MercadoLibre Backend Challenge

Motivación:

Debido a las necesidades y tecnologías que utilizamos en Meli creemos que hoy en día es necesario tener un conocimiento amplio de todos los componentes que intervienen en el desarrollo de un proyecto. Por este motivo, con este *challenge,* buscamos evaluar conocimientos sobre desarrollo (buenas prácticas, diseño modular, etc.) y conocimiento de arquitecturas web modernas, como APIs REST y microservicios. Por otro lado, el conocimiento de seguridad o la capacidad de adquirir conocimientos relacionados al campo.

Objetivo:

El equipo de seguridad de MercadoLibre necesita realizar escaneos automatizados de seguridad a miles de aplicaciones que hoy soportan a los servicios del sitio. Hay dos tipos de escaneos básicos DAST y SAST en esta oportunidad necesitamos realizar escaneos DAST.

El objetivo concreto es desarrollar una API REST que se comunique con una instancia de Zaproxy (suministrado en el docker-compose) y que tenga soporte a las siguientes acciones:

- Realizar un escaneo de vulnerabilidades sobre un set de urls usando Active Scan de Zaproxy.
- Poder consultar el estado de un escaneo y los resultados una vez terminado el mismo.

Consideraciones:

En los datos que devuelve la API construida tienen que persistir tanto los escaneos realizados como sus resultados más allá de lo que puede guardar Zaproxy. Esto es porque un desarrollo completo del proyecto de DAST tiene que ser independiente del motor de escaneo que utilice.
Por eso se valorará que el diseño sea independiente de Zaproxy y sea fácil de extender para usar otras tools.
También es parte del desafío el diseño de la API REST (los recursos, paths y métodos utilizados para cada acción).

Requerimientos técnicos:

- Utilizar una base de datos para persistencia.
- Desarrollar la API en Python, Ruby, Go, Java, C# o NodeJS.
- Utilizar docker-compose para levantar toda la app, incluyendo base de datos a utilizar y la herramienta Zaproxy.

Entregables:

- Código funcionando en repositorio privado de GitHub.
- Documentación mínima para correr el proyecto.
- Explicación del diseño y las decisiones tomadas.

Extras:

- Mecanismo para realizar escaneos programados.
- Diseño adaptable a distintos scanners, ejemplo Burp o Acunetix.
- Test unitarios y de integración.
- Propuesta de diseño para una aplicación escalable que pueda realizar N escaneos en paralelo, teniendo en cuenta cuestiones de performance y resiliencia.

Ayuda:

Se provee un docker-compose para levantar una base de datos (MySQL, pero puede cambiarse por otro de tu preferencia, relacional o no), Zaproxy y una app vulnerable (Juice Shop).