

PHPSUEXEC - What do I need to know?

The security hole of PHP

On most Apache servers, PHP runs as an Apache Module. As such, it runs directly in the user Nobody, but doesn't require the execute flag. This means that in order to execute a PHP file, it simply needs to be world readable. The problem is that this allows every other user on the server to read your PHP files!

Allowing other users to read your HTML files is not a problem, since they can be displayed in Internet Explorer. However, PHP files are not readable, they are parsed. Many scripts use a PHP file to store a database username and password. This means that on another server every client could read your PHP files, retrieve your password and access your databases.

ISPs close this hole by installing an Apache module called PHPsuexec, which executes PHP scripts under your username. Instead of using everyone's permissions it uses the owner's permissions. Thus you can change the permissions of your PHP scripts to 0700 or 0400 and still read and execute them. However, these scripts will no longer be accessible to any other users -- PHPsuexec will refuse to execute a script if it is world-writable to protect you from someone abusing one of your scripts. All servers will be running PHPsuexec within the near future.

You can easily tell if your server has PHPsuexec enabled by visiting your server's phpinfo page: <http://your.servername.com/phpinfo.php/> simply look for the following near the top of that page. (4th Box Down Server API) :- "**Server API Apache**" This means that your server is currently running PHP as an apache module. If within the phpinfo page you see the following: "**Server API CGI**" Then your server has a CGI installation of PHP with suexec enabled.

What is the difference?

Most sites will not be affected with the change, running PHP as cgi with suexec. PHPsuexec works in much the same way that cgi (Perl scripts etc) with suexec does, all applications being run under your account user name UID/GID, rather than in PHP's case as an apache module, the user "nobody".

This simply means that rules that apply to .cgi + .pl files on your current server, apply to PHP files also. The maximum permissions permitted on directories and PHP files are 755. Failing to have permissions set to a maximum of 755 on PHP files and their installation paths, will result in a 500 internal server error, when attempting to execute them.

File & Folder Permissions

As you can see from the table, the only required permission is 'owner-read' 0400, but if you need to write to that file, you need to also enable the owner-write permission 0600. It is **recommended that all your PHP files have chmod permission 0400 or 0600**. The execute permission is never required, and the 'group' and 'everyone' permissions can be left to 0.

	Read	Write	Execute
Group	✓	To Write ✗	
Everyo		✗	

PHPsuexec also validates the directories in which PHP files are located. A PHP file cannot be executed in a directory that is 'group-writable' or 'world-writable'. However, in order to access a directory, it must be 'world-executable', which is safe to do. So folders containing PHP files should have permissions 0755 or 0555.

	Read	Write	Execute
Group	✓	To Write ✗	✓
Everyo	✓	✗	✓

Questions

777 - Do I need directories set to this? My install script says that I do.

No, you do not need to have directories or files set to 777, even if your installation documents tell you that you do. Permissions of 755 will work in the same way - Scripts owned by your account user UID/GID will be able to write to your files, the same way that they can running under apache with 777 permissions.

If you have PHP applications / scripts that have directories set to 777, (required to write to them under PHP/apache module), they would need to be changed. Also we would need to change ownerships of all files owned by user "nobody" to the user name UID/GID for your account.

.htaccess

You cannot manipulate the `php.ini` settings with `.htaccess` when running PHP as `cgi/PHPsuexec`. If you are using `.htaccess` with `php_value` entries within it, you would receive an internal server 500 error when attempting to access the scripts. This is because PHP is no longer running as an Apache module and Apache will not handle those directives any longer. All PHP values should be removed from your `.htaccess` files to avoid this issue. Placing a `php.ini` file in its place will solve this issue. (Please see below.)

Default settings, I need Zend Optimizer or PHP to run with different options than the servers default settings, can I do this?

The server default settings with `php.ini` may restrict certain applications, it is possible to modify the settings and how PHP will run on your account, on a per directory basis. If you have an application that requires for example :-
`register_globals = On` Then by creating a file named `php.ini` within the directory that the script is located within, with the entry `register_globals = On` would allow you to run that script with your settings.

If you also require say Zend Optimizer to be installed for your application, you would have :

```
register_globals = on
zend_optimizer.optimization_level=15
zend_extension = "/usr/local/Zend/lib/ZendOptimizer.so"
```

You may copy the other variables from the `phpinfo` page as they appear within it and modify the settings as required for your scripts.

Some important relevant default PHP values are as follows:

```
register_globals = off
register_argc_argv = off
safe_mode = on
magic_quotes_gpc = off
```

All other settings can be viewed from your server's `phpinfo.php` page.

Quick trouble shooter

HELP my PHP script doesn't work or I have an error message

- 1 Check that the PHP script that you are attempting to execute has permissions of no more than 755 - 644 will work just fine normally, this is not something that will need to be changed in most cases.
- 2 Check that the `chmod` permissions for the folder that the script resides in are set to a maximum of 755. This also includes folders that the script would need to have access to also.
- 3 Check that the files are owned by you i.e. not owned by user 'nobody'. Certain applications having been run under PHP as an Apache module, may have files owned by the Apache user - In that case, submit a helpdesk ticket for the file ownerships to be changed.
- 4 Check that you do not have an `.htaccess` file with `php_values` within it. They will cause a 500 internal server error, when attempting to execute the script. The `php_values` will need to be removed from your `.htaccess` file and a `php.ini` put in its place, containing the PHP directives as explained above.

More Information How to install suPHP:
<http://www.phpasks.com/suphp/index.php>