

# UsingSQL

## Using SQL

spamd can use SQL user\_prefs by calling it with the -q or -Q flags. the following configuration options will be required in local.cf to point to the correct database.

These configuration options below are specific to SA 3.0. 2.63 does not use user\_scores\_sql\_custom\_query but does have other items to reference table names and such.

```
[root@mailgw /]# cd /etc/mail/spamassassin/

[root@mailgw spamassassin]# cat sql.cf
user_scores_dsn          DBI:mysql:logs:localhost:3306
user_scores_sql_password password
user_scores_sql_username username
user_scores_sql_custom_query SELECT preference, value FROM _TABLE_ WHERE username = _USERNAME_ OR
                             username = '$GLOBAL' OR username = CONCAT('%',_DOMAIN_) ORDER BY username ASC
```

Make sure your user\_scores\_sql\_custom\_query is **ALL ON ONE LINE!** I wrapped it for display purposes only.

See [sql/README](#) in your [SpamAssassin](#) tarball for more information on SQL configuration options.

## SQL Table Structure for User Prefs

Personally, I use the following table structure for my user prefs... This allows my web interface to add additional information to the items via the **descript** field, and also has added, added\_by, and modified fields.

```
CREATE TABLE `userpref` (
  `id` int(8) unsigned NOT NULL auto_increment,
  `username` varchar(128) NOT NULL default '',
  `preference` varchar(64) NOT NULL default '',
  `value` varchar(128) default NULL,
  `descript` varchar(128) default NULL,
  `added` datetime NOT NULL default '2003-01-01 00:00:00',
  `added_by` varchar(128) NOT NULL default '',
  `modified` timestamp(14) NOT NULL,
  UNIQUE KEY `id` (`id`),
  KEY `type` (`preference`),
  KEY `added_by` (`added_by`),
  KEY `preference` (`preference`),
  KEY `username` (`username`)
) TYPE=MyISAM COMMENT='Spamassassin Preferences';
```

The userpref table as recommended by [sql/userpref\\_mysql.sql](#) in your SA tarball is..

```
CREATE TABLE userpref (
  username varchar(100) NOT NULL default '',
  preference varchar(30) NOT NULL default '',
  value varchar(100) NOT NULL default '',
  prefid int(11) NOT NULL auto_increment,
  PRIMARY KEY (prefid),
  KEY username (username)
) TYPE=MyISAM;
```

Not alot of difference... use whatever you want. If you are planning on having 1000's of user prefs via SQL, i would scale it back and not have all the extra fields.

To get this table active in your mysql database, save the syntax above to a file named **userpref.sql** and run

```
# cat userpref.sql | mysql <DB>
```

where **<DB>** is the database you want to store this table. If you have not already created it, you will want to run something like..

```
# mysql
> CREATE DATABASE spamassassin;
> exit;
```

and then

```
# cat userpref.sql | mysql spamassassin
```

Once you have the table active, you need to assign access to it. All [SpamAssassin](#) needs is **SELECT** access to the userpref table. Running

```
# mysql spamassassin
> GRANT SELECT ON spamassassin.userpref TO username IDENTIFIED BY 'password';
> exit;
```

will allow connections from spamd to connect to the database as user **username** and a password of **password**... please change those to something **SANE!**

## Global, Per-Domain, and Per-User Preferences via SQL

[SpamAssassin](#) 3.0 supports the config option **user\_scores\_sql\_custom\_query** - Using this feature, one can accomplish any number of custom configuration for supporting tiered user preferences.

However, for proper sorting of userpref when using Per-Domain settings, one must be careful on how SQL sorts the results. To make this work properly, you may be required to change how you reference your **GLOBAL** and **PER-DOMAIN** config preferences. To achieve proper sorting of SQL prefs, I use the following custom\_query:

```
user_scores_sql_custom_query      SELECT preference, value FROM _TABLE_ WHERE username = _USERNAME_ OR
                                  username = '$GLOBAL' OR username = CONCAT('%', _DOMAIN_) ORDER BY username ASC
```

Please avoid line wrapping on user\_scores\_sql\_custom\_query.. copying and pasting the above line into a .cf will not work.

The reason I do it this way, as opposed to the standard vpopmail way (@GLOBAL), is so the preferences in my user\_pref database order correctly...

Note that if you are customizing your query, i.e. not using the suggested query, it *should not* return the roll 'username' or you'll receive an error like 'info: config: failed to parse line, skipping, in "(no file)": *username @domain config\_tag*'. The query must return preference and value sorted accordingly.

Here is what the query produces...

```
mysql> select username,preference,value from userpref WHERE
(username='$GLOBAL' OR username='%nmgi.com' OR
username='dallase@nmgi.com') ORDER by username ASC;
```

