# Contents

1 Introduction .............................................. 3  
   1.1 History ............................................. 3  
      1.1.1 LTS Version .................................... 3  
   1.2 Future .............................................. 4  
   1.3 Features ............................................ 4  
      1.3.1 CloudFlare and Proxy Servers ................. 4  
      1.3.2 Comments ....................................... 4  
      1.3.3 Pingbacks .................................... 4  
      1.3.4 Spam ........................................... 4  
      1.3.5 User Enumeration ............................... 4  
      1.3.6 Work-Arounds for Broken syslogd .......... 4  
      1.3.7 Blocking Users .................................. 4  
      1.3.8 mu-plugins Support ............................. 4  
2 Installation ............................................. 5  
   2.1 Is WP fail2ban Already Installed? .................. 5  
   2.2 Overview ............................................ 5  
      2.2.1 Premium ....................................... 6  
3 Configuration ........................................... 7  
   3.1 WP fail2ban ......................................... 7  
   3.2 Logging ............................................. 7  
      3.2.1 Choosing the Events to Log .................. 7  
      3.2.2 Advanced Users ................................. 7  
   3.3 fail2ban ............................................. 8  
      3.3.1 Filters ......................................... 9  
   3.4 mu-plugins Support ................................. 10  
      3.4.1 Loading Early .................................. 10  
      3.4.2 Forcing Usage .................................. 10  
4 Usage ..................................................... 13  
   4.1 Event Log ........................................... 13  
   4.2 Report: Events by Country .......................... 13  
5 Add-ons .................................................. 15  
6 Developers ............................................... 17
6.1 Overview .......................................................... 17
6.1.1 Register Plugin ............................................. 17
6.1.2 Register Message .......................................... 18
6.1.3 Log Message ................................................ 19
6.1.4 Design ......................................................... 19
6.2 Example ......................................................... 19

7 Release Notes .................................................... 21
7.1 4.2.0 .............................................................. 21
7.2 4.2.1 .............................................................. 21
7.2.1 Upgrade ....................................................... 21
7.3 4.2.2 .............................................................. 22
7.3.1 Upgrade ....................................................... 22
7.4 4.2.3 .............................................................. 22
7.4.1 Upgrade ....................................................... 22
7.5 4.2.4 .............................................................. 22
7.5.1 Upgrade ....................................................... 22
7.6 4.2.5 .............................................................. 22
7.6.1 Upgrade ....................................................... 22
7.7 4.2.5.1 ............................................................ 23
7.7.1 Upgrade ....................................................... 23
7.8 4.2.6 .............................................................. 23
7.8.1 Upgrade ....................................................... 23
7.9 4.2.7 .............................................................. 23
7.9.1 Upgrade ....................................................... 23
7.10 4.2.7.1 .......................................................... 23
7.10.1 Upgrade ....................................................... 23
7.11 4.2.8 .............................................................. 23
7.11.1 Upgrade ....................................................... 24

8 define() Constants ................................................ 25
8.1 All ........................................................................ 25
8.1.1 WP_FAIL2BAN_AUTH_LOG .................................. 25
8.1.2 WP_FAIL2BAN_BLOCKED_USERS .............................. 25
8.1.3 WP_FAIL2BAN_BLOCK_USER_ENUMERATION ............... 26
8.1.4 WP_FAIL2BAN_COMMENT_EXTRA_LOG ................. 26
8.1.5 WP_FAIL2BAN_COMMENT_LOG .............................. 26
8.1.6 WP_FAIL2BAN_HTTP_HOST .................................... 26
8.1.7 WP_FAIL2BAN_LOG_COMMENTS ............................ 27
8.1.8 WP_FAIL2BAN_LOG_COMMENTS_EXTRA ............... 27
8.1.9 WP_FAIL2BAN_PASSWORD_REQUEST ....................... 28
8.1.10 WP_FAIL2BAN_LOG Pingbacks ......................... 28
8.1.11 WP_FAIL2BAN_LOG_SPAM .................................. 28
8.1.12 WP_FAIL2BAN_OPENLOG_OPTIONS .................... 28
8.1.13 WP_FAIL2BAN_PASSWORD_REQUEST_LOG ............ 28
8.1.14 WP_FAIL2BAN_PINGBACK_ERROR_LOG ................ 28
8.1.15 WP_FAIL2BAN_PINGBACK_LOG ............................ 29
8.1.16 WP_FAIL2BAN_PLUGIN_AUTH_LOG ..................... 29
8.1.17 WP_FAIL2BAN_PLUGIN_COMMENT_LOG ................ 29
8.1.18 WP_FAIL2BAN_PLUGIN_LOG_AUTH ..................... 29
8.1.19 WP_FAIL2BAN_PLUGIN_LOG_COMMENT ................ 29
8.1.20 WP_FAIL2BAN_PLUGIN_LOG_OTHER .................. 30
8.1.21 WP_FAIL2BAN_PLUGIN_LOG_PASSWORD ............... 30
8.1.22 WP_FAIL2BAN_PLUGIN_LOG_REST ..................... 30
8.1.23 WP_FAIL2BAN_PLUGIN_LOG_SPAM .................................................. 30
8.1.24 WP_FAIL2BAN_PLUGIN_LOG_XMLRPC ......................................... 30
8.1.25 WP_FAIL2BAN_PLUGIN_OTHER_LOG ........................................... 31
8.1.26 WP_FAIL2BAN_PLUGIN_PASSWORD_LOG ......................................... 31
8.1.27 WP_FAIL2BAN_PLUGIN_REST_LOG ................................................ 31
8.1.28 WP_FAIL2BAN_PLUGIN_SPAM_LOG ................................................ 31
8.1.29 WP_FAIL2BAN_PLUGIN_XMLRPC_LOG ........................................... 31
8.1.30 WP_FAIL2BAN_PROXIES .............................................................. 32
8.1.31 WP_FAIL2BAN_REMOTE_ADDR ...................................................... 32
8.1.32 WP_FAIL2BAN_SPAM_LOG ........................................................... 32
8.1.33 WP_FAIL2BAN_SYSLOG_SHORT_TAG ............................................. 32
8.1.34 WP_FAIL2BAN_TRUNCATE_HOST ................................................... 33
8.1.35 WP_FAIL2BAN_XMLRPC_LOG ....................................................... 33

