

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

INTRODUCTION

You must have a working MailScanner set-up and have running copies of MySQL, Apache, and PHP with MySQL and GD support. For MailScanner to log to a MySQL database you need Perl-DBI and Perl-DBD-MySQL installed. Newer version of MailWatch (version 1.0.3) work with all versions of DBD-mysql

For MailWatch version 0.5.1 and earlier, MailWatch only works with DBD-mysql-2.1028, and DBD-mysql-2.1028 only works with version of MySQL Database less than 4.1. Thus, version 4.0.x must be used for MailWatch. You can obtain version 2.1028 of Perl-DBD-MySQL from here:

<http://search.cpan.org/CPAN/authors/id/J/JW/JWIED/DBD-mysql-2.1028.tar.gz>

To Install DBD-mysql-2.1028, the following must be executed exactly as shown:

1. `tar -xzf DBD-mysql-2.1028`
2. `cd DBD-mysql-2.1028`
3. `perl Makefile.PL -cflags=-I/usr/local/src/mysql/include --libs=-L/usr/local/src/mysql/lib -lmysqlclient`
4. `make`
5. `make test`
6. `make install`

PHP

PHP should have the following set in php.ini:

- `short_open_tag = On`
- `safe_mode = Off`
- `register_globals = Off`
- `magic_quotes_gpc = On`
- `magic_quotes_runtime = Off`
- `session.auto_start = 0`

INSTALLATION

All commands below should be run as the 'root'. It is assumed the file has been unpacked to /usr/local/src.

STEP 1: Create MySQL Database & Tables

```
cd /usr/local/src/mailwatch  
mysql -u root -p < create.sql
```

NOTE: The "create.sql" script that comes with MailWatch Version 1.0.3 appears to have a problem with the "users" table. Quite frankly, the "users" table doesn't work. Thus, I have used the "users" table creation from the "create.sql" script that comes with MailWatch Version 0.5.1. In fact, I have taken this a bit further and written my own "create.sql" script and called it "create_mailwatch.sql". This script automates all of the database related commands found in the following steps. Therefore, when this script is used, these listed commands below do not need to be executed. A copy of the script code can be found in Appendix C.

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

STEP 2: Create MySQL User

```
mysql -u root -p
mysql> GRANT ALL ON mailscanner.* TO mailwatch@localhost IDENTIFIED BY 'mailwatch'.
Mysql> GRANT FILE ON *.* TO mailwatch@localhost IDENTIFIED BY 'mailwatch';
mysql> FLUSH PRIVILEGES;
```

STEP 3: Edit MailWatch.pm

Edit "MailWatch.pm" and change the \$db_user and \$db_pass values accordingly. Then, either Move or Copy the modified "MailWatch.pm" file to the /usr/lib/MailScanner/MailScanner/CustomFunctions directory:

```
my($db_name) = "mailscanner";
my($db_host) = "localhost";
my($db_user) = "mailwatch";
my($db_pass) = "mailwatch";
```

```
mv MailWatch.pm /usr/lib/MailScanner/MailScanner/CustomFunctions
```

NOTE: For older versions of MailWatch (0.5.1), you also need to edit "CustomConfig.pm" and add the following to the top of the file just beneath the line that starts '\$VERSION = substr...': **require 'MailScanner/MailWatch.pm';**

STEP 4: Create a MailWatch Web user

Log on to "mailscanner" database as user mailwatch

```
mysql mailscanner -u mailwatch -p
Enter password: *****
USE mailwatch
INSERT INTO users VALUES ('mailwatch',md5('mailwatch'),'Mail Watch','A');
```

NOTE: This will probably fail if you use the "create.sql" script provide with MailWatch Version 1.0.3 (See NOTE in Step 1).

