ASIGNATURA DE MÁSTER:

DESARROLLO DE SOFTWARE SEGURO (MÁSTER EN INGENIERÍA INFORMÁTICA)

Curso 2017/2018

(Código:31106205)

1.PRESENTACIÓN

Lamentablemente los denominados “ciberataques” son noticia frecuente en los medios de comunicación. Según los datos publicados por el CERT (Computer Emergency Response Team) las vulnerabilidades de los sistemas informáticos reportadas cada año crecen y aumentan su grado de sofisticación.

En este curso se presentan métodos rigurosos, técnicas y herramientas para desarrollar e implantar software seguro. Los métodos incluyen el análisis de código para detectar las vulnerabilidades habituales, la revisión de código fuente mediante herramientas de análisis estático y buenas prácticas para desarrollar código seguro en lenguajes concretos de programación.

Los métodos y herramientas que se estudian sirven para la realización de pruebas, verificación y validación de software y sistemas, comprobando que se cumplen los requisitos funcionales y de seguridad. Con todos ellos, se aprende a validar y verificar el aseguramiento de la seguridad así como a establecer la diferencia entre vulnerabilidades y errores de programación. Las técnicas de test incluyen las pruebas de caja blanca, caja negra, pruebas contra ataques, amenazas y penetración, pruebas de resiliencia, etc.

2.CONTEXTUALIZACIÓN

La asignatura “Desarrollo de Software Seguro” se enmarca en el Máster Universitario en Ingeniería Informática, dentro del Módulo “Complementos en tecnología informática”. Es una asignatura optativa de 6 créditos que se imparte en el primer semestre. La relación con estas otras asignaturas del máster:

- Planificación y gestión de proyectos informáticos de I+D+i
- Sistemas empotrados
- Temas avanzados de redes e internet
- Cloud computing y gestión de los servicios de red
- Sistemas operativos de dispositivos móviles.

Se puede resumir indicando que esta asignatura de “Desarrollo de Software Seguro” supone una extensión de todas ellas cuando se trata de desarrollar un sistema software que debe tener la cualidad adicional de ser seguro. Obviamente esta cualidad debería ser exigible en cualquier desarrollo de software actual. Sin embargo, lamentablemente los aspectos de seguridad no son tenidos en cuenta y las vulnerabilidades de los sistemas aumentan cada día más.

Las Competencias de la asignatura se pueden dividir entre competencias generales y competencias específicas:

Competencias generales

(G.1) Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la ingeniería informática.

(G.2) Capacidad para la dirección de obras e instalaciones de sistemas informáticos, cumpliendo la normativa vigente y asegurando la calidad del servicio.

(G.4) Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

(G.5) Capacidad para la elaboración, planificación estratégica, dirección, coordinación y gestión técnica y económica de proyectos en todos los ámbitos de la Ingeniería en Informática siguiendo criterios de calidad y medioambientales.

Competencias específicas

(CB.6) Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.

(CB.7) Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares)relacionados con su área de estudio.

(CB.8) Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.

(CB.9) Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.

(CB.10) Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

(CT.1) Capacidad para emprender y liderar proyectos innovadores en entornos científicos, tecnológicos y multidisciplinares.

(CT.2) Capacidad para tomar decisiones y formular juicios basados en criterios objetivos (datos experimentales, científicos o de simulación disponibles).

(DG.2) Capacidad para la planificación estratégica, elaboración, dirección, coordinación, y gestión técnica y económica en los ámbitos de la ingeniería informática relacionados, entre otros, con: sistemas, aplicaciones, servicios, redes, infraestructuras o instalaciones informáticas y centros o factorías de desarrollo de software, respetando el adecuado cumplimiento de los criterios de calidad y medioambientales y en entornos de trabajo multidisciplinares.

3.REQUISITOS PREVIOS RECOMENDABLES

La formación previa que deberían tener los alumnos para el adecuado seguimiento de esta asignatura son los propios de ingreso al posgrado, haciendo especial recomendación en conocimientos de ingeniería de software y lenguajes de programación.

Es muy recomendable que se haya cursado la asignatura "Fundamentos de Programación" para facilitar la comprensión de los ejemplos de la bibliografía escritos en lenguaje C. Se recomienda que el alumno tenga preferiblemente alguna experiencia previa de programación con C++ para seguir los ejemplos de la bibliografía. Los conocimientos básicos de programación orientada a objetos se pueden obtener de la asignatura de Programación Orientada a Objetos perteneciente a la materia Fundamentos de la Programación.