8.2 Development ................................................................. 33
8.3 Logging ................................................................. 33
8.4 Plugins ............................................................... 33
8.5 Remote IPs ............................................................ 33
8.6 Reserved ............................................................... 33
8.7 syslog ................................................................. 33
8.8 Users ................................................................. 33

9 Facilities ............................................................. 35

10 Logfiles ............................................................ 37

11 Filter Files .......................................................... 39
  11.1 wordpress-hard.conf .................................................... 39
  11.2 wordpress-soft.conf ................................................... 40
  11.3 wordpress-extra.conf .................................................. 40
WP fail2ban is a WordPress plugin to write a myriad of events to syslog for integration with fail2ban.
CHAPTER 1

Introduction

1.1 History

As with many Open Source projects, P fail2ban started as way to scratch a particular itch. I had a dedicated server that was getting some unwelcome attention from various bots, and while it was trivial to configure fail2ban for ssh etc, WordPress was another story. Thus WP fail2ban was born late November 2011.

Since then it’s slowly but steadily accumulated features, and much to my surprise, gained a considerable number of installs (30,000+ at the time of writing) - I really had no idea so many other people would be interested!

Between versions 3.5 and 3.6 there was a bit of a delay. I switched my development environment from Windows 10 \(^1\) to a FreeBSD workstation and a Linux laptop, life then decided to take its turn and get in the way for a bit, all while the shadow of Gutenberg loomed large over the future of WordPress. With the advent of ClassicPress \(^2\) things started to look sunnier, so I dusted off the repo, put together some better documentation, braved the horrors of svn, and in November 2018 released 3.6 as a pseudo 7th anniversary present.

1.1.1 LTS\(^3\) Version

My plan was to retire the 3.x branch, but you know what they say about plans….

It turns out that at least one large hosting provider pre-installs WP\(^2\)b in mu-plugins. It's always great to see your work being used widely, but it would have been even better if they had let me know - I could have tested that combination before releasing version 4.

As a result there will now be a 3.7. This will be the very last version 3 release, it will be tailored to being pre-installed in mu-plugins, and it will be supported indefinitely.

---

\(^1\) It took me a while to realise that Microsoft really do want to turn Windows 10 into a toy, but I got there eventually.

\(^2\) In the interests of full disclosure: I’m a Founding Committee Member and at the time of writing, Security Team Lead.

\(^3\) Long-Term Support
1.2 Future

Version 4 was born from a desire to visualise the things WP/2b was logging; being entirely separate and distinct from the core functionality, adding this as freemium features seemed like a good plan. Time will tell.

This logical separation will continue for all future versions - if you were happy with the way 3.6 worked you’ll be happy with future versions too.

1.3 Features

1.3.1 CloudFlare and Proxy Servers

WP/2b can be configured to work with CloudFlare and other proxy servers. For a brief overview see WP_FAIL2BAN_PROXIES.

1.3.2 Comments

WP/2b can log both successful comments (see WP_FAIL2BAN_LOG_COMMENTS), and unsuccessful comments (see WP_FAIL2BAN_LOG_COMMENTS_EXTRA).

1.3.3 Pingbacks

WP/2b logs failed pingbacks, and can log all pingbacks. For a brief overview see WP_FAIL2BAN_LOG_PINGBACKS.

1.3.4 Spam

WP/2b can log comments marked as spam. See WP_FAIL2BAN_LOG_SPAM.

1.3.5 User Enumeration

WP/2b can block user enumeration. See WP_FAIL2BAN_BLOCK_USER_ENUMERATION.

1.3.6 Work-Arounds for Broken syslogd

WP/2b can be configured to work around most syslogd weirdness. For a brief overview see WP_FAIL2BAN_SYSLOG_SHORT_TAG and WP_FAIL2BAN_HTTP_HOST.

1.3.7 Blocking Users

WP/2b can be configured to short-cut the login process when the username matches a regex. For a brief overview see WP_FAIL2BAN_BLOCKED_USERS.

1.3.8 mu-plugins Support

WP/2b can easily be configured as a must-use plugin.
CHAPTER 2

Installation

2.1 Is WP fail2ban Already Installed?

WP fail2ban pre-installed in mu-plugins in a new DigitalOcean WordPress droplet.

2.2 Overview

WPf2b installs just like any other WordPress plugin - you need do nothing differently.
2.2.1 Premium

The Premium version installs via Freemius.

Database

Activating \textit{WPf2b} Premium creates two database tables:

- \texttt{wp\_fail2ban\_log}
- \texttt{wp\_fail2ban\_plugins}

\textit{WPf2b} Premium never drops the database tables - it’s your data.
Now you have WPf2b installed and activated it’s time to make it do something useful.

### 3.1 WP fail2ban

The Free version of WPf2b is configured by defining constants in wp-config.php. If you’re using the Premium version, or you know your way around wp-config.php already, skip ahead to Logging.

The first step is to check you can edit your wp-config.php file. If you’re not sure how to do that you’ll need to contact your hosting provider - for now you can skip ahead to configuring fail2ban.

The second step is to take a backup of wp-config.php. We’re not going to touch any other part of WordPress, so if anything goes wrong and your site stops working, restoring this backup should get you running again.

