**BanSpam plugin for Mantis 2.x**

Version 1.10

2024

Cas Nuy

[www.nuy.info](http://www.nuy.info)

The BanSpam plugin tries to block spammers:

- Ensuring issues/notes can only be raised in one(1) language

- Offering the option to block ip (ranges)

- Quarantining of new issues/notes

Blacklists can be uploaded in bulk using csv file.

Issues/notes can automatically allowed based upon a set number of issues/notes in the DB.

Only applies to language check.

**Contents**

[**License & Support** 3](#_Toc165983679)

[License 3](#_Toc165983680)

[Support 3](#_Toc165983681)

[**Installation** 4](#_Toc165983682)

[**Configuration** 5](#_Toc165983683)

[**During normal operation** 6](#_Toc165983684)

[**Manage User IP-Addresses** 7](#_Toc165983685)

[**Manage Banned IP’s** 8](#_Toc165983686)

[**Import IP Blacklist** 9](#_Toc165983687)

[**Manage Quarantined Issues** 10](#_Toc165983688)

[**Manage Quarantined Notes** 11](#_Toc165983689)

[**Enabling Admin approval for new users** 12](#_Toc165983690)

# **License & Support**

## License

This plugin is released under the [GPL v3 license](<http://opensource.org/licenses/GPL-3.0>).

## Support

Please visit https://github.com/mantisbt-plugins/BanSpam

# **Installation**

Copy the BanSpam directory into the plugins folder of your installation.

After copying to your webserver:

- Start Mantis as administrator

- Select manage

- Select manage Plugins

- Select Install behind BanSpam 1.10

- Click on the plugin name for further configuration (see next page).

In case you want to check already during login, a small change in a core script is required.

Please add a line to login.php. just after:

$t\_user\_id = auth\_get\_user\_id\_from\_login\_name( $f\_username );

Line to insert:

$continue = event\_signal( 'EVENT\_LOGIN\_CHECK' );

In case you want to check already during signup, a small change in a core script is required.Please add a line to signup.php. just after:

$f\_captcha = mb\_strtolower( trim( $f\_captcha ) );

Line to insert:

$continue = event\_signal( 'EVENT\_SIGNUP\_CHECK' );

# **Configuration**

A screenshot of a computer

Description automatically generated

The plugin can be configured with the following options:

- Set language to use for issues/notes (supported en-de-fr-es)

- Set fallback language in case of ambiguous result

(?? means abort, otherwise set to language defined above)

- What is the minimum number of characters to activate language check?

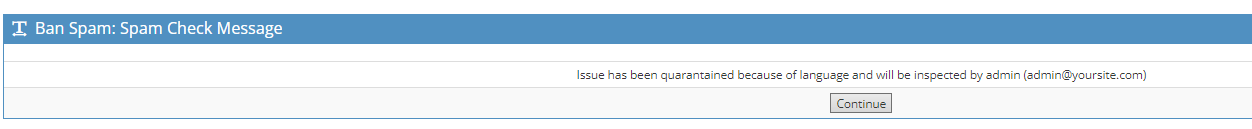
- Automatic approve text with less characters than the minimum stated above

- Do you want to check during logon yes/no?

- Do you want to check during signup?

- What is the contact e-mail (if any)?

# **During normal operation**



In case an issue or note is submitted while the language is not the required language, the reporter will be presented the above screen.

It is now up to authorized people to review quarantined items.

# **Manage User IP-Addresses**

A screenshot of a computer

Description automatically generated

Since mantis does not register the IP-address of its user, it is handled by this plugin.

On this page you can see all registered users with an IP-address.

In case the IP-Address is coloured RED, it means it is already banned for posting.

In all other cases you can ban an IP-Address straight from here.

Removing a ban can be done from the Manage Banned IP page.

# **Manage Banned IP’s**

A screenshot of a computer

Description automatically generated

Here you can:

* Add a new IP-range to ban from the system
* See which IP-ranges have been banned from the system and why
  + Banned ranged can be removed from here too.

\*\* Banning one IP-address can be done simply by setting that IP-address as the lower and upper limit of a range.

# **Import IP Blacklist**

A screenshot of a computer

Description automatically generated

You can use a simple CSV file to bulk upload ip-ranges to be banned.

In the Doc -directory you will find a sample csv sheet demonstrating how to set up this file.

The upload will ensure no duplicate range are loaded.

