18-19

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS

GUÍA DE ESTUDIO PÚBLICA

DESARROLLO DE SOFTWARE SEGURO

CÓDIGO 31105147

UNED

18-19

DESARROLLO DE SOFTWARE SEGURO

CÓDIGO 31105147

ÍNDICE

Nombre de la asignatura	DESARROLLO DE SOFTWARE SEGURO
Código	31105147
Curso académico	2018/2019
Título en que se imparte	MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS
Tipo	CONTENIDOS
Nº ETCS	9
Horas	225.0
Periodo	ANUAL
Idiomas en que se imparte	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

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.
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.

REQUISITOS Y/O RECOMENDACIONES PARA CURSAR ESTA ASIGNATURA

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, C++, Java o similares

EQUIPO DOCENTE

Nombre y Apellidos	JOSE ANTONIO CERRADA SOMOLINOS
Correo Electrónico	jcerrada@issi.uned.es
Teléfono	91398-6478
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	ING.DE SOFTWARE Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	DAVID JOSE FERNANDEZ AMOROS
Correo Electrónico	david@issi.uned.es
Teléfono	91398-8241
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	ING.DE SOFTWARE Y SISTEMAS INFORMÁTICOS

HORARIO DE ATENCIÓN AL ESTUDIANTE

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

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Competencias Básicas:

CB6 - 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

CB7 - 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

CB8 - 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

CB9 - 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

CB10 - 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.

Competencias Generales:

CG01 - Saber aplicar los conocimientos adquiridos y la capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares relacionados con la Ingeniería de Sistemas y la Ingeniería de Software.

CG02 - Demostrar una comprensión sistemática del campo de estudio de la Ingeniería de

Software o de la Ingeniería de Sistemas, y el dominio de las habilidades y métodos de investigación relacionados con dicho campo.
CG03 - Demostrar la capacidad de concebir, diseñar, poner en práctica y adoptar un proceso sustancial de investigación con seriedad académica.
CG04 - Ser capaz de realizar un análisis crítico, evaluación y síntesis de ideas nuevas y complejas.
CG05 - Saber comunicar sus conclusiones -y los conocimientos y razones últimas que las sustentan- a públicos especializados y no especializados, a sus colegas, a la comunidad académica en su conjunto y a la sociedad, de un modo claro y sin ambigüedades.
CG06 - Ser capaz de fomentar, en contextos académicos y profesionales, el avance tecnológico dentro de una sociedad basada en el conocimiento.
CG07 - Ser capaz 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.
CG08 - Realizar una contribución a través de una investigación original que amplíe las fronteras del conocimiento desarrollando un corpus sustancial, del que parte merezca la publicación referenciada a nivel nacional o internacional.
CG09 - Poseer las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

Competencias Específicas:
CE01 - Incorporar mejoras cualitativas sustanciales, bien sea en la elaboración de software o bien en el desarrollo e implantación de sistemas robóticos.
CE02 - Concebir, implementar implantar y supervisar nuevas soluciones a los problemas específicos que se le planteen en el ámbito de la investigación, innovación y desarrollo de software o de la robótica.

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.

CONTENIDOS

Tema 1: Introducción

Tema 2: Estudio de Vulnerabilidades

Tema 3: Plan Estratégico

Tema 4: Prácticas de Desarrollo

Tema 5: Buenas Prácticas de Programación

Tema 6: Gestión de Memoria en C y C++

Tema 7: Strings, Punteros y Manejo de Enteros

Tema 8: Otras vulnerabilidades en C y C++

Tema 9: Análisis Estático

Tema 10: Pruebas

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 ***Horario de atención al estudiante*** en esta guía docente.

SISTEMA DE EVALUACIÓN

TIPO DE PRIMERA PRUEBA PRESENCIAL

Tipo de examen: No hay prueba presencial

TIPO DE SEGUNDA PRUEBA PRESENCIAL

Tipo de examen2: No hay prueba presencial

CARACTERÍSTICAS DE LA PRUEBA PRESENCIAL Y/O LOS TRABAJOS

Requiere Presencialidad: No

Descripción

El sistema de evaluación está basado en el desarrollo de un proyecto o programa ad hoc en el que se deben estudiar y analizar los conceptos fundamentales de la asignatura.

Para la elaboración del proyecto/programa, y su evaluación correspondiente, existen 2 alternativas: por partes (Parcial) o final completa (Completa).

En ambos casos, en el desarrollo del proyecto/programa se deben incorporar los elementos fundamentales sobre seguridad estudiados en la asignatura. Por ello, no es importante la funcionalidad concreta del proyecto/programa y sin embargo es fundamental estudiar el mayor número de escenarios posibles con vulnerabilidades y los correspondientes mecanismos para evitarlas.

Es condición imprescindible para cualquiera de las modalidades de evaluación la presentación de una Entrega Obligatoria Inicial antes del 15 de Enero.

En el caso de evaluación parcial esta entrega corresponderá a las actividades estudiadas en los temas 1 y 2.