### 3.2 Logging

The key concept behind WPf2b is logging Events to syslog. If WPf2b doesn’t log an Event, or logs it to the wrong place, fail2ban won’t work as it should. If in doubt go with the defaults - they should work for most systems, and once you understand how the pieces fit together you can revisit this.

#### 3.2.1 Choosing the Events to Log

If you’re unfamiliar with fail2ban and syslog I recommend not enabling any extra logging to start with - skip ahead to configuring fail2ban. WPf2b automatically handles the most important things with sensible defaults that should work for most systems.

#### 3.2.2 Advanced Users
Events

Over the years \textit{WPf2b} has accumulated a lot of logging ability (and there’re even more on the way):

<table>
<thead>
<tr>
<th>Event</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth OK</td>
<td>WP_FAIL2BAN_AUTH_LOG</td>
</tr>
<tr>
<td>Auth Fail</td>
<td></td>
</tr>
<tr>
<td>Blocked User</td>
<td>WP_FAIL2BAN_BLOCKED_USERS</td>
</tr>
<tr>
<td>Blocked User Enumeration</td>
<td>WP_FAIL2BAN_BLOCK_USER_ENUMERATION</td>
</tr>
<tr>
<td>Comment</td>
<td>WP_FAIL2BAN_LOG_COMMENTS</td>
</tr>
<tr>
<td>Comment: Spam</td>
<td>WP_FAIL2BAN_LOG_SPAM</td>
</tr>
<tr>
<td>Attempted Comment: Post not found</td>
<td>WP_FAIL2BAN_LOG_COMMENTS_EXTRA</td>
</tr>
<tr>
<td>Attempted Comment: Closed post</td>
<td>WP_FAIL2BAN_LOG_COMMENTS_EXTRA</td>
</tr>
<tr>
<td>Attempted Comment: Trash post</td>
<td>WP_FAIL2BAN_LOG_COMMENTS_EXTRA</td>
</tr>
<tr>
<td>Attempted Comment: Draft post</td>
<td>WP_FAIL2BAN_LOG_COMMENTS_EXTRA</td>
</tr>
<tr>
<td>Attempted Comment: Password-protected post</td>
<td>WP_FAIL2BAN_LOG_COMMENTS_EXTRA</td>
</tr>
<tr>
<td>Pingback</td>
<td>WP_FAIL2BAN_LOG_PINGBACKS</td>
</tr>
<tr>
<td>Pingback error</td>
<td>WP_FAIL2BAN_PINGBACK_ERROR_LOG</td>
</tr>
</tbody>
</table>

You should consider enabling \textit{Comment: Spam} and \textit{Attempted Comment: Closed post}, and, if you don’t use WordPress’s commenting system at all, you should enable all the \textit{Attempted Comment} Events.

Facilities

By default, \textit{WPf2b} uses the following \texttt{syslog} Facilities and \textit{Levels}:

<table>
<thead>
<tr>
<th>What</th>
<th>Default</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth OK</td>
<td>LOG_AUTH</td>
<td>INFO</td>
</tr>
<tr>
<td>Auth Fail</td>
<td></td>
<td>NOTICE</td>
</tr>
<tr>
<td>Blocked User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blocked User Enum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment: Spam</td>
<td>LOG_USER</td>
<td>INFO</td>
</tr>
<tr>
<td>Comment: Post not found</td>
<td>LOG_AUTH</td>
<td>NOTICE</td>
</tr>
<tr>
<td>Comment: Closed post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment: Trash post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment: Draft post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comment: Password-protected post</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pingback</td>
<td>LOG_USER</td>
<td>INFO</td>
</tr>
<tr>
<td>Pingback error</td>
<td>LOG_AUTH</td>
<td>NOTICE</td>
</tr>
</tbody>
</table>

Unfortunately, there is no way of knowing \textit{a priori} which Facility goes where. There is a table of default locations of \texttt{Logfiles} for various OSs; if you’re running something not listed there and you know where the various Facilities go, please either submit a PR on GitHub, or let me know in the forum.

3.3 \texttt{fail2ban}

\texttt{fail2ban} can be tricky to configure correctly; with so many flavours of Linux it’s impossible to provide anything but general guidance.
3.3.1 Filters

The filter files included are intended only as a starting point for those who want WPf2b to work “out of the box”. There is no “one size fits all” configuration possible for fail2ban - what may be a soft failure for one site should be treated as a hard failure for another, and vice versa. Careful thought should be given to what is appropriate for your environment.

Typical Settings

1. Copy `wordpress-hard.conf` and `wordpress-soft.conf` to your `fail2ban/filters.d` directory

2. Edit `jail.local` to include something like:

```
[wordpress-hard]
enabled = true
filter = wordpress-hard
logpath = /var/log/auth.log
maxretry = 1
port = http,https

[wordpress-soft]
enabled = true
filter = wordpress-soft
logpath = /var/log/auth.log
maxretry = 3
port = http,https
```

Note: Make sure you change `logpath` to the correct log for your OS. If your OS uses systemd you may need to install a real syslog service.

3. Reload or restart fail2ban

*wordpress-hard.conf* and *wordpress-soft.conf*

There are some things that are almost always malicious, e.g. blocked users and pingbacks with errors. `wordpress-hard.conf` is designed to catch these so that you can ban the IP immediately.

Other things are relatively benign, like a failed login. You can’t let people try forever, but banning the IP immediately would be wrong too. `wordpress-soft.conf` is designed to catch these so that you can set a higher retry limit before banning the IP.

For the avoidance of doubt: you should be using both filters.

*wordpress-extra.conf*

Version 4 introduced a number of new logging options which didn’t fit cleanly into either of the hard or soft filters - they’re extra.

