Caelum
information and quality technologies

CMMI Institute Partner

Congreso CSTIC 2017

Be Agile, Be digital, Be Secure

Organiza: AEC
ASOCIACIÓN ESPAÑOLA PARA LA CALIDAD

Caelum
information and quality technologies
CMMI Institute Partner

Cómo mejorar la seguridad del software

Ramiro Carballo Gutiérrez
SCAMPI Lead Appraiser

CSTIC 2017
Be Agile, Be Digital, Be Secure

Quienes somos?

Caelum
information and quality technologies

CMMI Institute Partner

Partner del CMMI Institute (ISACA) en España:

CMMI DEV

CMMI SVC

CMMI ACQ

DATA MANAGEMENT MATURITY MODEL

ESTIMACIÓN DE PROYECTOS

ISO 15504 / ISO 12207

ISO 20000 / ITIL

ISO 27.001

MAGERIT V3

ESQUEMA NACIONAL DE SEGURIDAD

CSTIC 2017

Be Agile, Be Digital, Be Secure

Ciclo de Vida del Software Seguro

- Ciclo de Vida:
  - Secuencia de actividades de la vida de un producto desde su concepción hasta su retirada.
- Ciclo de Vida de Desarrollo de Software:
  - Etapas de la vida de una aplicación software, desde que el usuario expone sus necesidades, hasta que se despliega en el entorno de producción.
- Ciclo de Vida de Desarrollo de Software Seguro:
  - Integración de las actividades de desarrollo de software con los puntos de chequeo de seguridad y aquellas cuestiones que eviten que el software entregado incorpore vulnerabilidades.

Prácticas seguras del SSDL

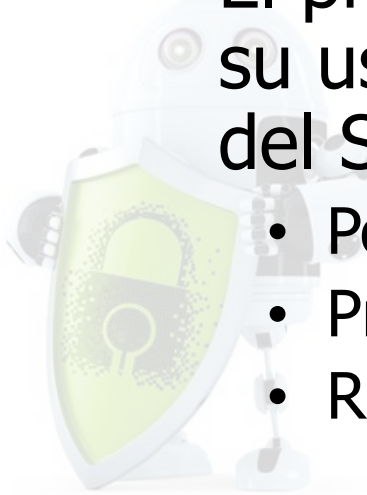
- Tres prácticas del ciclo de vida de desarrollo de software seguro (SSDL):
  - Análisis de la Arquitectura del Software
    - Representar la arquitectura del software con diagramas que faciliten identificar las amenazas y los riesgos, y construir un plan para reducirlos.
  - Revisiones de Código Fuente
    - Utilizar herramientas de análisis de código que apliquen reglas adaptadas al contexto de nuestras aplicaciones y hacer seguimiento de los resultados obtenidos.
  - Realización de Pruebas de Seguridad
    - Probar las vulnerabilidades introducidas durante la construcción del código, antes de realizar la entrega de las aplicaciones software.

Análisis de Arquitectura Software (I)

- Revisar las funcionalidades de seguridad de la arquitectura. Ej.
  - Autenticación
  - Control de acceso
  - Criptografía
- Realizar una revisión del diseño de las aplicaciones de más alto riesgo de la organización, detectar los defectos principales y elaborar un plan para resolverlos.
- El grupo de Seguridad del Software (SSG) debe liderar las revisiones de la arquitectura, con su experiencia en seguridad y con el apoyo de los arquitectos y desarrolladores del software revisado.

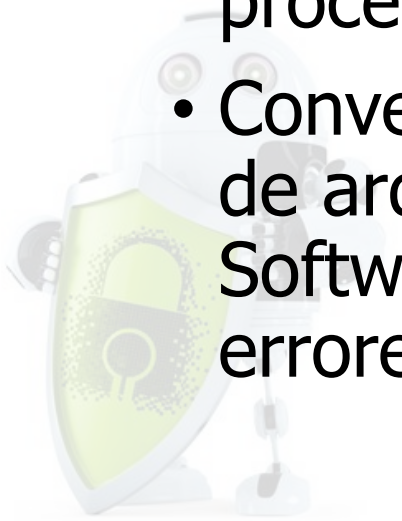
Análisis de Arquitectura Software (II)

