Reporte de Incidente
Vulnerabilidad SQL Injection en DVWA
ISO 27001

Introducción:

El presente reporte detalla la detección y el impacto de una vulnerabilidad de Inyección SQL en el sistema, durante una prueba de seguridad realizada en la aplicación Damn Vulnerable Web Aplicación (DVWA), este tipo de vulnerabilidad permite a un atacante ejecutar comandos SQL maliciosos, compromete la integridad, la confidencialidad de la base de datos y extraer múltiples registros con solo manipular el parámetro User ID; además recomienda acciones correctivas para mitigar el riesgo.

Descripción del Incidente:

Se ha detectado una vulnerabilidad de Inyección SQL que permite el acceso no autorizado a información sensible de la base de datos. Al introducir la cadena '1' OR '1'='1' en el campo de User ID, el sistema devuelve múltiples registros de usuarios, lo que indica que la aplicación no está validando o depurando correctamente las entradas del usuario antes de incorporarlas a las consultas SQL.

Proceso de Reproducción

1. Acceso a la Interfaz: Se accedió a la interfaz de usuario donde se solicita el User ID.
2. Inyección de Payload: En el campo User ID, se introdujo el payload de Inyección SQL: 1' OR '1'='1.
3. Observación de Resultados: Tras la ejecución de la consulta, el sistema devolvió una lista de usuarios, incluyendo:
    a. ID: 1' OR '1'='1, Nombre: admin, Apellido: admin
    b. ID: 1' OR '1'='1, Nombre: Gordon, Apellido: Brown
    c. ID: 1' OR '1'='1, Nombre: Hack, Apellido: Me
    d. ID: 1' OR '1'='1, Nombre: Pablo, Apellido: Picasso
    e. ID: 1' OR '1'='1, Nombre: Bob, Apellido: Smith

Estos resultados confirman que la inyección fue exitosa y que la consulta SQL original fue alterada para devolver todos los registros en la tabla de usuarios

Impacto del Incidente:

La vulnerabilidad de Inyección SQL representa un riesgo crítico para la seguridad del sistema. Sus posibles impactos incluyen:

Acceso no autorizado a datos: Un atacante puede leer, modificar o eliminar cualquier dato en la base de datos.

Escalada de privilegios: Si la base de datos tiene permisos elevados, un atacante podría obtener control sobre el servidor subyacente.

Denegación de servicio (DoS): Un atacante podría dañar o eliminar datos críticos, llevando a una interrupción del servicio.

Divulgación de información sensible: Credenciales de usuario, información personal identificable (PII) y otros datos confidenciales pueden ser expuestos.

Compromiso total del sistema: En escenarios avanzados, un atacante podría ejecutar comandos del sistema operativo a través de la base de datos.

Pérdida de confianza: Si esta vulnerabilidad es explotada en un entorno real, puede dañar la reputación de la organización.

Recomendación:

Para mitigar esta vulnerabilidad y prevenir futuros ataques de Inyección SQL, se recomienda implementar las siguientes medidas:

Consultas Parametrizadas/Sentencias Preparadas: Utilizar consultas parametrizadas o sentencias preparadas en todos los casos donde se interactúa con la base de datos. Esto asegura que la entrada del usuario sea tratada como un valor literal y no como parte del comando SQL.

Validación de Entrada: Implementar una validación estricta de la entrada del usuario en el lado del servidor, asegurándose de que los datos cumplan con el formato y tipo esperados (por ejemplo, validar que un User ID sea numérico si así se requiere).

Escape de Caracteres Especiales: Escapar adecuadamente los caracteres especiales en todas las entradas antes de construir consultas SQL dinámicas (aunque las consultas parametrizadas son el método preferido).

Principio de Mínimo Privilegio: Asegurar que las cuentas de usuario de la base de datos utilizadas por la aplicación tengan solo los permisos mínimos necesarios para realizar sus funciones.

Web Application Firewall (WAF): Considerar la implementación de un WAF para detectar y bloquear ataques conocidos de Inyección SQL y otros tipos de ataques web.

Auditorías de Código: Realizar auditorías de seguridad periódicas del código para identificar y corregir posibles vulnerabilidades.

Mensajes de Error Genéricos: Configurar la aplicación para que muestre mensajes de error genéricos al usuario final, evitando revelar detalles internos del sistema o de la base de datos.

Conclusión:

La vulnerabilidad de Inyección SQL identificada es una brecha de seguridad grave que requiere atención inmediata. Al implementar las recomendaciones mencionadas, la organización puede fortalecer significativamente su postura de seguridad, proteger los datos críticos y asegurar la continuidad de sus servicios. Es fundamental priorizar la remediación de esta vulnerabilidad para evitar un posible impacto devastador en la confidencialidad, integridad y disponibilidad del sistema