

en Español



# Software Assurance Maturity Model

Una guía para integrar seguridad en el desarrollo de software

VERSIÓN - 1.0

**Para obtener mayor información, por favor vea el sitio del proyecto en:**

<http://www.opensamm.org>

### **RECONOCIMIENTOS**

El modelo de madurez para el aseguramiento del software (SAMM por sus siglas en inglés) fue diseñado, desarrollado y escrito originalmente por Pravir Chandra ([chandra@owasp.org](mailto:chandra@owasp.org)), un consultor de seguridad independiente. La creación del primer borrador fue posible a través de los fondos de Fortify Software, Inc. Este documento es mantenido y actualizado actualmente por el proyecto OpenSAMM liderado por Pravir Chandra. Desde la publicación inicial, este proyecto se ha convertido en parte del proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés). Muchas gracias también a las muchas organizaciones que nos apoyaron (listadas en la contra-cubierta).

### **CONTRIBUIDORES Y REVISORES**

Este trabajo no podría haber sido posible sin el apoyo de muchos revisores y expertos que ofrecieron sus contribuciones y retroalimentación crítica. Ellos son (en orden alfabético):

Fabio Arciniegas	Brian Chess	Matteo Meucci	John Steven
Matt Bartoldus	Dinis Cruz	Jeff Payne	Chad Thunberg
Sebastian Deleersnyder	Justin Derry	Gunnar Peterson	Colin Watson
Jonathan Carter	Bart De Win	Jeff Piper	Jeff Williams
Darren Challey	James McGovern	Andy Steingruebl	

### **TRADUCTORES**

Francisco Aldrete	Miguel Pérez-Milicua	Aldo Salas
Luis Martínez Bacha	Alvaro Muñoz	

### **Edición de la Traducción**

Juan Carlos Calderón Rojas

Este es un proyecto de OWASP.



# OWASP

The Open Web Application Security Project

El proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad libre y abierta enfocada en mejorar la seguridad de los programas aplicativos. Nuestra misión es hacer la seguridad en aplicaciones “visible”, de manera que las personas y organizaciones puedan tomar decisiones informadas sobre los riesgos de seguridad en aplicaciones. Todos pueden participar en OWASP y todos nuestros materiales están disponibles bajo una licencia de software libre y abierto. La fundación OWASP es una organización caritativa sin ánimo de lucro 501(c)3 que asegura viabilidad continua y el apoyo a nuestro trabajo. Visite el sitio de OWASP en línea en <http://www.owasp.org>.

### **LICENCIA**



Este trabajo se publica bajo la licencia Creative Commons Attribution-Share Alike 3.0. Para ver una copia de la licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

# Resumen Ejecutivo

El modelo de madurez para el aseguramiento de software (SAMM por sus siglas en inglés) es un marco de trabajo abierto para ayudar a las organizaciones a formular e implementar una estrategia de seguridad para Software que sea adecuada a las necesidades específicas que está enfrentado la organización. Los recursos proveídos por el SAMM ayudarán a:

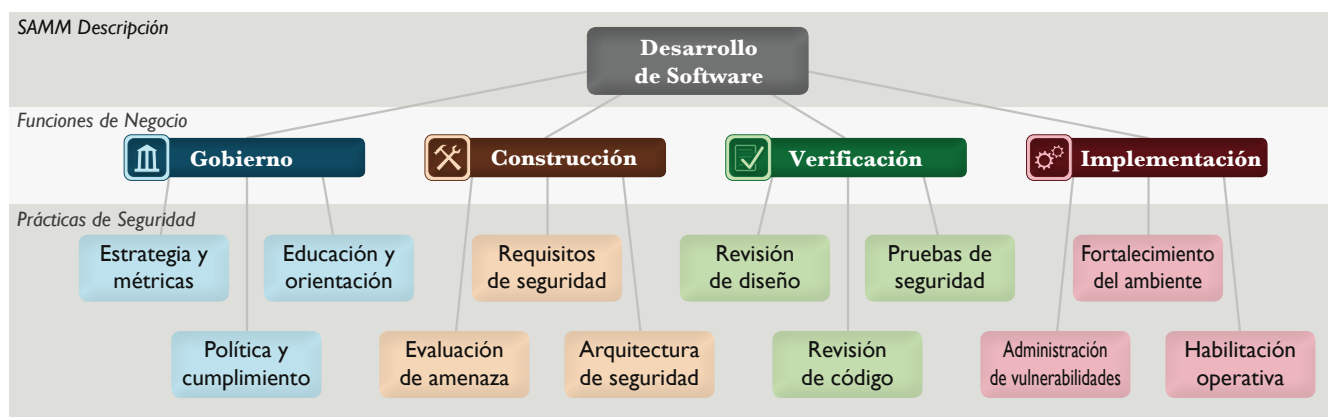
- ◆ *Evaluar las prácticas de seguridad en Software existentes en la organización*
- ◆ *Construir un programa de seguridad en Software balanceado en iteraciones bien definidas*
- ◆ *Demostrar mejoras concretas en el programa de aseguramiento de Software*
- ◆ *Definir y medir las actividades relacionadas con seguridad en la organización*

SAMM fue definido para ser flexible de manera que pueda ser utilizado por organizaciones pequeñas, medianas o grandes que utilicen cualquier estilo de desarrollo. Además, este modelo puede ser aplicado en toda la organización, en una sola línea de negocio o incluso en un proyecto en particular. Además de estos elementos, SAMM fue construido sobre los siguientes principios:

- ◆ *Cambios de comportamiento de una organización a través del tiempo.* Un programa de seguridad para Software exitoso debería ser creado en pequeños ciclos que entreguen ganancias tangibles en el aseguramiento de Software, al mismo tiempo, debe trabajar incrementalmente hacia metas de largo plazo.
- ◆ *No hay una sola receta que funcione para todas las organizaciones.* Un marco de seguridad en Software debe ser flexible y permitir a las organizaciones personalizar sus opciones basándose en su tolerancia a riesgo y la manera en la cual construye y usa el Software.
- ◆ *Los lineamientos relacionados a las actividades de seguridad deben ser específicos.* Todos los pasos en la construcción y medición del programa de aseguramiento deben ser simples, bien definidos y medibles. Este modelo también provee plantillas de planes de implementación para tipos comunes de organizaciones.

Las bases de este modelo están construidas alrededor de las funciones de negocio relacionadas al desarrollo de Software, se incluyen una serie de prácticas relacionadas a cada función (vea el diagrama abajo). Los bloques de construcción del modelo son los tres niveles de madurez definidos para cada una de las doce prácticas de seguridad. Estas definen una amplia variedad de actividades a las que una organización se puede adherir para reducir los riesgos de seguridad e incrementar el aseguramiento del Software. Se incluyen detalles adicionales para medir el desempeño exitoso de las actividades, entender los beneficios del aseguramiento asociado, estimar los costos de personal y otros costos.

Dado que SAMM es un proyecto abierto, el contenido se puede mantener siempre independiente de los vendedores y disponible libremente para que todo mundo lo use.



# Contenidos

<b>Resumen Ejecutivo</b> .....	<b>3</b>
<b>ENTENDIENDO EL MODELO</b> .....	<b>6</b>
<b>Funciones de Negocio</b> .....	<b>8</b>
Gobierno .....	10
Construcción .....	12
Verificación .....	14
Implementación .....	16
<b>APLICANDO EL MODELO</b> .....	<b>18</b>
<b>Usando los Niveles de Madurez</b> .....	<b>20</b>
<b>Realizando Revisiones</b> .....	<b>21</b>
<b>Creando Tarjetas de Calificaciones</b> .....	<b>26</b>
<b>Construyendo Programas</b> .....	<b>27</b>
Proveedor Independiente de Software .....	28
Proveedor de Servicios en Línea .....	29
Organización de Servicios Financieros .....	30
Organización de Gobierno .....	31
<b>LAS PRÁCTICAS DE SEGURIDAD</b> .....	<b>32</b>
Estrategia y métricas .....	34
Política y cumplimiento .....	38
Educación y orientación .....	42
Evaluación de amenaza .....	46
Requisitos de seguridad .....	50
Arquitectura de seguridad .....	54
Revisión de diseño .....	58
Revisión de código .....	62
Pruebas de seguridad .....	66
Administración de vulnerabilidades .....	70
Fortalecimiento del ambiente .....	74
Habilitación operativa .....	78
<b>CASOS DE ESTUDIO</b> .....	<b>82</b>
<b>VirtualWare</b> .....	<b>84</b>

## Si DESEA ...

### Revisar las prácticas existentes sobre aseguramiento de software

3 † Resumen Ejecutivo  
8-9 † Funciones de Negocio  
10-11 † Gobierno  
12-13 † Construcción  
14-15 † Verificación  
16-17 † Implementación  
21-25 † Realizando Revisiones  
26 † Creando Tarjetas de Calificaciones  
20 † Usando los Niveles de Madurez  
34-37 † Estrategia y métricas  
38-41 † Política y cumplimiento  
42-45 † Educación y orientación  
46-49 † Evaluación de amenaza  
50-53 † Requisitos de seguridad  
54-57 † Arquitectura de seguridad  
58-61 † Revisión de diseño  
62-65 † Revisión de código  
66-69 † Pruebas de seguridad  
70-73 † Administración de vulnerabilidades  
74-77 † Fortalecimiento del ambiente  
78-81 † Habilitación operativa  
27-31 † Construyendo Programas  
84-95 † VirtualWare

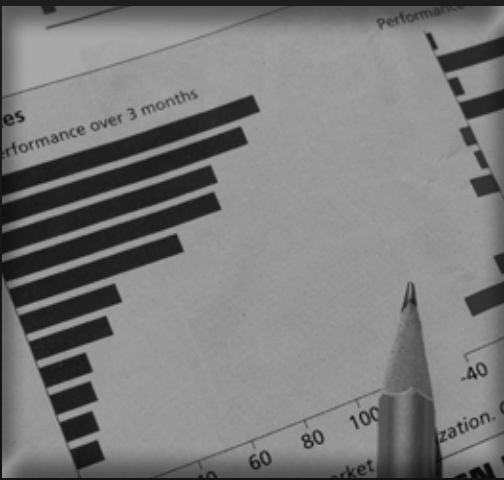
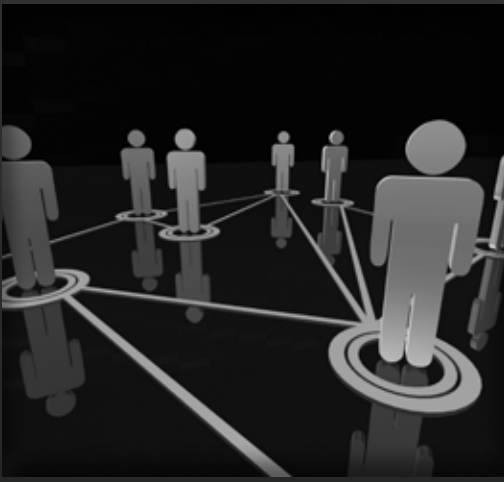
- † Leer
- ◇ Esquema

### Construir un plan estratégico para su organización

3 † Resumen Ejecutivo  
8-9 † Funciones de Negocio  
10-11 † Gobierno  
12-13 † Construcción  
14-15 † Verificación  
16-17 † Implementación  
20 † Usando los Niveles de Madurez  
27-31 † Construyendo Programas  
21-25 † Realizando Revisiones  
26 † Creando Tarjetas de Calificaciones  
84-95 † VirtualWare  
34-37 † Estrategia y métricas  
38-41 † Política y cumplimiento  
42-45 † Educación y orientación  
46-49 † Evaluación de amenaza  
50-53 † Requisitos de seguridad  
54-57 † Arquitectura de seguridad  
58-61 † Revisión de diseño  
62-65 † Revisión de código  
66-69 † Pruebas de seguridad  
70-73 † Administración de vulnerabilidades  
74-77 † Fortalecimiento del ambiente  
78-81 † Habilitación operativa

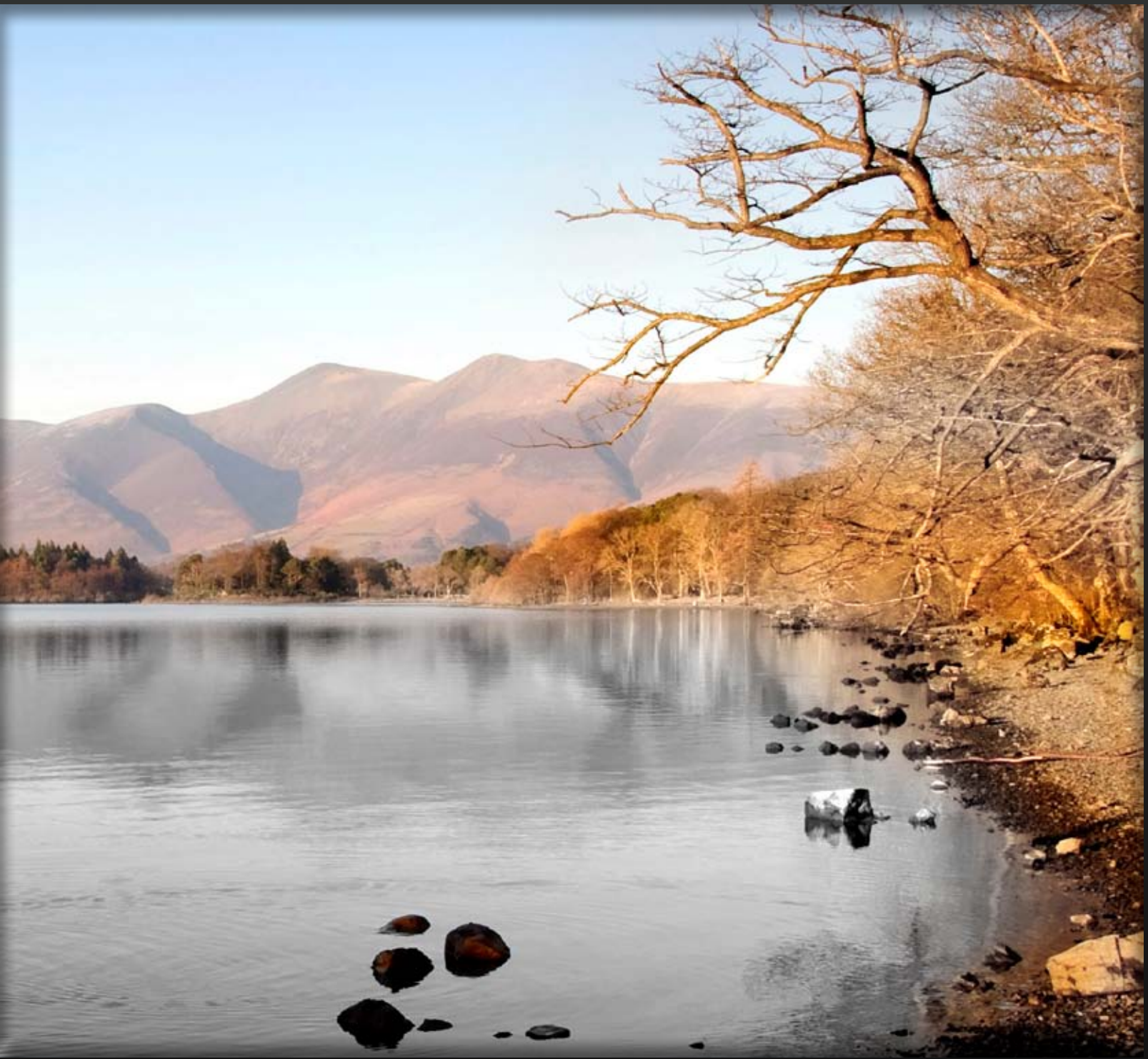
### Implementar o realizar actividades de seguridad

3 † Resumen Ejecutivo  
8-9 † Funciones de Negocio  
10-11 † Gobierno  
12-13 † Construcción  
14-15 † Verificación  
16-17 † Implementación  
20 † Usando los Niveles de Madurez  
34-37 † Estrategia y métricas  
38-41 † Política y cumplimiento  
42-45 † Educación y orientación  
46-49 † Evaluación de amenaza  
50-53 † Requisitos de seguridad  
54-57 † Arquitectura de seguridad  
58-61 † Revisión de diseño  
62-65 † Revisión de código  
66-69 † Pruebas de seguridad  
70-73 † Administración de vulnerabilidades  
74-77 † Fortalecimiento del ambiente  
78-81 † Habilitación operativa  
21-25 † Realizando Revisiones  
26 † Creando Tarjetas de Calificaciones  
27-31 † Construyendo Programas  
84-95 † VirtualWare



# Entendiendo el Modelo

Un vistazo a el horizonte



SAMM se construyó sobre una serie de prácticas de seguridad que están ligadas a las funciones de negocio centrales que están involucradas en el desarrollo de Software. Esta sección presenta estas funciones con sus correspondientes prácticas de seguridad. Después de cubrir el marco de trabajo a alto nivel, se discuten brevemente todas las prácticas de seguridad de cada nivel de madurez para darse una idea de cómo cada una puede mejorar iterativamente a través del tiempo.

# Funciones de Negocio

**Al nivel más alto, SAMM define cuatro funciones de negocio importantes.** Cada función de negocio (listada abajo) es una categoría de actividades relacionadas a las tareas específicas del desarrollo de software, dicho de otra manera, cualquier organización involucrada en el desarrollo de Software debe cumplir con cada una de esas funciones en cierto grado.

**Para cada función de negocio, SAMM define tres prácticas de seguridad.** Cada práctica de seguridad (listada al opuesto) es un área de actividades de seguridad que construyen las actividades de aseguramiento para las funciones de negocio relacionadas. Así que, en términos generales hay doce prácticas de seguridad que son las áreas de oportunidad a mejorar y comparar contra la funciones de desarrollo de Software del negocio.

**Para cada práctica de seguridad, SAMM define tres niveles de madurez como objetivos.** Cada nivel en las prácticas de seguridad esta caracterizado por un objetivo sucesivamente más sofisticado, definido por actividades específicas y mayores y mas exigente métricas de éxito que en el nivel anterior. Así mismo, cada práctica de seguridad puede ser mejorada independientemente, a través de actividades relacionadas que lleven a optimizaciones.



## Gobierno

El gobierno de TI está enfocado en los procesos y actividades relacionadas a como una organización gestiona las actividades de desarrollo de software global. Más específicamente, esto incluye preocupaciones que atraviesan los grupos implicados en el desarrollo, así como procesos de negocio que son establecidos a nivel de organización.

...continúa en página 10



## Construcción

Construcción se refiere a los procesos y actividades relacionados a como una organización define metas y crea software dentro de proyectos de desarrollo. En general, esto incluir la gestión de producto, reunión de requisitos de seguridad, especificación de arquitectura de alto nivel, diseño detallado e implementación.

...continúa en página 12



## Verificación

La verificación está enfocada en los procesos y actividades relacionadas a como una organización verifica y prueba artefactos producidos a través del desarrollo de Software. Esto típicamente incluye un trabajo de aseguramiento de calidad como lo son las pruebas, pero esto puede también incluir otras revisiones y actividades de evaluación.

...continúa en página 14



## Implementación

La implementación abarca los procesos y actividades relacionadas con la forma en que una organización administra la liberación de sistemas que han sido creados. Esto puede incluir el envío de productos a los usuarios finales, la instalación de los productos en ambientes internos o externos, y las operaciones normales de los sistemas en un ambiente de ejecución.

...continúa en página 16



Gobierno

**Estrategia y métricas** involucra la dirección estratégica global del programa de aseguramiento de software e instrumentación de procesos y actividades para recolectar métricas acerca de la postura de seguridad de una organización.

**Política y cumplimiento** involucra establecer una estructura de control y auditoría para seguridad y cumplimiento de regulaciones a lo largo de una organización para alcanzar un aseguramiento superior en software bajo construcción y en operación.

**Educación y orientación** involucra incrementar el conocimiento de seguridad entre el personal de desarrollo de software a través de entrenamiento y orientación en temas de seguridad pertinentes a funciones del trabajo individual.

Construcción

**Evaluación de amenazas** involucra identificar y caracterizar con precisión los ataques potenciales contra el software de una organización, con el fin de comprender mejor los riesgos y facilitar su gestión.

**Requisitos de seguridad** involucra promover la inclusión de las necesidades de seguridad durante el proceso de desarrollo de software a fin de especificar la funcionalidad correcta desde el principio.

**Arquitectura de seguridad** implica el fortalecimiento del proceso de diseño con actividades para promover diseños con seguridad en mente y los marcos de trabajo en que se basa el software.

Verificación

**Revisión de diseño** involucra la inspección de artefactos creados a partir del proceso de diseño para asegurar la provisión de mecanismos de seguridad adecuados y apegados a las expectativas de seguridad de la organización

**Revisión de código** involucra la evaluación del código fuente de una organización para ayudar en el descubrimiento de vulnerabilidades y actividades relacionadas a la mitigación como es el establecimiento de bases para las expectativas de la seguridad en programación.

**Pruebas de seguridad** involucra probar el software de la organización en su ambiente de ejecución para descubrir vulnerabilidades y establecer un estándar mínimo para la liberación del software.

Implementación

**Administración de vulnerabilidades** involucra establecer procesos consistentes para administrar reportes internos o externos de vulnerabilidades para limitar la exposición, recopilar datos y así mejorar el programa de aseguramiento.

**Fortalecimiento de ambientes** implica la implementación de controles para el ambiente operativo que rodea a los programas de una organización para reforzar la postura de seguridad de las aplicaciones que han sido implementadas.

**Habilitación operativa** implica identificar y capturar información relevante a la seguridad que necesita un operador para configurar, instalar y correr los programas de una organización.

### Niveles de Madurez

Cada una de las prácticas de seguridad tiene tres niveles de madurez bien definidos y un nivel inicial (cero) implícito. Los detalles de cada nivel difieren entre las prácticas pero generalmente representan:

- 0** Punto de inicio implícito, las actividades en la practica no se han realizado
- 1** Entendimiento inicial y provisión ad hoc de la práctica de seguridad
- 2** Incremento en la eficiencia y/o efectividad de la práctica de seguridad
- 3** Dominio amplio de la práctica de seguridad

### Notación

A través de este documento, los siguientes términos en mayúsculas se utilizarán como palabras reservadas que se refieren a los componentes del SAMM definidos en esta sesión. Si los términos aparecen sin mayúsculas, deben ser interpretados de acuerdo a el contexto en el que se encuentren:

- ◆ Función de Negocio también nombrado Función
- ◆ Práctica de Seguridad también nombrada Práctica
- ◆ Nivel de Madurez también nombrado como Nivel u Objetivo

# Gobierno

## Descripción de prácticas de seguridad



### Estrategia y métricas

La práctica de estrategia y métricas (SM por sus siglas en Inglés) está enfocada en establecer la estructura dentro de una organización para un programa de aseguramiento de software. Este es el paso más fundamental en la definición de objetivos de seguridad de una forma que sea medible y alineada con los riesgos de negocio reales de la organización. Al iniciar con perfiles de riesgo sencillos, una organización aumenta con el tiempo sus esquemas de clasificación de riesgos para aplicación y activos de datos. Con una perspectiva adicional sobre las medidas de riesgo relativo, una organización puede ajustar sus objetivos de seguridad a nivel de proyecto y elaborar planes de implementación granulares para que el programa de seguridad sea más eficiente. En los niveles más avanzados en esta práctica, una organización se basa en muchas fuentes de datos, tanto internos como externos, para recolectar métricas y retroalimentación cualitativa acerca del programa de seguridad. Esto permite un ajuste fino de la relación costo-beneficio a nivel del programa.



### Política y cumplimiento

La práctica de Política y Cumplimiento (PC por sus siglas en Inglés) está enfocada en comprender y cumplir con requisitos externos legales y regulatorios, además de implementar estándares de seguridad internos para asegurar el cumplimiento regulatorio de una manera que está alineada con los objetivos de negocio de la organización. Un tema importante a mejorar dentro de esta práctica es enfocarse en auditorías a nivel proyecto que reúnan información acerca del comportamiento de la organización para comprobar que las expectativas se están cumpliendo. Al introducir auditorías de rutina que comiencen sencillamente y crezcan en profundidad con el tiempo, el cambio organizacional es alcanzado de forma iterativa. De una forma sofisticada, la prestación de esta práctica implica entendimiento organizacional de estándares internos y cumplimientos externos y al mismo tiempo mantiene las aprobaciones de baja latencia con equipos de proyecto para asegurar que ningún proyecto esté operando fuera de las expectativas y sin visibilidad.



### Educación y orientación

La práctica de educación y orientación (EG por sus siglas en Inglés) está enfocada en el personal involucrado en el ciclo de vida de software con conocimiento y recursos para diseñar, desarrollar e implementar software seguro. Con acceso mejorado a la información, los equipos de proyecto estarán en mejores condiciones de para identificar proactivamente y mitigar los riesgos de seguridad específicos que apliquen para su organización. Un tema importante para la mejora a través de los objetivos es proporcionar entrenamiento para los empleados, ya sea con sesiones basadas en instructores o módulos basados en computadora. Conforme una organización progresa, una gran cantidad de entrenamiento es construido al empezar con los desarrolladores y moverse a otros roles en la organización, culminando con la adición de certificación basada en roles para asegurar la comprensión del material. Además del entrenamiento, esta práctica también requiere convertir información relevante a seguridad en lineamientos que sirvan como información de referencia para el personal. Esto construye un cimiento para establecer una expectativa base para las prácticas de seguridad en la organización, y después permite la mejora incremental una vez que el uso de los lineamientos ha sido adoptado.

# Gobierno

## Resumen de actividades

### Estrategia y métricas

...continúa en página 34



<b>OBJETIVOS</b>	<b>Establecer un plan estratégico unificado para la seguridad del software dentro de la organización</b>	<b>Medir el valor relativo de los datos y bienes, y elegir la tolerancia al riesgo</b>	<b>Alinear los gastos de seguridad con indicadores de negocio pertinentes y el valor de los activos</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Estimar el perfil global de riesgo del negocio</li> <li>B. Crear y mantener un plan de implementación para el programa de aseguramiento</li> </ul>	<ul style="list-style-type: none"> <li>A. Clasificar datos y aplicaciones basado en riesgo de negocio</li> <li>B. Establecer y medir los objetivos de seguridad por cada clasificación</li> </ul>	<ul style="list-style-type: none"> <li>A. Realizar comparaciones de costo periódicas a nivel industria</li> <li>B. Recolectar métricas históricas de gastos de seguridad</li> </ul>

### Política y cumplimiento

...continúa en página 38



<b>OBJETIVOS</b>	<b>Entender los motivos relevantes para el gobierno de TI y cumplimiento de regulaciones para la organización</b>	<b>Establecer base de seguridad y cumplimiento, y entender los riesgos por proyecto</b>	<b>Exigir cumplimiento de regulaciones y medir a los proyectos conforme a las políticas y estándares de la organización</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Identificar y monitorear los indicadores externos de cumplimiento</li> <li>B. Crear y mantener lineamientos de cumplimiento</li> </ul>	<ul style="list-style-type: none"> <li>A. Crear políticas y estándares para seguridad y cumplimiento</li> <li>B. Establecer la práctica de auditoría de proyecto</li> </ul>	<ul style="list-style-type: none"> <li>A. Crear puntos de control de cumplimiento para proyectos</li> <li>B. Adoptar una solución para la recolección de datos de auditoría</li> </ul>

### Educación y orientación

...continúa en página 42



<b>OBJETIVOS</b>	<b>Ofrecer acceso al personal de desarrollo a recursos alrededor de los temas de programación segura e implementación</b>	<b>Educar a todo el personal en el ciclo de vida de software con lineamientos específicos en desarrollo seguro para cada rol</b>	<b>Hacer obligatorio el entrenamiento de seguridad integral y certificar al personal contra la base de conocimiento.</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Realizar entrenamiento técnico de concientización en seguridad</li> <li>B. Crear y mantener lineamientos técnicos</li> </ul>	<ul style="list-style-type: none"> <li>A. Realizar entrenamiento de seguridad en aplicaciones específico para cada rol</li> <li>B. Utilizar mentores de seguridad para mejorar los equipos</li> </ul>	<ul style="list-style-type: none"> <li>A. Crear un portal formal de soporte de seguridad en aplicaciones</li> <li>B. Establecer exámenes o certificaciones por rol</li> </ul>

# Construcción

## Descripción de prácticas de seguridad



### Evaluación de amenaza

La Práctica de Evaluación de Amenazas (TA por sus siglas en inglés) se centra en la identificación y comprensión de los riesgos a nivel de proyecto, basándose en la funcionalidad del software a desarrollar y las características del entorno de ejecución. Desde los detalles de cada amenaza y los probables ataques contra cada proyecto, la organización en su conjunto opera más eficazmente por medio de mejores decisiones en la priorización de las iniciativas para la seguridad. Además, las decisiones de aceptación de riesgo son más informadas, y por lo tanto, mejor alineadas con el negocio. Al comenzar con modelados simples de amenaza y moverse a la creación de métodos más detallados de análisis de las amenazas y ponderación, la organización mejora con el tiempo. En última instancia, una organización sofisticada mantendría esta información estrechamente unida a los factores de compensación y de paso a los riesgos de las entidades externas. Esto proporciona una mayor amplitud de comprensión para los potenciales impactos debido a problemas de seguridad, mientras mantiene una estrecha vigilancia sobre el desempeño actual de la organización contra las amenazas conocidas.



### Requisitos de seguridad

La Práctica de Requisitos de Seguridad (SR por sus siglas en inglés) se centra en especificar proactivamente el comportamiento esperado del software con respecto a la seguridad. A través de la adición de las actividades de análisis a nivel de proyecto, los requisitos de seguridad se reúnen inicialmente basándose en el objetivo comercial del software. Conforme avanza una organización, se utilizan técnicas más avanzadas como las especificaciones de control de acceso para descubrir nuevos requisitos de seguridad que pueden no haber sido evidentes inicialmente para el desarrollo. En una forma sofisticada, la prestación de esta Práctica implica meter los requisitos de seguridad de la organización dentro de sus relaciones con los proveedores y luego auditar los proyectos para asegurar que todos se adhieren a las expectativas, respecto a la especificación de los requisitos de seguridad.



### Arquitectura de seguridad

La Práctica de Arquitectura de Seguridad (SA por sus siglas en inglés) se centra en medidas proactivas para una organización para diseñar y construir software seguro por defecto. Al mejorar el proceso de diseño de software con servicios y componentes reutilizables, el riesgo de seguridad global de desarrollo de software puede ser dramáticamente reducido. A partir de simples recomendaciones sobre los marcos de software y la consideración explícita de los principios de diseño seguro, una organización que evoluciona hacia el uso consistente de patrones de diseño para la funcionalidad de seguridad. Además, las actividades animan a los equipos de proyecto a una mayor utilización de los servicios de seguridad centralizados y de infraestructura. Como una organización que evoluciona con el tiempo, el suministro sofisticado de esta Práctica implica organizaciones construyendo plataformas de referencia para cubrir los tipos genéricos de software que construyen. Estos sirven como marcos en los que los desarrolladores pueden crear software a medida con menor riesgo de vulnerabilidad.

# Construcción

## Resumen de actividades

### Evaluación de amenaza

...continúa en página 46



<b>OBJETIVOS</b>	<b>Identificar y comprender las amenazas de alto nivel para la organización y los proyectos individuales</b>	<b>Aumentar la precisión de la evaluación de amenazas y mejorar la granularidad de la comprensión por proyecto</b>	<b>Comparar concretamente controles de compensación a cada amenaza contra el software interno y de terceros</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Desarrollar y mantener modelos de amenaza específicos a cada aplicación</li> <li>B. Elabore perfil de atacante desde la arquitectura de software</li> </ul>	<ul style="list-style-type: none"> <li>A. Desarrollar y mantener modelos de casos de abuso por proyecto</li> <li>B. Adoptar un sistema de ponderación para la medición de las amenazas</li> </ul>	<ul style="list-style-type: none"> <li>A. Evaluar explícitamente el riesgo de los componentes de terceros</li> <li>B. Elaboración de modelos de amenaza con controles de compensación</li> </ul>

### Requisitos de seguridad

...continúa en página 50



<b>OBJETIVOS</b>	<b>Considerar explícitamente la seguridad durante el procesamiento de captura de requisitos de software</b>	<b>Aumentar la granularidad de los requisitos de seguridad derivados de la lógica de negocio y riesgos conocidos</b>	<b>Exigir que se siga el proceso de requisitos de seguridad para todos los proyectos de software y dependencias de terceros</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Deducir los requisitos de seguridad a partir de la funcionalidad de negocios</li> <li>B. Evaluar la seguridad y los lineamientos de cumplimiento para regulaciones de los requisitos</li> </ul>	<ul style="list-style-type: none"> <li>A. Generar una matriz de control de acceso a los recursos y capacidades</li> <li>B. Especificar los requisitos de seguridad en base a los riesgos conocidos</li> </ul>	<ul style="list-style-type: none"> <li>A. Incorporar los requisitos de seguridad a acuerdos con proveedores</li> <li>B. Ampliar el programa de auditoría para los requisitos de seguridad</li> </ul>

### Arquitectura de seguridad

...continúa en página 54



<b>OBJETIVOS</b>	<b>Insertar consideraciones para lineamientos proactivos de seguridad en el proceso de diseño de software</b>	<b>Dirija el proceso de diseño de software hacia servicios seguros conocidos y diseños seguros desde la concepción</b>	<b>Controlar formalmente el proceso de diseño de software y validar la utilización de componentes de seguridad</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Mantener una lista de los marcos de trabajo de software recomendados</li> <li>B. Aplicar explícitamente los principios de seguridad para el diseño</li> </ul>	<ul style="list-style-type: none"> <li>A. Identificar y promover los servicios de seguridad e infraestructura</li> <li>B. Identificar los patrones de diseño de seguridad desde la arquitectura</li> </ul>	<ul style="list-style-type: none"> <li>A. Establecer arquitecturas y plataformas formales de referencia</li> <li>B. Validar el uso de marcos de trabajo, patrones, y plataformas</li> </ul>

# Verificación

## Descripción de prácticas de seguridad



### Revisión de diseño

La Práctica de Revisión de Diseño (DR por sus siglas en inglés) está enfocada en evaluar el diseño de software y arquitectura en busca de problemas relacionados a la seguridad. Esto permite a una organización el detectar problemas de arquitectura a principios del desarrollo de software, de esta manera, evitar grandes costos potenciales de re-trabajar después por cuestiones de seguridad. Comenzando con las actividades ligeras para construir un entendimiento de los detalles relevantes de la seguridad de una arquitectura, una organización evoluciona hacia una inspección más formal de los métodos que verifiquen la integridad en la provisión de mecanismos de seguridad. A nivel organización, los servicios de revisión de diseño son construidos y ofrecidos a los interesados. En una forma sofisticada, proveer esta práctica involucra una inspección de diseños detallada, a nivel de datos y la aplicación de las bases esperadas para conducir una evaluación de diseño y revisión de fallos antes de que el código sean aceptado.



### Revisión de código

La Práctica de revisión de código (CR por sus siglas en inglés) está enfocada en inspeccionar software al nivel de código fuente para encontrar vulnerabilidades de seguridad. Las vulnerabilidades a nivel de código son generalmente sencillas de entender. Pero incluso desarrolladores informados pueden fácilmente cometer errores que dejan el software abierto a un compromiso potencial. Para empezar, una organización usa listas de verificación sencillas y, por eficiencia, solo inspecciona los módulos más críticos del software. Sin embargo, conforme una organización evoluciona, utiliza la tecnología de automatización para mejorar dramáticamente la cobertura y la eficacia de las actividades de revisión de código. Una sofisticada disposición de esta Práctica involucra una integración más profunda de la revisión de código en el proceso de desarrollo para permitir equipos de proyecto encontrar problemas antes. Esto también permite a las organizaciones una mejor auditoría y conjunto de expectativas para los resultados de la revisión de código antes de que pueda hacerse la liberación del código.



### Pruebas de seguridad

La Práctica de Prueba de Seguridad (ST por sus siglas en inglés) está enfocada en la inspección de software en el ambiente de ejecución con el fin de encontrar problemas de seguridad. Estas actividades de pruebas refuerzan los casos de seguro para software verificándolo en el mismo contexto en el cual se espera será ejecutado, así hace visible las malas configuraciones operacionales o errores en la lógica de negocio que son difíciles de encontrar de otra manera. Empezando con una prueba de intrusión y casos de prueba a alto nivel basados en la funcionalidad del software, una organización evoluciona hacia el uso de pruebas de seguridad automatizadas para cubrir la amplia variedad de casos de prueba que podrían demostrar una vulnerabilidad en el sistema. En una forma avanzada, el ofrecer esta Práctica implica la personalización de las pruebas automatizadas para construir una serie de pruebas de seguridad que cubran a detalle las preocupaciones específicas sobre la aplicación. Con una visibilidad adicional a nivel organización, las pruebas de seguridad permiten a las organizaciones establecer las expectativas mínimas para los resultados de las pruebas de seguridad antes que la liberación de un proyecto sea aceptada.

# Verificación

## Resumen de actividades

### Revisión de diseño

...continúa en página 58



<b>OBJETIVOS</b>	<b>Apoyar en las revisiones de diseño de software para asegurarse que existan los lineamientos de mitigación para riesgos conocidos</b>	<b>Ofrecer evaluaciones de servicios para revisar el diseño del software contra buenas prácticas integrales de seguridad</b>	<b>Exija evaluar y valide los artefactos para desarrollar un entendimiento detallado de mecanismos de protección</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Identificar superficies de ataques de software</li> <li>B. Analizar el diseño contra requisitos de seguridad conocidos</li> </ul>	<ul style="list-style-type: none"> <li>A. Inspeccionar por completo la provisión de los mecanismos de seguridad</li> <li>B. Implementar el servicio de revisión de diseño para los equipos de proyecto</li> </ul>	<ul style="list-style-type: none"> <li>A. Desarrollar diagrama de flujo de datos para recursos sensible</li> <li>B. Establecer puntos de liberación para la revisión de diseño</li> </ul>

### Revisión de código

...continúa en página 62



<b>OBJETIVOS</b>	<b>Encontrar oportunamente vulnerabilidades básicas a nivel de código y otros problemas de seguridad de alto riesgo</b>	<b>Hacer revisiones de código más precisas y eficientes durante el desarrollo a través de la automatización</b>	<b>Exigir un proceso de revisión de código integral para descubrir riesgos específicos de la aplicación y a nivel del lenguaje</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Crear listas de verificación para la revisión de los requisitos de seguridad conocidos</li> <li>B. Realizar revisiones en código de puntos de alto riesgo</li> </ul>	<ul style="list-style-type: none"> <li>A. Utilizar herramientas automatizadas de análisis de código</li> <li>B. Integrar análisis de código en el proceso de desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>A. Personalizar el análisis de código para las preocupaciones específicas de la aplicación</li> <li>B. Establecer puntos de control para la liberación de las revisiones de código</li> </ul>

### Pruebas de seguridad

...continúa en página 66



<b>OBJETIVOS</b>	<b>Establecer el proceso para realizar pruebas de seguridad basándose en la implementación y los requisitos del software</b>	<b>Hacer pruebas de seguridad durante el desarrollo, más completas y eficientes a través de la automatización</b>	<b>Exigir pruebas de seguridad específicas a la aplicación para asegurarse que los lineamientos de seguridad están implementados antes de la publicación</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Deducir casos de prueba desde los requisitos de seguridad conocidos</li> <li>B. Conducir pruebas de intrusión en cada publicación del software</li> </ul>	<ul style="list-style-type: none"> <li>A. Utilizar herramientas automatizadas para pruebas de seguridad</li> <li>B. Integrar pruebas de seguridad en el proceso de desarrollo</li> </ul>	<ul style="list-style-type: none"> <li>A. Emplear automatización de pruebas de seguridad específicas de la aplicación</li> <li>B. Establecer puntos de control para la liberación de las revisiones de código</li> </ul>

# Implementación

## Descripción de prácticas de seguridad



### Administración de vulnerabilidades

La práctica de administración de vulnerabilidades (AV por sus siglas en Inglés) está enfocada en los procesos de una organización con respecto al manejo de reportes de vulnerabilidades e incidentes operativos. Al tener estos procesos establecidos, los proyectos de una organización tendrán expectativas consistentes y una mayor eficiencia para manejar estos eventos, en lugar de respuestas caóticas y sin uniformidad. Empezando con la asignación de roles en caso de un incidente, una organización genera un proceso de respuesta a incidentes más formal, que asegura la visibilidad y el seguimiento de los problemas que ocurran. Las comunicaciones también se mejoran para mejorar el entendimiento global de los procesos.

