WP fail2ban Manual

Charles Lecklider

Mar 13, 2019

Contents

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 *WPf2b* 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 *WPf2b* 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

WPf2b can be configured to work with CloudFlare and other proxy servers. For a brief overview see *WP_FAIL2BAN_PROXIES*.

1.3.2 Comments

WPf2b can log both successful comments (see *WP_FAIL2BAN_LOG_COMMENTS*), and unsuccessful comments (see *WP_FAIL2BAN_LOG_COMMENTS_EXTRA*).

1.3.3 Pingbacks

WPf2b logs failed pingbacks, and can log all pingbacks. For a brief overview see *WP_FAIL2BAN_LOG_PINGBACKS*.

1.3.4 Spam

WPf2b can log comments marked as spam. See *WP_FAIL2BAN_LOG_SPAM*.

1.3.5 User Enumeration

WPf2b can block user enumeration. See *WP_FAIL2BAN_BLOCK_USER_ENUMERATION*.

1.3.6 Work-Arounds for Broken syslogd

WPf2b 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

WPf2b can be configured to short-cut the login process when the username matches a regex. For a brief overview see *WP_FAIL2BAN_BLOCKED_USERS*.

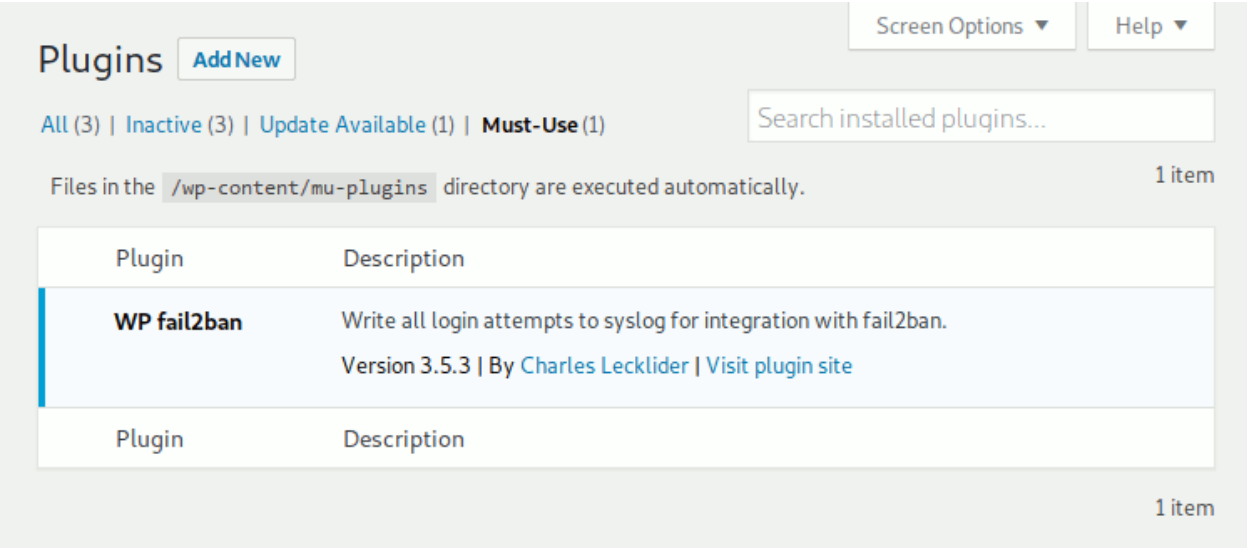
1.3.8 *mu-plugins* Support

WPf2b 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 *WPf2b* Premium creates one database table, `wp_fail2ban_log`.

WPf2b Premium never drops the database table - it's your data.

CHAPTER 3

Configuration

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 Permium 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 *WPf2b* has accumulated a lot of logging ability (and there're even more on the way):

Event	Reference
Auth OK	*WP_FAIL2BAN_AUTH_LOG*
Auth Fail	
Blocked User	*WP_FAIL2BAN_BLOCKED_USERS*
Blocked User Enumeration	*WP_FAIL2BAN_BLOCK_USER_ENUMERATION*
Comment	*WP_FAIL2BAN_LOG_COMMENTS*
Comment: Spam	*WP_FAIL2BAN_LOG_SPAM*
Attempted Comment: Post not found	*WP_FAIL2BAN_LOG_COMMENTS_EXTRA*
Attempted Comment: Closed post	*WP_FAIL2BAN_LOG_COMMENTS_EXTRA*
Attempted Comment: Trash post	*WP_FAIL2BAN_LOG_COMMENTS_EXTRA*
Attempted Comment: Draft post	*WP_FAIL2BAN_LOG_COMMENTS_EXTRA*
Attempted Comment: Password-protected post	*WP_FAIL2BAN_LOG_COMMENTS_EXTRA*
Pingback	*WP_FAIL2BAN_LOG_PINGBACKS*
Pingback error	*WP_FAIL2BAN_PINGBACK_ERROR_LOG*

You should consider enabling *Comment: Spam* and *Attempted Comment: Closed post*, and, if you don't use WordPress's commenting system at all, you should enable **all** the *Attempted Comment* Events.

