El material que viene en el quiz es el siguiente:

- El principio de Privilegio Mínimo significa asignar solo los permisos necesarios y por el tiempo justo para reducir riesgos de abuso o escalamiento.
    - Principio de Privilegio Mínimo: la cuenta de base de datos usada por la aplicación debe tener permisos estrictamente necesarios (solo lectura, por ejemplo).
- En Argon2id, el parámetro memory=64 MiB obliga al atacante a usar mucha memoria por intento, dificultando ataques con GPU.
    - Argon2id es un algoritmo moderno de hash de contraseñas, diseñado para ser seguro contra ataques de fuerza bruta y cracking con GPU/ASIC. En adicción combina lo mejor de Argon2i (resistente a ataques de canal lateral) y Argon2d (resistente a ataques con GPU).
- La Inyección SQL se evita usando sentencias parametrizadas, que separan los datos del código.
    - **Sentencias Parametrizadas:** separan los datos del código, impidiendo que el texto del usuario se ejecute como comando.
- Sandboxing consiste en ejecutar código en un entorno aislado para que, si algo falla, no afecte al resto del sistema.
    - Ejecutar código o procesos en un entorno aislado y con recursos/permisos limitados, de modo que, si ese código falla o está comprometido, el daño quede contenido y no afecte al sistema global
- Una vulnerabilidad TOCTOU aparece cuando se verifica un recurso y luego se usa sin revalidarlo, dejando tiempo para que un atacante lo modifique.
    - Es una vulnerabilidad de sincronización que ocurre cuando un programa revisa una condición (como permisos de un archivo) y luego, en otro momento, utiliza ese mismo recurso asumiendo que no ha cambiado. Un atacante puede aprovechar ese intervalo de tiempo para reemplazar o modificar el recurso verificado.
- El principio de Mediación Completa indica que cada acceso debe volver a verificarse, sin confiar en permisos anteriores.
    - (no pude encontrar algo más)
- Database Activity Monitoring (DAM) analiza en tiempo real el tráfico SQL para detectar consultas sospechosas.

    - **Database Activity Monitoring (DAM)** es una técnica o herramienta de seguridad que inspecciona, registra y analiza en tiempo real el tráfico SQL (consultas, comandos y respuestas) que circula hacia y desde las bases de datos.
    - DAM observa las consultas SQL que viajan por la red (por ejemplo, de un servidor web al servidor de base de datos).
- En un pipeline CI/CD seguro, si una herramienta como Bandit detecta una vulnerabilidad media o alta, el despliegue se bloquea automáticamente.
    - CI/CD (Continuous Integration / Continuous Deployment) es un conjunto de prácticas y herramientas que automatizan el proceso de construcción, prueba y despliegue de software.
        - Su objetivo es:
            - 1. Detectar errores a nivel preventivo
            - 2. Asegura calidad y seguridad en el código.
            - 3. Entrega actualizaciones periódicas sin interrumpir un servicio en particular.
    - **Ciberseguridad:** el pipeline CI/CD también actúa como una capa de defensa, verificando que el código cumpla estándares de seguridad antes de ser publicado.
    - **Bandit:** Analiza el código fuente buscando vulnerabilidades (e.g. eval(), contraseñas hardcodeadas). La siguiente imagen muestra cómo se podría utilizar bandit con un repositorio en lenguaje Python en GitHub. Esto se realiza antes de ejecutar commit en Git (local de tu ordenador).
    - **Pipeline CI/CD** ejecuta pruebas automáticas antes de aprobar un despliegue.
- Differential Privacy protege datos agregados añadiendo ruido estadístico para impedir identificar personas individuales.
    - **Differential Privacy:** Es una técnica matemática y estadística que permite analizar o compartir información de un conjunto de datos sin revelar información privada de ninguna persona individual. Agrega ruido estadístico para revelar información, pero no información privada.
- La gestión segura de secretos se logra almacenándolos en variables de entorno o servicios como AWS Secrets Manager, nunca dentro del código.

- ¿Qué es Secreto de Software?
  - Un secreto es cualquier dato confidencial que una aplicación necesita para funcionar, como:
    - 1. Contraseñas (de base de datos, APIs, correo, etc.)
    - 2. Claves API (Google, OpenAI, AWS…)
    - 3. Tokens de autenticación
    - 4. Certificados o claves privadas
  - Si estos Secretos quedan expuestos un atacante puede acceder a sistemas críticos o/y robar datos.

- Un Buffer Overflow ocurre cuando un programa escribe más datos en memoria de los permitidos, permitiendo ejecución de código malicioso.
  - Ocurre cuando un programa escribe más datos en una zona de memoria (búfer) de la que tiene asignada. Al sobrepasar los límites, se sobrescriben partes de memoria que podrían contener direcciones de retorno o variables críticas, lo que permite al atacante ejecutar código malicioso. Es un fallo clásico en lenguajes como C y C++, aunque puede simularse en otros entornos.

- En Docker, ejecutar contenedores con --read-only y como usuario no root refuerza el aislamiento y evita daños al sistema.
  - Contenedores (Docker) — aislamiento por namespaces + cgroups:
    - Docker usa:
      - ❖ namespaces (PID, mount, net, user, ipc) → aísla visibilidad de procesos, red y sistema de archivos.
      - ❖ cgroups → limita CPU, memoria, I/O.
      - ❖ capabilities → granulariza privilegios en lugar de ser root absoluto.
      - ❖ seccomp, AppArmor/SELinux → control de syscalls y políticas LSM.