De forma avanzada, la administración de vulnerabilidades implica una disección completa de los incidentes y los reportes de vulnerabilidades para obtener métricas detalladas e información sobre las causas raíz para proveer retroalimentación al comportamiento de la organización.



### Fortalecimiento del ambiente

La práctica de reforzamiento de ambientes (EA por sus siglas en Inglés) se enfoca en construir el aseguramiento del ambiente de ejecución que alberga los programas de la organización. Debido a que la operación segura de una aplicación se puede deteriorar por problemas en componentes externos, asegurar esta infraestructura base directamente mejora la postura de seguridad general del programa. Empezando con un simple seguimiento y distribución de información sobre el ambiente operativo para mantener mejor informados a los equipos de desarrollo, una organización evoluciona a métodos escalables para administrar la instalación de parches de seguridad y equipar el ambiente operativo con detectores tempranos de potenciales problemas de seguridad antes de que el daño se materialice. Conforme una organización avanza, el ambiente operativo se revisa y se refuerza con la instalación de herramientas de producción para agregar capas de defensa y redes de seguridad para limitar el daño en caso de que alguna vulnerabilidad sea explotada.



### Habilitación operativa

La práctica de habilitación de operativa (HO por sus siglas en Inglés) se enfoca en la recolección de información crítica de seguridad de los equipos de proyectos que construyen programas y en comunicar esta información a los usuarios y operadores del programa. Sin esta información, aún el programa diseñado más seguramente corre riesgos no planeados, ya que algunas características importantes y opciones de seguridad no serán conocidas en el sitio de publicación. Empezando con documentación preliminar para capturar los detalles más impactantes para los usuarios y operadores, una organización evoluciona hacia la construcción de guías completas de seguridad de operaciones que se entregan con cada distribución.

De forma avanzada, la habilitación de las operaciones también comprende pruebas a nivel organización para cada uno de los equipos de proyecto, esto, para asegurar que la información sea capturada y compartida de acuerdo a las expectativas.






# Implementación

## Resumen de actividades




### Administración de vulnerabilidades

...continúa en página 70

	 VM 1	 VM 2	 VM 3
<b>OBJETIVOS</b>	Entender el plan de alto nivel para responder a los reportes o incidentes de vulnerabilidades	Elaborar expectativas para prácticas de respuesta para mejorar la consistencia y las comunicaciones	Mejorar en análisis y la colección de datos en el proceso de respuesta para retroalimentación en la planeación proactiva
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Identificar un punto de contacto para problemas de seguridad</li> <li>B. Crear equipo(s) informal(es) de respuesta de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Establecer un proceso consistente de respuesta a incidentes</li> <li>B. Adoptar un proceso de divulgación de problemas de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Conducir análisis de causa raíz para incidentes</li> <li>B. Recolectar métricas por incidente</li> </ul>




### Fortalecimiento del ambiente

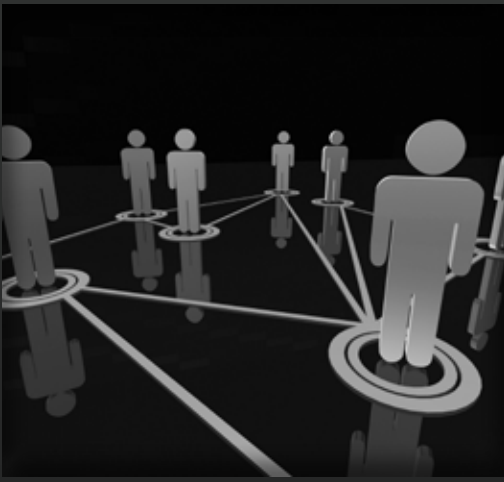
...continúa en página 74

	 EH 1	 EH 2	 EH 3
<b>OBJETIVOS</b>	Entender el ambiente operativo base para aplicaciones y componentes de sistemas	Mejorar la confianza en las operaciones de aplicaciones al reforzar el ambiente operativo.	Validar la salud de las aplicaciones y el estado de los ambientes operativos contra las mejores prácticas conocidas.
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Mantener una especificación de ambiente operativo</li> <li>B. Identificar e instalar actualizaciones y parches críticos de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Establecer un proceso rutinario de administración de parches</li> <li>B. Monitoreo del estado de configuración básico del ambiente</li> </ul>	<ul style="list-style-type: none"> <li>A. Identificar e implementar herramientas de protección relevantes para las operaciones</li> <li>B. Expandir el programa de auditoría hacia la configuración de ambientes</li> </ul>

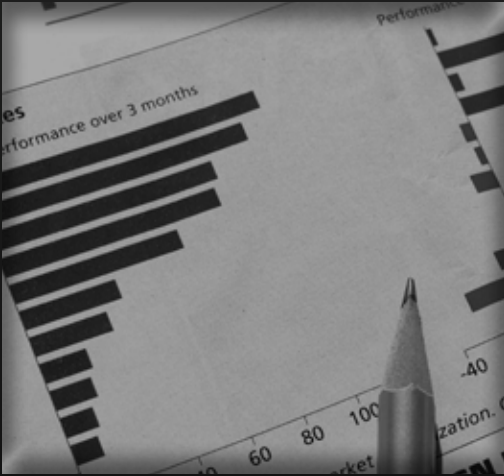
### Habilitación operativa

...continúa en página 78

	 OE 1	 OE 2	 OE 3
<b>OBJETIVOS</b>	Habilitar las comunicaciones entre los equipos de desarrollo y los operadores para datos críticos relevantes a seguridad	Mejorar las expectativas de operaciones seguras y continuas al proveer procedimientos detallados	Exigir la comunicación de información sobre seguridad y validar que los artefactos estén completos
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Capturar la información de seguridad crítica para el ambiente de publicación</li> <li>B. Documentar procedimientos para alertas de aplicación típicas</li> </ul>	<ul style="list-style-type: none"> <li>A. Crear procedimientos de administración de cambio por distribución</li> <li>B. Mantener guías formales de seguridad de operaciones</li> </ul>	<ul style="list-style-type: none"> <li>A. Expandir el programa de auditoría para información operativa</li> <li>B. Realizar firma de código para componentes de aplicaciones</li> </ul>



```
private $host;  
private $username;  
private $password;  
private $database;  
private $charset;  
  
private $link = null;  
  
public function connect()  
{  
    self::$link = mysql_connect(self::$host,  
    if (!self::$link)  
        throw new MySQLException("Connect failed");  
    mysql_query("SET CHARACTER SET " . self::$charset);  
    mysql_select_db(self::$database);  
}
```



# Aplicando el Modelo

Haciendo que todo trabaje en conjunto



Esta sección cubre varias aplicaciones útiles e importantes de SAMM. Dado el diseño central del modelo en sí, una organización puede usar SAMM como un marco de referencia para medir su programa de aseguramiento de Software y crear una tarjeta de calificaciones. Usando tarjetas de calificaciones, una organización puede demostrar mejoras a través de las iteraciones de desarrollo en el programa de aseguramiento. Y lo más importante, la organización puede usar las plantillas de planes de implementación de SAMM como guía para construir o mejorar su iniciativa de aseguramiento de Software.

# Usando los Niveles de Madurez

Cada una de las doce prácticas de seguridad tiene tres niveles de madurez. Cada nivel tiene muchos componentes que especifican los factores críticos para entender y alcanzar el nivel deseado. Más aun, los detalles prescritos hacen posible usar las definiciones de las prácticas de seguridad incluso fuera del contexto de usar SAMM para construir un programa de aseguramiento de Software.

## OBJETIVO

El objetivo es generar una postura general que capture la meta de aseguramiento al alcanzar el nivel de madurez asociado. Conforme el nivel se incrementa para una práctica en particular, los objetivos representan metas más sofisticadas. Esto, en cuanto a construir el aseguramiento para el desarrollo y publicación de software.

## ACTIVIDADES

Las actividades son requisitos indispensables para obtener cada nivel. Algunas fueron creadas para realizarse en toda la organización y algunas corresponden a acciones para cada uno de los equipos de proyecto. En cualquier caso, las actividades representan las funciones de seguridad principales, las organizaciones son libres para determinar como cumplirán con cada actividad.

## RESULTADOS

Los resultados demuestran las capacidades y capacidad de ejecución al obtener un nivel dado. En algunos casos, estos resultados son especificados concretamente y en otros se hace una descripción más cualitativa sobre las capacidades a incrementar.

## MÉTRICAS DE ÉXITO

Las métricas de éxito especifican ejemplos de mediciones que pueden ser usadas para verificar si una organización se desempeña en un nivel dado. La administración y recolección de datos se deja a elección de cada organización, pero se provee de las fuentes y niveles de datos recomendados.

## COSTOS

Los costos con descripciones cualitativas sobre los costos en que incurre una organización al obtener un nivel dado. Aunque los valores específicos variarán para cada organización, el objetivo es dar una idea de los costos iniciales y continuos asociados con operar en un nivel en particular.

## PERSONAL

Este tipo de propiedades para cada nivel indican los costos de operación en términos de recursos humanos para operar en un nivel dado.

- ◆ *Desarrolladores* - Individuos que desempeñan diseño e implementación detallada de software
- ◆ *Arquitectos* - Individuos que desempeñan trabajo de diseño de alto nivel e ingeniería de sistemas de alta escala.
- ◆ *Administradores* - Individuos haciendo la administración diaria del equipo de desarrollo.
- ◆ *Testadores de Calidad* - Los individuos realizando pruebas de aseguramiento de calidad y verificación previas a la publicación del software
- ◆ *Auditores de Seguridad* - Individuos con conocimiento técnico sobre seguridad del software que está siendo construido
- ◆ *Dueños de Negocio* - Individuos que realizan la toma de decisiones sobre el software y los requisitos del negocio
- ◆ *Soporte a Operaciones* - Individuos realizando soporte a clientes o soporte directo a las operaciones

## NIVELES RELACIONADOS

Los niveles relacionados son referencias a los niveles en otras prácticas que tienen el potencial de traslaparse dependiendo de la estructura organizacional y el avance en el programa de aseguramiento de software. Funcionalmente, estas indican sinergias y optimizaciones en la implementación de actividades si el nivel relativo es también un objetivo o ya está realizado.

# Realizando Revisiones

Al medir una organización contra las prácticas de seguridad definidas, obtenemos un panorama general de las actividades de seguridad existentes. Este tipo de revisiones es útil para entender la amplitud de las actividades de seguridad que se han creado en la organización. Más aun, permite que la organización utilice SAMM para crear un plan de implementación futuro para la mejora continua. Realizar una revisión es simplemente evaluar a una organización para determinar el nivel de madurez en el cual se está desempeñando. La profundidad hasta la cual se debe de revisar el desempeño de una organización variará dependiendo de los objetivos de la revisión, pero en general, hay dos estilos recomendados:

- ◆ **Ligero** - Las hojas de trabajo de cada práctica son evaluadas y las calificaciones son asignadas en base a las respuestas. Este tipo de revisión usualmente es suficiente para una organización que intenta comparar su programa de aseguramiento actual contra el SAMM y desea tener rápidamente una idea de donde están posicionados.
- ◆ **Detallado** - Después de completar la revisión de las hojas de trabajo, se realiza trabajo de auditoría adicional para verificar que las actividades prescritas por cada práctica existan y funcionen. Adicionalmente, dado que cada práctica también especifica métricas de éxito, esta información debe ser recolectada para asegurarse que la organización se está desempeñando como se espera.



Calificar una organización usando las hojas de trabajo para revisión es sencillo. Después de responder las preguntas, evalúe la columna de respuestas para determinar el nivel. El cual se indicaría con respuestas afirmativas para todas las preguntas sobre los marcadores a la derecha de la columna de respuestas.

Los programas de aseguramiento ya existentes no siempre consisten de actividades que ajustan exactamente en los límites de los niveles de madurez, por ejemplo, una organización que verifica si está en nivel 1 para una práctica en particular puede también tener actividades adicionales pero que no cumplen completamente con el nivel 2. Para esos casos, el calificación debe ser anotada con un símbolo de “+” para indicar que hay medidas implementadas mas allá de lo requerido por el nivel. Por ejemplo, una organización que esta realizando todas las actividades del nivel 1 para Habilitación Operativa , así como una actividad de nivel 2 o 3 sería asignado un calificación “1+”. Así mismo, una organización realizando todas las actividades para una práctica de seguridad, incluyendo algunas mas allá del alcance de SAMM se le daría un calificación de “3+”.






# Gobierno

## Hoja de trabajo para evaluación




### Estrategia y métricas

Si/No

◆ ¿Existe un programa de aseguramiento de la seguridad de software?		
◆ ¿Entienden la mayoría de los interesados en el negocio el perfil de riesgos de la organización?		
◆ ¿Está conciente la mayoría del personal de desarrollo de los planes futuros para el programa de aseguramiento?		
◆ ¿Están la mayoría de las aplicaciones y recursos organizadas por riesgo?		 <b>SM 1</b>
◆ ¿Son las calificaciones de riesgo utilizadas para adaptar las actividades de aseguramiento requeridas?		
◆ ¿La mayoría de la organización sabe lo se les requiere basado en calificación de riesgos?		 <b>SM 2</b>
◆ ¿Se recolectan datos por proyecto del costo de las actividades de aseguramiento?		
◆ ¿La organización compara regularmente los gastos de seguridad contra los de otras organizaciones?		 <b>SM 3</b>




### Política y cumplimiento

Si/No

◆ ¿La mayoría de los involucrados en el proyecto conocen el estado de cumplimiento del mismo?		
◆ ¿Son los requisitos de cumplimiento específicamente considerados por equipos de proyecto?		 <b>PC 1</b>
◆ ¿La organización utiliza un conjunto de políticas y estándares para controlar el desarrollo de software?		
◆ Los equipos de proyecto ¿Son capaces de solicitar una auditoría de cumplimiento con políticas y estándares?		 <b>PC 2</b>
◆ Los proyectos ¿Son auditados periódicamente para asegurar una base de cumplimiento con políticas y estándares?		
◆ ¿La organización usa auditorías sistemáticas para recolectar y controlar evidencia de cumplimiento?		 <b>PC 3</b>

### Educación y orientación

Si/No




◆ ¿La mayoría de los desarrolladores han recibido entrenamiento de alto nivel sobre concientización de seguridad?		
◆ ¿Cada equipo tiene acceso a mejores prácticas y orientación para desarrollo seguro?		 <b>EG 1</b>
◆ ¿Se les ha dado entrenamiento específico y orientación a la mayoría de los roles en el proceso de desarrollo?		
◆ La mayoría de los involucrados en el proyecto, ¿Son capaces de obtener mentores de seguridad para usar en sus proyectos?		 <b>EG 2</b>
◆ ¿Los lineamientos relacionados con seguridad son controlados centralizadamente y distribuidos consistentemente a lo largo de la organización?		
◆ ¿La mayoría de la gente es evaluada para asegurar un conjunto de habilidades básicas para prácticas de desarrollo seguro?		 <b>EG 3</b>

# Construcción

## Hoja de trabajo para evaluación




### Evaluación de amenaza

Si/No

♦ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?		
♦ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?		 TA 1
♦ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?		
♦ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?		
♦ ¿Los equipos de proyectos analizan regularmente los requisitos funcionales para descubrir probables abusos?		 TA 2
♦ Los equipos de proyecto, ¿Consideran específicamente los riesgos derivados de el software externo?		
♦ ¿Todos los mecanismos de protección y control son registrados y comparados con las amenazas?		 TA 3




### Requisitos de seguridad

Si/No

♦ La mayoría de los equipos de proyecto, ¿especifican algunos requisitos de seguridad durante el desarrollo?		
♦ ¿Obtienen los equipos de proyecto los requisitos de las mejores prácticas y guías de cumplimiento?		 SR 1
♦ La mayoría de los interesados ¿Revisan las matrices de control de acceso para los proyectos importantes?		
♦ ¿Están los equipos de proyecto especificando los requisitos basándose en la retroalimentación de otras actividades de seguridad?		 SR 2
♦ ¿Están la mayoría de los interesados revisando los acuerdos con proveedores para los requisitos de seguridad?		
♦ Los requisitos de seguridad ¿son especificados por los equipos de proyecto que están siendo auditados?		 SR 3

### Arquitectura de seguridad

Si/No




♦ ¿Cuentan los equipos de proyectos con una lista de los componentes de terceros recomendados?		
♦ ¿Están la mayoría de los equipos de proyecto conscientes de los principios de diseño seguro y su aplicación?		 SA 1
♦ ¿Hace publicidad de los servicios compartidos de seguridad como guía para equipos de proyectos?		
♦ ¿Están los equipos de proyectos previstos con los patrones de diseño prescriptivo basado en su arquitectura de aplicación?		 SA 2
♦ ¿Están los equipos de proyecto construyendo software a partir de plataformas y marcos de trabajo controlados?		
♦ ¿Están los equipos de proyecto siendo auditados para el uso de componentes seguros de arquitectura?		 SA 3

# Verificación

## Hoja de trabajo para evaluación




### Revisión de diseño

Si/No

◆ ¿Documentan los equipos de proyecto el perímetro de ataque de los diseños de software?		
◆ ¿Comprueban los equipos de proyecto los diseños de software contra los riesgos de seguridad conocidos?		
◆ ¿La mayoría de los equipos de proyecto analizan el diseño específicamente para los mecanismos de seguridad?		 <b>DR 1</b>
◆ ¿La mayoría de los interesados están consientes de cómo obtener una revisión de diseño formal?		 <b>DR 2</b>
◆ ¿El proceso de revisión de diseño incorpora un análisis detallado a nivel de datos?		
◆ ¿Las auditorias de proyecto rutinarias necesitan los lineamientos para los resultados de la revisión de diseño?		 <b>DR 3</b>




### Revisión de código

Si/No

◆ ¿La mayoría de los equipos de proyecto tienen listas de verificación basadas en los problemas más comunes?		
◆ Los equipos de proyecto ¿Generalmente realizan revisiones de algunos de los mayores riesgos en el código?		
◆ ¿Pueden la mayoría de los equipos de proyecto acceder a herramientas automatizadas de análisis de código para encontrar problemas de seguridad?		 <b>CR 1</b>
◆ ¿La mayoría de los interesados requieren y revisan constantemente los resultados de las revisiones de código?		 <b>CR 2</b>
◆ ¿La mayoría de los equipos de proyecto utilizan automatización para comprobar código contra los estándares de programación específicos de la aplicación?		
◆ ¿Las auditorias de rutina del proyecto necesitan lineamientos para los resultados de la revisión de código antes de la liberación?		 <b>CR 3</b>

### Pruebas de seguridad

Si/No

◆ ¿Están los proyectos especificando pruebas de seguridad basándose en los requisitos?		
◆ ¿La mayoría de los proyectos realizan pruebas de intrusión antes de la liberación?		
◆ ¿Están los interesados consientes del estado de las pruebas de seguridad antes de la liberación?		 <b>ST 1</b>
◆ ¿Están los proyectos usando automatización para evaluar los casos de prueba de seguridad?		
◆ ¿La mayoría de los proyectos siguen un proceso consistente para evaluar y reportar las pruebas de seguridad a los involucrados?		 <b>ST 2</b>
◆ Los casos de seguridad ¿son generados principalmente para la lógica específica de la aplicación?		
◆ ¿Las auditorias rutinarias requieren un estándar mínimo de resultados de las pruebas de seguridad?		 <b>ST 3</b>






# Implementación

## Hoja de trabajo para evaluación




### Administración de vulnerabilidades

Si/No

◆ ¿Tienen la mayoría de los proyectos un punto de contacto para problemas de seguridad?		
◆ ¿Tienen su organización un equipo asignado para respuestas a incidentes de seguridad?		
◆ ¿Conocen la mayoría de los equipos de proyecto su(s) punto(s) de contacto de seguridad y equipo(s) de respuesta?		 <b>VM 1</b>
◆ ¿Usa la organización un proceso consistente para reporte y manejo de incidentes?		
◆ ¿Conocen la mayoría de los interesados en el proyecto las publicaciones de seguridad relevantes y relacionadas con sus proyectos de sistemas?		 <b>VM 2</b>
◆ ¿Son inspeccionados la mayoría de los incidentes para encontrar la causa raíz y generar más recomendaciones?		
◆ La mayoría de los proyectos ¿obtienen y reportan consistentemente datos y métricas relacionadas con incidentes?		 <b>VM 3</b>




### Fortalecimiento del ambiente

Si/No

◆ ¿Documentan la mayoría de los proyectos algunos requisitos para el ambiente operativo?		
◆ ¿Revisan la mayoría de los proyectos actualizaciones de seguridad para componentes de sistemas de terceros?		 <b>EH 1</b>
◆ ¿Se usa un proceso consistente para aplicar actualizaciones y parches a dependencias críticas?		
◆ ¿Utilizan la mayoría de los proyectos la automatización para verificar la salud de aplicaciones y ambientes?		 <b>EH 2</b>
◆ Los interesados están enterados de opciones de herramientas adicionales para proteger sistemas mientras se ejecutan las operaciones?		
◆ Las auditorías de rutina verifican la salud de los ambientes base de la mayoría de los proyectos?		 <b>EH 3</b>

### Habilitación operativa

Si/No

◆ ¿Entrega notas de seguridad con la mayoría de las distribuciones de sistemas?		
◆ ¿Están documentadas las alertas de seguridad y las condiciones de error para la mayoría de los proyectos?		 <b>OE 1</b>
◆ ¿Están usando la mayoría de los proyectos un proceso de administración de cambio que es bien entendido?		
◆ ¿Entregan los equipos de proyecto una guía de seguridad de operaciones con cada liberación del producto?		 <b>OE 2</b>
◆ ¿Están la mayoría de los proyectos siendo auditados para verificar que cada entrega tenga la información de seguridad operativa apropiada?		
◆ ¿Se realiza rutinariamente la firma de código en los componentes de sistemas usando un proceso consistente?		 <b>OE 3</b>

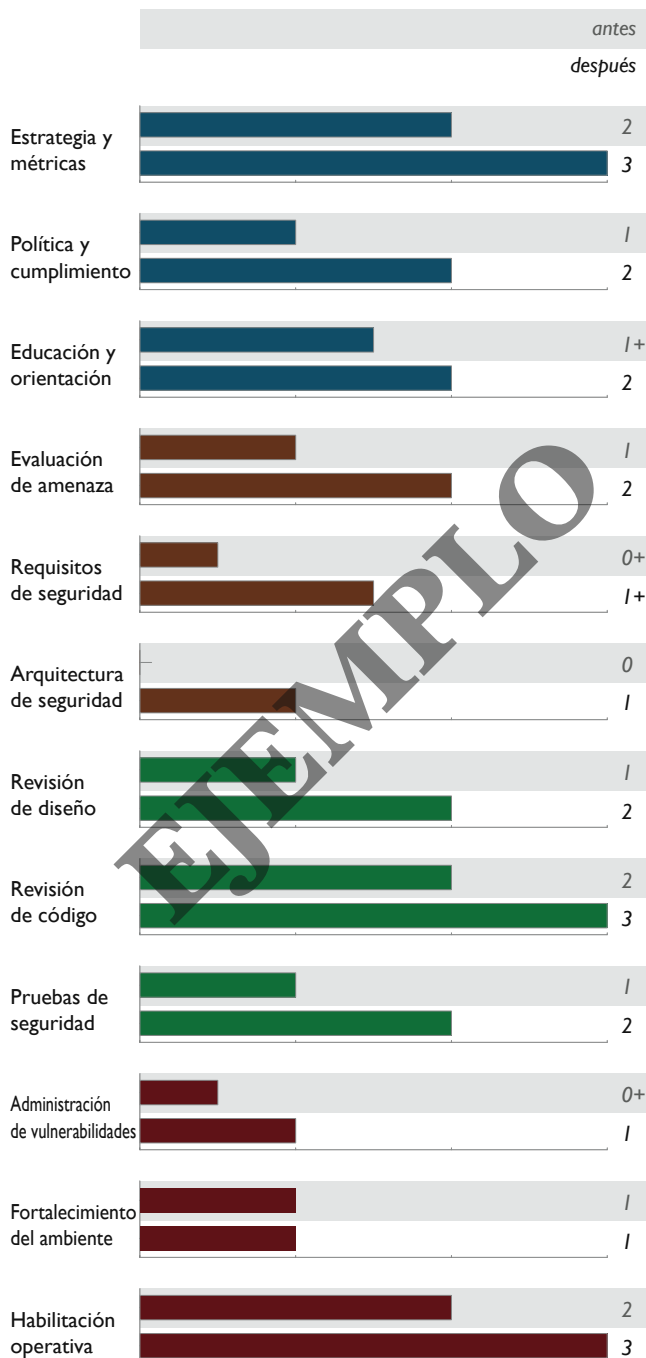
# Creando Tarjetas de Calificaciones

Basado en las calificaciones asignados a cada práctica de seguridad, una organización puede crear una tarjeta de calificaciones para captura esos valores. Funcionalmente, una tarjeta de calificaciones puede ser un conjunto simple de 12 calificaciones en un momento en particular. Sin embargo, seleccionar un intervalo de tiempo en el cual generar la tarjeta de calificaciones facilita el entendimiento de los cambios en el programa de aseguramiento durante un periodo de tiempo.

Se recomienda usar una tarjeta de calificaciones comparativa por varias razones:

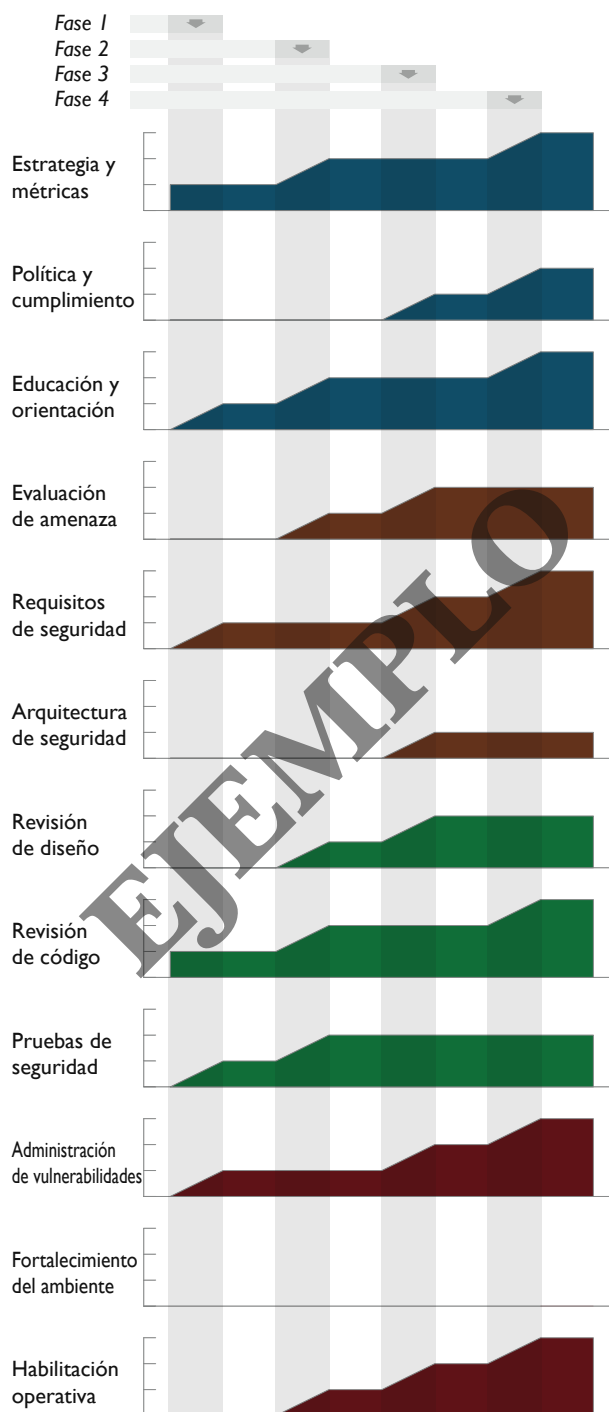
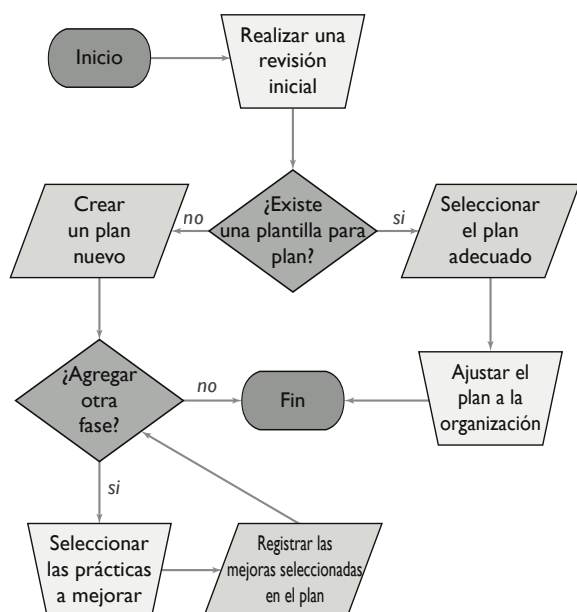
- ◆ *Análisis diferencial* - Registrar las calificaciones de la revisión detallada contra los niveles de desempeño esperados
- ◆ *Demostrar mejoras* - Registrar las calificaciones de antes y después de una iteración del programa de aseguramiento
- ◆ *Mediciones continuas* - Registrar las calificaciones sobre periodos de tiempo consistentes para un programa de aseguramiento que ya existe.

La figura a la derecha muestra un ejemplo de una tarjeta de calificaciones y muestra como un programa de aseguramiento de una organización cambió en el curso de un año. Si esa organización guardó la información sobre como planearon estar al final de año, ese hubiera sido otro conjunto de datos interesantes para graficar dado que ayudaría a mostrar que tanto tuvieron que cambiar los planes durante el año.



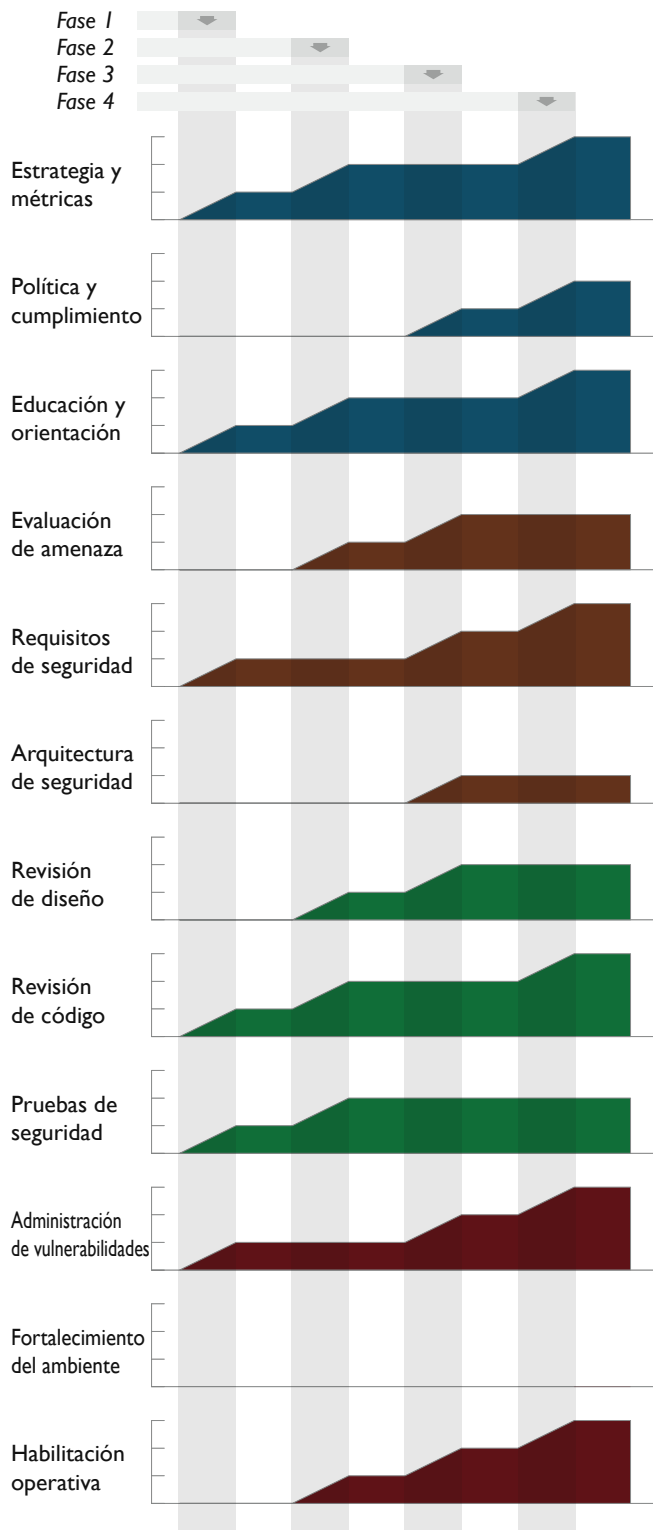
# Construyendo Programas

Uno de los usos principales del SAMM es ayudar a las organizaciones a construir programas de aseguramiento de Software. Ese proceso es sencillo y generalmente comienza con una revisión, si es que la organización ya está haciendo algunas actividades de aseguramiento. Se proveen varias plantillas de planes de implementación para organizaciones bien conocidas. Por lo tanto, la mayoría de las organizaciones pueden escoger el plan de implementación más apropiado y adaptarlo a sus necesidades. Para otro tipo de organizaciones, puede ser necesario que se cree un plan personalizado. Los planes de implementación (mostrados a la derecha) consisten en fases (las barras verticales) en las cuales varias prácticas van mejorando un nivel a la vez. Por lo tanto, construir un plan de implementación implica seleccionar cuales prácticas van a mejorar en cada fase planeada. Las organizaciones son libres de planear a futuro como lo deseen, pero los exhortamos a iterar basándose en las necesidades del negocio y la información específica del negocio para asegurar que los objetivos correspondan a las metas y tolerancia a riesgo del negocio. Después de que el plan de implementación está establecido, la construcción de un programa de aseguramiento es simple. Una organización comienza una fase de mejora y trabaja para llegar a los niveles establecidos realizando las actividades predefinidas. Al final de la fase, el plan debe ser ajustado basándose en lo que realmente fue realizado y entonces la siguiente fase puede comenzar.



# Proveedor Independiente de Software

## Plantilla para Plan de Implementación



### RAZONAMIENTO

Un proveedor independiente de software involucra en sus funciones principales de negocio la construcción y venta de componentes de software y aplicaciones. La motivación inicial de reducir las vulnerabilidades comunes que afectan a sus clientes y usuarios llevar a enfocarse inicialmente en actividades de revisión de código y pruebas de intrusión. Cambiando hacia una prevención más proactiva de los errores de seguridad en las especificaciones del producto. Con el tiempo, estas organizaciones van agregando actividades de levantamiento de requisitos de seguridad. También, para minimizar el impacto de cualquier problema de seguridad descubierto, la organización debe crear actividades de administración de vulnerabilidades. Conforme la organización madura, las actividades de transferencia de conocimiento para la habilitación operativa se agregan para instruir a los clientes y usuarios sobre la operación segura del software.

### Desarrollo Externo

Para las organizaciones con recursos de desarrollo externos, el no tener acceso al código, típicamente lleva a la priorización de actividades de levantamiento de requisitos de seguridad, en ves de actividades de revisión de código. Adicionalmente, hacer análisis avanzado de amenazas en fases tempranas permitirá a la organización aclarar mejor sus necesidades de seguridad a los desarrolladores externos contratados. Dada su experiencia en la configuración de software, esta generalmente es mas fuerte en el grupo externo, aun así los contratos deben construirse para tomar en cuenta las actividades relacionadas con la habilitación operativa.

### Aplicaciones conectadas a Internet

Las organizaciones construyendo aplicaciones que usan recursos en línea tienen riesgos adicionales provenientes de la infraestructura de Internet donde están publicados los sistemas. Para tomar en cuenta estos riesgos, las organizaciones deberían agregar en sus planes de implementación, actividades adicionales para el reforzamiento (hardening) del ambiente.

### Desarrollo de Controladores y Software de Dispositivos

Para las organizaciones construyendo controladores de bajo nivel o software para sistemas en dispositivos, las vulnerabilidades en el diseño de software pueden ser más dañinas y costosas de reparar. Por lo tanto, los planes de implementación deben ser modificados para enfatizar actividades de arquitectura y diseño seguro en fases tempranas de desarrollo.

### Organizaciones que Crecen con Adquisiciones

En una organización que ha crecido con una adquisición, frecuentemente hay varios equipos de desarrollos, que siguen modelos de desarrollo diferentes, con varios niveles de actividades relacionadas a la seguridad. Una organización como esta puede requerir planes de implementación separados para cada división o equipo de desarrollo para tomar en cuenta los diferentes puntos de inicio así como las preocupaciones de cada proyecto, si es que varios tipos de software se están desarrollando.

# Proveedor de Servicios en Línea

## Plantilla para Plan de Implementación

### RAZONAMIENTO

Un proveedor de servicios en línea incluye en sus funciones principales la construcción de aplicaciones Web y otras interfaces accesibles desde la red. Los motivos para validar la salud general del diseño sin restringir la innovación llevan a concentrarse inicialmente en las revisiones de diseño y las actividades de pruebas de intrusión. Dado que los sistemas críticos van a estar interconectados, las actividades de reforzamiento del ambiente también son agregadas en etapas tempranas e implementadas continuamente para tomar en cuenta los riesgos de los ambientes de publicación. Aunque puede variar según las actividades principales de las organizaciones, las actividades de cumplimiento y políticas podrían empezar a ser implementadas tempranamente y avanzar de acuerdo a la importancia de los factores externos de cumplimiento de regulaciones. Conforme una organización madura, las actividades de análisis de amenazas, requisitos de seguridad y arquitectura segura se agregan lentamente para ayudar a impulsar la seguridad proactiva, después de que algunas expectativas iniciales de seguridad se han definido.

### Desarrollo Externo

Para las organizaciones con recursos de desarrollo externos, el no tener acceso al código, típicamente lleva a la priorización de actividades en el levantamiento de requisitos de seguridad, en ves de actividades de revisión de código. Adicionalmente, hacer análisis avanzado de amenazas en fases tempranas permitirá a la organización aclarar mejor sus necesidades de seguridad a los desarrolladores externos contratados. Dada la experiencia en la configuración de software, esta generalmente será mas fuerte en el grupo externo, aun así los contratos deben construirse para tomar en cuenta las actividades relacionadas con la habilitación operativa.

