INSTITUTO TECNOLÓGICO DE TLALNEPANTLA

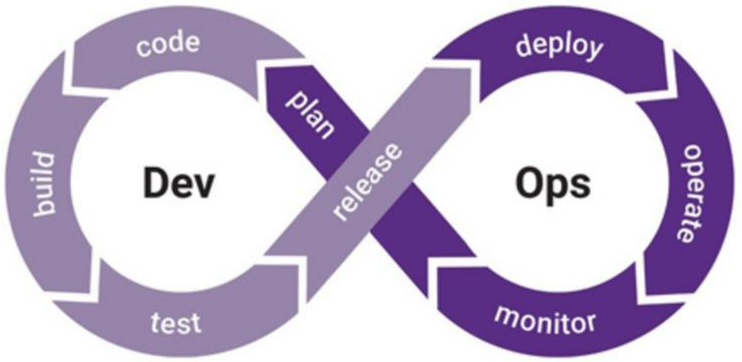
Software seguro

Docente: Hilda Díaz Rincón
Integrantes:
- Espino Horta Maria Jose
- Calderón Hernández Miguel Eduardo
- Alcántara Salazar Joel

T92 13/09/2021

S-SDLC – Secure Software Development Life Cycle

Software de
confianza y robusto frente a taques maliciosos, que realice solo las funciones para las que fue diseñado, que esté libre de vulnerabilidades, ya sean intencionalmente diseñadas o accidentalmente insertadas durante su ciclo de vida y se asegure su integridad, disponibilidad y confidencialidad”.



S-SDLC – Propiedades elementales Software Seguro

- Integridad: capacidad que garantiza que el código del

software, activos manejados, configuraciones y comportamientos no puedan ser o no hayan sido

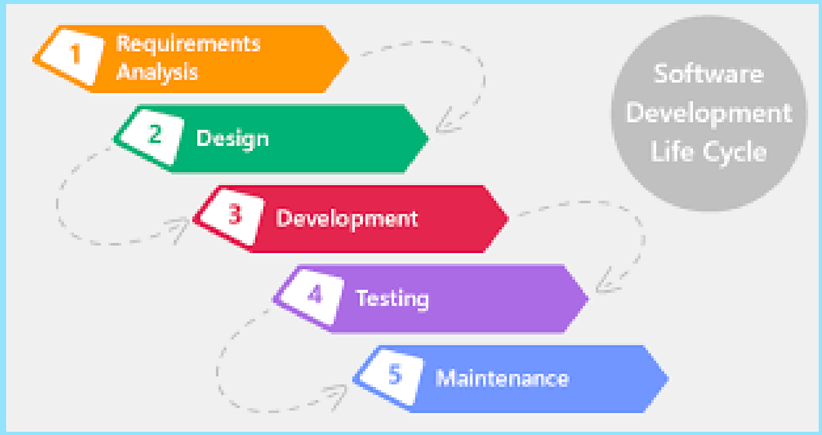
modificados o alterados.

- Disponibilidad: capacidad que garantiza que el software

es operativo y accesible por usuarios.

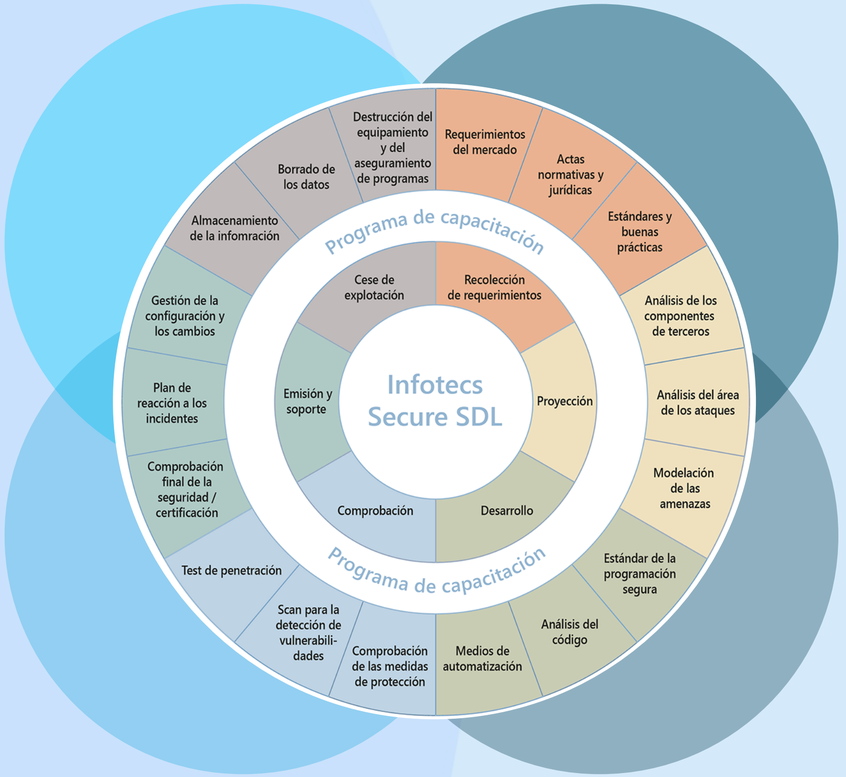
- Confidencialidad: capacidad de preservar que cualquiera

