

[NT] PHP Input Validation Vulnerabilities (addslashes, Windows Only)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-12/0071.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 12/21/04

To: list@securiteam.com

Date: 21 Dec 2004 18:42:57 +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 Input Validation Vulnerabilities (addslashes, Windows Only)

SUMMARY

PHP is vulnerable to meta character attacks. The bug could enable an attacker to read arbitrary files from the file system of a web server that hosts PHP scripts. In addition newer versions of PHP contain a bug that enables an attacker to manipulate the file name of uploaded files to perform directory traversal.

While both vulnerabilities exist in Windows and UNIX platform versions of PHP, they can only be successfully exploited on Windows systems.

DETAILS

Vulnerable Systems:

- * PHP version 4.3.9 (arbitrary file reading)
- * PHP version 4.3.6 up to 4.3.9 inclusive and PHP version 5.0.0 up to 5.0.2 inclusive (directory traversal)

Immune Systems:

- * PHP version 4.3.10
- * PHP version 5.0.3

Securiteam: [NT] PHP Input Validation Vulnerabilities (addslashes, Windows Only)

addslashes() Vulnerability:

Scope:

PHP version 4.3.9 contains a bug in the function addslashes(). addslashes() can be used to sanitize userinput and render it thus impossible for an attacker to influence scripts by injection meta characters. In the default configuration, magic_quotes_gpc is set to "On" which auto-magically performs addslashes() on every input value. However because of a bug, the NULL byte is not correctly encoded by addslashes, enabling an attacker to read arbitrary files from the file system, if user input is used within include() or require() directives.

Details:

Addslashes should turn a NULL byte (will be written as %00 in this advisory) into the string "\0" (backslash zero). In version 4.3.9 the NULL byte is encoded as "%00" (backslash null byte). Everything after the NULL byte is ignored in include and require directives so that an attacker can truncate the name of the file that is included in the PHP script. The last character however will always be the backslash. As in Windows the backslash is the path delimiter, this does not matter – the file named before the backslash is still loaded.

Example:

Consider the following PHP script:

```
<?
$whatever = addslashes($_REQUEST['whatever']);
include("/path/to/program/" . $whatever . "/header.htm");
?>
```

A malicious attacker might open the following URL, disclosing the boot.ini file:

<http://localhost/phpscript.php?whatever=../../../../boot.ini%00>

The trailing backslash from the escaped \%00 does for some reason not seem to be of concern to include().

Upload Path Traversal Vulnerability:

Scope:

PHP automatically sanitizes the file name of uploaded files removing everything before the last slash or backslash. This is done in order to prevent path traversal attacks with uploaded files. However if an attacker uploads a file containing a single quote and the attacked web server has magic_quotes turned on (which is default configuration) or performs an addslashes() directive on the name of the uploaded file, the quote is prefixed with a backslash. This occurs after PHP checks for backslashes in the filename. As the backslash is the path delimiter in windows, this behavior enables an attacker to traverse the path by one directory level.

Example:

If a file with the name "..'file.ext" is uploaded, PHP turns the name to "..\'file.ext" and the file is uploaded to the directory below of where the PHP script copies it.

Securiteam: [NT] PHP Input Validation Vulnerabilities (addslashes, Windows Only)

Vendor Status:

The vendor has been timely informed and has released patched versions of the software (PHP 4.3.10/PHP 5.0.3). Those can be downloaded from <http://www.php.net>.

Timeline:

- Oct. 08: Notified vendor of addslashes vulnerability
- Oct. 14: Vendor reply
- Nov. 02: Notified vendor of upload vulnerability
- Nov. 04: Vendor reply
- Nov. 20: Problems fixed in CVS
- Dec. 14: Release of patched versions 4.3.10/5.0.3

ADDITIONAL INFORMATION

The information has been provided by <<mailto:research@sec-consult.com>>
Daniel Fabian.

=====

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.