### Procesamiento de Pagos en Línea

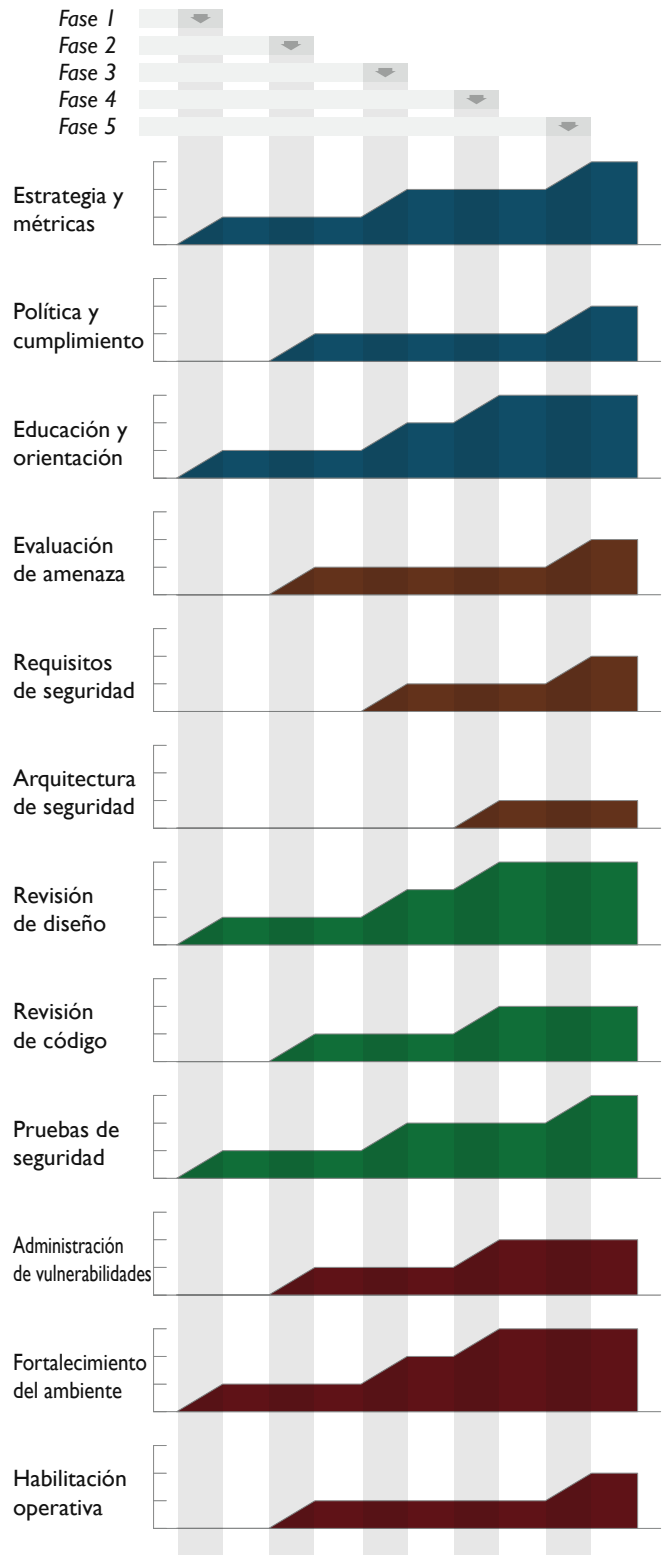
Las organizaciones que requieren estar en cumplimiento con el estándar de seguridad de datos de la industria de tarjetas de pago (PCI-DSS por sus siglas en Inglés) u otro estándar de pago en línea deben agregar actividades de cumplimiento y políticas en las fases iniciales de su plan de implementación. Esto permitiría a las organizaciones establecer actividades oportunamente que permitan el aseguramiento de software y faciliten que el plan se adapte en el futuro.

### Plataformas de Servicios Web

Para organizaciones construyendo plataformas de servicios Web, los errores de diseño pueden acarrear riesgos adicionales y pueden ser más costosos de mitigar. Por lo tanto, las actividades de análisis de amenazas, levantamiento de requisitos de seguridad y seguridad de la arquitectura deben ser agregadas en las fases iniciales del plan de implementación.

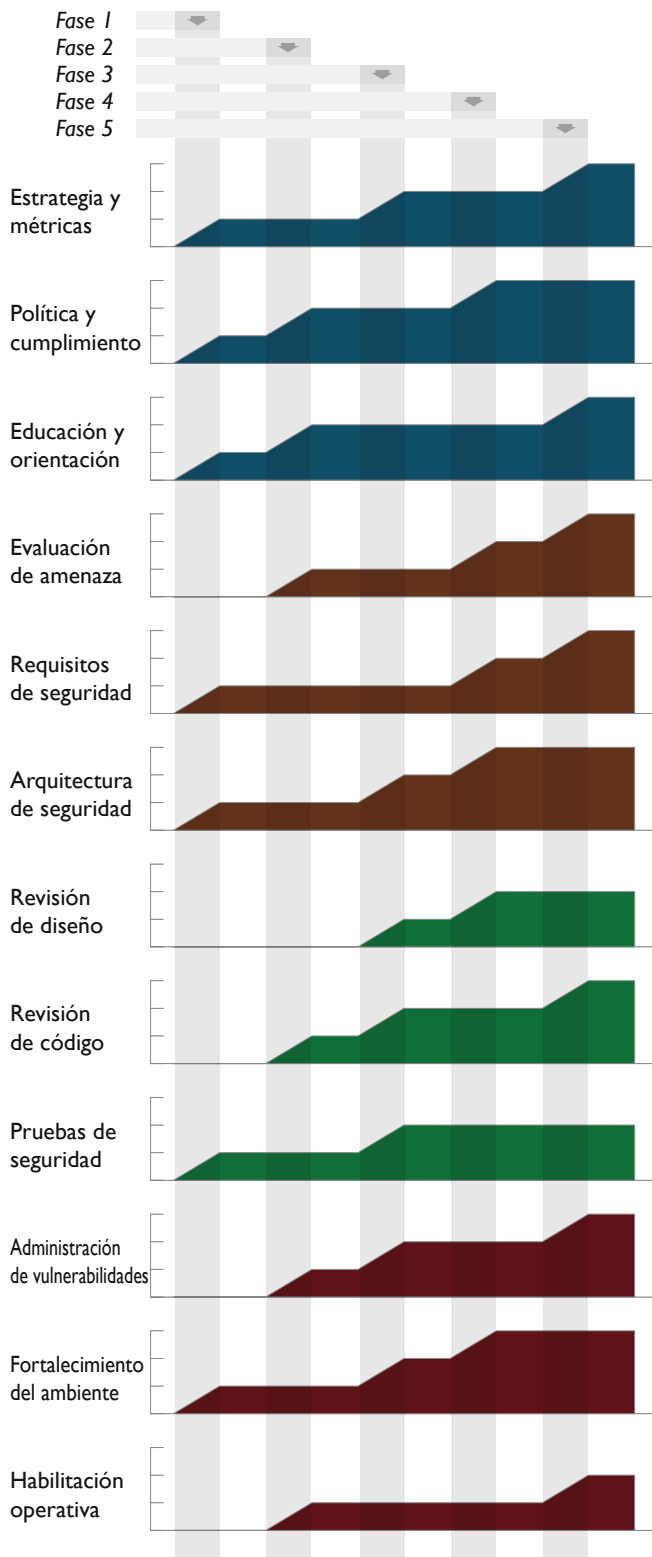
### Organizaciones que Crecen con Adquisiciones

En una organización que ha crecido con una adquisición, frecuentemente hay varios equipos de desarrollo que siguen modelos de desarrollo diferentes con varios niveles de actividades relacionadas a la seguridad. Una organización como esta puede requerir planes de implementación separados para cada división o equipo de desarrollo para tomar en cuenta los diferentes puntos de inicio, así como las preocupaciones de cada proyecto, si es que varios tipos de software se están desarrollando.



# Organización de Servicios Financieros

## Plantilla para Plan de Implementación



### RAZONAMIENTO

Una organización de servicios financieros incluye en sus funciones principales de negocio la construcción de sistemas como apoyo al procesamiento de transacciones financieras. En general, esto implica una mayor concentración en sistemas internos y de respaldo intercomunicados con los de los proveedores de datos externos. Inicialmente, el esfuerzo se enfoca en mejorar las prácticas relacionadas con el gobierno de TI, dado que hay servicios críticos que fijan las bases del programa de aseguramiento y ayudan a cumplir con los requisitos de cumplimiento para la organización. Dado que construir software seguro y confiable proactivamente es una meta común, las prácticas en la construcción se inicial tempranamente y se implementan a detalle conforme el programa madura. Las actividades de verificación también se implementan poco a poco durante el periodo de implementación del plan de implementación para manejar sistemas heredados sin crear expectativas irreales. Adicionalmente, esto ayuda a asegurar que se utilice el tiempo suficiente en construir prácticas más proactivas. Dado que una organización de servicios financieros frecuentemente opera el software que construye, se le da mayor apoyo a las prácticas de desarrollo en medio del plan, después a la implementación de un poco de gobierno, pero antes el mayor apoyo se le da a la construcción proactiva de prácticas.

### Desarrollo Externo

Para las organizaciones con recursos de desarrollo externos, las restricciones de acceso al código llevar típicamente a la priorización de las actividades en los requisitos de seguridad, en ves de actividades de revisión de código. Adicionalmente, haciendo análisis de amenazas en fases tempranas permitirá a la organización clarificar mejor sus necesidades de seguridad a los desarrolladores externos contratados. Dada la experiencia en la configuración de software, esta generalmente será mas fuerte en el grupo externo, aun así los contratos deben construirse para tomar en cuenta las actividades relacionadas con la habilitación operativa.

### Plataformas de Servicios Web

Para organizaciones construyendo plataformas de servicios Web, los errores de diseño pueden acarrear riesgos adicionales y por lo tanto ser más costosos de mitigar. Las actividades de verificación de amenazas, levantamiento de requisitos de seguridad y arquitectura segura deben ser creadas en etapas tempranas del plan de implementación.

### Organizaciones que Crecen con Adquisiciones

En una organización que ha crecido con una adquisición, frecuentemente hay varios equipos de desarrollo que siguen modelos de desarrollo diferentes con varios niveles de actividades relacionadas a la seguridad. Una organización como esta puede requerir planes de implementación separados para cada división o equipo de desarrollo para tomar en cuenta los diferentes puntos de inicio así como las preocupaciones de cada proyecto, si es que varios tipos de software se están desarrollando.

# Organización de Gobierno

## Plantilla para Plan de Implementación

### RAZONAMIENTO

Una organización de gobierno incluye en sus funciones principales de negocio el ser una organización afiliada al estado que construye software para soporte a proyectos del sector público. Inicialmente, las prácticas de gobierno se establecen para tener una idea de la carga para la organización de estar en cumplimiento con regulaciones existentes en el contexto del plan de implementación de mejoras. Dado el riesgo de exposición pública y la cantidad de código heredado, se le da énfasis primero a las pruebas de seguridad en las prácticas de verificación y después se desarrollan las revisiones de código o de diseño. Un énfasis similar se hace en las prácticas de construcción y desarrollo. Esto ayuda a establecer la administración de vulnerabilidades de la organización, mirando hacia reforzar la postura de seguridad en la habilitación operativa. Al mismo tiempo las actividades de seguridad proactiva, aun en construcción, se implementan para ayudar a evitar nuevos problemas en el software bajo desarrollo.

### Desarrollo Externo

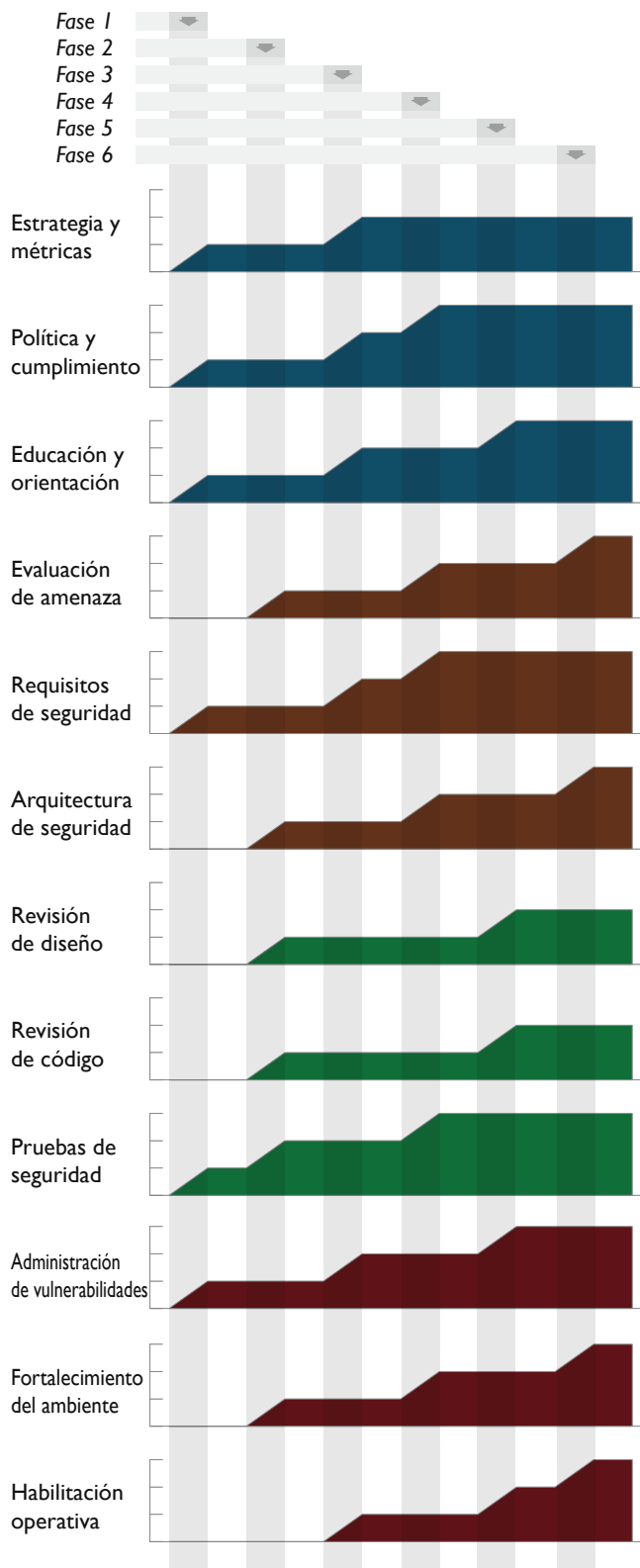
Para las organizaciones con recursos de desarrollo externos, las restricciones de acceso al código llevar típicamente a la priorización de las actividades de levantamiento de requisitos de seguridad, en ves de actividades de revisión de código. Adicionalmente, haciendo análisis de amenazas en fases tempranas permitirá a la organización aclarar mejor sus necesidades de seguridad a los desarrolladores externos contratados. Dada su experiencia en la configuración de software, esta generalmente es mas fuerte en el grupo externo, aun así los contratos deben construirse para tomar en cuenta las actividades relacionadas con la habilitación operativa.

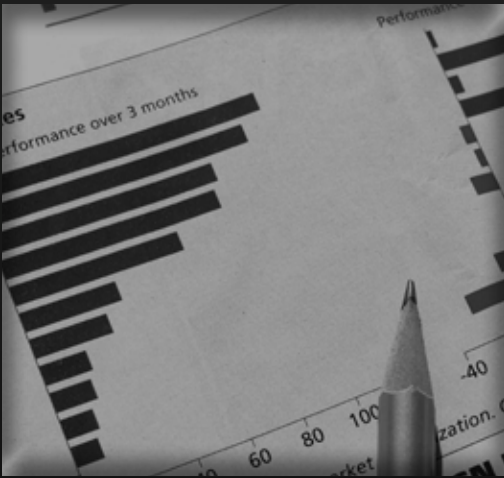
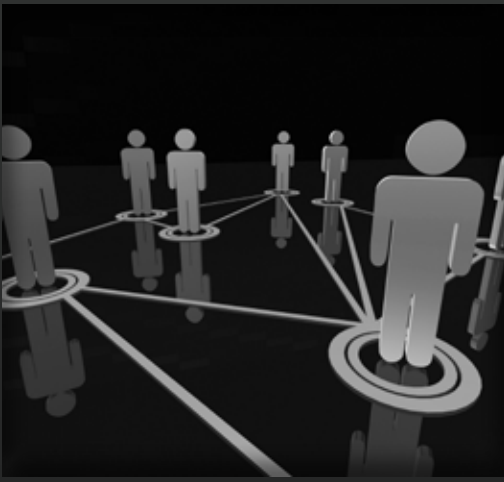
### Plataformas de Servicios Web

Para organizaciones construyendo plataformas de servicios Web, los errores de diseño pueden acarrear riesgos adicionales y por lo tanto ser más costoso de mitigar. Las actividades de verificación de amenazas, requisitos de seguridad y arquitectura segura deben ser creadas en etapas tempranas del plan de implementación

### Cumplimiento de Regulaciones

Para organizaciones bajo estrictas regulaciones que afectan los procesos de negocio, la construcción de la práctica de Cumplimiento y Políticas debe ser ajustada de acuerdo a los factores externos. Así mismo, las organizaciones bajo regulaciones menos estrictas podrían decidir evitar esta práctica a favor de otras.





# Las Prácticas de Seguridad




Una explicación de los detalles





Esta sección define los bloques de construcción del SAMM, los niveles de madurez bajo cada práctica de seguridad. Para cada práctica se enumeran los tres niveles en una tabla resumen. Seguida de la descripción de cada nivel, incluyendo explicaciones detalladas de las actividades requeridas, los resultados que la organización puede esperar al alcanzar cada nivel, las métricas de éxito para medir el desempeño, la inversión permanente en personal y los costos adicionales asociados.

# Estrategia y métricas

	 <b>SM 1</b>	 <b>SM 2</b>	 <b>SM 3</b>
<b>OBJETIVOS</b>	<b>Establecer un plan estratégico unificado para la seguridad del software dentro de la organización</b>	<b>Medir el valor relativo de los datos y bienes, y elegir la tolerancia al riesgo</b>	<b>Alinear los gastos de seguridad con indicadores de negocio pertinentes y el valor de los activos</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Estimar el perfil global de riesgo del negocio</li> <li>B. Crear y mantener un plan de implementación para el programa de aseguramiento</li> </ul>	<ul style="list-style-type: none"> <li>A. Clasificar datos y aplicaciones basado en riesgo de negocio</li> <li>B. Establecer y medir los objetivos de seguridad por cada clasificación</li> </ul>	<ul style="list-style-type: none"> <li>A. Realizar comparaciones de costo periódicas a nivel industria</li> <li>B. Recolectar métricas históricas de gastos de seguridad</li> </ul>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Existe un programa de aseguramiento de la seguridad de software?</li> <li>◆ ¿Entienden la mayoría de los interesados en el negocio el perfil de riesgos de la organización?</li> <li>◆ ¿Está conciente la mayoría del personal de desarrollo de los planes futuros para el programa de aseguramiento?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están la mayoría de las aplicaciones y recursos organizadas por riesgo?</li> <li>◆ ¿Son las calificaciones de riesgo utilizadas para adaptar las actividades de aseguramiento requeridas?</li> <li>◆ ¿La mayoría de la organización sabe lo se les requiere basado en calificación de riesgos?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Se recolectan datos por proyecto del costo de las actividades de aseguramiento?</li> <li>◆ ¿La organización compara regularmente los gastos de seguridad contra los de otras organizaciones?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Una lista concreta de los riesgos más críticos causados por software a nivel negocio.</li> <li>◆ Plan de implementación adaptado que solucione las necesidades de seguridad de la organización con el mínimo esfuerzo.</li> <li>◆ Entendimiento organizacional de cómo va a crecer el programa de aseguramiento a lo largo del tiempo.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Planes de aseguramiento personalizados por proyecto basados en el valor central para el negocio</li> <li>◆ Entendimiento organizacional de la importancia de la seguridad, de bienes de aplicación e información</li> <li>◆ Involucrados en el proyecto mejor informados respecto al entendimiento y aceptación de riesgos</li> </ul>	<ul style="list-style-type: none"> <li>◆ Información para tomar decisiones caso por caso sobre los gastos de seguridad</li> <li>◆ Estimaciones de pérdidas pasadas debidas a problemas de seguridad</li> <li>◆ Consideración por proyecto de gastos de seguridad contra pérdida potencial</li> <li>◆ Debida diligencia a nivel industria referente a seguridad</li> </ul>

# Estrategia y métricas



Establecer un plan estratégico unificado para la seguridad del software dentro de la organización

## ACTIVIDADES

### A. Estimar el perfil global de riesgo del negocio

Entreviste a los dueños del negocio e involucrados en el proyecto y cree una lista de peores escenarios a lo largo de varios bienes de aplicación y datos. De acuerdo a la manera en que la organización cree, use o venda software, la lista de peores escenarios puede variar ampliamente, pero los problemas comunes incluyen: robo o corrupción de información, fallas de servicio, pérdidas monetarias, ingeniería inversa, cuentas comprometidas, etc. Después de capturar ampliamente las ideas de los peores escenarios, organice y seleccione los más importantes basado en la información recolectada y conocimiento del negocio. Cualquier número puede ser seleccionado, pero trate de que sean al menos 3 y no más de 7 para hacer uso eficiente del tiempo y mantener enfocado el ejercicio. Elaborar una descripción de cada uno de los elementos seleccionados y documentar detalles de los factores que contribuyan a los peores escenarios, posibles factores agravantes y posibles factores mitigantes para la organización. El riesgo final del negocio debe ser revisado con los dueños del negocio y demás personas involucradas para su entendimiento.

### B. Crear y mantener un plan de implementación para el programa de aseguramiento

Entender los riesgos principales para la organización, evaluar el desempeño actual de la organización contra cada una de las doce prácticas. Asignar una calificación a cada práctica de 1, 2 o 3 basado en el objetivo correspondiente si es que la organización cumple con las métricas de éxito. Si no se cumple ninguna métrica, asignar una calificación de 0 a la práctica. Una vez que se tiene un buen entendimiento del estado actual, el siguiente objetivo es identificar las prácticas que serán mejoradas en la siguiente iteración. Selecciónelas basándose en el perfil de riesgo de negocio, otros manejadores de negocio, requisitos de cumplimiento, tolerancia de presupuesto, etc. Una vez que las prácticas son seleccionadas, los objetivos de la iteración son alcanzar el siguiente objetivo. Las iteraciones de mejora en el programa de aseguramiento deberán ser de aproximadamente 3-6 meses, pero se debe tener una sesión de estrategia de aseguramiento al menos cada 3 meses para revisar el progreso de las actividades, el desempeño contra las métricas de éxito y otros motivos de negocio que pudieran requerir cambiar el programa.

## EVALUACIÓN

- ◆ ¿Existe un programa de aseguramiento de la seguridad de software?
- ◆ ¿Entienden la mayoría de los interesados en el negocio el perfil de riesgos de la organización?
- ◆ ¿Está conciente la mayoría del personal de desarrollo de los planes futuros para el programa de aseguramiento?

## RESULTADOS

- ◆ Una lista concreta de los riesgos más críticos causados por software a nivel negocio.
- ◆ Plan de implementación adaptado que solucione las necesidades de seguridad de la organización con el mínimo esfuerzo.
- ◆ Entendimiento organizacional de cómo va a crecer el programa de aseguramiento a lo largo del tiempo.

## MÉTRICAS DE ÉXITO

- ◆ >80% de los involucrados sean informados en el perfil de riesgo del negocio en 6 meses
- ◆ >80% del personal informado en el plan de implementación del programa de aseguramiento en 3 meses
- ◆ >1 sesión estratégica del programa de seguridad en 3 meses.

## COSTOS

- ◆ Creación y mantenimiento del perfil de riesgo de negocio
- ◆ Evaluación trimestral del programa de aseguramiento

## PERSONAL

- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (4 días/año)
- ◆ Administradores (4 días/año)
- ◆ Dueños de Negocio (4 días/año)
- ◆ Testadores de Calidad (1 día/año)
- ◆ Auditor de seguridad (4 días/año)

## NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 1
- ◆ Auditoría de amenazas - 1
- ◆ Requisitos de seguridad - 2



## Medir el valor relativo de los datos y bienes, y elegir la tolerancia al riesgo

### EVALUACIÓN

- ◆ ¿Están la mayoría de las aplicaciones y recursos organizadas por riesgo?
- ◆ ¿Son las calificaciones de riesgo utilizadas para adaptar las actividades de aseguramiento requeridas?
- ◆ ¿La mayoría de la organización sabe lo que se requiere basado en calificación de riesgos?

### RESULTADOS

- ◆ Planes de aseguramiento personalizados por proyecto basados en el valor central para el negocio
- ◆ Entendimiento organizacional de la importancia de la seguridad, de bienes de aplicación e información
- ◆ Involucrados en el proyecto mejor informados respecto al entendimiento y aceptación de riesgos

### MÉTRICAS DE ÉXITO

- ◆ >90% de las aplicaciones y bienes de información evaluados contra la clasificación de riesgo en los últimos 12 meses
- ◆ >80% del personal informado de las calificaciones de riesgos de aplicación e información importantes en los últimos 6 meses
- ◆ >80% del personal informado de los planes de implementación importantes del programa de aseguramiento en los últimos 3 meses

### COSTOS

- ◆ Construcción o licenciamiento del esquema de clasificación de riesgos de aplicación e información
- ◆ Esfuerzo adicional de una planeación del programa más granular

### PERSONAL

- ◆ Arquitectos (2 días/año)
- ◆ Administradores (2 días/año)
- ◆ Dueños de Negocio (2 días/año)
- ◆ Auditor de seguridad (2 días/año)

### NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2
- ◆ Evaluación de amenaza - 2

- ◆ Análisis de diseño - 2

### ACTIVIDADES

#### A. Clasificar datos y aplicaciones basado en riesgo de negocio

Establecer un sistema simple de clasificación para representar el nivel de riesgo por aplicación. En su forma más simple, puede ser una organización de Alto/Medio/Bajo. Se pueden utilizar clasificaciones más sofisticadas, pero no debe haber más de siete categorías y deben representar una pendiente de alto a bajo impacto en los riesgos de negocio. Trabajar desde el perfil de riesgo de negocio en la organización, cree criterios de evaluación de proyecto que asigne a cada proyecto una de las categorías de riesgo. Un esquema de clasificación similar pero separado debe crearse para los activos de información y cada elemento debe ser priorizado y clasificado en función del impacto potencial de riesgo de negocio. Evalúe la información recolectada acerca de cada aplicación y asigne una categoría de riesgo a cada una basándose en los criterios de evaluación general y las categorías de riesgo de los activos de información en uso. Esto se puede hacer de forma centralizada por un grupo de seguridad o por equipos de proyecto individuales a través de un cuestionario personalizado para obtener la información necesaria. Se debe establecer un proceso continuo para la clasificación de riesgo de aplicaciones y bienes de información. Esto, para asignar categorías a nuevos bienes y mantener actualizada la información existente al menos 2 veces por año.

#### B. Establecer y medir los objetivos de seguridad por cada clasificación

Con un esquema establecido de clasificación para el portafolio de aplicaciones de la organización, los objetivos de seguridad, y las decisiones del plan de implementación del programa de aseguramiento pueden hacerse más granulares. El plan de implementación del programa de aseguramiento debe ser modificado para tener en cuenta cada categoría de riesgo de las aplicaciones al hacer énfasis en Prácticas específicas para cada categoría. Para cada iteración del programa de aseguramiento, esto típicamente tomaría la forma de priorizar Objetivos de alto nivel de mayor riesgo en la capa de aplicación y progresivamente moverse hacia los Objetivos menos rigurosos para otras categorías o categorías menores. Este proceso establece la tolerancia al riesgo de la organización, ya que tienen que tomar decisiones activas en cuanto qué Objetivos específicos se esperan de las aplicaciones en cada categoría de riesgo. Al elegir mantener aplicaciones de menor riesgo en niveles menores de desempeño con respecto a las Prácticas de Seguridad, se ahorran recursos a cambio de la aceptación de un riesgo ponderado. Sin embargo, no es necesario construir arbitrariamente un plan de implementación separado para cada categoría de riesgo, ya que eso puede llevar a ineficiencia en la administración en el mismo programa de aseguramiento.

## Alinear los gastos de seguridad con indicadores de negocio pertinentes y el valor de los activos

### ACTIVIDADES

#### A. Realizar comparaciones de costo periódicas a nivel industria

Investigar y reunir información acerca de los costos de seguridad en foros de comunicación de la industria, analistas de negocio y firmas consultoras, u otras fuentes externas. En particular, hay algunos factores clave que necesitan ser identificados. Primero, utilice la información recolectada para identificar la cantidad de esfuerzo promedio que es aplicada por organizaciones similares en la industria en materia de seguridad. Esto puede hacerse ya sea de arriba hacia abajo con estimados del porcentaje total del presupuesto, ingresos, etc. O puede hacerse de abajo hacia arriba al identificar actividades relacionadas con seguridad que son consideradas normales para su tipo de organización. En general, esto puede ser difícil de medir para ciertas industrias, así que recolecte información de tantas fuentes relevantes como le sea posible. El siguiente objetivo al investigar los costos de seguridad, es determinar si hay ahorros potenciales en productos de seguridad de terceros y servicios que la organización actualmente use. Al considerar la decisión de cambiar proveedores, tenga en cuenta los costos ocultos, tales como re-entrenamiento de personal u otros esfuerzos adicionales del programa. En general, estos ejercicios de comparación de costos deben ser hechos al menos anualmente antes de la sesión estratégica del programa de aseguramiento. La comparación de información debe ser presentada a los involucrados en el negocio para alinear el programa de seguridad con el negocio.

#### B. Recolectar métricas históricas de gastos de seguridad

Recolectar información específica del proyecto sobre el costo de incidentes de seguridad anteriores. Por ejemplo, tiempo y dinero gastados en limpiar una fuga de información, pérdida monetaria por interrupciones del sistema, multas y cargos de las agencias reguladoras, gastos únicos de seguridad por proyecto para herramientas o servicios, etc. Utilizando las categorías de riesgo de aplicaciones y el respectivo plan de implementación de cada una dentro del programa de aseguramiento, se puede estimar un costo base de seguridad para cada aplicación que incluya los costos asociados con la categoría de riesgo correspondiente al proyecto. Combinar la información de costos específicos a la aplicación con el modelo de costo general basado en la categoría de riesgo, después evalúe proyectos en busca de valores extremos, es decir, sumas desproporcionadas con respecto a la evaluación de riesgo. Estos indican, ya sea un error en la evaluación/clasificación de riesgo o la necesidad de ajustar el programa de aseguramiento de la organización para abordar las causas raíz para un costo de seguridad más efectivo. El seguimiento de la inversión en seguridad por proyecto debe hacerse trimestralmente en la sesión estratégica del programa de aseguramiento, y la información debe ser revisada y evaluada por los involucrados en el proyecto al menos anualmente. Los valores atípicos y otros gastos imprevistos deben ser discutidos para un afecto potencial en el plan de implementación del programa de aseguramiento.

### EVALUACIÓN

- ◆ ¿Se recolectan datos por proyecto del costo de las actividades de aseguramiento?
- ◆ ¿La organización compara regularmente los gastos de seguridad contra los de otras organizaciones?

### RESULTADOS

- ◆ Información para tomar decisiones caso por caso sobre los gastos de seguridad
- ◆ Estimaciones de pérdidas pasadas debidas a problemas de seguridad
- ◆ Consideración por proyecto de gastos de seguridad contra pérdida potencial
- ◆ Devida diligencia a nivel industria referente a seguridad

### MÉTRICAS DE ÉXITO

- ◆ >80% de los proyectos reportando los costos de seguridad en los últimos 3 meses
- ◆ >1 comparación de costos a nivel industria en el último año
- ◆ >1 evaluación histórica de gastos de seguridad en el último año

### COSTOS

- ◆ Construir o comprar licencias sobre inteligencia en la industria sobre programas de seguridad
- ◆ Esfuerzo adicional del programa por estimaciones, seguimiento y evaluación

### PERSONAL

- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Auditor de seguridad (1 día/año)

### NIVELES RELACIONADOS

- ◆ Administración de vulnerabilidades - I

# Política y cumplimiento



	<b>PC 1</b>	<b>PC 2</b>	<b>PC 3</b>
<b>OBJETIVOS</b>	Entender los motivos relevantes para el gobierno de TI y cumplimiento de regulaciones para la organización	Establecer base de seguridad y cumplimiento, y entender los riesgos por proyecto	Exigir cumplimiento de regulaciones y medir a los proyectos conforme a las políticas y estándares de la organización
<b>ACTIVIDADES</b>	<p>A. Identificar y monitorear los indicadores externos de cumplimiento</p> <p>B. Crear y mantener lineamientos de cumplimiento</p>	<p>A. Crear políticas y estándares para seguridad y cumplimiento</p> <p>B. Establecer la práctica de auditoría de proyecto</p>	<p>A. Crear puntos de control de cumplimiento para proyectos</p> <p>B. Adoptar una solución para la recolección de datos de auditoría</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿La mayoría de los involucrados en el proyecto conocen el estado de cumplimiento del mismo?</li> <li>◆ ¿Son los requisitos de cumplimiento específicamente considerados por equipos de proyecto?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿La organización utiliza un conjunto de políticas y estándares para controlar el desarrollo de software?</li> <li>◆ Los equipos de proyecto ¿Son capaces de solicitar una auditoría de cumplimiento con políticas y estándares?</li> </ul>	<ul style="list-style-type: none"> <li>◆ Los proyectos ¿Son auditados periódicamente para asegurar una base de cumplimiento con políticas y estándares?</li> <li>◆ ¿La organización usa auditorías sistemáticas para recolectar y controlar evidencia de cumplimiento?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Mayor seguridad para manejar una auditoría de un tercero con resultados positivos</li> <li>◆ Alineación de recursos internos basado en la prioridad de los requisitos de cumplimiento</li> <li>◆ Descubrimiento oportuno de requisitos de regulación en evolución que afectan a la organización</li> </ul>	<ul style="list-style-type: none"> <li>◆ Concientización para los equipos de proyecto en cuanto a expectativas para seguridad y cumplimiento</li> <li>◆ Dueños de negocio que entiendan mejor los riesgos específicos de cumplimiento en las líneas de producto</li> <li>◆ Enfoque optimizado para alcanzar el cumplimiento eficientemente con mejoras de seguridad de oportunidad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Visibilidad a nivel organización de riesgos aceptados debido al no cumplimiento</li> <li>◆ Garantía concreta para el cumplimiento a nivel proyecto</li> <li>◆ Seguimiento preciso de la historia de cumplimiento pasada del proyecto</li> <li>◆ Proceso de auditoría eficiente, aprovechando herramientas para disminuir el esfuerzo manual</li> </ul>

## Entender los motivos relevantes para el gobierno de TI y cumplimiento de regulaciones para la organización

### ACTIVIDADES

#### A. Identificar y monitorear los indicadores externos de cumplimiento

Mientras que una organización puede tener una amplia variedad de requisitos de cumplimiento, esta actividad está específicamente orientada alrededor de aquellos que, directa o indirectamente, afectan la forma en que la organización construye o usa software y/o información. Aprovechando personal interno enfocado en cumplimiento en caso de estar disponible. Basándose en el negocio central de la organización, realice una investigación e identifique los estándares reguladores de terceros con los cuales se requiere el cumplimiento o son considerados como normas de la industria. Las posibilidades incluyen Sarbanes-Oxley Act (SOX), PCI-DSS, HIPAA, etc. Después de leer y entender cada estándar recolecte requisitos específicos relacionados a software e información, y construya una lista consolidada que asigne cada manejador (estándar de tercero) a cada uno de sus requisitos específicos de seguridad. En esta etapa, trate de limitar la cantidad de requisitos al eliminar todo lo considerado como opcional o solo recomendado. Como mínimo, realice investigación al menos bianualmente para asegurar que la organización está siendo actualizada en cuanto a cambios en los estándares de terceros. Dependiendo de la industria y la importancia del cumplimiento, esta actividad puede variar en esfuerzo y personal involucrado, pero siempre debe hacerse explícitamente.

#### B. Crear y mantener lineamientos de cumplimiento

Basándose en la lista consolidada de requisitos de software y datos relacionados con información de los motivos de cumplimiento, elabore una lista creando una respuesta correspondiente a cada requisito. Estas respuestas son llamadas, a veces, declaraciones de control, cada respuesta debe capturar el concepto de lo que hace la organización para asegurar que el requisito se cumple (o para notar por qué no aplica). La práctica de una auditoría típica a menudo involucra verificar una declaración de control a ver si es suficiente y después medir el desempeño de la organización contra la declaración de control misma, es importante que representen las prácticas organizacionales reales. Además, muchos requisitos pueden ser cumplidos al instituir elementos de proceso simples para cubrir el cumplimiento antes de evolucionar a la organización a mejores niveles de aseguramiento. Trabaje la lista consolidada e identifique las principales deficiencias para alimentar los esfuerzos de planeación futuros con respecto a la creación del programa de aseguramiento. Comunicar información acerca de deficiencias de cumplimiento con los involucrados en el proyecto para asegurar la concientización sobre los riesgos en el no cumplimiento. Como mínimo, actualice y revise las declaraciones de control con los involucrados en el proyecto al menos bianualmente. Dependiendo del número de indicadores de cumplimiento, sería prudente realizar las actualizaciones más a menudo.

### EVALUACIÓN

- ◆ ¿La mayoría de los involucrados en el proyecto conocen el estado de cumplimiento del mismo?
- ◆ ¿Son los requisitos de cumplimiento específicamente considerados por equipos de proyecto?

### RESULTADOS

- ◆ Mayor seguridad para manejar una auditoría de un tercero con resultados positivos
- ◆ Alineación de recursos internos basado en la prioridad de los requisitos de cumplimiento
- ◆ Descubrimiento oportuno de requisitos de regulación en evolución que afectan a la organización

### MÉTRICAS DE ÉXITO

- ◆ >1 sesión de descubrimiento de cumplimiento en los últimos 6 meses
- ◆ Lista de cumplimiento completada y actualizada en los últimos 6 meses
- ◆ >1 sesión de revisión de cumplimiento con los involucrados en el proyecto en los últimos 6 meses

### COSTOS

- ◆ Creación y mantenimiento continuo de la lista de cumplimiento

### PERSONAL

- ◆ Arquitectos (1 día/año)
- ◆ Administradores (2 días/año)
- ◆ Dueños de Negocio (1-2 días/año)

### NIVELES RELACIONADOS

- ◆ Estrategia y métricas - I



## Establecer base de seguridad y cumplimiento, y entender los riesgos por proyecto

### EVALUACIÓN

- ◆ ¿La organización utiliza un conjunto de políticas y estándares para controlar el desarrollo de software?
- ◆ Los equipos de proyecto ¿Son capaces de solicitar una auditoría de cumplimiento con políticas y estándares?

