GOBIERNO REGIONAL DE LOS LAGOS

Acción de Futuro

SISTEMA DE SEGURIDAD DE LA INFORMACIÓN

PROCEDIMIENTO "PRINCIPIOS DE INGENIERÍA DE SISTEMA SEGURO"

CÓDIGO	**SSI-A.14.02.05**	**CLASIFICACIÓN INFORMACIÓN**		**Reservada**
			X	**Uso Interno**
				Pública
VERSIÓN	1.0	**FECHA DE LA VERSIÓN**	24-10-2019	
RESPONSABLE	Encargado de la Unidad de Informática.			

Historial de modificaciones

Creación/Modificaciones del Documento

Versión	Fecha de modificación	Autor	Motivo	Páginas modificadas	Firma
1.0	24-10-2019	Oscar Alejandro Oyarzo Pérez, Encargado de Seguridad de la Información	Creación de la primera versión del procedimiento	Todas	

Revisiones

Versión	Fecha	Autor	Observación	Páginas	Firma
1.0	26-10-2019	Javier Soto Mancilla, Apoyo Profesional DAF	Sin observaciones	Todas	

Visto Bueno

Versión	Fecha	Encargado	Firma
1.0	05-11-2019	Alejandro Montaña Ampuero, Administrador Regional del Gobierno Regional de Los Lagos	GOBIERNO REGIONAL ADMINISTRADOR REGIONAL DE LOS LAGOS
1.0	05-11-2019	Oscar Alejandro Oyarzo Pérez, Encargado de Seguridad de la Información	

Distribuciones

Versión	Fecha	Encargado	Observaciones
1.0	06-12-2019	Nicanor Bahamonde Loustau Profesional Unidad de Informática	Enviado por correo electrónico e instruida la publicación en intranet a la Unidad de Informática

Contenido

1. Objetivo

Establecer, documentar, mantener y aplicar los principios para la ingeniería de sistemas seguros para cualquier labor de implementación del sistema de información.

2. Alcance o ámbito de aplicación interno

El presente procedimiento, establecerá el lineamiento para todos los Desarrollos de Software que se realicen en el GORE Los Lagos tanto por profesionales internos como profesionales o empresas externas al Servicio.
Este procedimiento es aplicable a todos los funcionarios (planta, contrata, reemplazos y suplencia), personal a honorarios y terceros, es decir, proveedores, que presten servicios al Gobierno Regional de Los Lagos.

3. Roles y Responsabilidades.

ROLES	RESPONSABILIDADES
Encargado de Seguridad de la Información	• Responsable de la aprobación de este documento.
Encargado Unidad de Informática	• Encargado de velar por el cumplimiento de los lineamientos para un desarrollo seguro indicados en este documento.

4. Procedimiento

Los lineamentos expuestos en este documento están basados en los pilares de la Seguridad de la Información confidencialidad en cuanto a los accesos a los activos del sistema, los cuales deberán limitarse únicamente a usuarios autorizados, integridad basándose en que los activos del sistema solo podrán ser borrados o modificados por usuarios que se encuentren autorizados, disponibilidad en cuanto que se garantice a los usurarios accesos a los activos en un tiempo razonable y que estos siempre se encuentren disponible.

En las etapas de requerimientos, análisis y diseño.

Requisitos de seguridad:
Se debe identificar los objetivos y requisitos de seguridad que se requieren contemplar e implementar, estos se pueden determinar basado en lo siguiente:

- Arquitectura que va emplear la aplicación.
- Plataforma donde correrá la aplicación.
- Tipos de datos que se almacenan, consultarán o se transferirán, es decir se debe definir cuáles son confidenciales y/o públicos.
- Tipos de registros del sistema que debe generar, privilegios, perfiles de usuarios (lectura, escritura, modificación y eliminación).
- Definir el modo de autentificación y contraseñas.
- Contemplar los riesgos del proyecto en los cuales se debe considerar lo siguiente:
    - Mala planeación en los tiempos asignados.
    - Inhabilidades o incapacidad durante el desarrollo por parte del equipo de trabajo.