STEP 5: Install & Configure MailWatch

1. Move the entire mailscanner directory to the web server's root:

```
# mv -R mailscanner /var/www/html/
```

2. Check the permissions of /var/www/html/mailscanner/images and /var/www/html/images/cache. They should be ug+rwx and owned by root and in the same group as the web server user (nogroup).

```
# chown root:nogroup images
# chmod ug+rwx images
# chown root:nogroup images/cache
# chmod ug+rwx images/cache
```

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

3. Create “conf.php” by copying conf.php.example and edit the values to suit, you will need to set DB_USER and DB_PASS to the MySQL user and password that you created in Step 2. These should also match those same values in “MailWatch.pm”.

```
cp conf.php.example conf.php
```

```
// Database settings
define(DB_TYPE, 'mysql');
define(DB_USER, 'mailwatch');
define(DB_PASS, 'mailwatch');
define(DB_HOST, 'localhost');
define(DB_NAME, 'mailscanner');
```

NOTE: With MailWatch 1.0 and latter versions, you can use the quarantine more effectively when used with MailScanner version 4.43. Some code was added for MailWatch to keep track of messages quarantined by using a flag in the maillog table. This means that MailWatch 1.0 is much faster when you have a large quarantine directory. The new quarantine report requires the use of the new functionality, so you must upgrade if you want to run this.

The new quarantine flag is not used by default. If you have MailScanner Verions 4.43 or later, you can activate the new functionality by setting QUARANTINE_USE_FLAG to true in “conf.php”. If you do this, you must disable the “clean.quarantine” script supplied by MailScanner and use the new “quarantine_maint.php” script in the tools directory instead.

To clean the quarantine, set “QUARANTINE_DAYS_TO_KEEP” in “conf.php” and run “quarantine_maint –clean”. This should then be run daily from cron.

STEP 6: Setup MailScanner

1. Stop MailScanner: sh /etc/init.d/MailScanner stop
2. Next edit /etc/MailScanner/MailScanner.conf and make sure that the following options are set:

```
Always Looked Up Last = &MailWatchLogging
Detailed Spam Report = yes
Quarantine Whole Message = yes
Quarantine Whole Message As Queue Files = no
Include Scores In SpamAssassin Report = yes
Quarantine User = root
Quarantine Group = nogroup (same group as web server group)
Quarantine Permissions = 0660
```

3. Spam Actions and High Scoring Spam Actions should also have “store” as one of the keywords if you want to quarantine items for learning/viewing in MailWatch.
4. **(Optional)** To integrate Black & White listing into MailWatch, copy the “SQLBlackWhiteList.pm” file to /usr/lib/MailScanner/MailScanner/CustomFunctions directory. Edit the

MAILWATCH INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

“SQLBlackWhiteList.pm” file and change the connection string in the “CreateList” subroutine to match that which was set in “MailWatch.pm”. Finally, in “MailScanner.conf”, set these parameters as follows:

```
Is Definitely Not Spam = &SQKWhitelist  
Is Definitely Spam = &SQKBlacklist
```

5. Move the Bayesian Databases to /etc/MailScanner/bays and set-up permissions (root:nogroup) (skip this if you don't use bayes).
6. Edit /etc/MailScanner/spam.assassin.prefs.conf and set:

```
auto_whitelist_path           /etc/MailScanner/bays/bays/auto-whitelist  
auto_whitelist_file_mode     0660  
bayes_path                   /etc/MailScanner/bays/bays  
bayes_file_mode             0660
```

6. Create the 'new' bayes directory, make the directory owned by the same group as the web server user and make the directory setgid:

```
mkdir /etc/MailScanner/bayes  
chown root:nogroup /etc/MailScanner/bayes  
chmod g+rws /etc/MailScanner/bayes
```

7. Copy the existing bayes databases and set the permissions:

```
cp /root/.spamassassin/bayes_* /etc/MailScanner/bayes  
chown root:nogroup /etc/MailScanner/bayes/bayes_*  
chmod g+rw /etc/MailScanner/bayes/bayes_*
```

8. Test SpamAssassin to make sure that it is using the new databases correctly:

```
spamassassin -D -p /etc/MailScanner/spam.assassin.prefs.conf --lint
```

9. Start MailScanner up again: sh /etc/init.d/MailScanner start && tail -f /var/log/mail/mail
You now have MailScanner logging to MySQL.

STEP 7: Test the MailWatch Interface

1. Point your browser to <http://<hostname>/mailscanner/>. You should be prompted for a username and password. Enter the details of the MailWatch web user that you created earlier, and you should see a list of the last 50 messages processed by MailScanner.
2. Update the SpamAssassin Rules table. MailWatch keeps a list of all the SpamAssassin rules and descriptions which are displayed on the “Message Detail” page. To show the descriptions, you need to run the updater every time you add new rules or upgrade SpamAssassin.
3. Click on the “Other” menu, select “Update SpamAssassin Rule Descriptions” and click “Run Now”.