### RESULTADOS

- ◆ Concientización para los equipos de proyecto en cuanto a expectativas para seguridad y cumplimiento
- ◆ Dueños de negocio que entiendan mejor los riesgos específicos de cumplimiento en las líneas de producto
- ◆ Enfoque optimizado para alcanzar el cumplimiento eficientemente con mejoras de seguridad de oportunidad

### MÉTRICAS DE ÉXITO

- ◆ >75% del personal informado en políticas y estándares en los últimos 6 meses
- ◆ >80% de los involucrados deben estar concientes del estado de cumplimiento de políticas y estándares

### COSTOS

- ◆ Construcción o licencia de estándares internos
- ◆ Esfuerzo adicional por el proyecto para cumplimiento con estándares internos y auditorías

### PERSONAL

- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Auditores de Seguridad (2 días/proyecto/año)

### NIVELES RELACIONADOS

- ◆ Educación y orientación - 1 & 3
- ◆ Estrategia y métricas - 2
- ◆ Requisitos de seguridad - 1 & 3
- ◆ Arquitectura de seguridad - 3
- ◆ Análisis de código - 3
- ◆ Análisis de diseño - 3
- ◆ Fortalecimiento del ambiente - 3

### ACTIVIDADES

#### A. Crear políticas y estándares para seguridad y cumplimiento

Empezar con los lineamientos de cumplimiento actuales, revisar los estándares reguladores y notar cualquier requisito opcional o recomendado. Además, la organización debe hacer una pequeña investigación para descubrir cualquier cambio potencial en los requisitos de cumplimiento que sea relevante. Aumentar la lista con cualquier requisito adicional basándose en los indicadores de negocio. A menudo es más fácil consultar guías existentes proveídas al personal de desarrollo y obtener un conjunto de mejores prácticas. Agrupar requisitos comunes/similares y re-escribir cada grupo como declaraciones más generalizadas/simplificadas que cumplan con todos los manejadores de cumplimiento y que proporcionen más seguridad. Trabajar a través de este proceso con cada grupo para crear un conjunto de políticas internas y estándares que puedan ser asignadas directamente a manejadores de cumplimiento y mejores prácticas. Es importante para el conjunto de políticas y estándares que no contengan requisitos que son muy difíciles o costosos de ser cumplidos por los equipos. Una heurística útil es que aproximadamente el 80% de los proyectos debe ser capaz de cumplir con interrupción mínima. Esto requiere que un buen programa de comunicación sea establecido para anunciar las nuevas políticas/estándares y asistir a los equipos con cumplimiento en caso de ser necesario.

#### B. Establecer la práctica de auditoría de proyecto

Crear un proceso simple de auditoría para que los equipos soliciten y reciban una auditoría contra los estándares internos. Las auditorías son realizadas comúnmente por auditores de seguridad pero también pueden ser realizadas por personal de seguridad inteligente siempre que tengan conocimiento sobre los estándares internos. Basado en los indicadores de riesgo de negocio, los proyectos pueden ser priorizados concurrentemente en una lista de auditorías, de tal forma que el software de alto riesgo sea auditado antes o más frecuentemente. Adicionalmente, los proyectos de bajo riesgo pueden tener requisitos de auditoría internos menos estrictos para hacer la práctica de la auditoría más efectiva referente al costo. En general, cada proyecto activo debe tener una auditoría al menos bianualmente. Generalmente, las auditorías subsecuentes serán más simples de realizar si se mantiene información suficiente sobre la aplicación. Anunciar este servicio a los dueños de negocio y otros involucrados, para que ellos puedan solicitar una auditoría para sus proyectos. Los resultados detallados por solicitud de los estándares internos deben ser entregados a los involucrados para evaluación. Cuando sea práctico, los resultados de la auditoría pueden contener explicaciones de impacto y recomendaciones para remediación.



## Exigir cumplimiento de regulaciones y medir a los proyectos conforme a las políticas y estándares de la organización

### ACTIVIDADES

#### A. Crear puntos de control de cumplimiento para proyectos

Una vez que la organización ha establecido estándares internos para seguridad, el siguiente nivel de ejecución es establecer puntos particulares en el ciclo de vida del proyecto donde un proyecto no puede pasar hasta que sea auditado contra los estándares internos y esté en cumplimiento. Usualmente, la compuerta de cumplimiento es puesta en el punto de liberación de software de tal forma que no se permite publicar la versión del software hasta que el cumplimiento de regulaciones haya sido pasado. Es importante proporcionar suficiente tiempo para que se lleve a cabo la auditoría y la remediación, así que generalmente la auditoría debe iniciar antes, por ejemplo cuando la liberación es entregada a QA. A pesar de ser una compuerta firme de cumplimiento, el software antiguo u otros proyectos especializados pueden no ser capaces de cumplir, así que una aprobación de excepción debe ser creada. No más del 20% de los proyectos deben tener una excepción.

#### B. Adoptar una solución para la recolección de datos de auditoría

Las organizaciones que realizan auditorías periódicas generan una gran cantidad de información a lo largo del tiempo. La automatización debe ser utilizada para asistir en la recolección, administrar el cotejo para almacenamiento y recuperación, y para limitar el acceso individual a información sensible. Para muchos requisitos concretos de los estándares internos, las herramientas existentes como analizadores de código, herramientas de pruebas de intrusión, software de monitoreo, etc., pueden ser personalizadas y aprovechadas para automatizar las pruebas de cumplimiento contra los estándares internos. El objetivo de automatizar las validaciones de cumplimiento es mejorar la eficiencia de la auditoría así como permitir a más personal el auto-evaluarse antes de que una auditoría formal se lleve a cabo. Adicionalmente, las verificaciones automáticas son menos propensas a errores y permiten menor latencia en el descubrimiento de problemas. El almacenamiento de información debe permitir acceso centralizado a la información de las auditorías actuales e históricas por proyecto. Las soluciones automatizadas deben también proporcionar capacidades de control de acceso detallado para permitir el acceso solo a individuos aprobados con propósitos válidos de negocio para acceder a la información de la auditoría. Todas las instrucciones y procedimientos relacionados a acceder la información de cumplimiento así como solicitar privilegios de acceso deben ser anunciadas a los equipos. Algún tiempo adicional puede ser requerido inicialmente por los auditores de seguridad para introducir a los equipos de proyecto en este proceso.

### EVALUACIÓN

- ◆ Los proyectos ¿Son auditados periódicamente para asegurar una base de cumplimiento con políticas y estándares?
- ◆ ¿La organización usa auditorías sistemáticas para recolectar y controlar evidencia de cumplimiento?

### RESULTADOS

- ◆ Visibilidad a nivel organización de riesgos aceptados debido al no cumplimiento
- ◆ Garantía concreta para el cumplimiento a nivel proyecto
- ◆ Seguimiento preciso de la historia de cumplimiento pasada del proyecto
- ◆ Proceso de auditoría eficiente, aprovechando herramientas para disminuir el esfuerzo manual

### MÉTRICAS DE ÉXITO

- ◆ >80% de los proyectos en cumplimiento con políticas y estándares como se vio en la auditoría
- ◆ <50% del tiempo por auditoría, comparado con la auditoría manual

### COSTOS

- ◆ Construir o comprar licencias de herramientas para automatizar la auditoría contra estándares internos
- ◆ Mantenimiento continuo de puntos de control de auditoría y proceso de excepciones




### PERSONAL

- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)

### NIVELES RELACIONADOS

- ◆ Educación y orientación - 3
- ◆ Análisis de código - 2
- ◆ Pruebas de seguridad - 2

# Educación y orientación

			
<b>OBJETIVOS</b>	Ofrecer acceso al personal de desarrollo a recursos alrededor de los temas de programación segura e implementación	Educar a todo el personal en el ciclo de vida de software con lineamientos específicos en desarrollo seguro para cada rol	Hacer obligatorio el entrenamiento de seguridad integral y certificar al personal contra la base de conocimiento.
<b>ACTIVIDADES</b>	<p>A. Realizar entrenamiento técnico de concientización en seguridad</p> <p>B. Crear y mantener lineamientos técnicos</p>	<p>A. Realizar entrenamiento de seguridad en aplicaciones específico para cada rol</p> <p>B. Utilizar mentores de seguridad para mejorar los equipos</p>	<p>A. Crear un portal formal de soporte de seguridad en aplicaciones</p> <p>B. Establecer exámenes o certificaciones por rol</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿La mayoría de los desarrolladores han recibido entrenamiento de alto nivel sobre concientización de seguridad?</li> <li>◆ ¿Cada equipo tiene acceso a mejores prácticas y orientación para desarrollo seguro?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Se les ha dado entrenamiento específico y orientación a la mayoría de los roles en el proceso de desarrollo?</li> <li>◆ La mayoría de los involucrados en el proyecto, ¿Son capaces de obtener mentores de seguridad para usar en sus proyectos?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Los lineamientos relacionados con seguridad son controlados centralizadamente y distribuidos consistentemente a lo largo de la organización?</li> <li>◆ ¿La mayoría de la gente es evaluada para asegurar un conjunto de habilidades básicas para prácticas de desarrollo seguro?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Mejor concientización de los desarrolladores sobre los problemas más comunes a nivel código</li> <li>◆ Mantener software con mejores prácticas de seguridad elementales</li> <li>◆ Establecer lineamientos base para saber cómo llevar a cabo la seguridad entre el personal técnico</li> <li>◆ Habilitar verificaciones de seguridad cualitativas para una base de datos de conocimiento de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Concientización de extremo a extremo sobre los problemas que crean vulnerabilidades de seguridad en el producto, diseño y código</li> <li>◆ Construir planes para remediar vulnerabilidades y fallas de diseño en proyectos en curso</li> <li>◆ Habilitar controles de seguridad en las fases de requisitos, diseño y desarrollo</li> <li>◆ Mayor comprensión de los problemas de seguridad alientan a una planeación de seguridad más proactiva</li> </ul>	<ul style="list-style-type: none"> <li>◆ Remediación eficiente de vulnerabilidades en bases de código en curso y heredado</li> <li>◆ Entendimiento rápido y mitigación contra nuevos ataques y amenazas</li> <li>◆ Juzgue la comprensión de seguridad del personal y meda contra un estándar común</li> <li>◆ Establezca incentivos justos hacia la concientización de seguridad</li> </ul>

## Ofrecer acceso al personal de desarrollo a recursos alrededor de los temas de programación segura e implementación

### ACTIVIDADES

#### A. Realizar entrenamiento técnico de concientización en seguridad

Ya sea interna o externamente, lleve a cabo entrenamiento en seguridad para el personal técnico que cubra los principios básicos de seguridad en aplicaciones. Generalmente, esto se puede lograr vía instructor en 1 o 2 días o con entrenamiento basado en computadora con módulos teniendo la misma cantidad de tiempo por desarrollador. El contenido del curso debe cubrir información tanto conceptual como técnica. Temas apropiados incluyen mejores prácticas de alto nivel para la validación de entradas, codificación de salida, manejo de errores, registro, autenticación, autorización. La cobertura adicional de vulnerabilidades de software comunes también es deseable como una lista de los 10 problemas mayores relacionados al software que está siendo desarrollado (aplicaciones Web, dispositivos embebidos, aplicaciones cliente-servidor, etc.). Cuando sea posible, usar muestras de código y ejercicios de laboratorio en el lenguaje de programación específico aplicable a la compañía. Para desplegar dicho entrenamiento, es recomendado exigir entrenamiento de seguridad anual y después tener cursos (ya sea por instructor o computadora) con la frecuencia necesaria basándose en el número de desarrolladores.

#### B. Crear y mantener lineamientos técnicos

Para el personal de desarrollo, reúna una lista de documentos aprobados, sitios Web, y notas técnicas que proporcionen consejos de seguridad específicos a la tecnología. Estas referencias pueden ser reunidas de muchos recursos públicos en Internet. En casos donde las tecnologías sean muy especializadas o propietarias estén presentes, utilice personal experto en seguridad para crear notas a lo largo del tiempo para crear una base de conocimientos. Asegúrese que la administración esté conciente de los recursos e informe al personal acerca del uso esperado. Trate de mantener los lineamientos ligeros y actualizados para evitar desorden e irrelevancia. Una vez que se ha establecido un nivel de confort, pueden ser usados como una lista de verificación cualitativa para asegurar que los lineamientos han sido leídos, entendidos y seguidos en el proceso de desarrollo.

### EVALUACIÓN

- ◆ ¿La mayoría de los desarrolladores han recibido entrenamiento de alto nivel sobre concientización de seguridad?
- ◆ ¿Cada equipo tiene acceso a mejores prácticas y orientación para desarrollo seguro?

### RESULTADOS

- ◆ Mejor concientización de los desarrolladores sobre los problemas más comunes a nivel código
- ◆ Mantener software con mejores prácticas de seguridad elementales
- ◆ Establecer lineamientos base para saber cómo llevar a cabo la seguridad entre el personal técnico
- ◆ Habilitar verificaciones de seguridad cualitativas para una base de datos de conocimiento de seguridad

### MÉTRICAS DE ÉXITO

- ◆ >50% de los desarrolladores informados en problemas de seguridad el último año
- ◆ >75% de los desarrolladores expertos/arquitectos informados en problemas de seguridad el último año
- ◆ Lanzar orientación técnica dentro de los 3 meses del primer entrenamiento

### COSTOS

- ◆ Construcción del curso o licencia
- ◆ Mantenimiento continuo de la orientación técnica

### PERSONAL

- ◆ Desarrolladores (1-2 días/año)
- ◆ Arquitectos (1-2 días/año)

### NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2
- ◆ Requisitos de seguridad - 1
- ◆ Arquitectura de seguridad - 1



## Educar a todo el personal en el ciclo de vida de software con lineamientos específicos en desarrollo seguro para cada rol

### EVALUACIÓN

- ◆ ¿Se les ha dado entrenamiento específico y orientación a la mayoría de los roles en el proceso de desarrollo?
- ◆ La mayoría de los involucrados en el proyecto, ¿son capaces de obtener mentores de seguridad para usar en sus proyectos?

### RESULTADOS

- ◆ Concientización de extremo a extremo sobre los problemas que crean vulnerabilidades de seguridad en el producto, diseño y código
- ◆ Construir planes para remediar vulnerabilidades y fallas de diseño en proyectos en curso
- ◆ Habilitar controles de seguridad en las fases de requisitos, diseño y desarrollo
- ◆ Mayor comprensión de los problemas de seguridad alientan a una planeación de seguridad más proactiva

### MÉTRICAS DE ÉXITO

- ◆ >60% del personal de desarrollo entrenado en el último año
- ◆ >50% del personal de administración/ análisis entrenado en el último año
- ◆ >80% de desarrolladores expertos/ arquitectos entrenados en el último año
- ◆ >3.0 en la escala de Likert en la utilidad de los cursos de entrenamiento

### COSTOS

- ◆ Elaboración o compra de licencia de una biblioteca de entrenamiento
- ◆ Personal experto en seguridad para entrenamiento

### PERSONAL

- ◆ Desarrolladores (2 días/año)
- ◆ Arquitectos (2 días/año)
- ◆ Administradores (1-2 días/año)
- ◆ Dueños de Negocio (1-2 días/año)
- ◆ Testadores de Calidad (1-2 días/año)
- ◆ Auditores de Seguridad (1-2 días/año)

### NIVELES RELACIONADOS

- ◆ Administración de vulnerabilidades - 1
- ◆ Análisis de diseño - 2
- ◆ Arquitectura de seguridad - 2

### ACTIVIDADES

#### A. Realizar entrenamiento de seguridad en aplicaciones específico para cada rol

Realizar entrenamiento de seguridad que se enfoque en la seguridad en aplicaciones aplicable a cada rol. Generalmente, esto se puede lograr vía instructor en 1 o 2 días o con entrenamiento basado en computadora con diferentes módulos que tengan la misma cantidad de tiempo por persona. Para administradores y los que especifican los requisitos de seguridad, el contenido del curso debe hablar sobre la planeación de requisitos de seguridad, manejo de vulnerabilidades e incidentes, modelado de amenazas, y diseño de casos para uso indebido/abuso. El entrenamiento para los testadores y auditores debe enfocarse en entrenar al personal para entender y analizar más efectivamente el software en busca de problemas relevantes a la seguridad. Como tal, debe contener técnicas para el análisis de código, arquitectura y análisis de diseño, análisis de tiempo de ejecución y planeación efectiva de pruebas de seguridad. Expandir el entrenamiento técnico dirigiéndose a desarrolladores y arquitectos para incluir otros temas relevantes tales como patrones de diseño seguros, entrenamiento específico para herramientas, modelado de amenazas y técnicas de auditorías de software. Para desplegar dicho entrenamiento, se recomienda exigir entrenamiento de seguridad anual y entrenamiento especializado periódico. Los cursos deben estar disponibles (ya sea por instructor o computadora) con la frecuencia necesaria basándose en el número de personas por rol.

#### B. Utilizar mentores de seguridad para mejorar los equipos

Usando expertos, ya sea internos o externos, haga que el personal experto en seguridad esté disponible para consultas. Además, este recurso de ayuda debe ser anunciado internamente para asegurarse que el personal esté conciente de su disponibilidad. El personal técnico puede ser creado al reclutar individuos experimentados en la organización para usar un 10% de su tiempo para actividades de consulta. Los mentores deben comunicarse entre ellos para asegurarse que están concientes del área de especialización de cada uno de ellos y dirigir las preguntas de acuerdo a ello. Mientras los mentores pueden ser usados en cualquier punto del ciclo de vida de software, los momentos apropiados para usarlos incluyen la concepción inicial del producto, antes de terminar los aspectos funcionales, cuando surjan problemas durante el desarrollo, planeación de pruebas, y cuando ocurran incidentes de seguridad operativa. A lo largo del tiempo, la red interna de recursos de entrenamiento puede ser usado como puntos de contacto para comunicar información relevante a seguridad a lo largo de la organización así como ser recursos locales que tengan más familiaridad con los equipos de proyecto en curso que la que un equipo centralizado podría tener.

## Hacer obligatorio el entrenamiento de seguridad integral y certificar al personal contra la base de conocimiento.

### ACTIVIDADES

#### A. Crear un portal formal de soporte de seguridad en aplicaciones

Construir sobre recursos escritos de temas relevantes a seguridad de aplicaciones, crear y promocionar un repositorio centralizado (usualmente un sitio Web interno). Los lineamientos mismos pueden ser creados en cualquier forma que tenga sentido para la organización, pero debe establecerse una junta de aprobación y procesos de control para cambios directos. Más allá del contenido estático en forma de listas de mejores prácticas, guías específicas para herramientas, preguntas frecuentes, y otros artículos, el portal debe contener componentes interactivos como listas de correo, foros o wikis para permitir que los recursos internos tengan comunicación cruzada en cuanto a temas relevantes de seguridad y tengan la información catalogada para referencia futura. El contenido debe ser catalogado y fácil de buscar basándose en varios factores comunes como plataforma, lenguaje de programación, bibliotecas o marcos de trabajo de terceros, etapas del ciclo de vida, etc. Los equipos creando software deben alinearse tempranamente a sí mismos a los lineamientos específicos que van a seguir durante el desarrollo del producto. En auditorías de producto, la lista de lineamientos aplicables y discusiones relacionadas con el producto deben ser usadas como criterio de auditoría.

#### B. Establecer exámenes o certificaciones por rol

Ya sea por rol o entrenamiento por módulo/clase, cree y administre exámenes de aptitud que evalúen a las personas por comprensión y utilización del conocimiento de seguridad. Típicamente, los exámenes deben ser creados basados en roles y establecer un resultado mínimo de aprobación cerca del 75% correcto. Mientras que al personal le debe ser requerido tomar el entrenamiento aplicable o actualizarse en los cursos anualmente, los exámenes de certificación deben ser requeridos mínimo bianualmente. Basándose en criterio de pasó/falló o desempeño excepcional, el personal debe ser calificado en capas de tal forma que otras actividades relacionadas con seguridad pudieran requerir individuos de un nivel particular de certificación que apruebe antes de que la actividad sea completada, por ejemplo un desarrollador no certificado no puede pasar un diseño a implementación sin la aprobación explícita de un arquitecto certificado. Esto proporciona visibilidad por proyecto para dar seguimiento a decisiones de seguridad con responsabilidad individual. En general, esto proporciona una base para recompensar o sancionar al personal por tomar buenas decisiones de negocio acerca de seguridad de aplicación.

### EVALUACIÓN

- ◆ ¿Los lineamientos relacionados con seguridad son controlados centralizadamente y distribuidos consistentemente a lo largo de la organización?
- ◆ ¿La mayoría de la gente es evaluada para asegurar un conjunto de habilidades básicas para prácticas de desarrollo seguro?

### RESULTADOS

- ◆ Remediación eficiente de vulnerabilidades en bases de código en curso y heredado
- ◆ Entendimiento rápido y mitigación contra nuevos ataques y amenazas
- ◆ Juzgue la comprensión de seguridad del personal y meda contra un estándar común
- ◆ Establezca incentivos justos hacia la concientización de seguridad

### MÉTRICAS DE ÉXITO

- ◆ >80% del personal certificado en el año anterior

### COSTOS

- ◆ Crear o comprar licencias de evaluación para certificación
- ◆ Mantenimiento continuo y control de cambios para el portal de seguridad de aplicaciones
- ◆ Recursos humanos y esfuerzo adicional para implementar certificación de los empleados

### PERSONAL

- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Testadores de Calidad (1 día/año)
- ◆ Auditores de Seguridad (1 día/año)

### NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2 & 3

# Evaluación de amenaza

	 <b>TA 1</b>	 <b>TA 2</b>	 <b>TA 3</b>
<b>OBJETIVOS</b>	Identificar y comprender las amenazas de alto nivel para la organización y los proyectos individuales	Aumentar la precisión de la evaluación de amenazas y mejorar la granularidad de la comprensión por proyecto	Comparar concretamente controles de compensación a cada amenaza contra el software interno y de terceros
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Desarrollar y mantener modelos de amenaza específicos a cada aplicación</li> <li>B. Elabore perfil de atacante desde la arquitectura de software</li> </ul>	<ul style="list-style-type: none"> <li>A. Desarrollar y mantener modelos de casos de abuso por proyecto</li> <li>B. Adoptar un sistema de ponderación para la medición de las amenazas</li> </ul>	<ul style="list-style-type: none"> <li>A. Evaluar explícitamente el riesgo de los componentes de terceros</li> <li>B. Elaboración de modelos de amenaza con controles de compensación</li> </ul>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?</li> <li>◆ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?</li> </ul>	<ul style="list-style-type: none"> <li>◆ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?</li> <li>◆ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?</li> <li>◆ ¿Los equipos de proyectos analizan regularmente los requisitos funcionales para descubrir probables abusos?</li> </ul>	<ul style="list-style-type: none"> <li>◆ Los equipos de proyecto, ¿Consideran específicamente los riesgos derivados de el software externo?</li> <li>◆ ¿Todos los mecanismos de protección y control son registrados y comparados con las amenazas?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Comprensión de alto nivel de los factores que pueden conducir a resultados negativos</li> <li>◆ Mayor conciencia de las amenazas entre los equipos de proyecto</li> <li>◆ Inventario de las amenazas para su organización</li> </ul>	<ul style="list-style-type: none"> <li>◆ Comprensión granular de posibles amenazas a proyectos individuales</li> <li>◆ Marco para tomar mejores decisiones de compensación dentro de los equipos de proyecto</li> <li>◆ Capacidad de priorizar los esfuerzos de desarrollo dentro de un equipo de proyecto sobre la base de ponderación de riesgo</li> </ul>	<ul style="list-style-type: none"> <li>◆ Examen más profundo del perfil de riesgo total para cada proyecto de software</li> <li>◆ Comparación detallada de las características de aseguramiento contra las amenazas establecidas para cada proyecto de software</li> <li>◆ Artefactos para documentar la debida diligencia basándose en la función de negocio de cada proyecto de software</li> </ul>

# Evaluación de amenaza



Identificar y comprender las amenazas de alto nivel para la organización y los proyectos individuales

## ACTIVIDADES

### A. Desarrollar y mantener modelos de amenaza específicos a cada aplicación

Basándose puramente en el objetivo comercial de cada proyecto de software y el perfil de riesgo del negocio (si está disponible), identifique probables escenarios catastróficos para el software en desarrollo en cada equipo de proyecto. Esto puede llevarse a cabo utilizando árboles de ataque simples o por medio de un proceso de modelado de amenazas más formal tal como STRIDE de Microsoft, Trike, etc. Para construir los árboles de ataque, identifique cada uno de los escenarios catastróficos en una sola frase y etiquételos como los objetivos de alto nivel de un atacante. De cada objetivo de atacante identificado, identifique las condiciones preestablecidas que se deben tener para cada objetivo a realizar. Esta información debe ser registrada en las ramas debajo de cada objetivo en el que cada rama es o un operador lógico AND o un OR lógico de las declaraciones que figuran debajo. Una rama AND indica que cada uno de los nodos secundarios que dependerá directamente debe ser verdad a fin de realizar el nodo padre. Una rama OR indica que cualquiera de los nodos secundarios que dependen directamente debe ser verdad para alcanzar el nodo padre. Independientemente de el tipo de modelado de amenazas, revise cada uno de requisitos funcionales tanto históricos como actuales, para aumentar el árbol de ataque e indicar los fallos de seguridad pertinentes para cada uno. Haga una lluvia de ideas para diseccionar de forma iterativa cada escenario de fracaso en todas las formas posibles en que un atacante podría ser capaz de alcanzar cada uno de los objetivos. Después de la creación inicial, el modelo de amenazas de una aplicación debe ser actualizado cuando se produzcan cambios significativos en el software. Esta evaluación debe llevarse a cabo con los desarrolladores senior y arquitectos, así como uno o más auditores de seguridad.

### B. Elabore perfil de atacante desde la arquitectura de software

Inicialmente, realice una evaluación para identificar todas las amenazas probables en la organización en base a los proyectos de software. Para esta evaluación, limite las amenazas a los agentes de mal intencionados y omita otros riesgos, tales como las vulnerabilidades conocidas, las debilidades potenciales, etc. Por lo general, puede iniciar teniendo en cuenta los agentes externos y sus motivaciones para atacar. A esta lista, agregue personal interno que podrían causar daños y sus motivaciones para atacar. Basado en la arquitectura del proyecto(s) de software en estudio, puede ser más eficiente llevar a cabo este análisis una vez por cada tipo de arquitectura en lugar de para cada proyecto por separado, ya que las aplicaciones de la arquitectura y el objetivo comercial, generalmente serán susceptibles a amenazas similares. Esta evaluación debe llevarse a cabo con los dueños de negocio y otras partes interesadas, pero también incluir uno o más auditores de seguridad para una perspectiva adicional sobre las amenazas. Al final, el objetivo es tener una lista concisa de los agentes de amenaza y sus motivaciones para el ataque.

## EVALUACIÓN

- ◆ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?
- ◆ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?

## RESULTADOS

- ◆ Comprensión de alto nivel de los factores que pueden conducir a resultados negativos
- ◆ Mayor conciencia de las amenazas entre los equipos de proyecto
- ◆ Inventario de las amenazas para su organización

## MÉTRICAS DE ÉXITO

- ◆ >50% de los participantes en el proyecto informados sobre los modelos de la amenaza de los proyectos importantes para los últimos 12 meses
- ◆ >75% de los participantes en el proyecto informados sobre los perfiles de atacante para las arquitecturas relevantes para la organización

## COSTOS

- ◆ Construcción y mantenimiento de los artefactos del proyecto para los modelos de amenaza

## PERSONAL

- ◆ Dueños de Negocio (1 día/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Auditores de Seguridad (2 días/año)
- ◆ Administradores (1 día/año)

## NIVELES RELACIONADOS

- ◆ Estrategia y métricas - 1
- ◆ Requisitos de seguridad - 2



## TA 2

# Evaluación de amenaza

Aumentar la precisión de la evaluación de amenazas y mejorar la granularidad de la comprensión por proyecto

### EVALUACIÓN

- ◆ La mayoría de los proyectos de su organización, ¿considera y documenta probables amenazas?
- ◆ ¿Su organización comprende y documenta los tipos de atacantes a los que se enfrenta?
- ◆ ¿Los equipos de proyectos analizan regularmente los requisitos funcionales para descubrir probables abusos?

### RESULTADOS

- ◆ Comprensión granular de posibles amenazas a proyectos individuales
- ◆ Marco para tomar mejores decisiones de compensación dentro de los equipos de proyecto
- ◆ Capacidad de priorizar los esfuerzos de desarrollo dentro de un equipo de proyecto sobre la base de ponderación de riesgo

### MÉTRICAS DE ÉXITO

- ◆ >75% de los equipos de proyecto con amenazas identificadas y evaluadas
- ◆ >75% de los involucrados en el proyecto informados sobre los modelos de amenaza y abuso en los proyectos relevantes para los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional del proyecto para el mantenimiento de modelos de amenazas y perfiles de atacantes

### PERSONAL

- ◆ Auditor de seguridad (1 día/año)
- ◆ Dueño del Negocio (1 día/año)
- ◆ Administradores (1 día/año)

### NIVELES RELACIONADOS

- ◆ Estrategia y métricas - 2
- ◆ Arquitectura de seguridad - 2

### ACTIVIDADES

#### A. Desarrollar y mantener modelos de casos de abuso por proyecto

Considerar además las amenazas a la organización, realizar un análisis más formal para determinar la posible utilización indebida o abuso de funciones. Normalmente, este proceso se inicia con la identificación de escenarios de uso normal, por ejemplo, los diagramas de caso de uso, si están disponibles. Si una técnica de caso de abuso formal no se utiliza, generar un conjunto de casos de abuso para cada escenario, comenzando con una declaración de uso normal y formas de intercambio de ideas en que la declaración podría ser negada, como un todo o en parte. La forma más sencilla de comenzar es insertar la palabra "no" en la declaración de uso de tantas maneras como sea posible, normalmente en torno a nombres y verbos. Cada escenario de uso debería generar varias declaraciones de caso de abuso. Siga trabajando en las declaraciones de casos de abuso para incluir cualquier preocupación específica a la aplicación basándose en la función de negocio del software. El objetivo final es que el conjunto completo de las declaraciones de abuso formen un modelo de patrones de uso que deben ser rechazadas por el software. Si se desea, estos casos de abuso pueden ser combinados con los modelos de la amenaza existentes. Después de la creación inicial, los modelos de casos de abuso deben ser actualizados para los proyectos activos durante la fase de diseño. Para los proyectos existentes, los nuevos requisitos deben ser analizados para determinar los posibles abusos, y los proyectos existentes de forma oportunista deberían construir los casos de abuso de funciones establecidas donde sea práctico.

#### B. Adoptar un sistema de ponderación para la medición de las amenazas

Con base en los perfiles de agresor establecidos, identifique un sistema de clasificación para permitir la comparación relativa entre las amenazas. Inicialmente, esto puede ser una simple clasificación alto-medio-bajo basada en el riesgo comercial, pero cualquier escala puede utilizarse siempre que no haya más de 5 categorías. Después de la identificación de un sistema de clasificación, desarrolle criterios de evaluación que permitan a cada amenaza tener una puntuación. Con el fin de hacer esto correctamente, factores adicionales acerca de cada amenaza deben ser considerados, más allá de la motivación. Los factores importantes incluyen el capital y los recursos humanos, privilegio de acceso inherente, la capacidad técnica, las metas pertinentes sobre el(los) modelo(s) de amenaza, la probabilidad de un ataque exitoso, etc. Después de asignar a cada amenaza a una calificación, utilice esta información para priorizar las actividades de mitigación del riesgo en el ciclo de vida del desarrollo. Una vez construido por el equipo de proyecto, debe ser actualizado durante el diseño de nuevas funciones o actividades que modifiquen el código.



# Evaluación de amenaza



Comparar concretamente controles de compensación a cada amenaza contra el software interno y de terceros

## ACTIVIDADES

### A. Evaluar explícitamente el riesgo de los componentes de terceros

Realizar una evaluación de su código de software e identificar los componentes que son de origen externo. Normalmente, estos incluyen proyectos de código abierto, el software empaquetado (COTS por sus siglas en inglés) adquirido, y servicios en línea que utiliza su software. Para cada componente identificado, elaborar perfiles de atacantes para cada proyecto de software basándose en el daño potencial de los componentes de terceros. Con base en los perfiles de atacante recientemente identificados, actualizar los modelos de amenazas de software para incorporar los posibles riesgos basados en los nuevos objetivos o capacidades de atacante. Además de los escenarios de amenaza, también examinar las formas en que las vulnerabilidades o errores de diseño en el software de terceros puedan afectar a su código y diseño. Elaborar los modelos amenaza en consecuencia, con los riesgos potenciales de las vulnerabilidades y el conocimiento del perfil actualizado del atacante. Después de haberlo realizado, inicialmente para un proyecto, este debe ser actualizado y revisado durante la fase de diseño o en cada ciclo de desarrollo. Esta actividad debe ser realizada por un auditor de seguridad con los interesados pertinentes, técnicos y de negocio.

### B. Elaboración de modelos de amenaza con controles de compensación

Realizar una evaluación para identificar formalmente los factores que evitan directamente las condiciones necesarias para la transacción representada por los modelos de amenaza. Estos factores atenuantes son los controles de compensación que, formalmente, hacen frente a los riesgos directos del software. Los factores pueden ser las características técnicas del software en sí, pero también pueden ser elementos del proceso en el ciclo de vida del desarrollo, características de infraestructura, etc. Si los árboles de ataque se están usando, la relación lógica que representa cada rama será un AND o un OR. Por lo tanto, al mitigar contra sólo una condición previa en una rama AND, los padres y todos los nodos hoja conectados pueden ser marcados como mitigados. Sin embargo, todos los nodos hijos de un nodo OR deben evitarse antes de que el padre puede ser marcado como mitigado. Independientemente de la técnica de modelado de amenazas, identifique los controles de compensación y haga anotaciones directamente en los modelos de la amenaza. El objetivo es maximizar la cobertura en términos de controles y que se marquen las partes del modelo de amenazas como resueltas. Para cualquier camino viable restante, identifique los posibles controles de compensación para la retroalimentar la estrategia corporativa. Después de realizado, inicialmente para un proyecto, este debe ser actualizado y revisado durante la fase de diseño o de cada ciclo de desarrollo. Esta actividad debe ser realizada por un auditor de seguridad los interesados pertinentes, técnicos y de negocio.

## EVALUACIÓN

- ◆ Los equipos de proyecto, ¿Consideran específicamente los riesgos derivados de el software externo?
- ◆ ¿Todos los mecanismos de protección y control son registrados y comparados con las amenazas?

## RESULTADOS

- ◆ Examen más profundo del perfil de riesgo total para cada proyecto de software
- ◆ Comparación detallada de las características de aseguramiento contra las amenazas establecidas para cada proyecto de software
- ◆ Artefactos para documentar la debida diligencia basándose en la función de negocio de cada proyecto de software

## MÉTRICAS DE ÉXITO

- ◆ >80% de los equipos de proyecto con los modelos de amenazas actualizados antes de cada ciclo de aplicación
- ◆ >80% de los equipos de proyecto con el inventario actualizado de los componentes de terceros antes de cada lanzamiento
- ◆ >50% de todos los incidentes de seguridad identificados a priori por los modelos de amenazas en los últimos 12 meses

## COSTOS

- ◆ Esfuerzo adicional del proyecto para el mantenimiento de modelos de amenazas detallados y perfiles atacante ampliados
- ◆ Descubrimiento de todas las dependencias de terceros



## PERSONAL

- ◆ Dueños de Negocio (1 día/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Auditores de Seguridad (2 días/año)
- ◆ Administradores (1 día/año)

## NIVELES RELACIONADOS

- ◆ Requisitos de seguridad - 2 & 3

# Requisitos de seguridad

	 <b>SR 1</b>	 <b>SR 2</b>	 <b>SR 3</b>
<b>OBJETIVOS</b>	Considerar explícitamente la seguridad durante el procesamiento de captura de requisitos de software	Aumentar la granularidad de los requisitos de seguridad derivados de la lógica de negocio y riesgos conocidos	Exigir que se siga el proceso de requisitos de seguridad para todos los proyectos de software y dependencias de terceros
<b>ACTIVIDADES</b>	<p>A. Deducir los requisitos de seguridad a partir de la funcionalidad de negocios</p> <p>B. Evaluar la seguridad y los lineamientos de cumplimiento para regulaciones de los requisitos</p>	<p>A. Generar una matriz de control de acceso a los recursos y capacidades</p> <p>B. Especificar los requisitos de seguridad en base a los riesgos conocidos</p>	<p>A. Incorporar los requisitos de seguridad a acuerdos con proveedores</p> <p>B. Ampliar el programa de auditoría para los requisitos de seguridad</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ La mayoría de los equipos de proyecto, ¿especifican algunos requisitos de seguridad durante el desarrollo?</li> <li>◆ ¿Obtienen los equipos de proyecto los requisitos de las mejores prácticas y guías de cumplimiento?</li> </ul>	<ul style="list-style-type: none"> <li>◆ La mayoría de los interesados ¿Revisan las matrices de control de acceso para los proyectos importantes?</li> <li>◆ ¿Están los equipos de proyecto especificando los requisitos basándose en la retroalimentación de otras actividades de seguridad?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están la mayoría de los interesados revisando los acuerdos con proveedores para los requisitos de seguridad?</li> <li>◆ Los requisitos de seguridad ¿son especificados por los equipos de proyecto que están siendo auditados?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Alineación de alto nivel de los esfuerzos de desarrollo con los riesgos de negocio</li> <li>◆ Captura ad hoc de las mejores prácticas de seguridad de la industria como requisitos explícitos</li> <li>◆ Concientización entre los interesados de las medidas adoptadas para mitigar el riesgo de software</li> </ul>	<ul style="list-style-type: none"> <li>◆ Conocimiento detallado de los escenarios de ataque contra la lógica de negocios</li> <li>◆ Esfuerzo de desarrollo prioritario para las características de seguridad basadas en los ataques probables</li> <li>◆ Decisiones más educadas para compensar entre las características y los esfuerzos de seguridad</li> <li>◆ Las partes interesadas pueden evitar de mejor forma los requisitos funcionales que tienen fallas de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Sentar formalmente las bases para las expectativas de seguridad de código externo</li> <li>◆ Información centralizada sobre los esfuerzos de seguridad emprendidas por cada equipo de proyecto</li> <li>◆ Habilidad para alinear los recursos a los proyectos basándose en el riesgo de aplicación y los requisitos de seguridad deseados</li> </ul>

# Requisitos de seguridad



SR 1

Considerar explícitamente la seguridad durante el procesamiento de captura de requisitos de software

## ACTIVIDADES

### A. Deducir los requisitos de seguridad a partir de la funcionalidad de negocios

Lleve a cabo una revisión de los requisitos funcionales que especifican la lógica de negocio y el comportamiento general de cada proyecto de software. Después de reunir los requisitos para un proyecto, realice una evaluación para obtener los requisitos de seguridad pertinentes. Incluso si el software está siendo construido por un tercero, estos requisitos, una vez identificados, deben incluirse en los requisitos funcionales entregados a los vendedores. Para cada requisito funcional, un auditor de seguridad debe llevar a los involucrados en el desarrollo por el proceso de señalar de manera explícita cualquier expectativa con respecto a la seguridad. Normalmente, las preguntas a aclarar para cada requisito incluyen las expectativas de seguridad de datos, control de acceso, integridad de la transacción, la criticidad de la función empresarial, la separación de funciones, tiempo de actividad, etc. Es importante asegurarse de que todos los requisitos de seguridad siguen los mismos principios generales para escribir buenos requisitos. En concreto, deben ser específicos, medibles, y razonables. Realizar este proceso para todos los nuevos requisitos en proyectos activos. Para las características existentes, se recomienda llevar a cabo el mismo proceso como un análisis de las carencias, para potenciar la reutilización en el futuro de nuevos factores de seguridad.

