SQL Injection

Computer Security CSE 481

Tatiana Ensslin - November 10, 2015

Introduction

The goal of this lab is to become familiar with basic SQL injections in php code that will allow for an attacker to modify a user's info, and log in as a user without their password.

Turn Off the Countermeasure

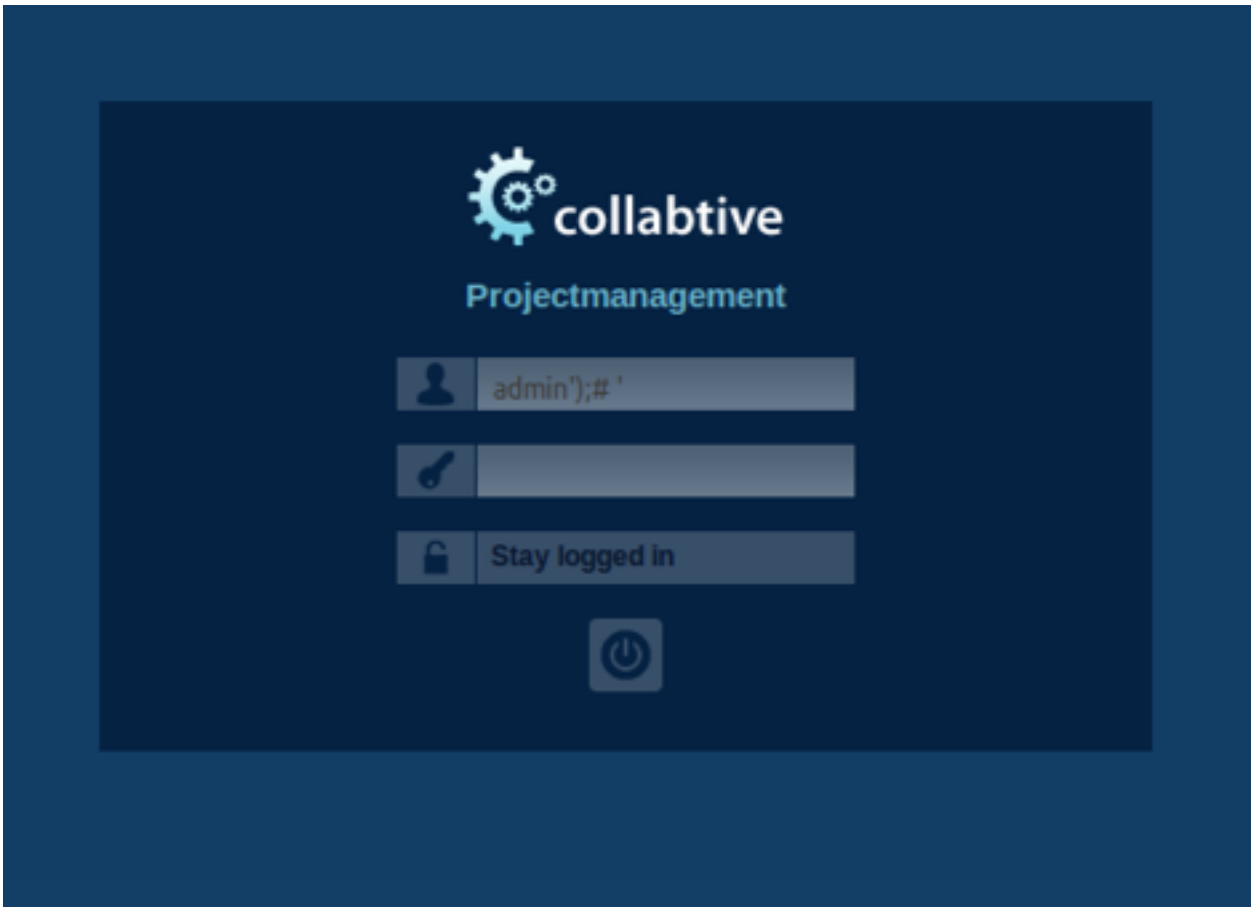
This is important because if you don't turn off the countermeasure and restart the server, your SQL injection won't work. This is a protective measure that needs to be turned off from the server.

```
[11/09/2015 18:52] seed@ubuntu:/etc/php5/apache2$ sudo gedit php.ini
^[[A
```

Task 1: SQL Injection Attack on SELECT Statements

By entering admin');# ' or admin' OR 1=1);# ' you can log in as admin without a password. This is because the # stops the parsing of the AND pass section. and the admin' fills in the first parameter, while setting the email section of $user to be true with 1=1;

It is not possible to find a way to modify the database.