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

STEP 8: Setup Mail Queue Watcher (Optional)

1. You can get MailWatch to watch and display your sendmail queue directories. All you need to do is copy mailq.php (from the root of the mailwatch tarball - not from the mailscheduler directory - they are different!) to /usr/local/bin and set-up a cron-job to run it.
2. Edit mailq.php first to change the require line to point to the location of functions.php, then:

```
cp mailq.php /usr/local/bin
crontab -e
0,5,10,15,20,25,30,35,40,45,50,55 * * * * root /usr/local/bin/mailq.php
```

STEP 9: Setup Sendmail Relay Log Watcher (Optional)

1. You can get MailWatch to watch your sendmail logs and store all message relay information which is then displayed on the "Message Detail" page which helps debugging and makes it easy for a Helpdesk to actually see where a message was delivered to by the MTA and what the response back was.

```
cp tools/sendmail_relay.php /usr/local/bin
nohup /usr/local/bin/sendmail_relay.php 2>&1 > /dev/null &
```

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

APPENDIX A: CONFIGURE MAILSCANNER TO LOG TO A REMOTE MYSQL DATABASE

This document presumes that you will have a server acting as a database with PHP and MySQL installed on it, and multiple MailScanner gateways logging to the database server.

1. Follow steps 1 to 6 from the INSTALL instructions above on the database server.

2. Create a mailscanner user and password on the database server:

```
mysql mailscanner
GRANT ALL ON mailscanner.* TO mailscanner IDENTIFIED BY 'mailscanner';
FLUSH PRIVILEGES;
```

3. On each MailScanner gateway, you'll need to make sure that the mysql client, perl, perl DBI and perl DBD-MySQL, are installed:

```
rpm -qa | grep "mysql"
mysql-3.23.54a-11
```

```
rpm -qa | grep "perl-DB"
perl-DBD-MySQL-2.1021-3
perl-DBI-1.32-5
```

4. From one of the MailScanner gateways, verify you can connect to the db:

```
mysql mailscanner -u mailscanner -h <db_hostname> -p
Enter password: *****
```

If you get a mysql> prompt, you can connect correctly (enter \q to quit).

5. On each MailScanner gateway, do steps 7 to 10 from the INSTALL instructions above. For step 7, you will also need to edit CustomConfig.pm and change the \$db_host, \$db_user and \$db_pass variables to your local settings.

6. To complete the set-up run steps 11 to 15 on the database server.

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

APPENDIX B: PER-USER FILTERING

MailWatch for MailScanner 0.5 has introduced the preliminary support for per-user filters. This allows an administrator to create MailWatch users that can either see everything, limit views to certain domains, certain e-mail addresses or to a particular regular expression.

In Version 0.5, you have to set everything up manually using SQL. In later versions of MailWatch, you will be able to create users using the Web Interface. The "users" table now contains an extra column named "type". This can have one of four values:

- 'A' - Administrator (can view everything, any filters are ignored)
- 'D' - Domain Admin (can view domains)
- 'U' - User (can view specific addresses)
- 'R' - Regexp (can view items matching regexp)

Next there is a new table called "user_filters". This has four columns:

- username - this should match the user from the 'users' table.
- filter - this is the text of the filter e.g. 'smf@f2s.com' or 'domain.com'.
- verify_key - this is for future use, it should be set to md5(rand()).
- active - this can be set to 'Y' or 'N', if 'N' the rule is ignored.

You can create as many rules per user as you like, they will be OR'd together and AND'd for each "from_address" and "to_address" for a match.

EXAMPLES

To create an administrator:

```
INSERT INTO users VALUES ('<username>',md5('<password>'),'<fullname>','A');
```

To create a domain admin for the domains 'fsl.com' and 'f2s.com':

```
INSERT INTO users VALUES ('<username>',md5('<password>'),'<fullname>','D');  
INSERT INTO user_filters VALUES ('<username>','fsl.com',md5(rand()),'Y');  
INSERT INTO user_filters VALUES ('<username>','f2s.com',md5(rand()),'Y');
```

To create a user that can view the addresses 'foo@bar.com' and 'foo@bar.co.uk':

```
INSERT INTO users VALUES ('<username>',md5('<password>'),'<fullname>','U');  
INSERT INTO user_filters VALUES ('<username>','foo@bar.com',md5(rand()),'Y');  
INSERT INTO user_filters VALUES ('<username>','foo@bar.co.uk',md5(rand()),'Y');
```