Facilities

By default, *WPf2b* uses the following `syslog` Facilities and *Levels*:

What	Default	Level
Auth OK	*LOG_AUTH*	INFO
Auth Fail		NOTICE
Blocked User		
Blocked User Enum		
Comment	*LOG_USER*	INFO
Comment: Spam	*LOG_AUTH*	NOTICE
Comment: Post not found		
Comment: Closed post		
Comment: Trash post		
Comment: Draft post		
Comment: Password-protected post		
Pingback	*LOG_USER*	INFO
Pingback error	*LOG_AUTH*	NOTICE

Unfortunately, there is no way of knowing *a priori* which Facility goes where. There is a table of default locations of *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 fail2ban

`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
```

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 plugins[1],
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`:

```
lrwxr-xr-x  1  www  www  38  4 Nov 16:24 wp-fail2ban.php -> ../plugins/wp-fail2ban/wp-
↪fail2ban.php
```

or for the Premium version:

```
lrwxr-xr-x  1  www  www  38  4 Nov 16:24 wp-fail2ban.php -> ../plugins/wp-fail2ban-
↪premium/wp-fail2ban.php
```

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 work[2].

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`:

```
lrwxr-xr-x  1  www  www  38  4 Nov 16:24 wp-fail2ban.php -> wp-fail2ban/wp-fail2ban.
↪php
```

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`:

