



The Handbook

Codex

http://codex.wordpress.org/Combating_Comment_Spam/Denying_Access

Version Date

6 August 2005

Combating Comment Spam/Denying Access

While denying access to comment spammers may be seen as drastic action, there are ways to prevent access to spammers while still allowing comments to be posted.

Keep in mind that most spammers use random IPs. Blocking spammers by their IP does break up spam floods, but it also **increases the chance of blocking legitimate users**.

Deny Access to Spammer IPs

When a comment is sent to your weblog, the IP address is included in the packet of information that travels with that comment across the internet. Think of it as a phone number, and the WordPress comment moderation acts like call display to show you where the comment is coming from.

It should be noted that spammers are notorious for hijacking IP addresses, so it is possible that the IP address attached to a spam item is, in fact, "stolen" from a legitimate internet-connected device.

If you watch the IP addresses carefully, you may notice that there is only a slight variation in some of the numbers. For example, you might see:

- **192.168.0.1**
- **192.168.0.2**
- **192.168.0.3**

And other sequential or similar number orders. You have the ability to add a simpler IP address to your comment spam word list by dropping one or more of the IP numbers, thusly: **192.168** -- in this way, any IP address that starts with **192.168** will be screened as spam regardless of the numbers that appear with this "wildcard". It saves you having to type in lots of individual numbers. Be careful with how generic you make your wildcard IP numbers though, because just using **192.** would probably eliminate legitimate IP addresses to comment.

The `.htaccess` file - which also controls your permalinks - can be used to completely block an IP from even seeing your site. You can place this either in your site root, or the directory where your blog is (if they are different).

Below is an example of the `.htaccess` that is present in the root directory of a website.

```
<Limit GET>
order allow,deny
deny from 123.123.123.123
deny from 456.456.456.*
deny from 789.789.*.*
allow from all
</Limit>
```

```
deny from 123.123.123.123
    Access is denied to that IP alone
```

```
deny from 456.456.456.*
    Access is denied to ALL users whose IPs start with 456.456.456
```

deny from 789.789.*.*

Again, anyone at all with an IP that starts 789.789 is blocked.

So a total of 256*256 unique IP addresses are blocked

If you do start blocking IPs, then a blocked visitor will see a **403 error page**. Try to make sure that such a page has your contact details listed. Check your hosting to see how to make a custom 403 (or see below too).

If you start blocking IPs with the * wildcard, at least give someone the chance to email you to say you may have made a mistake.

If you want to check that someone is blocked, get a friend's IP, tell them what you are doing, and block them (until you know it works).

Deny Access to No Referrer Requests

When your readers comment, the `wp-comments-post.php` file is accessed, does it's thing, and creates the post. The user's browser will send a "referrer" line about this.

When a **spam-bot** comes in, it hits the file directly and **usually** does not leave a referrer. This allows for some nifty detection and action direct from the server. If you are not familiar with Apache directives, then write the following in your root directory `.htaccess` file::

```
RewriteEngine On
RewriteCond %{REQUEST_METHOD} POST
RewriteCond %{REQUEST_URI} .wp-comments-post\.php*
RewriteCond %{HTTP_REFERER} !.*yourdomain.com.* [OR]
RewriteCond %{HTTP_USER_AGENT} ^-$
RewriteRule (.*) ^http://%{REMOTE_ADDR}/$ [R=301,L]
```

This will:

1. Detect when a POST is being made
2. Check to see if the post is on `wp-comments-post.php`
3. Check if the **referrer** is in your domain or if **no referrer**
4. Send the spam-bot BACK to it's originating server's IP address.

NOTE 1: In the 4th line, change `yourdomain.com` to your **domain.xxx** without the `www` or any prefix for that matter.

NOTE 2: There is a slim chance that someone's browser will not send the referral, but this is extremely rare.

This essentially *deflects* the spam-bot back on itself.

