INGENIERÍA DE SEGURIDAD US
E.T.S. Ingeniería Informática. Universidad de Sevilla
Avda Reina Mercedes s/n. 41012 Sevilla
Tlf/Fax 954 557 139 **E-mail** lsi@lsi.us.es **Web**
www.lsi.us.es

PAI-3. SEGURIDAD DEL SOFTWARE EN EL DESARROLLO DE SISTEMAS

INTRODUCCIÓN

Se pretende verificar las posibles **vulnerabilidades de seguridad** que existen en varios desarrollos debido a múltiples razones. Se propone detectar por ejemplo:

- Acceso más allá de la memora asignada (buffer overflow)
- Underflow y Overflow de tipos
- Lectura de objetos no inicializados
- No validación de entradas
- Fuga de recursos o de información
- Fallo en los valores devueltos en los métodos
- Liberación adecuada de todos los recursos
- Buffer y array underflow
- Path Manipulation

Inicialmente las buenas prácticas para el análisis de código fuente consistían en leer línea a línea para verificar la calidad, rendimiento, tamaño y productividad, pero esto no es suficientemente efectivo, ni fiable hoy en día, pues se trata de resolver dos problemas fundamentales relacionados con las interacciones entre componentes software:

1. Los modernos sistemas TI constan de cientos de componentes, elaborados por equipos diferentes y docenas de desarrolladores. Medir la calidad del software a través de sistemas requiere múltiples tecnologías, y la clara identificación de los límites de las aplicaciones.
2. Los bugs más insidiosos y peligrosos de las interacciones entre componentes software de los sistemas complejos no se pueden detectar por la mayoría de las herramientas que son desplegadas por desarrolladores personales, por lo que se requiere herramientas más sofisticadas.

En la actualidad existen multitud de herramientas que sirven para analizar el código fuente para diferentes lenguajes de programación para buscar vulnerabilidades, pudiendo citarse los siguientes:

- **Open Source**, *SonarQube, Yasca, VisualCodeGrepper, Agnitio, Coverity, RIPS, OWASP Code Crawler, Findbugs, etc.*
- **Comerciales**, *AppScan, bugScout, Veracode, SafeVal, Kiuwan, HP Fortify, etc.*

Algunas de las estas herramientas se basan en los **estándares de seguridad de mayor prestigio** en la industria tales como **CWE**, **CVSS**, **OWASP** y **WASC (**Web Application Security Consortium**).** Frecuentemente producen resultados que son **falsos positivos** donde la herramienta informa sobre una vulnerabilidad que no es tal, esto ocurre pues la herramienta no puede asegurar la seguridad de los datos fluyen a través de la aplicación desde la entrada a la salida. También pueden resultar **falsos negativos** donde la herramienta no informa de determinadas

vulnerabilidades, esto puede ocurrir por no estar configurada la herramienta con la regla correspondiente a dicha vulnerabilidad.

Objetivos

Cada Equipo deberá analizar el su código presentado al PAI-2 y/o algún proyecto desarrollado en otra asignatura, y los dos proyectos desarrollados que se proponen en EV (C y Java).

- **Tarea 1. Detección y solución. Listado de vulnerabilidades de ciberseguridad identificadas en el código fuente**, indicando la localización de los mismos, **el grado de prioridad de cada una a la hora de tener un ranking para resolverlas y la dimensión de seguridad que se podría ver comprometida**, y **propuestas de soluciones** para cada una de las vulnerabilidades detectadas (seleccione las 6 vulnerabilidades que considere más importantes en cada proyecto).
- **Tarea 2. Toma de Decisión Tecnológica ¿Cuáles son las alternativas tecnológicas (herramientas/plugins de análisis estático) que han estimado más convenientes para la resolución de las consultas y los criterios técnicos que les han llevado a determinar finalmente una o varias de ellas para cada lenguajes estudiado?.** En el análisis riguroso que se haya realizado de las diferentes alternativas tecnológicas debe tener en cuenta la eficiencia (tiempo necesario), eficacia (número de vulnerabilidades obtenidas), tasa de falsos positivos, usabilidad, y el informe de resultados obtenidos. **(estudiar al menos 2 herramientas)**

Normas del entregable

- Cada grupo debe entregar a través de la Plataforma de Enseñanza Virtual y en la **actividad** preparada para ello un archivo zip, nombrado ***PAI3-EquipodeTrabajoX*.zip**, que deberá contener al menos los ficheros siguientes:
  - ✓ **Documento en formato pdf que contenga un informe/resumen del proyecto** con los detalles más importantes de las decisiones, soluciones adoptadas y/o implementaciones desarrolladas, así como el resultado y análisis de las pruebas realizadas (máximo 15 páginas).
  - ✓ **Código fuente analizados y configuraciones de las herramientas utilizadas.**
- El plazo de entrega de dicho proyecto finaliza el **día 30 de noviembre a las 8:30 horas.**
- Los proyectos entregados fuera del plazo establecidos serán considerados inadecuados por el cliente y por tanto entrarán en penalización por cada día de retraso entrega de 15% del total, hasta agotarse los puntos.
- **El cliente no se aceptará envíos realizados por email, ni mensajes internos de la enseñanza virtual, ni correo interno de la enseñanza virtual.**

Métricas de valoración

Para facilitar el desarrollo de los equipos de trabajo el cliente ha decidido listar las métricas que se tendrán en cuenta para valorar los entregables de cada grupo de trabajo:

- **Documento (30%)**
  - Calidad del informe presentado y justificaciones
- **Solución aportada (70%)**
  - Cumplimiento de requisitos establecidos
  - Respuesta al conjunto de preguntas planteadas
  - Calidad de pruebas realizadas y resultados