Desarrollo Seguro y Operaciones
DEVSECOPS_JAV

Duración 4 Days

Este curso abarca el proceso de desarrollo de software seguro, alineado con la metodología DevSecOps. Se incorporan diversas herramientas, el lenguaje de programación Java, así como demostraciones y actividades prácticas para fortalecer el aprendizaje.Al finalizar, sabrás cómo implementar un proceso de desarrollo de software seguro, integrando prácticas de codificación segura, distintos tipos de pruebas (como análisis estático y dinámico), además de una serie de buenas prácticas para el desarrollo y monitoreo.

Objetivos del curso
Al finalizar el curso, serás capaz de:
Implementar un proceso de desarrollo de forma segura siguiendo la metodología DevSecOps.

Perfil de audiencia
Programadores.
Equipos de Operaciones.
Personas involucradas en el mantenimiento de aplicaciones y/o sistemas.
Analistas de sistemas.

Prerrequisitos
Conocimientos previos en programación con Java.
Experiencia en metodologías de desarrollo ágil.
Conocimiento de algunas herramientas comunes en el proceso de desarrollo

Versión de la tecnología
Spring boot 3.4.5
Docker 28.0.4
Kubernetes1.32
Java 17

Temario del Curso

Capítulo 1: Introducción a DevSecOps

- 1.1. ¿Qué es DevSecOps?
  - 1.1.1. Definición y objetivos
  - 1.1.2. Diferencias entre DevOps y DevSecOps
  - 1.1.3. Importancia de la seguridad en el desarrollo de software
- 1.2. Principios fundamentales de DevSecOps
  - 1.2.1. Principios de DevSecOps
  - 1.2.2. Conceptos de seguridad
- 1.3. Introducción a las amenazas comunes
  - 1.3.1. OWASP Top
  - 101.3.2. CWE/SANS Top 25
  - 1.3.3. MITRE ATTACKResumenReferencias bibliográficas

Desarrollo Seguro y Operaciones
DEVSECOPS_JAV

Duración 4 Days

Capítulo 2: Fundamentos de seguridad en el ciclo de vida del software

- 2.1. Seguridad en el ciclo de vida del desarrollo de software (SDLC)
- 2.2. ¿Qué es el modelado de amenazas?
  - 2.2.1. Identificación de activos, amenazas y vulnerabilidades
  - 2.2.2. Evaluación de riesgos
- 2.3. Modelos de amenazas
  - 2.3.1. STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege)
  - 2.3.2. DREAD (Damage, Reproducibility, Exploitability, Affected Users, Discoverability)
- 2.4. Herramientas para modelado de amenazas
  - 2.4.1. Microsoft Threat Modeling Tool
  - 2.4.2. OWASP Threat Dragon
- 2.5. Principios de diseño seguro
  - 2.5.1. Minimización de la superficie de ataque
  - 2.5.2. Principio de privilegio mínimo
  - 2.5.3. Defensa en profundidad
  - 2.5.4. Falla segura (Fail Secure)
  - 2.5.5. Validación de entradas
  - 2.5.6. Gestión de sesiones y autenticación
  - 2.5.7. Cifrado de datos sensibles
  - 2.5.8. Ejemplos de diseños inseguros y cómo evitarlos
  - 2.5.9. Codigo limpio
  - 2.5.10. Patrones de diseño

Práctica

1. Modelado de amenazas con Microsoft Threat Modeling Tool

Capítulo 3: Seguridad en APIs

- 3.1. Introducción a la seguridad en APIs
  - 3.1.1. Tipos de APIs (REST, SOAP)
  - 3.1.2. Principales riesgos en APIs (OWASP API Security Top 10)
- 3.2. Prácticas seguras en el diseño de APIs
  - 3.2.1. Validación de entradas y salidas
  - 3.2.2. Autenticación y autorización en APIs
  - 3.2.3. Uso de HTTPS y cifrado
- 3.3. Pruebas de seguridad en APIs
  - 3.3.1. Herramientas como Postman, OWASP ZAP, Insomnia

Práctica

2. Análisis de seguridad y codificación de un API Rest Spring Boot

Capítulo 4: Gestión de dependencias y librerías

- 4.1. Riesgos asociados a dependencias externas
  - 4.1.1. Vulnerabilidades en librerías de terceros
  - 4.1.2. Ejemplos de ataques como Log4Shell

