SecureBox - Documento ERS

1. Introducción

SecureBox es un sistema de almacenamiento seguro de secretos diseñado para garantizar la confidencialidad, integridad y disponibilidad de los datos. El sistema permite gestionar múltiples contenedores de secretos, protegiéndolos mediante cifrado AES y autenticación robusta. Además, ofrece funcionalidad de respaldo en la nube (Google Drive) para recuperación ante pérdidas.

2. Requisitos Funcionales

2.1. Gestión de Autenticación

[RF-1] El sistema permite la autenticación del usuario mediante contraseña, utilizando la función *scrypt* para derivar una clave segura a partir de la contraseña y un *salt* generado de forma segura con *os.urandom*.

[RF-2] El sistema implementa un mecanismo de retraso exponencial ante múltiples intentos fallidos de autenticación, para mitigar ataques de fuerza bruta.

[RF-3] El sistema almacena de forma segura la clave derivada y el *salt* en un archivo con permisos restringidos.

2.2. Gestión del Vault

[RF-4] El sistema permite la creación, edición y eliminación de contenedores, donde cada contenedor agrupa un conjunto de secretos.

[RF-5] El vault se cifra utilizando AES-256 en modo CBC. Para ello, se usa una DEK (Data Encryption Key) que se deriva y protege mediante una KEK (Key Encryption Key) obtenida a partir de la contraseña del usuario.

[RF-6] El sistema verifica la integridad del vault utilizando HMAC-SHA256 durante las operaciones de carga y descarga.

[RF-7] Los datos del vault persisten en un archivo local que permanece cifrado en disco en todo momento.

2.3. Integración con la Nube

[RF-8] El sistema soporta la autenticación **mediante OAuth2** para interactuar con Google Drive, gestionando de forma adecuada los tokens de acceso.

[RF-9] El sistema permite la subida del archivo cifrado del vault a Google Drive, almacenándolo en una carpeta dedicada.

[RF-10] El sistema permite la descarga del vault desde Google Drive, facilitando la recuperación de datos en caso de pérdida de la copia local.

2.4. Interfaz de Usuario

[RF-11] El sistema proporciona una interfaz (CLI) interactiva con menús jerárquicos.

[RF-12] La visualización de contenedores y su contenido se presentan en un formato claro y legible, mostrando información relevante como el ID, nombre y los secretos descifrados.

3. Requisitos No Funcionales

3.1. Seguridad

[RNF-1] El sistema debe ser resistente a ataques offline mediante el uso de *scrypt* con parámetros configurados (N=2^14, r=8, p=1).

[RNF-2] Los secretos deben estar protegidos mediante un cifrado doble: la DEK se protege con una KEK derivada de la contraseña del usuario.

[RNF-3] La integridad del vault debe ser comprobada en cada operación (carga y descarga) mediante HMAC-SHA256.

[RNF-4] Los archivos locales que almacenan datos sensibles deben contar con permisos restringidos (0o600).

3.2. Calidad del Código y Mantenimiento

[RNF-5] El diseño debe seguir un enfoque orientado a objetos con una clara separación de responsabilidades entre módulos.

[RNF-6] Todo el código debe estar documentado internamente, con comentarios de cabecera en cada función y clase.

3.3. Disponibilidad y Recuperación

[RNF-7] El sistema debe permitir la realización de copias de seguridad manuales en Google Drive, asegurando la disponibilidad y recuperación de datos en caso de pérdida de la copia local.z

[RNF-8] Se debe incluir un mecanismo para detectar y notificar posibles inconsistencias o modificaciones no autorizadas en el vault.