- Indebida clasificación del tipo de información (confidencial/publica), roles y permisos.
- Definición de requerimientos pocos claros o incompletos.

- Definir acciones correctivas y/o preventivas del proyecto, como las siguientes:
  - Establecer tiempos de holgura previniendo imprevisto.
  - Definir responsables que puedan suplir la ausencia de personas que conforman el equipo de trabajo.

Lista de chequeo

Se debe realizar una verificación para verificar el cumplimiento de Desarrollo de Software Seguro.

De ser necesario según la particularidad de cada proyecto se pueden adicionar o modificar los chequeos.

Administración de autentificación y contraseñas

A continuación, se indican lo que se debe tener en cuenta para la administración, autentificación y manejo de contraseñas.

- Si los usuarios requieren realizar algún tipo de transacciones, se debe requerir un proceso de autentificación previa dentro del portal o aplicación.
- Los controles de autentificación se deben realizar en sistemas confiables (por ejemplo, en el servidor).
- Se deben establecer y utilizar servicios de autentificación estándares.
- Todos los controles de autentificación deben fallar en forma segura. En el caso de existir un error u excepción del sitio web no deberá proveer información relacionada con información de autentificación del usuario.
- Si la aplicación administra almacenamiento de credenciales, se debe asegurar que únicamente se almacena el salty hash de la contraseña y que el registro que guarda la contraseña y claves solo pueda ser escrito por la aplicación.
- El hash de la contraseña debe ser implementado en un sistema confiable (ej. Servidor).
- Validar los datos de la autentificación únicamente después de haber completado todos los datos de entrada, especialmente en implementaciones secuenciales.
- Las respuestas a fallos en la autentificación, no deben indicar que parte de la autentificación es la incorrecta. Por ejemplo, utilizar mensajes como "" usuario y/o contraseña inválidos "en vez de "contraseña inválida" o "usuario inválido".
- Utilizar autentificación para conexiones a sistemas externos que involucren información o funciones sensibles.
- Las credenciales de autentificación para acceder a servicios externos a las aplicaciones deben ser cifradas y almacenadas en ubicaciones protegidas en un sistema confiable. El código fuente NO es una ubicación segura.
- Utilizar únicamente solicitudes del tipo HTTP POST para transmitir las credenciales de autentificación.
- Utilizar únicamente conexiones cifradas o datos cifrados para el envío de contraseñas que no sean temporales,
- No se debe desplegar en pantalla las contraseñas ingresadas.
- Bloquear cuentas después de un número establecido de intentos fallidos de ingreso al sistema. El número ideal para el bloqueo de cuentas son 3 intentos fallidos.
- Cambio y reinicio de contraseñas requiere de los mismos niveles de control, que aquellos asociados a la creación y autentificación de cuentas.
- Las preguntas para el reinicio de contraseñas deben contemplar un amplio rango de respuesta aleatorias.
- Si se utiliza reinicio por correo electrónico, únicamente enviar un link o contraseñas temporales a cuentas previamente registradas.
- Las contraseñas y link temporales deben tener un período de validez corto.
- Forzar el cambio de contraseñas temporales después de su utilización.
- Notificar a los usuarios cada vez que se realice un reinicio de contraseñas.
- Implementar monitoreo para identificar ataques a múltiples cuentas utilizando la misma contraseña. Este tipo de ataque debe quedar registrado en los logs del sistema.

- Utilizar autentificación multi-factor para cuentas más sensibles o de mayor valor.
- Se deben implementar controles para la administración de sesiones.
- Los controles de administración de sesiones deben utilizar algoritmos que generen identificadores suficientemente aleatorios.
- La función de logout debe terminar completamente con la sesión o conexión asociada y debe estar disponible en todas las páginas protegidas por autentificación.
- Se debe establecer un tiempo de vida de la sesión lo más corto posible. En la mayoría de los casos, nunca debería ser superior a varias horas.
- Generar un autentificador de sesión luego de cada re autentificación.
- No permitir inicio de sesiones concurrentes con el mismo usuario.
- No exponer identificadores de sesión en URLs, mensajes de error no logs.
- Generar un nuevo de identificador de sesión y desactivar el anterior de forma periódica. De esta forma se puede mitigar algunos escenarios de ribo de sesiones donde el identificador se ve comprometido.

