	Subdirección Ingeniería, IDC & Ofimática Seguridad de la Información	Versión 1.
	Nombre del archivo: Documento Recomendaciones de Seguridad para Portal WEB – APLICACIÓN SARLAFT	Fecha de Elaboración:
	Elaboró : Martha Lucia Duque Carvajal	20/05/2015

## DOCUMENTO RECOMENDACIONES DE SEGURIDAD DE LA INFORMACIÓN

### PORTAL WEB – APLICACIÓN SARLAFT

#### 1. INTRODUCCION

Con el ánimo de implementar un adecuado nivel de Seguridad para el Sistema de Información de SARLAFT que se está planeando, y con el fin de reducir el riesgo de interrupción del sistema debido a controles de seguridad técnicamente inadecuados, a continuación por parte de Seguridad de la Información, se describen los aspectos mínimos que se deben tener presente durante el desarrollo y puesta en marcha de la solución.

Este documento está dividido en tres secciones con base en la información suministrada en el documento de factibilidad y en los lenguajes y motores de base de datos que se utilizarán.

#### 2. LINEAMIENTOS BASE DE SEGURIDAD

##### 2.1 Requisitos de Seguridad para Framework de Desarrollo Yii


###### 2.1.1 Items a considerar

###### ➤ Administración

Los mecanismos de gestión de un sistema, solo deben ser accesibles desde segmentos de red de gestión o administrativos, para lo cual se considera conveniente restringir el acceso desde el parámetro “actionLogin” del controlador. Para dar cumplimiento con esta recomendación, se deben agregar las IP permitidas a los parámetros generales de la aplicación, editando el archivo **`./protected/config/main.php`**.

###### ➤ Archivos

- Los archivos temporales no deben desplegarse al ambiente de producción, para lo cual se debe tener presente lo siguiente:
  - Eliminar los archivos de pruebas unitarias, que se encuentran en la carpeta **`./protected/tests/`**.
  - Eliminar archivos innecesarios como **`./protected/yiic.bat`**, **`./protected/yiic`**.
  - Eliminar cualquier vista, modelo o controlador de prueba.
- Restringir el tamaño máximo de los archivos subidos.

	Subdirección Ingeniería, IDC & Ofimática Seguridad de la Información	Versión 1.
	Nombre del archivo: Documento Recomendaciones de Seguridad para Portal WEB – APLICACIÓN SARLAFT	Fecha de Elaboración:
	Elaboró : Martha Lucia Duque Carvajal	20/05/2015

Para validar el tamaño máximo que puede tener un archivo a la hora de ser subido usando un formulario en **Yii** se utiliza el validador **file** con el atributo **maxSize**

- Validar que archivos subidos al servidor estén libre de código malicioso.

Para aplicar esta recomendación, se debe descargar la extensión **clamscanvalidator** desde <http://www.yiiframework.com/extension/clamscanvalidator/> y se descomprime en la carpeta **/protected/extensions/** de la aplicación.

A través de esta validación, se consultará el demonio del antivirus ClamAV y si el archivo subido es malicioso, se mostrará un mensaje de error.

## ➤ Autenticación

- El proceso de autenticación no debe emitir mensajes de error que permitan distinguir si es un error de usuario o de contraseña.

Durante el proceso de autenticación, no se debe diferenciar entre un error de usuario o un error de contraseña, dado que un usuario malintencionado podría obtener una lista de usuarios válidos y realizar un ataque de fuerza bruta más efectivo.

En caso de un error en la validación de las credenciales durante la fase de autenticación, se debe mostrar un error genérico.

- Debe validarse que el sujeto que realiza las acciones de registro, autenticación y re-establecimiento de contraseña es un ser humano, para lo cual se sugiere utilizar un “reCaptcha” en los formularios sensibles de la aplicación


Para aplicar esta sugerencia, se debe descargar la extensión **reCaptcha** para Yii framework desde <http://www.yiiframework.com/extension/recaptcha/>. La instalación se realiza copiando la carpeta en la ruta **/protected/extensions/** y se modifica el Modelo del formulario y se adiciona el código respectivo a los métodos “**rules** y **attributeLabels**” del Modelo.

