Derrick Sung

**SQLInjection**

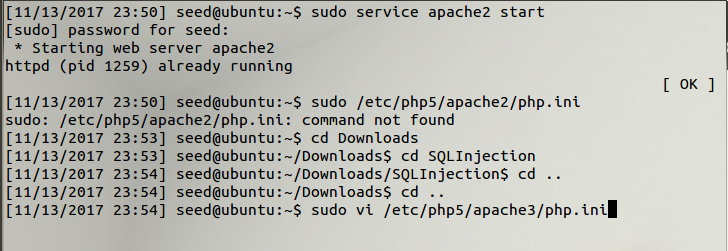
**NOTE\*: These are website references are used to make the code and SQL Injection that is screenshotted below, to work with the prepare statement.**https://stackoverflow.com/questions/43354342/sql-injection-example-php-and-mysqlhttps://websitebeaver.com/prepared-statements-in-php-mysqli-to-prevent-sql-injection

1. **Open terminal and start apache server:**

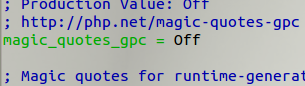
sudo service apache2 start

**Then turn off the counter measure:**

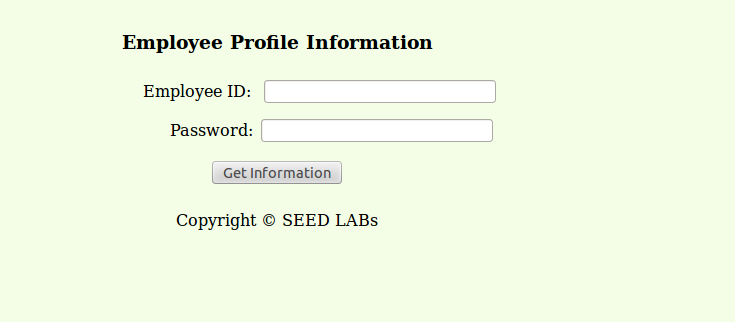
Sudo vi /etc/php5/apache2/php.ini



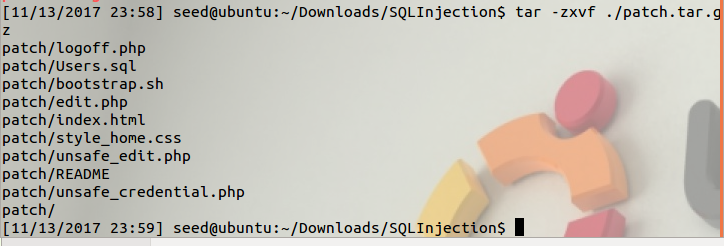
1. **Find the line containing magic\_quotes\_gpc and turn it off. Then restart the Apache server.**



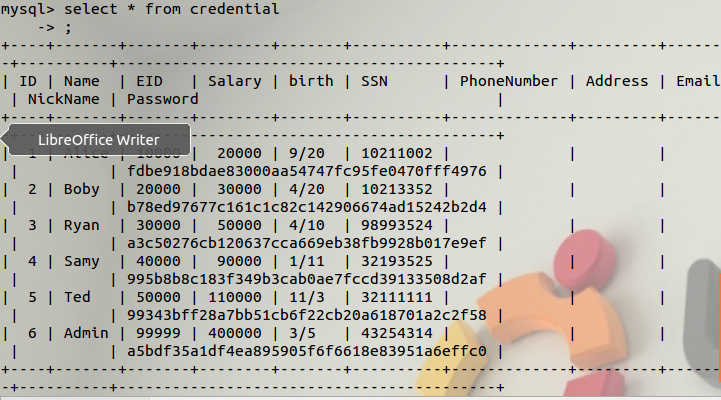
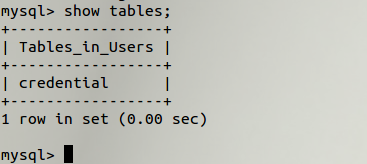
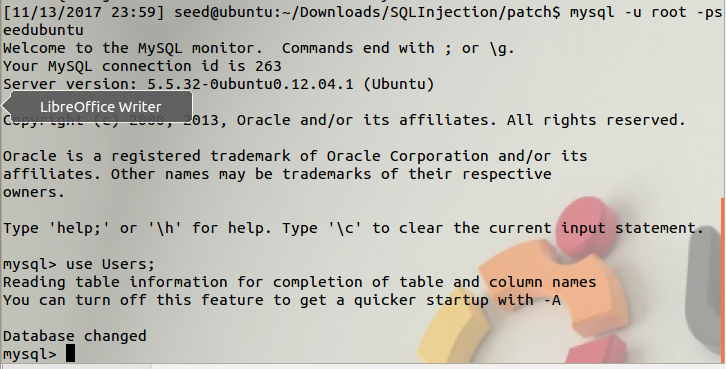
1. **Access the website: http://www.SEEDLabSQLInjection.com**