### B. Evaluar la seguridad y los lineamientos de cumplimiento para regulaciones de los requisitos

Determinar las mejores prácticas de la industria que los equipos de proyecto debe tratar como requisitos. Estos pueden ser seleccionados de las lineamientos a disposición del público, las lineamientos / standards / políticas internas o externas, o el cumplimiento de los requisitos establecidos. Es importante no tratar de introducir muchas más mejores prácticas en cada iteración del desarrollo ya que existe un impacto en el tiempo del diseño e implementación. El enfoque recomendado es añadir lentamente las mejores prácticas en ciclos sucesivos de desarrollo para reforzar perfil de seguridad global del software en el tiempo. Para los sistemas existentes, agregar factores de mejores prácticas de seguridad puede ser una tarea compleja.

Cuando sea posible, añada los requisitos de seguridad de manera oportunista cuando se añaden nuevas características. Como mínimo, la realización de los análisis para identificar las mejores prácticas de aplicación se debe hacer para ayudar a estimular los esfuerzos futuros de planeación.

Esta revisión debe ser realizada por un auditor de seguridad con la participación de los interesados de negocios.

Desarrolladores senior, arquitectos, técnicos y otros interesados también deben participar para brindar conocimientos de diseño e implementación en el proceso de decisión.

## EVALUACIÓN

- ◆ La mayoría de los equipos de proyecto, ¿especifican algunos requisitos de seguridad durante el desarrollo?
- ◆ ¿Obtienen los equipos de proyecto los requisitos de las mejores prácticas y guías de cumplimiento?

## RESULTADOS

- ◆ Alineación de alto nivel de los esfuerzos de desarrollo con los riesgos de negocio
- ◆ Captura ad hoc de las mejores prácticas de seguridad de la industria como requisitos explícitos
- ◆ Concientización entre los interesados de las medidas adoptadas para mitigar el riesgo de software

## MÉTRICAS DE ÉXITO

- ◆ >50% de los equipos de proyecto con los requisitos de seguridad definidos explícitamente

## COSTOS

- ◆ Esfuerzo adicional del proyecto producto de la adición de los requisitos de seguridad para cada ciclo de desarrollo

## PERSONAL

- ◆ Auditor de seguridad (2 días/año)
- ◆ Dueño del Negocio (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Arquitectos (1 día/año)

## NIVELES RELACIONADOS

- ◆ Educación y orientación - 1
- ◆ Política y cumplimiento - 2
- ◆ Análisis de diseño - 1
- ◆ Análisis de código - 1
- ◆ Pruebas de seguridad - 1



## SR 2

# Requisitos de seguridad

Aumentar la granularidad de los requisitos de seguridad derivados de la lógica de negocio y riesgos conocidos

### EVALUACIÓN

- ◆ La mayoría de los interesados ¿Revisan las matrices de control de acceso para los proyectos importantes?
- ◆ ¿Están los equipos de proyecto especificando los requisitos basándose en la retroalimentación de otras actividades de seguridad?

### RESULTADOS

- ◆ Conocimiento detallado de los escenarios de ataque contra la lógica de negocios
- ◆ Esfuerzo de desarrollo prioritario para las características de seguridad basadas en los ataques probables
- ◆ Decisiones más educadas para compensar entre las características y los esfuerzos de seguridad
- ◆ Las partes interesadas pueden evitar de mejor forma los requisitos funcionales que tienen fallas de seguridad

### MÉTRICAS DE ÉXITO

- ◆ >75% de todos los proyectos con modelos de casos de abuso actualizados dentro de los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional del proyecto a partir de la construcción y mantenimiento de modelos de casos de abuso

### PERSONAL

- ◆ Auditor de seguridad (2 días/año)
- ◆ Administradores (1 día/año)
- ◆ Arquitectos (2 días/año)
- ◆ Dueños de Negocio (1 día/año)

### NIVELES RELACIONADOS

- ◆ Evaluación de amenaza - 1 & 3
- ◆ Estrategia y métricas - 1

### ACTIVIDADES

#### A. Generar una matriz de control de acceso a los recursos y capacidades

Basándose en el objetivo comercial de la aplicación, identificar funciones de usuario y de operador. Además, construir una lista de recursos y las capacidades a través de la recopilación de todos los activos de datos y funciones específicas de aplicación que están protegidos por alguna forma de control de acceso. En una matriz simple con funciones en un eje y los recursos en el otro, considere las relaciones entre cada función y cada uno de los recursos y haga constar en cada intersección la conducta correcta del sistema en términos de control de acceso de acuerdo con los involucrados en el proceso. Para los recursos de datos, es importante tener en cuenta los derechos de acceso en términos de creación, el acceso de lectura, actualización y supresión. Para los recursos que son características, el registro de los derechos de acceso probablemente será específico a la aplicación, pero al menos debe indicar se debe permitir el acceso al rol para la función. Esta matriz de permisos servirá como un artefacto para documentar los derechos de control de acceso correctos para la lógica de negocio de todo el sistema. Como tal, debe ser creada por los equipos de proyecto con la aportación de las partes interesadas en el negocio. Después de la creación inicial, debe ser actualizada por las partes interesadas del negocio antes de cada lanzamiento, pero por lo general hacia el comienzo de la fase de diseño.

#### B. Especificar los requisitos de seguridad en base a los riesgos conocidos

Revise explícitamente los artefactos existentes que indican riesgos de seguridad de la organización o riesgos específicos al proyecto, con el fin de entender mejor el perfil de riesgo global para el software. Cuando estén disponibles, registre los recursos como el perfil de alto nivel de riesgo del negocio, los modelos individuales de amenaza de aplicación, los resultados de la revisión del diseño, revisión de código, pruebas de seguridad, etc. Además de revisar los artefactos existentes, use los modelos de casos de abuso para la aplicación, puede servir para estimular la identificación de las necesidades de seguridad concretas que, directa o indirectamente mitigan las situaciones de abuso. Este proceso debe llevarse a cabo por los dueños de negocio y los auditores de seguridad según sea necesario. En última instancia, la nociones de riesgo que conducen a nuevos requisitos de seguridad, debe convertirse en un paso dentro del fase de planeación donde los nuevos riesgos descubiertos sean evaluados específicamente por los equipos de proyecto.

# Requisitos de seguridad



Exigir que se siga el proceso de requisitos de seguridad para todos los proyectos de software y dependencias de terceros

## ACTIVIDADES

### A. Incorporar los requisitos de seguridad a acuerdos con proveedores

Más allá de los tipos de requisitos de seguridad ya identificados por el análisis anterior, beneficios de seguridad adicionales se pueden derivar de acuerdos con terceras partes. Por lo general, los requisitos y tal vez el diseño de alto nivel se desarrollarán internamente, mientras que el diseño detallado y la implementación es a menudo dejada a los proveedores. Basándose en la división del trabajo específico para cada componente desarrollado exteriormente, identifique las actividades específicas de seguridad y criterios de evaluación técnica a añadir a los contratos de proveedores. Comúnmente, se trata de un conjunto de actividades de Revisión del Diseño, Revisión de Código, y de Seguridad Prácticas de prueba. Las modificaciones en el lenguaje del acuerdo deben ser manejadas caso por caso, con cada proveedor, ya que añadir requisitos adicionales, por lo general, supondrá un aumento en el costo. El costo de cada actividad potencial de seguridad debe ser equilibrada contra el beneficio de la actividad, como el uso de un componente o sistema que está siendo considerado.

### B. Ampliar el programa de auditoría para los requisitos de seguridad

Incorpore controles de integridad en los requisitos de seguridad de los proyectos de auditoría rutinaria. Como esto puede ser difícil de determinar sin un conocimiento específico del proyecto, la auditoría debería centrarse en la comprobación de los artefactos del proyecto, como la evidencia de que los requisitos o documentación de diseño se llevaron a cabo. En particular, cada requisito funcional debe ser anotado con los requisitos de seguridad basándose en los impulsores del negocio, así como los escenarios de abuso esperados. Los requisitos generales del proyecto deben contener una lista de requisitos generados a partir de las mejores prácticas, lineamientos y estándares. Además, debe haber una lista clara de los requisitos de seguridad incumplidos y una línea de tiempo estimado para su prestación en futuras versiones. Esta auditoría se debe realizar durante cada iteración de desarrollo, idealmente hacia el final del proceso de requisitos, pero se deben realizar antes de que un lanzamiento se pueda realizar.

## EVALUACIÓN

- ◆ ¿Están la mayoría de los interesados revisando los acuerdos con proveedores para los requisitos de seguridad?
- ◆ Los requisitos de seguridad ¿son especificados por los equipos de proyecto que están siendo auditados?

## RESULTADOS

- ◆ Sentar formalmente las bases para las expectativas de seguridad de código externo
- ◆ Información centralizada sobre los esfuerzos de seguridad emprendidas por cada equipo de proyecto
- ◆ Habilidad para alinear los recursos a los proyectos basándose en el riesgo de aplicación y los requisitos de seguridad deseados

## MÉTRICAS DE ÉXITO

- ◆ >80% de los proyectos pasando los requisitos de seguridad de auditoría en los últimos 6 meses
- ◆ >80% de los acuerdos con proveedores analizados contra los requisitos de seguridad contractuales en los últimos 12 meses

## COSTOS

- ◆ Aumento de los costos del desarrollo externalizado de los requisitos de seguridad adicionales
- ◆ Esfuerzo adicional del proyecto a partir de la liberación de puertas de los requisitos de seguridad

## PERSONAL

- ◆ Auditor de seguridad (2 días/año)
- ◆ Administradores (2 días/año)
- ◆ Dueños de Negocio (1 día/año)

## NIVELES RELACIONADOS

- ◆ Evaluación de amenaza - 3
- ◆ Política y cumplimiento - 2

# Arquitectura de seguridad

	 SA 1	 SA 2	 SA 3
<b>OBJETIVOS</b>	Insertar consideraciones para lineamientos proactivos de seguridad en el proceso de diseño de software	Dirija el proceso de diseño de software hacia servicios seguros conocidos y diseños seguros desde la concepción	Controlar formalmente el proceso de diseño de software y validar la utilización de componentes de seguridad
<b>ACTIVIDADES</b>	A. Mantener una lista de los marcos de trabajo de software recomendados B. Aplicar explícitamente los principios de seguridad para el diseño	A. Identificar y promover los servicios de seguridad e infraestructura B. Identificar los patrones de diseño de seguridad desde la arquitectura	A. Establecer arquitecturas y plataformas formales de referencia B. Validar el uso de marcos de trabajo, patrones, y plataformas
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Cuentan los equipos de proyectos con una lista de los componentes de terceros recomendados?</li> <li>◆ ¿Están la mayoría de los equipos de proyecto conscientes de los principios de diseño seguro y su aplicación?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Hace publicidad de los servicios compartidos de seguridad como guía para equipos de proyectos?</li> <li>◆ ¿Están los equipos de proyectos previstos con los patrones de diseño prescriptivo basado en su arquitectura de aplicación?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están los equipos de proyecto construyendo software a partir de plataformas y marcos de trabajo controlados?</li> <li>◆ ¿Están los equipos de proyecto siendo auditados para el uso de componentes seguros de arquitectura?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Prevención ad hoc de las dependencias inesperadas</li> <li>◆ Las partes interesadas son conscientes del incremento de los riesgos de proyecto debido a las bibliotecas y los marcos de trabajo elegidos</li> <li>◆ Se establece un protocolo dentro del desarrollo para la aplicación proactiva de mecanismos de seguridad en el diseño</li> </ul>	<ul style="list-style-type: none"> <li>◆ Comparación detallada de los activos contra las funciones de usuario para fomentar una mejor compartimentación en el diseño</li> <li>◆ Bloques de construcción reutilizables para el suministro de protecciones de seguridad y funcionalidad</li> <li>◆ Aumento de la confianza en los proyectos de software por la utilización de técnicas de diseño preestablecidas para la seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Plataformas de desarrollo a la medida de aplicaciones que brindan protecciones de seguridad</li> <li>◆ Expectativas organizacionales de todo el esfuerzo de seguridad proactiva en el desarrollo</li> <li>◆ Los interesados con mejores condiciones de tomar decisiones de negociación basadas en la necesidad de negocio para el diseño seguro</li> </ul>



### Insertar consideraciones para lineamientos proactivos de seguridad en el proceso de diseño de software

#### ACTIVIDADES

##### A. Mantener una lista de los marcos de trabajo de software recomendados

A través de proyectos de software dentro de la organización, identificar bibliotecas de software y los marcos de trabajo comúnmente usados. En general, esto no tiene por qué ser una búsqueda exhaustiva de las dependencias, sino más bien hay que centrarse en la captura de los componentes de alto nivel que se utilizan con mayor frecuencia. De la lista de componentes, agrúpelos en categorías funcionales basándose en las características principales proporcionadas por el componente de terceros. También identifique la prevalencia de uso de cada componente a través de equipos de proyectos para ponderar la confianza en el código de terceros. Utilizando esta lista ponderada como una guía, cree una lista de componentes para anunciarlos a toda la organización como componentes recomendados. Varios factores deben contribuir a las decisiones para su inclusión en la lista recomendada. Aunque la lista puede ser creada sin la realización de investigaciones en concreto, es aconsejable inspeccionar el historial de incidentes, los antecedentes para responder a las vulnerabilidades, la adecuación de la funcionalidad de la organización, la excesiva complejidad en el uso de los componentes de terceros, etc. para cada uno. Esta lista deberá ser creada por los desarrolladores senior y arquitectos, pero también debe incluir los aportes de los administradores y auditores de seguridad. Después de la creación, la lista de componentes recomendados se comparará con las categorías funcionales que deben ser objeto de publicidad en la organización. En última instancia, el objetivo es proporcionar configuraciones predeterminadas conocidas a los equipos de proyecto.

##### B. Aplicar explícitamente los principios de seguridad para el diseño

Durante el diseño, el personal técnico del equipo de proyecto debe utilizar una lista corta de los principios rectores en materia de seguridad como una lista de control frente a los diseños detallados del sistema. Normalmente, los principios de seguridad incluyen la defensa a profundidad, aseguramiento del eslabón más débil, el uso de valores predeterminados seguros, simplicidad en el diseño de la funcionalidad de seguridad, fallos seguros, el equilibrio de la seguridad y facilidad de uso, ejecución con privilegios mínimos, evitar la seguridad por oscuridad, etc. En particular, para las interfaces de perímetro, el equipo de diseño debe considerar cada principio en el contexto del sistema en su totalidad e identificar las características que se pueden agregar para reforzar la seguridad en cada tipo de interfaz. Generalmente, estos deben ser tan limitados que sólo requieren de una pequeña cantidad de esfuerzo adicional más allá del costo normal de la aplicación de los requisitos funcionales y cualquier cosa más grande debe ser observado y previsto para versiones futuras. Si bien este proceso debe ser realizado por cada equipo de proyecto después de haber sido entrenados en seguridad, es útil para incorporar más personal con conocimientos de seguridad para auxiliar en la toma de decisiones de diseño.

#### EVALUACIÓN

- ◆ ¿Cuentan los equipos de proyectos con una lista de los componentes de terceros recomendados?
- ◆ ¿Están la mayoría de los equipos de proyecto conscientes de los principios de diseño seguro y su aplicación?

#### RESULTADOS

- ◆ Prevención ad hoc de las dependencias inesperadas
- ◆ Las partes interesadas son conscientes del incremento de los riesgos de proyecto debido a las bibliotecas y los marcos de trabajo elegidos
- ◆ Se establece un protocolo dentro del desarrollo para la aplicación proactiva de mecanismos de seguridad en el diseño

#### MÉTRICAS DE ÉXITO

- ◆ >80% del personal de desarrollo informado sobre las recomendaciones del marco de trabajo de software durante el pasado año
- ◆ >50% de los proyectos de presentando proactivamente la aplicación de los principios de seguridad para el diseño

#### COSTOS

- ◆ Construcción, mantenimiento, y la concientización sobre las recomendaciones del marco de trabajo de software
- ◆ Esfuerzo adicional del proyecto a partir del análisis y aplicación de los principios de seguridad

#### PERSONAL

- ◆ Arquitectos (2-4 días/año)
- ◆ Desarrolladores (2-4 días/año)
- ◆ Auditores de Seguridad (2-4 días/año)
- ◆ Administradores (2 días/año)

#### NIVELES RELACIONADOS

- ◆ Educación y orientación - I



Dirija el proceso de diseño de software hacia servicios seguros conocidos y diseños seguros desde la concepción

## EVALUACIÓN

- ◆ ¿Hace publicidad de los servicios compartidos de seguridad como guía para equipos de proyectos?
- ◆ ¿Están los equipos de proyectos previstos con los patrones de diseño prescriptivo basado en su arquitectura de aplicación?

## RESULTADOS

- ◆ Comparación detallada de los activos contra las funciones de usuario para fomentar una mejor compartimentación en el diseño
- ◆ Bloques de construcción reutilizables para el suministro de protecciones de seguridad y funcionalidad
- ◆ Aumento de la confianza en los proyectos de software por la utilización de técnicas de diseño preestablecidas para la seguridad

## MÉTRICAS DE ÉXITO

- ◆ >80% de los proyectos con la matriz de permisos actualizada en los últimos 6 meses
- ◆ >80% de los equipos de proyecto informados sobre las pautas de seguridad aplicables, en los últimos 6 meses

## COSTOS

- ◆ Construir o comprar licencias de los patrones de seguridad aplicables
- ◆ Esfuerzo adicional del proyecto en curso por el mantenimiento de la matriz de permisos

## PERSONAL

- ◆ Arquitectos (2-4 días/año)
- ◆ Desarrolladores (1-2 días/año)
- ◆ Administradores (1-2 días/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Auditores de Seguridad (1-2 días/año)

## NIVELES RELACIONADOS

- ◆ Educación y orientación - I

## ACTIVIDADES

### A. Identificar y promover los servicios de seguridad e infraestructura

Las organizaciones deben identificar las infraestructuras o los servicios compartidos con funcionalidad de seguridad.

Estos suelen incluir inicio de sesión único en los servicios, sistemas de directorio corporativo, control de acceso o derechos de los servicios y sistemas de autenticación. Mediante la recopilación y evaluación de sistemas reutilizables, ensamblar una lista de tales recursos y clasificarlas por el mecanismo de seguridad que cumplen. También es útil tener en cuenta cada uno de los recursos en términos de por qué un equipo de desarrollo se desearía integrar en ella, es decir, los beneficios de utilizar el recurso compartido. Si existen múltiples recursos en cada categoría, una organización debe seleccionar y estandarizar uno o más de servicios compartidos por categoría. Dado que los desarrollos de software futuros se basarán en estos servicios seleccionados, cada uno debe ser cuidadosamente auditado para garantizar que la postura básica de seguridad es comprendida. Para cada servicio seleccionado, los lineamientos de diseño deberían ser creados por los equipos de desarrollo, para comprender cómo ellos se integran con el sistema. Después de que estas guías están montadas, deben ponerse a disposición de los equipos de desarrollo mediante la capacitación, orientación, lineamientos y normas. Los beneficios de hacerlo incluyen la promoción de los sistemas seguros conocidos, una guía de seguridad simplificada para los equipos de diseño del proyecto, y caminos más claros para la construcción del aseguramiento alrededor de las aplicaciones que utilizan los servicios de seguridad compartidos.

### B. Identificar los patrones de diseño de seguridad desde la arquitectura

A través de los proyectos de software en una organización, cada uno debe ser clasificados en función del tipo de arquitectura genérica. Categorías comunes incluyen aplicaciones cliente-servidor, sistemas de dispositivos, aplicaciones de escritorio, aplicaciones Web, plataformas de servicios Web, sistemas de transaccionales intermedios, aplicaciones mainframe, etc. dependiendo de la especialidad de sus organizaciones, es posible que se tengan que realizar categorías más detalladas con base en el lenguaje, o la arquitectura del procesador, o incluso en la era de publicación. Para el tipo genérico de arquitectura de Software, un conjunto de patrones de diseño genéricos que representa la implementación de métodos adecuados de funcionalidad de seguridad se pueden obtener y aplicar a los diseños individuales de los proyectos de software de la organización. Estos patrones de diseño de seguridad representan definiciones generales de los elementos de diseño genérico que pueden ser investigados o ser comprados, a menudo es más eficaz si estos patrones son personalizados para ser más específicos a su organización. Los patrones de ejemplo incluyen un subsistema de inicio de sesión único, un modelo de delegación cruzado, un diseño reforzado de interfaz, un modelo de autorización para la separación de funciones, un modelo de registro centralizado, etc. El proceso de identificación de los patrones aplicables y apropiados debería llevarse a cabo por los arquitectos, desarrolladores senior y otras personas con habilidad técnica durante la fase de diseño.



# Arquitectura de seguridad



## Controlar formalmente el proceso de diseño de software y validar la utilización de componentes de seguridad

### ACTIVIDADES

#### A. Establecer arquitecturas y plataformas formales de referencia

Después de promover la integración con los servicios compartidos de seguridad y de trabajar con patrones de seguridad específicos a cada tipo de arquitectura, se debe seleccionar de los equipos de proyecto una colección de código implementando estas piezas de funcionalidad y se deben utilizar como base de la base de código compartido. Esta base de código compartida inicialmente puede comenzar como una colección de las librerías recomendadas que cada proyecto suele usar y puede crecer con el tiempo en uno o más marcos de software que representan plataformas de referencia sobre el que los equipos de proyectos construyen su software. Algunos ejemplos de plataformas de referencia incluyen marcos de modelo-vista-controlador de aplicaciones Web, las bibliotecas de apoyo transaccional a sistemas back-end, los marcos para las plataformas de servicios Web, los andamios (scaffolding) para aplicaciones cliente-servidor, marcos de trabajo para software intermediario (middleware) con lógica de negocios agregable, etc. Otro método de construcción de plataformas de referencia inicial es seleccionar un proyecto en particular al principio del ciclo de vida y tener al personal de seguridad con experiencia trabajando para construir la funcionalidad de seguridad de una manera genérica para que pueda ser extraída del proyecto y utilizarse en otras partes de la organización. Independientemente del enfoque de creación, las plataformas de referencia tienen ventajas en términos de velocidad de auditoría y revisiones relacionadas con la seguridad, aumentando de la eficiencia en el desarrollo, y reducción de el esfuerzo de mantenimiento. Arquitectos, desarrolladores senior y otras partes interesadas con habilidades técnicas deben participar en el diseño y creación de plataformas de referencia. Después de la creación, un equipo debe continuamente darle mantenimiento y actualizarlas.

#### B. Validar el uso de marcos de trabajo, patrones, y plataformas

Durante las auditorías de rutina de los proyectos, realice un análisis adicional de los artefactos del proyecto para medir el uso de los marcos, patrones de diseño, los servicios compartidos de seguridad, y las plataformas de referencia recomendadas. Aunque se lleva a cabo durante las auditorías de rutina, el objetivo de esta actividad es recoger información de los equipos de proyecto así como medir su esfuerzo individual en la seguridad proactiva. En general, es importante verificar varios factores con los equipos de proyecto. Identificar el uso de marcos no recomendados para determinar si puede haber falta en las recomendaciones frente a las necesidades de funcionalidad de la organización. Examine patrones de diseño no utilizados o mal empleados y módulos de plataforma de referencia para determinar si se necesitan cambios. Además, puede haber más o diferentes funcionalidades que a los equipos de proyecto les gustaría ver implementadas en las plataformas de referencia conforme evolucione la organización.

Este análisis puede realizarse por cualquier personal de seguridad técnico experto. Las métricas recolectadas de cada proyecto, deben reunirse para el análisis de los administradores y las partes interesadas.

### EVALUACIÓN

- ◆ ¿Están los equipos de proyecto construyendo software a partir de plataformas y marcos de trabajo controlados?
- ◆ ¿Están los equipos de proyecto siendo auditados para el uso de componentes seguros de arquitectura?

### RESULTADOS

- ◆ Plataformas de desarrollo a la medida de aplicaciones que brindan protecciones de seguridad
- ◆ Expectativas organizacionales de todo el esfuerzo de seguridad proactiva en el desarrollo
- ◆ Los interesados con mejores condiciones de tomar decisiones de negociación basadas en la necesidad de negociación para el diseño seguro

### MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos activos utilizando plataformas de referencia
- ◆ >80% de los proyectos reportando el uso de los marcos de trabajo, patrones y plataformas en los últimos 6 meses
- ◆ >3.0 Likert sobre la utilidad de la orientación / plataformas comunicadas por los equipos de proyecto

### COSTOS

- ◆ Construir o licenciar la(s) plataforma(s) de referencia
- ◆ Mantenimiento y apoyo de plataformas de referencia en curso
- ◆ Esfuerzo adicional del proyecto de la validación del uso durante la auditoría en curso

### PERSONAL

- ◆ Administradores (1 día/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Arquitectos (3-4 días/año)
- ◆ Desarrolladores (2-3 días/año)
- ◆ Auditores de Seguridad (2 días/año)

### NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2
- ◆ Análisis de diseño - 3
- ◆ Análisis de código - 3
- ◆ Pruebas de seguridad - 3

# Revisión de diseño

	 <b>DR 1</b>	 <b>DR 2</b>	 <b>DR 3</b>
<b>OBJETIVOS</b>	<b>Apoyar en las revisiones de diseño de software para asegurarse que existan los lineamientos de mitigación para riesgos conocidos</b>	<b>Ofrecer evaluaciones de servicios para revisar el diseño del software contra buenas prácticas integrales de seguridad</b>	<b>Exija evaluar y valide los artefactos para desarrollar un entendimiento detallado de mecanismos de protección</b>
<b>ACTIVIDADES</b>	<p>A. Identificar superficies de ataques de software</p> <p>B. Analizar el diseño contra requisitos de seguridad conocidos</p>	<p>A. Inspeccionar por completo la provisión de los mecanismos de seguridad</p> <p>B. Implementar el servicio de revisión de diseño para los equipos de proyecto</p>	<p>A. Desarrollar diagrama de flujo de datos para recursos sensible</p> <p>B. Establecer puntos de liberación para la revisión de diseño</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Documentan los equipos de proyecto el perímetro de ataque de los diseños de software?</li> <li>◆ ¿Comprueban los equipos de proyecto los diseños de software contra los riesgos de seguridad conocidos?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿La mayoría de los equipos de proyecto analizan el diseño específicamente para los mecanismos de seguridad?</li> <li>◆ ¿La mayoría de los interesados están consientes de cómo obtener una revisión de diseño formal?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿El proceso de revisión de diseño incorpora un análisis detallado a nivel de datos?</li> <li>◆ ¿Las auditorías de proyecto rutinarias necesitan los lineamientos para los resultados de la revisión de diseño?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Entendimiento a alto nivel de las implicaciones de seguridad en el perímetro de la arquitectura</li> <li>◆ Habilitar a los equipos de desarrollo para autoevaluar diseños contra las buenas prácticas de seguridad</li> <li>◆ Procesos ligeros para conducir revisiones de diseño a nivel de proyecto</li> </ul>	<ul style="list-style-type: none"> <li>◆ El servicio de evaluaciones ofrecidas formalmente y consistentemente revise la arquitectura</li> <li>◆ Resalte las fallas de seguridad en el modo de mantenimiento y sistemas antiguos</li> <li>◆ Entendimiento más profundo de los interesados del proyecto en como el software provee protección y aseguramiento</li> </ul>	<ul style="list-style-type: none"> <li>◆ Una vista granular de los puntos débiles en el diseño del sistema para fomentar una mejor compartamentalización</li> <li>◆ Concientización a nivel organización de los proyectos frente a las expectativas de seguridad base para la arquitectura</li> <li>◆ Comparación entre proyectos por eficiencia y progreso hacia la mitigación de fallas conocidas</li> </ul>

# Revisión de diseño



Apoyar en las revisiones de diseño de software para asegurarse que existan los lineamientos de mitigación para riesgos conocidos

## ACTIVIDADES

### A. Identificar superficies de ataques de software

Para cada proyecto de software, crear una visión simplificada de la arquitectura general. Normalmente, este debe ser creado sobre en base a los artefactos del proyecto, tales como requisitos de alto nivel y diseño de documentos, entrevistas con el personal técnico, o revisión del código a nivel de modular. Es importante captar los módulos de alto nivel en el sistema, pero una regla de oro para la granularidad es garantizar que el diagrama de todo el sistema que se examina quepa en una página. Desde el punto de vista de la única página de la arquitectura, analizar cada componente en términos de accesibilidad de las interfaces de los usuarios autorizados, los usuarios anónimos, los operadores, los roles específicos de la aplicación, etc. Los componentes que proporcionan las interfaces también deben considerarse en el contexto de verse en una sola página para encontrar puntos de delegación funcional o de paso de datos a través de otros componentes en el diagrama. Agrupe interfaces y componentes con perfiles de accesibilidad similares y registre esto como la superficie de ataque de software. Para cada interfaz, elabore más el diagrama de una página para anotar cualquier funcionalidad relacionada con la seguridad. Basándose en los grupos de interfaces que comprenden la superficie de ataque, compruebe el modelo por una consistencia a nivel de diseño para ver como se aseguran las interfaces con acceso similar. Cualquier ruptura en la consistencia puede ser anotada como un fallo en la evaluación. Este análisis debe de ser conducido por un equipo experto en seguridad, ya sea interno al proyecto o externo. Típicamente, después de la creación inicial, el análisis de diagrama y de la superficie de ataque solo necesita ser actualizado durante la fase de diseño cuando se hagan adiciones o cambios a las interfaces del sistema.

### B. Analizar el diseño contra requisitos de seguridad conocidos

Los requisitos de seguridad, formal o informalmente identificados, pueden ser identificados y colectados. Adicionalmente, identifique e incluya cualquier suposición de seguridad sobre la cual recaiga la operación segura del sistema. Revise cada elemento en la lista de requisitos de seguridad contra la página del diagrama de la arquitectura del sistema. Elabore el diagrama para mostrar las características a nivel de diseño que solucionan cada requerimiento de seguridad. Separados, los diagramas granulares pueden ser creados para simplificar la captura de esta información si el sistema es grande y/o complicado. El objetivo general es verificar que cada requerimiento de seguridad conocido haya sido solucionado en el diseño del sistema. Cualquier requerimiento de seguridad que no está claramente proporcionado en el nivel de diseño debe de ser anotado como una falla en la evaluación. Este análisis debe de ser conducido por un equipo experto en seguridad con la aportación de arquitectos, desarrolladores, gerentes, y dueños de negocio, según se necesite. Esto debe de ser actualizado durante la fase de diseño cuando haya cambios en los requisitos de seguridad o en el diseño del sistema a alto nivel.

## EVALUACIÓN

- ◆ ¿Documentan los equipos de proyecto el perímetro de ataque de los diseños de software?
- ◆ ¿Comprueban los equipos de proyecto los diseños de software contra los riesgos de seguridad conocidos?

## RESULTADOS

- ◆ Entendimiento a alto nivel de las implicaciones de seguridad en el perímetro de la arquitectura
- ◆ Habilitar a los equipos de desarrollo para autoevaluar diseños contra las buenas prácticas de seguridad
- ◆ Procesos ligeros para conducir revisiones de diseño a nivel de proyecto

## MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos con análisis de superficies de ataque actualizadas en los últimos 12 meses
- ◆ >50% de los proyectos con análisis de requisitos de seguridad actualizados a nivel de diseño en los últimos 12 meses

## COSTOS

- ◆ Expansión y mantenimiento de los diagramas de arquitectura de cada proyecto
- ◆ Esfuerzo adicional del proyecto para la inspección del diseño de los requisitos de seguridad y las superficies de ataque

## PERSONAL

- ◆ Arquitectos (2-3 días/año)
- ◆ Desarrolladores (1-2 días/año)
- ◆ Administradores (1 día/año)
- ◆ Auditor de seguridad (1 día/año)

## NIVELES RELACIONADOS

- ◆ Requisitos de seguridad - I



## DR 2

# Revisión de diseño

Ofrecer evaluaciones de servicios para revisar el diseño del software contra buenas prácticas integrales de seguridad

### EVALUACIÓN

- ◆ ¿La mayoría de los equipos de proyecto analizan el diseño específicamente para los mecanismos de seguridad?
- ◆ ¿La mayoría de los interesados están consientes de cómo obtener una revisión de diseño formal?

### RESULTADOS

- ◆ El servicio de evaluaciones ofrecidas formalmente y consistentemente revise la arquitectura
- ◆ Resalte las fallas de seguridad en el modo de mantenimiento y sistemas antiguos
- ◆ Entendimiento más profundo de los interesados del proyecto en como el software provee protección y aseguramiento

### MÉTRICAS DE ÉXITO

- ◆ >80% de los interesados informo el estado de las solicitudes de revisión en los últimos 6 meses
- ◆ >75% de los proyectos sometidos a revisión de diseño en los últimos 12 meses

### COSTOS

- ◆ Construcción, entrenamiento y mantenimiento del equipo de revisión de diseño
- ◆ Esfuerzo adicional del proyecto para la revisión de actividades

### PERSONAL

- ◆ Arquitectos (1-2 días/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Auditores de Seguridad (2-3 días/año)

### NIVELES RELACIONADOS

- ◆ Educación y orientación - 2
- ◆ Estrategia y métricas - 2

### ACTIVIDADES

#### A. Inspeccionar por completo la provisión de los mecanismos de seguridad

Por cada interfaz en un módulo en el diagrama de alto nivel de la arquitectura, itere a través de los mecanismos de seguridad formales y analice que el sistema esté provisto de seguridad. Este tipo de análisis puede ser desarrollado en interfaces internas p. ej. entre niveles, como también en las externas, p. ej. Aquellas que comprenden la superficie de ataque. Los seis principales mecanismos de seguridad a considerar son autenticación, autorización, validación de entradas, codificación de salidas, manejo de errores y registro de eventos. En su caso, también considere los mecanismos de criptografía y manejo de sesión. Por cada interfaz, determine donde, en el diseño del sistema, cada mecanismo es provisto y anote cualquier característica faltante o no clara como una falla. Este análisis debe de ser conducido por un equipo de seguridad experto con la asistencia del equipo de proyecto con el conocimiento específico de la aplicación. Este análisis debe de ser ejecutado una vez por publicación, usualmente, hacia el final de la fase de diseño. Después del análisis inicial, las liberaciones subsecuentes son requeridas para actualizar las fallas basadas en los cambios hechos durante el ciclo de desarrollo.

#### B. Implementar el servicio de revisión de diseño para los equipos de proyecto

Instituir un proceso por el cual los interesados del proyecto puedan solicitar una revisión de diseño. Este servicio puede ser proporcionado de manera centralizada en la organización o distribuido entre el equipo existente, pero todos los revisores deben de ser entrenados en desarrollar las revisiones completa y consistentemente. El servicio de revisión debe de ser manejado centralizadamente así la cola de las solicitudes de revisión pueden de ser priorizados por los altos directivos, arquitectos, e interesados que están familiarizados con el perfil de riesgo general del negocio para la organización. Esto permite la priorización de las revisiones del proyecto en alineación con el riesgo general del negocio. Durante una revisión de diseño, el equipo de revisión debe trabajar con el equipo de proyecto para recolectar suficiente información para formular un entendimiento de la superficie del ataque, compare los requisitos de seguridad específicos contra los elementos de diseño, y verifique los mecanismos de seguridad en los módulos de interfaces.

# Revisión de diseño



Exija evaluar y valide los artefactos para desarrollar un entendimiento detallado de mecanismos de protección

## ACTIVIDADES

### A. Desarrollar diagrama de flujo de datos para recursos sensible

Basándose en la función de negocio del proyecto de software, conduzca un análisis para identificar detalles en el comportamiento del sistema alrededor de funcionalidades de alto riesgo. Típicamente, la funcionalidad de alto riesgo correlaciona las características que implementan la creación, acceso, actualización, y eliminación de información sensible. Más allá de la información, la funcionalidad de alto riesgo también incluye lógica de negocio de naturaleza crítica específica a la aplicación, desde el punto de vista de una negación de servicio o vulnerabilidad. Por cada fuente de datos o función de negocio identificada, seleccione y use una notación estandarizada para registrar los módulos relevantes de software, fuentes de datos, actores, y mensajes que fluyen a través de ellos. Suele ser útil iniciar con el diseño de un diagrama a alto nivel e iterativamente agregar detalles relevantes mientras se remueven elementos que no corresponden al recurso sensible. Con los diagramas de flujo de datos creados por el proyecto, conduzca un análisis sobre ellos para determinar los cuellos de botella en el diseño. Generalmente, estos serán módulos de software individuales que manejan datos de diferentes niveles de sensibilidad o aquellos que dan acceso a varias funciones de negocio de varios niveles de criticidad.

### B. Establecer puntos de liberación para la revisión de diseño

Teniendo establecido un programa consistente de revisión de diseño, el siguiente paso de ejecución es establecer un punto de control en el ciclo de desarrollo de software donde el proyecto no puede pasar hasta que una revisión de diseño es conducida, y las fallas son revisadas y aceptadas. Con el fin de lograr esto, conjunto básico de expectativas debe de ser establecido p. ej. a ningún proyecto con fallas de alta severidad le será permitido pasar y todas las demás fallas deben de ser aceptadas por el dueño del negocio. Generalmente, las revisiones de diseño deben de ocurrir hacia el final de la fase de diseño para ayudar a la pronta detección de problemas de seguridad, por esto debe de ocurrir antes de que las liberaciones puedan ser hechas por el equipo del proyecto. Para sistemas antiguos o proyectos inactivos, un proceso de excepción puede ser creado para permitir a esos proyectos continuar con las operaciones, pero con un explícito tiempo de asignación para que cada uno sea revisado para identificar cualquier vulnerabilidad escondida en los sistemas existentes. Las excepciones deben de ser limitadas a no más del 20% de todos los proyectos.

## EVALUACIÓN

- ◆ ¿El proceso de revisión de diseño incorpora un análisis detallado a nivel de datos?
- ◆ ¿Las auditorías de proyecto rutinarias necesitan los lineamientos para los resultados de la revisión de diseño?