Además es necesario dominar el inglés técnico (leer y escribir) para manejar con facilidad las fuentes bibliográficas.

4.RESULTADOS DE APRENDIZAJE

La asignatura está enfocada al desarrollo y mantenimiento de software seguro y sin vulnerabilidades. Por tanto, los resultados de aprendizaje que se espera que el estudiante pueda alcanzar son:

- Identificar las principales causas de vulnerabilidad conocidas y desarrollar el código seguro que las evite.
- Conocer y saber aplicar un conjunto de métodos, técnicas y herramientas que permitan probar que el software desarrollado cumple los requisitos de funcionalidad y seguridad.
- Aplicar métodos para verificar formalmente la corrección de componentes de software crítico seguro.
- Realizar, junto con las pruebas tradicionales, otras adicionales específicas de seguridad.
- Usar modelos de penetración, patrones de ataque, de abuso o mal uso del sistema en la fase de pruebas.
- Conocer los procedimientos y programas de mantenimiento de software para que continúe cumpliendo con los requisitos de funcionalidad y seguridad.

5.CONTENIDOS DE LA ASIGNATURA

Tema 1: Estudio de Vulnerabilidades

1.1 Los aspectos estudiados en este tema son los siguientes:

1.2 Errores de programación más peligrosos según el CWE/SANS Top 25

1.3 Conceptos de seguridad

Tema 2: Prácticas de Desarrollo

Los aspectos estudiados en este tema son los siguientes:

2.1 Prácticas recomendadas:

2.2 El ciclo de vida del desarrollo seguro

2.3 Entrenamiento en seguridad

2.4 Requisitos

2.5 Diseño

2.6 Implementación

2.8 Verificación

Tema 3: Gestión de Memoria en C y C++

Los aspectos estudiados en este tema son los siguientes:

3.1 Errores más comunes de gestión de memoria

3.2 Buffer overflow

3.3 Stack smashing

3.4 Validación de entradas

Tema 4: Strings, Punteros y Manejo de Enteros

Los aspectos estudiados en este tema son los siguientes:

4.1 Errores de manejo de strings

4.2 Errores de overflow de enteros

4.3 Subterfugios con punteros

Tema 5: Otras vulnerabilidades en C y C++

Los aspectos estudiados en este tema son los siguientes:

5.1 Errores de formateado de Entrada/Salida de datos

5.2 Errores de secuenciado de Entrada/Salida de datos

5.3 Errores de manejo de ficheros

5.4 Errores de concurrencia

6.EQUIPO DOCENTE

- DAVID JOSE FERNANDEZ AMOROS
- JOSE ANTONIO CERRADA SOMOLINOS

7.METODOLOGÍA

La docencia de esta asignatura se impartirá a distancia, siguiendo el modelo educativo propio de la UNED. El principal instrumento docente será la plataforma aLF en la que se habilitarán diversos foros para canalizar las consultas y comentarios.

Las actividades a realizar por parte del alumno se desglosan de la siguiente manera:

Actividades formativas	Horas
Estudio de contenidos	60
Tutorías	10
Actividades en la plataforma virtual	5
Trabajos individuales	30
Trabajos en equipo	15
Prácticas informáticas	30

Elaboración de informes	0
Resolución de casos	0

Además, el estudiante podrá realizar consultas al equipo docente a través del correo, teléfono y presencialmente en los horarios establecidos para estas actividades. Ver apartado de *Tutorización* en esta guía docente.

8.BIBLIOGRAFÍA BÁSICA

ISBN(13): 9780321822130
Título: SECURE CODING IN C AND C++ (Second Edition)
Autor/es: Robert C. Seacord ;
Editorial: ADDISON WESLEY

Buscarlo en libreria virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9781439826966
Título: SECURE AND RESILENT SOFTWARE DEVELOPMENT
Autor/es: Mark S. Merkow And Lakshmikanth Raghavan ;
Editorial: CRC Press

Buscarlo en libreria virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

Comentarios y anexos:

Los dos libros están accesibles desde el portal de la UNED. Hay que autenticarse, y a partir de ahí.