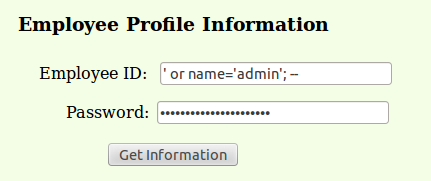
1. **Extract the tar file given from the SEED labs website.  
     
   boot up the .sh: ./bootstrap.sh**



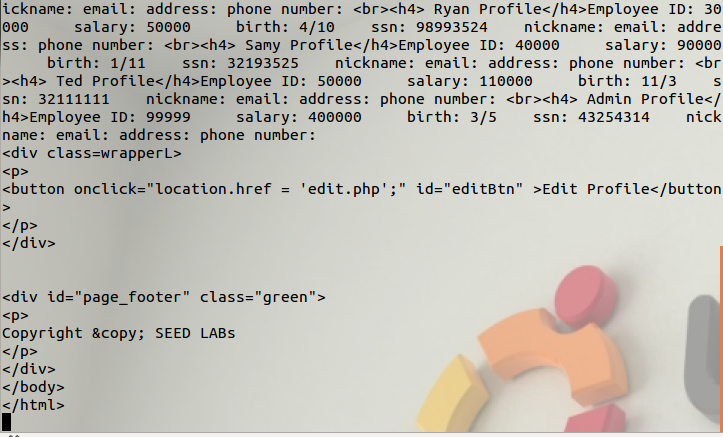
1. **Hop into MySQL:  
     
   Search for the tables of credentials, and search table for Alice.**



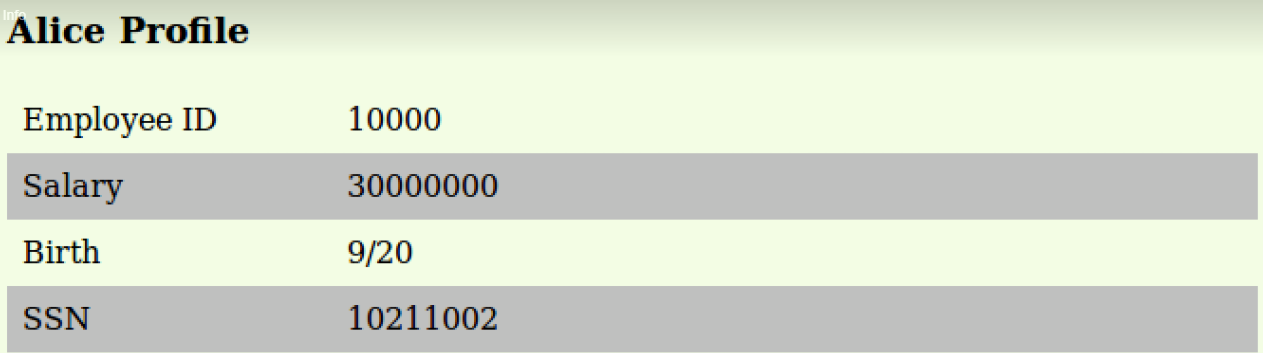
1. **Use SQL Injection on the website as admin.  
     
