TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

seguridad-virtualización

INVESTIGACIÓN

GRUPO: 7US

PRESENTA:

RUFINO MENDOZA VAZQUEZ - 21620198

ANA MICHEL LEÓN LEÓN - 21620112

ROSA SALAZAR DOROTEO - 18620216

FERNANDA RUIZ HERAS - 21520151

DOCENTE

ING. EDWARD OSORIO SALINA

Tlaxiaco, Oax.,16 de septiembre del 2024.

"Educación, ciencia y tecnología, progreso día con día"®

1. ¿Qué es la inyección de SQL y cómo funciona?

SQL Injection (SQLi) es una técnica de ataque cibernético en la que un atacante inserta o “inyecta” código SQL malicioso en una consulta SQL a través de un formulario web o cualquier otro punto de entrada de datos en una aplicación. Esto puede permitir al atacante manipular la consulta SQL y acceder, modificar o eliminar datos en la base de datos.

Conceptos Clave de SQL Injection:

Inyección de Código: La inyección ocurre cuando el atacante incluye comandos SQL en los campos de entrada de datos que la aplicación no valida ni filtra adecuadamente. Por ejemplo, si una aplicación web permite a los usuarios buscar datos mediante un formulario, un atacante podría introducir una entrada como ' OR '1'='1 para manipular la consulta SQL y obtener acceso no autorizado a información.

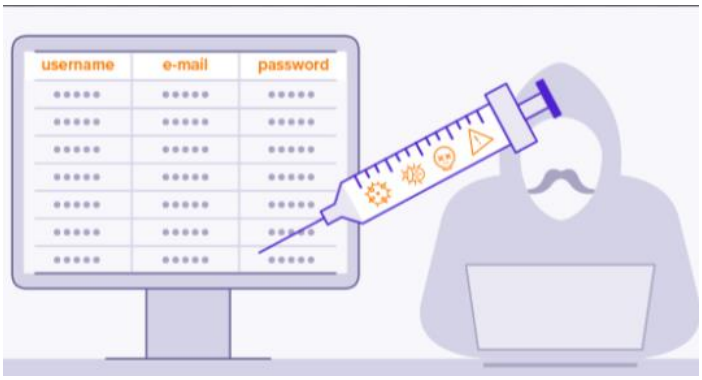
Tipos de SQL Injection:

Inyección Basada en Error: Utiliza errores de la base de datos para obtener información sobre la estructura de la base de datos.

Inyección Ciega: No devuelve errores de la base de datos, pero el atacante puede inferir información basándose en los comportamientos observados.

Inyección de Tiempo: Basada en los tiempos de respuesta del servidor para deducir información sobre la base de datos.

Inyección Basada en Unión: Utiliza la cláusula UNION para combinar resultados de múltiples consultas y extraer datos.



Consecuencias:

Acceso No Autorizado: Los atacantes pueden obtener acceso a datos sensibles como contraseñas, datos financieros, etc.

Modificación de Datos: Cambiar o eliminar datos importantes.

Divulgación de Datos: Robar información confidencial.

Control Total del Sistema: En algunos casos, obtener control completo del servidor o la base de datos.

Prevención de SQL Injection:

Uso de Consultas Preparadas y Parametrizadas: Las consultas preparadas y parametrizadas (también conocidas como consultas con parámetros) permiten separar el código SQL de los datos de entrada. En lugar de incluir directamente los datos del usuario en la consulta SQL, se utilizan parámetros que son procesados de manera segura por el motor de base de datos.

2. **Conceptos de bases de datos seguras y cómo se pueden implementar**

Las bases de datos seguras son fundamentales para proteger la información sensible y garantizar la confidencialidad, integridad y disponibilidad de los datos. A continuación, se presentan algunos de los conceptos claves:

Seguridad de las redes

- Los **Firewalls** son la primera línea de defensa en la seguridad de bases de datos DiD. Lógicamente, un firewall es un separador o limitador del

tráfico de red, que puede configurarse para aplicar la directiva de seguridad de datos de su organización. Si utiliza un firewall, aumentará la seguridad del sistema operativo, ya que proporciona un cuello de botella en el que pueden concentrarse las medidas de seguridad.

