Disable PHP Expose / Disable PHP Credits page showing up with including some.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000 variable to a PHP powered WebServer

Author : admin

Many people are not aware that by default PHP 5 has enabled variable in the **php.ini** file which allows a third party person to query **PHP** for a few hard coded variables which displays various infos on PHP. Some of the infos displayed are:
PHP Credits

To see that request in your browser to a PHP powered webserver.

http://domainname.com/some.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000

PHP Group
Thies C. Arntzen, Stig Bakken, Shane Caraveo, Andi Gutmans, Rasmus Lerdorf, Sam Ruby, Sascha Schumann, Zeev Suraski, Jim Winstead, Andrei Zmievski

Language Design & Concept
Andi Gutmans, Rasmus Lerdorf, Zeev Suraski

PHP 5 Authors	
Contribution	**Authors**
Zend Scripting Language Engine	Andi Gutmans, Zeev Suraski
Extension Module API	Andi Gutmans, Zeev Suraski, Andrei Zmievski
UNIX Build and Modularization	Stig Bakken, Sascha Schumann, Jani Taskinen
Win32 Port	Shane Caraveo, Zeev Suraski, Wez Furlong
Server API (SAPI) Abstraction Layer	Andi Gutmans, Shane Caraveo, Zeev Suraski
Streams Abstraction Layer	Wez Furlong, Sara Golemon
PHP Data Objects Layer	Wez Furlong, Marcus Boerger, Sterling Hughes, George Schlossnagle, Ilia Alshanetsky

SAPI Modules	
Contribution	**Authors**
AOLserver	Sascha Schumann
Apache 1.3 (apache_hooks)	Rasmus Lerdorf, Zeev Suraski, Stig Bakken, David Sklar, George Schlossnagle, Lukas Schroeder
Apache 1.3	Rasmus Lerdorf, Zeev Suraski, Stig Bakken, David Sklar
Apache 2.0 Filter	Sascha Schumann, Aaron Bannert

This would reveal you the PHP version number + the PHP authors who took active part in the development of the current PHP release.
That could be even counted as a non-critical security flaw since it reveals PHP version and many companies nowadays prefers that the technology backing up their websites stays private.

Some other hard coded variables that can be requested to a PHP enabled server are:

http://domainname.com/some.php?=PHPE9568F34-D428-11d2-A769-00AA001ACF42
This will show you that the Apache Server or the questionable Server is configured to server PHP pages.

http://domainname.com/some.php?=PHPE9568F35-D428-11d2-A769-00AA001ACF42

This simple method could be another security threat if you're a security freak. It will reveal your Webserver is running PHP with Zend2 Framework extension enabled.

http://domainname.com/some.php?=PHPE9568F36-D428-11d2-A769-00AA001ACF42

This last one reveals again the PHP logo a bit jagged.
Be aware that this 4 are enabled by default in **php.ini** on PHP version 5.x.
Therefore from a security standpoint as well as to show off your professionalism you can disable it simply by editing your **php.ini** and changing the variable **expose_php = Off** .To do that quickly on Debian running Apache 2.2.x from the command line issue the commands:

```
debian-server:~# sed -e "s#expose_php = On#expose_php = Off#g" /etc/php5/apache2/php.ini >
/etc/php5/apache2/php.ini.1;
debian-server:~# mv /etc/php5/apache2/php.ini.1 /etc/php5/apache2/php.ini
```

That's it now the annoying information concerning PHP Credits, PHP Logos, and PHP Zend Framework Logos won't be exposed any more for pranksters.

Historically speaking in PHP version 4. There were are 3 really funny pictures hard coded into the PHP library. If you're running PHP and you want to check them out you have to do a request to your server like the one below:

http://domainname.com/?=PHPE9568F36-D428-11d2-A769-00AA001ACF42

The following funny pictures should appear right away :)

Now take some time and test the hidden requests on your PHP powered servers :)