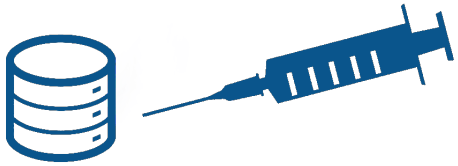
ATAQUE
SQL INJECTION



Un ataque SQL Injection es un **método de infiltración de código, ante la falta de validación de campos, en operaciones sobre una base de datos.**

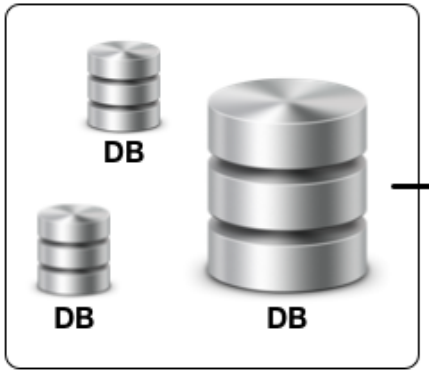
1.

BASES DE DATOS

BASES DE DATOS

- Serie de datos organizados y relacionados entre sí
- Objetivo: explotarlos por los usuarios o la empresa / organización
- Componentes: Hardware, Software y **Datos**.

BASE DE DATOS SQL



SQL DATABASE

Table Table Table

First Name	Last Name	Address	City	Age
Mickey	Mouse	123 Fantasy Way	Anaheim	73
Bat	Man	321 Cavern Ave	Gotham	54
Wonder	Woman	987 Truth Way	Paradise	39
Donald	Duck	555 Quack Street	Mallard	65
Bugs	Bunny	567 Carrot Street	Rascal	58
Wiley	Coyote	999 Acme Way	Canyon	61
Cat	Woman	234 Purrfect Street	Hairball	32
Tweety	Bird	543	Itotltaw	28

Rows

Columns

CONSULTAS SQL

- Crear tabla: *CREATE TABLE 'opweb' (…)*
- Insertar información: *insert into opweb (field1,field2) values (data1,data2)*
- Consultas: *select * from opweb group_by field1*
- Especificando campos: *select field1,field2 from opweb*
- Filtrando valores: *select * from opweb where field1 > value*
- Combinar resultados: *select … UNION select …*

2.

INTRODUCCIÓN A SQL INJECTION

INTRODUCCIÓN

- Técnica consistente en inyectar código malicioso en aplicaciones web
- Una persona no autorizada busca tener acceso a la información
- Posibles objetivos: usuarios y contraseñas
  - Acceso al servidor
  - Acceso al panel de administración
  - Ingeniería social con los usuarios
  - Robo de datos de visitantes

PROBLEMAS QUE PROVOCA

- *Confidencialidad*: acceder a información sensible supone una pérdida de confidencialidad
- *Autenticación*: si el acceso a una zona restringida de una web es débil, con este ataque se puede acceder sin conocer usuario ni contraseña
- *Integridad*: de la misma forma que este ataque permite leer información, también es posible realizar cambios o incluso borrar toda información.

FUNCIONAMIENTO

- Posible vector de ataque: *formularios de login*
- Envío de esta información a una sentencia SQL

*sql = SELECT * FROM usuarios WHERE usuario = '$usuario' and password = '$pass';*

FUNCIONAMIENTO

Si no se toman las medidas necesarias, podría aceptar lógica en sus campos:

*sql = SELECT * FROM usuarios WHERE usuario = 'opweb' and password = '1234' OR '1'='1';*

La contraseña del usuario *opweb* no será *1234*

Pero *1 si es igual a 1*, por lo que la sentencia es correcta. Especial atención a las comillas.

3.

BBDD & FUNCIONES DE INFORMACIÓN

FUNCIONES DE INFORMACIÓN

Devuelven información de la base de datos

- Devuelve una cadena con la **versión** del servidor MySQL: *VERSION()*
- Devuelve el nombre de la **base de datos**: *DATABASE()*
- Nombre de **usuario** y *host*: *USER()*

INFORMATION_SCHEMA

Base de datos que almacena información acerca del resto de bases de datos que mantiene el servidor MySQL.

Formada por tablas de información como:

- *TABLES:* Información de las tablas de la bbdd
- *COLUMNS:* Inf. de las columnas en tablas
- *STATISTICS:* Inf. de los índices de las tablas
- *USER_PRIVILEGES:* permisos globales

4.

PREVENCIÓN

PREVENCIÓN

Uso de la función:

*string **mysql_real_escape_string** (string cadena […])*

Caracteres a escapar: ***\x00, \n, \r, \, ’, ”***

```
$consulta = "SELECT * FROM usuarios WHERE
usuario='{$_POST['usuario']}' AND password='{$_POST['password']}'";
mysql_query($consulta);
```

```
$_POST['usuario'] = mysql_real_escape_string($_POST['usuario']);
$_POST['password'] =
mysql_real_escape_string($_POST['password']);
```