VÍCTOR ÁLVAREZ FERNÁNDEZ

Unidad:

Unidad 1 - Introducción a los SGBD

Práctica:

P11-SQL-Injection

Fecha:

3 de Enero de 2026

Índice

Introducción

¿Qué es una SQL Injection?

Es una vulnerabilidad de seguridad, que permite al atacante interferir en las consultas que una aplicación realiza a una Base de Datos.

Objetivos Principales SQL Injection

I. **Suplantación de Identidad**

II. **Acceso a Datos Sensibles**

III. **Alteración de Datos**

IV. **Escalada en Privilegios dentro la Aplicación Web**

Ataque en la autenticación de usuarios

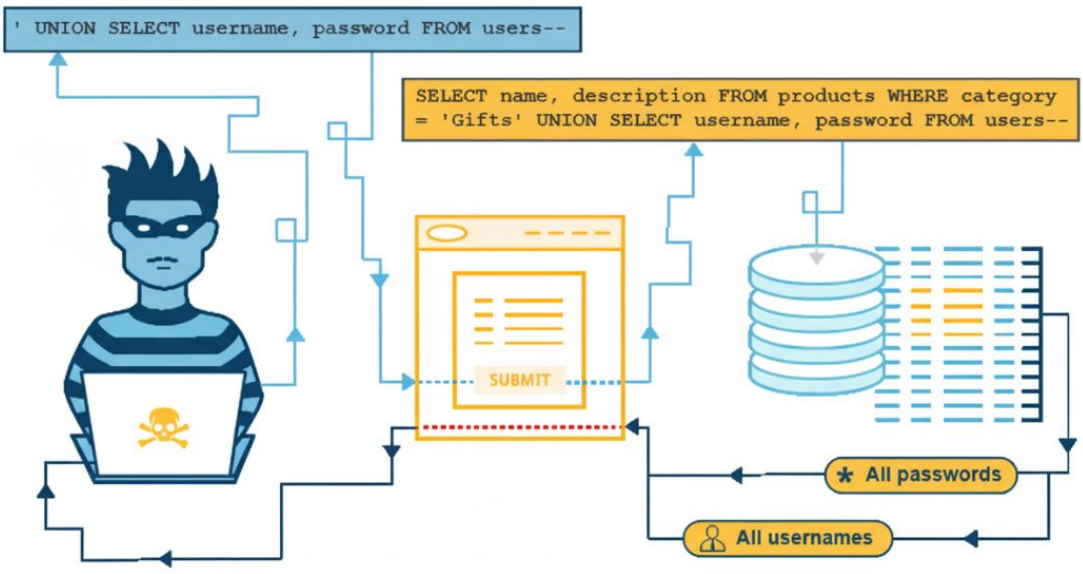
Este tipo de ataque se produce cuando en un servicio o aplicación vía web existe una validación de acceso para los usuarios mediante 'username/e-mail' y 'password'.

Si la configuración para este tipo de consultas de acceso no se ha realizado de manera correcta, un potencial 'usuario malicioso' podría acceder a la Base de Datos para utilizar, alterar o eliminar su contenido.

Ejemplo de mala configuración de una consulta de este tipo:

SELECT * FROM usuarios WHERE nombre = '" + usuario + "' AND clave = '" + password + "';.

El *uso de comillas y concatenaciones* abre la posibilidad a un programador de introducir código, que le dé acceso a una Base de Datos para consultarla y/o a realizar cualquier tipo de manipulación en ella.



Arquitectura tres niveles: Ideal para Aplicaciones Web

Los ataques SQL Injection tienen una mayor probabilidad de producirse en *arquitecturas SGBD de tres niveles*. Estas arquitecturas utilizan un nivel intermedio entre el Servidor que acoge la Base de Datos y los equipos Cliente que realizan las consultas.

La arquitectura de tres niveles es la más utilizada para la implementación de aplicaciones web y requiere de un Servidor Intermedio (Servidor Web), que acoge la instalación de este tipo de aplicaciones.