For example, if your site doesn’t use WordPress comments at all, you could add the rules matching attempted comments to the hard filter. Again, there is no “one size fits all” for these rules.
### 3.4 mu-plugins Support

There are two main reasons for using `mu-plugins`:

1. You need to load *WPf2b* before other security plugins¹.
2. You don’t trust the site administrators.

#### 3.4.1 Loading Early

One of the better ways is to install *WPf2b* as usual and then create a symlink in `mu-plugins`:

```bash
# ln -s ../plugins/wp-fail2ban/wp-fail2ban.php
# ls -l
```

```
```

or for the Premium version:

```bash
# ln -s ../plugins/wp-fail2ban-premium/wp-fail2ban.php
# ls -l
```

```
```

This has the advantage that you can update *WPf2b* as usual without having to update `mu-plugins` directly. For the free version you don’t need to activate *WPf2b*, but you do for the Premium version.

#### 3.4.2 Forcing Usage

The main objective here is to stop people fiddling with things, so there are necessarily some restrictions on configuring *WPf2b*.

*WPf2b* must be configured in `wp-config.php` - you can’t use the Premium config UI; not only does it make no sense, it won’t work².

The actual configuration itself is simple; for the **Free** version:

1. Extract the **Free** version of *WPf2b* into a directory called `wp-fail2ban` within `mu-plugins`.
2. symlink `wp-fail2ban.php`:

```bash
# ln -s wp-fail2ban/wp-fail2ban.php
# ls -l
```

```
```

3. **Keep WPf2b up-to-date.**

For the **Premium** version:

1. Extract the **Premium** version of *WPf2b* into a directory called `wp-fail2ban-premium` within `mu-plugins`.
2. symlink `wp-fail2ban.php`:

```bash
# ln -s wp-fail2ban-premium/wp-fail2ban.php
# ls -l
```

```
```

¹ For example, WordFence, which assumes it’s the only one.
² It may look like it works now, but in a future release it will be blocked.
3. Keep *WPf2b* up-to-date.

Keeping *WPf2b* up-to-date

It’s that last step that catches out most people - WordPress doesn’t check **mu-plugins** for updates, so by configuring *WPf2b* in this way you are **taking responsibility** for keeping *WPf2b* up-to-date. I do my best, but I cannot guarantee there will never be a critical problem with *WPf2b* - you and you alone are responsible for checking for updates and installing them.
CHAPTER 4

Usage

4.1 Event Log

4.2 Report: Events by Country
CHAPTER 5

Add-ons

Gravity Forms
New in version 4.2.
Version 4.2 introduced the ability for 3rd-party plugins to integrate with WPf2b.

6.1 Overview

The basic steps are:

6.1.1 Register Plugin

**Description**

**Usage**

```php
try {
    do_action('wp_fail2ban_register_plugin', 'my-plugin-slug', 'My Plugin Name');
} catch(
    \LengthException $e)
{
    // slug or name too long
} catch(
    \RuntimeException $e)
{
    // database error
}
```

**Parameters**

- `wp_fail2ban_register_plugin` WPf2b action.
- `my-plugin-slug` The plugin slug to register. Must be < 256 chars.
- `My Plugin Name` The display name of the plugin being registered. Must be < 256 chars.
Exceptions

LengthException Either the slug or name is too long.

6.1.2 Register Message

Example

```php
$args = [
    'slug' => 'my-plugin-msg-slug-1',
    'fail' => 'hard',
    'priority' => LOG_NOTICE,
    'event_class' => 'Password',
    'event_id' => 0x001F,
    'message' => 'My message with ___VAR1___ and ___VAR2___',
    'vars' => [
        'VAR1' => '\d+',
        'VAR2' => '*.'
    ]
];
try {
    do_action('wp_fail2ban_register_message', 'my-plugin-slug', $args);
} catch (InvalidArgumentException $e) {
    // Missing entry or invalid type
} catch (UnexpectedValueException $e) {
    // Invalid value
}
```

Details

do_action

wp_fail2ban_register_message WPf2b action.

my-plugin-slug The plugin slug used in Register Plugin.

$args

slug Message slug.

fail Recommended action.

priority syslog priority to use. Only the following priorities are supported:

- LOG_CRIT
- LOG_ERR
- LOG_WARNING
- LOG_NOTICE
- LOG_INFO
- LOG_DEBUG

event_class Class of Event. This is one of:

Auth Authentication-related Events. Note that Blocking Events will have their own class in the future.
Comment Comment-related Events.
XMLRPC XML-RPC-related Events.
Password Password-related Events.
REST REST API-related Events.
Spam Spam-related Events.

**event_id** Event ID - 16 bits you can do with as you please.
**message** Message with substitutions. Note that ” from <IP>” is appended.
**vars** An array of substitutions mapped to regular expressions.

When logging a message the substitutions are checked and substituted if present. The regex will be used to generate a matching rule for *fail2ban*.

### 6.1.3 Log Message

### 6.1.4 Design

To allow 3rd-party plugins to add support for *WPf2b* more easily, the API uses actions. This avoids the need to check if *WPf2b* is installed, then import a file, check for versions, and so on. Integration code can be written that will work if *WPf2b* is installed, and do nothing if not.

**Note:** Because *do_action* has no return value *WPf2b* will throw an Exception if there is an error.

### 6.2 Example

```php
/**
 * ...
 */

function myplugin_wpf2b_register() {
    // Register the plugin
    try {
        do_action('wp_fail2ban_register_plugin', 'my-plugin-slug', 'My Plugin Name');
    } catch (LengthException $e) {
        // slug or name too long
    } catch (RuntimeException $e) {
        // database error
    }

    // Register a message
    $args = [
        'slug' => 'my-plugin-msg-slug-1',
        'fail' => 'hard',
        'priority' => LOG_NOTICE,
        'event_class' => 'Password',
        'event_id' => 0x001F,
        'message' => 'My message with ___VAR1___ and ___VAR2___',
        'vars' => [
            'VAR1' => '\d+',
        ],
    ];
}
```

(continues on next page)
'VAR2' => '*.'

