

Forum:[Hack Forums](#)

Thread: [\[Newb-friendly\] Pentesting Admin Panels \[3 ways & Demo target\]](#)

Th3PonyWizard - September 23, 2013, 10:59 am

Note: This tutorial originally was written back when I still was a part of Legion, you know, before they turned out to be crap after the so called "Cleanup". I promised I'll leave it there for a couple of months, which I did. This isn't exactly new, I have already a shit tons of other stuff to add which I won't (XML Injections, XSS tamperation and exploitation, DNS Hijacking), oh well. This is REALLY Newb-friendly, I'm going into some really basic shit that I feel isn't being stressed enough, while still being there once a while.

Note: This tutorial requires you to know how PHP data Methodologies work (\$_GET, \$_POST), in order to understand whatever I'll be presenting. If you don't, you might get confused at some point hence I'd recommend learning PHP before reading this. Worst case scenario I will answer any questions to some degree.

This tutorial can be repeated live on my website. <http://Th3PonyWizard.me/HF/>

Have it open while reading, you'll have the option to repeat the steps yourself to get a better grasp.

Want to see all of the source code, for any page? You know, how shit is actually behaving server-sided? Now you can, use my source.php

Like this - <http://th3ponywizard.me/HF/source.php?file=source.php>

Sup. At this little write-up I'll bring something more technical than usual, which is a thing I tend to avoid at during my writeups. Oh well. Today we shall look upon 3 very common security holes newbie programmers tend to leave around: Not properly checking session, using unreliable data transferring methods (The PHP Referer Header), and some SQLi for the finish. Lets begin.

Chapter 1

What ARE sessions?

PHP has been supporting the \$_SESSION in a very comfortable and easy way; It's job is to save temporary information about the surfer while he is surfing. After a configured amount of time in which the user is in-active (I believe the default is when he closes his browser), the session value is being deleted.

Scenario: You're logging in to facebook (fagget). If you didn't press "Remember me", your browser will not receive a cookie, but only a session ID name which is something like "Logged-in". The server keeps the "logged-in" value as TRUE (1), as long as you're browsing facebook. If you closed the window of the browser, the browser will drop the session and you'll have to re-log, unless you pressed "Remember me" which created an encrypted two-sided token that facenook issued you, which allows you to connect to your user without putting the user & pass coords. (Unless they're retarded and put the user+pass in the cookies but I doubt it. Maybe later on I'll do a tutorial about Cookie and Session hijacking, but that's a completely different thing~) The session is being stored almost-completely server-sided, unlike cookies which are being stored client-sided. The only part of the session that is being stored client-side is the ID name itself, but not it's value.

We got that? Good. PHP codes with sessions will always start with this function

PHP Code

```
<?php
session_start();
```

So, how do we implement those sessions into our beloved login screen? Have a look at the one I coded.

This tutorial's live example address is <http://Th3PonyWizard.com/HF/>
Lets look at my page, which implements a session use.

<http://th3ponywizard.me/HF/login.php>

If at any point, you have any interest to view the source code (Both client and server-sided of my pages), navigate through

<http://th3ponywizard.me/HF/source.php>

And use 'file' as a GET parameter to your wanted file, like this.

<http://th3ponywizard.me/HF/source.php?file=login.php>

Lets copy that one off.

PHP Code

```
<?php
session_start();
if ($_SESSION['admin'] == 1) {
    session_destroy();
}
?>Code <html>
<head><title>Ugh</title></head>
<body>
<form method="post">
Password: <input type="password" name="pass">
<input type="submit" value="submit" name="sub">
</form>PHP Code
<?php
session_start();
if ($_SESSION['admin'] == 0) {
    if (isset($_POST['pass'])) {
        if ($_POST['pass'] == "HF") {
            $_SESSION['admin'] = 1;
            echo "Horray, you're logged in! Go to <a href=\"admin.php\">DIS LINK</a>";
        } else {
            echo "Nope";
        }
    }
    else {
        echo "You're already logged in. Press submit one more time, it'll log you off.";
        if(isset($_POST['sub'])) {
            $_SESSION['admin'] = 0; } } ?>This, although being quite clumsy (Paradon) is your average example
of a login screen that applies sessions into it.
```

Couple of things to note:

- 1) You see the first part of the code? It checks whether the user is already logged in. If he is, it will destroy it's session thus forcing him to log back in.
- 2) couple of IF statements at the buttom, they check whether:
 - a. You're not admin. If you are, your session destroys thus you're not.
 - b. If there's actually a value passed into the password user input.

c. If the passed password value is the true one

If all of those three are getting TRUE, our page creates the session to our user and changes it to 1 (true), we're moving to admin.php.

