## Securing PHP

Posted on October 17, 2011 by James Cunningham

On any modern linux distribution, it is very easy to get a LAMP stack running for your application. It can be done in as little as 5 commands (on CentOS at least).

```
yum install httpd mysql mysql-server php php-mysql
chkconfig --levels 235 httpd on
chkconfig --levels 235 mysqld on
service httpd start
service mysqld start
```

# Configuration

While the provided PHP configuration (found in /etc/php.ini) is okay, it's not great, and can be improved by changing some of the settings specified below.

## Safe Mode

Safe Mode in PHP is not security, and time and time again its vulntrabilities have surfaced. Not only is it not security, it can prevent applications from operating as intended and this led to it being dropped from PHP in future releases.

```
; Safe Mode
; http://www.php.net/manual/en/ini.sect.safe-mode.php#ini.safe-mode
safe_mode = Off
```

## Root Directory/open_basedir

PHP allows you to specify the root directory for an applications files, it is the open_basedir directive and will only allow the application to access certain directories (through fopen etc). In the past, open_basedir has been fairly easy to circumvent however has improved of late.

```
; open_basedir, if set, limits all file operations to the defined directory
; and below.  This directive makes most sense if used in a per-directory
; or per-virtualhost web server configuration file. This directive is
; *NOT* affected by whether Safe Mode is turned On or Off.
; http://www.php.net/manual/en/ini.sect.safe-mode.php#ini.open-basedir
open_basedir = /www:/tmp/phpupload
```

## Disable Dangerous Functions

There are all manors of functions which could be considered dangerous, mainly functions which allow you to execute system commands. If you are not using these functions, why leave them enabled? Disabling functions means they can no longer be used, and trying to execute them will issue a warning.

```
; This directive allows you to disable certain functions for security reasons.
; It receives a comma-delimited list of function names. This directive is
; *NOT* affected by whether Safe Mode is turned On or Off.
; http://www.php.net/manual/en/ini.sect.safe-mode.php#ini.disable-functions
disable_functions = exec,passthru,shell_exec,system,proc_open,popen,parse_ini_file,show_
```

## Expose PHP

For reasons unbeknownst to me, by default PHP will issue a header containing the fact PHP was used, and the version you are using. Essentially telling a potential attacker which exact version of PHP to look for vulnerabilities for, or identifying yourself to an automated attacker bot.

```
; Decides whether PHP may expose the fact that it is installed on the server
; (e.g. by adding its signature to the Web server header).  It is no security
; threat in any way, but it makes it possible to determine whether you use PHP
; on your server or not.
; http://www.php.net/manual/en/ini.core.php#ini.expose-php
expose_php = Off
```

## Resource Limits

PHP has built in resource limits, prevent a script from taking up either too much memory, or too much time.

```
; Maximum execution time of each script, in seconds
; http://www.php.net/manual/en/info.configuration.php#ini.max-execution-time
max_execution_time = 15

; Maximum amount of time each script may spend parsing request data. It's a good
; idea to limit this time on productions servers in order to eliminate unexpectedly
; long running scripts.
; Default Value: -1 (Unlimited)
; Development Value: 60 (60 seconds)
; Production Value: 60 (60 seconds)
; http://www.php.net/manual/en/info.configuration.php#ini.max-input-time
max_input_time = 30

; Maximum amount of memory a script may consume (128MB)
; http://www.php.net/manual/en/ini.core.php#ini.memory-limit
memory_limit = 8M

; Maximum size of POST data that PHP will accept.
; http://www.php.net/manual/en/ini.core.php#ini.post-max-size
post_max_size = 8M
```

## Displaying Errors

When writing applications in PHP, error message reporting is very useful however on a running applications it can expose sensitive internals of your application. By disabling error messages, this is not the case.

```
display_errors = Off
```

### Register Globals

Stop enabling register globals. It is disabled by default and being removed from PHP for a reason.

```
register_globals = Off
```

### Dynamic Libraries

It is possible with PHP to load a library or extension at runtime through PHP, however as with the dangerous functions, this could allow an attacker to execute dangerous code. If you are not using it, best just to disable it.

```
; Whether or not to enable the dl() function.  The dl() function does NOT work
; properly in multithreaded servers, such as IIS or Zeus, and is automatically
; disabled on them.
; http://www.php.net/manual/en/info.configuration.php#ini.enable-dl
enable_dl = Off
```

### URL FOpen

Through FOpen, you are able to access internet URL's as if they were local files and even include remote files for execution. But an attack could potentially use this to include and execute malicious code. Reading a remote file and verifying its contents before using the data is typically considered safe, but an include using URL's is not.

```
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; http://www.php.net/manual/en/filesystem.configuration.php#ini.allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like http:// or ftp://) as files.
; http://www.php.net/manual/en/filesystem.configuration.php#ini.allow-url-include
allow_url_include = Off
```

## Consider mod_php alternatives

Believe it or not, there are other ways to run PHP that with mod_php and apache. A popular alternative is to run PHP through FastCGI, which PHP has built in through PHP-FPM. This has the added advantage of a lower overhead, as the PHP interpreter is always running and not loaded on each request, and allows you to run PHP as a different user and from that point UNIX file permissions take over access to certain files and directories. Although other ways to exist to run PHP as a different user, such as suPHP and suExec.

### Hardened PHP

It is also possible to install a patch to PHP called Suhosin, which includes greater security and offers more flexible configuration options. (Thanks elliotcarlson from Hacker News)

This entry was posted in Code, Technology and tagged php, secure, security. Bookmark the permalink.

Like 3 people liked this.

Login

## Add New Comment

Sort by popular now

## Showing 11 comments

**Iongion**

Another step that should be done when display_errors is Off would be to enable error logging to file, this way you do not loose valuable information for fixing issues that might occur.

4 hours ago Like Reply

**James Cunningham**

I should have explained, that I have a followup post about monitoring and tracking PHP errors in production.

4 hours ago in reply to Iongion 1 Like Like Reply

**Iongion**

It makes no sense to block "parse_ini_file", there was a bogus bug in php long time ago and most of the ones recommending to remove it don't understand how it works.

4 hours ago Like Reply

**Philip Tellis**

set allow_url_fopen to Off. It automatically follows redirects, which includes redirects to files on your local filesystem. Better to use curl with FOLLOW_LOCATION turned off, and manually verify and follow all redirects.

5 hours ago Like Reply

**Chris Reed**

Nice shoutout out Hardened PHP, which has been the default install for FreeBSD since 6.2 I believe.

5 hours ago Like Reply

**greenlight**

I believe it's also default on Gentoo

5 hours ago in reply to Chris Reed Like Reply

**jameswade**

You disable_functions includes show_source but not highlight_file?

5 hours ago Like Reply

**stolen data**

Half of these things are already, by default, set to the option/mode you describe, and some of the settings are quite likely to impose problems in not particularly rare cases.

6 hours ago Like Reply

**James Cunningham**

At least not in CentOS, I had to change all these values (except enable_dl()). These settings will not fit everyone, but should work fine with most applications.

6 hours ago in reply to stolen data Like Reply

**Scott MacVicar**

Whats the reasoning for disabling the curl_*() functions? Everything else seems reasonable tbh.

6 hours ago Like Reply

**James Cunningham**

You are right, those shouldn't be there. Removed them now. I lifted some of these values off one of my production servers and we have those disabled.

6 hours ago in reply to Scott MacVicar Like Reply

✉ Subscribe by email RSS