Lab10 SQL Injection Attack Lab

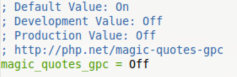
Yukui Ye

SUID: 439644268

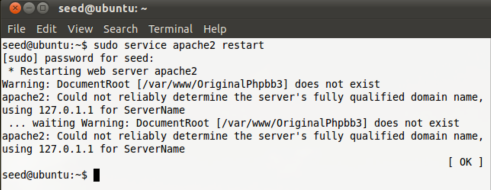
**Task1: SQL Injection Attack on SELECT Statements**

Setup: Turn off build-in SQL injection protection

Set magic\_quotes\_gpc value to ‘off’ in file /etc/php5/apache2/php.ini in order to turn it off.



Then restart it:

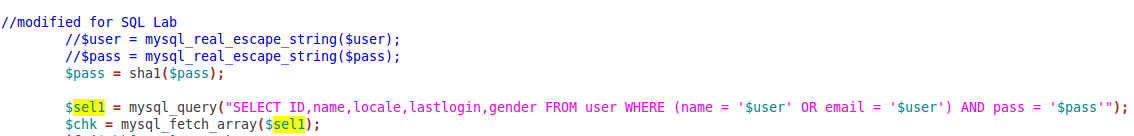


Task1.1: Log into another person’s account without knowing the password

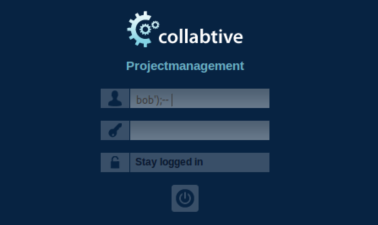
Check the login part code to see how it login works.

Macintosh HD:Users:yukuiye:Library:Application Support:QQ:Users:394852457:QQ:Temp.db:83455807-12E7-4351-9E7F-27674B0F1631.png

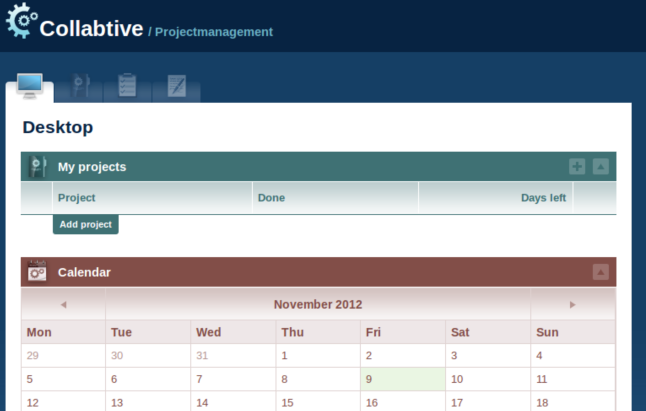
Following is the PHP codes run at server side.



From the above code, we can tell how server send query to Database. But we can still guess to login without knowing the user’s passward by typing “bob’);-- “( space after the comment’--’ )



By doing above thing, we successfully log into user’s account.

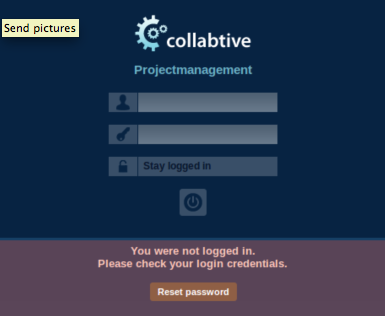


Task1.2: Find a way to modify the database.

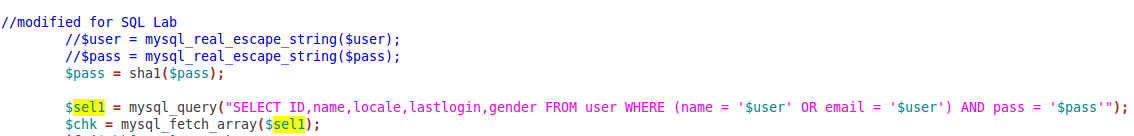
We can use semicolon to finish a SQL statement and then append update/insert/delete statement right after semicolon in order to modify the database by SQL injection. Therefore the database will receive two statements at the same time. I tried the following sentence in the log in window.

peter’);update user set company=’SQLinjection’ where id=4;--

It all turned out to be failed as the screenshot showed below.

