ASIGNATURA DE MÁSTER:

DESARROLLO DE SOFTWARE SEGURO

Curso 2017/2018

(Código:31105147)

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.

2.CONTEXTUALIZACIÓN

La asignatura “Desarrollo de Software Seguro” se enmarca en el Máster Universitario en Investigación en Ingeniería de Software y Sistemas Informáticos. En particular, es una de las asignaturas de la Materia “Ingeniería del desarrollo de Software”.

Esta asignatura supone una extensión de los aspectos de desarrollo de software 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 Generales de la asignatura son (la medida de la intensidad con que se aplica cada competencia sigue una escala de 1 a 3):

CG1: Competencias de gestión y planificación (intensidad 2)

CG2: Competencias cognitivas superiores (intensidad 3)

CG3: Competencias de gestión de la calidad y la innovación (intensidad 3)

CG4: Competencias de expresión y comunicación (intensidad 2)

CG5: Competencias en el uso de las herramientas y recursos de la Sociedad del Conocimiento (intensidad 3)

CG6: Trabajo en equipo desarrollando distinto tipo de funciones o roles (intensidad 2)

CG7: Compromiso ético, especialmente relacionado con la deontología profesional (intensidad 2)

Las Competencias Específicas Disciplinares de la asignatura son (la medida de la

intensidad con que se aplica cada competencia sigue una escala de 1 a 3):

CED1: Conocer porqué el desarrollo de software seguro es el mejor método para asegurar la seguridad de los sistemas informáticos (intensidad 3)

CED2: Identificar las peculiaridades en cada fase del ciclo de vida para lograr el desarrollo de software seguro (intensidad 2)

CED3: Conocer métodos de verificación y pruebas de las especificaciones de un software seguro (intensidad 3)

CED4: Conocer los procedimientos y programas de mantenimiento de software para que se continúe cumpliendo con los requisitos de seguridad (intensidad 2)

Las Competencias Específicas Profesionales de la asignatura son (la medida de la intensidad con que se aplica cada competencia sigue una escala de 1 a 3):

CEP1: 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 (intensidad 2)

CEP2: Usar modelos de penetración, patrones de ataque, de abuso o mal uso del sistema en la fase de pruebas (intensidad 2)

CEP3: Realizar, junto con las pruebas tradicionales, otras adicionales específicas de seguridad. (intensidad 2)

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.

Se recomienda que el alumno tenga preferiblemente alguna experiencia previa de programación con lenguajes C y C++.

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: Introducción

Los aspectos estudiados en este tema son los siguientes:

- Visión panorámica de las vulnerabilidades y sus costes
- Propiedades del software seguro y resiliente

Bibliografía de estudio:

- Capitulos 1 y 2 del libro [Merkow 2010]

Tema 2: Estudio de Vulnerabilidades

Los aspectos estudiados en este tema son los siguientes:

- *Errores de programación más peligrosos según el CWE/SANS Top 25*
- *Conceptos de seguridad*

Bibliografía de estudio:

- *Apéndice A del libro [Merkow 2010]*
- *Capítulo 1 del libro [Seacord 2013]*
- *Artículo [Tsipenyuk 2005] disponible en formato pdf en la plataforma aLF*

Tema 3: Plan Estratégico

Los aspectos estudiados en este tema son los siguientes:

- Seguridad y resiliencia a lo largo del ciclo de vida
- Puntos de ataque y seguridad perimetral
- Buenas prácticas según OWASP (Open Web Application Security Project)

Bibliografía de estudio:

- Capítulos 3 y 4 del libro [Merkow 2010]

Tema 4: Prácticas de Desarrollo

Los aspectos estudiados en este tema son los siguientes:

- Buenas prácticas para análisis de requisitos, diseño arquitectónico y de detalle. Por ejemplo:
  - Casos de uso/abuso
  - Modelado de amenazas
  - Análisis de riesgos
  - Revisión de diseño
  - Defensa en profundidad

Bibliografía de estudio:

- *Capítulo 5 del libro [Merkow 2010]*
- *Capítulo 9 del libro [Seacord 2013]*

Tema 5: Buenas Prácticas de Programación

Los aspectos estudiados en este tema son los siguientes:

- Los 10 riesgos de seguridad más críticos según OWASP
- Plataforma ESAPI (OWASP Enterprise Security API)
- Cross-Site Scripting (XSS)
- Inyección de ataques
- Autentificación y gestión de sesión

Bibliografía de estudio:

- *Capítulo 6 del libro [Merkow 2010]*

Tema 6: Gestión de Memoria en C y C++

Los aspectos estudiados en este tema son los siguientes:

- Errores más comunes de gestión de memoria
  - Buffer overflow
  - Stack smashing
- Validación de entradas

