

**GASIC 8– Seguridad en el Desarrollo y Adquisición de Software**

**Preguntas de Auditoría**

**Funcionalidad de Software Existente**

**Competencias Básicas del Auditor**

1. **Conocimiento Profundo en Desarrollo de Software Seguro:** Entender los principios de seguridad en el desarrollo de software, incluyendo buenas prácticas de codificación, diseño seguro, y consideraciones de seguridad en el ciclo de vida del desarrollo.
2. **Experiencia en Identificación de Vulnerabilidades:** Tener habilidades para identificar y evaluar vulnerabilidades en el código y el diseño del software, incluyendo análisis estático y dinámico, pruebas de penetración y revisión de arquitecturas.
3. **Conocimiento en Criptografía y Protección de Datos:** Comprender los principios de la criptografía y cómo se aplica en la protección de datos sensibles dentro del software, incluyendo la gestión segura de claves y la integridad de los datos.
4. **Habilidades en Pruebas de Seguridad y Evaluación de Riesgos:** Ser capaz de realizar pruebas de seguridad exhaustivas para evaluar la resistencia del software frente a posibles ataques, así como evaluar los riesgos asociados con el software.
5. **Conocimientos en Cumplimiento y Normativas:** Estar familiarizado con los requisitos de cumplimiento y normativas relevantes en el ámbito de la seguridad del software, como GDPR, PCI DSS, y normas ISO/IEC.
6. **Capacidad Analítica:** Ser capaz de analizar datos de seguridad y tendencias para identificar posibles amenazas y vulnerabilidades emergentes en el software.
7. **Habilidades de Comunicación Técnica:** Ser capaz de comunicar de manera efectiva los hallazgos de seguridad a los desarrolladores, gerentes y otros interesados, y proponer recomendaciones claras y prácticas.
8. **Conocimientos en Gestión de Incidentes y Respuesta a Incidentes:** Entender los procedimientos y prácticas para la gestión de incidentes de seguridad en el software, incluyendo la capacidad de respuesta rápida y eficaz.

# Visión General

La gestión y evaluación de la funcionalidad del software existente son esenciales para mantener la seguridad en entornos que integran componentes de terceros y código abierto. Adoptar prácticas estructuradas para evaluar, configurar y actualizar estos componentes garantiza su compatibilidad y seguridad dentro del software de la organización. Desde la perspectiva de auditoría, es crucial que estos procesos estén documentados y que los componentes cumplan con los estándares de seguridad establecidos.

**1.1 Evaluación y Procedencia de Componentes de Software**

Los componentes de software de terceros deben revisarse en función de su propósito previsto y su procedencia. Esto incluye analizar los riesgos asociados a cada componente y verificar que cumplan con las configuraciones seguras definidas por la organización. Además, mantener repositorios para alojar componentes aprobados permite un control centralizado y garantiza que los desarrolladores utilicen únicamente versiones examinadas. Desde el punto de vista de auditoría, se revisará si estos repositorios están actualizados y si los componentes aprobados cumplen con los estándares de seguridad.

**1.2 Gestión de Repositorios y Componentes**

La organización debe establecer y mantener repositorios dedicados para componentes aprobados, tanto comerciales como de código abierto. Estos repositorios deben estar documentados y accesibles para los desarrolladores, quienes deben seguir las prácticas de desarrollo seguro al utilizarlos. Un auditor verificará si las listas de componentes aprobados están alineadas con las políticas internas, y si los procesos de selección e inclusión de componentes están formalizados.

**1.3 Actualización y Monitoreo de Vulnerabilidades**

Actualizar periódicamente los componentes de software es fundamental para mitigar riesgos asociados con vulnerabilidades conocidas. La detección automática de vulnerabilidades en la cadena de herramientas debe integrarse para identificar problemas antes de que afecten la seguridad del software. Además, es importante mantener versiones anteriores de los componentes hasta que las transiciones a nuevas versiones se completen de manera segura. Un auditor evaluará si estas actualizaciones son consistentes y si se manejan conforme a los procedimientos establecidos.