<http://th3ponywizard.me/HF/source.php?file=admin.php>

PHP Code

```
<?php
session_start();
if ($_SESSION['admin'] == 1) {
    echo "This is the admin panel, just without the panel.";
}
else {
    die("login through login.php"); }
?>
```

As you can see, there's a session check (That session we created at login.php in case the dude got the pass right) whether our user has the admin session as "1" = true. If he does, he sees the "admin panel". If he doesn't, he gets the other message.

This is an admin/login system done right. Look at a admin page done NOT right.

<http://th3ponywizard.me/HF/source.php?file=admin2.php>

PHP Code

```
<?php
/*There's no inclusion for the footer.php file here, nor any actual check whether the session is in place*/
echo "This is the admin panel, just without the panel";
?>Yes. This is it. I've wrote a complete damn tutorial about guessing admin-related URLs and hoping they're not checking sessions from the login system. This is quite a common issue, programmers forget to check the dude if he's admin before presenting him the page, and as a result you get in and do whatever you want. The best solution is to create a footer.php file, have this or some shit
```

PHP Code

```
<?php
session_start();
if ($_SESSION['admin'] != 1) {
    die("LOGIN, DUMBASS"); }
?>and include(footer.php) it into any admin-related file.
```

Chapter 2

The HTTP protocol has a very interesting parameter called REFERER.

How do you make a good CEO-based to your website, to check to see where are your user traffic-income is coming from? You find out their HTTP referer.

I can go into detail about why, how and when it exists, but why? Wikipedia did a great job summoning it up.

https://en.wikipedia.org/wiki/HTTP_referer

And if you happen to check my challenge, I based it on the same thing -

<https://hackforums.net/showthread.php?tid=3645138>

(Nobody except one teacher and one student solved it. Great.)

So lets get working. Oh wait, I'd recommend getting either: Hackbar (Firefox addon) / Tamper data (Firefox addon) / Live HTTP Headers (Firefox addon). For this, the best IMO is hackbar but anything

that allows you to edit your user agent is good.

Lets check our page.

<http://th3ponywizard.me/HF2/source.php?file=login.php>

PHP Code

```
<?php
session_start();
if ($_SESSION['admin'] == 1) {
session_destroy();
}
?>Code <html>
<head><title>Ugh v2</title></head>
<body>
<form method="post">
Password: <input type="password" name="pass">
<input type="submit" value="submit" name="sub">
</form>PHP Code
<?php
session_start();
if ($_SESSION['admin'] == 0) {
if (isset($_POST['pass'])) {
if ($_POST['pass'] == rand()) {
$_SESSION['admin'] = 1;
echo "Horray, you're logged in! Go to <a href=\"admin.php\">DIS LINK</a>";
} else {
echo "Nope";
}}}
else {
echo "You're already logged in. Press submit one more time, it'll log you off.";
if(isset($_POST['sub'])) {
$_SESSION['admin'] = 0;} } ?>Looks familiar? This is our (recycled) login page. Ignore the sessions,
they're not relevant for us now. Lets skip to the admin.php page.
```

<http://th3ponywizard.me/HF2/source.php?file=admin.php>

PHP Code

```
<?php
if(isset($_SERVER['HTTP_REFERER'])) {
if ($_SERVER['HTTP_REFERER'] == "http://th3ponywizard.me/HF2/login.php") {
echo "This is an admin panel. If you're seeing this, you managed to either log in or hack yourself in"
; }}
?>
```

Bingo. Look at our first if statement, we first check if there's any actual input to the HTTP_REFERER, because if it's empty yet you print it out you'll get an Undefined Index error. So, presuming we have a value into the REFERER, we continue to our second IF statement, which basically checks if our REFERER is the login page.

What's the logic? It's simple. If the user came from the login page to the admin page, it's likely to presume he just logged in (programmer logic here) so yeah. We know that REFERER is part of the user agent data that's part of the HTTP packet that your browser always sends, so we know we can edit it.

Get into <http://th3ponywizard.me/HF2/admin.php>

Open HackBar, press "enable referer".

We did we came from? That's right, <http://th3ponywizard.me/HF2/login.php>

Quote: This is an admin panel. If you're seeing this, you managed to either log in or hack yourself in Bang, we're in.

Chapter 3 (Bonus!)

<http://th3ponywizard.me/HF3/>

SQL Injections. This is probably known, and will make most of you guys roll eyes, but shit is still active as long as people won't bother switching to PDO/MySQLi/Filter user input. SQLi has been there for ages and isn't planning to move anywhere.