```
lrwxr-xr-x  1  www  www  38  4 Nov 16:24 wp-fail2ban.php -> wp-fail2ban-premium/wp-
↪fail2ban.php
```

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.

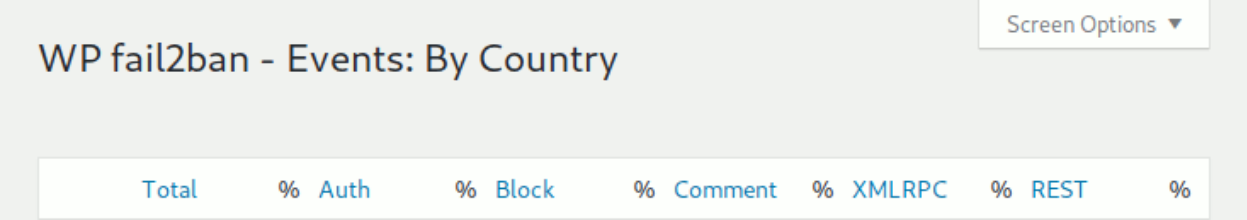
[1] For example, WordFence, which assumes it's the only one.
[2] It may look like it works now, but in a future release it will be blocked.

CHAPTER 4

Usage

4.1 Event Log

4.2 Report: Events by Country



CHAPTER 5

Release Notes

5.1 1.0.0

- Initial release.

5.2 1.1.0

- Minor cosmetic updates.

5.3 1.2.0

- Fix harmless warning.

5.4 1.2.1

- Update FAQ.

5.5 2.0.0

- Add *experimental* support for X-Forwarded-For header; see *WP_FAIL2BAN_PROXIES*.
- Add *experimental* support for regex-based login blocking; see *WP_FAIL2BAN_BLOCKED_USERS*.

5.5.1 Upgrade

This is an experimental release. If your current version is working and you're not interested in the new features, skip this version - wait for 2.1.0. For those that do want to test this release, note that *wordpress.conf* has changed - you'll need to copy it to *fail2ban/filters.d* again.

5.6 2.0.1

- Bugfix in *experimental* *WP_FAIL2BAN_PROXIES* code.

5.6.1 Upgrade

Bugfix in experimental code; still an experimental release.

5.7 2.1.0

- Add support for blocking user enumeration; see *WP_FAIL2BAN_BLOCK_USER_ENUMERATION*.
- Add support for CIDR notation in *WP_FAIL2BAN_PROXIES*.

5.7.1 Upgrade

The *wordpress.conf* filter has been updated; you will need to update your *fail2ban* configuration.

5.8 2.1.1

- Minor bugfix.

5.9 2.2.0

- Custom authentication log is now called *WP_FAIL2BAN_AUTH_LOG*.
- Add logging for pingbacks; see *WP_FAIL2BAN_LOG_PINGBACKS*.
- Custom pingback log is called *WP_FAIL2BAN_PINGBACK_LOG*.

5.9.1 Upgrade

BREAKING CHANGE: *WP_FAIL2BAN_LOG* has been renamed to *WP_FAIL2BAN_AUTH_LOG*.

Pingbacks are getting a lot of attention recently, so *WPf2b* can now log them. The *wordpress.conf* filter has been updated; you will need to update your *fail2ban* configuration.

5.10 2.2.1

- Fix stupid mistake with *WP_FAIL2BAN_BLOCKED_USERS*.

5.10.1 Upgrade

Bugfix.

5.11 2.3.0

- Bugfix in *experimental* *WP_FAIL2BAN_PROXIES* code (thanks to KyleCartmell).

5.11.1 Upgrade

Fix for *WP_FAIL2BAN_PROXIES*; if you're not using it you can safely skip this release.

5.12 2.3.2

- Bugfix *WP_FAIL2BAN_BLOCKED_USERS*.

5.13 3.0.0

- Add *WP_FAIL2BAN_SYSLOG_SHORT_TAG*.
- Add *WP_FAIL2BAN_HTTP_HOST*.
- Log XML-RPC authentication failure.
- Add better support for MU deployment.

5.13.1 Upgrade

BREAKING CHANGE: The *fail2ban* filters have been split into two files. You will need up update your *fail2ban* configuration.

5.14 3.0.1

- Fix regex in *wordpress-hard.conf*.

5.15 3.0.2

- Prevent double logging in WP 4.5.x for XML-RPC authentication failure

5.16 3.0.3

- Fix regex in *wordpress-hard.conf*.

5.16.1 Upgrade

You will need up update your *fail2ban* filters.

5.17 3.5.0

- Add *WP_FAIL2BAN_OPENLOG_OPTIONS*.
- Add *WP_FAIL2BAN_LOG_COMMENTS* and *WP_FAIL2BAN_COMMENT_LOG*.
- Add *WP_FAIL2BAN_LOG_PASSWORD_REQUEST*.
- Add *WP_FAIL2BAN_LOG_SPAM*.
- Add *WP_FAIL2BAN_TRUNCATE_HOST*.
- *WP_FAIL2BAN_BLOCKED_USERS* now supports an array of users with PHP 7.

5.17.1 Upgrade

You will need up update your *fail2ban* filters.

5.18 3.5.1

- Bugfix for *WP_FAIL2BAN_BLOCK_USER_ENUMERATION*.

5.18.1 Upgrade

Bugfix: disable *WP_FAIL2BAN_BLOCK_USER_ENUMERATION* in admin area. . . .

5.19 3.5.3

- Bugfix for *wordpress-hard.conf*.

5.19.1 Upgrade

You will need up update your *fail2ban* filters.

5.20 3.6.0

- The *Filter Files* are now generated from PHPDoc in the code. There were too many times when the filters were out of sync with the code (programmer error) - this should resolve that by bringing the patterns closer to the code that emits them.
- Added PHPUnit tests. Almost 100% code coverage, with the exception of *WP_FAIL2BAN_PROXIES* which is quite hard to test properly.
- Bugfix for *wordpress-soft.conf*.
- Add *WP_FAIL2BAN_XMLRPC_LOG*.
- Add *WP_FAIL2BAN_REMOTE_ADDR*.
- *WP_FAIL2BAN_PROXIES* now supports an array of IPs with PHP 7.

5.20.1 Upgrade

You will need up update your *fail2ban* filters.

5.21 4.0.0

Not released.

5.22 4.0.1

- Add extra features via Freemius. **This is entirely optional.** *WPf2b* works as before, including new features listed here.
- Add settings summary page.
- Add *WP_FAIL2BAN_PASSWORD_REQUEST_LOG*.
- Add *WP_FAIL2BAN_SPAM_LOG*.
- Add *WP_FAIL2BAN_LOG_COMMENTS_EXTRA* - enable logging for attempted comments on posts which are:
  - not found,
  - closed for commenting,
  - in the trash,
  - drafts,
  - password protected
- Block user enumeration via REST API.

5.22.1 Upgrade

To take advantage of the new features you will need up update your *fail2ban* filters; existing filters will continue to work as before.

CHAPTER 6

define() Constants

6.1 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`:

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

6.2 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`:

```
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']);
```

6.3 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`:

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

6.4 WP_FAIL2BAN_COMMENT_EXTRA_LOG

New in version 4.0.5.

Default: LOG_AUTH

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

6.5 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`:

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

See also:

WP_FAIL2BAN_LOG_COMMENTS.

6.6 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.

6.7 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`:

```
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*.

