Materia:

Desarrollo movil Integral

Grupo:

10-B

Nombre del Alumno:

Rendon Gurrola Alan Misael

Docente:

Ray Brunette Parra Galaviz

Fecha de Elaboración:

15/01/2024

Secure Coding Principles Specification

Especificación de principios de codificación segura

Codificación segura, el principio de diseñar código que se adhiera a las mejores prácticas de seguridad del código, salvaguarde y proteja el código publicado de vulnerabilidades conocidas, desconocidas e inesperadas, como vulnerabilidades de seguridad, la pérdida de secretos de la nube, credenciales integradas, claves compartidas, datos comerciales confidenciales y personales. información identificable (PII).

Refleja una comprensión más amplia entre los desarrolladores, los equipos de seguridad y DevOps de que la seguridad del código debe aplicarse como parte integral de CI/CD, supporting continuous changes both in code and in infrastructure, providing visibility into all seen and hidden components of a given environment.

Los incidentes de seguridad pueden tener graves consecuencias para negocios y personas. Un código creado sin seguir buenas prácticas de seguridad puede provocar daños económicos y robos de identidad, entre otros perjuicios.

En sectores específicos, como finanzas, salud, energía y transporte, los efectos negativos pueden ser mayores.

Las compañías que se relacionan con sus clientes a través de webs y aplicaciones móviles pueden perder la confianza de estos por problemas de seguridad. Una de las peores cosas que pueden pasar es poner en peligro a los clientes. Por este motivo, consolidar buenas prácticas de desarrollo es una prioridad.

Desde un punto de vista tecnológico, el daño que se causa a una web o app puede afectar a otros sistemas, como por ejemplo la base de datos del backend o el sistema del usuario.

Cuando se trata de hacer desarrollos, no se puede reinventar la rueda. Así, los expertos han alcanzado consensos sobre qué prácticas garantizan un desarrollo seguro.

Una de las guías más seguidas es OWASP Secure Coding Practices. Se trata de un documento muy práctico, con buenas prácticas descritas a modo de checklist en un total de 14 áreas con impacto en el ciclo de desarrollo.

Los 14 ámbitos de seguridad analizados por OWASP son:

- Validación de entradas
- Codificación de salidas
- Gestión de autenticación y contraseñas
- Administración de sesiones
- Control de acceso
- Prácticas criptográficas
- Gestión de errores y logs
- Protección de datos
- Seguridad en las comunicaciones
- Configuración del sistema
- Seguridad de base de datos
- Gestión de archivos
- Gestión de memoria
- Prácticas generales para el desarrollo de código
- Los desarrolladores siguen las prácticas de codificación segura de OWASP
- Validación

Validación de entradas y codificación de salidas

Una buena práctica destacada en este ámbito es la necesidad de identificar todas las fuentes de datos, y de clasificarlas según si son de confianza o no son de confianza. Todos los datos que vengan de fuentes que no son de confianza (bases de datos, secuencias de archivo, etc), habría que validarlos. Cualquier error en la validación debe llevar al rechazo del dato de entrada.

Gestión de autenticación y contraseñas

Las contraseñas son un punto débil en muchos sistemas, razón que explica la generalización de la autenticación multifactor. Ahora bien, las contraseñas son necesarias. De hecho, hoy son la acreditación más común para garantizar la seguridad.

Como regla general, hay que pedir autenticación para todos los recursos y páginas, excepto para aquellas que se han clasificado como públicas.

Administración de sesiones

La duración de las sesiones debe ser lo más breve posible, buscando un equilibro entre las necesidades de seguridad del sistema y las necesidades del usuario/cliente.

Habría que utilizar tokens para reforzar la gestión de sesiones para operaciones delicadas del lado del servidor, como por ejemplo la gestión de cuentas de usuario. Los tokens previenen los ataques de Cross Site Request Forgery.

Control de acceso

Las decisiones de acceso deben basarse en permisos, y no en exclusiones. En otras palabras, la idea es que el acceso se deniegue por defecto, y que el sistema identifique las condiciones bajo las cuales se permite un acceso. Por tanto, a los usuarios que no pueden demostrar autorización, se les debe denegar el acceso.

Prácticas criptográficas

En caso de filtración de datos, es fundamental la encriptación de datos con algoritmos criptográficos y seguir buenas prácticas de gestión segura de claves.

Hay muchas librerías disponibles para ayudar a implementar la encriptación. Ahora bien, es importante usar solo algoritmos y librerías estándar.

Gestión de errores y logs

Los errores de código pueden apuntar muchas veces a vulnerabilidades, así que la gestión y el registro de errores (logs) son dos técnicas muy útiles.

Protección de datos

Muchos ciberataques buscan acceder a datos sensibles. De ahí que sean cruciales las buenas prácticas de codificación en protección de datos.

Seguridad en las comunicaciones

Una de las buenas prácticas destacadas es implementar la encriptación para todas las transmisiones de información sensible. Esto debería incluir TLS para proteger la conexión. Además, se puede recurrir a una encriptación discreta de archivos sensibles o conexiones no basadas en HTTP.

Configuración del sistema

La guía de OWASP para un desarrollo seguro también describe buenas prácticas para la seguridad de los sistemas con los que se escribe el código. La idea principal es tenerlo todo actualizado y trabajar solo con las funcionalidades y archivos estrictamente necesarios. Otro consejo obvio, pero que no está de más recordar, es llevar un sistema de control de cambios y versiones.

Seguridad de base de datos

Una recomendación de seguridad destacada es el empleo de consultas parametrizadas, para evitar inyecciones SQL. Los datos de entrada en el sistema son interpretados como parámetros y no contienen código ejecutable.

Gestión de archivos y de memoria

Una gestión de archivos segura exige autenticación antes de transferir los archivos al servidor. También hay que hacer una validación de los tipos de archivo, a partir de los file headers (datos contenidos en el archivo con información que lo identifica), no a partir de las extensiones.

BIBLIOGRAFIA

Torrejón, M. (2022, noviembre 30). Buenas prácticas para un desarrollo seguro. El blog de Omatech; Omatech. https://www.omatech.com/blog/2022/11/30/buenas-practicas-para-un-desarrollo-seguro/