de sus características, activos manejados, están ocultos a usuarios no autorizados



Ciclo de Vida del Desarrollo de productos de Ciberseguridad

Realización de medidas para el desarrollo de programas seguros en todas las etapas del ciclo de vida (SDLC – Secure Software Development Lifecycle) es una condición de obligatorio cumplimiento para garantizar la competitividad en el mercado para las compañías que se dedican al desarrollo de



Recopilación de los requerimientos

- **Requerimientos del mercado**

La recopilación de los requerimientos del mercado incluye la detección de problemas del usuario, que pueden ser resueltos con ayuda del producto o del servicio. En esta etapa es muy importante definir muchos detalles, por ejemplo, el tipo de datos a proteger, las funciones necesarias del producto, las posibles restricciones relacionadas con el medio de funcionamiento del producto etc.

- **Legislación vigente**

La consideración de toda la legislación y normas en la etapa temprana facilita la disminución de los gastos, los riesgos de no compatibilidad, así como la fecha de puesta en el mercado del producto en su conjunto.

- **Estándares y mejores prácticas**

A pesar de que la legislación por lo general no exige el cumplimiento de los estándares y las mejores prácticas, su uso es una de las condiciones importantes para asegurar la competitividad de los productos y de la compañía.

Proyección

En esta etapa se analiza detalladamente los requerimientos recolectados con el objetivo de definir la forma de realización de esos requerimientos en el producto, así como se determina la forma más confiable y segura de su realización. Si está planificado el uso de programas de terceros, primero se revisan de forma detallada y se realiza un análisis del área de ataques (además de que se elabora el modelo del atacante) y se genera el modelo de las amenazas.

- Análisis de componentes de terceros
- Análisis del área de los ataques
- Creación del modelo de amenazas

Desarrollo

El desarrollo de aseguramiento de programas seguro ante todo está orientado a que no surjan errores en el código en la etapa de desarrollo, lo que se puede lograr mediante:

- La introducción de un rígido estándar de programación segura y el control de su cumplimiento mediante el análisis del código de programación.
- El empleo de instrumentos de automatización modernos (analizadores estáticos, compiladores, sistemas de control de las versiones, etc.) en el ambiente de desarrollo del aseguramiento de programas

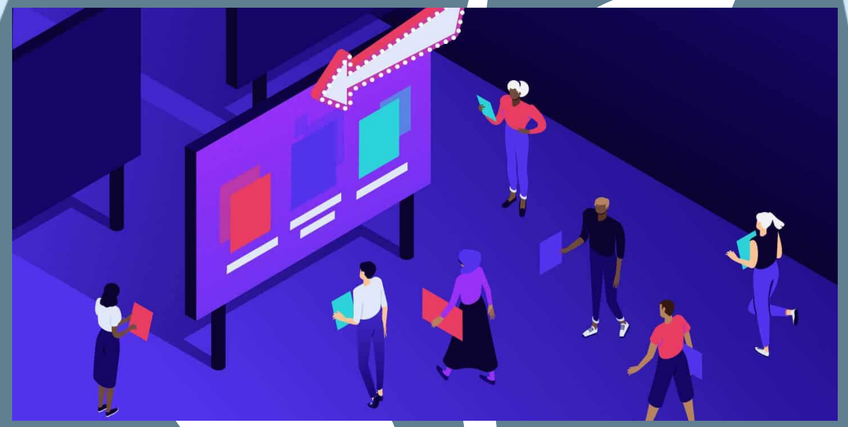
Comprobación

- El objetivo de esta etapa es la comprobación de que el producto cumple con los requerimientos de nivel necesario de seguridad. Para la valoración funcional del producto es conveniente realizar una comprobación (test) integracional, regresiva y modular.

