

Re: Looking for general advice on security

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2006-04/msg01851.html>

- *From:* Schraalhans Keukenmeester <firstname_DOT_lastname_AT_xs4all_DOT_nl>
 - *Date:* Fri, 28 Apr 2006 16:39:24 +0200
-

tony@xxxxxxx wrote:

I'm designing a survey form page that will be fairly complex and am becoming confident enough with PHP now to tackle most things.
(Thanks to everyone here who has helped)

Before I go too far with this I was wondering if anyone could perhaps offer advice or point me to any documents/web pages that could help with ensuring the security of the form/page and site. It is likely that the form will come under attack I expect.

Even comments about the best chmod settings are welcome.

I'd rather not have to wade through another history of the internet book with the words "and be security conscious by using SSL" on the last page which is what most advice I've found so far boils down to.

I've located standard advice such as using PHP strip-tags on input fields and other PHP specific stuff but was wondering how best to get interactive with the security.

Are there any PHP libraries perhaps that help with this?

I'm thinking of things like verifying users ID while they are online without having them email and preventing bots from getting in and things like that.

Any input on this would be most welcome.

thanks

tony

Consider using https for starters, if the page collects more or less sensitive information.

It may be a good practice from security point of view to have ALL your form-vars parsed by a separate handler/filter first before handing them to your operational code. In other words, there should never be any referral to \$_POST["anyname"] (or \$_GET) in your actual processing code. The handler should:

Check type consistency

Re: Looking for general advice on security

Truncate data to intended maximum length/size

Strip possibly dangerous chars, esp wrt MySQL injection. (mod_security on your webserver can help with this further)

Discard any variable with a name not used in your form, and possibly take further action (like blocking client ip)

Further: keep track of amount of submits by an address or address range in a given timeframe. If this exceeds a reasonable value, block the address (temporarily?). Or have a lookup in your db how long it was since the last attempt for ip w.x.y.z and introduce a delay for the user.

To prevent bot-action, include a verification field with hard to scan characters.

keep db passwords, names etc. in an include file outside your docroot.
having non-standard names is a good thing from security point of view.

Translate form fieldnames to different ones you use in your db.

Don't use \$_GLOBALS.

Set safe mode on if it's not already the default mode on your server.

The server should run as a separate user of course. Never root, nor a regular user. Be careful with symlinks jumping out of the tree.

Set rights to a minimum requirement. I use rw- r-- --- root:apache on anything that does not need to be modified by the webserver. Consider setting the immutable bit on critical files as well.

Chrooting the server and/or mysql is worth considering on any production/high visibility server. (Securityfocus has great walkthroughs on this). Of course only applicable if you have access to your own server as root.

I'm not a big fan of too much interactivity and additional user-checking. It gets complicated soon and may even give a false sense of security if there's holes in it you may have overlooked.

Anything helpful here ? Hope so.

GL!

Sh.

--

Backbone Scoliosis

.