

## Aclaraciones Práctica 3: Mitigación SQLi

### Contenido

<b>Creación y Poblado de la base de datos .....</b>	<b>2</b>
<b>Programación de una pequeña aplicación de Login insegura en PHP con parámetros POST .....</b>	<b>4</b>
<b>Programación de una pequeña aplicación de búsqueda de usuarios insegura en PHP con parámetros GET .....</b>	<b>6</b>
<b>Uso de la herramienta automatizada SQLMAP.....</b>	<b>8</b>
<b>Ver permisos de Usuarios en la base de datos .....</b>	<b>9</b>
<b>Configurar contraseña al Usuario root de la Base de datos .....</b>	<b>10</b>
<b>Validar qué permisos de administrador tiene un usuario .....</b>	<b>11</b>

## Creación y Poblado de la base de datos

Llegados a este punto, nos conectaremos por ssh a la máquina donde esté alojada la BBDD, en este caso a nuestra máquina de bee-box:

```
ssh bee@192.168.1.13
```

Una vez conectados, mediante el comando mysql, nos conectaremos a la base de datos como root.

```
sudo mysql -u root -p  
bug
```

Creamos base de datos galileo:

```
CREATE DATABASE galileo;  
USE galileo;
```

Crear nuevo usuario *sqluser* que utilizaremos para conectarnos desde nuestra aplicación

```
CREATE USER sqluser@localhost IDENTIFIED BY 'password';
```

Darle permisos sobre la BBDD

```
GRANT ALL ON galileo.* TO sqluser@localhost;
```

Crear tabla usuarios:

```
CREATE TABLE usuarios (  
    idusuario int,  
    nombre varchar(255),  
    apellidos varchar(255),  
    password varchar(255)  
);
```

### Insertar Usuarios:

```
INSERT INTO usuarios(idusuario,nombre,apellidos,password) VALUES  
("1","javier","Perez Jambrina","javi77");  
INSERT INTO usuarios(idusuario,nombre,apellidos,password) VALUES  
("2","sofia","Alfonso Lucas","sflul2");  
INSERT INTO usuarios(idusuario,nombre,apellidos,password) VALUES  
("3","julio","Narvaez lopez","jul2lio");  
INSERT INTO usuarios(idusuario,nombre,apellidos,password) VALUES  
("4","eva","Santos Fernandez ","evaeva11");
```

### Confirmación de que los datos se han insertado correctamente:

```
SELECT * from usuarios;
```

## Programación de una pequeña aplicación de Login insegura en PHP con parámetros POST

Para la realización de la siguiente parte, es necesario que la versión de PHP del servidor sea inferior a PHP v7, puesto que algunas funciones se han deprecado a partir de esa versión. (Para ver la versión de nuestro servidor usaremos el comando `php -v`)

Creamos formulario de login en PHP y la consulta de esos datos a la base de datos. Se ha realizado en los ficheros `index.php` y `content.php` que se adjuntan en Google Classroom.

Para subir los ficheros a nuestro servidor PHP, a debemos crear un nuevo directorio dentro de la ruta de Apache `/var/www`, en nuestro caso lo hemos llamado `/practica3`. Podemos usar el comando `scp` para subir los ficheros.

```
scp index.php bee@192.168.1.11:/var/www/practica3
scp content.php bee@192.168.1.11:/var/www/practica3
```

Probamos a realizar la siguiente inyección sobre nuestro formulario de login (Nota: `--` es equivalente a `#`)

' OR 1=1 -- -
---------------

Debemos sanitizar todas las entradas con la función, por ejemplo con:

```
mysqli_real_escape_string($conexión,$variable);
```

&lt;?php

```
if(isset($_POST['username']) && isset($_POST['passwd']))
{
    $username=$_POST['username'];
    $passwd=$_POST['passwd'];

    $connection = mysqli_connect("localhost", "sqliuser", "password", "galileo");

    echo 'Connected successfully';
    echo nl2br("\n\n");

    $username = mysqli_real_escape_string($connection, $username);
    $passwd = mysqli_real_escape_string($connection, $passwd);

    $sql = "SELECT nombre,apellidos,password FROM usuarios WHERE nombre='".$username."' and
password='".$passwd."'";

    $result = mysqli_query($connection, $sql);

    $row = mysqli_fetch_array($result, MYSQLI_BOTH);

    echo $row["nombre"]." " . $row["apellidos"]." " . $row["password"];

}
else
    echo "Error en parámetros";

?>
```

Si nos arroja algún error o la pantalla queda en blanco, deberemos instalar:

```
sudo apt-get install php-mysqli
```

## Programación de una pequeña aplicación de búsqueda de usuarios insegura en PHP con parámetros GET

Creamos formulario de login en PHP de GET y la consulta de esos datos a la base de datos. Se ha realizado en los ficheros `index_get.php` y `content_get.php` que se adjuntan en Google Classroom. Debes renombrarlos a `index.php` y `content.php` respectivamente.

