UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

PROGRAMA DE ESTUDIO

DESARROLLO DE SOFTWARE SEGURO	**0682**	**8°, 9°**	**06**
Asignatura	Clave	Semestre	Créditos

Ingeniería Eléctrica	**Ingeniería en Computación**	**Ingeniería en Computación**
División	Departamento	Carrera en que se imparte

Asignatura:		**Horas:**		**Total (horas):**	
Obligatoria		Teóricas	3.0	Semana	3.0
Optativa	X	Prácticas	0.0	16 Semanas	48.0

Aprobado: Consejo Técnico de la Facultad — Fecha: 25 de febrero, 17 de marzo y 16 de junio de 2005
Consejo Académico del Área de las Ciencias Físico Matemáticas y de las Ingenierías — 11 de agosto de 2005

Modalidad: Curso.

Asignatura obligatoria antecedente: Ninguna.

Asignatura obligatoria consecuente: Ninguna.

Objetivo(s) del curso:
El alumno comprenderá, analizará y aplicará los principios, tendencias y técnicas para el desarrollo de software seguro.

Temario

NÚM.	NOMBRE	HORAS
1.	Introducción a la seguridad del software	3.0
2.	Administración de los riesgos en la seguridad del software	5.0
3.	Código abierto o cerrado	5.0
4.	Principios guías del software seguro	5.0
5.	Auditoria de software	6.0
6.	Código seguro	12.0
7.	Pruebas de software	6.0
8.	Derechos de autor en México (software)	6.0
		48.0
	Prácticas de laboratorio	0.0
	Total	48.0

1 Introducción a la seguridad del software

Objetivo: El alumno conocerá y comprenderá los fundamentos teóricos, tendencias y metas de la seguridad en el software.

Contenido:

- **1.1** Concepto de Software
- **1.2** Casos reales de fallas en el software
- **1.3** Futuro del software
- **1.4** Fuentes para información de vulnerabilidades
  - **1.4.1** Buqtraq
  - **1.4.2** CERT Advisores
  - **1.4.3** RISK Digest
- **1.5** Tendencias técnicas que afectan a la Seguridad del Software
- **1.6** Breanking and patch (romper y actualizar)
- **1.7** Metas de la Seguridad enfocadas al Software
  - **1.7.1** Prevención
  - **1.7.2** Auditable y trazable
  - **1.7.3** Monitoreo
  - **1.7.4** Privacidad y Confidencialidad
  - **1.7.5** Seguridad Multiniveles
  - **1.7.6** Anonimato
  - **1.7.7** Autenticación
  - **1.7.8** Integridad
- **1.8** Conocer al enemigo
- **1.9** Metas de proyecto de Software

2 Administración de los riesgos en la seguridad del software

Objetivo: El alumno *conocerá* e *identificará* los riesgos que se tienen al poner en práctica la seguridad del software, así como los mecanismos para la evaluación del desarrollo de sistemas seguros.

Contenido:

- **2.1** Descripción de la administración de los riesgos en la Seguridad del Software
- **2.2** Administración de los riesgos en la seguridad del Software en la práctica
  - **2.2.1** Pruebas de Caja Negra
  - **2.2.2** Equipo Rojo
- **2.3** Criterios Comunes

3 Código abierto o cerrado

Objetivo: El alumno *conocerá* los mecanismos que emplea la industria del software para proteger sus códigos, tanto de los competidores como de los *crackers*; así como las ventajas y desventajas del código abierto.

Contenido:

- **3.1** Seguridad por Oscuridad
- **3.2** Ingeniería en Reversa
- **3.3** Código Fuente Abierto
- **3.4** Falacias del código abierto

4 Principios guías del software seguro

Objetivo: El alumno *conocerá* e *identificará* los principios más importantes que deben estar presentes usando se diseña o construye un sistema seguro, evitando los problemas más comunes de seguridad.

Contenido:

- **4.1** Principio 1. Reducir las líneas débiles
- **4.2** Principio 2. Defensa por pasos o capas
- **4.3** Principio 3. Seguramente fallará
- **4.4** Principio 4. Menos privilegios
- **4.5** Principio 5. Segmentación
- **4.6** Principio 6. Mantenerlo simple
- **4.7** Principio 7. Promover la privacía
- **4.8** Principio 8. Ocultar secretos es difícil
- **4.9** Principio 9. Transparentar el código
- **4.10** Principio 10. Usar recursos comunes

5 Auditoria de software

Objetivo: El alumno *conocerá* e *identificará* las etapas que se requieren para poder llevar a cabo la auditoría de software una vez que éste ha sido terminado; así como las herramientas que permiten realizar auditoría al código fuente.

Contenido:

- **5.1** Definición de Arquitectura de Seguridad
- **5.2** Principios de la Arquitectura de Seguridad
- **5.3** Análisis de la Arquitectura de Seguridad
  - **5.3.1** Diseño
  - **5.3.2** Implementación
  - **5.3.3** Automatización y pruebas
  - **5.3.4** Árboles de Ataque
  - **5.3.5** Reporte del Análisis
- **5.4** Implementación del Análisis de Seguridad
  - **5.4.1** Auditoría de Código Fuente
  - **5.4.2** Herramientas de Auditoría de Seguridad de Código

6 Código seguro

Objetivo: El alumno *conocerá, identificará* y *aplicará* diferentes lenguajes de programación que le permitan *analizar*, *diseñar* y *desarrollar* las diferentes técnicas de código de seguro.