## RESULTADOS

- ◆ Una vista granular de los puntos débiles en el diseño del sistema para fomentar una mejor compartimentalización
- ◆ Concientización a nivel organización de los proyectos frente a las expectativas de seguridad base para la arquitectura
- ◆ Comparación entre proyectos por eficiencia y progreso hacia la mitigación de fallas conocidas

## MÉTRICAS DE ÉXITO

- ◆ >80% de proyectos con los flujos de datos actualizados en los últimos 6 meses
- ◆ >75% de proyectos pasando por una auditoría de revisión de diseño en los últimos 6 meses

## COSTOS

- ◆ Trabajo adicional de proyectos para el mantenimiento de los diagramas de flujo
- ◆ Trabajo adicional de la organización por retrasos de proyectos causados por fallas en las auditorías de revisiones de diseño

## PERSONAL

- ◆ Desarrolladores (2 días/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1-2 días/año)
- ◆ Dueños de Negocio (1-2 días/año)
- ◆ Auditores de Seguridad (2-3 días/año)

## NIVELES RELACIONADOS

- ◆ Arquitectura de seguridad - 3
- ◆ Análisis de código - 3

# Revisión de código

	 <b>CR 1</b>	 <b>CR 2</b>	 <b>CR 3</b>
<b>OBJETIVOS</b>	<b>Encontrar oportunamente vulnerabilidades básicas a nivel de código y otros problemas de seguridad de alto riesgo</b>	<b>Hacer revisiones de código más precisas y eficientes durante el desarrollo a través de la automatización</b>	<b>Exigir un proceso de revisión de código integral para descubrir riesgos específicos de la aplicación y a nivel del lenguaje</b>
<b>ACTIVIDADES</b>	<p>A. Crear listas de verificación para la revisión de los requisitos de seguridad conocidos</p> <p>B. Realizar revisiones en código de puntos de alto riesgo</p>	<p>A. Utilizar herramientas automatizadas de análisis de código</p> <p>B. Integrar análisis de código en el proceso de desarrollo</p>	<p>A. Personalizar el análisis de código para las preocupaciones específicas de la aplicación</p> <p>B. Establecer puntos de control para la liberación de las revisiones de código</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿La mayoría de los equipos de proyecto tienen listas de verificación basadas en los problemas más comunes?</li> <li>◆ Los equipos de proyecto ¿Generalmente realizan revisiones de algunos de los mayores riesgos en el código?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Pueden la mayoría de los equipos de proyecto acceder a herramientas automatizadas de análisis de código para encontrar problemas de seguridad?</li> <li>◆ ¿La mayoría de los interesados requieren y revisan constantemente los resultados de las revisiones de código?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿La mayoría de los equipos de proyecto utilizan automatización para comprobar código contra los estándares de programación específicos de la aplicación?</li> <li>◆ ¿Las auditorías de rutina del proyecto necesitan lineamientos para los resultados de la revisión de código antes de la liberación?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Inspección de las vulnerabilidades de código comunes que conducen al un probable descubrimiento o ataque</li> <li>◆ Revisión ligera de errores de codificación que conducen a encontrar impactos severos a la seguridad</li> <li>◆ Diligencia básica a nivel de código para el aseguramiento de la seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ El desarrollo permite consistentemente auto verificar las vulnerabilidades de seguridad a nivel de código</li> <li>◆ Resultados de análisis de rutina para compilar datos históricos de hábitos de programación segura por equipo</li> <li>◆ Los interesados están consientes de las vulnerabilidades no mitigadas para apoyar un mejor análisis de negociación</li> </ul>	<ul style="list-style-type: none"> <li>◆ Incrementar la confianza en la precisión y aplicabilidad de los resultados del análisis de código</li> <li>◆ Lineamientos organizacionales para las expectativas de programación segura</li> <li>◆ Equipos de proyecto con un objetivo a alcanzar para juzgar la seguridad a nivel de código</li> </ul>

# Revisión de código



Encontrar oportunamente vulnerabilidades básicas a nivel de código y otros problemas de seguridad de alto riesgo

## ACTIVIDADES

### A. Crear listas de verificación para la revisión de los requisitos de seguridad conocidos

De los requisitos de seguridad para un proyecto, deriva una ligera lista de verificación para una revisión de código de seguridad. Estas pueden ser comprobaciones específicas a las inquietudes de seguridad alrededor de los requisitos funcionales o comprobar por las buenas prácticas de codificación segura basadas en la implementación del lenguaje, plataforma, tipo típico de tecnología, etc. Debido a estas variaciones, a menudo se necesitan un conjunto de listas de verificación para cubrir los diferentes tipos de desarrollo de software en la organización. Independientemente, de que haya sido creada a partir de los recursos disponibles públicamente o comprados, los técnicos involucrados en el desarrollo como lo son gerentes de desarrollo, arquitectos, desarrolladores y auditores de seguridad deben de revisar las listas de verificación por eficiencia y viabilidad. Esto es importante para mantener las listas cortas y simples, teniendo como objetivo el captar problemas de alta prioridad que son sencillos de encontrar en el código ya sea manualmente o con herramientas simples de búsqueda. Las herramientas de análisis de código se deben usar también para lograr este mismo fin, pero puede también ser personalizada para reducir el conjunto general de verificaciones de seguridad a uno pequeño, un valioso conjunto con el fin de hacer el proceso de exploración y revisión eficiente. Los desarrolladores deben de ser informados brevemente sobre las metas de las listas de verificación aplicables a su función de trabajo.

### B. Realizar revisiones en código de puntos de alto riesgo

Dado que las vulnerabilidades a nivel de código pueden tener un incremento dramático en el impacto si ocurren en partes críticas para la seguridad del software, equipos de proyecto debe revisar módulos de alto riesgo por vulnerabilidades comunes. Ejemplos comunes de funcionalidades de alto riesgo incluyen módulos de autenticación, puntos de reforzamiento de controles de acceso, esquemas de manejo de sesión, interfaces externas, validadores de entradas, y analizadores de datos, etc. Utilizando las listas de verificación de las revisiones de código, el análisis puede ser ejecutado como una parte normal del proceso de desarrollo donde los miembros del equipo de proyecto son asignados a módulos específicos para revisión cuando se hacen cambios. Se puede usar también auditores de seguridad y herramientas automatizadas de revisión. Durante los ciclos de desarrollo donde el alto riesgo en código es cambiado y revisado, los gerentes de desarrollo pueden priorizar la remediación apropiadamente con la aportación de otros involucrados en el proyecto.

## EVALUACIÓN

- ◆ ¿La mayoría de los equipos de proyecto tienen listas de verificación basadas en los problemas más comunes?
- ◆ Los equipos de proyecto ¿Generalmente realizan revisiones de algunos de los mayores riesgos en el código?

## RESULTADOS

- ◆ Inspección de las vulnerabilidades de código comunes que conducen al un probable descubrimiento o ataque
- ◆ Revisión ligera de errores de codificación que conducen a encontrar impactos severos a la seguridad
- ◆ Diligencia básica a nivel de código para el aseguramiento de la seguridad

## MÉTRICAS DE ÉXITO

- ◆ >80% de los equipos de proyecto informados en las listas de verificación de las revisiones de código en los últimos 6 meses.
- ◆ >50% de los equipos de proyecto ejecutan revisiones en código de alto riesgo en los últimos 6 meses.
- ◆ >3.0 Likert sobre la utilidad de las listas de verificación de las revisiones de código reportadas por los desarrolladores.

## COSTOS

- ◆ Expansiones o licencias de listas de verificaciones de revisiones de código
- ◆ Esfuerzo adicional de los proyectos para actividades de revisiones de código en código de alto riesgo

## PERSONAL

- ◆ Desarrolladores (2-4 días/año)
- ◆ Arquitectos (1-2 días/año)
- ◆ Administradores (1-2 días/año)
- ◆ Dueños de Negocio (1 día/año)

## NIVELES RELACIONADOS

- ◆ Requisitos de seguridad - I



# CR 2

# Revisión de código

Hacer revisiones de código más precisas y eficientes durante el desarrollo a través de la automatización

## EVALUACIÓN

- ◆ ¿Pueden la mayoría de los equipos de proyecto acceder a herramientas automatizadas de análisis de código para encontrar problemas de seguridad?
- ◆ ¿La mayoría de los interesados requieren y revisan constantemente los resultados de las revisiones de código?

## RESULTADOS

- ◆ El desarrollo permite consistentemente auto verificar las vulnerabilidades de seguridad a nivel de código
- ◆ Resultados de análisis de rutina para compilar datos históricos de hábitos de programación segura por equipo
- ◆ Los interesados están consientes de las vulnerabilidades no mitigadas para apoyar un mejor análisis de negociación

## MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos con revisiones de código y aprobación de los involucrados en el desarrollo en los últimos 6 meses
- ◆ >80% de los proyectos con acceso a resultados de revisiones de código automáticas en el último mes

## COSTOS

- ◆ Investigación y selección de soluciones de análisis de código
- ◆ Costo inicial y mantenimiento de la integración de la automatización
- ◆ Esfuerzo adicional del proyecto para revisiones de automatizadas de código y mitigación de vulnerabilidades

## PERSONAL

- ◆ Desarrolladores (1-2 días/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1-2 días/año)
- ◆ Auditores de Seguridad (3-4 días/año)

## NIVELES RELACIONADOS

◆

## ACTIVIDADES

### A. Utilizar herramientas automatizadas de análisis de código

Muchas vulnerabilidades de seguridad a nivel de código son complejas de entender y requieren una inspección cuidadosa para descubrirlas. Sin embargo, hay muchas soluciones de automatización útiles disponibles para automáticamente analizar código por errores y vulnerabilidades. Existen productos comerciales y de código abierto disponibles para cubrir lenguajes de programación y marcos de trabajo populares. La selección de una solución apropiada de análisis de código está basada en varios factores incluyendo una profunda y precisa inspección, usabilidad del producto, modelo de uso, expansión y características de personalización, aplicabilidad a la arquitectura de la organización, la pila de tecnologías, etc. Utilice las aportaciones del equipo experto de seguridad así como las de los desarrolladores y gerentes de desarrollo en el proceso de selección, y revise los resultados generales con los involucrados en el desarrollo.

### B. Integrar análisis de código en el proceso de desarrollo

Una vez que se selecciona la solución de análisis, esta debe de ser integrada en el proceso de desarrollo para fomentar a los equipos de proyecto para utilizar sus capacidades. Una manera efectiva para lograr esto es configurar la infraestructura para que las exploraciones se ejecuten automáticamente en el tiempo de construcción o desde el código en el repositorio de código del proyecto. De esta manera, los resultados estarán disponibles pronto, permitiendo a los equipos de desarrollo autoevaluarse en el camino previo a la liberación. Un potencial problema con los sistemas antiguos o grandes proyectos en curso es que la exploración de código típicamente reportara las fallas en los módulos que no fueron actualizados en la liberación. Si las exploraciones automáticas se configuran para ejecutarse periódicamente, una estrategia efectiva para evitar esfuerzo adicional en la revisión es limitarse a considerar las fallas que hayan sido agregadas, removidas, o cambiadas desde la exploración previa. Es crítico no ignorar el resto de los resultados, sin embargo, los gerentes de desarrollo deben tomar las aportaciones de los auditores de seguridad, los interesados, y el equipo del proyecto para formular un plan concreto para solucionar el resto de las fallas. Si las fallas no solucionadas de la revisión de código permanecen en la liberación, estas deben de ser revisadas y aceptadas por los interesados del proyecto.



# Revisión de código



Exigir un proceso de revisión de código integral para descubrir riesgos específicos de la aplicación y a nivel del lenguaje

## ACTIVIDADES

### A. Personalizar el análisis de código para las preocupaciones específicas de la aplicación

Las herramientas de exploración de código son impulsadas por una base de conocimiento de reglas para verificar el código basándose en el API (Interfaz de programación del lenguaje) y librerías comúnmente usadas, pero tienen la limitada habilidad de entender un API personalizada y diseñada para aplicar verificaciones análogas. Sin embargo, a través de la personalización, una exploración de código puede ser un poderoso y genérico motor de análisis para encontrar problemas de seguridad de la organización específicos al proyecto. Mientras los detalles pueden variar entre herramientas en términos de facilidad y poder de análisis personalizados, las exploraciones de código personalizadas generalmente involucran verificaciones específicas para ser ejecutadas en APIs específicas y sitios con llamadas a función. Comprobaciones pueden incluir análisis para verificar la adherencia a estándares de codificación internos, los datos contaminados y sin comprobar que pasan a interfaces personalizadas, seguimiento y verificación de manejo de información sensible, uso correcto de una API interna, etc. Las verificaciones de uso de bases de código compartidas son un lugar efectivo para comenzar las exploraciones personalizadas, ya que la creación de los verificadores puede ser utilizada a través de múltiples proyectos. Para personalizar una herramienta para un código base, un auditor de seguridad puede inspeccionar ambos, códigos y diseños de alto nivel para identificar candidatos a verificadores para discutir con el equipo de desarrollo y los involucrados en la implementación.

### B. Establecer puntos de control para la liberación de las revisiones de código

Para establecer lineamientos de seguridad a nivel de código a todos los proyectos de software, un punto en particular en el ciclo de desarrollo de software puede ser establecido como un punto de control donde el estándar mínimo para los resultados de las revisiones de código deben ser cumplidos para poder hacer la liberación. Para comenzar, este estándar debe de ser sencillo de cumplir, por ejemplo escogiendo uno o dos tipos de vulnerabilidades y estableciendo el estándar que el proyecto no debe pasar con cualquier falla correspondiente. Con el tiempo, estos lineamientos estándar deberán de ser mejorados agregando criterios adicionales para pasar el punto de control. Generalmente, los puntos de control de revisiones de código deben ocurrir hacia el final de la fase de implementación, pero debe de ocurrir antes de la liberación. Para sistemas antiguos o proyectos inactivos, se debe crear un proceso de excepción para permitir a estos proyectos continuar con las operaciones, pero con un explícito periodo de asignación de las fallas. Las excepciones deben de ser limitadas a no más del 20% de todos los proyectos.

## EVALUACIÓN

- ◆ ¿La mayoría de los equipos de proyecto utilizan automatización para comprobar código contra los estándares de programación específicos de la aplicación?
- ◆ ¿Las auditorías de rutina del proyecto necesitan lineamientos para los resultados de la revisión de código antes de la liberación?

## RESULTADOS

- ◆ Incrementar la confianza en la precisión y aplicabilidad de los resultados del análisis de código
- ◆ Lineamientos organizacionales para las expectativas de programación segura
- ◆ Equipos de proyecto con un objetivo a alcanzar para juzgar la seguridad a nivel de código

## MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos usando personalizaciones de análisis de código
- ◆ >75% de los proyectos pasando por una auditoría de revisión de código en los últimos 6 meses

## COSTOS

- ◆ Expansión y mantenimiento de comprobaciones personalizadas de revisión de código
- ◆ Esfuerzo adicional de los proyectos para la auditoría de revisión de código
- ◆ Esfuerzo adicional por retrasos de proyectos causados por auditorías de revisión de código fallidas




## PERSONAL

- ◆ Arquitectos (1 día/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Auditores de Seguridad (1-2 días/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Administradores (1 día/año)

## NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2
- ◆ Arquitectura de seguridad - 3

# Pruebas de seguridad

	 <b>ST 1</b>	 <b>ST 2</b>	 <b>ST 3</b>
<b>OBJETIVOS</b>	Establecer el proceso para realizar pruebas de seguridad basándose en la implementación y los requisitos del software	Hacer pruebas de seguridad durante el desarrollo, más completas y eficientes a través de la automatización	Exigir pruebas de seguridad específicas a la aplicación para asegurarse que los lineamientos de seguridad están implementados antes de la publicación
<b>ACTIVIDADES</b>	<p>A. Deducir casos de prueba desde los requisitos de seguridad conocidos</p> <p>B. Conducir pruebas de intrusión en cada publicación del software</p>	<p>A. Utilizar herramientas automatizadas para pruebas de seguridad</p> <p>B. Integrar pruebas de seguridad en el proceso de desarrollo</p>	<p>A. Emplear automatización de pruebas de seguridad específicas de la aplicación</p> <p>B. Establecer puntos de control para la liberación de las revisiones de código</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Están los proyectos especificando pruebas de seguridad basándose en los requisitos?</li> <li>◆ ¿La mayoría de los proyectos realizan pruebas de intrusión antes de la liberación?</li> <li>◆ ¿Están los interesados consientes del estado de las pruebas de seguridad antes de la liberación?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están los proyectos usando automatización para evaluar los casos de prueba de seguridad?</li> <li>◆ ¿La mayoría de los proyectos siguen un proceso consistente para evaluar y reportar las pruebas de seguridad a los involucrados?</li> </ul>	<ul style="list-style-type: none"> <li>◆ Los casos de seguridad ¿son generados principalmente para la lógica específica de la aplicación?</li> <li>◆ ¿Las auditorías rutinarias requieren un estándar mínimo de resultados de las pruebas de seguridad?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Verificación independiente de los mecanismos esperados de seguridad alrededor de funciones críticas para el negocio</li> <li>◆ Alto nivel de diligencia debido a pruebas de seguridad.</li> <li>◆ Crecimiento de un conjunto de pruebas de seguridad para cada proyecto de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Más profunda y más consistente verificación de funcionalidad de seguridad del software</li> <li>◆ Equipo de desarrollo posibilitado para autoevaluar y corregir problemas antes de la liberación</li> <li>◆ Interesados mejor informados de vulnerabilidades abiertas cuando se hace la toma de decisión de aceptación de riesgos</li> </ul>	<ul style="list-style-type: none"> <li>◆ Lineamientos a nivel organización para un desempeño esperado de las aplicaciones contra los ataques</li> <li>◆ Conjuntos de pruebas de seguridad personalizados para mejorar la precisión de los análisis automatizados</li> <li>◆ Equipos de proyecto consientes de las metas objetivo para la resistencia contra ataques</li> </ul>

# Pruebas de seguridad



## Establecer el proceso para realizar pruebas de seguridad basándose en la implementación y los requisitos del software

### ACTIVIDADES

#### A. Deducir casos de prueba desde los requisitos de seguridad conocidos

De los requisitos de seguridad conocidos para un proyecto, identificar un conjunto de casos de prueba para comprobar la correcta funcionalidad del software. Típicamente, estos casos de prueba son obtenidos de los potenciales problemas de seguridad alrededor de los requisitos funcionales y la lógica de negocio del sistema. Pero pueden ser también incluidas pruebas genéricas para vulnerabilidades comunes relacionadas a la implementación del lenguaje o las tecnologías existentes. Con frecuencia, es más efectivo usar el tiempo de los equipos de proyecto para construir casos de prueba específicos a la aplicación y utilizar los recursos disponibles públicamente o comprar bases de conocimiento para seleccionar los casos de prueba generales aplicables para la seguridad. Aunque no se exige, las herramientas automatizadas de prueba de seguridad pueden ser utilizadas también para cubrir los casos de prueba de seguridad generales. Este planteamiento de casos de prueba debe de ocurrir durante las fases de levantamiento de requisitos y/o diseño, pero deben de ocurrir antes de las pruebas finales, antes de la liberación. Los casos de prueba candidatos, deben de ser revisados por aplicabilidad, eficacia, viabilidad y relevancia en el desarrollo, seguridad y equipo de aseguramiento de calidad.

#### B. Conducir pruebas de intrusión en cada publicación del software

Usar un conjunto de casos de prueba por proyecto ya identificados, las pruebas de intrusión se deben realizar para evaluar el desempeño del sistema contra cada caso. Es común que esto ocurra durante la fase de pruebas antes de la liberación del software. Los casos de pruebas de intrusión pueden incluir pruebas específicas a la aplicación para comprobar la robustez de la lógica de negocio, así como las pruebas contra las vulnerabilidades más comunes para comprobar el diseño y la implementación. Una vez especificados, los casos de prueba de seguridad pueden ser ejecutados por el equipo experto de aseguramiento de calidad o equipo de desarrollo, pero la primera ejecución de los casos de seguridad por un equipo de proyecto debe de ser monitoreada por un auditor de seguridad para asistir y preparar a los miembros de equipo. Antes de la liberación o implementación, los interesados deben de revisar los resultados de las pruebas de seguridad y aceptar los riesgos indicados por las fallas en las pruebas de seguridad en el momento de la liberación. En casos posteriores, una línea de tiempo concreta debe de ser establecida para resolver las diferencias que vayan apareciendo.

### EVALUACIÓN

- ◆ ¿Están los proyectos especificando pruebas de seguridad basándose en los requisitos?
- ◆ ¿La mayoría de los proyectos realizan pruebas de intrusión antes de la liberación?
- ◆ ¿Están los interesados consientes del estado de las pruebas de seguridad antes de la liberación?

### RESULTADOS

- ◆ Verificación independiente de los mecanismos esperados de seguridad alrededor de funciones críticas para el negocio
- ◆ Alto nivel de diligencia debido a pruebas de seguridad.
- ◆ Crecimiento de un conjunto de pruebas de seguridad para cada proyecto de seguridad

### MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos especificando casos de pruebas de seguridad en los últimos 12 meses
- ◆ >50% de los interesados informados en el estatus del proyecto acerca de las pruebas de seguridad en los últimos 6 meses

### COSTOS

- ◆ Expansión o licencia de casos de prueba de seguridad
- ◆ Esfuerzo adicional de los proyectos para el mantenimiento y evaluación de casos de prueba de seguridad

### PERSONAL

- ◆ Testadores de Calidad (1-2 días/año)
- ◆ Auditor de seguridad (1-2 días/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Dueños de Negocio (1 día/año)

### NIVELES RELACIONADOS

- ◆ Requisitos de seguridad - I



## ST 2

# Pruebas de seguridad

Hacer pruebas de seguridad durante el desarrollo, más completas y eficientes a través de la automatización

### EVALUACIÓN

- ◆ ¿Están los proyectos usando automatización para evaluar los casos de prueba de seguridad?
- ◆ ¿La mayoría de los proyectos siguen un proceso consistente para evaluar y reportar las pruebas de seguridad a los involucrados?

### RESULTADOS

- ◆ Más profunda y más consistente verificación de funcionalidad de seguridad del software
- ◆ Equipo de desarrollo posibilitado para autoevaluar y corregir problemas antes de la liberación
- ◆ Interesados mejor informados de vulnerabilidades abiertas cuando se hace la toma de decisión de aceptación de riesgos

### MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos con pruebas de seguridad y aprobación de los interesados en los últimos 6 meses
- ◆ >80% de los proyectos con acceso a resultados de pruebas de seguridad automatizadas en el último mes

### COSTOS

- ◆ Investigación y selección de soluciones de pruebas de seguridad automatizadas
- ◆ Costo inicial y mantenimiento de la integración de la automatización
- ◆ Esfuerzo adicional de los proyectos para pruebas de seguridad automatizadas y mitigación

### PERSONAL

- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1-2 días/año)
- ◆ Auditores de Seguridad (2 días/año)
- ◆ Testadores de Calidad (3-4 días/año)

### NIVELES RELACIONADOS



### ACTIVIDADES

#### A. Utilizar herramientas automatizadas para pruebas de seguridad

Con el fin de probar contra los problemas de seguridad, un potencialmente gran número de casos de entradas deben de ser comprobados contra cada interfaz del software, lo cual puede hacer muy difícil hacer pruebas de seguridad efectivas usando implementación manual de casos de prueba. Por lo tanto, se deben usar herramientas automatizadas de pruebas de seguridad para probar automáticamente el software, resultando en pruebas de seguridad más eficientes y con resultados de mayor calidad. Productos comerciales y de código abierto están disponibles y deben ser revisados para su adecuación a la organización. La selección de una herramienta adecuada está basada en varios factores incluyendo robustez y precisión de la construcción de casos de prueba, eficacia en la prueba de los tipos de arquitectura importantes para la organización, personalización para cambiar o agregar casos de prueba, calidad y usabilidad de vulnerabilidades encontradas para el desarrollo de la organización, etc. Utilizar aportaciones del equipo técnico experto en seguridad, así como del equipo de desarrollo y de aseguramiento de la calidad en el procesos de selección y revisar los resultados generales con los involucrados en el desarrollo de software.

#### B. Integrar pruebas de seguridad en el proceso de desarrollo

Con herramientas para ejecutar pruebas de seguridad automatizadas. Los proyectos en la organización deben de ejecutar pruebas rutinariamente y revisar los resultados durante el desarrollo. De manera que puedan hacerlo escalable con una bajo esfuerzo, las herramientas de pruebas de seguridad pueden ser configuradas para ejecutarse automáticamente de forma periódica, p. ej. de manera nocturna o semanal, y las fallas deben de ser inspeccionadas como vayan ocurriendo. Conducir pruebas de seguridad tan pronto como las fases de requisitos o diseño pueden ser benéficas. Mientras, tradicionalmente, son usadas para pruebas de casos funcionales, este tipo de enfoque en el desarrollo de conducción de pruebas involucra la identificación y ejecución de casos de prueba de seguridad relevantes tempranamente en el ciclo de desarrollo, usualmente durante el diseño. Con la ejecución automática de los casos de prueba de seguridad, los proyectos entran en la fase de implementación con un número de casos fallidos para funcionalidad no existente. La implementación está completa cuando todas las pruebas son exitosas. Esto provee una clara y adelantada meta a los desarrolladores temprano en el ciclo de desarrollo, por lo tanto, se reduce el riesgo de retrasos en liberaciones debido a problemas de seguridad o aceptaciones de riesgos forzadas para cumplir los plazos del proyecto. Por cada liberación de proyecto, los resultados de las pruebas de seguridad automatizadas y manuales deben de ser presentadas a los directivos e involucrados del negocio para su revisión. Sí hay fallas sin solucionarse que permanezcan como riesgos aceptados para la liberación, los involucrados en el desarrollo y los directivos deben de trabajar juntos para establecer un tiempo concreto para solucionarlos.

# Pruebas de seguridad



**Exigir pruebas de seguridad específicas a la aplicación para asegurarse que los lineamientos de seguridad están implementados antes de la publicación**

## ACTIVIDADES

### A. Emplear automatización de pruebas de seguridad específicas de la aplicación

Ya sea por la personalización de las herramientas de pruebas de seguridad o las mejoras en la ejecución de herramientas de casos de prueba genéricos, los equipos de proyecto deben iterar formalmente a través de requisitos de seguridad y construir un conjunto de revisores automáticos para probar la seguridad de la lógica de negocio implementada. Adicionalmente, muchas herramientas de pruebas de seguridad automatizadas pueden ser extremadamente mejoradas en precisión y profundidad de la cobertura si son personalizables, para que entiendan más a detalle las interfaces específicas de software en el proyecto que se está probando. Además, los requisitos específicos a la organización sobre cumplimiento o estándares técnicos puedan ser codificados como una batería central de pruebas de cumplimiento para hacer una auditoría de colección de datos y administración de proyecto visiblemente más simple. Los equipos de proyecto deben enfocarse en la expansión granular de casos de prueba de seguridad basándose en la funcionalidad del negocio del software, y un equipo a nivel organización dirigido por un auditor de seguridad debe enfocarse en la especificación de casos de prueba automatizados para el cumplimiento de regulaciones y estándares internos.

### B. Establecer puntos de control para la liberación de las revisiones de código

Para evitar que el software se publique con errores de seguridad sencillos de encontrar, un punto en particular en el ciclo de desarrollo de software debe ser identificado como un punto de control donde deben pasar por un conjunto establecido de casos de prueba de seguridad para poder hacer la publicación del proyecto. Esto establece un lineamiento para los tipos de pruebas de seguridad que se esperan de todos los proyectos para pasar. Ya que agregar demasiados casos de prueba desde el inicio puede resultar en una burbuja de costos por el esfuerzo adicional que implica, inicie escogiendo uno o dos problemas de seguridad e incluya una amplia variedad de casos de prueba para cada uno con la expectativa de que ningún proyecto pueda pasar si falla en cualquiera de las pruebas. Con el tiempo, este lineamiento debe ser mejorado seleccionando problemas de seguridad adicionales y agregando una variedad de casos de prueba correspondientes. Generalmente, este punto de control en las pruebas de seguridad deben ocurrir hacia el final de la fase de implementación, pero debe ocurrir antes de la liberación. Para sistemas antiguos o proyectos inactivos, un proceso de excepción puede ser creado para permitir a esos proyectos continuar con las operaciones, pero con una explícita asignación de tiempo para mitigar las fallas. Las excepciones deben ser limitadas a no más del 20% de todos los proyectos.

## EVALUACIÓN

- ◆ Los casos de seguridad ¿son generados principalmente para la lógica específica de la aplicación?
- ◆ ¿Las auditorías rutinarias requieren un estándar mínimo de resultados de las pruebas de seguridad?

## RESULTADOS

- ◆ Lineamientos a nivel organización para un desempeño esperado de las aplicaciones contra los ataques
- ◆ Conjuntos de pruebas de seguridad personalizadas para mejorar la precisión de los análisis automatizados
- ◆ Equipos de proyecto consientes de las metas objetivo para la resistencia contra ataques

## MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos usando pruebas de seguridad personalizadas
- ◆ >75% de los proyectos pasando por todas las pruebas de seguridad en los últimos 6 meses

## COSTOS

- ◆ Expansión y mantenimiento de personalización de automatización de pruebas de seguridad
- ◆ Esfuerzo adicional de los proyectos en curso para los procesos de auditorías de pruebas de seguridad
- ◆ Esfuerzo adicional por retrasos de proyecto debido a fallas en las auditorías de pruebas de seguridad




## PERSONAL

- ◆ Arquitectos (1 día/año)
- ◆ Desarrolladores (1 día/año)
- ◆ Auditores de Seguridad (1-2 días/año)
- ◆ Testadores de Calidad (1-2 días/año)
- ◆ Dueños de Negocio (1 día/año)
- ◆ Administradores (1 día/año)

## NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2
- ◆ Arquitectura de seguridad - 3

# Administración de vulnerabilidades

	 <b>VM 1</b>	 <b>VM 2</b>	 <b>VM 3</b>
<b>OBJETIVOS</b>	Entender el plan de alto nivel para responder a los reportes o incidentes de vulnerabilidades	Elaborar expectativas para prácticas de respuesta para mejorar la consistencia y las comunicaciones	Mejorar en análisis y la colección de datos en el proceso de respuesta para retroalimentación en la planeación proactiva
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Identificar un punto de contacto para problemas de seguridad</li> <li>B. Crear equipo(s) informal(es) de respuesta de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Establecer un proceso consistente de respuesta a incidentes</li> <li>B. Adoptar un proceso de divulgación de problemas de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Conducir análisis de causa raíz para incidentes</li> <li>B. Recolectar métricas por incidente</li> </ul>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Tienen la mayoría de los proyectos un punto de contacto para problemas de seguridad?</li> <li>◆ ¿Tienen su organización un equipo asignado para respuestas a incidentes de seguridad?</li> <li>◆ ¿Conocen la mayoría de los equipos de proyecto su(s) punto(s) de contacto de seguridad y equipo(s) de respuesta?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Usa la organización un proceso consistente para reporte y manejo de incidentes?</li> <li>◆ ¿Conocen la mayoría de los interesados en el proyecto las publicaciones de seguridad relevantes y relacionadas con sus proyectos de sistemas?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Son inspeccionados la mayoría de los incidentes para encontrar la causa raíz y generar más recomendaciones?</li> <li>◆ La mayoría de los proyectos ¿obtienen y reportan consistentemente datos y métricas relacionadas con incidentes?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Procesos sencillos establecidos para manejar vulnerabilidades o incidentes de alta prioridad</li> <li>◆ Marcos de trabajo para notificaciones a los interesados y reporte de eventos con impacto a la seguridad.</li> <li>◆ Seguimiento proactivo de alto nivel para el manejo de problemas de seguridad.</li> </ul>	<ul style="list-style-type: none"> <li>◆ Plan de comunicación para manejar reportes de vulnerabilidades de terceros</li> <li>◆ Proceso claro para liberar parches de seguridad a los operadores de los programas</li> <li>◆ Proceso formal para dar seguimiento, manejar y comunicar internamente los incidentes</li> </ul>	<ul style="list-style-type: none"> <li>◆ Retroalimentación detallada para la mejora organizacional después de cada incidente</li> <li>◆ Estimación inicial del costo de vulnerabilidades y su explotación</li> <li>◆ Los interesados podrán tomar mejores decisiones basados en tendencias históricas de incidentes</li> </ul>

## Entender el plan de alto nivel para responder a los reportes o incidentes de vulnerabilidades

### ACTIVIDADES

#### A. Identificar un punto de contacto para problemas de seguridad

Para cada división dentro de la organización o para cada proyecto, establecer un punto de contacto que sirva como un concentrador de comunicaciones para información de seguridad. Mientras generalmente esta responsabilidad no toma mucho tiempo de un individuo, el propósito de tener un punto de contacto predeterminado es para agregar estructura y gobernabilidad para la administración de vulnerabilidades. Ejemplos de incidentes que pueden causar la utilización incluyen la recepción de un reporte de vulnerabilidades de una entidad externa, la explotación u otra falla de seguridad de un programa, el descubrimiento interno de vulnerabilidades de alto riesgo, etc. En caso de un evento, el contacto más cercano entraría como un recurso extra y consejero al proyecto afectado para proveer guía técnica y reportar a otros involucrados avances en el progreso de mitigación. El punto de contacto debería ser elegido entre el personal técnico o administrativo de seguridad con un amplio conocimiento de los proyectos de sistemas de la organización. Una lista de estos puntos de contacto de seguridad asignados debería ser mantenida centralmente y actualizada al menos cada seis meses. Además, anunciar y darle publicidad a esta lista le permite al personal de la organización pedir ayuda y trabajar directamente uno con el otro en problemas de seguridad.

#### B. Crear equipo(s) informal(es) de respuesta de seguridad

De la lista de individuos asignados como puntos de contacto para seguridad o de personal dedicado a la seguridad, selecciona un pequeño grupo para servir como un equipo técnico de respuesta centralizado para seguridad. Las responsabilidades del equipo incluyen tomar directamente el liderazgo de los incidentes de seguridad o los reportes de vulnerabilidades y ser responsables de la priorización, mitigación y reporte a los interesados. Dada su responsabilidad, cuando es necesario los miembros del equipo de respuesta de seguridad también son responsables de la comunicación y reportes ejecutivos durante un incidente. Es probable que la mayoría del tiempo, el equipo de respuesta de seguridad no estará operando en esta capacidad, aunque deben ser lo suficientemente flexibles para poder responder rápidamente o que exista un proceso sencillo para delegar el incidente a otro miembro del equipo. El equipo de respuesta debería tener una junta al menos anual para repasar con todos los puntos de contacto de seguridad el proceso de respuesta y las expectativas de alto nivel sobre los reportes de seguridad de los equipos de proyecto.

### EVALUACIÓN

- ◆ ¿Tienen la mayoría de los proyectos un punto de contacto para problemas de seguridad?
- ◆ ¿Tienen su organización un equipo asignado para respuestas a incidentes de seguridad?
- ◆ ¿Conocen la mayoría de los equipos de proyecto su(s) punto(s) de contacto de seguridad y equipo(s) de respuesta?

### RESULTADOS

- ◆ Procesos sencillos establecidos para manejar vulnerabilidades o incidentes de alta prioridad
- ◆ Marcos de trabajo para notificaciones a los interesados y reporte de eventos con impacto a la seguridad.
- ◆ Seguimiento proactivo de alto nivel para el manejo de problemas de seguridad.

### MÉTRICAS DE ÉXITO

- ◆ >50% de la organización informada sobre el punto de contacto de seguridad más cercano en los últimos 6 meses.
- ◆ >1 junta del equipo de respuesta de seguridad y puntos de contacto en los últimos 12 meses

### COSTOS

- ◆ Esfuerzo adicional variable y continuo del personal que conforma los roles de puntos de contacto de seguridad
- ◆ Identificación del equipo apropiado de respuesta de seguridad

### PERSONAL

- ◆ Auditores de Seguridad (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Dueños de Negocio (1 día/año)

### NIVELES RELACIONADOS

- ◆ Educación y orientación - 2
- ◆ Estrategia y métricas - 3



## Elaborar expectativas para prácticas de respuesta para mejorar la consistencia y las comunicaciones

### EVALUACIÓN

- ◆ ¿Usa la organización un proceso consistente para reporte y manejo de incidentes?
- ◆ ¿Conocen la mayoría de los interesados en el proyecto las publicaciones de seguridad relevantes y relacionadas con sus proyectos de sistemas?

### RESULTADOS

- ◆ Plan de comunicación para manejar reportes de vulnerabilidades de terceros
- ◆ Proceso claro para liberar parches de seguridad a los operadores de los programas
- ◆ Proceso formal para dar seguimiento, manejar y comunicar internamente los incidentes

### MÉTRICAS DE ÉXITO

- ◆ >80% de los equipos de proyectos actualizados con el proceso de respuesta a incidentes en los últimos 6 meses
- ◆ >80% de los interesados actualizados sobre la divulgación de problemas de seguridad en los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional de las organizacionales derivados del proceso de respuesta a incidentes

### PERSONAL

- ◆ Auditores de Seguridad (3-5 días/año)
- ◆ Administradores (1-2 días/año)
- ◆ Dueños de Negocio (1-2 días/año)
- ◆ Soporte/Operadores (1-2 días/año)

### NIVELES RELACIONADOS



### ACTIVIDADES

#### A. Establecer un proceso consistente de respuesta a incidentes

Extendiéndose desde el equipo informal de respuesta de seguridad, documente explícitamente el proceso de respuesta a incidentes de la organización, así como los procedimientos que se esperan que los miembros del equipo sigan. Además, cada miembro del equipo de respuesta de seguridad debe ser entrenado en este material, al menos una vez al año. Hay varios principios para un buen proceso de respuesta a incidentes e incluyen una priorización inicial para evitar daño adicional, administración de los cambios y aplicación de actualizaciones, administrar el personal de proyectos y a otros involucrados en el incidente, recolección y preservación de evidencia forense, limitar la comunicación del incidente a los interesados, reportes bien definidos a los interesados y/o árboles de comunicación, etc. Con los equipos de desarrollo, los encargados de seguridad deberían trabajar juntos para conducir el análisis técnico para verificar hechos y suposiciones sobre cada reporte de incidente o vulnerabilidad. De la misma forma, cuando los equipos de proyecto detectan un incidente o vulnerabilidad de alto riesgo, deberían seguir un proceso interno que los ponga en contacto con algún miembro del equipo de respuesta de seguridad.

#### B. Adoptar un proceso de divulgación de problemas de seguridad

Para la mayoría de las organizaciones, no es deseable dejar que las noticias de un problema de seguridad se hagan públicas, pero hay varias formas importantes en las cuales se deben llevar las comunicaciones internas y externas sobre problemas de seguridad. La primera y más común es con la creación e implementación de parches de seguridad para los sistemas producidos por la organización. Generalmente, si todos los proyectos de sistemas se usan solo internamente, entonces el problema es menos crítico, pero para todos los contextos donde los sistemas los operan personas externas a la organización, debe existir un proceso de liberación de parches. Debe proveer varios factores, incluyendo la administración del cambio y pruebas de regresión antes de la liberación del parche, el anuncio a los operadores/usuarios con una categoría asignada de criticalidad para el parche, detalles técnicos escasos para que la vulnerabilidad no pueda ser entendida y explotada directamente, etc. Otro camino para comunicaciones externas es con empresas externas que reportan vulnerabilidades de seguridad en los sistemas de una organización. Al adoptar y publicar externamente el proceso esperado con tiempos de respuesta, las organizaciones que reportan sus vulnerabilidades son alentadas a seguir prácticas de divulgación responsable. Finalmente, muchos estados y países requieren comunicaciones externas para incidentes que involucren el robo de datos de información personalmente identificable y otros tipos de datos sensibles. Si este tipo de incidente ocurre, el equipo de respuesta de seguridad debería trabajar con los administradores y los involucrados del negocio para determinar los pasos apropiados a seguir.



## Mejorar en análisis y la colección de datos en el proceso de respuesta para retroalimentación en la planeación proactiva

### ACTIVIDADES

#### A. Conducir análisis de causa raíz para incidentes

Aunque potencialmente consume mucho tiempo, el proceso de respuesta a incidentes debería ser aumentado para incluir análisis adicionales para identificar las fallas de seguridad claves. Estas causas raíz pueden ser problemas técnicos como vulnerabilidades a nivel de código, errores de configuración, etc. o pueden ser problemas con las personas/procesos como ingeniería social, fallas al seguir los procesos, etc. Una vez que se identifica una causa raíz para un incidente, debe ser usada como una herramienta para encontrar otras debilidades potenciales en la organización donde un incidente similar podría haber ocurrido. Para cada debilidad adicional identificada se deben comunicar recomendaciones adicionales para mitigaciones proactivas como parte del cierre del esfuerzo del incidente original. Todas las recomendaciones basadas en el análisis de causa raíz deben ser revisados por la administración y los interesados relevantes del negocio ya sea para calendarizar actividades de mitigación o tomar nota de la aceptación de los riesgos.

