Recomendaciones de Seguridad en el Desarrollo de Software

Al Iniciar un Proyecto

Incorporar un enfoque de seguridad desde el inicio del proyecto reduce los esfuerzos de desarrollo y la cantidad de vulnerabilidades que heredan los sistemas productivos. También puede ayudar a acortar los plazos para la puesta en producción y aumenta los niveles de seguridad de la aplicación.
Principales aspectos a considerar en este enfoque,

Van a Atacar la aplicación

Se debe operar bajo la premisa de que la aplicación va a recibir ataques variados periódicamente. Dependiendo de la criticidad de la aplicación, puede aumentar la intensidad y sofisticación de los ataques.

Algún ataque va a funcionar

Se debe suponer que algún ataque va a funcionar. Ya sea por errores de diseño o *bugs* (defectos) en la implementación o por vulnerabilidades en la infraestructura de la que depende la aplicación.

Considerar la posibilidad real de que un atacante evadirá los controles de seguridad establecidos...

y cuando un atacante obtenga acceso al sistema debemos considerar el impacto que esto tendrá en el “negocio” o en los servicios que brinda el organismo.

Privacidad de Información

La información personal de sus usuarios, una vez “filtrada” no volverá a ser privada. Una filtración de datos puede afectar los procesos y la reputación de la organización. Una vez tomada la decisión de almacenar datos privados, debe asumirse la responsabilidad de proteger su confidencialidad mediante controles de seguridad y técnicas de desarrollo seguro.

Considerar la Seguridad en Cada Decisión

Cada cambio sobre la aplicación implica nuevos riesgos. Al tomar decisiones sobre la arquitectura del sistema, debe hacerse un análisis de los riesgos implicados. Los riesgos generados con los cambios pueden ser difíciles de detectar. Las decisiones importantes requieren la intervención de los miembros más experimentados del equipo y de los responsables de seguridad.

Intervención del Equipo de Seguridad

La participación del equipo de seguridad, puede ofrecer una visión especializada sobre los riesgos implicados en el desarrollo y ayudar a prevenir fallos asegurando el diseño de la aplicación. La ventaja de incorporar la visión de seguridad antes de la implementación es que se ahorra el esfuerzo de modificar código o rediseñar procesos para corregir vulnerabilidades.

Los responsables de seguridad deberán brindar entrenamiento sobre las buenas prácticas de desarrollo seguro al resto de los miembros del equipo. Todas las partes con algún tipo de responsabilidad en el proyecto de desarrollo, pueden beneficiarse recibiendo entrenamiento de seguridad y comprendiendo los riesgos inherentes a producir nuevas aplicaciones.

Principios para un Diseño Seguro.

Algunas vulnerabilidades pueden incorporarse a la aplicación a partir de decisiones de diseño riesgosas. Aplicar principios de diseño seguro ayuda a prevenir este tipo de fallos. Principios a considerar en este enfoque:

Minimizar la Superficie de Ataque.

Cada línea de código es un *bug* en potencia. Cada *bug* puede abrir una vulnerabilidad y ser explotado. Se debe reducir la cantidad de componentes y librerías de las que se dependerá. Esta práctica afecta la cantidad final de errores en el código.

Diseñar para ser Mantenido

Habrá que corregir *bugs*. La arquitectura de la aplicación puede perjudicar o ayudar en su mantenimiento. Será más viable corregir vulnerabilidades de seguridad en un sistema con arquitectura simple y mantenimiento sencillo.

Factores como la complejidad de módulos y arquitectura afectan a la seguridad de la aplicación por el efecto que tienen sobre su mantenibilidad. La rapidez para corregir *bugs* afecta la “ventana de vulnerabilidad”, el plazo durante el que se conoce una falla de seguridad del sistema sin corregir.

El eslabón más débil.

Generalmente, los atacantes utilizarán inicialmente las vulnerabilidades más fáciles de explotar. La arquitectura de seguridad de su aplicación será tan resistente como lo sea el componente con la mínima seguridad.

