

[UNIX] PHP 4.x Session Spoofing

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-01/0077.html>

From: support@securiteam.com

Date: 01/15/02

From: support@securiteam.com

To: list@securiteam.com

Date: Tue, 15 Jan 2002 23:21:11 +0100 (CET)

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

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> -- Know that you're safe.

PHP 4.x Session Spoofing

SUMMARY

The default configuration of PHP 4.x does not store its SessionID in a secure environment but rather in the /tmp directory. This would allow an attacker with access to the /tmp directory to spoof existing SessionIDs, thus gaining privileged access to other people's accounts.

DETAILS

Background:

What are sessions:

A session ID is required to identify people. It is passed over to the browser and then is either part of the url or is stored as a cookie. With every request, the browser also sends this ID over to the server that makes it possible to see which requests came from which user. Using the IP is not reliable for identification, because many people can come over a proxy and have the same IP.

Sessions are now also used for authentication purposes. Because there is no reliable way of keeping a permanent connection to the user, a login procedure is simulated using sessions. As long as the user is "logged in", the session-ID replaces any user/password combination. Because session-IDs are difficult to predict (that is why they are so terribly long), they are considered secure.

Securiteam: [UNIX] PHP 4.x Session Spoofing

Session support in PHP:

Since PHP4, there is a native support for sessions, which was derived from the PHPLib. Instead of using a SQL backend to store these IDs, they chose to store them as files in /tmp.

Every session is stored in a file like

```
sess_g35g5g54gg45wg85
```

Where "g35g5g54gg45wg85" is the actual SessionID. Someone could now easily spoof these sessions, because he now knows the IDs. He would even be able to read the contents of these files, because PHP very often runs as module (i.e. every executed PHP script inherits the user permissions of apache), thus you only have to write a PHP script which reads out these files.

Workaround:

It is suggested that you create a directory called

```
mkdir /tmp/php_sessions/
```

You have to adjust the path in php.ini for this. Then chown it to apache

```
chown www-data: php_sessions
```

And make sure to take away "r". r means "listing a directory". Apache only has to be able to "go into it" = x = 1, and "write" = w = 2. 1 + 2 = 3, so

```
chmod 300 php_sessions
```

Now, although apache is able to create and read sessions, it is not anymore possible to list the directory.

The PHP-developers are informed about this issue, and there is a discussion about various security issues in PHP-Dev.

ADDITIONAL INFORMATION

The information has been provided by Michel Lang and
<mailto:daniel@lorch.cc> Daniel Lorch.

=====

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.

Securiteam: [UNIX] PHP 4.x Session Spoofing

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

- **Previous message:** support@securiteam.com: "[NT] Internet Explorer SuperCookies P3P Bypass and Cookie Controls"
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)