Si realizamos una petición, podremos ver la URL que se forma y que admite parámetros dentro de la propia URL (Por GET)

```
http://192.168.1.13/actividad3/content.php?id=1
```

Para realizar la inyección, bastaría con colocar el siguiente contenido en el formulario o directamente en la URL. (Nota, no es necesario la comilla simple puesto que el campo *id* es numérico y no de tipo texto). (Nota: -- - es equivalente a #)

<code>1 OR 1=1 -- -</code>
----------------------------

En este punto deberíamos sanitizar la entrada para proteger frente al ataque de inyección SQL, de la siguiente forma:

Debemos cambiar la sentencia usando `mysqli_prepare()`,  
`mysqli_stmt_bind_param()`, `mysqli_stmt_execute()`,  
`mysqli_stmt_get_result()`, `mysqli_fetch_array()`;

```
<?php

if(isset($_GET['id']))
{
    $id = $_GET['id'];

    $connection = mysqli_connect("localhost", "sqluser", "password", "galileo");

    $sql = "Select * from usuarios where idusuario=$id";

    if($stmt = mysqli_prepare($connection,$sql))
    {

        mysqli_stmt_bind_param($stmt, 'i', $id);
        mysqli_stmt_execute($stmt);
        $result= mysqli_stmt_get_result($stmt);
        $row= mysqli_fetch_array($result,MYSQLI_BOTH);

        if($row)
        {
            echo $row["nombre"]." ".$row["apellidos"]." ".$row["password"];
        }

    }
    else
        echo "Error en parámetros";

?>
```

## Uso de la herramienta automatizada SQLMAP

Sacar nombre de Base de datos

```
sqlmap -u http://192.168.1.13/actividad3/content.php?id=1 -dbs
```

Ver las tablas de la base de datos

```
sqlmap -u http://192.168.1.13/actividad3/content.php?id=1 -random-agent -  
level 5 -D galileo --tables;
```

Ver los campos de una tabla

```
sqlmap -u http://192.168.1.13/actividad3/content.php?id=1 -random-agent -  
level 5 -D -D galileo -D galileo -T usuarios -columns
```

Ver el valor de una columna

```
sqlmap -u http://192.168.1.13/actividad3/content.php?id=1 -D galileo -T  
usuarios -C nombre,password -dump
```

Ver el valor del usuario de la Base de Datos y su contraseña (SQLmap realiza un ataque de fuerza bruta utilizando el diccionario *rockyou.txt*)

```
sqlmap -u http://192.168.1.13/actividad3/content.php?id=1 --users --passwords
```



## Ver permisos de Usuarios en la base de datos

Para saber que usuario está utilizando la aplicación web, es recomendable buscar el fichero *db\_credentials.php*.

En nuestro caso, el usuario se encuentra dentro del propio fichero *content.php*

Nos conectamos a la base de datos con root

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

Listamos los usuarios existentes

```
select user,host from mysql.user;
```

Mostramos permisos para un usuario concreto:

```
show grants for sqliuser@localhost;
```

Revocamos los permisos al usuario

```
revoke ALL ON *.* from sqliuser@localhost;
```

Le damos permiso sólo sobre las tablas de la base de datos que necesita:

```
grant all privileges on galileo.* to junior@localhost WITH grant option;
```

## Configurar contraseña al Usuario root de la Base de datos

Validamos si el usuario root tiene contraseña seteada:

```
select authentication_string from mysql.user where user="root";
```

Le configuramos una contraseña 'no recomendable':

```
alter user 'root'@'localhost' identified by "1234";
```

Validamos de nuevo si el usuario root tiene contraseña seteada:

```
select authentication_string from mysql.user where user="root";
```

Reiniciamos el Servicio Mysql:

```
service mysql restart;
```

Nos conectamos a la base de datos con el usuario root y su nueva contraseña:

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

## Validar qué permisos de administrador tiene un usuario

Para ver los permisos

```
sudo -l
```

A continuación, podemos consultar el fichero para ver los permisos de ese usuario:

```
cat /etc/sudoers
```

Dependiendo de los permisos que tengamos con nuestro usuario, podremos realizar una escalada de privilegios utilizando como referencia para buscar el comando para realizarlo la siguiente web:

<https://gtfobins.github.io/#na>