- Utilizar un cuestionario de riesgos para facilitar la revisión de las funcionalidades de seguridad y el diseño de la aplicación . Ej.
  - Lenguaje de programación
  - Usuarios potenciales
  - Entorno móvil
- El proceso de análisis de arquitectura debe estar definido para su uso fácil incluso para técnicos ajenos al grupo de Seguridad del Software (SSG) y así ayudar a enfocar las revisiones hacia:
  - Posibles ataques
  - Propiedades de seguridad
  - Riesgos asociados



Análisis de Arquitectura Software (III)

- Estandarizar la documentación de los diseños: flujos de datos, diagramas UML, ayuda a tener una imagen clara de los activos de información que se deben proteger.
- Que el grupo de Seguridad del Software (SSG) soporte y sea el mentor del resto de equipos de desarrollo en la aplicación del proceso de Análisis de Arquitectura.
- Convertir los resultados del Análisis de Arquitectura en patrones de arquitectura estándar, en los que un Comité de Diseño de Software Seguro se ha preocupado de evitar que se produzcan errores similares en el futuro.



Revisión de Seguridad del Código (I)

- Se deben realizar revisiones de código por parte del Grupo de Seguridad del Software (SSG), para aplicaciones de alto riesgo.
  - Más adelante, otros miembros del equipo de desarrollo pueden hacer estas revisiones
  - Se pueden apoyar en herramientas o ser revisiones manuales.
  - Se debe evolucionar según avanza la tecnología revisada.
- Obligar a que se revise el código de todos los proyectos:
  - Las entregas se bloquearán si un proyecto no se ha revisado o si el resultado de la revisión fue negativo.
  - Se puede mezclar el uso de herramientas automáticas para los proyectos de bajo riesgo con el uso de revisiones manuales para los de riesgo mayor.

Revisión de Seguridad del Código (II)

- Generar informes centralizados para aprender de los resultados.
  - Tener un repositorio centralizado de errores de seguridad.
  - Que permita realizar informes de tendencia a nivel de organización.
  - Que permita reconocer las mejoras en la seguridad global
  - Los informes de revisión de código se pueden cruzar con otras medidas del ciclo de vida del desarrollo seguro de software relacionadas con:
    - Pruebas de penetración
    - Pruebas de seguridad
    - Pruebas de caja negra
    - Pruebas de caja blanca

Revisión de Seguridad del Código (III)

- Objetivos para "NOTA" en la revisión de seguridad del código:
  - Publicar la lista de los 10 defectos más buscados "de esta casa". Algunos métodos como OWASP no suele reflejar lo que más le preocupa a una empresa en concreto.
  - Montar una "factoría" de revisores – correctores de código. Con motores de análisis estático combinado con dinámico, en un flujo industrial de proceso de revisión.
  - Automatizar la detección de código malicioso que pueden elaborar nuestros propios desarrolladores, mediante el reconocimiento de patrones de código, para lo que el análisis manual se queda corto.
  - Reforzar el uso de estándares de codificación, de manera preventiva.

Pruebas de Seguridad (I)

- Asegurar que las pruebas de QA analizan los valores límites de los rangos de las condiciones:
  - Es ir más allá de las pruebas puramente funcionales
  - Hay que pensar como los malos y probar valores con mala intención.
  - ¿Qué pasa que intentas entrar con la contraseña incorrecta una vez? ¿y otra? ¿y otra…?
- Hay que dirigir las pruebas a
  - Los requisitos de seguridad: ¿puedo acceder a algo sin tener permiso?
  - Y a las funcionalidades de seguridad.
  - Las funciones de nuevos modelos de servicios en la nube…

Pruebas de Seguridad (II)

- Mezclar las pruebas de calidad con las pruebas de seguridad:
  - Integrando las pruebas de caja negra en el proceso de aseguramiento de la calidad.
  - Haciendo que las pruebas las ejecuten desde calidad, pero se cuente con el Grupo de Seguridad del Software (SSG) para interpretar los resultados.
  - Compartiendo los resultados de seguridad con los equipos de calidad, que tienen una forma de trabajar metódica y orientada a la mejora continua.
  - Incluyendo casos de prueba de seguridad dentro de la ejecución automática de las pruebas de QA.