## ➤ Autorización

- Los privilegios para objetos nuevos deben establecerse según el principio de mínimo privilegio (umask),
- Los privilegios para un sujeto no pueden ser aumentados por el mismo sujeto, por lo tanto se requieren definir roles de usuarios.
- Los privilegios para objetos con contenido o funcionalidad sensible deben tener acceso restringido.

Para poder implementar las tres recomendaciones anteriores, se puede hacer uso del mecanismo de control de acceso basado en roles denominado RBAC.

Con base en la granularidad que se implemente en el sistema y los roles de los funcionarios que van a ingresar, los elementos de autorización pueden ser clasificados como operaciones, tareas y roles.

	Subdirección Ingeniería, IDC & Ofimática Seguridad de la Información	Versión 1.
	Nombre del archivo: Documento Recomendaciones de Seguridad para Portal WEB – APLICACIÓN SARLAFT	Fecha de Elaboración:
	Elaboró : Martha Lucia Duque Carvajal	20/05/2015

Para definir un elemento de autorización, se puede utilizar uno de los siguientes métodos, dependiendo del tipo de elemento:

- CAuthManager::createRole
- CAuthManager::createTask
- CAuthManager::createOperation

## ➤ Bitácoras

- Los eventos excepcionales deben ser registrados en bitácoras y ser clasificados por severidad.

Para poder efectuar el registro en bitácoras, se puede hacer uso del método estático **Yii::log**, el cual recibe tres parámetros, el mensaje a registrar y el nivel.

Es de anotar, que este método registra los mensajes en memoria, por lo tanto para almacenarlos en un archivo físico, se debe utilizar el enrutador **CFileLogRoute.php**, para lo cual se debe modificar el archivo **./protected/config/main.php**,

- Los eventos con severidad de depuración no deben estar habilitados en producción.

Para deshabilitar los mensajes de depuración de la aplicación en el archivo **index.php** se configura en "**False**" la línea de código que tiene el parámetro '**Yii DEBUG**'.

## ➤ Codificación


- Debe validarse que la salida de información esté codificada según el lenguaje correspondiente (HTML, JS, .escaping.) evitando la presencia de XSS.

XSS, del inglés **Cross-site scripting** es un tipo de vulnerabilidad típico de aplicaciones web, en el que un usuario malintencionado puede inyectar en páginas vistas por el usuario código **JavaScript**, lo cual puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, comprometiendo la integridad del sistema. Con el ánimo de eliminar la presencia de XSS se recomienda usar siempre el método estático **CHtml::encode (Nombre del campo)**. Y si se llegara a permitir el uso de etiquetas HTML se hace uso de la clase **CPurifier**

## ➤ Datos

- La información sensible debe ser transportada por un canal seguro, para lo cual se recomienda hacer uso de un protocolo seguro y de certificado digital.

## ➤ Validación

	Subdirección Ingeniería, IDC & Ofimática Seguridad de la Información	Versión 1.  Fecha de Elaboración: 20/05/2015
	Nombre del archivo: Documento Recomendaciones de Seguridad para Portal WEB – APLICACIÓN SARLAFT	
	Elaboró : Martha Lucia Duque Carvajal	

Debe validarse la entrada de información antes de ser usada, de tal forma que se minimice el ingreso de código malicioso y de ciertos vectores de ataque, para lo cual se puede hacer uso del método rules() en los modelos definidos para el uso de la aplicación

- Las peticiones a acciones en una aplicación no deben seguir un patrón discernible, se debe prevenir los ataques Cross Site Request Forgery en una aplicación desarrollada con Yii.

Un ataque Cross-Site Request Forgery (CSRF) ocurre cuando un sitio web malicioso obliga al navegador del usuario a realizar una acción no deseada en un sitio confiable. Por lo anterior, Yii implementa un sistema de protección CSRF para ayudar a prevenir los ataques mediante el método POST, el cuál almacena un valor aleatorio en la cookie, y luego se compara el valor de la cookie con el que es enviado mediante el formulario en la petición POST. Es de anotar, que por defecto, este sistema de protección esta deshabilitado , por lo tanto para habilitarlo se debe configurar el componente de aplicación **CHttpRequest**. Y para crear un formulario se recomienda utilizar **CHtml::form** en vez de escribir directamente las etiquetas HTML