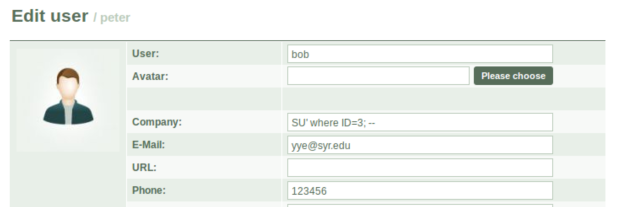


REASON: First check the PHP codes again:

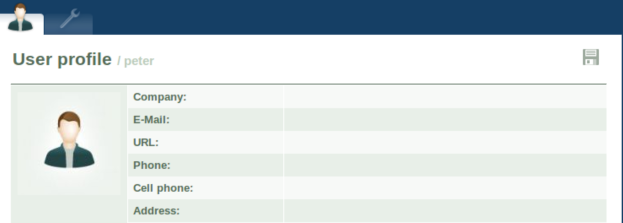


By checking the code, we can see that it uses mysql\_query to send statement to database. But mysql\_query can only send one query to database at one time to run. Multiple queries are not supported. Therefore, it not allow to select and update in the same time when using mysql\_query.

**Task2: SQL Injection on UPDATE Statements**

Step1:The main idea is the same as task1, which is to achieve the goal by using‘ -- ’to comment statement, and add our own where condition. First, we login as peter(peter’s original profile is nothing), change the user name to bob, and add “ SU’where ID=3;-- “at the field of company, AND email information as well as phone number. 

After we click send buttom, go to peter’s profile, still nothing just as original one.



Then we login as bob to check bob profile, we surprisingly find that all the information we write before in peter’s profile is showed up here. In one word, bob’s profile has been successfully modified.

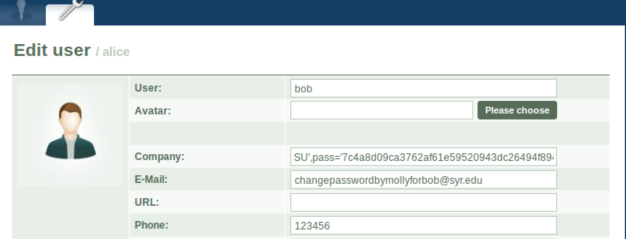


Step2: Now, we are trying to modify the password of bob. Login as alice and modify the profile. Change name to bob. ATTENTION: The password we are going to write in the company field should be hashed. By searching for the internet, I found that string” 123456” is “7c4a8d09ca3762af61e59520943dc26494f8941b”

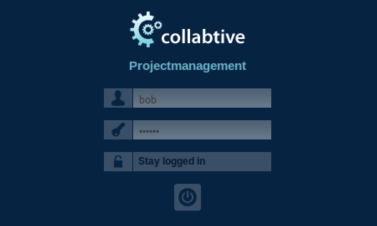
Macintosh HD:Users:yukuiye:Library:Application Support:QQ:Users:394852457:QQ:Temp.db:D143531F-2303-4C1D-BC86-1C951484DA60.png

In the company field, type like this :

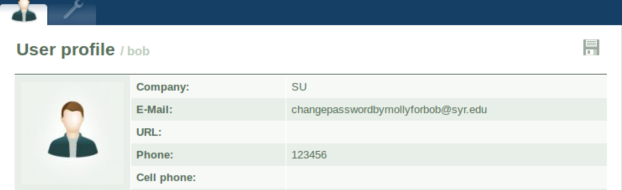
SU’,pass=’7c4a8d09ca3762af61e59520943dc26494f8941b’ where ID=3 --



Now login bob with our password 123456



We can successfully login and changed the bob’s profile.



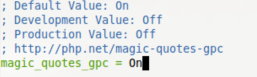
**Task3: Countermeasures**

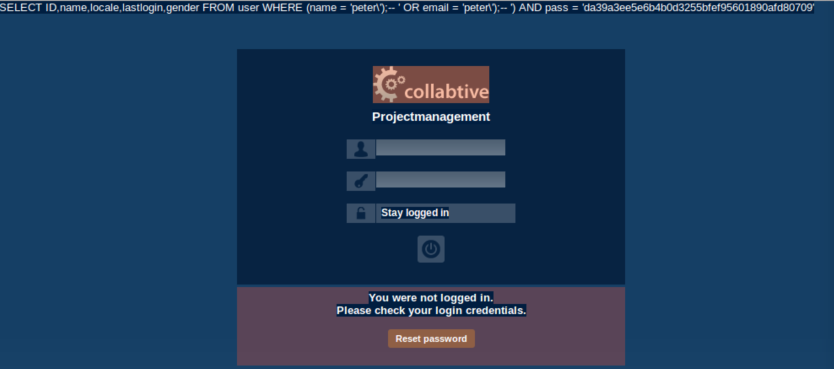
I add debug output information in login function at PHP file in order to what will happen with following countermeasures.