Pruebas de Seguridad (III)

- Para sacar "NOTA" en las pruebas de seguridad:
  - Enfocar las pruebas a lo que nos diga el análisis de riesgos:
    - Según el resultado del Análisis de la Arquitectura que habíamos procedimentado.
    - Priorizando sobre los componentes que presentan los riesgos más altos.
  - Darle más importancia a la cobertura de las pruebas:
    - Medir la cobertura de código de las pruebas de seguridad nos permite conocer la proporción de código que no se ha probado.
    - Podemos utilizar el indicador de cobertura para conseguir que las pruebas de seguridad alcancen mayor profundidad.
    - Se demuestra que las pruebas de caja negra tienen una cobertura muy baja.

Más allá del ciclo de vida SSDL (I)

- Si nos quedamos en
  - Análisis
  - Diseño
  - Código
  - Pruebas
- Hemos dejado la seguridad en manos de los desarrolladores.

¿¡ SON LOS HÉROES DE LA SEGURIDAD DEL SOFTWARE !?

Métrica v2.1 sólo habla de ingeniería -> swCMM - Métrica v3

¿QUIERO TENER UNA EMPRESA DE HÉROES?

Más allá del ciclo de vida SSDL (II)

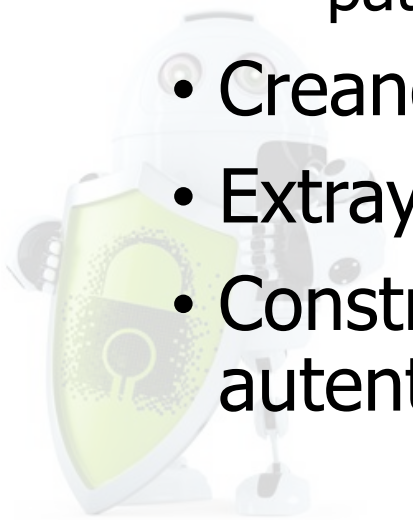
- Se necesita el apoyo de toda la organización mediante:
  - La entrega segura del software.
  - La inteligencia relacionada con la seguridad del software:
  - El gobierno de la seguridad del software.

La entrega segura de software

- La entrega de software debe asegurar:
  - Que los expertos en seguridad realizan pruebas de penetración enfocadas en explotar vulnerabilidades de la entrega final.
  - Que se realiza sobre un entorno seguro:
    - Con un sistema operativo y una plataforma adecuadamente parcheada
    - Con los firewalls adecuados para las aplicaciones web
    - Con documentación de configuración e instalación
    - Con monitorización de la disponibilidad de las aplicaciones, gestión del cambio,…
  - Con gestión de la configuración y de las vulnerabilidades
    - Para asegurar que la última versión no contiene una vulnerabilidad que ya habíamos corregido.
    - Con gestión de incidencias, seguimiento de defectos, control de versiones, etc.

Inteligencia de la seguridad

- Creando modelos de ataque basando en información recogida de cómo piensan los atacantes:
  - modelado de amenazas
  - desarrollo y refinamiento de casos de abuso
  - clasificación de datos y
  - patrones de ataque específicos de cada tecnología.
- Creando patrones de diseños y funcionalidades seguros.
- Extrayendo los requisitos de seguridad de la organización.
- Construyendo estándares sobre las funciones principales: autenticación, validación de entradas, etc.



Gobierno de seguridad del software

- Mediante una estrategia de mejora de la seguridad del software en la que
  - Se fijan unos objetivos de seguridad del software
  - Se elabora un plan, con recursos, plazos,
  - Se identifican las metas y los indicadores para seguirlos.
- Identificando los controles para asegurar el cumplimiento de compromisos, contratos, estándares, SLAs, siguiendo una política de seguridad del software y auditándola.
- Proporcionando formación sobre seguridad a los equipos de desarrollo.

Muchas gracias.

Ramiro Carballo

CMMI SCAMPI Lead Appraiser

Tlf. (+34) 639078817

e-mail: rcarballo@caelum.es

https://www.linkedin.com/in/ramirocarballo/

Twitter: @rcarballo_CMMI