Precisamente, el que esté pensada para albergar aplicaciones web, y por consiguiente conectada a Internet, aumenta la probabilidad de aparición de usuarios malintencionados.

Otras arquitecturas

Por supuesto, un ataque SQL Injection se puede dar en *arquitecturas de dos niveles*, donde es bien sabido que no hay un Servidor Intermedio. Esta arquitectura está pensada para 'circuitos de usuarios cerrados' e incluso no expuestos a Internet; aunque esto no implica que puedan intentar acceder usuarios con fines maliciosos a través de esta vulnerabilidad, si es que existiera en el SGBD instalado y configurado en el Servidor.

De igual modo, se puede dar en una *arquitectura de un sólo nivel*, aunque la probabilidad de un ataque de este tipo es aún más reducida por muchas razones entre las que podemos esgrimir la dificultad de acceso al equipo que tiene instalado el SGBD; o incluso el interés que puede tener el acceso o manipulación a este tipo de datos.

Recordamos que la arquitectura de un solo nivel es utilizada por pequeñas empresas o usuarios domésticos, que aprovechan el mismo equipo para la instalación, configuración, y administración del SGBD, desde donde también acceden los usuarios finales.

TalkTalk Telecom Group

TalkTalk Telecom Group era una de las mayores empresas de telecomunicaciones en el Reino Unido. En octubre de 2015, la compañía sufrió una brecha de seguridad masiva que expuso los datos personales de más de 150.000 clientes.

El ataque no fue para nada sofisticado. *Los atacantes utilizaron una técnica de Inyección SQL en las páginas de acceso de los clientes y en Bases de Datos heredadas de una adquisición previa (Tiscali)*.

Cronología de los hechos

I. ***El fallo***: **El personal encargado de la programación de los formularios web no tuvo la brillante idea de incluir campos de consulta más profesionales para la validación de usuarios. Esto permitió que un atacante insertara comandos SQL maliciosos en los campos de texto.**

II. ***La ejecución***: **El servidor, al no distinguir entre un dato (como un nombre de usuario) y una instrucción de código, ejecutó los comandos directamente en la base de datos.**

III. ***Herramientas***: **Se cree que los atacantes utilizaron herramientas automatizadas como sqlmap, que escanean y explotan estas vulnerabilidades de forma sistemática.**

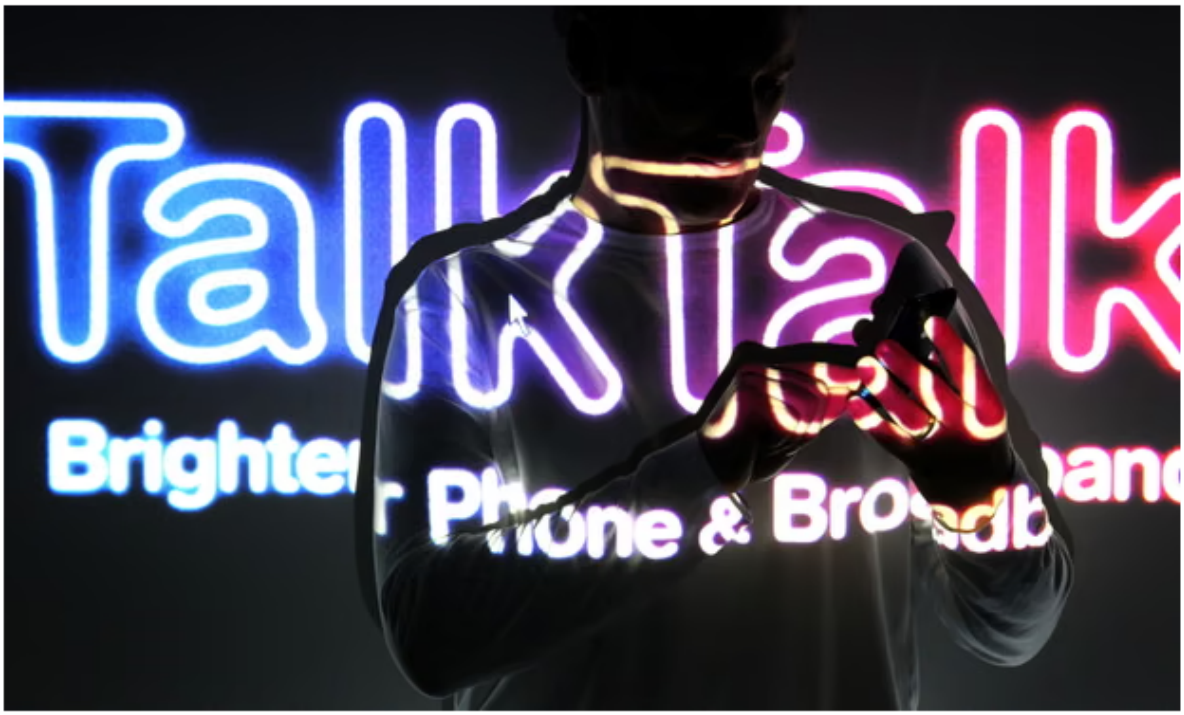
La brecha de seguridad provocó que los atacantes lograran extraer datos personales de miles de clientes, incluyendo nombres, domicilios y fechas de nacimiento; sin embargo, el daño más crítico fue la obtención de detalles bancarios y números de cuenta de más de 15.000 usuarios. Este incidente subrayó el peligro de no cifrar campos sensibles dentro de las tablas del SGBD, permitiendo que la información fuera legible de forma inmediata tras el éxito de la inyección SQL.