To create a view against a regular expression:

```
INSERT INTO users VALUES ('<username>',md5('<password>'),'<fullname>','R');  
INSERT INTO user_filters VALUES ('<username>','<regexp>',md5(rand()),'Y');
```

MAILWATCH INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

APPENDIX C: create_mailwatch.sql

```
-- #####
-- # /usr/local/src/mailwatch/create_mailwatch.sql
-- #####
-- #####
-- # Usage: mysql -u root -p < /usr/local/src/mailwatch/create_mailwatch.sql
-- # Usage: mysql -u root -p mailscanner < /usr/local/src/mailwatch/create_mailwatch.sql
-- #####
-- #####
-- # CREATE MAILSCANNER DATABASE & DATABASE USERS
-- #####
DROP DATABASE mailscanner;
CREATE DATABASE mailscanner;
GRANT ALL ON mailscanner.* TO mailwatch@localhost IDENTIFIED BY 'mailwatch';
GRANT FILE ON *.* TO mailwatch@localhost IDENTIFIED BY 'mailwatch';
FLUSH PRIVILEGES;
SET PASSWORD FOR mailwatch@localhost=OLD_PASSWORD('mailwatch');
USE mailscanner;

-- #####
-- # CREATE TABLE AUDIT_LOG
-- #####
CREATE TABLE audit_log (
  timestamp timestamp(14) NOT NULL,
  user varchar(20) NOT NULL default "",
  ip_address varchar(15) NOT NULL default "",
  action text NOT NULL
) TYPE=MyISAM;

-- #####
-- # CREATE TABLE BLACKLIST
-- #####
CREATE TABLE blacklist (
  id int(11) NOT NULL auto_increment,
  to_address text,
  to_domain text,
  from_address text,
  PRIMARY KEY (id),
  UNIQUE KEY blacklist_uniq (to_address(100),from_address(100))
) TYPE=MyISAM;

-- #####
-- # CREATE TABLE GEOIP_COUNTRY
-- #####
CREATE TABLE geoip_country (
  begin_ip varchar(15) default NULL,
  end_ip varchar(15) default NULL,
  begin_num bigint(20) default NULL,
  end_num bigint(20) default NULL,
  iso_country_code char(2) default NULL,
  country text,
  KEY geoip_country_begin (begin_num),
  KEY geoip_country_end (end_num)
) TYPE=MyISAM;

-- #####
-- # CREATE TABLE INQ
-- #####
CREATE TABLE inq (
  id text,
  cdate date default NULL,
  ctime time default NULL,
  from_address text,
```

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

```
to_address text,  
subject text,  
message text,  
size text,  
priority text,  
attempts text,  
lastattempt text,  
hostname text,  
KEY inq_hostname (hostname(50))  
) TYPE=MyISAM;
```

```
-- #####  
-- # CREATE TABLE MAILLOG  
-- #####  
CREATE TABLE maillog (  
  timestamp timestamp(14) NOT NULL,  
  id text,  
  size bigint(20) default '0',  
  from_address text,  
  from_domain text,  
  to_address text,  
  to_domain text,  
  subject text,  
  clientip text,  
  archive text,  
  isspam tinyint(1) default '0',  
  ishighspam tinyint(1) default '0',  
  issaspam tinyint(1) default '0',  
  isrbldspam tinyint(1) default '0',  
  isfp tinyint(1) default '0',  
  isfn tinyint(1) default '0',  
  spamwhitelisted tinyint(1) default '0',  
  spamblacklisted tinyint(1) default '0',  
  sascore decimal(7,2) default '0.00',  
  spamreport text,  
  virusinfected tinyint(1) default '0',  
  nameinfected tinyint(1) default '0',  
  otherinfected tinyint(1) default '0',  
  report text,  
  ismcp tinyint(1) default '0',  
  ishighmcp tinyint(1) default '0',  
  issamcp tinyint(1) default '0',  
  mcpwhitelisted tinyint(1) default '0',  
  mcpblacklisted tinyint(1) default '0',  
  mcpsascore decimal(7,2) default '0.00',  
  mcpreport text,  
  hostname text,  
  date date default NULL,  
  time time default NULL,  
  headers text,  
  quarantined tinyint(1) default '0',  
  KEY maillog_datetime_idx (date,time),  
  KEY maillog_id_idx (id(20)),  
  KEY maillog_clientip_idx (clientip(20)),  
  KEY maillog_from_idx (from_address(200)),  
  KEY maillog_to_idx (to_address(200)),  
  KEY maillog_host (hostname(30)),  
  KEY from_domain_idx (from_domain(50)),  
  KEY to_domain_idx (to_domain(50)),  
  KEY maillog_quarantined (quarantined)  
) TYPE=MyISAM;
```

```
-- #####  
-- # CREATE TABLE MCP_RULES  
-- #####
```

