OWASP SAMM

Procesos para ingeniería de datos.

Introducción

OWASP SAMM (Software Assurance Maturity Model) es un marco de trabajo que ayuda a las organizaciones a **evaluar, formular e implementar estrategias enfocadas en seguridad de software**, desde una perspectiva del Software Development Lifecycle (SDLC) existente.

El modelo de maduración OWASP SAMM está hecho para dar soporte a:

1. **Evaluar** el estado actual del software.
2. Definir la **estrategia** (**objetivo**) que la organización debe implementar.
3. Formular un **plan de ejecución** con los objetivos y metodología.
4. Consejos prescriptivos en cómo **implementar actividades** particulares.

Introducción

OWASP SAMM provee un medio para **conocer el estado actual de una organización respecto al aseguramiento de software**, y conocer recomendaciones para alcanzar un nivel de madurez mayor.

OWASP SAMM es lo suficientemente flexible para **determinar el nivel de madurez y prácticas de seguridad** de la organización basado en sus necesidades, características, experiencia, tamaño y conocimiento técnico.

Introducción

SAMM se compone de **15 prácticas de seguridad** organizadas en **5 funciones comerciales**. Cada práctica incluye **actividades divididas en 3 niveles de madurez (inicial, definido y optimizado)**. Las actividades de menor nivel de madurez son más sencillas de implementar y requieren menor formalización que las de niveles superiores.

Descripción general

Cada función comercial es una **categoría de actividades que cualquier organización de desarrollo de software debe cumplir** en algún nivel.

Cada función comercial tiene tres prácticas de seguridad, las cuáles son **áreas con actividades relacionadas a la seguridad**, que impactan la función comercial.

Streams **cubren distintos aspectos de una práctica de seguridad**, tiene sus propios objetivos, permiten **relacionar las actividades según el nivel de madurez**.

