

SDLC

Desarrollo Seguro y Modelado de Amenazas aplicado al Ciclo de Vida del Desarrollo del Software

El modelado de amenazas es una técnica formal, estructurada y repetible que permite determinar y ponderar los riesgos y amenazas a los que estará expuesta una aplicación y que además, puede aplicarse al Ciclo de Vida del Desarrollo de Software (SDLC).

Objetivos generales

Este curso está orientado a conocer los principales modelos de desarrollo de software y modelado de amenazas para poder abordar los principales tipos de vulnerabilidades existentes así como la forma de analizarlas, explotarlas y solucionarlas desde el inicio de cualquier proyecto de desarrollo.

Requisitos previos

Conocimiento básico sobre análisis de sistemas y programación.

Público Objetivo

Desarrolladores, administradores de bases de datos, líderes de proyecto, analistas de sistemas y personas vinculadas al análisis, diseño, arquitectura y desarrollo de aplicaciones.

Finalizado el curso, el asistente será capaz de:

- Entender el ciclo de desarrollo seguro de aplicaciones
- Entender las amenazas y vulnerabilidades a las que está expuesta cualquier aplicación
- Reconocer y detectar de manera proactiva las falencias del desarrollo de las aplicaciones
- Diseñar y desarrollar modelos de amenazas sobre el ciclo de vida del software
- Desarrollar aplicaciones seguras y orientadas a evitar vulnerabilidades

Este curso complementa el otro curso **“OWASP: Desarrollo Seguro: práctica orientada a prevención de OWASP Top 10”** dictado por **Segu-Info**.

Material a entregar

Presentaciones utilizadas durante el curso en formato digital.

Instructor

Cristian Borghello es Licenciado en Sistemas (UTN), desarrollador, [Certified Information Systems Security Professional \(CISSP\)](#), [Certificate of Cloud Security Knowledge \(CCSK\)](#), [Microsoft MVP \(Most Valuable Professional\) Security](#).

Actualmente es Director de **Segu-Info** y **Segu-Kids** y se desempeña como consultor independiente en Seguridad de la Información.

Escribe para diversos medios especializados e investiga en forma independiente sobre Seguridad Informática y de la Información.

Ha brindado cursos y dictado congresos y seminarios nacionales e internacionales sobre la temática.

Duración y Modalidad

Este curso teórico/práctico tiene una duración de **32 horas** y la modalidad de cursada es presencial. También puede ser desarrollado **in-company**.

Temario

Estado en la seguridad del software

- Principios del diseño de software seguro
- Modelos de desarrollo
- Etapas del desarrollo (IATF 3.1, ISO 12207 y NIST 800-64)
- Fuentes de fallas en los sistemas informáticos
- Eficiencia en la implementación de mejores prácticas y Frameworks
- S-SDLC (Secure Software Development LifeCycle)
- ROSI asociado al ciclo de desarrollo seguro

Identificación de amenazas

- Gestión de riesgos según NIST SP 800-30
- Identificación de amenazas
- Threat Modeling (modelado de amenazas)
- ISAM (The Integral Secure Agile Methodology)
- OWASP Development Guide
- CLASP (Comprehensive, Lightweight Application Security Process)
- Caso de (ab)uso
- Ataques basados en red: Eavesdropping, Tampering, Spoofing, Hijacking, Observing
- Tipos de ataque: Interposición, Sniffing, Replay

Modelado de amenazas

- Modelo de Microsoft STRIDE (Spoofing, Tampering, Repudiation, Info Disclosure, DoS, Elevation of Privilege)
- Modelo de Microsoft DREAD (Damage, Reproducibility, Exploitability, Affected Users, Discoverability)
- Microsoft Truthworthy Computing
- Microsoft SD³ + C
- Microsoft PD³ + C
- Microsoft Security Development Lifecycle
- Microsoft Threat Analysis & Modeling (TAM)
- Microsoft Threat Modeling Tool (TMT)
- Ejemplo práctico de desarrollo con un caso de uso utilizando TAM y TMT

Análisis y práctica de desarrollo seguro

- CMM y OSSTMM
- Programación "defensiva"
- Certificación de productos
- Testing Black y White Box
- Objetos y Código fuente
- Fallas de implementación vs defectos de diseño
- Revisión de código

- Revisión del punto de entrada
- Superficie de ataque (RASQ)
- Los 33 principios de seguridad NIST 800-27
- OWASP Top 10, CWE y SANS Top 20
- Análisis de código estático y dinámico
- Debilidades según NIST 800-268
- Ingeniería reversa, desensambladores y descompiladores
- Ofuscación de código
- Buffer Overflow, SQL Injection, XSS
- Fuzzing
- Hashing y HMAC