MAILWATCH

INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

```
CREATE TABLE mcp_rules (  
  rule char(100) NOT NULL default "",  
  rule_desc char(200) NOT NULL default "",  
  PRIMARY KEY (rule)  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE MATALOG  
-- #####  
CREATE TABLE mtaolog (  
  timestamp datetime default NULL,  
  host text,  
  type text,  
  msg_id varchar(20) default NULL,  
  relay text,  
  dsn text,  
  status text,  
  delay time default NULL,  
  UNIQUE KEY mtaolog_uniq (timestamp,host(10),type(10),msg_id,relay(20)),  
  KEY mtaolog_timestamp (timestamp),  
  KEY mtaolog_type (type(10))  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE OUTQ  
-- #####  
CREATE TABLE outq (  
  id text,  
  cdate date default NULL,  
  ctime time default NULL,  
  from_address text,  
  to_address text,  
  subject text,  
  message text,  
  size text,  
  priority text,  
  attempts text,  
  lastattempt text,  
  hostname text,  
  KEY outq_hostname (hostname(50))  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE SA_RULES  
-- #####  
CREATE TABLE sa_rules (  
  rule varchar(100) NOT NULL default "",  
  rule_desc varchar(200) NOT NULL default "",  
  PRIMARY KEY (rule)  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE SAVED_FILTERS  
-- #####  
CREATE TABLE saved_filters (  
  name text NOT NULL,  
  col text NOT NULL,  
  operator text NOT NULL,  
  value text NOT NULL,  
  username text NOT NULL,  
  UNIQUE KEY unique_filters (name(20),col(20),operator(20),value(20),username(20))  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE SPAMSCORES
```

MAILWATCH INSTALLATION PROCEDURE

By Mark E. Donaldson
(Adapted From the MailWatch Documentation)

```
-- #####  
CREATE TABLE spamscores (  
  user varchar(40) NOT NULL default "",  
  lowspamscore decimal(10,0) NOT NULL default '0',  
  highspamscore decimal(10,0) NOT NULL default '0',  
  PRIMARY KEY (user)  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE USER_FILTERS  
-- #####  
CREATE TABLE user_filters (  
  username varchar(60) NOT NULL default "",  
  filter text,  
  verify_key varchar(32) NOT NULL default "",  
  active enum('N','Y') default 'N',  
  KEY user_filters_username_idx (username)  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE USERS (NEW) DOES NOT WORK  
-- #####  
-- # CREATE TABLE users (  
-- # username varchar(60) NOT NULL default "",  
-- # password varchar(32) default NULL,  
-- # fullname varchar(50) NOT NULL default "",  
-- # type enum('A','D','U','R','H') default NULL,  
-- # quarantine_report tinyint(1) default '0',  
-- # spamscore tinyint(4) default '0',  
-- # highspamscore tinyint(4) default '0',  
-- # noscan tinyint(1) default '0',  
-- # quarantine_rcpt varchar(60) default NULL,  
-- # PRIMARY KEY (username)  
-- # ) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE USERS (OLD) WORKS  
-- #####  
CREATE TABLE users (  
  username varchar(20) NOT NULL default "",  
  password varchar(32) default NULL,  
  fullname varchar(50) NOT NULL default "",  
  type enum('A','D','U','R') NOT NULL default 'A',  
  PRIMARY KEY (username)  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE TABLE WHITELIST  
-- #####  
CREATE TABLE whitelist (  
  id int(11) NOT NULL auto_increment,  
  to_address text,  
  to_domain text,  
  from_address text,  
  PRIMARY KEY (id),  
  UNIQUE KEY whitelist_uniq (to_address(100),from_address(100))  
) TYPE=MyISAM;  
  
-- #####  
-- # CREATE MAILWATCH WEB USERS  
-- #####  
INSERT INTO users VALUES ('mailwatch',md5('mailwatch'),'Mail Watch','A');
```