#### B. Recolectar métricas por incidente

Al tener un proceso centralizado para manejar los reportes de problemas y vulnerabilidades de alta prioridad, una organización es capaz de tomar mediciones de las tendencias en el tiempo y determinar el impacto y eficiencia de las iniciativas para garantizar la seguridad. Los registros de incidentes pasados deben ser almacenados y revisados al menos cada 6 meses. Agrupar incidentes similares y simplemente listar el conteo general de cada tipo de problema. Mediciones adicionales de los incidentes incluyen la frecuencia de proyectos de sistemas afectados por incidentes, tiempo de inactividad de los sistemas, y costo por la pérdida de utilización, recursos humanos usados en el manejo y limpieza del incidente, estimados de los costos a largo plazo como multas de regulaciones o daños a la marca, etc. Para causas raíz que fueron problemas técnicos en su naturaleza, también es útil identificar que tipo de práctica proactiva, de revisión u operacional pudo haberla detectado antes o minimizado el daño. Esta información es retroalimentación concreta para el proceso de planeación del programa ya que representa el impacto real en la seguridad que la organización ha sentido a través del tiempo.

### EVALUACIÓN

- ◆ ¿Son inspeccionados la mayoría de los incidentes para encontrar la causa raíz y generar más recomendaciones?
- ◆ La mayoría de los proyectos ¿obtienen y reportan consistentemente datos y métricas relacionadas con incidentes?

### RESULTADOS

- ◆ Retroalimentación detallada para la mejora organizacional después de cada incidente
- ◆ Estimación inicial del costo de vulnerabilidades y su explotación
- ◆ Los interesados podrán tomar mejores decisiones basados en tendencias históricas de incidentes

### MÉTRICAS DE ÉXITO

- ◆ >80% de incidentes documentados con causas raíz y recomendaciones adicionales en los últimos 6 meses
- ◆ >80% de incidentes incluidos en las métricas en los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional de organizaciones para conducir investigaciones más profundas y análisis de incidentes
- ◆ Costos regulares organizacionales de recolección y revisión de métricas de incidentes




### PERSONAL

- ◆ Auditores de Seguridad (3 días/año)
- ◆ Administradores (2 días/año)
- ◆ Dueños de Negocio (2 días/año)

### NIVELES RELACIONADOS

- ◆ Estrategia y métricas - 3

# Fortalecimiento del ambiente

	 <b>EH 1</b>	 <b>EH 2</b>	 <b>EH 3</b>
<b>OBJETIVOS</b>	<b>Entender el ambiente operativo base para aplicaciones y componentes de sistemas</b>	<b>Mejorar la confianza en las operaciones de aplicaciones al reforzar el ambiente operativo.</b>	<b>Validar la salud de las aplicaciones y el estado de los ambientes operativos contra las mejores prácticas conocidas.</b>
<b>ACTIVIDADES</b>	<ul style="list-style-type: none"> <li>A. Mantener una especificación de ambiente operativo</li> <li>B. Identificar e instalar actualizaciones y parches críticos de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>A. Establecer un proceso rutinario de administración de parches</li> <li>B. Monitoreo del estado de configuración básico del ambiente</li> </ul>	<ul style="list-style-type: none"> <li>A. Identificar e implementar herramientas de protección relevantes para las operaciones</li> <li>B. Expandir el programa de auditoría hacia la configuración de ambientes</li> </ul>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Documentan la mayoría de los proyectos algunos requisitos para el ambiente operativo?</li> <li>◆ ¿Revisan la mayoría de los proyectos actualizaciones de seguridad para componentes de sistemas de terceros?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Se usa un proceso consistente para aplicar actualizaciones y parches a dependencias críticas?</li> <li>◆ ¿Utilizan la mayoría de los proyectos la automatización para verificar la salud de aplicaciones y ambientes?</li> </ul>	<ul style="list-style-type: none"> <li>◆ Los interesados están enterados de opciones de herramientas adicionales para proteger sistemas mientras se ejecutan las operaciones?</li> <li>◆ Las auditorías de rutina verifican la salud de los ambientes base de la mayoría de los proyectos?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Claro entendimiento de expectativas operativas en el equipo de desarrollo</li> <li>◆ Riesgos de alta prioridad de la infraestructura mitigados en un periodo de tiempo bien establecido</li> <li>◆ Operadores de sistemas con un plan de alto nivel para mantenimiento de seguridad crítico para la infraestructura</li> </ul>	<ul style="list-style-type: none"> <li>◆ Verificación granular de características de seguridad de sistemas en las operaciones</li> <li>◆ Expectativas formales de tiempos para mitigación de riesgo en infraestructura</li> <li>◆ Interesados enterados consistentemente del estado de las operaciones actuales de proyectos de sistemas</li> </ul>	<ul style="list-style-type: none"> <li>◆ Ambiente operativo reforzado con verificaciones de seguridad por capas</li> <li>◆ Metas establecidas y medidas para el mantenimiento y desempeño operativo</li> <li>◆ Reducción de la probabilidad de un ataque exitoso por medio de fallas en dependencias externas</li> </ul>



## Entender el ambiente operativo base para aplicaciones y componentes de sistemas

### ACTIVIDADES

#### A. Mantener una especificación de ambiente operativo

Para cada proyecto, se debe crear y mantener una definición concreta de las plataformas operativas esperadas. Dependiendo de la organización, esta especificación debe ser creada en conjunto con el personal de desarrollo, los involucrados en el desarrollo, grupos de soporte y operaciones, etc. Comience esta especificación capturando todos los detalles que deben ser ciertos sobre el ambiente operativo basándose en la función de negocio del sistema. Estos pueden incluir factores como la arquitectura del procesador, versiones de sistema operativo, sistemas requeridos, sistemas que generan conflictos, etc. Además, se deben registrar todas las opciones conocidas configurables por los usuarios y operadores sobre el ambiente operativo que afectan la forma en la que el sistema se va a comportar. Además, identifique cualquier suposición relevante sobre el ambiente operativo que se haya hecho en el diseño e implementación del proyecto y capture todas esas suposiciones en la especificación. Esta especificación debe ser revisada y actualizada por lo menos cada 6 meses para proyectos activos o más frecuentemente si se hacen cambios al diseño del sistema o al ambiente operativo esperado.

#### B. Identificar e instalar actualizaciones y parches críticos de seguridad

La mayoría de las aplicaciones son sistemas corren sobre un gran número de otros sistemas compuestos por bibliotecas de lenguajes de programación incluidas, componentes de terceros y marcos de trabajo de desarrollo, sistemas operativos base, etc. Debido a que las fallas de seguridad contenidas en cualquier módulo de ese gran número de sistemas afectan la seguridad general del sistema de la organización, se deben instalar las actualizaciones de seguridad críticas para estos elementos. De tal manera, se deben realizar investigaciones regulares o monitoreo constante sobre dependencias de alto riesgo. Al identificar una actualización o parche crítico que impacte la postura de seguridad del sistema, se debe hacer un plan para que los usuarios y operadores afectados actualicen sus instalaciones. Dependiendo del tipo de proyecto de sistema, los detalles de este plan pueden variar.

### EVALUACIÓN

- ◆ ¿Documentan la mayoría de los proyectos algunos requisitos para el ambiente operativo?
- ◆ ¿Revisan la mayoría de los proyectos actualizaciones de seguridad para componentes de sistemas de terceros?

### RESULTADOS

- ◆ Claro entendimiento de expectativas operativas en el equipo de desarrollo
- ◆ Riesgos de alta prioridad de la infraestructura mitigados en un periodo de tiempo bien establecido
- ◆ Operadores de sistemas con un plan de alto nivel para mantenimiento de seguridad crítico para la infraestructura

### MÉTRICAS DE ÉXITO

- ◆ >50% de los proyectos con especificaciones actualizadas de sus ambientes en los últimos 6 meses
- ◆ >50% de proyectos con una lista actualizadas de parches de seguridad críticos relevantes en los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional del proyecto para la construcción y mantenimiento de especificaciones de ambientes operativos
- ◆ Esfuerzo adicional del proyecto para el monitoreo e instalación de actualizaciones críticas de seguridad

### PERSONAL

- ◆ Desarrolladores (1-2 día/año)
- ◆ Arquitectos (1-2 día/año)
- ◆ Administradores (2-4 día/año)
- ◆ Soporte/Operadores (3-4 días/año)

### NIVELES RELACIONADOS

- ◆ Habilitación operativa - 2



# EH 2

## Fortalecimiento del ambiente

Mejorar la confianza en las operaciones de aplicaciones al reforzar el ambiente operativo.

### EVALUACIÓN

- ◆ ¿Se usa un proceso consistente para aplicar actualizaciones y parches a dependencias críticas?
- ◆ ¿Utilizan la mayoría de los proyectos la automatización para verificar la salud de aplicaciones y ambientes?

### RESULTADOS

- ◆ Verificación granular de características de seguridad de sistemas en las operaciones
- ◆ Expectativas formales de tiempos para mitigación de riesgo en infraestructura
- ◆ Interesados enterados consistentemente del estado de las operaciones actuales de proyectos de sistemas

### MÉTRICAS DE ÉXITO

- ◆ >80% de equipos de proyecto informados del proceso de administración de parches en los últimos 12 meses
- ◆ >80% de interesados informados del estado actual de parches en los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional de la organización para administración de parches y monitoreo
- ◆ Construcción o licencias de herramientas de monitoreo de infraestructura

### PERSONAL

- ◆ Arquitectos (1-2 días/año)
- ◆ Desarrolladores (1-2 días/año)
- ◆ Dueños de Negocio (1-2 días/año)
- ◆ Administradores (1-2 días/año)
- ◆ Soporte/Operadores (3-4 días/año)

### NIVELES RELACIONADOS

◆

### ACTIVIDADES

#### A. Establecer un proceso rutinario de administración de parches

Moviéndose a un proceso más formal que la aplicación personalizada de actualizaciones y parches de seguridad, de debe crear un proceso continuo in la organización para aplicar actualizaciones consistentemente a las dependencias de sistemas en los ambientes operativos. En la forma más básica, el proceso debe apuntar a ofrecer una garantía por un lapso de tiempo entre la liberación y la aplicación de actualizaciones y parches. Para hacer este proceso eficiente, las organizaciones normalmente aceptan una alta latencia para actualizaciones de baja prioridad, por ejemplo un máximo de 2 días para parches críticos hasta un máximo de 30 días para parches de baja prioridad. Esta actividad debe ser conducida por personal de soporte y operaciones, pero las juntas de rutina con los equipos de desarrollo también deben realizarse para mantener el proyecto al día sobre cambios en el pasado y actualizaciones calendarizadas. Además el personal de desarrollo debe compartir una lista de componentes de terceros de los cuales el proyecto depende internamente, de manera que el personal de soporte y operaciones pueda monitorear estos también para indicarles a los equipos de desarrollo cuando una actualización sea requerida.

#### B. Monitoreo del estado de configuración básico del ambiente

Dada la complejidad del monitoreo y la administración de parches a través de la variedad de componentes que componen la infraestructura de un proyecto de sistemas, las herramientas de automatización deben usarse para monitorear la solidez de la configuración de los sistemas. Hay herramientas comerciales y de código abierto disponibles para proveer este tipo de funcionalidad, de manera que los equipos de proyecto seleccionen una solución basada en las necesidades del negocio. Típicamente los criterios de selección incluyen la facilidad de implementación y personalización, aplicabilidad a las plataformas y tecnologías usadas en la organización, características incluidas para la administración de cambios y alertas, recolección de métricas y seguimiento de tendencias, etc. Además de la verificación del servidor y la plataforma, la automatización del monitoreo debe ser personalizada para realizar verificaciones de salud específicos de la aplicación y la configuración. El personal de soporte y operaciones debe trabajar con arquitectos y desarrolladores para determinar la cantidad óptima de monitoreo para un proyecto de sistemas determinado. Finalmente, una vez implementada una solución para el monitoreo del estado de configuración de un ambiente, las alertas inesperadas o cambios de configuración deben ser recolectados y revisados regularmente por los interesados semanalmente o por lo menos una vez cada cuarto.

# Fortalecimiento del ambiente



Validar la salud de las aplicaciones y el estado de los ambientes operativos contra las mejores prácticas conocidas.

## ACTIVIDADES

### A. Identificar e implementar herramientas de protección relevantes para las operaciones

Para construir un mejor caso de seguridad de sistemas en su ambiente operativo, se puede usar otras herramientas para mejorar la postura de seguridad del sistema en conjunto. Los ambientes operativos pueden variar dramáticamente, así que debe considerarse una tecnología de protección apropiada en el contexto del proyecto. Las herramientas de protección comúnmente usadas incluyen cortafuegos (firewalls) de aplicaciones Web, pasarelas (gateways) de seguridad de XML para servicios Web, paquetes antimodificación y de confusión (obfuscation) para sistemas de cliente-servidor o para dispositivos, sistemas de detección/prevención de intrusión a redes para infraestructura antigua (legacy), herramientas forenses de conjuntos de registros, herramientas de verificación de integridad de servidores, etc. Basados en el conocimiento del proyecto y la organización, los técnicos involucrados deberían trabajar con el personal de soporte y operaciones para identificar y recomendar herramientas de protección a los interesados del negocio para algunas operaciones seleccionadas. Si se considera una inversión valiosa en términos de reducción de riesgo contra el costo de implementación, los interesados deberían acordar planes para un piloto, instalación generalizada y mantenimiento continuo.

### B. Expandir el programa de auditoría hacia la configuración de ambientes

Cuando se conducen auditorías de rutina a nivel proyecto, expanda la revisión para incluir la inspección de artefactos relacionados al reforzamiento del ambiente operativo. Más allá de una especificación actualizada para el ambiente operativo, las auditorías deberían inspeccionar el estado actual de parches y datos históricos desde la auditoría anterior. Al conectar las herramientas de monitoreo, las auditorías pueden también verificar factores clave sobre la administración de la configuración de aplicaciones y cambios históricos. Las auditorías deberían inspeccionar también el uso de herramientas de protección de operaciones contra las que están disponibles para el tipo de arquitectura del sistema. Las auditorías de infraestructura deben ocurrir en cualquier punto después de la liberación e instalación inicial del proyecto, pero deben ocurrir por lo menos cada 6 meses. Para sistemas antiguos o proyectos sin desarrollo activo, las auditorías de infraestructura aún deben ser realizadas y revisadas por los interesados del negocio. Se debe crear un proceso de excepción para permitir que proyectos de caso especial continúen sus operaciones, pero con un tiempo explícitamente asignado para la mitigación de los hallazgos. Las excepciones deben estar limitadas a no más del 20% de todos los proyectos.

## EVALUACIÓN

- ◆ Los interesados están enterados de opciones de herramientas adicionales para proteger sistemas mientras se ejecutan las operaciones?
- ◆ Las auditorías de rutina verifican la salud de los ambientes base de la mayoría de los proyectos?

## RESULTADOS

- ◆ Ambiente operativo reforzado con verificaciones de seguridad por capas
- ◆ Metas establecidas y medidas para el mantenimiento y desempeño operativo
- ◆ Reducción de la probabilidad de un ataque exitoso por medio de fallas en dependencias externas

## MÉTRICAS DE ÉXITO

- ◆ >80% de los interesados informados de las herramientas relevantes de protección de las operaciones en los últimos 6 meses
- ◆ >75% de los proyectos pasaron las auditorías de infraestructura en los últimos 6 meses

## COSTOS

- ◆ Investigación y selección de soluciones de protección de operaciones
- ◆ Construcción o licencias de herramientas de protección para operaciones
- ◆ Esfuerzo adicional de las operaciones de mantenimiento de herramientas de protección
- ◆ Esfuerzo adicional del proyecto de auditorías de infraestructura




## PERSONAL

- ◆ Dueños de Negocio (1 día/año)
- ◆ Administradores (1-2 días/año)
- ◆ Soporte/Operadores (3-4 días)

## NIVELES RELACIONADOS

- ◆ Política y cumplimiento - 2

# Habilitación operativa

	 <b>OE 1</b>	 <b>OE 2</b>	 <b>OE 3</b>
<b>OBJETIVOS</b>	<b>Habilitar las comunicaciones entre los equipos de desarrollo y los operadores para datos críticos relevantes a seguridad</b>	<b>Mejorar las expectativas de operaciones seguras y continuas al proveer procedimientos detallados</b>	<b>Exigir la comunicación de información sobre seguridad y validar que los artefactos estén completos</b>
<b>ACTIVIDADES</b>	<p>A. Capturar la información de seguridad crítica para el ambiente de publicación</p> <p>B. Documentar procedimientos para alertas de aplicación típicas</p>	<p>A. Crear procedimientos de administración de cambio por distribución</p> <p>B. Mantener guías formales de seguridad de operaciones</p>	<p>A. Expandir el programa de auditoría para información operativa</p> <p>B. Realizar firma de código para componentes de aplicaciones</p>
<b>EVALUACIÓN</b>	<ul style="list-style-type: none"> <li>◆ ¿Entrega notas de seguridad con la mayoría de las distribuciones de sistemas?</li> <li>◆ ¿Están documentadas las alertas de seguridad y las condiciones de error para la mayoría de los proyectos?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están usando la mayoría de los proyectos un proceso de administración de cambio que es bien entendido?</li> <li>◆ ¿Entregan los equipos de proyecto una guía de seguridad de operaciones con cada liberación del producto?</li> </ul>	<ul style="list-style-type: none"> <li>◆ ¿Están la mayoría de los proyectos siendo auditados para verificar que cada entrega tenga la información de seguridad operativa apropiada?</li> <li>◆ ¿Se realiza rutinariamente la firma de código en los componentes de sistemas usando un proceso consistente?</li> </ul>
<b>RESULTADOS</b>	<ul style="list-style-type: none"> <li>◆ Mejoras personalizadas a la postura de seguridad de sistemas a través de un mejor entendimiento de la operación correcta de sistemas</li> <li>◆ Operadores y usuarios enterados de su rol para asegurar una instalación segura</li> <li>◆ Comunicaciones mejoradas entre los desarrolladores de sistemas y los usuarios para información crítica de seguridad</li> </ul>	<ul style="list-style-type: none"> <li>◆ Guía detallada para cambios relevantes de seguridad entregados con las distribuciones de sistemas</li> <li>◆ Repositorio de información actualizado con procedimientos de operación segura por aplicación</li> <li>◆ Alineación de las expectativas de operaciones entre los desarrolladores, operadores y usuarios</li> </ul>	<ul style="list-style-type: none"> <li>◆ Entendimiento organizacional de las expectativas de documentación relevantes a seguridad</li> <li>◆ Interesados más capacitados para tomar decisiones basadas en la retroalimentación de la instalación y operaciones</li> <li>◆ Operadores y usuarios capaces de verificar independientemente la integridad de las entregas de sistemas</li> </ul>

# Habilitación operativa



## Habilitar las comunicaciones entre los equipos de desarrollo y los operadores para datos críticos relevantes a seguridad

### ACTIVIDADES

#### A. Capturar la información de seguridad crítica para el ambiente de publicación

Con conocimiento de sistemas específicos, los equipos de proyecto deberían identificar cualquier información de configuración y operaciones relevante para la seguridad y comunicarla a los usuarios y operadores. Esto habilita la postura de seguridad de sistemas en los sitios de instalación para que funcionen de la misma forma en que los diseñadores del equipo lo planearon. Este análisis debe comenzar con los arquitectos y desarrolladores haciendo una lista de características de seguridad incorporadas en el sistema. De esa lista, la información sobre las opciones de configuración y su impacto en la seguridad debe ser capturada también. Para proyectos que ofrecen varios modelos de instalación diferentes, la información de las ramificaciones de seguridad de cada uno debe ser incluida para informar a los usuarios y operadores sobre el impacto de sus elecciones. Sobre todo, la lista debe ser sencilla y con el objetivo de capturar la información más crítica. Una vez creada, debe ser revisada por el equipo de proyecto y los interesados del negocio para obtener consentimiento. Además, es recomendable revisar esta lista con operadores o usuarios seleccionados para asegurar que la información es entendible y realizable. Los equipos de proyecto deberían revisar y actualizar esta información con cada entrega, pero deben hacerlo al menos cada 6 meses.

#### B. Documentar procedimientos para alertas de aplicación típicas

Con conocimiento específico de formas en las cuales los sistemas se comportan, los equipos de proyecto deben identificar los mensajes de error y alerta más importantes que requieren atención del usuario/operador. De cada evento identificado, se debe capturar la información relacionada a las acciones apropiadas del usuario/operador en respuesta al evento. De un conjunto potencialmente grande de eventos que el sistema podría generar, selecciona el conjunto de más alta prioridad basándose en la relevancia del sistema en términos del propósito de negocio. Esto debe incluir cualquier evento relacionado a seguridad, pero también puede incluir errores y alertas críticas relacionadas a la salud del sistema y al estado de la configuración. Para cada evento, se deben capturar recomendaciones de cómo hacer para informar a los usuarios y operadores de los pasos a seguir requeridos y las causas raíz potenciales del evento. Estos procedimientos deben ser revisados por el equipo de proyecto y actualizarse en cada liberación mayor del producto y/o cada 6 meses, pero pueden ser realizada más frecuentemente, por ejemplo con cada publicación de versión.

### EVALUACIÓN

- ◆ ¿Entrega notas de seguridad con la mayoría de las distribuciones de sistemas?
- ◆ ¿Están documentadas las alertas de seguridad y las condiciones de error para la mayoría de los proyectos?

### RESULTADOS

- ◆ Mejoras personalizadas a la postura de seguridad de sistemas a través de un mejor entendimiento de la operación correcta de sistemas
- ◆ Operadores y usuarios enterados de su rol para asegurar una instalación segura
- ◆ Comunicaciones mejoradas entre los desarrolladores de sistemas y los usuarios para información crítica de seguridad

### MÉTRICAS DE ÉXITO

- ◆ >50% de proyectos con información de publicación segura actualizada en los últimos 6 meses
- ◆ >50% de proyectos con procedimientos operativos para eventos actualizados en los últimos 6 meses

### COSTOS

- ◆ Esfuerzo adicional para el mantenimiento de información de seguridad de instalaciones
- ◆ Esfuerzo adicional del proyecto para el mantenimiento de procedimientos operativos críticos

### PERSONAL

- ◆ Desarrolladores (1-2 días/año)
- ◆ Arquitectos (1-2 días/año)
- ◆ Administradores (1 día/año)
- ◆ Soporte/Operadores (1 día/año)

### NIVELES RELACIONADOS

◆



Mejorar las expectativas de operaciones seguras y continuas al proveer procedimientos detallados

## EVALUACIÓN

- ◆ ¿Están usando la mayoría de los proyectos un proceso de administración de cambio que es bien entendido?
- ◆ ¿Entregan los equipos de proyecto una guía de seguridad de operaciones con cada liberación del producto?

## RESULTADOS

- ◆ Guía detallada para cambios relevantes de seguridad entregados con las distribuciones de sistemas
- ◆ Repositorio de información actualizado con procedimientos de operación segura por aplicación
- ◆ Alineación de las expectativas de operaciones entre los desarrolladores, operadores y usuarios

## MÉTRICAS DE ÉXITO

- ◆ >50% de proyectos con procedimientos actualizados de administración de cambios en los últimos 6 meses
- ◆ >80% de interesados informados del estado de las guías de seguridad operativa en los últimos 6 meses

## COSTOS

- ◆ Esfuerzo adicional de mantenimiento para procedimientos de administración de cambio
- ◆ Esfuerzo adicional de proyecto para el mantenimiento de guías de seguridad operativa

## PERSONAL

- ◆ Desarrolladores (1-2 días/año)
- ◆ Arquitectos (1-2 días/año)
- ◆ Administradores (1 día/año)
- ◆ Soporte/Operadores (1 día/año)

## NIVELES RELACIONADOS

- ◆ Fortalecimiento del ambiente - I

## ACTIVIDADES

### A. Crear procedimientos de administración de cambio por distribución

Para actualizar más formalmente a los usuarios y operadores sobre cambios relevantes en los sistemas, cada liberación debe incluir procedimientos de administración de cambios relevantes para actualizaciones e instalaciones de primera vez. Sobre todo, la meta es capturar los pasos esperados que aseguren que la instalación será correcta y no incurrir en tiempo muerto excesivo o en degradación de la postura de seguridad. Para construir estos procedimientos durante el desarrollo, los equipos de proyecto deben establecer un proceso interno sencillo para capturar los puntos relevantes que impactarían la publicación del software. Es recomendable tener este proceso en pie desde el inicio del ciclo de desarrollo para que esta información pueda ser comprendida tan pronto como sea identificada desde las fases de requisitos, diseño e implementación. Antes de cada liberación, el equipo debe revisar la lista completa para verificar su completitud y factibilidad. Para algunos proyectos, los procedimientos de cambio extensivos que acompañan a una liberación pueden especificar algún manejo especial, como la construcción de rutinas de actualización automática para evitar errores durante la instalación.

### B. Mantener guías formales de seguridad de operaciones

Comenzando con la información capturada sobre eventos críticos de sistemas y los procedimientos para manejar cada uno, los equipos de proyecto deben construir y mantener guías formales que registren toda la información relevante de seguridad que los usuarios y operadores necesitan conocer. Inicialmente, esta guía debe construirse con la información conocida sobre el sistema, como las opciones de configuración de seguridad, procedimientos de manejo de eventos, guías de instalación y actualización, especificaciones de ambiente operativo, etc. Extendiendo esto, la guía formal operativa debe explicar cada uno de estos para cubrir más detalles de forma que la mayoría de los usuarios y operadores estén informados para todas las preguntas que puedan tener. Para sistemas grandes y complejos esto puede ser un reto, así que los equipos de proyecto deben trabajar con los interesados del negocio para determinar el nivel apropiado de documentación. Además, los equipos de proyecto deben documentar todas las recomendaciones de publicación que mejoren la seguridad. La guía de seguridad operativa, después de la creación inicial, debe ser revisada por los equipos de proyecto y actualizada con cada liberación.



# Habilitación operativa



Exigir la comunicación de información sobre seguridad y validar que los artefactos estén completos

## ACTIVIDADES

### A. Expandir el programa de auditoría para información operativa

Al realizar auditorías de rutina a nivel proyecto, expanda la revisión para incluir la inspección de artefactos relacionados a la habilitación operativa de la seguridad. Los proyectos deben verificarse para asegurar que tienen guías de seguridad operativa actualizadas y completas relevantes a los detalles del sistema. Estas auditorías deben comenzar hacia el final del ciclo de desarrollo previo a la publicación, pero deben ser completadas y pasadas antes de que la liberación pueda llevarse a cabo. Para sistemas antiguos o proyectos inactivos, este tipo de auditoría debe realizarse y se debe hacer un esfuerzo único para cerrar los hallazgos y verificar el cumplimiento de la auditoría, después de lo cual ya no se requerirán auditorías subsiguientes de habilitación operativa. Los resultados de la auditoría deben ser revisados con los interesados del negocio antes de la liberación. Se debe crear un proceso de excepción para permitir que los proyectos que no pasen una auditoría continúen con la publicación, pero estos proyectos deben tener un plan con fechas concretas para la mitigación de los hallazgos. Las excepciones deben estar limitadas a no más del 20% de todos los proyectos activos.

### B. Realizar firma de código para componentes de aplicaciones

Aunque se usa frecuentemente con sistemas de propósito especial, la firma de código permite a los usuarios y operadores realizar verificaciones de integridad en sistemas de manera que puedan verificar criptográficamente la autenticidad de un módulo o distribución. Al firmar los módulos de sistemas, el equipo de proyecto permite que las instalaciones operen con un mayor grado de seguridad contra cualquier corrupción o modificación del sistema instalado en su ambiente operativo. La firma de código incurre en esfuerzo adicional para la administración de credenciales de firmas para la organización. Una organización debe seguir procesos administrativos de llave segura para asegurar la confidencialidad continua de las llaves de firma. Cuando se manejan llaves criptográficas, los interesados del proyecto también deben considerar planes para manejar problemas operativos comunes relacionados con la criptografía como rotación, compromiso o pérdida de llaves. Ya que la firma de código no es apropiada para todo, los arquitectos y desarrolladores deben trabajar con los auditores de seguridad y los interesados del negocio para determinar cuales partes del sistema deben ser firmados. Conforme el proyecto evoluciona, esta lista debe ser revisada con cada liberación, especialmente cuando se agregan nuevos módulos o se hacen cambios a componentes previamente firmados.

## EVALUACIÓN

- ◆ ¿Están la mayoría de los proyectos siendo auditados para verificar que cada entrega tenga la información de seguridad operativa apropiada?
- ◆ ¿Se realiza rutinariamente la firma de código en los componentes de sistemas usando un proceso consistente?

## RESULTADOS

- ◆ Entendimiento organizacional de las expectativas de documentación relevantes a seguridad
- ◆ Interesados más capacitados para tomar decisiones basadas en la retroalimentación de la instalación y operaciones
- ◆ Operadores y usuarios capaces de verificar independientemente la integridad de las entregas de sistemas

## MÉTRICAS DE ÉXITO

- ◆ >80% de los proyectos con guías de seguridad operativa actualizadas en los últimos 6 meses
- ◆ >80% de los interesados informados sobre opciones y estado de la firma de código en los últimos 6 meses

## COSTOS

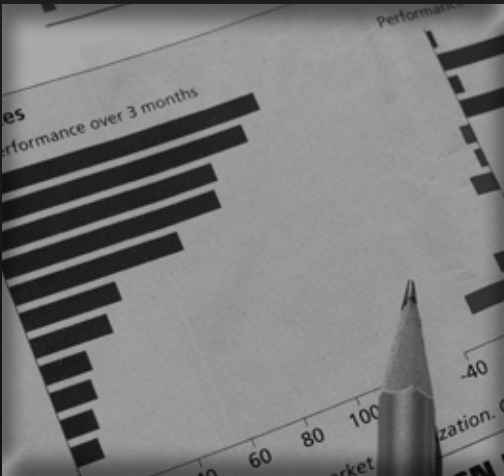
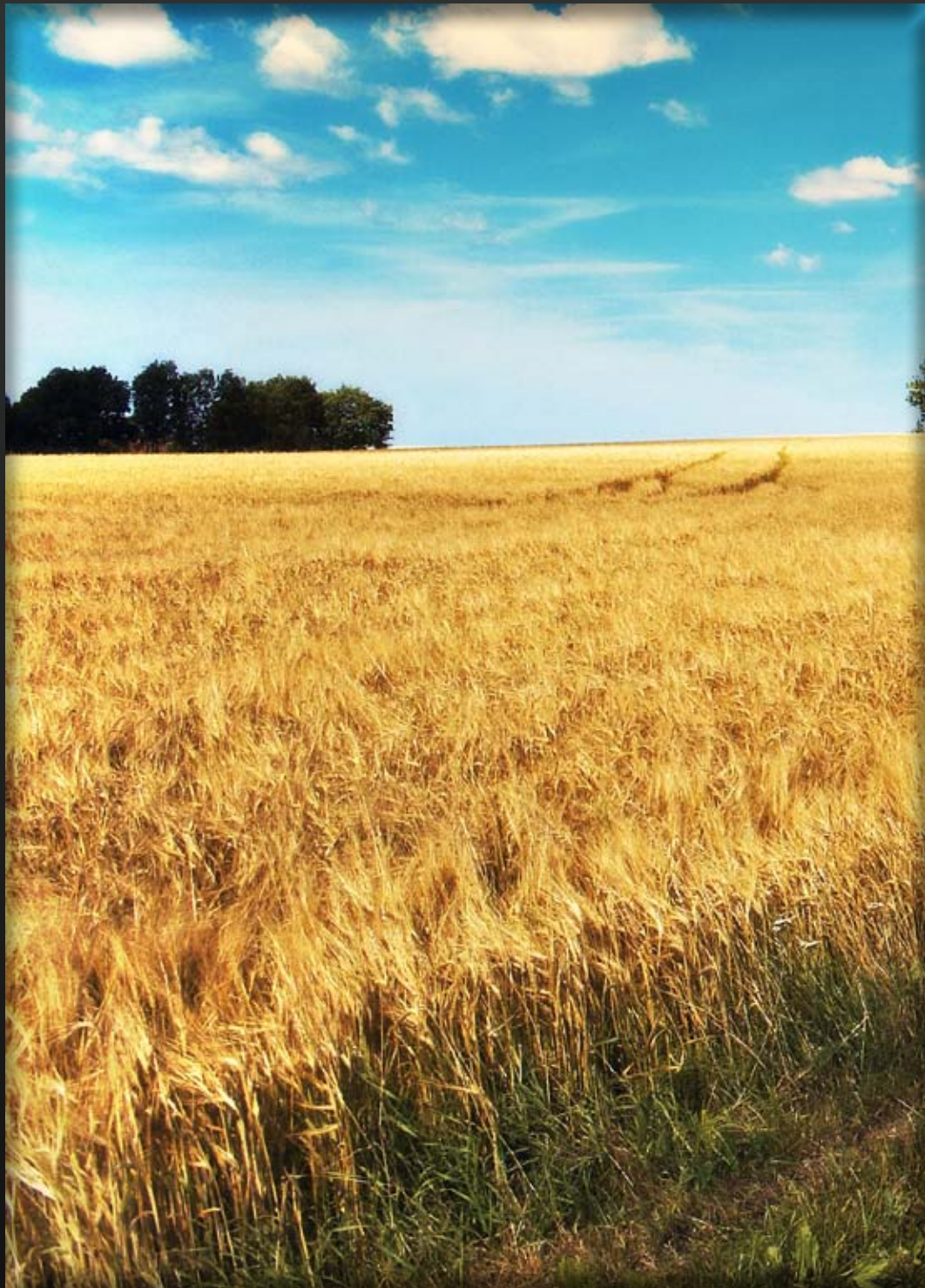
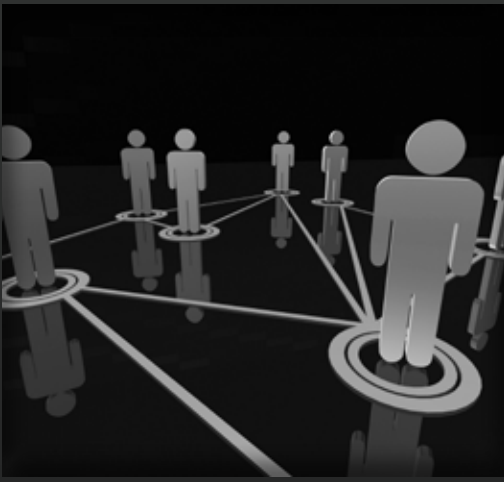
- ◆ Costos de proyecto regulares de las auditorías de guías operativas
- ◆ Costos organizacionales regulares de administración de credenciales de firma de código
- ◆ Costos de proyecto regulares de identificación y firma de módulos de código

## PERSONAL

- ◆ Desarrolladores (1 día/año)
- ◆ Arquitectos (1 día/año)
- ◆ Administradores (1 día/año)
- ◆ Auditores de Seguridad (1-2 días/año)

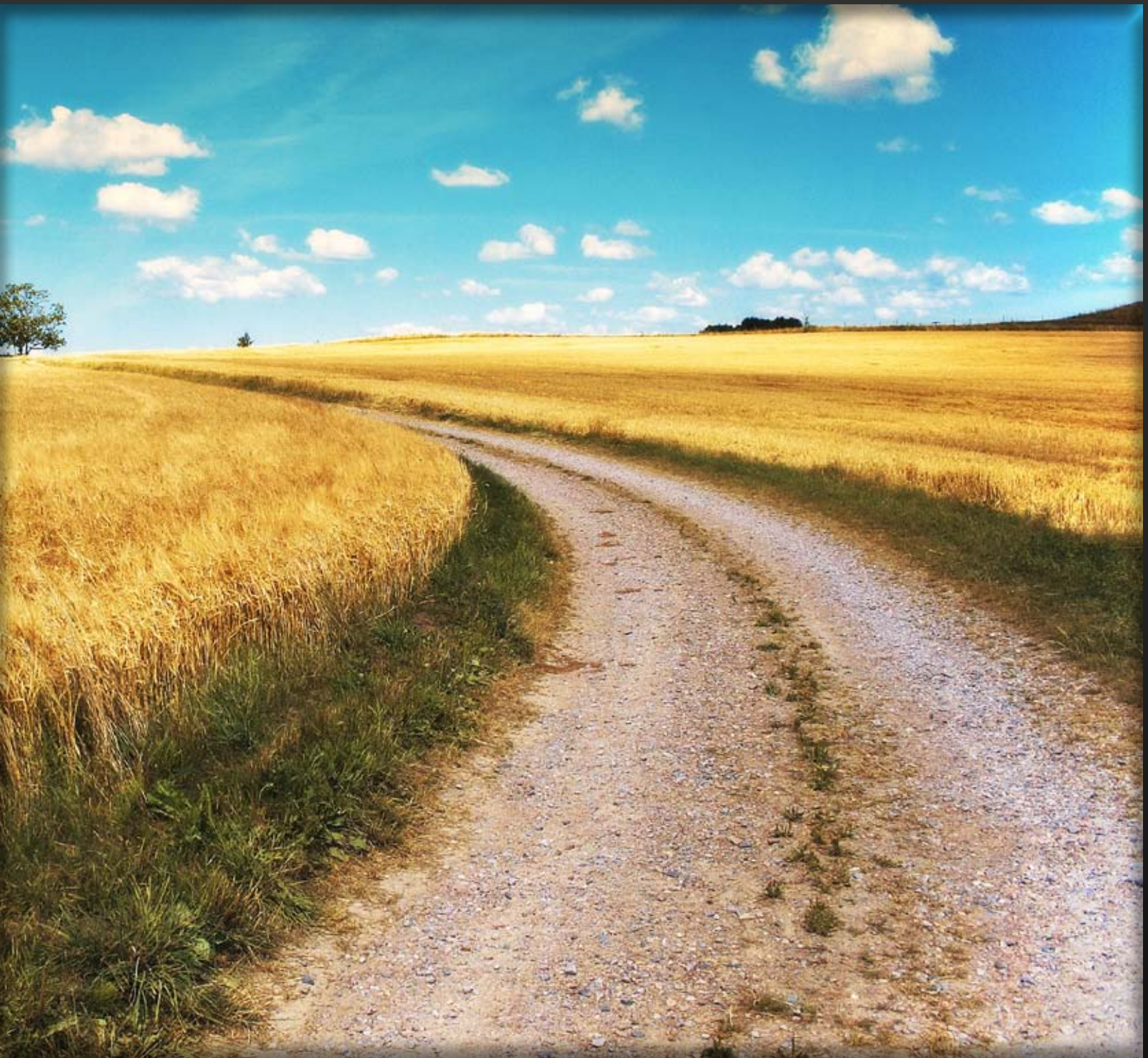
## NIVELES RELACIONADOS

◆



# Casos de Estudio

Un recorrido por escenarios de ejemplo



Esta sección habla sobre los escenarios en los cuales se explica la aplicación del SAMM en el contexto de un negocio específico. Usando las plantillas de Plan de implementación como guía, los casos de estudio explican como una organización puede adaptar mejores prácticas y tomar en cuenta los riesgos específicos de la organización cuando construye un programa de aseguramiento de software.

# VirtualWare

## Caso de estudio: Proveedor de software independiente de tamaño mediano

### PERFIL DE NEGOCIO

VirtualWare es un líder en su mercado al proporcionar plataformas de aplicación virtuales integradas para ayudar a las organizaciones a consolidar sus interfaces de aplicación en un solo ambiente. Su tecnología es proporcionada como una aplicación de servidor y un cliente de escritorio construido para múltiples ambientes incluyendo plataformas Microsoft, Apple y Linux.

La organización es de tamaño mediano (200 -1000 empleados) y tiene una presencia global con oficinas en la mayoría de los principales países del mundo.

