Seguridad Informática

Tema 9 – Contramedidas de usuario, administrador y desarrollador

Universidad Rey Juan Carlos

Isaac Lozano Osorio **Isaac.lozano@urjc.es**

30/04/2024

Índice

Universidad Rey Juan Carlos

- Buenas prácticas para el usuario y administrador.
- Buenas prácticas para el desarrollador.
- Metodologías de desarrollo seguro.

Índice

Universidad Rey Juan Carlos

- **Buenas prácticas para el usuario y administrador.**
- Buenas prácticas para el desarrollador.
- Metodologías de desarrollo seguro.

Buenas prácticas (usuario)

Universidad Rey Juan Carlos

- **Usuario**:
  - Uso de contraseñas seguras (gestores de contraseña).
  - Uso de antivirus.
  - Usar aplicaciones y sistemas operativos que estén actualizados.
- **Administrador**:
  - Hay que establecer seguridad física.
  - Formar y concienciar a los usuarios.
  - Control de acceso y autenticación.
  - Permitir únicamente acceso remoto encriptado.
    - Sistemas asimétricos.
    - Sistemas simétricos.

Buenas prácticas (usuario)

Universidad Rey Juan Carlos

- Forzar la utilización de contraseñas seguras.
- Controlar el envejecimiento y la expiración de estas contraseñas.
- Uso de Shells restringidas.
- Establecer gestión segura de cuentas y filosofía del mínimo privilegio.
- Utilización de perfiles y roles, cuentas especiales, eliminación de cuentas obsoletas.
- Encriptación a diferentes niveles.

Índice

Universidad Rey Juan Carlos

- Buenas prácticas para el usuario y administrador.
- **Buenas prácticas para el desarrollador.**
- Metodologías de desarrollo seguro.

Buenas prácticas (desarrollador)

Universidad Rey Juan Carlos

- Comprobar los operandos antes de utilizarlos.
- Comprobar los rangos antes de realizar conversiones de tipos.
- No ignorar valores de retorno ni excepciones.
- Validación de datos de entrada:
  - Normalización de cadenas y de datos.
  - Canonización de rutas, rutas que sólo incluyan caracteres alfanuméricos.
- Evitar usar variables públicas de clases, interfaces, paquetes, etc.

Buenas prácticas (desarrollador)

Universidad Rey Juan Carlos

- No mostrar información sensible en las excepciones.
- Ni en el código, se pueden emplear desensambladores y des-compiladores para localizarla.
- Borrar ficheros temporales.
- Liberar recursos cuando no se necesitan.
- Generar números aleatorios “fuertes”.
- Fundamental: conocer en profundidad el lenguaje de programación utilizado.

Índice

- Buenas prácticas para el usuario y administrador.
- Buenas prácticas para el desarrollador.
- **Metodologías de desarrollo seguro.**

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- Existen analizadores estáticos de código que son capaces de detectar las vulnerabilidades más extendidas:

FlawFinder	PMD	LAPSE (OWASP)
Sonar Cloud	AppScan (IBM)	Fortify (HP)

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- También existen diferentes guías que recopilan las mejores prácticas para el desarrollo de aplicaciones seguras en diferentes lenguajes de programación:
  - ISO: Programming languages – Guide for the Use of the Ada Programming Language in High Integrity Systems.
  - MISRA: Guidelines for the use of the C language in critical systems.
  - CERT: C Secure Coding Standard, Secure Coding in C and C++ y Oracle Secure Coding Standard for Java.
  - OWASP: Secure Coding Practices Quick Reference Guide.

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

Top 10 de mejores prácticas para el desarrollo seguro según CERT:

1. Validar las entradas de fuentes de datos no fiables.
2. Ojo a los warnings.
3. Hay que diseñar una arquitectura orientada a las políticas de seguridad.
4. Regla KISS.
5. Negación por defecto.
6. Principio del privilegio mínimo.
7. Sanear los datos que se mandan a otros subsistemas.
8. Practicar la defensa en profundidad.
9. Usar técnicas eficaces de garantía de calidad.
10. Adoptar un estándar de codificación segura.

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- Con el análisis estático o los check-lists que comprueban la aplicación de esas mejores prácticas, pero no se ejecuta el código que se está analizando.
  - Pruebas de caja blanca.

- Las técnicas de análisis dinámico de código sí que lo ejecutan, por lo que se monitoriza su comportamiento para ver si supera un conjunto de requisitos (verificación formal).
  - La ventaja frente al análisis estático es que se está probando el programa en un escenario de ejecución real, donde pueden aparecer vulnerabilidades/debilidades no detectados en el análisis estático.

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- Pero cuando se trata de metodologías completas, que tengan en cuenta todo el ciclo de vida del software y no sólo la codificación, tenemos:

Building Security in Maturity Model

OWASP Software Assurance Maturity Model (SAMM)

Microsoft Security Development Lifecycle (SDL)

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

Building Security in Maturity Model (BSIMM- 11)

