

[UNIX] PHP Multiple Vulnerabilities (File Upload, parse_str() register_global bypassing, phpinfo XSS)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-11/0014.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/02/05

To: list@securiteam.com

Date: 2 Nov 2005 12:29:31 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

PHP Multiple Vulnerabilities (File Upload, parse_str() register_global bypassing, phpinfo XSS)

SUMMARY

" <<http://www.php.net/>> PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML."

By crafting special upload file, it is possible to overwrite the global array of PHP allowing attackers to inject different content to PHP global variables. By generating multiple requests to a code that uses parse_str() it is possible for attackers to open the register_global to be activated in PHP. A XSS in phpinfo() function of PHP allow attackers to steal information from users.

DETAILS

Vulnerable Systems:

- * PHP 4 version 4.4.0 and prior
- * PHP 5 version 5.0.5 and prior

[UNIX] PHP Multiple Vulnerabilities (File Upload, parse_str() register_global bypassing, phpinfo XSS)

Securiteam: [UNIX] PHP Multiple Vulnerabilities (File Upload, parse_str() register_global bypassing, phpinfo XSS)

Immune Systems:

* PHP 4 version 4.4.1

File-Upload \$GLOBALS Overwrite:

In PHP 4.3.11 some code was added to disallow overwriting the \$GLOBALS array when register_globals is turned on. Unfortunately there was a hole in this protection. The introduced code did only affect the globalization of the GET, POST and COOKIE variables. However it was overseen, that the rfc1867 file upload code within PHP also registers global variables, which can be used by an attacker to overwrite the \$GLOBALS array by simply sending a multipart/form-data POST request containing a fileupload field with the name 'GLOBALS'.

Until now it was not realized, how dangerous the problem is. This is also one of the reasons why all PHP <= 4.3.10 packages shipped with various distributions are still vulnerable to the normal \$GLOBALS overwrite, which was fixed in PHP 4.3.11.

Describing the impact of \$GLOBALS overwrite vulnerabilities and why it does not only affect installations, where register_globals is turned on, why it allows remote code execution in a lot of PHP applications and why this is also a threat for applications that allow local file includes and are running in a SAFE_MODE or open_basedir environment is out of the scope of this advisory.

register_globals Activation Vulnerability in parse_str():

When parse_str() is called with only one parameter it parses the supplied string, as if it were the query string passed via a URL and sets variables in the global scope. This is achieved by internally switching register_globals on, while the string is parsed.

Unfortunately it could be possible for an external attacker to trigger the memory_limit request termination during such a call to parse_str() by sending a lot of request variables to consume enough memory to trigger the limit. (It is described elsewhere how it is possible to consume a lot of memory with a small request body). If the request shutdown is executed during the call to parse_str() the register_globals flag is left on, for the rest of the lifetime of the involved web server process.

Because the flag is only internally changed and this has nothing to do with setting ini variables, the script is not able to detect that register_globals is on in an easy way. This tricks a lot of register_globals deregistration layers, because they usually only get activated when the ini_get() functions returns that register_globals is turned on.

This vulnerability is rated low, because calls to parse_str() with only one parameter are very rare. Additionally even if register_globals is turned on without the script knowing, this is only a security problem if the affected script does not properly initialize its variables.

Securiteam: [UNIX] PHP Multiple Vulnerabilities (File Upload, parse_str() register_global bypassing, phpinfo XSS)

XSS in phpinfo()

The phpinfo() function outputs a large amount of information about the current state of PHP. This includes information about PHP compilation options and extensions, the PHP version, server information and environment (if compiled as a module), the PHP environment, OS version information, paths, master and local values of configuration options and request variables, HTTP headers, and the PHP License.

Because phpinfo() leaks a lot of information to the viewer it is not recommended to leave a script executing phpinfo() on a production server. However in reality phpinfo() scripts are left open on a lot of servers. While this is already bad enough, there is also a problem when request variables of a certain form are displayed. With a properly crafted URL, that contains a stacked array assignment it is e.g. possible to inject HTML code into the output of phpinfo(), which could result in the leakage of domain cookies (e.g. session identifiers).

ADDITIONAL INFORMATION

The information has been provided by <mailto:sesser@hardened-php.net> Stefan Esser.

The original article can be found at:

<http://www.hardened-php.net/advisory_202005.79.html>
http://www.hardened-php.net/advisory_202005.79.html,
<http://www.hardened-php.net/advisory_192005.78.html>
http://www.hardened-php.net/advisory_192005.78.html,
<http://www.hardened-php.net/advisory_182005.77.html>
http://www.hardened-php.net/advisory_182005.77.html

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.