CERTuy
CENTRO NACIONAL DE RESPUESTA A INCIDENTES DE SEGURIDAD INFORMÁTICA

agesic
DESARROLLANDO EL URUGUAY DIGITAL

Uruguay **Presidencia**

Desarrollo seguro: ¿qué es y qué ventajas tiene?

Versión 1 - mayo 2020

Índice

Introducción

El proceso de desarrollo de software ha tenido como prioridad crear y mantener software funcional. Existen diversas metodologías para ayudar en este proceso, pero son pocas las que tienen en cuenta que el software tenga un aspecto seguro, además del funcional.

Resolver los problemas en la etapa de mantenimiento resulta más costoso y complejo. Esto es así ya que solucionar un fallo introducido en la etapa de diseño puede impactar la funcionalidad de la aplicación, por lo que requiere más trabajo y modificaciones sobre el código. Según un estudio reciente hecho por IBM System Science Institute, el costo de resolver una falla en la etapa de mantenimiento puede ser hasta cien veces superior al valor de resolverlo en la etapa de diseño.

Por otra parte, el alcance de una falla en la seguridad de un sistema puede extenderse más allá del sistema en sí mismo, afectando también a otros sistemas conectados con él, como backends o bases de datos, o también utilizarse como parte de una campaña de distribución de malware u otro tipo de ataques a terceros.

Para abordar este problema se empieza a utilizar el concepto de "ciclo de desarrollo seguro", que a grandes rasgos consiste en introducir conceptos de seguridad en todas las etapas del proceso de desarrollo. A modo de ejemplo, se puede introducir el análisis de arquitectura en la etapa de diseño, el análisis de requerimientos de seguridad junto al análisis de los demás requerimientos y la ejecución de pruebas de seguridad junto a las pruebas funcionales.

Ventajas del ciclo de desarrollo seguro

La principal ventaja de adoptar un proceso de desarrollo seguro es la reducción general del riesgo en una organización y el aumento de su confianza. Asimismo, se pueden mencionar otras:

- Introducir la seguridad en el proceso de desarrollo supone un enfoque proactivo que resulta en un software más seguro, ya que, al haberse considerado la seguridad desde un primer momento, permite detectar y solucionar tempranamente las fallas.

- La detección temprana de fallas trae aparejada una reducción de costos, ya que las modificaciones, el tiempo y la complejidad de su solución son menores.

- También permite identificar causas y evitar que se repitan errores comunes de seguridad, por lo que se ve reducida la probabilidad de vulnerabilidades en el producto final.

- Incluir conceptos de seguridad en la etapa de diseño también puede ayudar a mitigar el impacto de la explotación de alguna vulnerabilidad no detectada en el ciclo de desarrollo.

- Genera conciencia de seguridad en los equipos involucrados en el ciclo de desarrollo.

- Hace posible obtener mediciones tangibles sobre el estado de la seguridad en el

software. Esto permite la toma de decisiones informadas en lo que refiere a seguridad, así como su conocimiento y registro.

Herramientas para alcanzar un ciclo de desarrollo seguro

Actualmente, existen distintos marcos de referencia, estándares y herramientas que pueden ser adaptados al ciclo de desarrollo de una organización en base a sus objetivos y prioridades.

OWASP - SAMM

Software Assurance Maturity Model es el marco de referencia de OWASP y está orientado a que las organizaciones logren evaluar, formular e implementar una estrategia para la seguridad del software en una forma medible y efectiva.

Soporta el ciclo de desarrollo completo y no está limitado a una tecnología específica. Es un sistema flexible, que permite una adopción gradual proponiendo tres niveles de madurez. Las organizaciones pueden usarlo para conocer su nivel de madurez actual de seguridad en su software y definir estrategias y prioridades para avanzar al siguiente nivel.

OWASP - ASVS

Application Security Verification Standard es un marco de referencia para requerimientos y controles de seguridad. Consiste en una guía técnica que provee una lista de requerimientos funcionales y no funcionales para el desarrollo seguro y que ofician como base para el testing de controles técnicos de seguridad. Puede usarse como un checklist para medir el estado de seguridad de una aplicación o también adaptarse a sus casos de uso.

ASVS propone tres niveles de verificación, de acuerdo a la criticidad de la información tratada, y cada nivel contiene una lista de requerimientos de seguridad asociada.

Microsoft SDL

Es una metodología desarrollada por Microsoft y aplicada a su proceso de desarrollo interno. Consiste en una serie de prácticas y consideraciones de alto nivel a ser aplicadas en cada una de las fases del proceso de desarrollo.

NIST SSDF

Aunque se trata de un borrador al momento de la redacción, SSDF pretende ser un marco de referencia basado en algunos de los mencionados anteriormente entre otros. El enfoque, en este caso, es elaborar requerimientos y lineamientos de alto nivel que puedan ser aplicados a cualquier metodología de desarrollo.

Recomendaciones para comenzar con un ciclo de desarrollo seguro.

- Implementar capacitaciones en prácticas de desarrollo seguro y marcos de referencia disponibles.
- Adoptar el marco de referencia más ajustado a las necesidades de la organización.
- Determinar qué políticas existen actualmente en la organización y qué tan efectivas son.
- Implementar entrenamientos por roles a los equipos involucrados en el ciclo de desarrollo.
- Tener en cuenta la seguridad del producto al momento de hacer los análisis de diseño y requerimientos y también en la elaboración de casos de prueba.
- Definir metas de seguridad de realizables y posibles de ser medidas de forma objetiva.
- Implementar análisis y escaneo de vulnerabilidades y hacer un seguimiento de ellas.
- Implementar escaneos estáticos sobre el código fuente.

Referencias

- IBM System Science Institute, “Integrating software assurance into the software development life cycle (SDLC)”. Disponible en internet: https://www.researchgate.net/figure/IBM-System-Science-Institute-Relative-Cost-of-Fixing-Defects_fig1_255965523
- (Fecha de última consulta: 8 de mayo de 2020).

- MICROSOFT, Microsoft Security Development Lifecycle. Disponible en internet: https://www.microsoft.com/en-us/securityengineering/sdl/ (Fecha de última consulta: 8 de mayo de 2020).

- NIST, “Mitigating the Risk of Software 1 Vulnerabilities by Adopting a Secure 2 Software Development Framework (SSDF)”. Disponible en internet: https://csrc.nist.gov/CSRC/media/Publications/white-paper/2019/06/07/mitigating-risk-of-software-vulnerabilities-with-ssdf/draft/documents/ssdf-for-mitigating-risk-of-software-vulns-draft.pdf (Fecha de última consulta: 8 de mayo de 2020).

- OWASP, OWASP Application Security Verification Standard. Disponible en internet: https://owasp.org/www-project-application-security-verification-standard/ (Fecha de última consulta: 8 de mayo de 2020).

- OWASP, OWASP SAMM. Disponible en internet: https://owasp.org/www-project-samm/ ((Fecha de última consulta: 8 de mayo de 2020).