<http://th3ponywizard.me/HF3/login.php>

As usual, our code is:

PHP Code

```
<?php
session_start();
if ($_SESSION['admin'] == 1) {
    session_destroy();
}
?>Code <html>
<head><title>Ugh v3</title></head>
<body>
<form method="post">
    Password: <input type="password" name="password">
    <input type="submit" value="submit" name="sub">
</form>PHP Code
<?php
session_start();
$con = mysql_connect('localhost', 'thponywi_pony', '123456');
mysql_select_db("thponywi_HF");
$password = $_POST['password'];
$query = "SELECT * FROM `hue` WHERE `password` = '$password'";
$query2 = mysql_query($query) or die(mysql_error());
if ((mysql_num_rows($query2)) > 0) {
    $_SESSION['admin'] = 1;
    echo "Horray, you're logged in! Go to <a href=\"admin.php\">DIS LINK</a>";
}
else {
    echo "nope";
}
if(isset($_POST['sub'])) {
    $_SESSION['admin'] = 0;
}
mysql_close();
```

?>Ah, yes. Implementation of an SQL system.

Lets look a bit more closely, specifically, here:

PHP Code

```
<?php
$password = $_POST['password'];
$query = "SELECT * FROM `hue` WHERE `password` = '$password'";
```

You don't need to be a genius to figure out this is inject-able :3

The user input labeled as password (back from our HTML form) is being imported directly into our query instruction, without any validation / filtration checks, everything goes inside dangerous or not.

Before proceeding, lets check our MySQL database and table.

More on SELECT statement: <http://dev.mysql.com/doc/refman/5.0/en/select.html>

Code SELECT *

FROM `hue` Database thponywi_HF

Table structure for table hue

Column Type Null Default

id int(11) No

admin text No

password text No

Dumping data for table hue

1 Th3PonyWizard HF

Pass is HF. Lets check it out at our login.php page.

<http://th3ponywizard.me/HF3/login.php>

Quote:Horray, you're logged in! Go to DIS LINK

As you can see, everything is functioning. Now, knowing there's room for an injection, lets try out one. Going back to our code:

PHP Code

```
<?php
```

```
$password = $_POST['password'];
```

```
$query = "SELECT * FROM `hue` WHERE `password` = '$password'";
```

We can note that it is a STRING based. Why? Because the variable is being put in-between quotes just like a string, which is what it is. Hence we will first close the string by injecting our own quote. First of all I'd recommend tempering and messing around with what we have, just for the sake of observation.

Code password='output:

Quote: Password:

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 ' ' ' at line 1

The error is caused because of a non-dual amount of quotes at the string: 3, look:

```
$query = "SELECT FROM `hue` WHERE `password` = ' ' '";
```

The pink quote is the quote we injected, it closes our string.

The red quote is part of the script, but because of the already-inserted pink quote, the quote re-opens the string without closing him with another quote hence we get a syntax error.

Note: Don't rely on errors. You see that one only because I used a condition, whether the query fails thus an error will show up to the visitor. At some injections, the output error will not be shown.

After vafirying it's indeedy an SQLi, string based, we're going to inject something along the lines of Code password=something' OR '1' = '1

Explanation

```
SELECT FROM `hue` WHERE `password` = 'something' OR '1' = '1'";
```

We did something that's quite simple: Beginning with providing an obviously-false password, but then closing our pseudo-string and providing a condition. The SQL condition is OR, on which you can learn here:

http://www.w3schools.com/sql/sql_and_or.asp

Basically, it checks whether each of the two parts of the query is true. If one of the parts of the query's statements in each side is true, then the OR statement will return TRUE. Obviously, the

"something" pass is wrong, but 1=1 is always 1=1 so it's always TRUE, hence it returned TRUE.

WRONG PASSWORD. <- IF THAT IS WRONG, THEN CHECK THIS: 1=1.

So, trying it?

Quote:Horray, you're logged in! Go to DIS LINK

We're in.

Mini-Chapter 4 (Tips 'n notes!)

1) Everything is logged. Most HTTPD services (Apache, IIS, nginx) will save up access logs. Going CTRL+F for /admin.php and crossing IPs is pretty damn easy to get you caught.

2)Google is your friend. Applying the site: filter and combining it with other filters can easily reveal the admin login page. If it's not cached, you're out of lock, but check robots.txt files just in case. Retard believe it's better to block google bots from caching the login page whilst giving it more attention with the robots file.

3) Get your IPs ready. A properly coded login page will block the IP after a numerous failed attempts.

4) Check all user-provided input, whether you're the sysadmin blocking or the attacker, well, attacking. You might be sending "innocent" POST / COOKIE values such as "is_admin" which is more than under-stable how to abuse if possible. You'd be surprised but yes, this exists. User input is the devil, keep messing with it.

5) Brute-force is usually NOT something you want to consider. Get a formula that's consisted of your PC sending an HTTP request with your BF, the server's time to figure out and send a response, and (if needed) the amount of time it takes you to log in to a new proxy. You can figure out it'll take a couple hundred years to hack it, and that's assuming he only used English, numbers, and special chars. Yeah, it depends on the flags and method of BF, but it's usually not effective at all.