   You will be able to gain access to admin as such.**



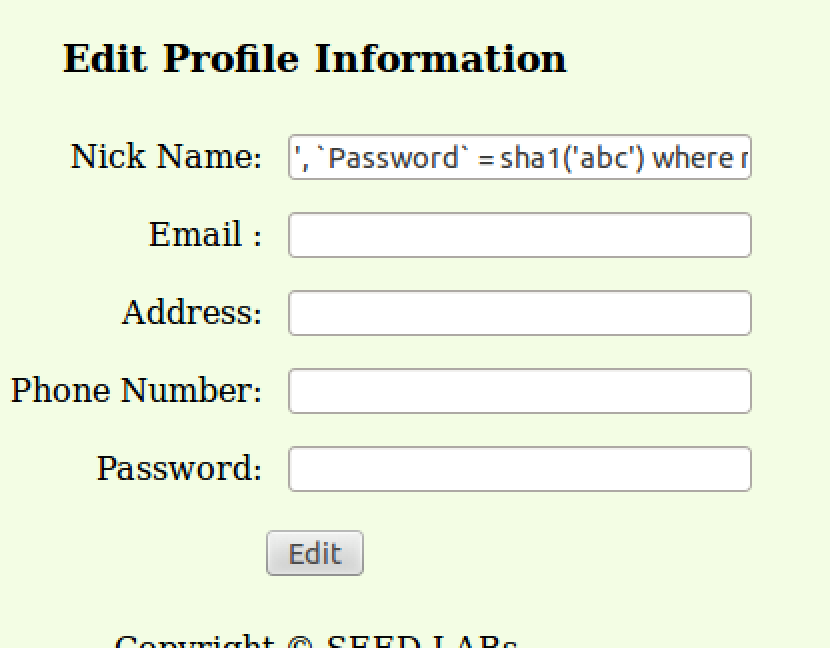
1. **Open a new terminal, and see if you can gain access to the website with curl.**curl ’www.SeedLabSQLInjection.com/unsafe\_credential.php?EID=’+or+name%3D’admin’%3B+--+&Password=’  
   **In the terminal, it should look like html code of the website.**