Task3.1: Escaping Special Characters using magic\_quotes\_gpc

Set magic\_quotes\_gpc to on, and restart server.

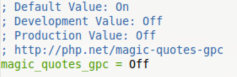


when you press “comment a” on the keyboard, it immediately show code on the top of the window.

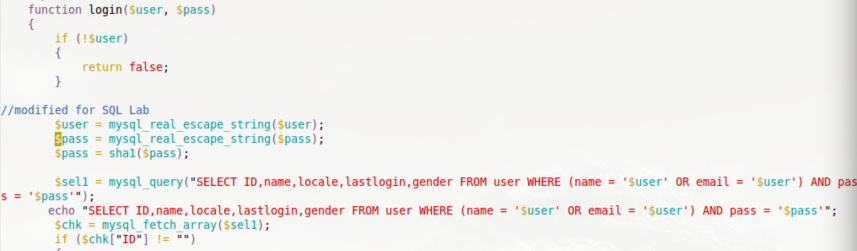
We could see there is blackslash before single quote that we input in user name field. This setting escape the quote automatically, therefore even if we use’--’to comment the statement, we cannot match the quote. Thus, we cannot login.

Task3.2: Escaping Special Characters using mysql real escape string

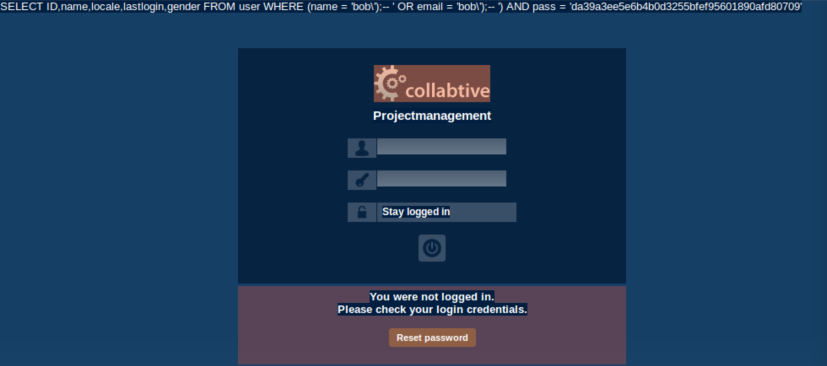
Set magic\_quotes\_gpc to Off.



Uncomment mysql\_real\_escape\_string in PHP file for login section.

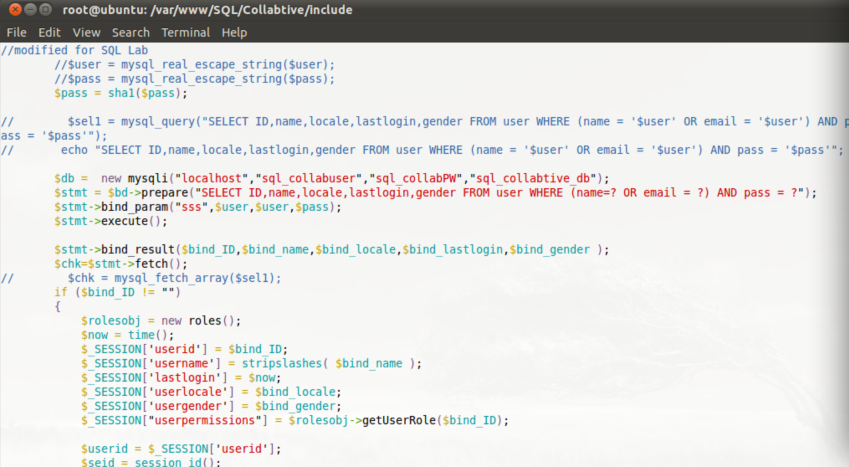


Do the same task in Task1, but we cannot login successfully, The code shows on the top of the window is the same as the previous task, there is blackslash before single quote that we input in user name field. Therefore, we cannot login.



Task3.3: Prepare Statement

The modified code for login section.



This mechanism separates the values from the statement, we cannot login a user without password. However we can still login as property. It sends values by parameters, and database treats each of them as data, it would not be part of statement codes.