];
try {
    do_action('wp_fail2ban_register_message', 'my-plugin-slug', $args);
} catch (InvalidArgumentException $e) {
    // Missing entry or invalid type
} catch (UnexpectedValueException $e) {
    // Invalid value
}
}
add_action('wp_fail2ban_register', __NAMESPACE__.'\myplugin_wpf2b_register');

/**
 * function myplugin_foobar()
 *
 * $vars = [
 *     'VAR1' => 12345,
 *     'VAR2' => 'xyz'
 * ];
 * do_action('my-plugin-slug', 'my-plugin-msg-slug-1', $vars);
 */
Release Notes

7.1 4.2.0

- Not released.

7.2 4.2.1

- Completed support for `WP_FAIL2BAN_COMMENT_EXTRA_LOG`.
- Add support for 3rd-party plugins; see Developers.
  - Add-on for Contact Form 7 (experimental).
  - Add-on for Gravity Forms (experimental).
- Change logging for known-user with incorrect password; previously logged as unknown user and matched by hard filters (due to limitations in older versions of WordPress), now logged as known user and matched by soft.
- Bugfix for email-as-username - now logged correctly and matched by soft, not hard, filters.
- Bugfix for regression in code to prevent Free/Premium conflict.

7.2.1 Upgrade

You do not need to update your filters from 4.1.0.

7.3 4.2.2

- Fix 5.3 compatibility.
7.3.1 Upgrade

You do not need to update your filters from 4.1.0.

7.4 4.2.3

- Workaround for some versions of PHP 7.x that would cause `define()` to be ignored.
- Add config note to settings tabs.
- Fix documentation links.

7.4.1 Upgrade

This is a bugfix release. You do not need to update your filters from 4.1.0.

7.5 4.2.4

- Add filter for login failed message.
- Fix logging spam comments from admin area.
- Fix Settings link from Plugins page.
- Update Freemius library

7.5.1 Upgrade

This is a minor release. You do not need to update your filters from 4.1.0.

7.6 4.2.5

- Properly fix PHP 5.3 support; tested on CentOS 6. Does not support any UI or Premium features.
- Fix potential issue with `WP_FAIL2BAN_BLOCK_USER_ENUMERATION` if calling REST API or XMLRPC from admin area.

7.6.1 Upgrade

This is a minor release. You do not need to update your filters from 4.1.0.

7.7 4.2.5.1

- Fix premium activation issue with PHP < 7.0.
7.7.1 Upgrade
This is a premium-only patch release. If you are on PHP 7.0 or later you do not need to upgrade.

7.8 4.2.6
- Add support for Remote Tools add-on.
- Add support for the new ClassicPress security page.
- Improved user enumeration blocking.

7.8.1 Upgrade
This is a minor release. You do not need to update your filters from 4.1.0.

7.9 4.2.7
- Fix error when blocking user enumeration via REST.
- Fix buttons on Settings tabs.

7.9.1 Upgrade
This is a bugfix release. You do not need to update your filters from 4.1.0.

7.10 4.2.7.1
- Fix error when blocking user enumeration via oembed (h/t @wordpressfab).

7.10.1 Upgrade
This is a bugfix release. You do not need to update your filters from 4.1.0.

7.11 4.2.8
- Add link to new support forum.
- Fix user enumeration conflict with Gutenberg (h/t @dinghy).
- Fix notices wrt admin menu (h/t @marioivangf).
- Fix harmless XDebug notice (h/t @dinghy).
- Update Freemius library.
7.11.1 Upgrade

This is a bugfix release. You do not need to update your filters from 4.1.0. This is the last release that supports PHP 5.3.
8.1 All

8.1.1 WP_FAIL2BAN_AUTH_LOG

New in version 2.2.0.

By default, WPf2b uses LOG_AUTH for logging authentication success or failure. However, some systems use LOG_AUTHPRIV instead, but there’s no good run-time way to tell. If your system uses LOG_AUTHPRIV you should add the following to wp-config.php:

```php
define('WP_FAIL2BAN_AUTH_LOG', LOG_AUTHPRIV);
```

8.1.2 WP_FAIL2BAN_BLOCKED_USERS

New in version 2.0.0.

The bots that try to brute-force WordPress logins aren’t that clever (no doubt that will change), but they may only make one request per IP every few hours in an attempt to avoid things like fail2ban. With large botnets this can still create significant load.

Based on a suggestion from @jmadea, WPf2b now allows you to specify a regex that will shortcut the login process if the requested username matches.

For example, putting the following in wp-config.php:

```php
define('WP_FAIL2BAN_BLOCKED_USERS', '^admin$');
```

will block any attempt to log in as admin before most of the core WordPress code is run. Unless you go crazy with it, a regex is usually cheaper than a call to the database so this should help keep things running during an attack.

WPf2b doesn’t do anything to the regex other than make it case-insensitive.

If you’re running PHP 7, you can now specify an array of users instead:
define('WP_FAIL2BAN_BLOCKED_USERS', ['admin', 'another', 'user']);

### 8.1.3 WP_FAIL2BAN_BLOCK_USER_ENUMERATION

New in version 2.1.0.

Changed in version 4.0.0: Now also blocks enumeration via the REST API.

Brute-forcing WP requires knowing a valid username. Unfortunately, WP makes this all but trivial.

Based on a suggestion from @geeklol and a plugin by @ROIBOT, WPf2b can now block user enumeration attempts. Just add the following to `wp-config.php`:

```php
define('WP_FAIL2BAN_BLOCK_USER_ENUMERATION', true);
```

### 8.1.4 WP_FAIL2BAN_COMMENT_EXTRA_LOG

New in version 4.0.5.

Default: LOG_AUTH

```php
define('WP_FAIL2BAN_COMMENT_EXTRA_LOG', LOG_LOCAL5);
```

### 8.1.5 WP_FAIL2BAN_COMMENT_LOG

New in version 3.5.0.