TIP: If you want to see this work, and you know the absolute path to your root directory, then do this:

```
RewriteEngine On
RewriteLog /absolute/path/to/your/wwwroot/public_html/rewrite_log.txt
RewriteLogLevel 2
```

When the `RewriteRule` is activated, you will get something like this in `rewrite_log.txt`:

```
65.197.28.xxx- - [06/Feb/2005:10:59:34 --0500]
[yourdomain.com/sid#80054890][rid#804b6a50/initial] (2) init rewrite engine with
requested uri /wp-comments-post.php
65.197.28.xxx- - [06/Feb/2005:10:59:34 --0500]
```

```
[yourdomain.com/sid#80054890][rid#804b6a50/initial] (2) rewrite /press/wp-
comments-post.php -> http://65.197.28.xxx/
65.197.28.xxx- - [06/Feb/2005:10:59:34 --0500]
[yourdomain.com/sid#80054890][rid#804b6a50/initial] (2) explicitly forcing
redirect with http://65.197.28.170/
65.197.28.xxx- - [06/Feb/2005:10:59:34 --0500]
[yourdomain.com/sid#80054890][rid#804b6a50/initial] (1) escaping
http://65.197.28.xxx/ for redirect
65.197.28.xxx- - [06/Feb/2005:10:59:34 --0500]
[yourdomain.com/sid#80054890][rid#804b6a50/initial] (1) redirect to
http://65.197.28.xxx/ [REDIRECT/301]
```

Taken from an actual log

Deny Access Referrer Spammers

Many bloggers show referrer's to their site or links from which people came to visit their site. Spammers exploit this and indiscriminately spam blogs (even bloggers who do not have this feature enabled) with referral links pointing to their spammy sites. They end up wasting your resources, polluting your legitimate referrer's list and slowing down access for your readers.

In an effort to economize their resources, spammers often send out comment spam bots with their spam referrers for that two-in-one-shot effect. Consequently, you can block quite a few comment spam bots by blocking the referrer spam.

Once you know which referrer URL you'd like to block, and believe me you'll know, you can keep them out by adding the following into your .htaccess file:

```
SetEnvIfNoCase Via evil-spam-proxy spammer=yes
SetEnvIfNoCase Referer evil-spam-domain.com spammer=yes
SetEnvIfNoCase Referer evil-spam-keyword spammer=yes
SetEnvIfNoCase Via pinappleproxy spammer=yes
SetEnvIfNoCase Referer doobu.com spammer=yes
SetEnvIfNoCase Referer poker spammer=yes
```

```
Order allow,deny
allow from all
deny from env=spammer
```

The aforementioned .htaccess rules were brought to you by [Tom Raftery](http://www.tomrafteryit.net/using-htaccess-to-minimise-comment-and-referrer-spam/) (<http://www.tomrafteryit.net/using-htaccess-to-minimise-comment-and-referrer-spam/>), who originally used regular rewrite conditions and later decided that "using SetEnvIfNoCase instead of RewriteCond - seems to be quite effective (especially for referrers)."

Plugins for blocking Referrer Spam can be found on the [Plugins List for Comment Spam Prevention](#).

Using a Custom 403

It's a regular webpage, and again using .htaccess you tell the server to show your page, not the default one.

The .htaccess should have this:

ErrorDocument 403 /errors/403.html

Create a directory called "errors", for example, and put your custom 403 message into that. Again, to test, block a friend and ask for feedback.

You create a custom 404 the same way.

Caution: Read any and all documentation that your host has regarding use of custom pages - your server requirements may differ from the above suggestions! If in doubt, contact your host directly for assistance.

Using PHP Code to Block

You can add this to the top of any PHP page, putting the actual IP address where the xxx or yyy is.

```
<?php
$block = array("xxx.xxx.xxx.xxx", "yy.yy.y.yyy");

if (in_array ($_SERVER['REMOTE_ADDR'], $block)) {
    header("Location: http://google.com/");
    exit();
}
?>
```

Resources

- [Combating Comment Spam](#)
- [Comment Spam](#)
- [Spam Words](#)
- [Spam Plugin Tools](#)