Administración de acceso

- **Autenticación** es el proceso de demostrar que el usuario es quien dice ser introduciendo el ID de usuario y la contraseña correctos. Algunas soluciones de seguridad permiten a los administradores administrar de forma centralizada las identidades y permisos de los usuarios de la base de datos. Esto incluye la minimización de almacenamiento de contraseñas y permite directivas centralizadas de rotación de contraseñas.
- **Autorización** permite a cada usuario acceder a determinados objetos de datos y realizar ciertas operaciones en la base de datos como leer, pero no modificar datos, modificar, pero no borrar datos, o borrar datos.
- **Control de acceso** es realizado por el administrador del sistema, que asigna permisos a un usuario dentro de una base de datos. Los permisos se administran idealmente agregando cuentas de usuario a roles de base de datos y asignando permisos a nivel de base de datos a dichos roles. Por ejemplo, de seguridad de nivel de fila (RLS) permite a los administradores de bases de datos restringir el acceso de lectura y escritura a filas de datos en función de la identidad, la pertenencia a roles o el contexto de ejecución de consultas de un usuario. RLS centraliza la lógica de acceso en la propia base de datos, lo que simplifica el código de la aplicación y reduce el riesgo de revelación accidental de datos.

Protección contra amenazas

- **Auditoría** realiza un seguimiento de las actividades de la base de datos y ayuda a mantener el cumplimiento de las normas de seguridad registrando los eventos de la base de datos en un registro de auditoría. Esto le permite supervisar las actividades en curso de la base de datos, así como analizar e investigar la actividad histórica para identificar posibles amenazas o sospechas de abuso y violaciones de la seguridad.

- **Detección de amenazas** descubre las actividades anómalas de la base de datos que indican una posible amenaza para la seguridad de la base de datos y puede mostrar información sobre eventos sospechosos directamente al administrador.

Protección de la información

- **Cifrado de datos** protege los datos confidenciales convirtiéndolos a un formato alternativo, de modo que solo las partes interesadas puedan descifrarlos y devolverlos a su forma original y acceder a ellos. Aunque el cifrado no resuelve los problemas de control de acceso, mejora la seguridad al limitar la pérdida de datos cuando se eluden los controles de acceso. Por ejemplo, si el equipo anfitrión de la base de datos está mal configurado y un usuario malintencionado obtiene datos sensibles, como números de tarjetas de crédito, esa información robada podría ser inútil si está encriptada.
- **Copia de seguridad y recuperación de la base de datos** es fundamental para proteger la información. Este proceso implica hacer copias de seguridad de la base de datos y de los archivos de registro de forma periódica y almacenar las copias en un lugar seguro. La copia de seguridad y el archivo están disponibles para restaurar la base de datos en caso de fallo o infracción de la seguridad.
- **Seguridad física** limita estrictamente el acceso al servidor físico y a los componentes de hardware. Muchas organizaciones con bases de datos locales usan salas cerradas con acceso restringido para el hardware del servidor de base de datos y los dispositivos de red. También es importante limitar el acceso a los soportes de copia de seguridad almacenándolos en un lugar seguro fuera de las instalaciones.

IMPLEMENTACIÓN

Para implementar estos conceptos, se recomienda:

- **Evaluar Riesgos:** Realizar un análisis de riesgos para identificar vulnerabilidades específicas de la base de datos.

- **Definir Políticas:** Establecer políticas claras de seguridad y procedimientos de respuesta a incidentes.
- **Capacitar al Personal:** Proporcionar capacitación regular sobre buenas prácticas de seguridad a todos los empleados.
- **Revisar y Actualizar:** Realizar revisiones periódicas de las medidas de seguridad implementadas y actualizarlas según sea necesario.

BIBLIOGRAFIA:

Avast. (s.f.). Obtenido de https://www.avast.com/es-es/c-sql-injection#:~:text=La%20inyecci%C3%B3n%20de%20SQL%20es,la%20informaci%C3%B3n%20de%20los%20usuarios

Microsoft. (s.f.). Obtenido de https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-database-security