Contenido:

- **6.1** Definición de Código Seguro
- **6.2** Lenguaje Ensamblador
- **6.3** Lenguajes de Programación
- **6.4** Técnicas de Código Seguro
  - **6.4.1** Buffer Overflows
  - **6.4.2** Heap Overflows
  - **6.4.3** Formato de cadena
  - **6.4.4** Exploits
  - **6.4.5** Race conditions
  - **6.4.6** SQL injection
  - **6.4.7** Cross Site & Cross-Domain Scripting
  - **6.4.8** Fault Injection

7 Pruebas de software

Objetivo: El alumno *conocerá* e *identificará* las fases y los diferentes tipos de pruebas que se realizan al software.

Contenido:

- **7.1** Fases de las Pruebas de Software
  - **7.1.1** Modelado del ambiente del software
  - **7.1.2** Selección de escenarios de prueba
  - **7.1.3** Ejecución y evaluación de los escenarios de prueba
  - **7.1.4** Medición del progreso de las pruebas
- **7.2** Prácticas de las Pruebas de Software
  - **7.2.1** Básicas
    - **7.2.1.1** Especificaciones funcionales
    - **7.2.1.2** Revisión e inspección
    - **7.2.1.3** Entrada formal y criterios de salida
    - **7.2.1.4** Prueba funcional
    - **7.2.1.5** Pruebas multiplataforma
    - **7.2.1.6** Ejecución automatizada de prueba
    - **7.2.1.7** Programas beta
  - **7.2.2** Fundamentales
    - **7.2.2.1** Escenarios de usuario
    - **7.2.2.2** Pruebas de utilidad
    - **7.2.2.3** Requerimientos para la planificación de la prueba
    - **7.2.2.4** Generación automatizada de la prueba
  - **7.2.3** Incrementales
    - **7.2.3.1** Cobertura de código
    - **7.2.3.2** Generador de ambiente automatizado
    - **7.2.3.3** Diagrama del estado de la prueba
    - **7.2.3.4** Simulación de falla en la memoria
    - **7.2.3.5** Pruebas estadísticas
    - **7.2.3.6** Métodos semiformales
    - **7.2.3.7** Registro de la prueba para el código
    - **7.2.3.8** Benchmark
    - **7.2.3.9** Generación de errores (bugs)

8 Derechos de autor en México (software)

Objetivo: El alumno conocerá y aprenderá cómo proteger y registrar un programa de cómputo En México a través la Ley Federal del Derecho de Autor.

Contenido:

- **8.1** Ley Federal del Derecho de Autor (LFDA) en México
  - **8.1.1** Definición
  - **8.1.2** Artículos para la protección jurídica del software
  - **8.1.3** Derechos que se confieren a través de la LFDA
    - **8.1.3.1** Derechos morales
    - **8.1.3.2** Derechos patrimoniales
- **8.2** 8.2. Instituto Nacional del Derecho de Autor (INDAUTOR)
  - **8.2.1** Definición
  - **8.2.2** Ubicación del INDAUTOR
- **8.3** Dirección General de Asuntos Jurídicos de la UNAM (DGAJ)
  - **8.3.1** Definición
  - **8.3.2** Relación con el INDAUTOR
  - **8.3.3** Ubicación de la DGAJ
- **8.4** Registro del software
  - **8.4.1** Procedimiento y requerimientos para registrar software en el INDAUTOR.
  - **8.4.2** Procedimiento y requerimientos para registrar software en la DGAJ.
  - **8.4.3** Ventajas y desventajas al registrar software
- **8.5** Violación a los Derechos de Autor
- **8.6** Leyes que brindan protección jurídica al software en caso de violación.
- **8.7** Sociedades de Gestión Colectiva
  - **8.7.1** ¿Qué es una Sociedad de Gestión Colectiva?
  - **8.7.2** Procedimiento y requerimientos para registrar una Sociedad de Gestión Colectiva.
  - **8.7.3** Obligaciones y privilegios al formar parte de una Sociedad de Gestión Colectiva.

Bibliografía básica: **Temas para los que se recomienda:**

GRAFF G., Mark, VIEGA, John **Todos**
Building Secure Software
U.S.A.
Addison-Wesley, 2001

GRAFF G., Mark, VAN WYK, Kenneth R. **Todos**
Secure Coding
U.S.A.
O'Reilly, 2003

HOWARD, Michael, LEBLANC, David **Todos**
Writing Secure Code
2nd. Edition
U.S.A.
Microsoft Press, 2003

Bibliografía complementaria:

CORTÉS AGUILAR, Claudia, GARCÍA MORALES, Mayra **8**
Tesis : Propuesta de Reforma de Ley y Procedimientos para la Protección Jurídica de los Programas de Cómputo en México a través de la Ley Federal del Derecho de Autor
México
UNAM-FI, 2005

GARFINKEL, Simson, SCHWARTZ, Gene, SPAFOORD, Gene **6**
Practical Unix & Internet Security
3rd edition
USA
O'Reilly, 2003

MESSIER, Matt; VIEGA, John **6**
Secure Programming Cookbook for C and C++
USA
O'Reilly, 2003

Sugerencias didácticas:

Exposición oral	X	Lecturas obligatorias	X
Exposición audiovisual	X	Trabajos de investigación	X
Ejercicios dentro de clase	X	Prácticas de taller o laboratorio	
Ejercicios fuera del aula	X	Prácticas de campo	
Seminarios		Otras	

Forma de evaluar:

Exámenes parciales	X	Participación en clase	X
Exámenes finales	X	Asistencias a prácticas	
Trabajos y tareas fuera del aula	X	Otras	

Perfil profesiográfico de quienes pueden impartir la asignatura

El profesor deberá contar con licenciatura, preferentemente de las siguientes carreras; Ingeniería en computación, Ingeniería en Comunicaciones y Electrónica, Ingeniaría en Telecomunicaciones, Ingeniaría en Ciencias Computacionales o formación equivalente y contar con amplia experiencia en desarrollo de software seguro así como de proyectos y aplicaciones de la seguridad informática.