6) Check source code. The whole login system might be laying around completely client-sided (/w Javascript for example), which is standing available for your alteration or at least for your observation.

##EOF##

This text has been written by me (Th3PonyWizard) with the GNU Free Documentation License, allowing you to share, modify, store, and do whatever the fuck you want with this, but selling it isn't an option. (Who the F will buy this anyway)

Huge thanks for Zixem for patching up my code and making it as retarded as it should be. (<3)

All testing, writing, and producing was done at my own website, under the explicit allowance of my box owner. I do not endure nor give two fucks about what you do with this knowledge, nor I responsible in case you'll fuck up and you will, sooner or later.

Terry A. Davis - September 23, 2013, 12:50 pm

"selling it isn't an option. (Who the F will buy this anyway)"

tree fiddy for dis her wrk!

On topic, I loved it all and at least I walked away having learnt something.
Thanks for a very nice share

Th3PonyWizard - September 23, 2013, 1:55 pm

(Sep 23, 2013 12:50 PM)Perfect Cell Wrote: [\[CITE\]](#) "selling it isn't an option. (Who the F will buy this anyway)"

tree fiddy for dis her wrk!

On topic, I loved it all and at least I walked away having learnt something.

Thanks for a very nice shareAt some point during writing I usually get bored and decide to shitpost all over the ending. Jokes aside, copyrights is srs bsns.

And sure thing m8.

•**Lucifer** - September 23, 2013, 1:59 pm

Looks like it took A LOT of time and effort

Ill read it when I got time but nice work :)

Terry A. Davis - September 23, 2013, 3:44 pm

(Sep 23, 2013 01:55 PM)Th3PonyWizard Wrote: [\[CITE\]](#) At some point during writing I usually get bored and decide to shitpost all over the ending. Jokes aside, copyrights is srs bsns.

And sure thing m8.

Even shit posting is good posting for a man with a scat fetish.

You know what? I will make a sollemn vow never to say taht again...

Aura - September 23, 2013, 3:52 pm

Excellent tutorial man !

Bookmarked for future references and to use, of course ! :)

Th3PonyWizard - September 23, 2013, 11:59 pm

Thanks guys. Bumping for future interest.

Melkor333 - September 24, 2013, 1:28 am

thhx alot for this Tut!

just bookmarked it =)

cheetos804 - September 24, 2013, 3:04 am

That sure will help me in the future! thanks, i am bookmarking this. :D

Zixem - September 24, 2013, 3:52 pm

(Sep 23, 2013 10:59 AM)Th3PonyWizard Wrote: [\[CITE\]](#) Huge thanks for Zixem for patching up my code and making it as retarded as it should be. (<3)owhhh <3

Spoiler ([Click to View](#))

PHP Code

```
<?php
```

```
$con = mysql_connect('localhost', 'thponywi_pony', '123456');
```

```
skids gonna skid. jez sayin'
```

Th3PonyWizard - September 25, 2013, 12:13 am

(Sep 24, 2013 03:52 PM)Zixem Wrote: [\[CITE\]](#) owhhh <3

Spoiler ([Click to View](#))

PHP Code

```
<?php
```

```
$con = mysql_connect('localhost', 'thponywi_pony', '123456');
```

skids gonna skid. jez sayin'
>implying MySQL service accepts remote connections
>implying thponywi_pony has UPDATE permission
fack off m8

HaxOr? - January 8, 2014, 10:19 am

Sick tutorial dude, good job! <3 <3

Th3PonyWizard - January 8, 2014, 10:22 am

(Jan 8, 2014 11:19 AM)HaxOr? Wrote: [\[CITE\]](#) Sick tutorial dude, good job! <3 <3*bumps your butt*
pls no touching only teasing

Im gone brah - January 8, 2014, 10:49 am

[img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img][img]http://fi.somethingawful.com/safs/smilies/b/3/ducksiren.001.gif[/img]

Aura - January 8, 2014, 10:51 am

Your webpage isn't working anymore newb.
All I get is you yelling at me, you meany ?

Quote:HELLO DIS IS A TEMPORARY PLACEHOLDER FOR SOMETHING THAT IS GOING TO BE MY PAGE YES I AM YELLING

Th3PonyWizard - January 8, 2014, 2:01 pm

(Jan 8, 2014 11:51 AM)Aur? Wrote: [\[CITE\]](#) Your webpage isn't working anymore newb.
All I get is you yelling at me, you meany ?it was never supposed to be working, I have nothing to put there s:

BeggFoMercy - January 8, 2014, 2:47 pm

Awesome thread bish , Keep it up :) .

Th3PonyWizard - May 8, 2014, 9:25 am

The tutorial is valid although the website is dead.