Desarrollo Seguro y Operaciones
DEVSECOPS_JAV

Duración 4 Days

4.2. Gestión segura de dependencias
4.2.1. Uso de herramientas como Snyk, OWASP Dependency-Check
4.2.2. Actualización y monitoreo continuo de dependencias
4.3. Licencias de software y su impacto en la seguridad

Práctica

2. Analizar dependencias y librerias del microservicio cliente usando Snyk y OWASP Dependency Check

Capítulo 5: Gestión de secretos y configuraciones

5.1. Riesgos asociados a la exposición de secretos
5.1.1. Claves API, contraseñas, tokens
5.2. Herramientas para la gestión de secretos
5.2.1. AWS Secrets Manager, Azure Key Vault
5.3. Prácticas seguras en la configuración de aplicaciones
5.3.1. Separación de configuraciones sensibles del código
5.3.2. Uso de variables de entorno

Práctica

3. Gestión de secretos con Azure Keyvault del microservicio cliente

Capítulo 6: Integración de seguridad en CI/CD

6.1. Introducción a CI/CD (Integración y Entrega Continua)
6.2. Pruebas6.2.1. TDD6.2.2. Code review
6.2.3. Escaneo de código estático (SAST). Herramientas como SonarQube
6.2.4. Escaneo dinámico de aplicaciones (DAST). Herramientas: OWASP ZAP
6.2.5. Análisis de composición de software (SCA)
6.3. Automatización de pruebas de seguridad
6.3.1. Herramientas como SonarQube, OWASP ZAP, Snyk, etc.
6.4. Gestión de secretos y credenciales en pipelines

Práctica

4. Añadir seguridad en CI/CD para el microservicio cliente

Capítulo 7: Seguridad en el despliegue

7.1. Introducción a la seguridad en entornos de despliegue
7.2. Configuración segura de servidores y entornos
7.2.1. Hardening de servidores
7.2.2. Herramientas de gestion de postura de seguridad

Práctica

5. Hardening básico en Ubuntu:20.04

Capítulo 8: Seguridad en Infraestructura como Código (IaC)

8.1. Introducción a IaC y su importancia en DevSecOps
8.2. Principales herramientas de IaC: Terraform, Ansible, CloudFormation
8.3. Escaneo y auditoría de IaC
8.3.1. Detectar configuraciones inseguras
8.3.2. Herramientas como Checkov y Terrascan

Desarrollo Seguro y Operaciones
DEVSECOPS_JAV

Duración 4 Days

8.4. Prácticas seguras en IaC
8.4.1. Gestión de permisos y roles
8.4.2. Políticas de seguridad en la nube

Práctica

6. Configuración y uso de Checkov para el escaneo de una plantilla de Terraform

Capítulo 9: Seguridad en Contenedores y Kubernetes

9.1. Introducción a contenedores y Kubernetes
9.2. Seguridad en contenedores
9.2.1. Escaneo de imágenes de contenedores (Trivy)
9.2.2. Gestión de vulnerabilidades en imágenes
9.3. Seguridad en Kubernetes
9.3.1. Configuración segura de clústeres
9.3.2. Políticas de red y RBAC (Control de Acceso Basado en Roles)
9.3.3. Herramientas como kubescape
9.4. Gestión de secretos en contenedores y Kubernetes

Práctica

7. Análisis de seguridad de Docker y Kubernetes usando Trivy, Checkov y kubescape

Capítulo 10: Seguridad en la nube

10.1. Introducción a la seguridad en entornos cloud
10.2. Principales riesgos en la nube (según OWASP Cloud Top 10)
10.3. Herramientas de monitoreo y auditoría en la nube
10.3.1. AWS Security Hub, Azure Security Center, Google Cloud Security Command Center
10.4. Prácticas seguras en la nube
10.4.1. Gestión de identidades y accesos (IAM)
10.4.2. Configuración de firewalls y políticas de red

Práctica

8. Explorando opciones de seguridad en la nube con Azure Security Center

Capítulo 11: Monitoreo y respuesta a incidentes

11.1. Introducción al monitoreo continuo en DevSecOps
11.2. Herramientas de monitoreo y logging
11.2.1. Prometheus, Grafana.
11.3. Respuesta a incidentes en entornos DevOps
11.3.1. Planificación y ejecución de planes de respuesta1
1.3.2. Automatización de respuestas a incidentes
11.4. Gestión de vulnerabilidades y parches

Práctica

9. Monitoreo de microservicio cliente con Grafana y Prometheus