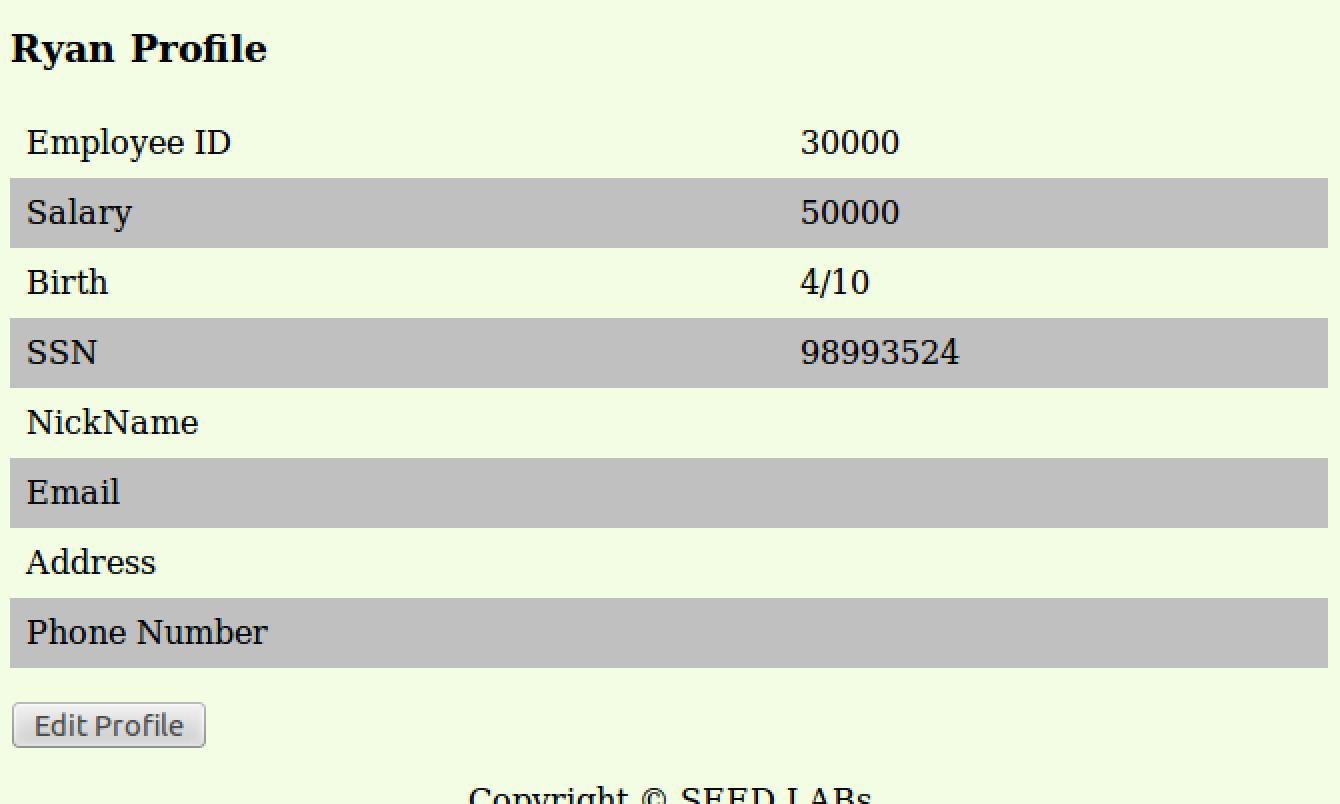
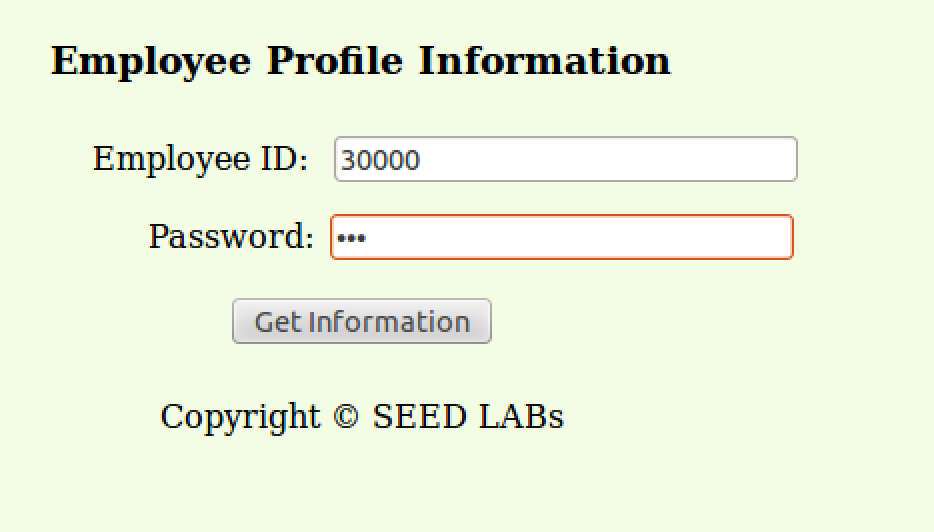
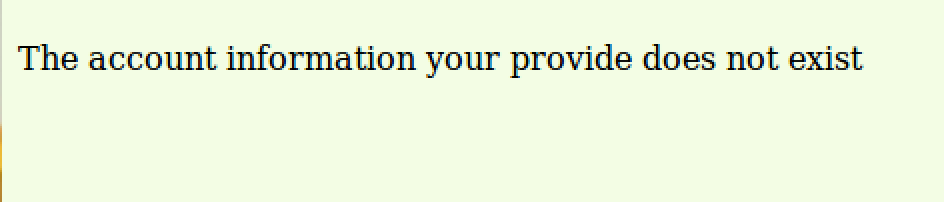
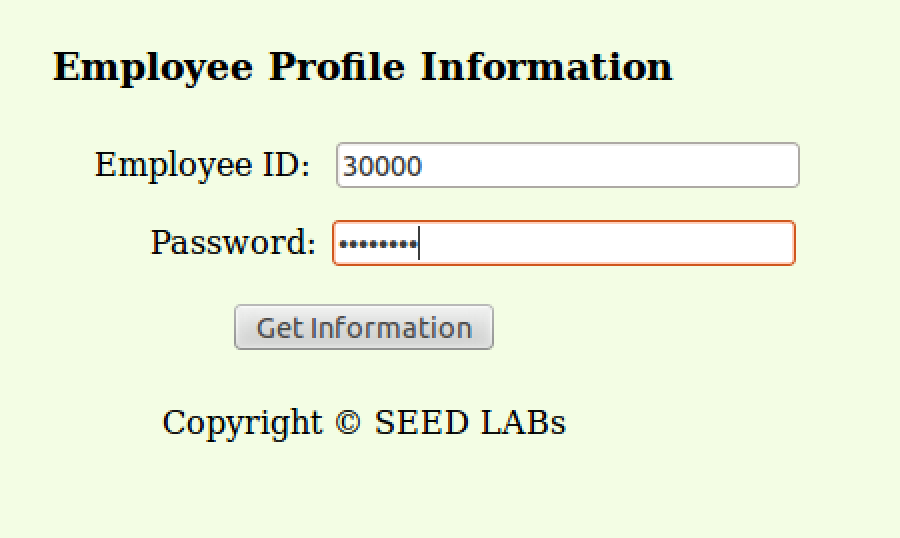
1. **Sign in as Alice, click ‘Edit Profile’, and use SQL Injection to update salary from 20000 to 30000000.**‘, `salary` = 30000000 where name =’Alice’ ; #’



