PROGRAMACIÓN ANDROID

Actividad 2.5: Manejo de elementos de seguridad

I. ANTECEDENTES GENERALES

Recinto:	Laboratorio de informática.
Aprendizaje esperado:	Resolver situaciones de riesgo, detectando y corrigiendo vulnerabilidades durante el proceso de desarrollo de App.
Criterios de evaluación:	- Testea App en Java, para Android, según especificaciones acordadas. - Corrige App en Java, para Android, según especificaciones. - Cumple de manera responsable y a lo largo del tiempo, los compromisos adquiridos, en diversos contextos, presentando un trabajo bien hecho.

II. PRESENTACIÓN DE LA ACTIVIDAD

En esta actividad de investigación sumativa, analizaran los elementos de vulnerabilidad en el proceso de desarrollo de una app, realizando tests de vulnerabilidad y trabajando los elementos de Best Practices, Security Tips y Security Improvement Program.

III. INSTRUCCIONES

- Tipo de actividad: Sumativa (20%).
- Modalidad de trabajo: Individual.
- Preparación: Familiarízate con las APIs y servicios de seguridad en Android y revisa las herramientas de análisis de seguridad recomendadas por tu docente, como OWASP ZAP, MobSF, y SonarQube.
- Desarrollo:
  - Integrar Google Maps.
  - Implementación de funcionalidades.
  - Subir proyecto a GitHub.
- Entregables: Enlace al repositorio de GitHub con el código de la aplicación y archivo README.md con la documentación del proyecto.
- Plazo de entrega: Según fecha definida por docente.

IV. DESARROLLO DE LA ACTIVIDAD

Análisis de Vulnerabilidades

1. Identificación de vulnerabilidades:
   - Realizar un análisis de seguridad en la aplicación Android, conforme a lo siguiente:
     Existen varios análisis, uno de ellos es para Aplicaciones Móviles:
     Utilizando MobSF (Mobile Security Framework) para realizar un análisis estático y dinámico específico para aplicaciones móviles.
     Con MobSF:
     Descarga el APK de la aplicación Android.
     Carga el APK en MobSF para realizar un análisis estático.
     Revisa los informes generados por MobSF, que incluyen detalles sobre permisos, configuraciones inseguras, dependencias inseguras, etc.
     Realiza un análisis dinámico utilizando un emulador o dispositivo físico conectado a MobSF.
     Documenta las vulnerabilidades encontradas.

```
# Vulnerabilities Report

## Summary
- Total vulnerabilities found: X
- Critical: Y
- High: Z
- Medium: A
- Low: B

## Detailed Findings

### 1. Vulnerability Title
- **Description:** Detailed description of the vulnerability.
- **Severity:** Critical/High/Medium/Low
- **Impact:** Potential impact of the vulnerability.
- **Steps to Reproduce:** Steps to replicate the vulnerability.
- **Remediation:** Suggested fixes or mitigations.

### 2. Vulnerability Title
- **Description:** Detailed description of the vulnerability.
- **Severity:** Critical/High/Medium/Low
- **Impact:** Potential impact of the vulnerability.
- **Steps to Reproduce:** Steps to replicate the vulnerability.
- **Remediation:** Suggested fixes or mitigations.

... (repeat for each vulnerability)
```

   - Utilizar herramientas de análisis estático y dinámico para identificar posibles vulnerabilidades.
   - Documentar las vulnerabilidades encontradas en un archivo vulnerabilities.md.

2. Tests de vulnerabilidad:
   - Realizar pruebas de vulnerabilidad utilizando herramientas como OWASP ZAP, MobSF, o cualquier otra herramienta de seguridad.
   - Generar un reporte detallado de las pruebas realizadas y los resultados obtenidos.
   - Guardar el reporte en el repositorio Github con el nombre vulnerability_report.pdf.

Implementación de Best Practices

1. Aplicar Prácticas Recomendadas:
   - Asegurar el código y la infraestructura de la aplicación siguiendo las Best Practices (Buenas Prácticas).
   - Implementar cifrado para datos sensibles.
   - Asegurar la comunicación de red utilizando HTTPS.
   - Validar y sanitizar todas las entradas del usuario.

2. Documentación de Best Practices:
   - Crear un documento best_practices.md que detalle las Best Practices implementadas en la aplicación.
   - Explicar cómo cada práctica mejora la seguridad de la aplicación.

Security Tips

1. Implementar consejos de Seguridad:
   - Integrar medidas específicas para mejorar la seguridad de la aplicación basadas en los Security Tips.
   - Proteger la aplicación contra ataques de inyección SQL.
   - Implementar autenticación y autorización seguras.
   - Proteger la aplicación contra ataques de red (e.g., MITM).

2. Documentación de Security Tips:
   - Crear un documento security_tips.md que detalle los Security Tips implementados.
   - Explicar cómo cada tip mejora la seguridad de la aplicación.

App Security Improvement Program

1. Desarrollo de un Programa de Mejora de Seguridad:
   - Diseñar un programa estructurado para evaluar y mejorar la seguridad de la aplicación.
   - Establecer un proceso de revisión periódica de la seguridad.
   - Identificar métricas clave para medir la seguridad de la aplicación.

2. Documentación del Programa de Mejora de Seguridad:
   - Crear un documento security_improvement_program.md que detalle el programa y las mejoras realizadas.
   - Incluir un plan de acción para futuras mejoras de seguridad.

Entrega:

- Subir todo el código, documentación y reportes al repositorio GitHub.
- Asegurarse de que el repositorio esté bien organizado y documentado.
- Incluir un archivo README.md con una descripción del proyecto, las vulnerabilidades identificadas, las mejoras implementadas y cómo ejecutar la aplicación de manera segura.

Ejemplo de README.md:

```
# Nombre del Proyecto #

# Descripción
Este proyecto es una aplicación Android que implementa medidas de seguridad para proteger contra vulnerabilidades comunes.

## Vulnerabilidades Identificadas
- Inyección SQL
- Comunicación no segura
- Exposición de datos sensibles

## Mejoras Implementadas
- Cifrado de datos sensibles
- Comunicación segura (HTTPS)
- Validación y sanitización de entradas

## Documentación
- [Vulnerabilidades](vulnerabilities.md)
- [Best Practices](best_practices.md)
- [Security Tips](security_tips.md)
- [Security Improvement Program](security_improvement_program.md)

## Cómo Ejecutar la Aplicación de Forma Segura
1. Clonar el repositorio
2. Importar el proyecto en Android Studio
3. Ejecutar la aplicación en un dispositivo o emulador
4. Asegurarse de que los permisos necesarios están configurados

## Reporte de Vulnerabilidades El reporte detallado de las pruebas de vulnerabilidad realizadas se encuentra en el archivo `vulnerability_report.pdf`.
```

Preguntas de cierre

1. ¿Qué nuevos conocimientos y habilidades has adquirido en la protección de aplicaciones Android contra amenazas de seguridad?
2. ¿Cómo podrías utilizar los conocimientos adquiridos para mejorar la seguridad en aplicaciones móviles en diferentes entornos, como aplicaciones financieras, de salud, o de comercio electrónico?
3. ¿Lograste implementar las mejores prácticas y los consejos de seguridad de manera efectiva? ¿Qué desafíos encontraste en el proceso y cómo los superaste?