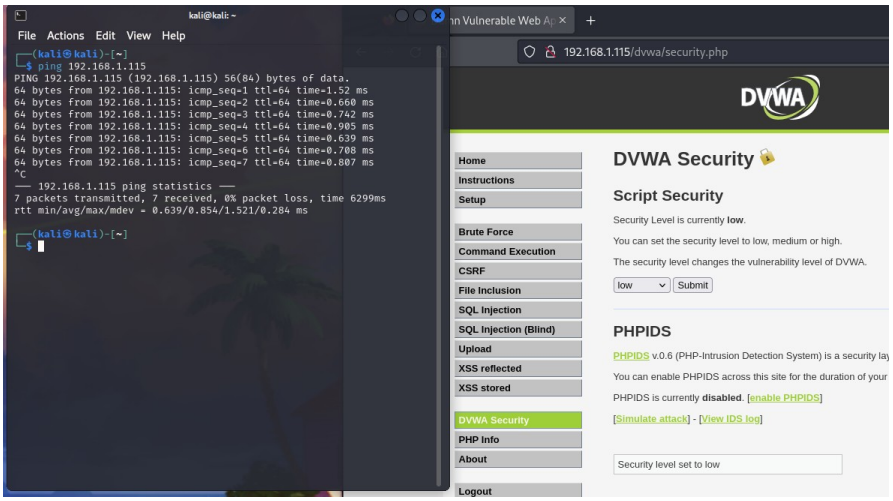
L’obiettivo dell’esercizio è l’analisi e lo sfruttamento di due tipologie di vulnerabilità comuni nelle applicazioni web:

XSS Riflesso (Cross-Site Scripting): Esplorare come un sito web possa eseguire codice JavaScript malevolo inserito tramite input utente.
SQL Injection (SQLi): Testare l’iniezione di comandi SQL per accedere a dati sensibili non protetti e comprendere i rischi derivanti da query SQL non sanificate.

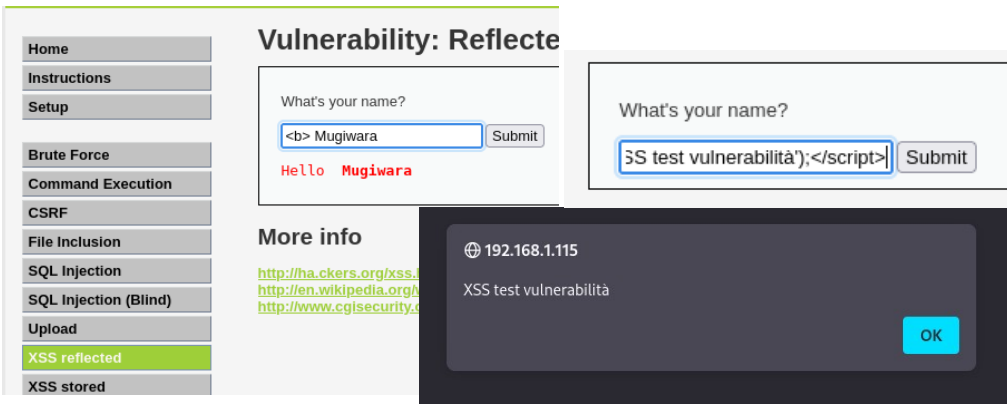
Utilizzo una macchina Kali Linux e una macchina DVWA.
Verifico la connessione tra Kali Linux e la macchina DVWA tramite il comando ping e imposto il livello di sicurezza su “LOW” per facilitare i test di vulnerabilità..



Nella sezione XSS reflected troviamo un campo di input nel quale possiamo fare i nostri test
Test con uno script Semplice:

Come primo test, ho utilizzato un XSS di base per generare un pop-up, verificando così la presenza della vulnerabilità:

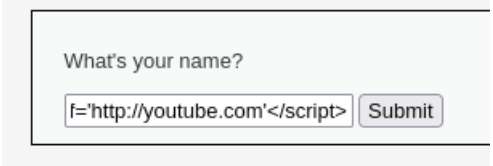
<script>alert('XSS');</script>



Il browser ha eseguito il codice JavaScript dell’attacante, evidenziando che l’input non è stato correttamente filtrato e che il campo è vulnerabile ad attacchi XSS.

Ho eseguito un ulteriore test più avanzato per verificare il livello di controllo ottenibile:
Verifico la possibilità di indirizzare la vittima verso un sito da me prestabilito, in questo caso ho scelto youtube, ma avrei potuto inserire un sito clonato o altro.

