**Propuesta de Proyecto: Aplicación de Seguridad Informática para Análisis de Vulnerabilidades**

**Descripción del Proyecto**

Desarrollo de una aplicación modular que centralice herramientas de seguridad informática para el análisis de vulnerabilidades en páginas web y entornos cloud, con un enfoque en un sistema innovador de "toquen de antesala". La aplicación priorizará funcionalidades esenciales debido a las limitaciones de tiempo (320 horas totales, 80 horas por persona para un equipo de 4). Se proporcionarán reportes básicos de vulnerabilidades detectadas y recomendaciones para su mitigación.

**Objetivos**

1. **Centralización básica**: Integrar herramientas de código abierto para escanear vulnerabilidades en páginas web y entornos cloud.
2. **Prototipo de "toquen de antesala"**: Implementar un módulo inicial que analice y clasifique archivos antes de su almacenamiento.
3. **Mitigación de riesgos**: Generar recomendaciones básicas para corregir vulnerabilidades detectadas.
4. **Diferenciación**: Posicionar la herramienta como una solución proactiva y modular, destacando el sistema de toquenes.

**Innovación: Sistema de "Toquen de Antesala"**

El "toquen de antesala" es un filtro inicial que analiza archivos entrantes antes de su almacenamiento. Para este proyecto, se implementará una versión simplificada con las siguientes características:

* **Análisis de tipo de archivo**: Identifica formatos comunes (p. ej., .js, .pdf, .yaml) y evalúa riesgos básicos.
* **Encaminamiento**: Dirige archivos a un módulo de análisis específico según su tipo.
* **Inocuación básica**: Cuarentena archivos sospechosos basándose en reglas simples (p. ej., tamaño, extensión).
* **Interconexión limitada**: Un único toquen secundario para análisis, debido a restricciones de tiempo.

**Valor Añadido y Posicionamiento**

**Ventaja Competitiva**

* **Proactividad simplificada**: El toquen de antesala ofrece un enfoque preventivo, aunque limitado en esta fase inicial.
* **Modularidad**: Diseñado para ser escalable, permitiendo agregar más toquenes en el futuro.
* **Facilidad de uso**: Interfaz intuitiva para usuarios no expertos, combinando herramientas existentes en una sola plataforma.

**Posicionamiento**

* **Público objetivo**: Pymes y desarrolladores individuales que necesitan una solución sencilla para análisis de vulnerabilidades.
* **Diferenciador**: Sistema de toquenes como base para una solución proactiva, frente a herramientas reactivas como Nessus o Burp Suite.
* **Limitaciones**: Debido al alcance reducido, la herramienta competirá en usabilidad y potencial de escalabilidad, no en profundidad de análisis.

**Alcance Técnico (Realista para 320 Horas)**

**Tecnologías Propuestas**

* **Backend**: Python con FastAPI (rápido y ligero) para el sistema de toquenes y API.
* **Frontend**: React con Tailwind CSS para una interfaz básica y responsiva.
* **Análisis de vulnerabilidades**:
  + Web: OWASP ZAP (escaneo automatizado de sitios web).
  + Cloud: ScoutSuite (análisis de configuraciones en AWS, limitado a credenciales preconfiguradas).
* **Base de datos**: SQLite (ligera, suficiente para prototipo).
* **Seguridad de archivos**: Reglas básicas con YARA para el toquen de antesala.

**Funcionalidades Priorizadas**

1. **Toquen de antesala** (80 horas):
   * Clasificación de archivos por extensión y tamaño.
   * Cuarentena de archivos sospechosos.
   * Encaminamiento a un toquen secundario para análisis.
2. **Análisis de vulnerabilidades** (100 horas):
   * Escaneo web con OWASP ZAP (limitado a URLs proporcionadas por el usuario).
   * Escaneo cloud con ScoutSuite (solo AWS, configuraciones básicas).
3. **Interfaz de usuario** (80 horas):
   * Panel para iniciar escaneos y cargar archivos.
   * Visualización de resultados en formato tabular.
4. **Reportes** (60 horas):
   * Generación de reportes en texto con vulnerabilidades detectadas y recomendaciones predefinidas.

**Flujo del Sistema**

1. **Carga de archivo**: El usuario sube un archivo o ingresa una URL/configuración cloud.
2. **Toquen de antesala**: Analiza el archivo, lo clasifica y decide si pasa a cuarentena o al toquen secundario.
3. **Análisis**: El toquen secundario usa OWASP ZAP o ScoutSuite para generar resultados.
4. **Reporte**: Se muestra un informe en la interfaz con vulnerabilidades y mitigaciones básicas.

**Plan de Desarrollo (Colaborativo)**

**Equipo y Roles**

* **Persona 1**: Backend (FastAPI, toquen de antesala, integración con YARA).
* **Persona 2**: Análisis de vulnerabilidades (configuración de OWASP ZAP y ScoutSuite).
* **Persona 3**: Frontend (React, Tailwind CSS, diseño de interfaz).
* **Persona 4**: Reportes y coordinación (generación de reportes, pruebas, documentación).

**Cronograma**

* **Semana 1 (80 horas)**:
  + Configuración inicial: Entorno de desarrollo, FastAPI, React, SQLite.
  + Investigación: Configuración de OWASP ZAP, ScoutSuite, YARA.
  + Asignación de tareas detalladas.
* **Semana 2 (80 horas)**:
  + Desarrollo del toquen de antesala (clasificación y cuarentena).
  + Configuración inicial de OWASP ZAP y ScoutSuite.
  + Prototipo de interfaz (panel básico).
* **Semana 3 (80 horas)**:
  + Integración del toquen secundario con herramientas de análisis.
  + Desarrollo de la interfaz (visualización de resultados).
  + Generación de reportes básicos.
* **Semana 4 (80 horas)**:
  + Pruebas funcionales: Escaneos, análisis de archivos, reportes.
  + Corrección de errores y optimización.
  + Documentación final y presentación.

**Requerimientos**

* **Hardware**: Computadoras personales con Docker para pruebas de OWASP ZAP y ScoutSuite.
* **Software**: Python 3.9+, Node.js, React, SQLite, OWASP ZAP, ScoutSuite, YARA.
* **Limitaciones**: No se implementarán toquenes complejos ni análisis avanzados debido al tiempo disponible.

**Siguientes Pasos**

1. Configurar el entorno de desarrollo colaborativo (GitHub, Docker).
2. Realizar una reunión inicial para confirmar roles y cronograma.
3. Desarrollar el prototipo del toquen de antesala como prueba de concepto.