FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

PROGRAMA DE CURSO

Código	Nombre
CC5315	Seguridad de Software

Nombre en Inglés
Software Security

SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	0	7

Requisitos	Carácter del Curso
CC3301 Programación de Software de Sistemas, CC4401 Ingeniería de Software	Electivo para ICC

Resultados de Aprendizaje

Al termino del curso el estudiante demuestra que es capaz de:

1. Analizar ataques y vulnerabilidades de software con sus consecuencias tanto personales, técnicas y económicas con ejemplos de la vida real, comprendiendo la importancia de la seguridad.
2. Evaluar el estado de la seguridad de aplicaciones en casos reales, analizando los distintos elementos de estas.
3. Aplicar herramientas y bibliotecas que le permiten probar y reforzar la seguridad de las aplicaciones del software resolviendo problemas.

Metodología Docente	Evaluación General
El curso se desarrollará bajo la estrategia activo-participativa, en donde se utilizarán la siguiente metodología: • Clases expositivas. • Aprendizaje basado en proyectos (Proyecto de semestre). • Clases de Simulación (Hands-On).	Se utilizarán las siguientes instancias de evaluación: • Controles y examen. • Proyecto de semestre.

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	**Amenazas al Software**	2

Contenidos	Resultados de Aprendizaje de la Unidad	Referencias a la Bibliografía
1. Ataques al software y sus consecuencias. 2. ¿Por qué es tan difícil hacer software seguro? 3. Principios y Propiedades de seguridad	El estudiante demuestra que: 1. Justifica la importancia de la seguridad en el desarrollo de software en general. 2. Identifica las causas de la dificultad para desarrollar software seguro en cualquier contexto.	[1], [2], [5]

Número	Nombre de la Unidad	Duración en Semanas
2	**Taxonomía de Vulnerabilidades**	6

Contenidos	Resultados de Aprendizaje de la Unidad	Referencias a la Bibliografía
1. Vulnerabilidades típicas, clasificadas en una taxonomía conocida. Cómo reconocerlas y mitigarlas: • Validación de Entradas • Abuso de Funcionalidad • Características de Seguridad • Estado y Tiempo • Manejo de Errores • Calidad del Código • Encapsulamiento • Ambiente	El estudiante demuestra que: 1. Identifica y clasifica vulnerabilidades en el software. 2. Valora las vulnerabilidades en base a sus detalles, daño que pueden causar y su mitigación.	[1], [3]

Número	Nombre de la Unidad	Duración en Semanas
3	**Análisis de seguridad**	2

Contenidos	Resultados de Aprendizaje de la Unidad	Referencias a la Bibliografía
1. Modelamiento de amenazas 2. Documentación del análisis	El estudiante demuestra que: 1. Analiza el estado de la seguridad de un software después del desarrollo de éste. 2. Clasifica amenazas y riesgos por niveles de prioridad en situaciones que se le presentan.	[4], [5]

Número	Nombre de la Unidad	Duración en Semanas
4	**Desarrollo y herramientas de software seguro**	5

Contenidos	Resultados de Aprendizaje de la Unidad	Referencias a la Bibliografía
1. Herramientas de desarrollo de software seguro. • Criptografía aplicada • Fuzzers, analizadores estáticos • Descompilación • Sandboxing • Herramientas integradas en API's y frameworks. 2. Buenas prácticas para desarrollar software seguro. • Manuales de buenas practicas • Consejos y Principios de software seguro • Respuesta a incidentes	El estudiante demuestra que: 1. Utiliza herramientas automatizadas que le permitan proteger las aplicaciones desarrolladas en el entorno empresarial. 2. Integra buenas prácticas de seguridad al desarrollo de software. 3. Conoce procedimientos de respuesta a incidentes de seguridad que afecten al software creado.	[1], [2], [3],[5], [4]

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

Bibliografía General

1. [McGraw06] Gary McGraw, “Software Security: Building Security In”, Addison-Wesley Professional, 2006.

2. [Graff03] Mark G. Graff, Kenneth R. van Wyk: “Secure Coding: Principles and Practices”, O'Reilly, 2003.

3. [Sins10] Michael Howard, David LeBlanc, John Viega: “24 Deadly Sins of Software Security. Programming Flaws and How to Fix Them”, McGraw Hill, 2010.

4. [SDL09] Michael Howard, Steve Lipner: “The Security Development Lifecycle”, Microsoft Press, 2009.

5. [Viega02] John Viega, Gary McGraw, “Building Secure Software: How to Avoid Security Problems The Right Way”, Addison-Wesley, 2002.

Vigencia desde:	2012
Elaborado por:	Cristián Rojas
Revisado por:	Alejandro Hevia