Emisión y soporte del producto

LEl lanzamiento del producto al mercado se realiza solamente después de la realización exitosa de la comprobación final de la seguridad (FSR – Final Security Review), ejecutada por un equipo compuesto por los desarrolladores del producto y expertos en seguridad. Además, nosotros hemos desarrollado un plan de reacción a los incidentes para la erradicación de los posibles problemas de seguridad en nuestros productos.a

Salida de explotación



Cuando llega el momento de término de la explotación del producto (en el caso de su obsolescencia, necesidad de renovación u otras causas) no se puede olvidar que la información a proteger debe mantenerse protegida. Para ello es necesario planificar de forma anticipada y definir el proceso de salida del producto del mercado

Los 10 principios básicos para un desarrollo seguro

1. Partir siempre de un modelo de permisos mínimos, es mejor ir escalando privilegios por demanda de acuerdo a los perfiles establecidos en las etapas de diseño.
2. 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.
3. 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.
4. 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.
5. Todos los accesos que se hagan a los sistemas deben ser validados.

Los 10 principios básicos para un desarrollo seguro

6. Para intercambiar información sensible utilizar protocolos para cifrar las comunicaciones, y en el caso de almacenamiento la información confidencial debería estar cifrada utilizando algoritmos fuertes y claves robustas.
7. Cualquier funcionalidad, campo, botón o menú nuevo debe agregarse de acuerdo a los requerimientos de diseño. De esta forma se evita tener porciones de código que resultan siendo innecesarias.
8. La información almacenada en dispositivos móviles debería ser la mínima, y más si se trata de contraseñas o datos de sesión. Este tipo de dispositivos son los más propensos a ser que se pierdan y por lo tanto su información puede ser expuestas más fácilmente.
9. Cualquier cambio que se haga debería quedar documentado, esto facilitará modificaciones futuras.
10. Poner más cuidado en los puntos más vulnerables, no hay que olvidar que el nivel máximo de seguridad viene dado por el punto más débil.

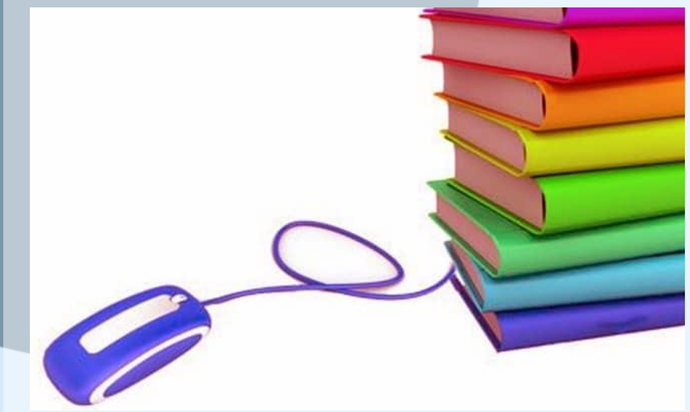
El arte del desarrollo seguro

Cuando se piensa en desarrollo seguro, desarrollo se software seguro, se suele acudir al código, buscando líneas de código concretas, o buenas prácticas que nos salven de un posible incidente de seguridad. Esta practica es muy beneficiosa para el software, pero también es incompleta.

Principio de diseño seguro: Seguir el principio de mínimo privilegio

- Un usuario o proceso, debe tener disponible la información y recursos que requiere para el desempeño de su función de su función y no mas. La funcionalidad disponible debe ser solo y exclusivamente la necesaria, cualquier posibilidad de acceso o ejecución de una acción

Referencias:



- Auditor. (2021, abril 24). El arte del desarrollo seguro (diseño seguro). Auditoriadecodigo.com. https://auditoriadecodigo.com/desarrollo-seguro-es-diseno-seguro/

- Ciclo de Vida del Desarrollo de la Seguridad. (s/f). Infotecs.mx. Recuperado el 13 de septiembre de 2021, de https://infotecs.mx/support/security-development-lifecycle/

- Enero, B. D. C. (s/f). GUÍA DE DESARROLLO SEGURO. Gov.co. Recuperado el 13 de septiembre de 2021, de https://www.uspec.gov.co/wp-content/uploads/2021/01/A3-GU-02_Guia_de_Desarrollo_Seguro_V02.pdf

- Fast Track. (s/f). Secure software development life cycle. Owasp.org. Recuperado el 13 de septiembre de 2021, de https://owasp.org/www-pdf-archive/OWASP-LATAMTour-Patagonia-2016-rvfigueroa.pdf

- Los 10 principios básicos para un desarrollo seguro. (s/f). Welivesecurity.com. Recuperado el 13 de septiembre de 2021, de https://www.welivesecurity.com/la-es/2014/02/28/10-principios-basicos-para-desarrollo-seguro/

¡Stickers!