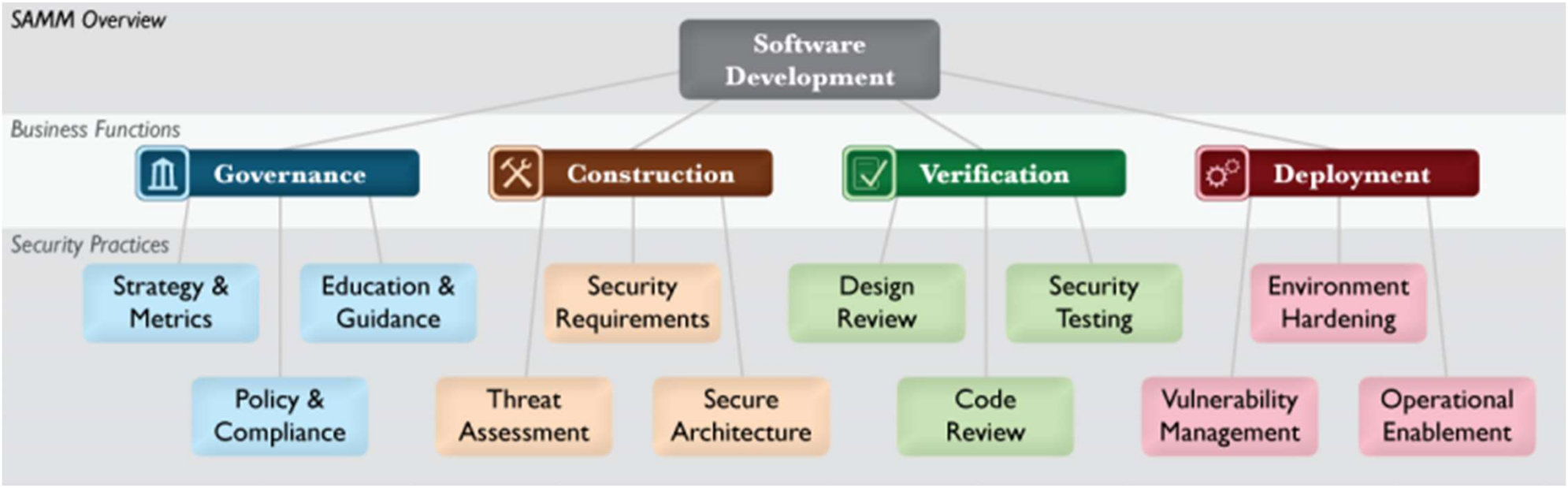
- Estudio de las iniciativas de desarrollo de software seguro.
- Ampliamente adoptado por las empresas: Google, Visa, Intel…

- Determina cuál es la situación de la empresa respecto a las actividades de desarrollo de software seguro
- Cómo puede mejorar las acciones en esa área.

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

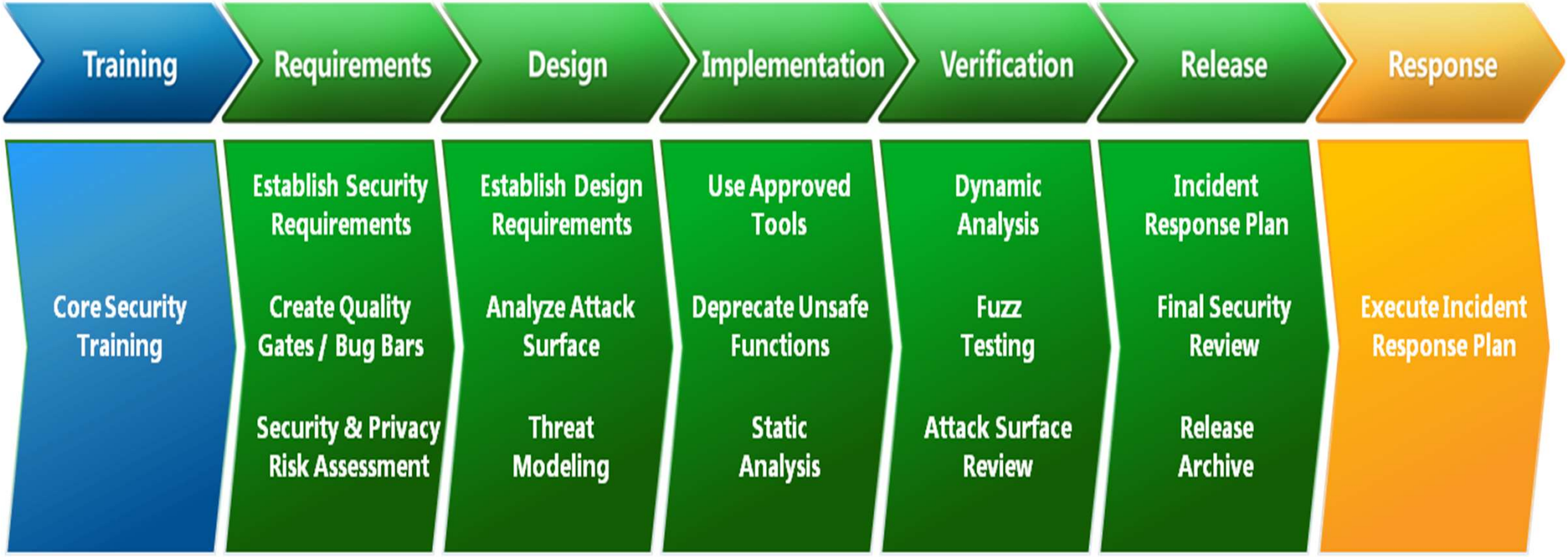
- **OWASP SAMM**



Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- **Security Development Lifecycle (SDL)**



Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- Aspectos comunes:

Procesos repetibles

Resultados medibles

Trazabilidad

Formación y concienciación de los desarrolladores

SD3+C

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

SD3+C

Security by Default

Secuity by Design

Security in Deployment

Communication/ Community

Metodologías de desarrollo seg.

Universidad Rey Juan Carlos

- El desarrollo seguro es un reto importante en la actualidad en el caso de las aplicaciones para móviles.
- Las metodologías que se usan en el desarrollo de aplicaciones tradicionales o web no sirven, porque existen problemas muy específicos.
- Existen diferentes proyectos, que identifican las vulnerabilidades y debilidades más importantes para después proponer nuevas metodologías.