ENTORNOS DE DESARROLLO

FORO EVALUABLE I

ALUMNO CESUR

24/25

Alejandro Muñoz de la Sierra

PROFESOR

Diego Tinedo Rodríguez

Introducción

En el desarrollo de software, la seguridad es muy importante y debe ser considerada desde el inicio hasta el mantenimiento. A medida que los sistemas se vuelven más complicados y conectados, los riesgos de vulnerabilidades aumentan, como se vio con el caso Log4Shell en 2021. Este evento, que afectó a la biblioteca Log4j en muchas aplicaciones, destaca la necesidad de tener estrategias de seguridad firmes en todas las partes del ciclo de vida del software.

El ciclo de vida del desarrollo tiene fases claras, como análisis, diseño, codificación, pruebas, explotación y mantenimiento. Cada fase es importante para proteger el sistema de ataques o errores. Sin embargo, algunas fases afectan más la prevención de vulnerabilidades desde el principio.

En este texto, se argumentará que la fase de diseño es la más importante para asegurar la seguridad del software, ya que en esta etapa se crean las bases del proyecto. Con un enfoque preventivo en el diseño, no solo se reducen los riesgos en etapas futuras, sino que también se optimizan los costos y se evita tener que hacer complejas correcciones en producción.

Fase de desarrollo seleccionada: Diseño

La seguridad en el desarrollo de software es muy importante y debe pensarse en todas las etapas. Sin embargo, la fase de diseño es crucial para asegurar que el sistema sea sólido desde el inicio. En esta etapa se establecen las bases del sistema, definiendo su estructura, cómo interactúan los componentes y las políticas de seguridad que afectarán al proyecto.

Justificación:

Prevención de vulnerabilidades desde el inicio
En la fase de diseño, se decide cómo se organizará el software y qué medidas de seguridad se incorporarán. Una mala decisión aquí puede resultar en brechas de seguridad que sean difíciles y caras de corregir más adelante.
Ejemplo: Si no se elige un sistema de autenticación fuerte desde el inicio, la aplicación podría ser accesible sin autorización.

Incorporación de principios de diseño seguro :

Durante esta etapa, se deben incluir estrategias de seguridad, tales como:

- Principio de menor privilegio: Asegura que cada usuario o componente acceda solo a lo necesario.
- Defensa en profundidad: Establece múltiples capas de seguridad para reducir riesgos.
- Validación de entrada: Protege contra ataques como inyecciones SQL y otros problemas como los causados por la vulnerabilidad Log4Shell.

Aplicar estos principios desde el diseño ayuda a evitar complicaciones y costos posteriores.

Definición de arquitectura y estándares:

El diseño debe considerar herramientas y librerías que sean seguras.
Ejemplo: Si se hubiera planificado un sistema de gestión de dependencias, este habría vigilado librerías externas como Log4j y habría avisado sobre vulnerabilidades. Establecer una política de actualización continua habría reducido riesgos asociados a Log4Shell.

Reducción de costos y riesgos:

Encontrar y corregir problemas de seguridad durante el diseño es mucho más barato que hacerlo en la fase de producción. Los errores descubiertos en producción, como el caso de Log4Shell, conllevan altos costos financieros y un posible daño reputacional complicado de recuperar.

Conclusión

La seguridad en el desarrollo de software es un proceso que nunca termina y debe incluir todas las etapas del proyecto, desde el inicio hasta el mantenimiento. Sin embargo, la fase de diseño es la más importante para evitar vulnerabilidades desde el principio. Durante el diseño, se crean las bases arquitectónicas, las normas de acceso y los principios de seguridad que sostendrán al sistema contra amenazas.

Casos como Log4Shell, que afectaron muchas aplicaciones en todo el mundo, muestran que no tratar problemas de seguridad en las primeras fases puede llevar a consecuencias graves y costosas en producción. Diseñar con un enfoque en la seguridad desde el comienzo no solo reduce estos riesgos, sino que también mejora los costos, asegura que se cumplen las normas y aumenta la confianza en la calidad del software.

En conclusión, aunque todas las etapas del desarrollo de software son importantes para mantener un sistema seguro, incorporar medidas de prevención durante el diseño es clave para crear aplicaciones fuertes y resistentes a las amenazas actuales y futuras. Este enfoque activo garantiza sistemas más seguros y un producto final más confiable y de mejor calidad.

Referencias

https://www.wbassetstudio.com/blog/diseno-de-software-fases-y-modelos/#elementor-toc__heading-anchor-4

https://fidiaspro.com/las-10-fases-que-debe-seguir-el-desarrollo-de-un-buen-software/

https://secureframe.com/es-es/blog/secure-by-design

https://learn.microsoft.com/es-es/azure/well-architected/security/principles

https://www.linkedin.com/pulse/7-pasos-para-implementar-patrones-de-dise%C3%B1o-seguros/

https://www.cisa.gov/sites/default/files/2024-10/principles_approaches_for_security-by-design-default_508c%20%285%29_ES.pdf

https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay

https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/

https://icariatechnology.com/la-calidad-del-software-y-sus-estandares-mas-importantes/

https://www.hiberus.com/crecemos-contigo/los-estandares-de-calidad-del-software-mas-importantes/