Research

Adrian Rybaczuk

29 April 2025

1 Cloud Storage

1.1 Do czego użyjemy

C loud Storage chcemy użyć do przechowywania haseł użytkownika. Użytkownik powinien wybrać docelowy provider, którego chce użyć do przechowywania haseł. Po wybraniu providera, powinien podać dane dostępowe do danego providera. Np. w wypadku S3 powinien podać nazwę bucketa, oraz swoje ID i klucz dostępu. Następnie powinien podać login i hasło dostępu do danych. Na podstawie którego zostanie zapisany na urządzeniu lokalnym plik konfiguracyjny dostępu do danego providera. Program powinien być napisany w sposób który pozwoli dostęp do więcej niż jednego providera lub pozwoli na łatwą implementację nowych providerów.

Aktualnie dostępne providery na rynku to:

- S3
- Google Cloud
- Azure Blob Storage

Źródło możemy tu znaleźć inne płatne opcje

Opcje open source:

- minio
- Storj
- SwiftStack

Źródło

2 Bezpieczeństwo danych

2.1 Bezpieczeństwo dostępu do danych

Poza bezpiecznym przechowywaniem danych dostępowych do providerów, bezpieczeństwo przechowywanych danych
zależy od zewnętrznych providerów.

2.2 Web Crypto API

Dane: `https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API` Do projektu zamierzam uzyc Web Crypto API do szyfrowania danych.

3 Platformy docelowe

Platformy są wstępnie posortowane po priorytetach.

3.1 Web Addon

3.2 Desktop

3.3 Android

3.4 iOS

4 Cechy

4.1 Prostota

4.2 Bezpieczeństwo

4.3 Przenośność

5 Podobne projekty

5.1 Mopass

link: https://phodal.github.io/mopass/

5.2 Amazon KMS and DynamoDB

Case: `https://towardsaws.com/diy-serverless-password-manager-using-aws-kms-and-amazon`
Problem z tym jest taki że zyskujemy 2 warstwy pomiędzy aws lambda oraz
kms (w tym możemy użyć s3)

6 Funkcjonalności

1. Dodawanie haseł
2. Generowanie haseł
3. Przeglądanie haseł
4. Edytowanie haseł
5. Usuwanie haseł
6. Importowanie haseł
   Przyjmuje plik csv z hasłami
7. Eksportowanie haseł
   Exportuje hasla do pliku csv po wybraniu providera, oraz zakresu haseł i wpisania hasla dostepu
8. Synchronizacja haseł
   Synchronizacja odbywa się automatycznie, wymaga dostępu do internetu, oraz włączonej funkcji synchronizacji w ustawieniach.
9. Wyszukiwanie haseł
10. Multifactor authentication Oprocz hasla do repozytorium, bedzie mozliwosc uzycia drugiego czynnika uwierzytelniania TOTP/HOTP
11. Przenośność Przy pomocy kodu QR z zaszyfrowanymi danymi dostępu do repozytorium, mozna przeniesc repozytorium na inne urzadzenie