By default, WPf2b uses LOG_USER for logging comments. If you’d rather it used a different facility you can change it by adding something like the following to `wp-config.php`:

```php
define('WP_FAIL2BAN_COMMENT_LOG', LOG_LOCAL3);
```

See also:

- `WP_FAIL2BAN_LOG_COMMENTS`
- `WP_FAIL2BAN_LOG_COMMENTS_EXTRA`

### 8.1.6 WP_FAIL2BAN_HTTP_HOST

New in version 3.0.0.

This is for some flavours of Linux where WP_FAIL2BAN_SYSLOG_SHORT_TAG isn’t enough.

If you configure your web server to set an environment variable named WP_FAIL2BAN_SYSLOG_SHORT_TAG on a per-virtual host basis, WPf2b will use that in the syslog tag. This allows you to configure a unique tag per site in a way that makes sense for your configuration, rather than some arbitrary truncation or hashing within the plugin.

**Note:** This feature has not been tested as extensively as others. While I’m confident it works, FreeBSD doesn’t have this problem so this feature will always be second-tier.
8.1.7 WP_FAIL2BAN_LOG_COMMENTS

New in version 3.5.0.

`WPf2b` can now log comments. To enable this feature, add the following to `wp-config.php`:

```php
define('WP_FAIL2BAN_LOG_COMMENTS', true);
```

The comment ID and IP will be written to `WP_FAIL2BAN_COMMENT_LOG` and matched by `wordpress-extra.conf`.

See also:
- `WP_FAIL2BAN_COMMENT_LOG`

8.1.8 WP_FAIL2BAN_LOG_COMMENTS_EXTRA

New in version 4.0.0.

`WPf2b` can optionally log the following comment-related events:

- **Not found**  Attempted comment on a non-existent post

  `WP_FAIL2B_EVENT_COMMENT_NOT_FOUND`

- **Closed**  Attempted comment on a post with closed comments

  `WP_FAIL2B_EVENT_COMMENT_CLOSED`

- **Trash**  Attempted comment on a post in Trash

  `WP_FAIL2B_EVENT_COMMENT_TRASH`

- **Draft**  Attempted comment on a Draft post

  `WP_FAIL2B_EVENT_COMMENT_DRAFT`

- **Password-protected**  Attempted comment on a password-protected post

  `WP_FAIL2B_EVENT_COMMENT_PASSWORD`

To enable this feature OR the event constants; for example, to enable `Closed` and `Draft`:

```php
define('WP_FAIL2BAN_LOG_COMMENTS_EXTRA', WPF2B_EVENT_COMMENT_CLOSED | WPF2B_EVENT_COMMENT_DRAFT);
```

The Post ID and IP will be written to `WP_FAIL2BAN_COMMENT_LOG` and matched by `wordpress-extra.conf`.

8.1. All
8.1.9 WP_FAIL2BAN_LOG_PASSWORD_REQUEST

New in version 3.5.0.

WPf2b can log password reset requests. Add the following to wp-config.php:

```php
define('WP_FAIL2BAN_LOG_PASSWORD_REQUEST', true);
```

The username and IP will be written to WP_FAIL2BAN_PASSWORD_REQUEST_LOG and matched by wordpress-extra.conf.

8.1.10 WP_FAIL2BAN_LOG_PINGBACKS

New in version 2.2.0.

Based on a suggestion from @maghe, WPf2b can now log pingbacks. To enable this feature, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_LOG_PINGBACKS', true);
```

By default, WPf2b uses LOG_USER for logging pingbacks. If you’d rather it used a different facility you can change it by adding something like the following to wp-config.php:

```php
define('WP_FAIL2BAN_PINGBACK_LOG', LOG_LOCAL3);
```

8.1.11 WP_FAIL2BAN_LOG_SPAM

New in version 3.5.0.

WPf2b can now log spam comments. To enable this feature, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_LOG_SPAM', true);
```

The comment ID and IP will be written to WP_FAIL2BAN_SPAM_LOG and matched by wordpress-hard.conf.

See also:
- WP_FAIL2BAN_SPAM_LOG

8.1.12 WP_FAIL2BAN_OPENLOG_OPTIONS

New in version 3.5.0.

8.1.13 WP_FAIL2BAN_PASSWORD_REQUEST_LOG

New in version 4.0.0.

8.1.14 WP_FAIL2BAN_PINGBACK_ERROR_LOG

New in version 4.0.5: Reserved for future use.

Default: LOG_AUTH
8.1.15 WP_FAIL2BAN_PINGBACK_LOG

New in version 2.2.0.
See `WP_FAIL2BAN_LOG_PINGBACKS`.

8.1.16 WP_FAIL2BAN_PLUGIN_AUTH_LOG

New in version 4.2.0.
Facility for “Auth” class plugin messages.
See also:
- `WP_FAIL2BAN_PLUGIN_LOG_AUTH`
- `Facilities`

8.1.17 WP_FAIL2BAN_PLUGIN_COMMENT_LOG

New in version 4.2.0.
Facility for “Comment” class plugin messages.
See also:
- `WP_FAIL2BAN_PLUGIN_LOG_COMMENT`
- `Facilities`

8.1.18 WP_FAIL2BAN_PLUGIN_LOG_AUTH

New in version 4.2.0.
To enable logging plugin “Auth” class messages, add the following to `wp-config.php`:

```
define('WP_FAIL2BAN_PLUGIN_LOG_AUTH', true);
```

See also:
- `WP_FAIL2BAN_PLUGIN_LOG_AUTH`

8.1.19 WP_FAIL2BAN_PLUGIN_LOG_COMMENT

New in version 4.2.0.
To enable logging plugin “Comment” class messages, add the following to `wp-config.php`:

```
define('WP_FAIL2BAN_PLUGIN_LOG_COMMENT', true);
```