6.8 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

Closed Attempted comment on a post with closed comments

Trash Attempted comment on a post in Trash

Draft Attempted comment on a Draft post

Password-protected Attempted comment on a password-protected post

To enable this feature OR the Event IDs; for example, to enable *Closed* and *Draft*:

```
define('WP_FAIL2BAN_LOG_COMMENTS_EXTRA', 0x00020004 | 0x00020010);
```

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

6.9 WP_FAIL2BAN_LOG_PASSWORD_REQUEST

New in version 3.5.0.

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

```
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*.

6.10 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`:

```
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`:

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

6.11 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`:

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

The comment ID and IP will be written to *WP_FAIL2BAN_AUTH_LOG* and matched by *wordpress-hard.conf*.

6.12 WP_FAIL2BAN_OPENLOG_OPTIONS

New in version 3.5.0.

6.13 WP_FAIL2BAN_PASSWORD_REQUEST_LOG

New in version 4.0.0.

6.14 WP_FAIL2BAN_PINGBACK_ERROR_LOG

New in version 4.0.5: Reserved for future use.

Default: LOG_AUTH

```
define('WP_FAIL2BAN_PINGBACK_ERROR_LOG', LOG_LOCAL3);
```

6.15 WP_FAIL2BAN_PINGBACK_LOG

New in version 2.2.0.

See *WP_FAIL2BAN_LOG_PINGBACKS*.

6.16 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`:

```
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!

6.17 WP_FAIL2BAN_REMOTE_ADDR

New in version 3.6.0.

Some themes and plugins anonymise requests

6.18 WP_FAIL2BAN_SPAM_LOG

New in version 4.0.0.

6.19 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:

```
define('WP_FAIL2BAN_SYSLOG_SHORT_TAG', true);
```

to `functions.php` will make *WPf2b* 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.

6.20 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:

```
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.

6.21 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:

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

CHAPTER 7

Facilities

While the full list of facilities is reproduced here for completeness, using anything but **LOG_AUTH**, **LOG_AUTHPRIV**, and/or **LOG_LOCAL0**..**7** is unlikely to have the desired results.

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

CHAPTER 8

Logfiles

OS	Level	LOG_AUTH	LOG_AUTHPRIV	LOG_USER
CentOS 7			`/var/log/secure`	
FeeBSD	INFO	`/var/log/auth/log`	`/var/log/auth/log`	•
	NOTICE	`/var/log/auth/log`	`/var/log/auth/log`	`/var/log/messages`
Ubuntu 18	(all)	`/var/log/auth.log`	`/var/log/auth.log`	`/var/log/syslog`

CHAPTER 9

Filter Files

9.1 wordpress-hard.conf

```
# Fail2Ban filter for WordPress hard failures
# Auto-generated: 2019-03-13T01:12:18+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
```

9.2 wordpress-soft.conf

```
# Fail2Ban filter for WordPress soft failures
# Auto-generated: 2019-03-13T01:12:18+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
```

9.3 wordpress-extra.conf

```
# Fail2Ban filter for WordPress extra failures
# Auto-generated: 2019-03-13T01:12:18+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>$
            ^%(__prefix_line)sComment attempt on password-protected post \d+ from
↪<HOST>$
            ^%(__prefix_line)sPassword reset requested for .* from <HOST>$

ignoreregex =

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

(continues on next page)

(continued from previous page)

```
#
# Author: Charles Lecklider
```

CHAPTER 10

Premium

10.1 Events

Event	`action`	`ref_id`
Auth OK	0x00010001	*(null)*
Auth Fail	0x00010002	*(null)*
Blocked User[1]	0x00010004	*(null)*
Blocked User Enum[1]	0x00010008	*(null)*
Comment	0x00020000	`comment_ID`
Comment: Spam	0x00020001	`comment_ID`
Comment: Post not found	0x00020002	`post_ID`
Comment: Closed post	0x00020004	`post_ID`
Comment: Trash post	0x00020008	`post_ID`
Comment: Draft post	0x00020010	`post_ID`
Comment: Password-protected post	0x00020020	`post_ID`
Pingback	0x00040001	*(null)*
Pingback error	0x00040002	*(null)*

[1] These will change in v4.1.0.