

StatsPlugin

Stats Plugin

NOTE: This module requires SpamAssassin 3.1+

Stats Plugin provides real-time statistics generated by [SpamAssassin](#). Statistics are stored inside of a MySQL table (defined as stats) which contains the following information:

\$day \$username \$domain \$ham \$spam

Each entry is rotated daily, meaning when the day changes a new entry is placed into the table with the new date and the following fields are set to zero: ham, spam.

To Begin Using

To begin using the Stats Plugin you will need to create a table for the plugin to write to. Here is the necessary schema:

Warning: The code below is only compatible with MySQL 4.1+! (Just like most of the SQL code for [SpamAssassin 3.1](#))

If you are using MySQL 4.0 and below you can download the old plugin, which contains a possible race condition [here](#).

```
CREATE TABLE stats (
  day date NOT NULL default'',
  username varchar(100) NOT NULL default '',
  domain varchar(100) default '',
  spam int(20) default '0',
  ham int(20) default '0',
  PRIMARY KEY (day,username, domain)
) TYPE=MyISAM
```

Once the table has been created you will need to add the following options into your local.cf or other configuration file:

- You will need to replace the following variables with settings for your configuration*

```
# Configure SQL for statistical storage
use_stats                1
user_stats_dsn            DBI:mysql:spamassassin_beta:sql_hostname
user_stats_sql_username   sql_username
user_stats_sql_password   sql_password
user_stats_sql_table      sql_table
```

Then the plugin will need to be added to your init.pre or other .pre file:

```
# Stats Plugin - store stats in a MySQL DB
#
loadplugin Mail::SpamAssassin::Plugin::Stats
```

Finally, we need to put the plugin inside of the [SpamAssassin](#) plugin directory. You can copy the code below, but I would recommend downloading it from my [site](#).

```
=head1 NAME

package Mail::SpamAssassin::Plugin::Stats - Keep Real Time SpamAssassin Stastics

=head1 SYNOPSIS

    loadplugin Mail::SpamAssassin::Plugin::Stats

    use_stats                1
    user_stats_dsn            DBI:mysql:spamassassin_beta:sql_hostname
    user_stats_sql_username   sql_user
    user_stats_sql_password   sql_pass
    user_stats_sql_table      stats
```

=head1 DESCRIPTION

This SpamAssassin plugin records real-time, user-level statistics. The statistics are stored inside of a MySQL database and are rotated daily. Each entry inside the table contains the current date, user, domain, number of hams (legitimate mail) and the number of spams (unsolicited mail).

=cut

```
package Mail::SpamAssassin::Plugin::Stats;
```

```
use strict;
```

```
use warnings;
```

```
use bytes;
```

```
use Mail::SpamAssassin;
```

```
use Mail::SpamAssassin::Logger;
```

```
use vars qw(@ISA);
```

```
@ISA = qw(Mail::SpamAssassin::Plugin);
```

```
sub new {
```

```
    my ($class, $mailsa) = @_;
```

```
    $class = ref($class) || $class;
```

```
    my $self = $class->SUPER::new($mailsa);
```

```
    bless ($self, $class);
```

```
    $self->set_config($mailsa->{conf});
```

```
    $self;
```

```
}
```

```
sub set_config {
```

```
    my ($self, $conf) = @_;
```

```
    my @cmds = ();
```

```
    push (@cmds, {
```

```
        setting => 'use_stats',
```

```
        default => 1,
```

```
        type => $Mail::SpamAssassin::Conf::CONF_TYPE_NUMERIC,
```

```
    });
```

```
    push (@cmds, {
```

```
        setting => 'user_stats_dsn',
```

```
        type => $Mail::SpamAssassin::Conf::CONF_TYPE_STRING,
```

```
    });
```

```
    push (@cmds, {
```

```
        setting => 'user_stats_sql_username',
```

```
        type => $Mail::SpamAssassin::Conf::CONF_TYPE_STRING,
```

```
    });
```

```
    push (@cmds, {
```

```
        setting => 'user_stats_sql_password',
```

```
        type => $Mail::SpamAssassin::Conf::CONF_TYPE_STRING,
```

```
    });
```

```
    push (@cmds, {
```

```
        setting => 'user_stats_sql_table',
```

```
        type => $Mail::SpamAssassin::Conf::CONF_TYPE_STRING,
```

```
    });
```

```
    $conf->{parser}->register_commands(\@cmds);
```

```
}
```

```

sub check_end {
    my ($self, $params) = @_;
    my $pms = $params->{permsgstatus};

    return 0 unless ($pms->{conf}->{use_stats});

    dbg("stats: Executing stats-plugin");

    my $dsn = $self->{main}->{conf}->{user_stats_dsn};
    if (!defined($dsn)) {
        dbg("stats: no DSN specified; HALT!");
        return 1;
    }

    require DBI;

    my $main = $self->{main};
    my $dbuser = $main->{conf}->{user_stats_sql_username};
    my $dbpass = $main->{conf}->{user_stats_sql_password};
    my $table = $main->{conf}->{user_stats_sql_table};

    my $f_spam = 'spam';
    my $f_ham = 'ham';
    my $f_username = 'username';
    my $f_domain = 'domain';
    my $f_day = 'day';
    my $isspam;
    my $user;
    my $domain;

    my $username = $self->{main}->{username};
    $username = lc($username);
    my $score = $pms->{score};
    my $required_score = $main->{conf}->{required_score};

    dbg("stats: Splitting $username based on @");
    ($user,$domain) = split /@/, $username;
    if (!defined($domain)) {
        $domain = '';
    }

    dbg("stats: User: $user Domain: $domain");
    dbg("stats: Message Score: $score out of $required_score");

    if ($score >= $required_score ) {
        $isspam = 1;
    }
    else {
        $isspam = 0;
    }

    dbg("stats: IsSpam is $isspam");

    my $dbh = DBI->connect($dsn, $dbuser, $dbpass, {'PrintError' => 0});

    if ($dbh) {
        &execute_stats($user, $domain, $dbh, $table, $f_spam, $f_ham, $f_username, $f_domain, $f_day, $isspam);
        $dbh->disconnect();
    }
    else {
        die "stats: SQL error: " . DBI->errstr . "\n";
    }
}

0;
}

sub execute_stats {
    my ($user, $domain, $dbh, $table, $f_spam, $f_ham, $f_username, $f_domain, $f_day, $isspam) = @_;

    my $column = ($isspam) ? $f_spam : $f_ham;

    my $sql = "INSERT into $table set $f_day = curdate(), $f_username='$user', $column=1";

```

```
$sql .= ", $f_domain='$domain'" if $domain;
$sql .= " ON DUPLICATE KEY UPDATE $column = $column + 1";

dbg("stats: config: SQL executing $sql");
my $rv = $dbh->do($sql);

if ($rv) {
    dbg("stats: Updated $column for $user $domain");
}
else {
    die "stats: SQL error: " . $dbh->errstr . "\n";
}
}

1;
```

Contact

Use this plugin at your own risk. I cannot guarantee it will not cause you issues!

If you have any problems with this plugin, suggestions, or would like to submit modifications to it please contact me at:

jamesk ~~at~~ okeating ~~dot~~ net