Contraseñas para software de desarrollo
Las contraseñas de desarrollos de sistemas desarrollados por el GORE Los Lagos deben regirse por el documento "Estándar de seguridad en cuentas de usuario".

En las etapas de desarrollo, testing o pruebas y aprobación del sistema

Requisitos de seguridad:
Se debe identificar los objetivos y requisitos de seguridad que se requieren contemplar e implementar, estos se pueden determinar basado en lo siguiente:

- Se debe contar con ambientes de producción, testing o pruebas y desarrollo de forma
- independiente.
- Los desarrolladores deben realizar su trabajo exclusivamente en ambiente de desarrollo.
- Los nombres de dominio para los ambientes de producción, testing y desarrollo deben ser diferentes para así evitar confusión durante la ejecución de las pruebas, puesta en producción y desarrollo.

En el desarrollo

- Autentificarse adecuadamente: La información confidencial y los sistemas informáticos solo deben ser accesibles por las personas con los roles definidos en la etapa de diseño.
- Comprobar las entradas: Se debe verificar y controlar los datos que son ingresados en los aplicativos, estos deben estar dentro del rango de valores definidos, es decir si son numéricos que lo sean y que no sobrepasen la longitud determinada, sobre todo en los que son de tipo cadenas.
- Valores límites de salida: Se debe controlar la salida de los métodos, que el dato resultante se una operación este dentro de los parámetros definidos antes de asignarlo.
- Formato de salida: Los formatos de salida no deben ser cambiados por funciones debido a que estos pueden ocasionar errores en el buffer.
- Se debe proteger la información que se muestra sobre los procesos activos, debido a que muchos sistemas operativos permiten a un usuario observar la información de procesos que pertenecen a otros usuarios. Esta información puede incluir argumentos de línea de comandos o la configuración de la variable de entorno, lo que puede provocar un ataque contra el software por parte de otros usuarios si entre esos datos se incluye información confidencial.
- Se debe evitar el uso de datos reales de carácter personal en las pruebas anteriores a la implantación o modificación de un sistema, salvo que se garantice el nivel de seguridad correspondiente al tipo de información contratada.

Buenas prácticas en el desarrollo de software

Buenas prácticas en el desarrollo de software

- Emplear nombres descriptivos, en la declaración de variables.
- La aplicación deberá generar y almacenar un log de auditoria sobre las tablas y transacciones críticas, que permitan consultar como mínimo: Id del usuario, fecha, hora, dirección ip, tabla involucrada, acción ejecutada (modificación, creación, eliminación, etc.).
- Los comentarios que contenga el código fuente, deben ir enfocados a describir funcionalidades que se está programando, en los bloques de códigos extensos es recomendables dividirlos e introducir un comentario al principio con el fin de guiar al desarrollador, seria optimo que exista una línea blanca de separación, estos comentarios no deberían ser excesivos.
- No excederse con el número de niveles de instrucciones anidadas.
- No mezclar datos con código.
- Evitar el usar métodos con muchos parámetros, en caso que sea necesario es recomendable contemplar la creación de una clase que tenga las propiedades requeridas.
- Se debe considerar la validación de todos los parámetros de las API exportadas, verificando que sean válidos, esto incluye los datos que parece ser coherentes pero que están más allá de intervalo de valores aceptados, como los tamaños de buffer excesivos.
- Se debe considerar emplear API criptográficas por firmas reconocidas, en lugar de escribir su propio software criptográfico, con ello los desarrolladores podrán concentrarse en la generación de aplicaciones.
- Cuando una funcionalidad se requiera implementar en diferentes aplicaciones se recomienda crear una función, una rutina, un servicio o un componente que sea reutilizable para cualquier aplicación.

