Para abordar en profundidad cada uno de los puntos clave en la definición y documentación de una arquitectura modular para tu proyecto de seguridad informática, te detallo el proceso, buenas prácticas y ejemplos aplicados a tu caso:

1. Identificar y documentar los módulos funcionales

Proceso y Buenas Prácticas

Analiza las funcionalidades principales del sistema y sepáralas en módulos independientes, cada uno con una única responsabilidad clara.

Ejemplo aplicado a tu proyecto:

Módulo de Análisis Web: Encargado de escanear vulnerabilidades en aplicaciones web.

Módulo de Análisis Cloud: Analiza recursos y configuraciones en entornos cloud.

Módulo Toquen de Antesala: Realiza un pre-análisis de archivos antes de su almacenamiento.

Módulo de Gestión de Resultados: Almacena y organiza los resultados de los análisis.

Módulo de Interfaz Web: Presenta la información y permite la interacción con el usuario.

Módulo de Mitigación y Recomendaciones: Genera y presenta las acciones correctivas ante vulnerabilidades detectadas.

Checklist

Listar todos los módulos funcionales.

Definir la responsabilidad única de cada módulo.

Documentar entradas, salidas y función principal de cada módulo.

2. Definir interfaces y protocolos de comunicación

Proceso y Buenas Prácticas

Cada módulo debe funcionar como una “caja negra”: solo expone lo necesario a través de interfaces bien definidas.

Define para cada módulo:

Entradas: Qué datos recibe y en qué formato.

Salidas: Qué datos devuelve y en qué formato.

Protocolo: ¿Se comunican por API REST, mensajes, archivos temporales, base de datos compartida, etc.?

Ejemplo:

El módulo de análisis web recibe una URL vía API REST y devuelve un JSON con los resultados.

El toquen de antesala recibe archivos por una ruta temporal, los analiza y notifica al módulo de gestión de resultados mediante un mensaje o evento.

Checklist

Definir para cada módulo sus entradas y salidas.

Especificar el protocolo de comunicación (REST, archivos, eventos, etc.).

Documentar ejemplos de uso de cada interfaz.

3. Asignar responsables de desarrollo para cada módulo

Proceso y Buenas Prácticas

Asigna a cada integrante del equipo un módulo principal, según su experiencia o interés.

Cada responsable debe:

Liderar el desarrollo y pruebas de su módulo.

Documentar el funcionamiento y la integración.

Coordinarse con los responsables de módulos relacionados.

Checklist

Asignar un responsable por módulo.

Documentar los responsables en el plan de proyecto.

Establecer canales de comunicación y revisión cruzada.

4. Redactar un documento de arquitectura modular

Proceso y Buenas Prácticas

El documento debe incluir:

Visión general: Objetivo y alcance del sistema.

Lista de módulos: Función, entradas, salidas y responsable de cada uno.

Diagrama de arquitectura: Representa gráficamente los módulos y sus relaciones.

Interfaces y protocolos: Detalle de la comunicación entre módulos.

Flujos de datos: Cómo circula la información por el sistema.

Puedes estructurarlo siguiendo las vistas recomendadas: lógica (módulos y relaciones), procesos (concurrencia), implementación (componentes y dependencias), datos (modelo de datos), despliegue (infraestructura).

Checklist

Redactar visión y alcance.

Incluir la lista y descripción de módulos.

Añadir diagramas de arquitectura y flujos.

Documentar interfaces y protocolos de comunicación.

Revisar y actualizar el documento periódicamente.

5. Validar la arquitectura con el equipo

Proceso y Buenas Prácticas

Organiza una reunión de revisión donde cada responsable exponga su módulo.

Analizad en conjunto:

La coherencia de las interfaces.

La claridad de las responsabilidades.

Posibles solapamientos o dependencias innecesarias.

Ajusta la arquitectura según el feedback recibido.

Checklist

Presentar el documento de arquitectura al equipo.

Recoger feedback y propuestas de mejora.

Realizar los ajustes necesarios.

Aprobar la arquitectura final antes de iniciar el desarrollo.

Referencia a mejores prácticas y ejemplos:

La arquitectura modular facilita el desarrollo, la escalabilidad y el mantenimiento, permitiendo que cada módulo evolucione de forma independiente y segura. Documentar y validar estos puntos garantiza un proyecto más robusto y colaborativo