**1.4 Mantenimiento y Soporte Continuo**

Los componentes de software deben ser monitoreados para garantizar que estén activos y mantenidos, lo que incluye verificar que no hayan alcanzado el fin de su vida útil. Este monitoreo debe incluir la revisión de vulnerabilidades recién identificadas y la confirmación de la integridad de los componentes mediante mecanismos como firmas digitales. Desde una perspectiva de auditoría, estos procesos deben estar documentados y demostrar que los componentes se gestionan de manera proactiva para garantizar su seguridad.

**1.5 Pruebas y Validación**

El código de los componentes debe ser revisado, analizado y probado regularmente para validar su funcionalidad y cumplimiento con los estándares de la organización. Estas actividades deben ser registradas y formar parte del proceso de evaluación de riesgos asociados con el uso de componentes de terceros. Un auditor verificará si las pruebas se realizan de manera sistemática y si los hallazgos se integran en las decisiones sobre la utilización de los componentes.

# Modelo de Madurez

|  |  |
| --- | --- |
| NIVEL 1 [Impredecible y reactivo]: La actividad se completa, pero a menudo se retrasa y supera el presupuesto. | 1. Revisar y evaluar componentes de software de terceros en el contexto de su uso esperado. 2. Determinar configuraciones seguras para componentes de software y ponerlas a disposición para que los desarrolladores puedan utilizar las configuraciones fácilmente. 3. Obtener información de procedencia para cada componente de software y analizar esa información para evaluar mejor el riesgo que el componente puede introducir. 4. Establecer uno o más repositorios de software para alojar componentes de código abierto aprobados y examinados. 5. Mantener una lista de componentes de software comercial aprobados por la organización y versiones de componentes junto con sus datos de procedencia. |
| NIVEL 2 [Gestionado a nivel de proceso]: Las actividades se planifican, ejecutan, miden y controlan. | 1. Seguir las prácticas de seguridad establecidas por la organización para el desarrollo de software seguro al crear y mantener los componentes. 2. Mantener uno o más repositorios de software para estos componentes. 3. Designar qué componentes deben incluirse en el software a desarrollar. |
| NIVEL 3 [Proactivo antes que reactivo]: Los estándares de toda la organización brindan orientación a través de proyectos, programas y carteras. | 1. Implementar procesos para actualizar los componentes de software implementados a versiones más nuevas y conservar las versiones anteriores de los componentes de software hasta que todas las transiciones desde esas versiones se hayan completado con éxito. 2. Comprobar periódicamente si existen vulnerabilidades conocidas públicamente en los módulos y servicios de software que los proveedores aún no han solucionado. 3. Incorporar en la cadena de herramientas la detección automática de vulnerabilidades conocidas en componentes de software. 4. Utilizar los resultados existentes de servicios comerciales para examinar los módulos y servicios de software. 5. Asegurarse de que cada componente de software todavía se mantenga activamente y no haya llegado al final de su vida útil; esto debería incluir nuevas vulnerabilidades encontradas en el software que se está reparando. 6. Confirmar la integridad de los componentes de software mediante firmas digitales u otros mecanismos. 7. Revisar, analizar y/o probar código. |
| NIVEL 4 [Gestionado Cuantitativamente]: La organización está basada en datos con objetivos cuantitativos de mejora del desempeño que son predecibles |  |
| NIVEL 5 [Optimizado]: La organización se centra en la mejora continua y está diseñada para girar y responder a las oportunidades y los cambios. |  |