username	preference	value
\$GLOBAL	required_hits	4.00
\$GLOBAL	subject_tag	[SPAM-_HITS_]-
\$GLOBAL	score USER_IN_WHITELIST	-10
\$GLOBAL	whitelist_from	*@sonicwall.com
\$GLOBAL	score USER_IN_BLACKLIST	10
\$GLOBAL	report_safe	0
\$GLOBAL	use_razor2	1
\$GLOBAL	use_pyzor	1
\$GLOBAL	use_dcc	1
\$GLOBAL	skip_rbl_checks	1
\$GLOBAL	use_bayes	1
\$GLOBAL	ok_locales	en
\$GLOBAL	ok_languages	en
\$GLOBAL	whitelist_from	*@nai.com
\$GLOBAL	use_auto_whitelist	1
\$GLOBAL	rewrite_header Subject	[SPAM-_HITS_]-
%nmgi.com	use_bayes	1
%nmgi.com	whitelist_from	*@hp.com
%nmgi.com	score USER_IN_WHITELIST	-10
%nmgi.com	score USER_IN_BLACKLIST	10
%nmgi.com	ok_locales	en
%nmgi.com	ok_languages	en
%nmgi.com	use_razor2	1
%nmgi.com	use_pyzor	1
%nmgi.com	use_dcc	1
%nmgi.com	skip_rbl_checks	1
%nmgi.com	report_safe	0
%nmgi.com	required_hits	5
dallase@nmgi.com	use_razor2	1
dallase@nmgi.com	use_bayes	1
dallase@nmgi.com	use_pyzor	1
dallase@nmgi.com	use_dcc	1
dallase@nmgi.com	skip_rbl_checks	0
dallase@nmgi.com	whitelist_from	*@paypal.com
dallase@nmgi.com	ok_languages	en
dallase@nmgi.com	ok_locales	en
dallase@nmgi.com	score USER_IN_WHITELIST	-10
dallase@nmgi.com	score USER_IN_BLACKLIST	10
dallase@nmgi.com	required_hits	4.10
dallase@nmgi.com	report_safe	0
dallase@nmgi.com	use_auto_whitelist	1
dallase@nmgi.com	rewrite_header Subject	[SPAM-_HITS_]-

Doing it this way guarantees proper sorting of prefs, so the last required\_hits found would be that of the user if defined, or that of the domain if defined... And if neither exist, it will assume global required\_hits.

Obviously, you'd need to rewrite a couple lines in your web administration packages that allows users to modify their SQL prefs with proper GLOBALS (\$GLOBAL) and DOMAIN (%nmgi.com) references...

## Testing your Setup

```
# mysql spamassassin;
> INSERT INTO userpref (username,preference,value) VALUES ('$GLOBAL','required_hits','5.0');
> INSERT INTO userpref (username,preference,value) VALUES ('%nmgi.com','required_hits','4.5');
> INSERT INTO userpref (username,preference,value) VALUES ('dallase@nmgi.com','required_hits','4.0');
> exit;
```

make sure spamd is running with the -q flag and that your SQL preferences are correct in your local.cf.

To debug it, make sure spamd is not running, and from the command line, start spamd in foreground (no -d flag) mode with full debugging enabled (-D), and watch for the SQL debug info...

```
# spamd -D -q
```

From another console, run

```
# echo -e "From: user\nTo:user\nSubject: Test\n\n" | spamc -u '$GLOBAL'
From: user
To:user
Subject: Test
X-Spam-Status: No, hits=1.3 required=5.0
```

and you should see required hits is set to 5.0

```
# echo -e "From: user\nTo:user\nSubject: Test\n\n" | spamc -u '%nmgi.com'
From: user
To:user
Subject: Test
X-Spam-Status: No, hits=1.3 required=4.5
```

should display required hits of 4.5

```
# echo -e "From: user\nTo:user\nSubject: Test\n\n" | spamc -u 'dallase@nmgi.com'
From: user
To:user
Subject: Test
X-Spam-Status: No, hits=1.3 required=4.0
```

should display required hits of 4.0

As you can see, the 'required' hits changed on each of them because it pulled the SQL preferences. How you get spamc to call -u with the proper information is up to you. In qmail-scanner, I do the following.. in the sub spamassassin function before it opens up the spamc call -

```
if ($recips =~ m/\./) {
    $sa_domain = &recips_in_same_domain($recips);
    if (defined $sa_domain && $sa_domain ne "") {
        $sa_user = "%" . $sa_domain;
    }
    else {
        $sa_user = "\$GLOBAL";
    }
}
else {
    $sa_user = $recips;
}
$sa_user =~ s/[^0-9a-z\._\-\=\+\@\$\%]/_/gi;
$sa_user =~ /^( [0-9a-z\._\-\=\+\@\$\%]+ )$/i;
$sa_user = tolower($1);

open(SA, "$spamc_binary $spamc_options -u \"$sa_user\" < $scandir/$wmaildir/new/$file_id|")
|| &tempfail("cannot run $spamc_binary < $scandir/$wmaildir/new/$file_id - $!");
```

my recips\_in\_same\_domain function looks like this... (i'm sure someone can write it much smarter)

```

sub recips_in_same_domain {
    my $recips = shift;
    my @recip=split(/\./,$recips);
    my (%doms,$user,$domain);
    my $count=0;

    if ($recips !~ m/\./) { return; };

    foreach my $r (@recip) {
        if ($r =~ m/\./) {
            ($user,$domain) = split(/\./,$r);
        }
        else {
            $user=$r;
            $domain='localhost';
        }
        $doms{$domain} = (int $doms{$domain} + 1);
    }

    foreach my $key (keys %doms) { $count++; }
    if ($count == 1) {
        return $domain;
    }
    else {
        return;
    }
}

```

Just add that function at the bottom of `qmail-scanner-queue.pl` somewhere so sub `spamassassin` can call it.

Fun Fun! Thats all for now.. Hopefully I'll add a little information on a web interface soon.

content by [DallasEngelken](#)