1. **Use SQL Injection to change Ryan’s password (because we hate him.)**‘, `Password` = sha1(‘abc’) where name =’Ryan’ ; #’



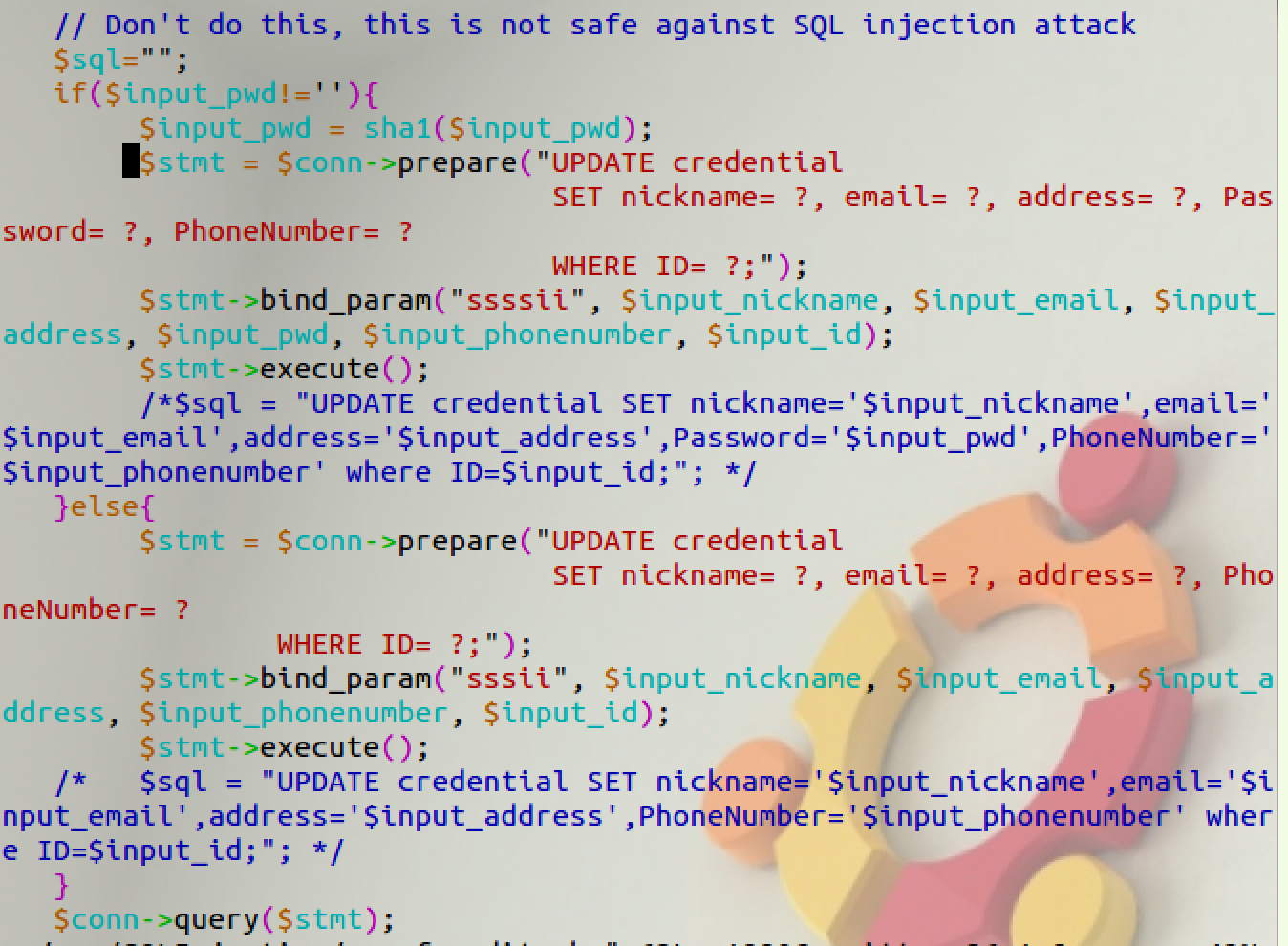
1. **Attempt to login as Ryan with his old password ‘seedryan’  
     
