Ministerio de Cultura y Educación **(Programa del año 2020)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
(OPTATIVA) OP.SEGURIDAD DE SISTEMAS DE SOFTWARE	LIC.CS.COMP.	32/12	2020	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BERON, MARIO MARCELO	Prof. Responsable	P.Adj Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	2 Hs	1 Hs	3 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
22/09/2020	18/12/2020	13	75

IV - Fundamentación

El amplio crecimiento que las tecnologías de la información y de la comunicación han experimentado en este último tiempo ha provocado una utilización masiva de los sistemas de software en todos los ámbitos de la sociedad. En la actualidad la gran mayoría de las actividades se realizan a través de la utilización de medios informáticos lo cual permite realizar tareas que tiempo atrás eran impensadas.Los usuarios realizan compras por Internet, inscriben a sus hijos en las instituciones educativas, solicitan turnos a médicos, pagan sus cuentas desde sus hogares, entre otras tantas alternativas.
Muchas labores sociales han cambiado rotundamente debido a los avances en el desarrollo del hardware y del software. Es posible observar que existen trabajos en donde las tareas se realizan desde el hogar, comercios donde no son necesarios funcionarios que atiendan a los clientes, etc. Las organizaciones realizan tareas similares pero además construyen sistemas de software que hacen posible que los usuarios realicen sus labores a distancia. Como es posible percibir, el software ocupa un lugar central en la sociedad moderna. Los sistemas se hacen cada vez más complejos y grandes lo cual implica la participación de diferentes equipos cada uno de ellos compuesto por muchas personas. Asociado con los beneficios que la tecnología ha producido a la sociedad se encuentran riesgos, algunos de los cuales son muy serios, que hacen los usuarios queden expuestos a acciones no deseadas. Claramente, la tecnología y en particular el software llevan adelante tareas que son críticas para todos los miembros de la sociedad. Basta con pensar los problemas que pueden causar una transacción bancaria errónea, un pago mal realizado o una ficha médica mal procesada, entre otras tantas posibilidades. Dichos problemas pueden ser causados de manera no intencional, por fallas en el proceso de construcción de software. No obstante, el desperfecto también se puede producir de manera intencional. Sea intencional o no, se puede decir que cuando se produce un desperfecto o fuga de información se está en presencia de una violación a la seguridad. En el caso de que ocurra una falla de software la misma se puede deber a errores en la construcción, en el mantenimiento o en la migración o evolución del software.

Claramente es deseable que el software sea confiable, seguro, fácil de mantener y modificar. No obstante, esas metas son muy difíciles de alcanzar. De hecho existen líneas de investigación, tanto en el contexto académico como en el empresarial, que tienen como principal objetivo desarrollar métodos y estrategias para evitar los inconvenientes antes mencionados. Los accesos no deseados también están relacionados con la falta de calidad del software como así también a fallas en el proceso de construcción del mismo. Por las razones antes mencionadas, la seguridad de los sistemas de software han adquirido una importancia central en la actualidad y a posibilitado el surgimiento de una amplia variedad de investigaciones y desarrollos tecnológicos en el contexto de la seguridad de software. Lo antes mencionado hace que la seguridad informática sea un aspecto clave que los profesionales de informática necesitan dominar y aplicar a los sistemas de software.
En esta materia optativa se brindan los conceptos y técnicas esenciales para evitar, detectar y subsanar fallas en la seguridad de los sistemas de software de forma tal de nutrir a los estudiantes con las bases para la producción de software seguro.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso los alumnos deberán:
1. Conocer las principales fallas en la codificación de los sistemas de software.
2. Conocer estrategias de ataque y defensa.
3. Operar herramientas para implementar estrategias de protección de software.

VI - Contenidos

Unidad 1: Desarrollo de Software Seguro
Seguridad en los Sistemas de Software. Procesos de Desarrollo de Software Seguro. Estándares de Codificación Segura. Codificación Segura en C. Codificación Segura en Java. Patrones de Errores en la Codificación.

Unidad 2: Protección de Software
Concepción. Ataque y Defensa. Análisis de Programas. Ofuscación de Código. Aplicaciones de la Ofuscación de Código. Tipos de Ofuscación. Software a Prueba de Manipulaciones. Aplicaciones. Marca de Agua. Similaridad de Software. Análisis Forense de Software. Marca de Nacimiento. Técnicas de Protección por Hardware.

Unidad 3: Análisis de Programas
Análisis Estático. Análisis de Control de Flujo. Análisis de Flujo de Datos. Análisis de Dependencia de Datos. Análisis de Alias. Slicing. Interpretación Abstracta. Análisis Dinámico. Depuración. Profiling. Tracing. Emulación. Reconstitución de los Fuentes. Desensamble. Decompilación. Análisis Pragmático. Métricas de Estilo. Métricas de Complejidad de Software. Visualización de Software.

Unidad 4: Métodos de Ataque y Defensa
Estrategias de Ataque. Objetivo de Cracking. Motivación del Adversario. Por qué el atacante logra el objetivo? Cuál es la metodología del atacante? Qué herramientas usa el atacante? Qué técnicas usa el adversario? Estrategias de Defensa. Motivación. Primitivas de Protección: Cubrir. Duplicar. Dividir. Mezclar. Reordenar. Map. Indirecto. Mímica. Aviso.