Es una buena práctica identificar los puntos débiles de la seguridad de la aplicación mediante revisiones de diseño. Y reforzarlos ajustando el diseño o implementando controles de seguridad.

Seguridad por Defecto.

Los usuarios no reconfiguran los parámetros de seguridad. Se debe establecer los controles en su configuración más segura aceptable. Y eventualmente ofrecer la posibilidad de deshabilitar algunos de ellos. Se recomienda desalentar configuraciones inseguras.

Mantener la Usabilidad.

El buen software es invisible. Es esencial negociar un equilibrio entre controles de seguridad y la usabilidad del sistema. Este principio también se conoce como “Aceptabilidad psicológica”. Deberá evitarse generar experiencias de usuario que puedan confundir a los usuarios y llevarlos a tomar malas decisiones de seguridad.

Autorización para Todo por Defecto.

Por defecto, debe requerirse autorización para acceder los recursos de la aplicación. En caso de que se decidiera publicar ciertos recursos, debe fundamentarse la decisión en base a reglas de la organización. Y debería hacerse un análisis de los riesgos asociados a la decisión.

Principio de Mínimo Privilegio.

Establece que se debe asignar sólo los permisos necesarios y suficientes a un componente o usuario para realizar una acción específica. Se asigna los permisos lo más tarde posible y se deben revocar inmediatamente en cuanto no son necesarios.

Separación de Responsabilidades.

Cuando un componente falle, que no afecte el resto del sistema. Este principio consiste en aislar los privilegios de los componentes para evitar que interfieran entre sí en caso de que alguno sea vulnerable.

Defensa en Profundidad.

Consiste en establecer controles de seguridad consecutivos que seguirían en pie independientemente de que falle alguno de ellos. Este principio de diseño aumenta notablemente la resistencia del sistema, ofrece defensa contra ataques que encadenan múltiples técnicas de explotación.

Los Controles en el Cliente no son Suficientes.

Sin control sobre el dispositivo en que se ejecuta el código es inviable controlar su comportamiento. Cualquier control implementado en un equipo que le pertenece a terceros puede ser modificado o deshabilitado. Nunca se debe confiar en datos validados en un equipo cliente, un atacante podría manipularlos arbitrariamente.

Ayudar a los Administradores.

El software no puede defenderse a sí mismo de atacantes. La aplicación debe ofrecer herramientas útiles a los administradores para analizar el comportamiento de los usuarios y detectar casos de abuso. También debe permitir conceder, analizar y revocar permisos de acceso a usuarios.

Debe ofrecerse un registro de eventos de seguridad, preferiblemente con reducción de falsos positivos y ruido. Nunca mostrar información personal de los usuarios a los administradores. Se debe proteger especialmente interfaces de acceso administrativo, considerar implementar autenticación multifactor.

Diseño sin secretos.

La buena seguridad no requiere secretos. Se debe diseñar el sistema bajo la premisa de que eventualmente el público conocerá los detalles de su funcionamiento interno. Seguridad mediante oscuridad es el error de confiar en secretos como controles válidos. Existen técnicas de pruebas a ciegas e ingeniería inversa para obtener información sobre el funcionamiento de sistemas teóricamente secretos. No se debe confiar en el secreto de los mecanismos como medida válida de seguridad.

Modelado de Amenazas.

Consiste en revisar el diseño de la aplicación antes de empezar con la implementación para detectar posibles vulnerabilidades. Con la ventaja que permite corregirlas sin tener que modificar código. Se estudia el diseño tentativo del sistema, reduciéndolo a sus componentes principales.

- Se grafica la estructura y las relaciones de confianza que existen entre los componentes.
- Se producen diagramas de flujos de datos.
- Se definen posibles amenazas para las relaciones entre componentes.
- Se clasifica las posibles amenazas con criterios STRIDE, DREAD, OCTAVE.
- Se rankea las amenazas por criticidad.
- Se decide cómo corregirlas.

Ayuda a prevenir vulnerabilidades a nivel arquitectura de diseño. Ayuda a incorporar el principio de defensa en profundidad al diseño de seguridad. Se recomienda dar participación al equipo de seguridad de la organización.