Bibliografía de estudio:

- *Capítulos 2 y 4 del libro [Seacord 2013]*

Tema 7: Strings, Punteros y Manejo de Enteros

Los aspectos estudiados en este tema son los siguientes:

- Errores de manejo de strings
- Errores de overflow de enteros
- Subterfugios con punteros

Bibliografía de estudio:

- *Capítulos 3 y 5 del libro [Seacord 2013]*

Tema 8: Otras vulnerabilidades en C y C++

Los aspectos estudiados en este tema son los siguientes:

- Errores de formateado de Entrada/Salida de datos
- Errores de secuenciado de Entrada/Salida de datos
- Errores de manejo de ficheros

Bibliografía de estudio:

- *Capítulos 6 y 7 del libro [Seacord 2013]*

Tema 9: Análisis Estático

Los aspectos estudiados en este tema son los siguientes:

- Tipos de análisis estático
- Herramientas de análisis
  - Coverity
  - Fortify

Bibliografía de estudio:

- *Capítulo 8 del libro [Merkow 2010]*

Tema 10: Pruebas

Los aspectos estudiados en este tema son los siguientes:

- Buenas prácticas de pruebas de unidades
- Pruebas de penetración
- Fuzzing

Bibliografía de estudio:

- *Capítulo 9 del libro [Merkow 2010]*

6.EQUIPO DOCENTE

- JOSE ANTONIO CERRADA SOMOLINOS
- DAVID JOSE FERNANDEZ AMOROS

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 en los tres ámbitos siguientes:

- Actividades de contenido teórico: lectura de las orientaciones generales; lectura comprensiva de la bibliografía, material didáctico e información temática; e intercambio de información y consulta de dudas con el equipo docente
- Actividades de contenido práctico: manejo de herramientas informáticas y de ayuda a la presentación de resultados; intercambio de información con otros compañeros y el equipo docente sobre aspectos prácticos y participación, argumentación y aportación constructiva en los debates en foros
- Trabajo autónomo: búsqueda de información adicional en biblioteca, Internet, etc.; selección de la información útil; actividades, que el estudiante realiza de manera autónoma, orientadas a resolver ejercicios, prácticas, problemas o trabajos que se plantean específicamente en la asignatura; realización de memorias de las prácticas, trabajos y desarrollos.

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)
- El libro de Merkow está aquí: (enlace)

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

El artículo está disponible en el curso virtual 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

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.

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: *José Antonio Cerrada*

Horario: Jueves de 10:00 a 14:00

jcerrada@issi.uned.es,

Teléfono: 91 398 6478

También es posible una asistencia personalizada 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 las tareas que se describen en la parte 2 de esta guia de la asignatura para cada uno de los Temas. Dependiendo de la disponibilidad de tiempo y dedicación del alumno, existen dos modalidades de evaluación. Cada alumno deberá optar al comienzo del curso por una de las dos modalidades de evaluación. Esta elección se reflejará en la entrega inicial que deberá realizarse:

FECHA ENTREGA INICIAL: Antes del día 15 de Enero.

En esta entrega inicial, todos los alumnos deberán incluir los siguientes documentos según la modalidad elegida:

- *Evaluación Continua: Documentos correspondiente a los Tema 1 y 2*

- *Evaluación Completa: Documento con los* Requisitos Funcionales y de Seguridad *del Proyecto del Curso siguiendo las pautas indicadas en los Temas 1 y 2 con una extensión de entre 3 y 5 páginas.*

Asimismo, dependiendo de la modalidad elegida, los trabajos a realizar y las fechas serán los siguientes:

- *Evaluación Continua: El alumno deberá ir entregando mensualmente otros CUATRO conjuntos de documentos que se describen como tareas en los correspondientes Temas según el siguiente calendario:*

- *Temas 3 y 4 --- antes del 1 de Marzo*
- *Temas 5 y 6 --- antes del 1 de Abril*
- *Temas 7 y 8 --- antes del 1 de Mayo*
- *Temas 9 y10 --- antes del 1 de Junio*

- *Evaluación Completa: El alumno sólo hará la entrega de la documentación final completa del Proyecto del Curso en la siguiente fecha:*

- *Proyecto del Curso --- antes del 1 de Junio*

NOTA IMPORTANTE: En la Convocatoria de Septiembre no existe la modalidad de Evaluación Continua. Sólo se podrá realizar:

- *Evaluación Completa: El alumno que necesite utilizar la convocatoria de septiembre sólo podrá hacer la entrega de la documentación final completa del Proyecto del Curso en la siguiente fecha:*

- *Proyecto del Curso --- antes del 10 de Septiembre*

13.COLABORADORES DOCENTES

Véase equipo docente.