<script>window.location.href='http://youtube.com'</script>

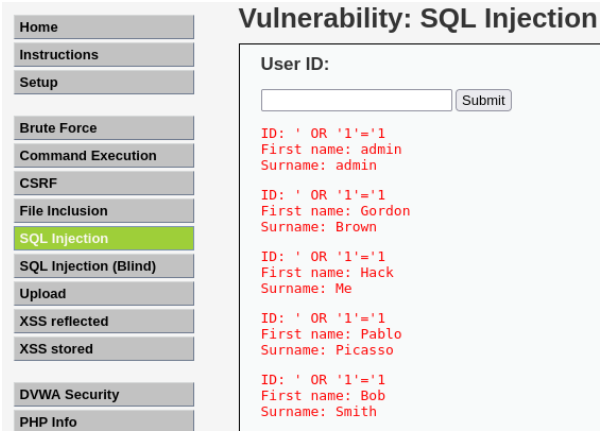


Il risultato è che l’indirizzamento avviene con successo

Nella sezione SQL Injection troviamo un campo di input nel quale possiamo fare i nostri test
Test con uno script SQL Injection:
È stato utilizzato uno script SQL Injection di base per bypassare i controlli e ottenere l’accesso ai dati:

' OR '1'='1
Inserendo questo script, il sistema ha accettato la query senza errori e ha restituito risultati che normalmente sarebbero stati nascosti, confermando la vulnerabilità.

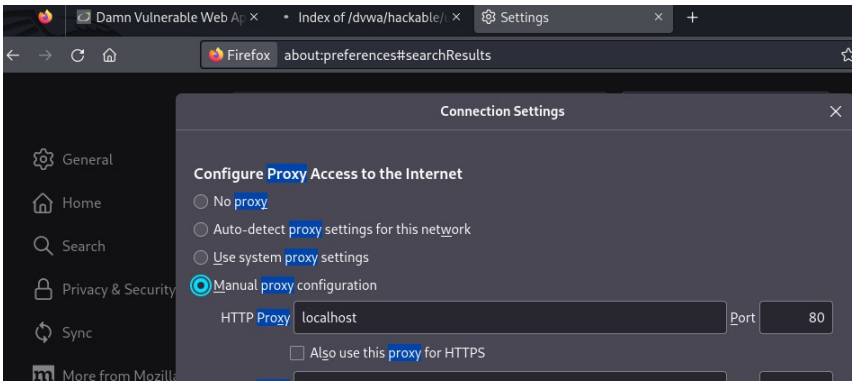
Lo script ha permesso di iniettare un comando SQL, questo tipo di vulnerabilità consente di accedere a informazioni riservate o persino di modificare i dati.



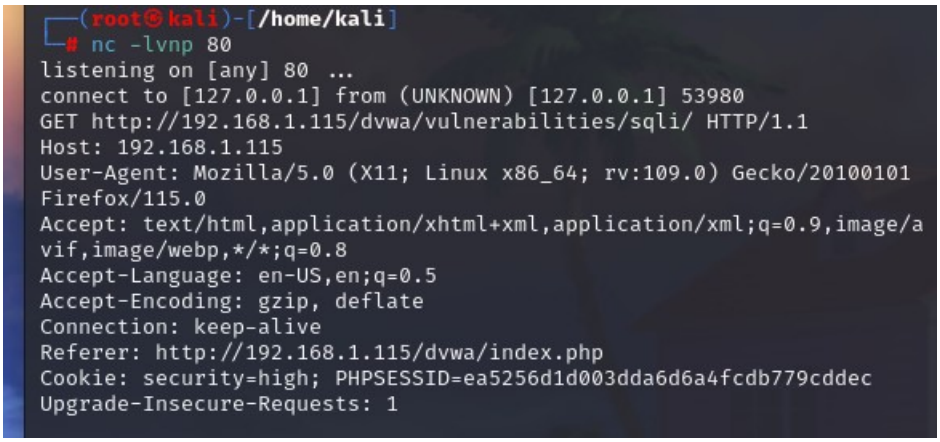
Come misura preventiva, è fondamentale filtrare o rimuovere caratteri speciali (<, >, &, ecc.) dall'input dell’utente.
CSP (Content Security Policy): Implementare politiche di sicurezza dei contenuti per limitare l'esecuzione di script non autorizzati.

Per monitorare il traffico delle informazioni utilizzo netcat per verificare i cookie di sessione e monitorare le richieste HTTP:

Ho dovuto impostare manualmente il proxy sul browser per permettere a netcat di intercettare la porta 80



Con il comando nc -lvnp 80 possiamo intercettare il traffico della vittima e monitorare le sue attività



Tramite netcat, è stato possibile visualizzare i dettagli delle richieste e intercettare i cookie di sessione inviati dalla DVWA, dimostrando come un attaccante potrebbe potenzialmente catturare i cookie di sessione di una vittima.