See also:
- `WP_FAIL2BAN_PLUGIN_COMMENT_LOG`
8.1.20 WP_FAIL2BAN_PLUGIN_LOG_OTHER

New in version 4.2.0.
To enable logging plugin “Other” class messages, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_PLUGIN_LOG_OTHER', true);
```

See also:
- WP_FAIL2BAN_PLUGIN_OTHER_LOG

8.1.21 WP_FAIL2BAN_PLUGIN_LOG_PASSWORD

New in version 4.2.0.
To enable logging plugin “Password” class messages, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_PLUGIN_LOG_PASSWORD', true);
```

See also:
- WP_FAIL2BAN_PLUGIN_PASSWORD_LOG

8.1.22 WP_FAIL2BAN_PLUGIN_LOG_REST

New in version 4.2.0.
To enable logging plugin “REST” class messages, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_PLUGIN_LOG_REST', true);
```

See also:
- WP_FAIL2BAN_PLUGIN_REST_LOG

8.1.23 WP_FAIL2BAN_PLUGIN_LOG_SPAM

New in version 4.2.0.
To enable logging plugin “Spam” class messages, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_PLUGIN_LOG_SPAM', true);
```

See also:
- WP_FAIL2BAN_PLUGIN_SPAM_LOG

8.1.24 WP_FAIL2BAN_PLUGIN_LOG_XMLRPC

New in version 4.2.0.
To enable logging plugin “XMLRPC” class messages, add the following to wp-config.php:

```php
define('WP_FAIL2BAN_PLUGIN_LOG_XMLRPC', true);
```
See also:

- WP_FAIL2BAN_PLUGIN_XMLRPC_LOG

8.1.25 WP_FAIL2BAN_PLUGIN_OTHER_LOG

New in version 4.2.0.
Facility for “Other” class plugin messages.
See also:

- WP_FAIL2BAN_PLUGIN_LOG_OTHER
- Facilities

8.1.26 WP_FAIL2BAN_PLUGIN_PASSWORD_LOG

New in version 4.2.0.
Facility for “Password” class plugin messages.
See also:

- WP_FAIL2BAN_PLUGIN_LOG_PASSWORD
- Facilities

8.1.27 WP_FAIL2BAN_PLUGIN_REST_LOG

New in version 4.2.0.
Facility for “REST” class plugin messages.
See also:

- WP_FAIL2BAN_PLUGIN_LOG_REST
- Facilities

8.1.28 WP_FAIL2BAN_PLUGIN_SPAM_LOG

New in version 4.2.0.
Facility for “Spam” class plugin messages.
See also:

- WP_FAIL2BAN_PLUGIN_LOG_SPAM
- Facilities

8.1.29 WP_FAIL2BANPLUGIN_XMLRPC_LOG

New in version 4.2.0.
Facility for “XML-RPC” class plugin messages.
See also:
• WP_FAIL2BAN_PLUGIN_LOG_XMLRPC
• Facilities

8.1.30 WP_FAIL2BAN_PROXIES

New in version 2.0.0.
Changed in version 4.0.0: Entries can be ignored by prefixing with #

The idea here is to list the IP addresses of the trusted proxies that will appear as the remote IP for the request. When defined:

• If the remote address appears in the WP_FAIL2BAN_PROXIES list, WPf2b will log the IP address from the X-Forwarded-For header
• If the remote address does not appear in the WP_FAIL2BAN_PROXIES list, WPf2b will return a 403 error
• If there’s no X-Forwarded-For header, WPf2b will behave as if WP_FAIL2BAN_PROXIES isn’t defined

To set WP_FAIL2BAN_PROXIES, add something like the following to wp-config.php:

```php
define('WP_FAIL2BAN_PROXIES', '192.168.0.42,192.168.42.0/24');
```

WPf2b doesn’t do anything clever with the list - beware of typos!

If you’re running PHP 7 you can use an array instead:

```php
define('WP_FAIL2BAN_PROXIES', [
    '192.168.0.42',
    '192.168.42.0/24'
]);
```

8.1.31 WP_FAIL2BAN_REMOTE_ADDR

New in version 3.6.0.

Some themes and plugins anonymise requests

8.1.32 WP_FAIL2BAN_SPAM_LOG

New in version 4.0.0.

8.1.33 WP_FAIL2BAN_SYSLOG_SHORT_TAG

New in version 3.0.0.

Some flavours of Linux come with a syslogd that can’t cope with the normal message format WPf2b uses; basically, they assume that the first part of the message (the tag) won’t exceed some (small) number of characters, and mangle the message if it does. This breaks the regex in the fail2ban filter and so nothing gets blocked.

Adding:

```php
define('WP_FAIL2BAN_SYSLOG_SHORT_TAG', true);
```
to `functions.php` will make WP/2b use `wp` as the syslog tag, rather than the normal `wordpress`. This buys you 7 characters which may be enough to work around the problem, but if it’s not enough you should look at `WP_FAIL2BAN_HTTP_HOST` or `WP_FAIL2BAN_TRUNCATE_HOST` too.

### 8.1.34 WP_FAIL2BAN_TRUNCATE_HOST

New in version 3.5.0.

If you’ve set `WP_FAIL2BAN_SYSLOG_SHORT_TAG` and defining `WP_FAIL2BAN_HTTP_HOST` for each virtual host isn’t appropriate, you can set `WP_FAIL2BAN_TRUNCATE_HOST` to whatever value you need to make `syslog` happy:

```php
define('WP_FAIL2BAN_TRUNCATE_HOST', 8);
```

This does exactly what the name suggests: truncates the host name to the length you specify. As a result there’s no guarantee that what’s left will be enough to identify the site.