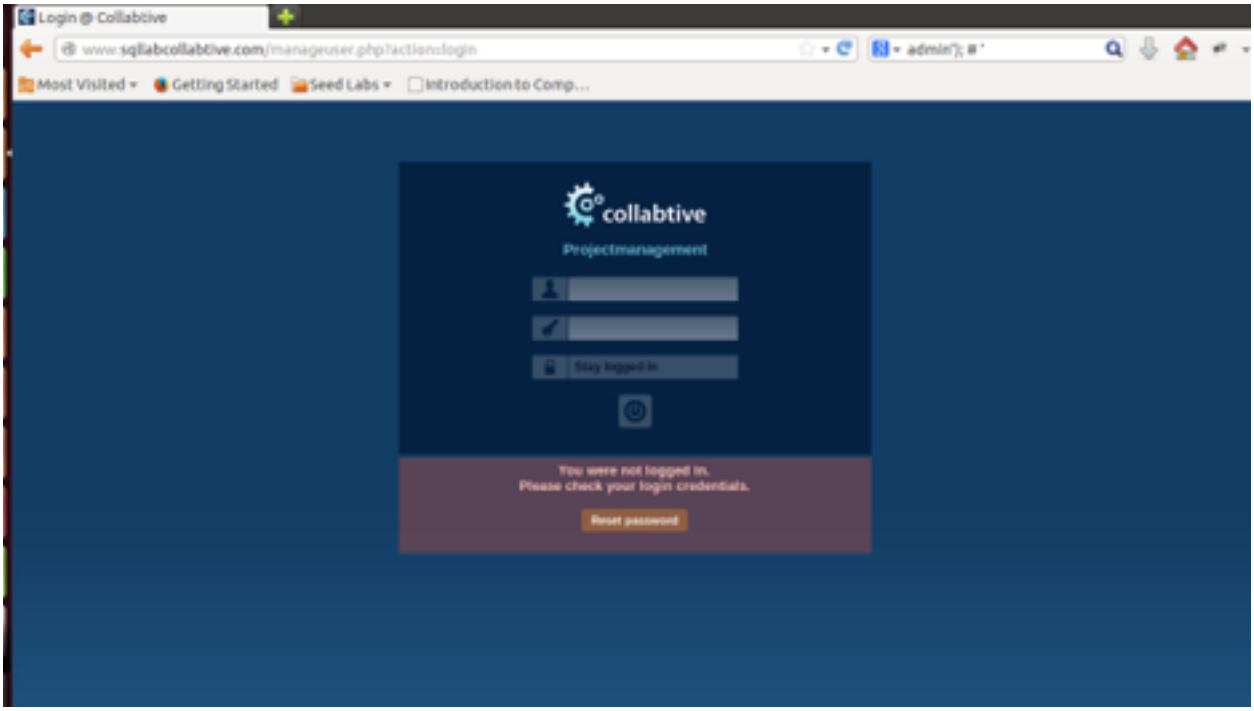
Task 2: Update Statement

For this section, we log in under a different user, and then update another users account info to allow us to log into their account with “pass.” For this, I logged in as Alice, and edited her info. I changed her name to ted in the edit form and entered: ', `pass` = '9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684' WHERE ID = 4 # ' for the company field. This was used in conjunction with knowing that the company field in the code was vulnerable to sql injection. I then logged out of Alice and was able to log in as Tes using pass in the password field.

Task 3: Prepare Statement

```
$db = new mysqli("localhost", "root", "seedubuntu", "sql_collabtive_db");
   $stmt = $db->prepare("SELECT ID,name,locale,lastlogin,gender FROM user
                    WHERE (name=? OR email=?) AND pass=?");
  $stmt->bind_param("sss", $user, $user, $pass);
  $stmt->execute();
  $stmt->bind_result($bind_ID, $bind_name, $bind_locale, $bind_lastlogin,
                  $bind_gender);
   $chk = $stmt->fetch();
      $chk = mysql_fetch_array($sel1);
         if ($chk["ID"] != '')
      {
         $rolesobj = new roles();
         $now = time();
        $_SESSION['userid'] = $bind_ID;

        $_SESSION['username'] = stripslashes($bind_name);
         $_SESSION['lastlogin'] = $now;
         $_SESSION['userlocale'] = $bind_locale;
        $_SESSION['usergender'] = $bind_gender;
        $_SESSION["userpermissions"] = $rolesobj->getUserRole($bind_ID);
```



Not allowing the admin');# ' statement or any sql injection to be done.

```
<!DOCTYPE html>
<html>
<body>

<img id="image" src="smiley.gif" width="304" height="228">

<script>
document.getElementById("image").src = "pic_mountain.jpg";
</script>

</body>
</html>
```