Buenas Practicas

- Validación de entradas
- Codificación de salidas
- Estandarización y reutilización de funciones de seguridad
- Manejos de errores y logs
- Estilo de programación

La seguridad no es un producto, es una sumatoria de personas, procesos y tecnología.

"Las empresas invierten millones en firewalls, cifrado y dispositivos para acceder de forma segura, y es dinero malgastado, porque ninguna de estas medidas corrige el nexo más débil de la cadena"

UNIVERSIDAD DE COLIMA

Principios de seguridad en el desarrollo de software

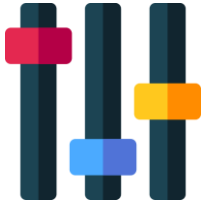
Creado por

Francisco Fidel Castro Barreto

¿Qué es el desarrollo seguro de software?

El desarrollo seguro de software es un modelo de trabajo que se basa en la realización de chequeos de seguridad continuos del proyecto en construcción, incluso desde sus fases iniciales y antes de que se escriba una sola línea de código.

El objetivo es, al fin y al cabo, asegurarnos de que impediremos el acceso al programa y a los datos almacenados por parte de usuarios carentes de permiso.



Principios básicos para un desarrollo

- Si se utiliza un lenguaje que no sea compilado, asegurarse de limpiar el código que se pone en producción, para que no contenga rutinas de pruebas, comentarios o cualquier tipo de mecanismo que pueda dar lugar a un acceso indebido.
- Nunca confiar en los datos que ingresan a la aplicación, todo debe ser verificado para garantizar que lo que está ingresando a los sistemas es lo esperado y además evitar inyecciones de código.
- Hacer un seguimiento de las tecnologías utilizadas para el desarrollo. Estas van evolucionando y cualquier mejora que se haga puede dejar obsoleta o inseguras versiones anteriores.

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.
- Ataques: Se debe operar bajo la premisa de que la aplicación va a recibir ataques variados periódicamente por lo tanto el software debe ser tolerante a fallos.
- Defensa en profundidad: Consiste en establecer controles de seguridad consecutivos que seguirían en pie independientemente de que falle alguno de ellos.