En la etapa de testing o pruebas

En esta etapa se debe considerar lo siguiente:

- Las pruebas funcionales de acuerdo a las funcionalidades solicitadas deben buscar o descartar errores que se presenten al utilizar el aplicativo o el módulo desarrollado.
- En el caso que el requerimiento venga de un mantenimiento se deben verificar todas las partes del sistema que se puedan ver afectadas por la integración de la nueva funcionalidad o cambio.
- Se debe probar el nivel de acceso al aplicativo, con el fin de valorar que únicamente accedan los usuarios autorizados y solo los módulos definidos de acuerdo a su rol.
- Las pruebas de seguridad funcional se deben basar en los requerimientos definidos con respecto a:
    - Autentificación solicitada.
    - Bloqueo automático de cuentas de acceso.
    - Lo registrado en los logs, así como en su almacenamiento.
    - Los mensajes de error que se deben presentar en las acciones validadas.
    - En la preparación de datos de pruebas, estos preferiblemente no deben ser reales; sin embargo, teniendo en cuenta que para algunas pruebas pueden ser complejas la preparación manual de datos y que puede ser ineficiente debido a la integridad referencial que deben mantener las relaciones de los registros y llegase a ser necesario emplear información real para las pruebas, esta información debe ser anonimizada, con el fin de resguardar la confidencialidad de la información solicitada.
    - Es necesario que se contemplen la posibilidad de realizar pruebas de stress de software, creando escenarios que la sometan la aplicación a su máximo rendimiento y consumo de recursos.
    - Las pruebas que se realicen a los aplicativos se debe designar una persona diferente al desarrollador del requerimiento.

Lista de chequeo
Se debe realizar una verificación para verificar el cumplimiento de Desarrollo de Software Seguro.
De ser necesario según la particularidad de cada proyecto se pueden adicionar o modificar los chequeos.

En las etapas de puesta en marcha, capacitación.
Requisitos de seguridad:
La capacitación o transferencia de conocimientos en la puesta en producción se debe contemplar las políticas de seguridad de la información, la importancia del buen uso del aplicativo, así como la confidencialidad que maneja el aplicativo, como las restricciones, roles y perfiles, manejos de usuarios y contraseñas con los que va a contar el aplicativo.

Lista de chequeo
Se debe realizar una verificación para verificar el cumplimiento de Desarrollo de Software Seguro.
De ser necesario según la particularidad de cada proyecto se pueden adicionar o modificar los chequeos.

4.1. Documentos de referencia

En la siguiente tabla, se presentan los documentos que se han utilizado como referencia, para la formulación del presente manual de procedimientos.

Código	Descripción
SSI-A.05.01.01	Política de Seguridad de la Información
NCh-ISO 27001	Tecnologías de la información – Técnicas de seguridad – Sistemas de gestión de la seguridad de la información.
NCh-ISO 27002	Tecnologías de la información – Técnicas de seguridad – Código de prácticas para los controles de seguridad de la información.

4.2. Registros de control del procedimiento.

a) Lista de chequeo etapas: requerimientos, análisis y diseño.
b) Lista de chequeo etapas: desarrollo, testing o pruebas y aprobación de sistemas.
c) Lista de chequeo etapas: puesta en marcha y capacitación.

5. Validez y gestión de documentos

Este documento es válido desde la fecha de su aprobación.

El responsable de este documento es el Encargado de la Unidad de Informática, que debe verificar, y si es necesario actualizar el documento, por lo menos una vez cada tres años.

Aprobado Por
GOBIERNO REGIONAL - ENCARGADO DE SEGURIDAD DE LA INFORMACIÓN - Región de Los Lagos GOBIERNO REGIONAL - COMITÉ DE SEGURIDAD DE LA INFORMACIÓN - Región de Los Lagos
Oscar Alejandro Oyarzo Pérez **Encargado de Seguridad de la Información**
05 de Noviembre de 2019