   It will show it does not exist.  
     
   Use the new password ‘abc’  
     
   It works!**



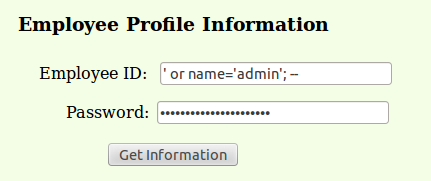
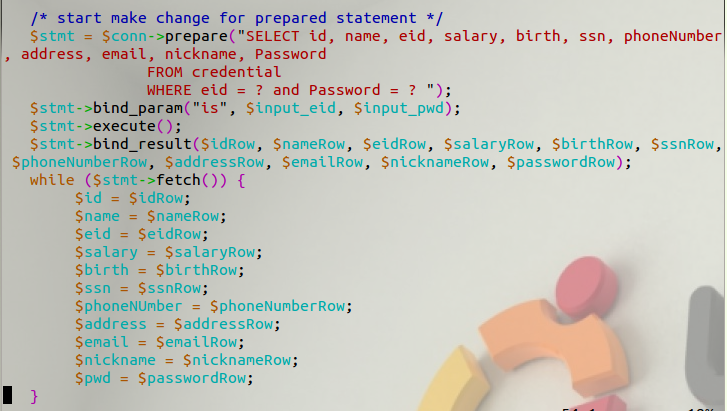
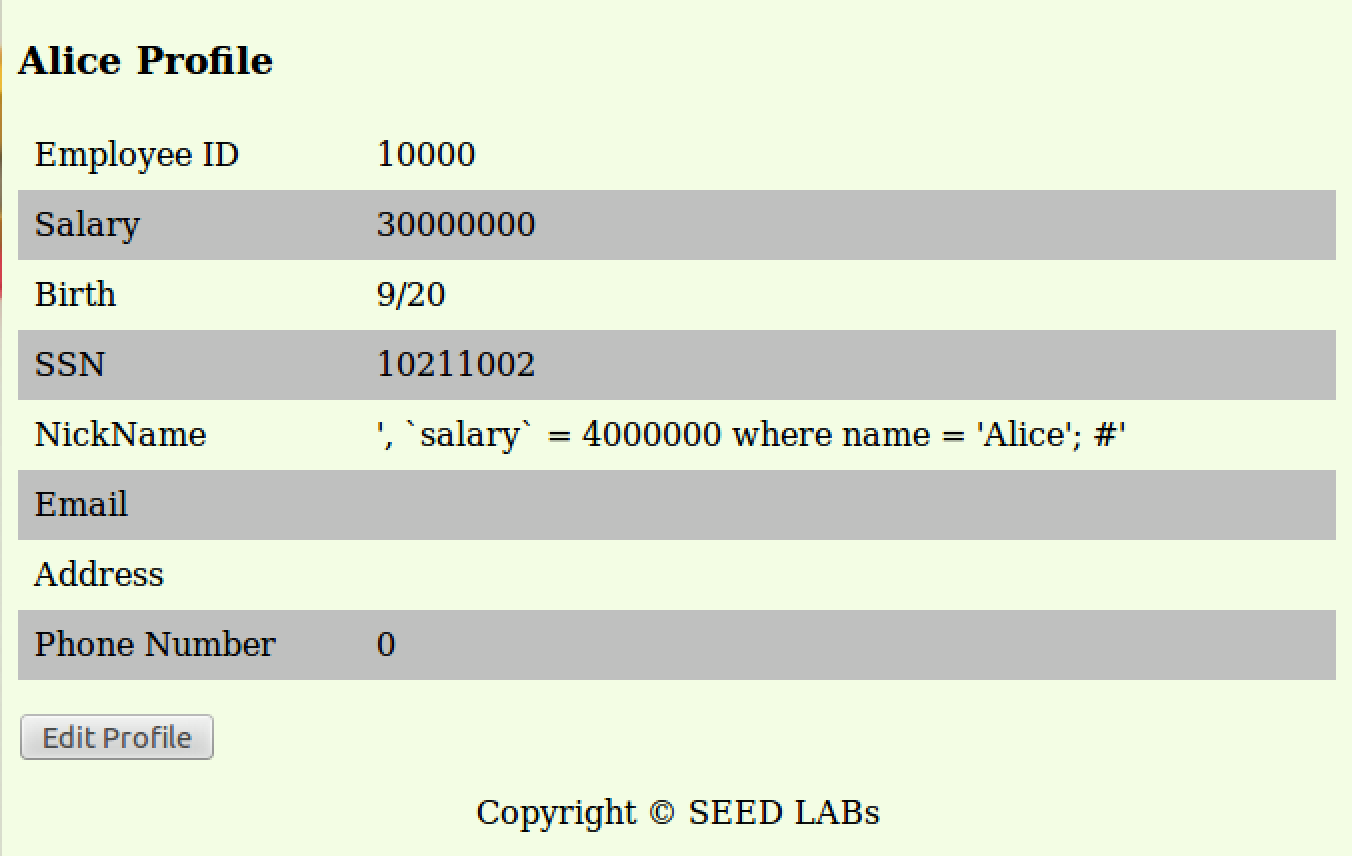
1. **To stop this from happening in the future, we must use prepare statements in our php code.  
   Head back into terminal and access:**

**sudo vi /var/www/SQLInjection/unsafe\_edits.php**

**Edit the Update code in the file to a prepare statement.  
   
Save.**



**This will stop Alice from attempting to update her salary or sabotage someone’s account. We can test this by going back to the edit profile page and use SQL Injection to update her salary again.  
  
  
It did not work. Her salary was not changed to a higher number.  
  
12. Next we can use a prepare statement to stop Alice from using SQLInjection to access accounts, and as admin.  
Edit the code in unsafe\_credential.php with a prepare statement.**sudo vi /var/www/SQLInjection/unsafe\_credential.php  
  
**Save.  
  
From here, this will prevent Alice from using SQL Injection to gain access to admin.**



**The website will show that the account information provided does not exist the next time you use SQL Injection to log in as admin.  
  
  
FINISH!**