### 8.1.35 WP_FAIL2BAN_XMLRPC_LOG

New in version 3.6.0.

This is for debugging and future development.

Attackers are doing weird things with XML-RPC, so this logs the raw post data to the file specified:

```php
define('WP_FAIL2BAN_XMLRPC_LOG', '/var/log/xml-rpc.log');
```

8.2 Development

8.3 Logging

8.4 Plugins

8.5 Remote IPs

8.6 Reserved

8.7 syslog

8.8 Users
While the full list of facilities is reproduced here for completeness, using anything but \texttt{LOG\_AUTH}, \texttt{LOG\_AUTHPRIV}, and/or \texttt{LOG\_LOCAL0..7} is unlikely to have the desired results.

<table>
<thead>
<tr>
<th>Facility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{LOG_AUTH}</td>
<td>security/authorization messages (use \texttt{LOG_AUTHPRIV} instead in systems where that constant is defined)</td>
</tr>
<tr>
<td>\texttt{LOG_AUTHPRIV}</td>
<td>security/authorization messages (private)</td>
</tr>
<tr>
<td>\texttt{LOG_CRON}</td>
<td>clock daemon (cron and at)</td>
</tr>
<tr>
<td>\texttt{LOG_DAEMON}</td>
<td>other system daemons</td>
</tr>
<tr>
<td>\texttt{LOG_KERN}</td>
<td>kernel messages</td>
</tr>
<tr>
<td>\texttt{LOG_LOCAL0...7}</td>
<td>reserved for local use, these are not available in Windows</td>
</tr>
<tr>
<td>\texttt{LOG_LPR}</td>
<td>line printer subsystem</td>
</tr>
<tr>
<td>\texttt{LOG_MAIL}</td>
<td>mail subsystem</td>
</tr>
<tr>
<td>\texttt{LOG_NEWS}</td>
<td>USENET news subsystem</td>
</tr>
<tr>
<td>\texttt{LOG_SYSLOG}</td>
<td>messages generated internally by syslogd</td>
</tr>
<tr>
<td>\texttt{LOG_USER}</td>
<td>generic user-level messages</td>
</tr>
<tr>
<td>\texttt{LOG_UUCP}</td>
<td>UUCP subsystem</td>
</tr>
</tbody>
</table>
### Logfiles

<table>
<thead>
<tr>
<th>OS</th>
<th>Level</th>
<th>LOG_AUTH</th>
<th>LOG_AUTHPRIV</th>
<th>LOG_USER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS 7</td>
<td></td>
<td></td>
<td>/var/log/secure</td>
<td></td>
</tr>
<tr>
<td>FreeBSD</td>
<td>INFO</td>
<td>/var/log/auth/log</td>
<td>/var/log/auth/log</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOTICE</td>
<td>/var/log/auth/log</td>
<td>/var/log/auth/log</td>
<td>/var/log/messages</td>
</tr>
<tr>
<td>Ubuntu 18</td>
<td>(all)</td>
<td>/var/log/auth.log</td>
<td>/var/log/auth.log</td>
<td>/var/log/syslog</td>
</tr>
</tbody>
</table>
11.1 wordpress-hard.conf

# Fail2Ban filter for WordPress hard failures
# Auto-generated: 2019-04-18T14:45:30+00:00
#

[INCLUDES]

before = common.conf

[Definition]

_daemon = (?:wordpress|wp)

failregex = ^%(__prefix_line)sAuthentication attempt for unknown user .* from <HOST>$
^%(__prefix_line)sREST authentication attempt for unknown user .* from <HOST>$
^%(__prefix_line)sXML-RPC authentication attempt for unknown user .* from <HOST>$
^%(__prefix_line)sSpam comment \d+ from <HOST>$
^%(__prefix_line)sBlocked user enumeration attempt from <HOST>$
^%(__prefix_line)sBlocked authentication attempt for .* from <HOST>$
^%(__prefix_line)sXML-RPC multicall authentication failure from <HOST>$
^%(__prefix_line)sPingback error .* generated from <HOST>$

ignoreregex =

# DEV Notes:
# Requires the 'WP fail2ban' plugin:
# https://wp-fail2ban.com/
#
# Author: Charles Lecklider
11.2 wordpress-soft.conf

# Fail2Ban filter for WordPress soft failures
# Auto-generated: 2019-04-18T14:45:30+00:00
#

[INCLUDES]
before = common.conf

[Definition]
_daemon = (?:wordpress|wp)

failregex = ^%(__prefix_line)sAuthentication failure for .* from <HOST>$
^%(__prefix_line)sREST authentication failure for .* from <HOST>$
^%(__prefix_line)sXML-RPC authentication failure for .* from <HOST>$

ignoreregex =

# DEV Notes:
# Requires the 'WP fail2ban' plugin:
# https://wp-fail2ban.com/
#
# Author: Charles Lecklider

11.3 wordpress-extra.conf

# Fail2Ban filter for WordPress extra failures
# Auto-generated: 2019-04-18T14:45:30+00:00
#

[INCLUDES]
before = common.conf

[Definition]
_daemon = (?:wordpress|wp)

failregex = ^%(__prefix_line)sComment \d+ from <HOST>$
^%(__prefix_line)sComment post not found \d+ from <HOST>$
^%(__prefix_line)sComments closed on post \d+ from <HOST>$
^%(__prefix_line)sComment attempt on trash post \d+ from <HOST>$
^%(__prefix_line)sComment attempt on draft post \d+ from <HOST>$_<HOST>$_
^%(__prefix_line)sComment attempt on password-protected post \d+ from
^%(__prefix_line)sPassword reset requested for .* from <HOST>$_

ignoreregex =

# DEV Notes:
# Requires the 'WP fail2ban' plugin:
# https://wp-fail2ban.com/

(continues on next page)
# 11.3. wordpress-extra.conf

# Author: Charles Leclider