message.

milter_sapi – message modification functions

- can only be called within `milter_eoh`
- appropriate `smfi` flag must be set
- return `true` or `false`

Function	Description
<code>smfi_addheader(string headerf, string headerv)</code>	Add a header to the message.
<code>smfi_chgheader(string headerf, string headerv)</code>	Change or delete a header.

milter sapi – message modification functions II


Function	Description
<code>smfi_addrcpt(string rcpt)</code>	Add a recipient to the envelope.
<code>smfi_delrcpt(string rcpt)</code>	Delete a recipient form the envelope.
<code>smfi_replacebody(string body)</code>	Replace the body of the message.

mlter sapi – data access functions

- can be called from any callback

Function	Description
smfi_getsymval(string macro)	Returns the value of the given macro or NULL if the macro is not defined.
smfi_setreply(string rcode, string xcode, string message)	Directly set the SMTP error reply code for this connection.

milter sapi – return values



- recipient-oriented – affects a single message recipient
- message-oriented – affects a single message
- connection-oriented – affects an entire connection

milter sapi – return values II

Return value	Description
SMFIS_CONTINUE	Default. Continue processing.
SMFIS_REJECT	<p>For a connection-oriented routine, reject this connection; call <code>milter_close</code>.</p> <p>For a message-oriented routine (except <code>milter_eom</code> or <code>milter_abort</code>), reject this message.</p> <p>For a recipient-oriented routine, reject the current recipient (but continue the current message).</p>
SMFIS_DISCARD	<p>For a message- or recipient-oriented routine, accept this message, but silently discard it.</p> <p>Should not be returned by a connection-oriented routine.</p>

milter sapi – return values III

SMFIS_ACCEPT	<p>For a connection-oriented routine, accept this connection without further filter processing; call <code>milter_close</code>.</p> <p>For a message- or recipient-oriented routine, accept this message without further filtering.</p>
SMFIS_TEMPFAIL	<p>Return a temporary failure.</p> <p>For message-oriented routine, except <code>milter_envfrom</code>, fail for this message.</p> <p>For a connection-oriented routine, fail for this connection, call <code>milter_close</code>.</p> <p>For a recipient-oriented routine, only fail for the current recipient; continue message processing</p>

Example



- all callbacks used
- available at
<http://cvs.php.net/co.php/php4/sapi/milter/milter.php>
- purpose: logging

Example

```
function milter_log($msg)
{
    $GLOBALS['log'] = fopen("/tmp/milter.log", "a");
    fwrite($GLOBALS['log'], date("[H:i:s d.m.Y]") .
"\t{$msg}\n");
    fclose($GLOBALS['log']);
}
```

```
function milter_init() {
    milter_log("-- startup --");
    milter_log("milter_init()");
    smfi_setflags(SMFIIF_ADDHDRS);
}
```

Example

```
function milter_connect($connect)
{
    milter_log("milter_connect('$connect')");
}
```

- called for every SMTP connection

```
function milter_helo($helo)
{
    milter_log("milter_helo('$helo')");
}
```

- called whenever client sends a HELO/EHLO command

Example

```
function milter_envfrom($args)
{
    milter_log("milter_envfrom(args[])");
    foreach ($args as $ix => $arg) {
        milter_log("\targs[$ix] = $arg");
    }
}
```

- called at beginning of each message
- args[0] is sender address
- later arguments are ESMTP arguments

Example

```
function milter_header($header, $value)
{
    milter_log("milter_header('$header', '$value')");
}
```

- zero or more times between milter_envrcpt and milter_eoh
- once per messageheader

```
function milter_eoh()
{
    milter_log("milter_eoh()");
}
```

- called once after all headers sent and processed

Example

```
function milter_body($bodypart)
{
    milter_log("milter_body('$bodypart')");
}
```

- zero or more times between milter_eoh and milter_eom
- significantly impacts filter performance
- later filters will see body changes made by earlier ones

Example

```
function milter_eom()  
{  
    milter_log("milter_eom()");  
    /* add PHP header to the message */  
    smfi_addheader("X-PHP", phpversion());  
}
```

- once per message
- most api functions only work within this callback

Example

```
function milter_abort()  
{  
    milter_log("milter_abort()");  
}
```

- called at any time during message processing

```
function milter_close()  
{  
    milter_log("milter_close()");  
}
```

- called once at end of connection

Outlook



- testing
- benchmarking
- some memory leaks