Taller de Software Seguro

Taller SOFTSEG

Este curso presenta una metodología integral acerca del desarrollo de software seguro. Abarca aspectos de los 3 ámbitos: personas, procesos y tecnologías necesarios para el desarrollo seguro durante todo el ciclo de vida de un proyecto de desarrollo de software. Desde el análisis de requerimientos hasta el retiro final, pasando por el diseño, implementación, liberación y operaciones. El curso cubre todos los conceptos y principios necesarios para la creación de software seguro. Dado que el software no es desarrollado en un ambiente aislado, este curso no sólo se enfoca en los aspectos de seguridad del desarrollo, también toma en cuenta aspectos relacionados a redes y nodos en los cuales el software se va a ejecutar. Adicionalmente define una estrategia de largo plazo para mejorar el estado general de la seguridad dentro de una organización, proporcionando soluciones tácticas. Este curso no está asociado a ninguna plataforma, tecnología o lenguaje en particular, sino que puede ser aplicado a cualquier proyecto de software.

Objetivos	Presentar de forma integral y holística los diferentes aspectos necesarios para el análisis, diseño, implementación, desarrollo, pruebas e implantación de forma segura. Mediante el entendimiento y aprendizaje de los principios necesarios para la creación de software seguro y confiable. El objetivo general de este curso consiste en brindar los conocimientos adecuados para llevar todo el ciclo de vida del desarrollo del software siguiendo los lineamientos, principios, recomendaciones y mejores prácticas para lograr software que sea lo más seguro posible sin descuidar otros aspectos fundamentales del mismo.
Audiencia	Este curso es ideal para desarrolladores de software y profesionales de seguridad, responsables de aplicar las mejores prácticas en cada fase del ciclo de vida del desarrollo de software. Así mismo es de gran ayuda a cualquier persona involucrada en cualquiera de las etapas del desarrollo de un proyecto de software.
Prerrequisitos	Para tomar este curso se recomienda tener por lo menos 2 años de experiencia como profesional en el ciclo de vida de desarrollo de software, en cualquiera de los ocho dominios por los que está conformado este curso.