En el caso de evaluación completa corresponderá con la elaboración de los requisitos del proyecto/programa a desarrollar con un análisis de requisitos, riesgos y vulnerabilidades a evitar. La adecuación de la propuesta a los contenidos y objetivos de aprendizaje en la asignatura será verificada por el equipo docente.

Según la modalidad de evaluación tendremos:

Evaluación Parcial: Es necesario entregar otros CUATRO trabajos cuyos contenidos se indican en las actividades de los Temas 3 y 4 antes del 1 de marzo, de los Temas 5 y 6, antes del 1 de abril, de los Temas 7 y 8 antes del 1 de mayo y de los Temas 9 y 10 antes del 1 de junio..

Evaluación Completa: Es necesario entregar un Proyecto Completo cuyos contenidos se indican en las actividades de los Temas 1 a 10 y los requisitos presentados en la entrega inicial. La entrega deberá realizarse antes del 1 de junio o bien antes del 10 de septiembre dependiendo de la convocatoria elegida para su presentación.

Criterios de evaluación

Los criterios con que se evalúan los contenidos tanto de cada trabajo parcial, como del Proyecto Completo, se refieren a los objetivos de la asignatura y a los resultados del aprendizaje esperados para ella:
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.
En particular, los contenidos de los trabajos (bien sean los parciales o el proyecto completo) se refieren a las actividades directamente vinculadas a unos temas concretos del programa de la asignatura; por lo que los criterios de la evaluación de cada parte se refieren al cumplimiento de los objetivos de los temas que le correspondan.
Adicionalmente, en el caso de la evaluación completa para el trabajo final, también se tendrá en cuenta su coherencia con la propuesta inicial aprobada por el equipo docente.

Ponderación de la prueba presencial y/o los trabajos en la nota final	Según el tipo de evaluación elegida la ponderación es la siguiente: Evaluación Parcial: 20% cada uno de los 5 trabajos Evaluación Completa: 100%. el trabajo final
Fecha aproximada de entrega	Las fechas están detalladas en el apartado de descripción

Comentarios y observaciones

Es **condición imprescindible** para cualquiera de las modalidades de evaluación la presentación de una **Entrega Obligatoria Inicial** por parte del estudiante antes del 15 de Enero. La adecuación de la propuesta a los contenidos y los objetivos de aprendizaje en la asignatura será verificada por el equipo docente.

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC?	No
Descripción	
Criterios de evaluación	
Ponderación de la PEC en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s? No

Descripción

Criterios de evaluación

Ponderación en la nota final

Fecha aproximada de entrega

Comentarios y observaciones

¿CÓMO SE OBTIENE LA NOTA FINAL?

La asignatura está superada si la nota final NF >= 5.

La condición necesaria para obtener un resultado suficiente en la evaluación es la presentacion de las correspondientes entregas en tiempo y forma y su valoración favorable.

Dependiendo de la modalidad de evaluación:

Evaluación Parcial: NF = 0.20 x (TP1 + TP2 + TP3 + TP4 + TP5) donde TP es el trabajo parcial correspondiente.

Evaluación Completa: NF = TF, donde TF es el Trabajo Final.

BIBLIOGRAFÍA BÁSICA

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

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

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.

BIBLIOGRAFÍA COMPLEMENTARIA

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 es la siguiente:

•The MITRE Corporation (MITRE). ***Common Weakness Enumeration. (2010)*** http://cwe.mitre.org/

•Grembi, Jason. ***Secure Software Development - A Security Programmer's Guide.*** Tutorial at 11th Semi-Annual Software Assurance Forum. Arlington, VA, November 2009. Software Engineering Institute, Carnegie Mellon University, 2009.
https://www.vte.cert.org/vteweb/go/2699.aspx

•Gerhart, Susan; Hogle, Jan; &Crandall, Jedidiah. ***How Do Buffer Overflow Attacks Work? (2002).***
http://nsfsecurity.pr.erau.edu/bom/
En este documento se incluye una introducción a los problemas de buffer overflows y stack-smashing con ejercicios y animaciones muy interesantes que el alumno puede y debe utilizar para el estudio de la asignatura.

•CERT. ***CERT Secure Coding Standards (2010).***
https://www.securecoding.cert.org/

•Miller, Barton P.; Cooksey, Gregory; &Moore, Fredrick. ***An Empirical Study of the Robustness of MacOS Applications Using Random Testing.*** ACM SIGOPS Operating Systems Review 41, 1 (January 2007): 78-86.

•Golze, Andreas; Sarbiewski, Mark; &Zahm, Alain. ***Optimize Quality for Business Outcomes: A Practical Approach to Software Testing.*** Wiley Publishing, 2008. El capítulo 8 sobre pruebas de seguridad es especialmente útil.

•Howard, Michael &LeBlanc, David. ***Writing Secure Code, 2nd ed.*** Microsoft Press, 2003.

•Howard, Michael; LeBlanc, David; &Viega, John. ***19 Deadly Sins of Software Security.*** McGraw-Hill, 2005.

RECURSOS DE APOYO Y WEBGRAFÍA

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.

IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.