### ORGANIZACIÓN

VirtualWare ha estado desarrollando su plataforma de software por más de 8 años. Durante este tiempo han limitado el riesgo de vulnerabilidades Web comunes debido al uso mínimo de interfaces Web. La mayoría de las plataformas de VirtualWare son ejecutadas ya sea en un sistema basado en servidor o clientes pesados ejecutándose en el escritorio.

Recientemente, VirtualWare inició varios nuevos proyectos, los cuales entregan sus interfaces de cliente y servidor a través de tecnología Web. Conocer la magnitud de los ataques comunes que se ven en la Web ha llevado a la organización a revisar su estrategia de seguridad de software y asegurarse de que aborda adecuadamente posibles amenazas hacia su organización.

Anteriormente la organización había tenido revisiones básicas del código de aplicación, y ha estado enfocada en el desempeño y funcionalidad en lugar de seguridad. Los desarrolladores de VirtualWare han estado utilizando varias herramientas de análisis de calidad de código para identificar errores y repararlos en el mismo código.

Con esto en mente, el equipo de alta dirección ha establecido un objetivo estratégico para analizar el estado actual de la seguridad de sus aplicaciones y determinar el mejor método para identificar, remover y evitar vulnerabilidades en ellas.

### AMBIENTE

VirtualWare desarrolla su tecnología de virtualización en una mezcla de Java, C++ y la tecnología .NET de Microsoft. Su tecnología de virtualización de aplicaciones ha sido escrita en C++ y ha tenido varias revisiones en busca de errores y seguridad, pero actualmente no existe un proceso formal para identificar y arreglar errores de seguridad conocidos o desconocidos.

VirtualWare ha decidido apoyar su tecnología Web en Java, aunque los sistemas de soporte (backend) están construidos usando tecnologías de Microsoft y C++. El equipo de desarrollo enfocado en las nuevas interfaces Web está compuesto principalmente por desarrolladores Java.

VirtualWare emplea más de 300 desarrolladores, con el personal dividido en equipos basado en los proyectos que van a trabajar. Hay 12 equipos con 20-40 desarrolladores por equipo. En cada equipo hay experiencia mínima en seguridad de software, y aunque los desarrolladores expertos realizan auditorías básicas en su código, la seguridad no es considerado un objetivo crítico en la organización.

Cada equipo en VirtualWare adopta un modelo diferente de desarrollo. Actualmente las 2 metodologías principales usadas son Ágile SCRUM, y modelos iterativos en Cascada. Hay mínima o ninguna guía por parte del departamento de IT o arquitectos de proyecto en cuanto a seguridad de software.

## RETOS CLAVE

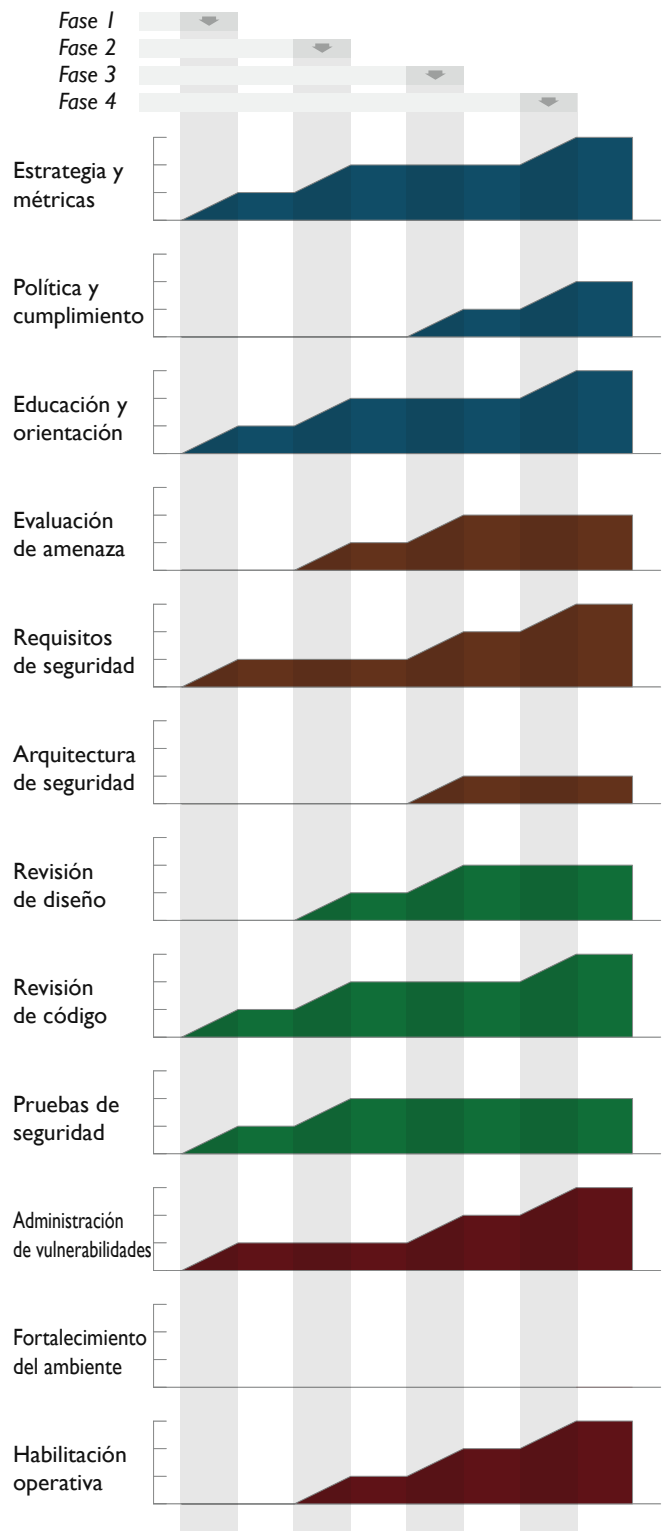
- ◆ Implementación rápida de funcionalidades para asegurar que mantienen su ventaja competitiva sobre sus rivales
- ◆ Experiencia limitada en conceptos de seguridad de software – actualmente un esfuerzo mínimo es asociado en tareas relacionadas con seguridad
- ◆ Los desarrolladores dejan la organización y son reemplazados con desarrolladores con menor experiencia
- ◆ Múltiples tecnologías son utilizadas en las aplicaciones, con aplicaciones heredadas que no han sido actualizadas desde que se construyeron originalmente
- ◆ No hay un entendimiento o ni siquiera existe una postura de seguridad o de riesgos que enfrenta la organización

VirtualWare quiso enfocarse en asegurar que sus nuevas aplicaciones Web serían entregadas de manera segura a sus clientes. Por lo tanto el enfoque inicial al implementar un programa de aseguramiento de la seguridad fue en la educación y concientización para sus equipos de desarrollo, así como proporcionar orientación técnica base en codificación segura y estándares de pruebas

La organización había recibido previamente requisitos de errores y vulnerabilidades a través de su dirección support@virtualware.net. Sin embargo, como esta era una dirección de soporte general, los requisitos existentes no siempre eran filtrados hacia los equipos apropiados dentro de la organización ni manejados correctamente. La necesidad de implementar un programa de respuesta formal a una vulnerabilidad de seguridad fue también identificada por VirtualWare.

## ESTRATEGIA DE IMPLEMENTACIÓN

La adopción de un programa de aseguramiento de seguridad dentro de una organización es una estrategia a largo plazo, e impacta significativamente en la cultura de los desarrolladores y el proceso tomado por el negocio para desarrollar y entregar aplicaciones de negocio. La adopción de esta estrategia es establecida en un período de 12 meses, y debido al tamaño de la organización será relativamente fácil de implementar en ese período.



## FASE I (MESES 0 – 3) – CONCIENTIZACIÓN Y PLANEACIÓN

VirtualWare identificó previamente que tenían conocimiento y concientización reducidas en cuanto a las amenazas de seguridad de aplicación para su organización y experiencia limitada de codificación segura. La primer fase de implementación en VirtualWare se enfocó en entrenar a los desarrolladores e implementar orientación y programas para identificar las vulnerabilidades de seguridad actuales.

Los equipos de desarrollo en VirtualWare tenían experiencia limitada en técnicas de codificación segura, por lo tanto fue desarrollado un programa de entrenamiento inicial que pudiera ser proporcionado a los desarrolladores dentro de la organización en técnicas de programación defensivas.







Con más de 300 desarrolladores y múltiples lenguajes soportados en la organización, uno de los retos claves para VirtualWare fue proporcionar un programa de educación que fuera lo suficientemente técnico para enseñar a los desarrolladores algunos de los conceptos básicos en codificación segura. El objetivo de este curso de educación inicial fue principalmente en técnicas de codificación y herramientas de pruebas. El curso desarrollado y entregado dentro de la organización duró 1 día y cubrió lo básico en codificación segura.

VirtualWare estaba conciente que tenían varias aplicaciones con vulnerabilidades y no contaban con una estrategia real en cómo identificar vulnerabilidades existentes y abordar los riesgos en un tiempo razonable. Una metodología básica de evaluación de riesgo fue adoptada y la organización llevó a cabo un análisis de las plataformas de aplicación existentes.

Esta fase también incluyó implementar varios conceptos para que el equipo de desarrollo mejorara sus herramientas de seguridad. Los equipos de desarrollo ya contaban con varias herramientas disponibles para realizar evaluaciones de tipo de calidad. Se realizó una investigación adicional sobre herramientas de análisis de código y herramientas de pruebas de seguridad.

### OBJETIVOS PRINCIPALES

Durante esta fase del proyecto, VirtualWare implementó las siguientes Prácticas y Actividades de SAMM.

 <b>SM 1</b>	A. Estimar el perfil global de riesgo del negocio B. Crear y mantener un plan de implementación para el programa de aseguramiento
 <b>EG 1</b>	A. Realizar entrenamiento técnico de concientización en seguridad B. Crear y mantener lineamientos técnicos
 <b>SR 1</b>	A. Deducir los requisitos de seguridad a partir de la funcionalidad de negocios B. Evaluar la seguridad y los lineamientos de cumplimiento para regulaciones de los requisitos
 <b>CR 1</b>	A. Crear listas de verificación para la revisión de los requisitos de seguridad conocidos B. Realizar revisiones en código de puntos de alto riesgo
 <b>ST 1</b>	A. Deducir casos de prueba desde los requisitos de seguridad conocidos B. Conducir pruebas de intrusión en cada publicación del software
 <b>VM 1</b>	A. Identificar un punto de contacto para problemas de seguridad B. Crear equipo(s) informal(es) de respuesta de seguridad

Para alcanzar estos niveles de madurez, VirtualWare implementó varios programas durante esta fase de implementación. Se adoptaron las siguientes iniciativas:

- ◆ Curso de 1 día en codificación segura (alto nivel) para todos los desarrolladores
- ◆ Construir un documento de orientación técnica para seguridad de aplicación para tecnologías usadas en la organización
- ◆ Crear un proceso de riesgo y realizar evaluaciones de riesgo de negocio a alto nivel para las plataformas de aplicación y analizar el riesgo de negocio
- ◆ Preparar lineamientos técnicos iniciales y estándares para desarrolladores
- ◆ Realizar análisis de código cortos en las plataformas de aplicación que representan un riesgo significativo para la organización
- ◆ Desarrollar casos de pruebas y uso para proyectos y evaluar los casos contra las aplicaciones;
- ◆ Nombrar un rol para iniciativas de seguridad de aplicación
- ◆ Generar un plan estratégico inicial para la siguiente fase del programa de aseguramiento

Debido a la cantidad limitada de experiencia en VirtualWare, la compañía contrató un grupo de consultores de seguridad de un tercero para asistir con la creación del programa de entrenamiento, y ayudar a escribir plan de implementación de el modelado de amenazas y el plan estratégico para la organización.

Uno de los retos claves enfrentados durante esta fase fue conseguir que los 300 desarrolladores tomaran el curso de un día de entrenamiento. Para alcanzar esto, VirtualWare tuvo 20 días de curso, con solo un pequeño grupo de desarrolladores de cada equipo asistiendo al curso a la vez. Esto redujo el impacto general en los recursos de personal durante el periodo de entrenamiento.

Durante esta fase del proyecto, VirtualWare invirtió un esfuerzo significativo en recursos en la adopción de un proceso de análisis de riesgo y revisando el riesgo de negocio de la organización. Aunque un esfuerzo considerable fue enfocado en estas tareas, fueron críticos para asegurar que los siguientes pasos implementados por VirtualWare estuvieran alineados con los riesgos de negocio enfrentados por la organización.

La dirección de VirtualWare recibió feedback positivo de la mayoría de los desarrolladores de la organización acerca del programa de formación. Aunque no se detalla, los desarrolladores sintieron que la formación inicial que recibieron les aportó los conocimientos básicos para ayudarles a escribir código seguro en su día a día.

## COSTOS DE IMPLEMENTACIÓN

Durante esta fase del proyecto se invirtió una cantidad significativa de recursos internos y costes. Hubo tres tipos diferentes de costes asociados a esta fase.

### Requerimientos de recursos internos

Esfuerzo de los recursos internos usados en la creación del contenido, talleres de trabajo y revisión de las iniciativas de seguridad de las aplicaciones durante esta fase. El esfuerzo se muestra en días totales por rol.

Desarrollador	14 días	Responsable del negocio	8 días
Arquitecto	10 días	Probador QA	3 días
Gestor	8 días	Auditor de seguridad	9 días

### Requerimientos de recursos formación (Formación por persona durante el periodo)

A cada desarrollador de VirtualWare se le requirió la asistencia a un curso de formación y por lo tanto, cada desarrollador dedico un día al programa de seguridad de las aplicaciones.

Desarrollador	1 día
---------------	-------

### Recursos externos

Debido a la falta de conocimiento dentro de VirtualWare, se externalizó la creación de contenido y la creación/impartición del programa de formación a los desarrolladores.

Consultor de seguridad	15 días	Consultor de formación	22 días
------------------------	---------	------------------------	---------

## FASE 2 (MESES 3 – 6) – FORMACIÓN Y PRUEBAS

VirtualWare identificó durante la primera fase un conjunto de vulnerabilidades en sus aplicaciones que podían ser explotadas por amenazas externas. Por lo tanto, una de las objetivos claves de esta fase fue implementar una batería de pruebas básicas y la capacidad de revisión para identificar las vulnerabilidades y corregirlas en el código.








La introducción de herramientas automatizadas para ayudar con la cobertura del código y la detección de defectos fue introducida como uno de los mayores retos en esta fase de implementación. Tradicionalmente, en el pasado, los desarrolladores habían usado herramientas automáticas con gran dificultad y por lo tanto la introducción de nuevas herramientas se percibió como un reto significativo.

Para asegurar un despliegue exitoso de las herramientas automáticas en la organización, VirtualWare lo afronto mediante un despliegue progresivo en varias fases. Las herramientas se entregarían a los responsables senior de los equipos primero y mas tarde se incorporarían otros desarrolladores progresivamente durante un periodo de tiempo. Los equipos fueron animados a adoptar las herramientas, sin embargo, no se estableció ningún proceso formal para su uso.

Esta fase de implementación también vio la introducción de un programa de formación y concienciación mas formal. Los desarrolladores que tomaron parte de la primera formación solicitaron una formación mas especifica en el área de servicios web y validación de datos. VirtualWare también implemento programas de formación adicionales para arquitectos y gestores, y adopto una campaña de concienciación dentro de la organización.

### OBJETIVOS PRINCIPALES

Durante esta fase del proyecto, VirtualWare implementó las siguientes Prácticas y Actividades de SAMM.

 <b>SM 2</b>	A. Clasificar datos y aplicaciones basado en riesgo de negocio B. Establecer y medir los objetivos de seguridad por cada clasificación
 <b>EG 2</b>	A. Realizar entrenamiento de seguridad en aplicaciones especifico para cada rol B. Utilizar mentores de seguridad para mejorar los equipos
 <b>TA 1</b>	A. Desarrollar y mantener modelos de amenaza especificos a cada aplicación B. Elabore perfil de atacante desde la arquitectura de software
 <b>DR 1</b>	A. Identificar superficies de ataques de software B. Analizar el diseño contra requisitos de seguridad conocidos
 <b>CR 2</b>	A. Utilizar herramientas automatizadas de análisis de código B. Integrar análisis de código en el proceso de desarrollo
 <b>ST 2</b>	A. Utilizar herramientas automatizadas para pruebas de seguridad B. Integrar pruebas de seguridad en el proceso de desarrollo
 <b>OE 1</b>	A. Capturar la información de seguridad critica para el ambiente de publicación B. Documentar procedimientos para alertas de aplicación típicas



Para llegar a estos niveles de madurez VirtualWare implemento varios programas durante esta fase de implementación. Se adoptaron las siguientes iniciativas:

- ◆ Cursos adicionales de educación y entrenamiento los testadores de QA, administradores y arquitectos
- ◆ Realizar la clasificación de datos y fijar metas de seguridad
- ◆ Desarrollar la metodología de análisis de riesgo en la forma de análisis de amenazas con árboles de ataques y perfiles
- ◆ Revisar e identificar los requisitos de seguridad por plataforma de aplicación
- ◆ Agregar herramientas automatizadas para asistir con la cobertura de código y el análisis de seguridad de las aplicaciones nuevas o existentes
- ◆ Revisar y mejorar los programas existentes de pruebas de intrusión
- ◆ Mejorar el ciclo de desarrollo de software existente para apoyar las pruebas de seguridad como parte del proceso de desarrollo

VirtualWare adaptó el programa de seguridad existente para crear una versión más pequeña y menos técnica, que se usó como programa inicial de seguridad para el negocio. Este es un curso más pequeño de 4 horas y se impartió a Administradores, dueños de negocio de la organización.

Una revisión de alto nivel de los programas existentes de revisión de código y pruebas de intrusión identificó que el proceso era inadecuado y necesitaba ser mejorado para proveer mejores pruebas y resultados de las vulnerabilidades de seguridad en aplicaciones. El equipo implementó un nuevo programa para realizar pruebas de intrusión y revisión de código. Como parte de ese programa, los desarrolladores Sr. de cada equipo de desarrollo dedicaron 4 días a realizar una revisión de alto de nivel en el código fuente de sus aplicaciones.

Los ejecutivos de VirtualWare entendieron que la infraestructura y aplicaciones están altamente integrados y durante esta fase, la parte operativa de las plataformas de aplicación (infraestructura) fue revisada. Esta fase se vieron los requisitos de infraestructura y la posible integración en las aplicaciones con el hardware e interfaces de aplicación recomendadas.

Durante esta fase el equipo de proyectos revisó el plan estratégico de implementación y la metodología de seguridad para aplicaciones. El objetivo de esta revisión y actualización fue clasificar formalmente los activos de datos y fijar los niveles apropiados de riesgo de negocio asociado con los activos de datos y las aplicaciones. Después de esto, el equipo de proyectos fue capaz de fijar las metas de seguridad para estas aplicaciones.

## COSTOS DE IMPLEMENTACIÓN

En esta fase del proyecto un monto significativo fue invertido en recursos internos y otros costos. Hubo 3 tipos diferentes de costos asociados a esta fase.

### Requisitos de Recursos Internos

El esfuerzo de los recursos internos fue usado en la creación de contenidos, talleres y revisión de iniciativas de seguridad en aplicaciones en esta fase. El esfuerzo se muestra en días totales por rol.

Desarrollador	8 días	Responsable del negocio	5 días
Arquitecto	10 días	Probador QA	3 días
Gestor	8 días	Auditor de seguridad	15 días
Soporte a operaciones	2 días		

### Requisitos de Recursos de Entrenamiento (Entrenamiento por persona por periodo)

Al personal adicional de VirtualWare se le pidió que atendiera a el curso de entrenamiento y por lo tanto varios roles tuvieron que dedicar tiempo a el entrenamiento en seguridad de aplicaciones.

Arquitecto	1 día	Gestor	1/2 día
Responsable del negocio	1/2 día		

### Recursos Externos

Dada la falta de conocimiento de VirtualWare, se usaron recursos externos para ayudar con la creación de contenido y crear/dar el programa de entrenamiento a los desarrolladores.

Consultor de seguridad	22 días	Consultor de formación	5 días
------------------------	---------	------------------------	--------

## FASE 3 (MESES 6-9) – ARQUITECTURA E INFRAESTRUCTURA

La tercera fase de la implementación del programa aseguramiento en VirtualWare continuó con las fase previas de implementación y se enfocó en el modelo de riesgos, arquitectura, infraestructura y capacidades de habilitación operativa.








El reto principal en esta fase fue establecer una mayor integración entre las plataformas de aplicación y el lado operativo de la organización. En la fase previa se introdujo a los equipos de VirtualWare en el manejo de vulnerabilidades y el lado operativo de la seguridad de aplicaciones. Durante esta fase VirtualWare se enfocó en la siguiente fase de estas tres áreas y agregó un proceso claro de respuesta a incidentes y procedimientos detallados para el control de cambios.

VirtualWare ha escogido iniciar con dos áreas para esta implementación. Aunque VirtualWare no es impactada por el cumplimiento regulatorio, cierto número de clientes han preguntado si las plataforma pueden ayudar a pasar el cumplimiento regulatorio. Se ha creado un equipo pequeño en VirtualWare para identificar los indicadores relevantes del cumplimiento y crear una lista de esos indicadores.

En las fases previas de VirtualWare agregó algunas herramientas automatizadas para ayudar con al revisión e identificación de vulnerabilidades. Aunque no era un aspecto principal en esta fase, los equipos de desarrollo adoptaron nuevas herramientas y han reportado que están iniciando a obtener beneficios de usar estas herramientas entre estos los grupos.

### OBJETIVOS PRINCIPALES

Durante esta fase del proyecto, VirtualWare implementó las siguientes Prácticas y Actividades de SAMM.

 <b>PC 1</b>	<ul style="list-style-type: none"> <li>A. Identificar y monitorear los indicadores externos de cumplimiento</li> <li>B. Crear y mantener lineamientos de cumplimiento</li> </ul>
 <b>TA 2</b>	<ul style="list-style-type: none"> <li>A. Desarrollar y mantener modelos de casos de abuso por proyecto</li> <li>B. Adoptar un sistema de ponderación para la medición de las amenazas</li> </ul>
 <b>SR 2</b>	<ul style="list-style-type: none"> <li>A. Generar una matriz de control de acceso a los recursos y capacidades</li> <li>B. Especificar los requisitos de seguridad en base a los riesgos conocidos</li> </ul>
 <b>SA 1</b>	<ul style="list-style-type: none"> <li>A. Mantener una lista de los marcos de trabajo de software recomendados</li> <li>B. Aplicar explícitamente los principios de seguridad para el diseño</li> </ul>
 <b>DR 2</b>	<ul style="list-style-type: none"> <li>A. Inspeccionar por completo la provisión de los mecanismos de seguridad</li> <li>B. Implementar el servicio de revisión de diseño para los equipos de proyecto</li> </ul>
 <b>VM 2</b>	<ul style="list-style-type: none"> <li>A. Establecer un proceso consistente de respuesta a incidentes</li> <li>B. Adoptar un proceso de divulgación de problemas de seguridad</li> </ul>
 <b>OE 2</b>	<ul style="list-style-type: none"> <li>A. Crear procedimientos de administración de cambio por distribución</li> <li>B. Mantener guías formales de seguridad de operaciones</li> </ul>

Para llevar a estos niveles de madurez VirtualWare implemento varios programas durante esta fase de implementación. Las siguientes iniciativas fueron adoptadas:

- ◆ Definir y publicar lineamientos técnicos sobre los requisitos de seguridad y aseguramiento de arquitectura para los proyectos dentro de la organización
- ◆ Identificar y documentar los requisitos de cumplimiento y regulaciones
- ◆ Identificar y crear lineamientos de seguridad para la infraestructura de aplicaciones
- ◆ Crear una lista definida de marcos de trabajo aprobados para el desarrollo
- ◆ Mejorar el proceso de modelado de amenazas existente usado en VirtualWare
- ◆ Adoptar un plan de respuesta a incidentes y preparar el proceso de publicación para las vulnerabilidades reportadas
- ◆ Agregar procedimientos de manejo de cambios y lineamientos formales para todos los proyectos

Para coincidir con la introducción de herramientas automatizadas de los desarrolladores (desde la fase anterior) se agregaron en la organización lineamientos técnicos formales sobre las técnicas de codificación segura. Estos fueron documentos técnicos específicos relacionados a los lenguajes y tecnología, ellos proveen guía sobre las técnicas de codificación segura en cada lenguaje/aplicación relevante.

Con una solución combinada de programas de educación y conscientización, las guías técnicas y la introducción de herramientas automatizadas para ayudar a los desarrolladores, VirtualWare inicio a ver una diferencia visible en el código que estaba siendo publicado en las versiones de producción de sus aplicaciones. Los desarrolladores dieron retroalimentación positiva sobre las herramientas y la educación puesta a su disposición en este programa.

Por primera vez el equipo de proyectos de VirtualWare se hizo responsable por la seguridad en el diseño de sus plataformas de aplicación. Durante esta fase se realizó una revisión formal del proceso de revisión y validación contra las mejores prácticas realizadas por cada equipo. Algunos equipos identificaron fallas relativas a la seguridad y el diseño de negocio que necesitaban ser revisadas. Se realizó un plan formal para asegurarse que las fallas fueran reparadas.

Durante esta fase del proyecto se agregó un plan formal de respuesta a incidentes y procedimientos de manejo de cambio. Este fue un proceso de implementación difícil y los equipos de VirtualWare batallaron inicialmente con el proceso dado que el impacto en la cultura y el lado operativo fue significativo. Sin embargo, con el tiempo, cada miembro de equipo identifico el valor en el nuevo proceso y los cambios fueron aceptados por el equipo durante el periodo de implementación.

## COSTOS DE IMPLEMENTACIÓN

En esta fase del proyecto un monto significativo fue invertido en recursos internos y otros costos. Hubo 2 tipos diferentes de costos asociados a esta fase.

### Requisitos de Recursos Internos

El esfuerzo de los recursos internos fue usado en la creación de contenidos, talleres y revisión de iniciativas de seguridad para aplicaciones. El esfuerzo se muestra en días totales por rol.

Desarrollador	5 días	Responsable del negocio	6 días
Arquitecto	7 días	Auditor de seguridad	10 días
Gestor	9 días	Soporte a operaciones	3 días

### Recursos Externos

Dada la falta de conocimiento en VirtualWare, se usaron recursos externos para ayudar con la creación de contenido, ayudar a los equipos, crear los procesos y lineamientos.

Consultor de seguridad	20 días
------------------------	---------

## FASE 4 (MESES 9-12) – GOBIERNO Y SEGURIDAD OPERATIVA

La cuarta fase de la implementación del programa de aseguramiento en VirtualWare continúa desde las fases anteriores al mejorar las funciones existentes de seguridad en la organización. Hasta ahora VirtualWare ha implementado varios procesos y mecanismos importantes en la seguridad de aplicaciones para asegurarse que las aplicaciones son desarrolladas y mantenidas de manera segura.

El enfoque principal en esta fase es en impulsar las bases de una estrategia de seguridad en aplicaciones efectiva y de largo plazo. Se implemento un programa de educación completo, al mismo tiempo se creó plan de implementación estratégico de largo plazo para VirtualWare.








El otro elemento clave en esta fase es la parte operativa de la implementación. Los ejecutivos de VirtualWare identificaron la necesidad de un plan de respuesta a incidentes y procesos de administración del cambio en la estrategia de largo plazo.

VirtualWare vio esta fase como las bases de su futuro a largo plazo. En esta fase la organización implemento varias medidas finales para cimentar los bloques de construcción existentes que han sido puestos en las fases anteriores. En el largo plazo esto asegurará que los procesos, conceptos y controles implementados continúen funcionando en la organización para asegurar los resultados más seguros para sus plataformas de aplicación.

VirtualWare escogió para esta fase el introducir a sus clientes a sus nuevas iniciativas de seguridad proveer detalles de una serie de programas a los clientes de VirtualWare sobre seguridad en aplicaciones, publicación segura de aplicaciones y reporte de vulnerabilidades en las aplicaciones de VirtualWare. La meta principal de estos programas es promover la confianza dentro la base de clientes y convencerlos de que las aplicaciones de VirtualWare están construidas con seguridad en mente y que VirtualWare puede ayudar a sus clientes al asegurar que los ambientes de aplicación de su tecnología son seguros.

### OBJETIVOS PRINCIPALES

Durante esta fase del proyecto, VirtualWare implementó las siguientes Prácticas y Actividades de SAMM.

 <b>SM 3</b>	A. Realizar comparaciones de costo periódicas a nivel industria B. Recolectar métricas históricas de gastos de seguridad
 <b>PC 2</b>	A. Crear políticas y estándares para seguridad y cumplimiento B. Establecer la práctica de auditoría de proyecto
 <b>EG 3</b>	A. Crear un portal formal de soporte de seguridad en aplicaciones B. Establecer exámenes o certificaciones por rol
 <b>SR 3</b>	A. Incorporar los requisitos de seguridad a acuerdos con proveedores B. Ampliar el programa de auditoría para los requisitos de seguridad
 <b>CR 3</b>	A. Personalizar el análisis de código para las preocupaciones específicas de la aplicación B. Establecer puntos de control para la liberación de las revisiones de código
 <b>VM 3</b>	A. Conducir análisis de causa raíz para incidentes B. Recolectar métricas por incidente
 <b>OE 3</b>	A. Expandir el programa de auditoría para información operativa B. Realizar firma de código para componentes de aplicaciones

Para alcanzar estos niveles de madurez implementó varios programas durante esta fase de la implementación. Las siguientes iniciativas fueron adoptadas:

- ◆ Crear requisitos de seguridad bien definidos y un programa de pruebas para todos los proyectos
- ◆ Crear e implementar un plan de respuesta a incidentes
- ◆ Revisar los procedimientos existentes de alerta para aplicación y documentar un proceso para capturar estos eventos
- ◆ Crear una publicación sobre liberación segura de aplicaciones para los clientes
- ◆ Revisar los gastos en seguridad actuales en los proyectos y determinar si el presupuesto de seguridad adecuado ha sido asignado para cada proyecto
- ◆ Implementar programas finales de educación y de conocimiento para los diferentes roles de aplicación
- ◆ Completar para la organización un plan de implementación estratégico a largo plazo sobre seguridad en aplicaciones

En fases anteriores VirtualWare publicó un plan formal de respuesta a incidentes para recibir las vulnerabilidades encontradas en su código. Durante esta fase, VirtualWare tomó los resultados de las vulnerabilidades enviadas y condujo revisiones de porque y como ocurrió el problema. Creó una serie de reportes para determinar cualquier tema en común entre las vulnerabilidades reportadas.

Como parte del esfuerzo continuo para asegurar que las aplicaciones son publicadas de manera segura internamente así como en las redes de los clientes, VirtualWare creó una serie de publicaciones basadas en los estándares de la industria, se proveyeron a los clientes y se recomendó que mejoraran sus ambientes. El propósito de los lineamientos es proveer ayuda a los clientes sobre cual es la mejor solución cuando publiquen sus aplicaciones.

Durante esta fase, VirtualWare implementó un corto entrenamiento basado en computadora de manera que los desarrolladores nuevos y existentes puedan mantener sus habilidades de seguridad en aplicaciones. Se exigió también que todos los roles asociados a aplicaciones tomen un curso obligatorio de 1 día al año. Esto se hizo para asegurar que las habilidades dadas a los desarrolladores no se pierdan y que los nuevos desarrolladores estén bien entrenados durante el tiempo que estén con la compañía.

Una de las funciones finales implementadas en VirtualWare fue completar y revisar el análisis de fallas “tal cual” (AS IS) y determinar que tan efectivos habían sido los últimos 12 meses. Durante este corto programa se enviaron cuestionarios a todos los miembros del equipo involucrados, también se hizo una revisión comparativa contra el SAMM. Las debilidades y fortalezas identificadas durante la revisión fueron documentadas en un plan estratégico de implementación de la organización y así se fijó la estrategia para los próximos 12 meses para VirtualWare.

## COSTOS DE IMPLEMENTACIÓN

En esta fase del proyecto un monto significativo fue invertido en recursos internos y otros costos. Hubo 2 tipos diferentes de costos asociados a esta fase.

### Requisitos de Recursos Internos

El esfuerzo de los recursos internos fue usado en la creación de contenidos, talleres y revisión de iniciativas de seguridad en aplicaciones en esta fase. El esfuerzo se muestra en días totales por rol.

Desarrollador	4 días	Responsable del negocio	6 días
Arquitecto	7 días	Probador QA	1 días
Gestor	9 días	Auditor de seguridad	11 días

### Recursos Externos

Dada la falta del conocimiento dentro de VirtualWare, se usaron recursos externos para ayudar con al implementación de esta fase, incluyendo la documentación, procesos y talleres.

Consultor de seguridad	22 días
------------------------	---------

## SEGUIMIENTO (MESES 12+)

En los pasados 12 meses VirtualWare ha iniciado a implementar varios programas de entrenamiento y educación para desarrollar lineamientos y políticas. En la fase final de implementación del programa de aseguramiento, VirtualWare empezó a publicarlo externamente y a trabajar con sus clientes para mejorar la seguridad de la plataforma de aplicación de sus clientes.

Los ejecutivos de VirtualWare fijaron un lineamiento para asegurar que el software desarrollado en la compañía era seguro, que el mercado sabía de las iniciativas de seguridad tomadas y a ayudar a sus clientes a asegurar sus plataformas de aplicación.

Para alcanzar estas metas administrativas los primeros doce meses fijaron el camino a seguir para una estrategia efectiva en VirtualWare y finalmente empezó a ayudar a sus clientes a asegurar sus ambientes de aplicación. Después VirtualWare creó varias iniciativas en la organización para asegurar que la compañía no regresara a los viejos hábitos. Algunos de estos programas incluyen:

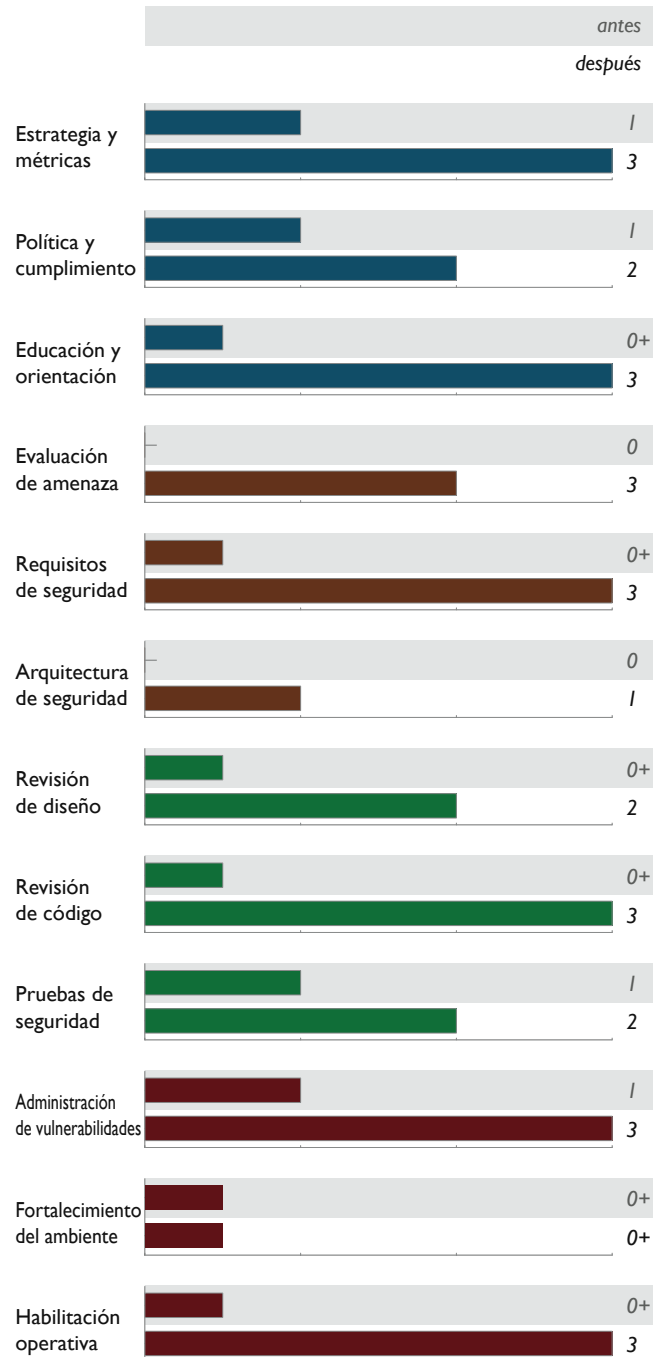
- ◆ Los dueños de negocio y líderes de equipo conocen el riesgo asociado con sus aplicaciones y requieren aprobarlas antes de su publicación.
- ◆ Los líderes de equipo necesitan pasar todas sus aplicaciones por un proceso formal de seguridad y los desarrolladores realizan revisiones de código semanales.
- ◆ Entrenamientos periódicos y anuales (incluyendo los CBTs) se proveen a todo el personal de los proyectos y los desarrolladores tienen que atender un curso por lo menos una vez al año.
- ◆ Un líder de seguridad en aplicaciones dedicado fue creado y es responsable ahora de las comunicaciones con el clientes y de las publicaciones y guías técnicas para los clientes.

Además VirtualWare ahora tiene una cultura de seguridad como parte de su ciclo de desarrollo de software, por lo que se asegura de que las aplicaciones desarrolladas y proveídas a los clientes son seguras y robustas. Un proceso efectivo se ha implementado donde las vulnerabilidades se pueden reportar y son administradas por la organización cuando se requiere.

Durante la fase final de implementación se realizó un análisis de carencias en los proyectos para identificar cualquier debilidad que apareciera durante la implementación. En particular dada la alta rotación del personal, VirtualWare necesitó de entrenar constantemente a los nuevos desarrolladores conforme iniciaban labores con la organización. Como objetivo principal para manejar este problema está el programa de inducción, que se agregó específicamente para los desarrolladores de manera que recibieran entrenamiento formal de seguridad cuando iniciaban con la organización. Esto también ayuda a crear una mentalidad de que la seguridad es importante dentro de la organización y su equipo de desarrollo.

## TARJETA DE CALIFICACIONES DE MADUREZ

La tarjeta de calificaciones de madurez se completó como una forma de autoevaluación durante la implementación del programa de aseguramiento para VirtualWare. El tarjeta de calificaciones final (mostrado a la derecha) representa el estado de VirtualWare en el momento que empezó y cuando termino este proyecto de mejora de cuatro fases.



**PARA OBTENER MAYOR INFORMACIÓN, POR FAVOR VEA EL SITIO DEL PROYECTO EN:**

<http://www.opensamm.org>

### **AUTOR Y LÍDER DE PROYECTO**

Pravir Chandra

### **CONTRIBUIDORES Y REVISORES**

Fabio Arciniegas

Matt Bartoldus

Sebastien Deleersnyder

Jonathan Carter

Darren Challey

Brian Chess

Dinis Cruz

Justin Derry

Bart De Win

James McGovern

Matteo Meucci

Jeff Payne

Gunnar Peterson

Jeff Piper

Andy Steingruebl

John Steven

Chad Thunberg

Colin Watson

Jeff Williams

### **PATROCINADORES**

Gracias a las siguientes organizaciones que han hecho contribuciones significativas a el proyecto SAMM.



### **PARTIDARIOS**

Gracias a las siguientes organizaciones por ayudar a revisar y apoyar el proyecto SAMM.

*Nota: OWASP y el proyecto SAMM no endosa ningún producto o servicio comercial*



### **LICENCIA**



Este trabajo se publica bajo la licencia Creative Commons Attribution-Share Alike 3.0. Para ver una copia de la licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.