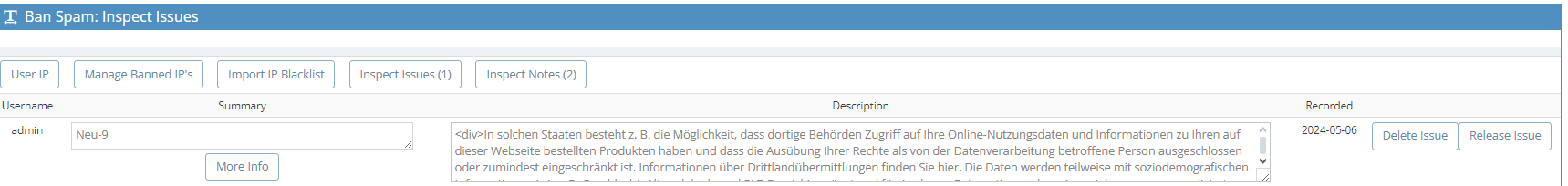
Be aware that the import will skip the first line, so please leave the header row in place.

Also make sure that you choose the correct column separator while using that function.

Default it is set to a comma but it just as well could be a semicolon.

An error message will popup in case you use the wrong separator.

# **Manage Quarantined Issues**



This page provides an overview of all issues that are currently quarantined. You can delete or release an issue here directly from this page.

In case more information is required, simply click on “More Info” and the following screen will appear.

A screenshot of a computer

Description automatically generated

Here more fields are shown in order to make a better judgement on releasing the issue. You can delete or release here directly from the page.

# **Manage Quarantined Notes**

A screenshot of a computer

Description automatically generated

This page provides an overview of all notes that are currently quarantined. You can delete or release a note here directly from this page.

# **Enabling Admin approval for new users**

In case you want to review new accounts before they gain access, you can do various things.

You can put the initial level for new accounts to “VIEWER” which ensures that new users can not post anything.

In case you want to manual approve new registrations, keep on reading.

When a new user register him/her-self on the Mantis site, the account is created like normal.  
Only difference is that the account is not yet enabled.  
That needs to be done by someone with rights to get in user management.  
These new accounts are trackable by filtering on the text as defined by $s\_new\_account (added to the.  
Do ensure that you want to see the disabled accounts too.

A screenshot of a computer

Description automatically generated

So all you need to do is to enable the account and preferably remove the temporary realname (not mandatory).  
That's it, now the new user can logon with the password provided earlier during the registration process.   
  
In order to enable this, some changes are required to a few standard Mantis scripts.

This does mean that these changes need to be made again after an update of Mantis itself.

Once the Mantis dev team comes up with an alternative, you may not need this anymore.

Installation  
In config/config.php add the following lines:

/\*\*  
 \* Initiate Admin approval for new users  
 \* When this is enabled, users that signup need to be validated by the admkin  
 \* @global integer $g\_admin\_approval  
 \*/  
$g\_admin\_approval = ON;

in core/constant\_inc.php add the following line:

define( 'ERROR\_ACCOUNT\_DISABLED', 817 );

In lang/strings\_english.txt add the following lines:

$s\_admin\_approval\_msg = 'Congratulations, you have registered successfully ! After review by the administrator , you will be sent a confirmation e-mail to verify your e-mail address. Visiting the link sent to you in this e-mail will activate your account.';  
$s\_new\_account = '=New Account=';  
$MANTIS\_ERROR[ERROR\_ACCOUNT\_DISABLED] = 'Your account has not yet been approved by the Administration, please come back later.';

In signup.php make the following change:  
Replace the line:

<?php echo lang\_get( 'password\_emailed\_msg' ) ?>

With:

<?php  
 if ( config\_get\_global( 'admin\_approval') ) {  
 echo lang\_get( 'admin\_approval\_msg' ) ;  
 } else {  
 echo lang\_get( 'password\_emailed\_msg' ) ;  
 }  
?>

in verify.php add the following.  
Find line stating:

$f\_confirm\_hash = gpc\_get\_string( 'confirm\_hash' );

After that insert:

# check if user approval is required, if so do not enable user yet  
 if (config\_get\_global( 'admin\_approval') ) {  
 return user\_create( $p\_username, $t\_password, $p\_email, auth\_signup\_access\_level(), false, false, lang\_get( 'new\_account' ) );  
 } else {  
 return user\_create( $p\_username, $t\_password, $p\_email, auth\_signup\_access\_level() );  
 }  
Finally in core user\_api.php lookup **function user\_signup**.  
Within that function, you simply replace the following line:

return user\_create( $p\_username, $t\_password, $p\_email, auth\_signup\_access\_level() );

with:

# check if user approval is required, if so do not enable user yet  
 if (config\_get\_global( 'admin\_approval') ) {  
 return user\_create( $p\_username, $t\_password, $p\_email, auth\_signup\_access\_level(), false, false, lang\_get( 'new\_account' ) );  
 } else {  
 return user\_create( $p\_username, $t\_password, $p\_email, auth\_signup\_access\_level() );  
 }