- El libro de Seacord está aquí: (enlace). En este primer libro se estudian los aspectos generales relativos a todo el ciclo de vida del desarrollo de software seguro y sus particularidades.
- El libro de Merkow está aquí: (enlace). Este segundo libro está dedicado específicamente a estudiar las vulnerabilidades y las técnicas de programación segura en el lenguaje C y C++.

Hay un artículo que forma parte de la biblografía recomendada:

- Tsipenyuk, Katrina; Chess, Brian & McGraw, Gary. *Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors.* IEEE Security & Privacy, 2005

Este artículo se utiliza para clasificar los ciberataques y está disponible en la plataforma aLF de la asignatura.

9.BIBLIOGRAFÍA COMPLEMENTARIA

Comentarios y anexos:

Aunque no se consideran necesarios para el estudio de la asignatura, los libros y documentos de esta bibliografía complementaria pueden ser muy interesantes para un estudio en mayor profundidad de la asignatura. La relación de documentos se incluye en la parte 2 de esta guia de la asignatura.

10.RECURSOS DE APOYO AL ESTUDIO

Los alumnos tendrán a su disposición los siguientes recursos de apoyo al estudio:

- Guía de la asignatura: Incluye el plan de trabajo y orientaciones para su desarrollo. Esta guía será accesible desde el curso virtual.
- Curso virtual: A través de esta plataforma los alumnos pueden consultar información de la asignatura, acceder a material complementario, enunciados de ejercicios resueltos para que el alumno pueda autoevaluar sus conocimientos, realizar consultas al equipo docente y/o tutores a través de los foros correspondientes e intercambiar información con el resto de compañeros.
- Tutorías. En el Centro Asociado al que pertenezca el estudiante, éste deberá consultar si existe la posibilidad de disponer de una tutoría presencial con un tutor/a que le atienda presencialmente.
- Biblioteca: el acceso a las bibliotecas de los Centros Asociados y de la Sede Central permitirán al estudiante encontrar la bibliografía que podrá serle de utilidad durante el proceso de aprendizaje. De particular interés es el acceso electrónico a la colección de Safari Books Online a la que tienen acceso los estudiantes de la UNED.

Se ofrecerán las herramientas necesarias para que, tanto el equipo docente como el alumnado, encuentren la manera de compaginar el trabajo individual y el aprendizaje cooperativo (Skype, Moodle, Alf, etc.) si este se considerará necesario. Además de ello se podrá contactar con el equipo docente por teléfono y correo electrónico.

11.TUTORIZACIÓN Y SEGUIMIENTO

La tutorización de los alumnos se llevará a cabo fundamentalmente a través de la plataforma aLF. Además se puede utilizar el correo electrónico y las consultas telefónicas:

Profesor: *David Fernández Amorós*

Horario: Jueves de 16:00 a 20:00

david@issi.uned.es,

Teléfono: 91 398 8241

Profesor: *José Antonio Cerrada*

Horario: Jueves de 16:00 a 20:00

jcerrada@issi.uned.es,

Teléfono: 91 398 6478

También es posible una asistencia personalizada (preferentemente previo aviso) en los días y horas de tutorización en la siguiente dirección:

Dpto. de Ingeniería de Software y Sistemas Informáticos

ETSI Informática, UNED

C/ Juan del Rosal, 16

28040 MADRID

12.EVALUACIÓN DE LOS APRENDIZAJES

Para evaluar los conocimientos adquiridos, el alumno deberá realizar una Prueba de Evaluación a Distancia (PEC) que determinará el 30% de la nota final de la asignatura junto con un examen que determinará el 70% restante de la nota.

La PEC consiste en una prueba tipo de test de 10 preguntas teóricas sobre cuestiones de desarollo de software seguro. Habrá dos intentos para superar la prueba, uno correspondiente a la convocatoria de febrero y otro más adelante para la convocatoria de septiembre.

El examen consistirá en una serie breve (de no más de cinco) preguntas sobre el contenido de la asignatura. Puede tratarse de desarrollos teóricos sobre temas de la bibliografía o de preguntas sobre situaciones concretas descritas en el enunciado, incluyendo ejemplos de código vulnerable.

13.COLABORADORES DOCENTES

Véase equipo docente.