ESQUEMA DEL CURSO

Dominio 1 - Conceptos de Software

Objetivos

Introducción

- 1.1 Seguridad Holística

1.1.1 Desafíos de implementación

1.1.2. Calidad y seguridad

- 1.2. Conceptos básicos de seguridad

1.2.1 Confidencialidad

1.2.2 Integridad

1.2.3 Disponibilidad

1.2.4 Autenticación

1.2.5 Autorización

1.2.6 No repudio

- 1.3. Conceptos de Diseño de Seguridad

1.3.1 Privilegio Mínimo

1.3.2 Principio de Separación de Deberes (O) Compartimentación

1.3.3 Defensa en Profundidad (O) Defensa en Capas

1.3.4 Fail Secure

1.3.5 Economía de mecanismos

1.3.6 Mediación completa

1.3.7 Diseño abierto

1.3.8 Mecanismos menos comunes

1.3.9 Aceptabilidad psicológica

1.3.10 Enlace más débil

1.3.11 Aprovechar los componentes existentes

- 1.4. Gestión de Riesgos

1.4.1 Términos y definiciones

1.4.2 Gestión de riesgos para el software

1.4.3 Manejo de riesgos

1.4.4 Concepto de gestión de riesgos: resumen

- 1.5. Políticas de seguridad: El "Qué" y el "Por qué" para la Seguridad

1.5.1 Alcance de las políticas de seguridad

1.5.2 Requisitos Previos para el Desarrollo de Políticas de Seguridad

1.5.2 Proceso de desarrollo de políticas de seguridad

- 1.6. Normas de seguridad

1.6.1 Tipos de normas de seguridad

1.6.2 Normas internas de codificación

1.6.3 Estándares NIST

1.6.4 Estándares federales de procesamiento de información (FIPS)

1.6.5 Normas ISO

1.6.6 Estándares PCI

1.6.7 Organización para el adelanto de Estándares de información estructurada (OASIS)

1.6.8 Beneficios de los estándares de seguridad

1.6.9 Mejores Prácticas

1.6.10 Proyecto de Seguridad de Aplicaciones Web Abiertas (OWASP)

1.6.11 Biblioteca de Infraestructura de Tecnología de la Información (ITIL)

- 1.7. Metodologías de desarrollo de software

1.7.1 Modelo en cascada

1.7.2 Modelos iterativos

1.7.3 Modelos de espiral

1.7.4 Metodologías de desarrollo ágil

- 1.8. Metodologías de Software Assurance

1.8.1 Método Socrático

1.8.2 Six Sigma (6 σ)

1.8.3 Integración del Modelo de Madurez de Capacidades (CMMI)

1.8.4 Evaluación operativamente crítica de amenazas, activos y vulnerabilidades (OCTAVE®)

1.8.5 AVANZAR y TEMER (STRIDE and DREAD)

1.8.6 Manual de Metodología (OSSTMM)

1.8.7 Método de Hipótesis de Falla (FHM)

1.8.8 Aplicación empresarial y marcos de seguridad

- 1.9. Regulaciones, Privacidad y Cumplimiento
- 1.10. Prisma Cloud

1.10.1 Enterprise Edition vs Compute Edition

- 1.11. SonarQube
- 1.12. AWS CloudFormation
- 1.13. Terraform – Azure DevOps

1.13.1 ¿Qué es Terraform?

1.13.2 Terraform components

1.13.3 Terraform en Azure

1.13.4 Instalación de Terraform

Resumen

Quiz

Referencias

Dominio 2: Requerimientos de Software Seguro

Objetivos

Introducción

- 2.1. Fuentes de requisitos de seguridad
- 2.2 Tipos de Requisitos de Seguridad

2.2.1 Requisitos de Seguridad Básicos

2.2.1.1 Requisitos de confidencialidad

2.2.2 Requisitos de Integridad

2.2.3 Requisitos de responsabilidad

2.2.4 Requerimientos Generales

- 2.2.4.1 Requisitos de gestión de sesión
- 2.2.4.2 Requisitos de gestión de errores y excepciones

- 2.2.5 Requerimientos Operacionales

2.2.5.1 Requisitos del entorno de implementación

2.2.5.2 Requisitos de archivo

2.2.5.3 Requisitos contra la piratería

- 2.2.6 Otros Requerimientos

2.2.6.1 Requisitos de secuencia y sincronización

.2.6.2 Requerimientos internacionales

2.2.6.3 Requisitos de adquisición

- 2.3 Elicitación de Necesidades de Protección (PNE)

2.3.1 Lluvia de Ideas

2.3.2 Encuestas (Cuestionarios y Entrevistas)

- 2.4 Descomposición de políticas
- 2.5 Tipos de datos

2.5.1 Etiquetado

2.5.2 Propiedad de los datos

2.5.3 Gestión del ciclo de vida de los datos (DLM)

- 2.6 Matriz Sujeto / Objeto

2.6.1 Modelado de casos de uso y mal uso

- 2.7. Matriz de Trazabilidad de Requisitos (RTM)

Resumen

Quiz

Práctica

Dominio 3: Diseño de Software Seguro

Objetivos

Introducción

- 3.1. La necesidad de un diseño seguro

3.1.1 Defectos contra errores

- 3.2 Software de Arquitectura con Conceptos Básicos de Seguridad
- 3.3 Diseño de confidencialidad

3.3.1 Algoritmos asimétricos

3.3.2 Certificados digitales

3.3.3 Firmas digitales

- 3.4 Diseño de Integridad

3.4.1 Hashing

3.4.2 Integridad referencial

3.4.3 Bloqueo de recursos

- 3.5 Diseño de Disponibilidad

3.5.1 Replicación

3.5.2 Conmutación por falla

- 3.6 Diseño de Escalabilidad
- 3.7 Diseño de Autenticación
- 3.8 Diseño de autorización
- 3.9 Diseño de Responsabilidad
- 3.10 Software de Arquitectura con Principios de Diseño Seguro
- 3.11 Otras Consideraciones de Diseño