La respuesta institucional fue contundente, con una multa de 400.000 libras, debido a que la vulnerabilidad se consideró evitable con prácticas básicas de seguridad. Más allá de la sanción, TalkTalk enfrentó una crisis financiera con costes operativos y de reparación que ascendieron a los 77 millones de libras, sumado a una pérdida de reputación que provocó la fuga de más de 100.000 clientes.

This article is more than **9 years old**

TalkTalk hit with record £400k fine over cyber-attack

Internet service provider handed fine by Information Commissioner's Office after security failings allowed customer data to be accessed 'with ease'



More than 150,000 TalkTalk customers had their personal details hacked in the attack in October 2015. Photograph: Andrew Milligan/PA

TalkTalk has been hit with a record £400,000 fine for the security failings that led to the company being hacked in October 2015.

The Information Commissioner's Office levied the fine saying that the attack "could have been prevented if TalkTalk had taken basic steps to protect customers' information".

The hack resulted in the attacker accessing the personal information of more than 150,000 customers of the internet service provider, including sensitive financial data for more than 15,000 people.

Texto plano en la validación de usuarios

El configurar el SGBD para que sólo acepte texto plano en la validación de usuarios, elimina la posibilidad de incursión mediante SQL Injection. Por este motivo, el uso de concatenaciones y comillas (simples o dobles) es una mala práctica y debe evitarse.

En su lugar se recomienda utilizar el símbolo *?*, que indica al SGBD que todo lo introducido por los campos de texto implicados se trate como texto plano.

Ejemplo de buena práctica de configuración de una consulta de este tipo:

SELECT * FROM usuarios WHERE nombre = ? AND password = ?;.

Además, el símbolo *?* es un comodín de seguridad que habilita al usuario a introducir tantos caracteres, y el tipo de los mismos, como desee.

Importancia de la Seguridad Informática

Es importante establecer otras medidas y prácticas de seguridad que dificulten el acceso malintencionado a través de métodos como el *registro de pulsaciones (keylogger)*, *ataques de fuerza bruta* y/o *diccionario*.

Dentro de las buenas prácticas a la hora de realizar los registros, *sería ideal configurar criterios para que los nuevos usuarios establezcan contraseñas más seguras que utilicen un número mínimo de caracteres; el uso de mayúsculas, minúsculas y caracteres especiales*; e incluso *fijar la caducidad de la contraseña para que esta se tenga que renovar cada cierto periodo de tiempo*.

Almacenar las *claves mediante algoritmos* sería también algo muy interesante, sobre todo si la Base de Datos maneja información que se pueda considerar sensible.