**Propuesta Ajustada para el Proyecto de Fin de Curso:** Plataforma de Seguridad Modular

**Equipo:** 4 personas

**Tiempo disponible:** 80 horas por persona (320 horas totales)

**Objetivo:**  Desarrollar un MVP (Minimum Viable Product) centrado en el "toquen de antesala" y un módulo básico de análisis de vulnerabilidades web, asegurando que sea funcional, demostrable y escalable.

📌 **Alcance Realista** (Priorizado)

Dado el tiempo limitado, nos enfocaremos en:

**1. Sistema de Toquen de Antesala** (Core del Proyecto)

✅ **Funcionalidad básica:**

- Analizar archivos antes de guardarlos en disco (sandboxing).

- Clasificar archivos (PDFs, scripts, ejecutables, etc.).

- Enrutar a módulos específicos (ej: si es un PDF, pasarlo a un "toquen" de análisis de documentos).

- Inoculación básica: Eliminar/metadatos sospechosos o bloquear archivos peligrosos.

🔧 **Tareas técnicas:**

- Crear un microservicio/API que reciba archivos y los procese.

- Implementar reglas simples (ej: firmas de malware conocidas, análisis de metadatos).

- Logs de actividad para auditoría.

**2. Módulo Básico de Análisis Web (Complementario)**

✅ **Funcionalidad básica:**

- Escanear URLs en busca de vulnerabilidades \*\*básicas\*\* (ej: detección de formularios sin CSRF, cabeceras inseguras).

- Generar un \*\*reporte simple\*\* con hallazgos y recomendaciones de mitigación.

🔧 **Tareas técnicas:**

- Usar herramientas open-source como \*\*OWASP ZAP (API)\*\* o scripts personalizados.

- Interfaz mínima (CLI o web simple) para ingresar una URL y ver resultados.

**3. Interfaz de Usuario** (Mínima pero Funcional)

- Consola de administración (CLI o web básica con Flask/FastAPI).

- Visualización de logs y reportes de análisis.

📅 **Distribución de Tareas** (Equipo de 4)

|  |  |  |
| --- | --- | --- |
| Rol | Tareas Principales | Tiempo |
| Backend (Toquen) | Diseñar API de procesamiento, lógica de enrutamiento, sanitización básica de archivos | 80h |
| Seguridad/Scanner | Integrar OWASP ZAP o scripts de análisis web, generar reportes | 80h |
| Frontend/CLI | Interfaz mínima (web o consola) para subir archivos y ver resultados | 60h |
| DevOps/Testing | Configurar entorno (Docker), pruebas de seguridad, logs | 60h |

(Las horas restantes se usarán para integración, pruebas y documentación.)

⚙️ **Stack Tecnológico** (Sencillo pero Eficiente)

- Lenguaje: Python (ideal para prototipado rápido).

- Backend: FastAPI (para el toquen de antesala) o Flask.

- Scanner Web: OWASP ZAP (API) o scripts con `requests`/`beautifulsoup`.

- Frontend:

- Opción 1: CLI con `argparse`/`click` (rápido).

- Opción 2: Web simple con HTML/JS o Streamlit (si hay tiempo).

- Despliegue: Docker (para empaquetar fácilmente).

📌 **Entregables Finales**

1. Toquen de Antesala Funcional (puede recibir un archivo, analizarlo y decidir si bloquearlo o limpiarlo).

2. Scanner Web Básico (analiza una URL y devuelve vulnerabilidades comunes).

3. Documentación Técnica (README, diagrama de flujo del toquen).

4. Demo Funcional (vídeo o presentación mostrando el sistema en acción).

🔎 **Posibles Extensiones** (Si Queda Tiempo)

- Añadir más tipos de análisis al toquen (ej: detección de macros maliciosas en Office).

- Integrar un dashboard con gráficos de resultados.

- Conexión con APIs de virustotal o hybrid-analysis.

🎯 **Conclusión**

Este enfoque garantiza un producto viable en el tiempo disponible, centrado en la innovación del "toquen de antesala" pero con un módulo de análisis web para demostrar utilidad práctica.

¡Es escalable! – Los demás módulos (cloud, OSINT, etc.) podrían añadirse en futuras iteraciones.