3.11.1 Diseño de interfaz

3.11.2 API

3.11.3 SMI

3.11.4 Interfaz fuera de banda

3.11.5 Interfaes de registro

3.11.6 Interconectividad

- 3.12 Procesos de Diseño

3.12.1 Evaluación de la Superficie de Ataque

3.12.2 Modelado de amenazas

- 3.13 Arquitecturas

3.13.1 Arquitectura Mainframe

3.13.2 Computación distribuida

3.13.3 Arquitectura orientada a servicios

3.13.4 Aplicaciones de Internet enriquecidas

3.13.5 Computación Omnipresente / Ubicua

3.13.6 Computación en la Nube

- 3.14 Seguridad en Aplicaciones Móviles

3.14.1 Descubrimiento de información

3.14.2 Denegación de servicio móvil

3.14.3 Autenticación

3.14.4 Autorización

Dominio 4: Implementación de Software Seguro

4.1.1 Arquitectura de Computadores

4.1.2 Evolución de los lenguajes de programación

4.1.2.1 Leguajes compilados

4.1.2.2 Leguajes interpretados

4.1.2.3 Leguajes híbridos

- 4.2 Controles y Vulnerabilidades Comunes de Software

4.2.1 Desbordamiento de búfer

4.2.2 Desbordamiento de pila (Stack Overflow)

4.2.3 Desbordamiento de búfer (Heap Overflow)

4.2.4 Defectos de Inyección

4.2.5 Autenticación Rota y Gestión de Sesiones

4.2.6 Secuencias de Comandos Entre Sitios (XSS)

4.2.6.1 XSS No Persistente o Reflejado

4.2.6.2 XSS Persistente o Almacenado

4.2.6.3 XSS Basado en DOM

4.2.7 Referencias De Objetos Directos Inseguros

4.2.8 Mala Configuración de Seguridad

4.2.9 Exposición de Datos Sensibles

4.2.10 Protección Insuficiente de Datos en Movimiento

4.2.11 Falsificación De Solicitudes Entre Sitios (CSRF)

4.2.12 Uso de Componentes Vulnerables Conocidos

4.2.13 Redirecciones y Reenvíos No Validados

4.2.14 Ataques de Archivos

4.2.15 Condición de Carrera (Race Condition)

4.2.16 Ataques de Canal Lateral

- 4.3 Prácticas De Codificación Defensiva: Conceptos y Técnicas

4.3.1 Validación de Entrada

4.3.2 Canonicalización

4.3.3 Higienización/Sanitización

4.3.4 Manejo de errores

4.3.5 APIs seguras

Dominio 5: Pruebas de Software Seguro

Dominio 6: Administración del ciclo de vida de software seguro

Dominio 7: Mantenimiento, Operaciones y Despliegue Seguro de Software

7.4.7 Análisis posterior al incidente

- 7.5 Manejo de problemas
- 7.6 Gestión del cambio
- 7.7 Gestión de parches y vulnerabilidades
- 7.8 Copias de seguridad, recuperación y archivo
- 7.9 Disposición
- 7.10 Políticas de fin de vida
- 7.11 Criterios de puesta de sol
- 7.12 Procesos de ocaso (obsolescencia)
- 7.13 Eliminación de información y desinfección de medios

Resumen

Quiz

Referencias

Dominio 8: Cadena de suministro y adquisición de software

Objetivos

Introducción

- 8.1 Adquisición de software y cadena de suministro.
- 8.2 Ciclo de vida de la adquisición
- 8.3 Modelos y beneficios de adquisición de software

8.3.1 Outsourcing

8.3.2 Servicios administrados

- 8.4 Metas del software de la cadena de suministro
- 8.5 Amenazas al software de la cadena de suministro

8.5.1 Amenazas de productos / datos:

8.5.2 Procesos / amenazas de flujo:

8.5.3 Amenazas a las personas

- 8.6 Gestión de riesgos de la cadena de suministro de software (SCRM)
- 8.7 Evaluación y gestión de riesgos de proveedores
- 8.8 Evaluación de respuesta
- 8.9 Controles contractuales

- 8.10 Propiedad y responsabilidades de la propiedad intelectual (PI)

8.10.1 Tipos de propiedad intelectual (PI)

- 8.11 Licencia (condiciones de uso y redistribución)

8.11.1 Código cerrado (Closed source)

8.11.2 Código abierto (Open source)

- 8.12 Desarrollo y prueba de software
- 8.13 Entrega, operaciones y mantenimiento de software

Resumen

Quiz

Referencias

Apendice A

- P&R Dominio 1 - Conceptos de Software Seguro
- P&R Dominio 2 - Requisitos de Software Seguro
- P&R Dominio 3 - Diseño de Software Seguro
- P&R Dominio 4 - Implementación Codificación Segura de Software
- P&R Dominio 5 - Prueba de Software Segura
- P&R Dominio 6 - Aceptación del Software
- P&R Dominio 7 - Implementación, Operaciones, Mantenimiento y Eliminación de Software
- P&R Dominio 8 - Cadena de suministro y Adquisición de Software

Debido a las constantes actualizaciones de los contenidos de los cursos por parte del fabricante, el contenido de este temario puede variar con respecto al publicado en el sitio oficial, sin embargo, Netec siempre entregará la versión actualizada de éste