1. **Preguntas de Auditoría para Funcionalidad de Software Existente**

|  |
| --- |
| Temática: Evaluación de Componentes de Software de Terceros |
| La revisión y evaluación de componentes de software de terceros es esencial para asegurar que estos componentes cumplan con los estándares de seguridad y funcionen correctamente en el contexto de su uso esperado. Evaluar estos controles garantiza que los componentes de terceros no introduzcan riesgos innecesarios y que se utilicen de manera segura dentro de la infraestructura de software de la organización.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Se revisan y evalúan regularmente los componentes de software de terceros en el contexto de su uso esperado? * ¿Existen procesos documentados para la evaluación de seguridad de estos componentes antes de su integración? * ¿Se determinan y documentan configuraciones seguras para los componentes de software? * ¿Están estas configuraciones seguras fácilmente disponibles para los desarrolladores? * ¿Se obtiene y analiza información de procedencia para cada componente de software utilizado? * ¿Cómo se utiliza esta información para evaluar y mitigar el riesgo que cada componente puede introducir? * ¿Se han establecido uno o más repositorios de software para alojar componentes de código abierto aprobados y examinados? * ¿Estos repositorios se mantienen y actualizan regularmente? * ¿Existe y se mantiene una lista de componentes de software comercial aprobados por la organización? * ¿Incluye esta lista versiones específicas de componentes y sus datos de procedencia? |

|  |
| --- |
| Temática: Prácticas de Seguridad en el Desarrollo de Componentes de Software |
| Seguir prácticas de seguridad establecidas es crucial para el desarrollo y mantenimiento de componentes de software seguros. Evaluar estos controles asegura que los desarrolladores apliquen consistentemente medidas de seguridad en todas las etapas del ciclo de vida del software, reduciendo así el riesgo de vulnerabilidades y fallos de seguridad.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Se siguen las prácticas de seguridad establecidas por la organización durante el desarrollo y mantenimiento de componentes de software? * ¿Existen políticas y procedimientos documentados que guíen estas prácticas? * ¿Se mantienen uno o más repositorios de software para los componentes desarrollados? * ¿Estos repositorios se supervisan y actualizan regularmente para asegurar su integridad? * ¿Se designa claramente qué componentes deben incluirse en el software a desarrollar? * ¿Existe un proceso formal para la selección y aprobación de estos componentes? |

|  |
| --- |
| Temática: Actualización y Mantenimiento de Componentes de Software |
| Implementar procesos para la actualización y mantenimiento de componentes de software es vital para asegurar que los sistemas permanezcan seguros y funcionales. Estos controles garantizan que los componentes se actualicen a versiones más nuevas, que se monitoricen las vulnerabilidades conocidas y que se mantenga la integridad y seguridad de los componentes a lo largo del tiempo.  Este levantamiento puede incluir análisis específicos tales como:   * ¿Existen procesos documentados para actualizar los componentes de software implementados a versiones más nuevas? * ¿Se conservan las versiones anteriores de los componentes de software hasta que todas las transiciones se hayan completado con éxito? * ¿Se comprueba periódicamente la existencia de vulnerabilidades conocidas públicamente en los módulos y servicios de software? * ¿Cómo se manejan las vulnerabilidades que los proveedores aún no han solucionado? * ¿Se ha incorporado en la cadena de herramientas la detección automática de vulnerabilidades conocidas en componentes de software? * ¿Cómo se gestionan los resultados de estas detecciones? * ¿Se utilizan los resultados de servicios comerciales para examinar los módulos y servicios de software? * ¿Cómo se integran estos resultados en los procesos de desarrollo y mantenimiento? * ¿Se asegura de que cada componente de software todavía se mantenga activamente y no haya llegado al final de su vida útil? * ¿Se incluyen las nuevas vulnerabilidades encontradas en el software en los procesos de reparación? * ¿Se confirma la integridad de los componentes de software mediante firmas digitales u otros mecanismos? * ¿Se revisan y verifican regularmente estas firmas digitales? * ¿Se revisa, analiza y prueba el código regularmente para identificar y corregir posibles vulnerabilidades? * ¿Existen procesos documentados para la revisión y prueba de código? |