Unidad 5: Aplicaciones
Extracción de la Información. Lexer. Parser. Árbol de Parse. Recorridos sobre el Árbol de Parse. Extracción de la Información con Gramáticas de Atributos, Listener y Visitors. Herramientas de Generación Automática de Lexers y Parsers. Operación de herramientas de generación automática de lexers, parsers y árbol de parse. Definición de recorridos al Árbol de Parse. Implementación de Gramáticas de Atributos para extraer información específica del dominio. Extracción de Información Específica del Dominio a través de la Implementación de: Gramáticas de Atributos, Listener y Visitors.

VII - Plan de Trabajos Prácticos

Unidad 1: Desarrollo de Software Seguro
Analizar las metodologías de desarrollo de software seguro utilizadas por las grandes empresas de desarrollo de software y describir las actividades que se incorporan para considerar la característica de seguridad de software.

Unidad 2: Protección de Software
Implementación de estrategias de protección de software: Ofuscación, Marca de Agua, etc.

Unidad 3: Análisis de Programas
Operación de herramientas de Análisis de Programa.
Implementación de Técnicas Básicas de Análisis de Programas.

Unidad 4: Métodos de Ataque y Defensa
Elaboración de estrategias de defensa ante situaciones hipotéticas de ataque.

Unidad 5: Aplicaciones
Construcción de aplicaciones que:
1. Detecten problemas sencillos en el código fuente de un sistema de software.
2. Protejan los activos del software.

VIII - Regimen de Aprobación

Condiciones de Regularización:
i) Aprobar los prácticos de laboratorio. Se otorgarán dos recuperaciones por cada práctico de laboratorio a todos los alumnos.
ii) Aprobar un práctico teórico-práctico con una nota mayor o igual a 6 (seis). Se otorgarán dos recuperaciones para cada evaluación parcial a todos los alumnos.
Condiciones de Aprobación:
a) El alumno debe:
a.1) Contar con la condición de regularización i).
a.2) Aprobar un práctico teórico-práctico con una nota mayor o igual a 7 (siete). Se otorgarán dos recuperaciones para cada práctico a todos los alumnos.
b) Aprobar una evaluación final integradora con una nota mayor o igual 7. Se otorgarán dos recuperaciones de esta evaluación a todos los alumnos.
c) Tener un porcentaje de asistencia del 80% a clases.
ii) Por examen final.
Alumnos Libres: No se aceptan alumnos con esta condición.

IX - Bibliografía Básica

[1] Alfred V. Aho,Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman Compilers: Principles, Techniques, and Tools. Pearson Education. 2006.
[2] Long, Fred; Mohindra, Dhruv; Seacord,Robert C.; Sutherland, Dean F; Svoboda, David. Java Coding Guidelines. Addison-Wesley. ISBN- 3: 978-0-321-93315-7. 2014.
[3] Seacord, Robert C. Secure Coding in C and C++. Second Edition. Addison-Wesley. ISBN-13: 978-0-321-82213-0. 2013.
[4] Eilam, E. Reversing. Secrets of Reverse Engineering. Wiley. ISBN-10: 0764574817, ISBN-13: 978-0764574818. 2005.
[5] Pressman, R; Maxim, B. Software Engineering: A Practitioner's Approach. McGraw-Hill. 8va Edición. ISBN-10:0078022126. ISBN-13: 978-0078022128. 2014.
[6] Chess, Brian; West,Jacob. Secure Programming with Static Analysis. ISBN: 0-321-42477-8. Addison-Wesley. 2007.
[7] Tomassetti, Federico. How to create pragmatic, lightweightlanguages. Leanpub. 2018.
[8] Howard, Michael; Lipner, Steve. THE SECURITY DEVELOPMENT LIFECYCLE. SDL: A Process for Developing Demonstrably More Secure Software. Microsoft Press. ISBN: 978-07356-2214-2. 2006.
[9] Application Security Verification Standard 3.0.1. OWASP. Open Web Application Security Project. 2016.

X - Bibliografia Complementaria

[1]

XI - Resumen de Objetivos

1. Conocer las principales fallas en la codificación de los sistemas de software.
2. Conocer estrategias de ataque y defensa.

3. Operar herramientas para implementar estrategias de protección de software.

XII - Resumen del Programa

Unidad 1: Desarrollo de Software Seguro
Unidad 2: Protección de Software
Unidad 3: Análisis de Programas
Unidad 4: Métodos de Ataque y Defensa
Unidad 5: Aplicaciones

XIII - Imprevistos

El DECNU 520/2020 de distanciamiento social, obligatorio y preventivo, establecido por el Gobierno Nacional y la necesidad de reajustar el Calendario Académico de la Universidad Nacional de San Luis, en lo referente al Segundo Cuatrimestre 2020, el Consejo Superior en su sesión del día 01/09/2020 estableció en el Artículo 1 de la Resolución Nro. 68/2020,que el Segundo Cuatrimestre sea de 13 semanas. A los efectos de que se impartan todos los contenidos y se respete el crédito horario establecido en el Plan de Estudios de la Carrera para esta asignatura, se establece que se de cómo máximo 6hs por semana distribuidas en teorías, prácticos de aula, práctico de máquina, trabajos tutoriales, consultas, hasta completar las 75hs. La metodología de la asignatura tiene las siguientes características:
*El dictado de las clases teóricas es mediante videoconferencias en plataformas tipo google meet, hanghout, skype, entre otras apoyadas con TIC.
*Los prácticos de aula y los prácticos de máquina se podrán realizar de forma individual o grupal, con 2 consultas por semana.

Para finalizar esta sección es importante mencionar que: La Modalidad de Dictado de Seguridad de los Sistemas de Software es No presencial.

XIV - Otros