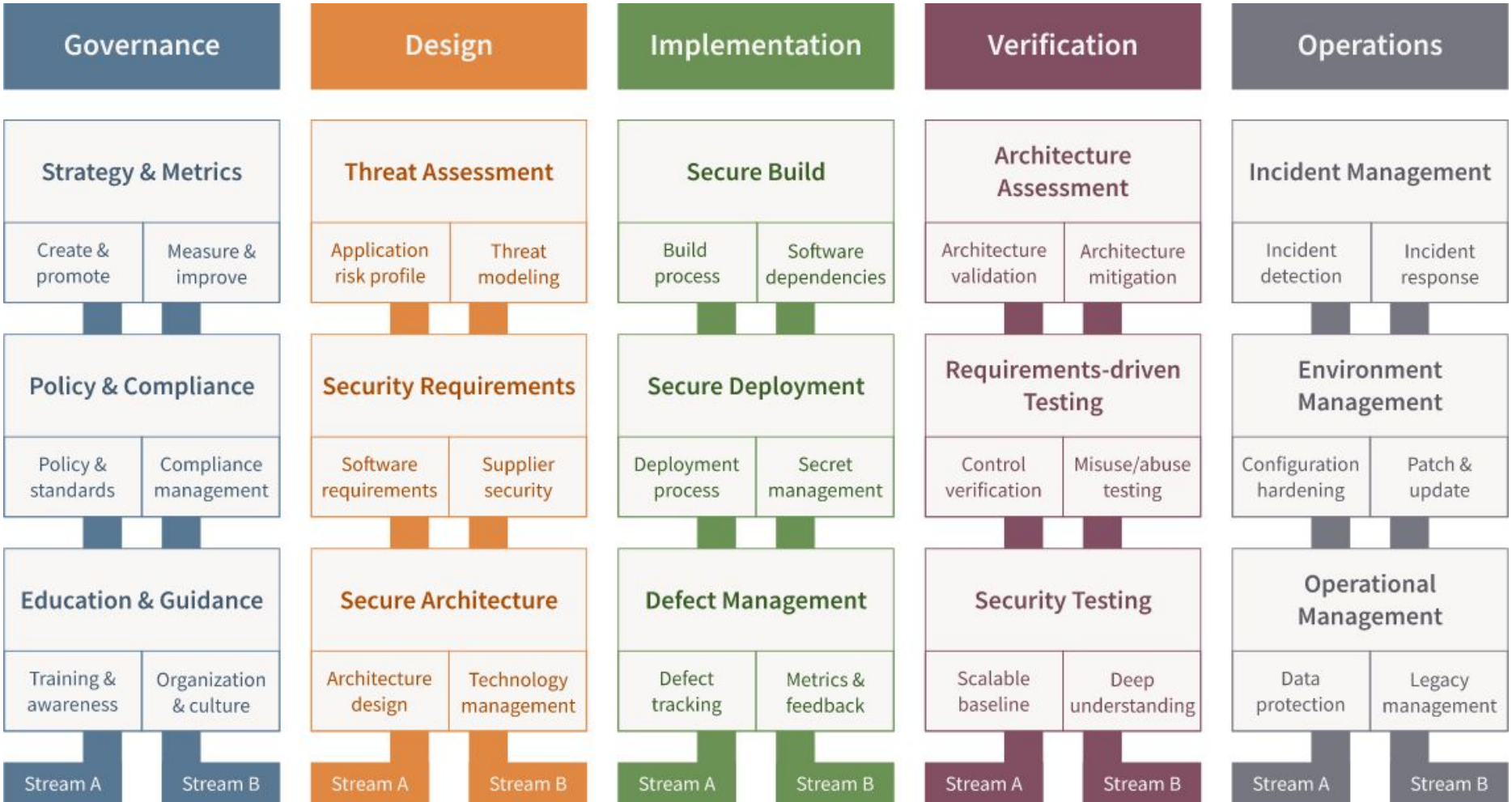
Funciones comerciales

Governance	Design	Implementation	Verification	Operations
Gestión de actividades de desarrollo de software desde una perspectiva de gobierno (políticas, gestión de riesgos, supervisión)	Planificar y diseñar sistemas de software (requisitos y arquitectura general)	Construcción y despliegue de software (secretos, dependencias de terceros)	Verificar la seguridad del software (pruebas, revisiones de código y evaluaciones de seguridad)	Gestión y el mantenimiento de la seguridad del software una vez que está en producción (protección de datos, actualizaciones)

Prácticas de seguridad

Governance	Design	Implementation	Verification	Operations
• Educación y orientación • Políticas y cumplimiento • Métricas y estrategias	• Arquitectura segura • Requisitos de seguridad • Evaluación de amenazas	• Gestión de defectos • Despliegue seguros • Construcción segura	• Pruebas de seguridad • Pruebas basadas en requisitos • Evaluación de arquitectura	• Gestión operacional • Gestión de ambientes • Gestión de incidentes

Modelo


Governance
Strategy & Metrics
Create & promote
Measure & improve
Policy & Compliance
Policy & standards
Compliance management
Education & Guidance
Training & awareness
Organization & culture
Stream A
Stream B
Design
Threat Assessment
Application risk profile
Threat modeling
Security Requirements
Software requirements
Supplier security
Secure Architecture
Architecture design
Technology management
Stream A
Stream B
Implementation
Secure Build
Build process
Software dependencies
Secure Deployment
Deployment process
Secret management
Defect Management
Defect tracking
Metrics & feedback
Stream A
Stream B
Verification
Architecture Assessment
Architecture validation
Architecture mitigation
Requirements-driven Testing
Control verification
Misuse/abuse testing
Security Testing
Scalable baseline
Deep understanding
Stream A
Stream B
Operations
Incident Management
Incident detection
Incident response
Environment Management
Configuration hardening
Patch & update
Operational Management
Data protection
Legacy management
Stream A
Stream B

Ejemplo de implementación

Función comercial: Diseño

Práctica de seguridad: Arquitectura segura

Nivel de madurez: 1 - Integrar guías de seguridad proactivas al proceso de diseño de software

Stream: B - Gestión de tecnología

Ejemplo de implementación

Diseño / Arquitectura segura / Nivel 1 / Stream B - Gestión de arquitectura

Beneficio: Transparencia en las tecnologías que agregan riesgos de seguridad.

Actividad: Identificar y evaluar las tecnologías, frameworks, herramientas e integraciones más importantes que se usan en cada aplicación.

Pregunta: ¿Se evalúa la calidad desde el punto de vista de seguridad de las tecnologías empleadas?

Criterios de calidad:

- Se tiene una lista con las tecnologías más importantes para cada aplicación
- Se identifica y da seguimiento a los riesgos de esas tecnologías
- Se asegura que los riesgos de esas tecnologías concuerdan con los estandares de la organización

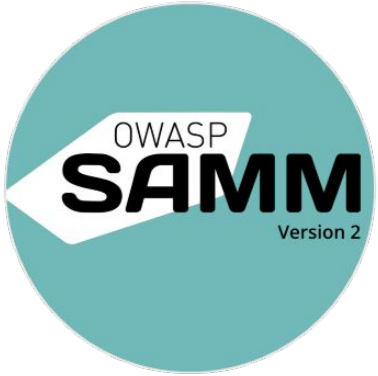
No / Si en algunas aplicaciones / Si en al menos la mitad de las aplicaciones/ Si para todas

Casos de uso


Desarrollo de
AplicacionesWeb


Mobile Banking
Login:
Password:


DESARROLLO DE
APLICACIONES
MÓVILES


OWASP
SAMM
Version 2


ProduktZero
PLATAFORMA DIGITAL
DE GOBIERNO


LEARN ONLINE