Cryptographie Appliquée Avancée

Lab 2 - Password manager

Département TIC - orientation TS

Professeur : Alexandre Duc

Assitante : Lucie Steiner

Auteur : Dejvid Muaremi

30 avril 2019

1 Introduction

Le but de ce laboratoire est de créer un gestionnaire de mot de passe cryptographiquement sûr. Ce gestionnaire est sûr tant que le mot du master password l'est aussi. Le gestionnaire peut être sous l'un des trois états suivants :

1. **Not Running** Son état par défaut avant d'être lancé.
2. **Unlocked** Son état lorsque l'utilisateur entrera le mot du master password. Une fois dans cet état, l'utilisateur peut récupérer tout ses mots de passes stocké sans avoir besoin de taper le mot du master password.
3. **Locked** L'utilisateur peut verrouillé le gestionnaire qui va libérer la mémoire et attendre une nouvelle saisie de mot de passe.

2 Modélisation du programme

Stockage des mots de passes

1. **master.txt** Contient le hash du master password ainsi que le salt de la clé secrète.
2. **sites.txt** Contient, en clair, les noms des sites dont les mots de passe sont stocké.
3. **passwords.txt** Contient les mots de passes chiffré avec la clé secrète.
4. **nonces.txt** Contient, en clair, les nonces des mots de passes.

Gestion du mot du master password Le master password, entré par l'utilisateur, est haché avec argon2. Ensuite, une clé secrète sera générée en utilisant un salt aléatoire puis le tout sera stocké dans le fichier correspondant.

Chiffrement des mots de passes Chaque mot de passe entré par l'utilisateur est chiffré puis stocké avec son nonce dans le fichier correspondant.

Changement de mot du master password Une fois que l'utilisateur aura saisi le nouveau mot de passe, des fichiers temporaire seront créé afin de contenir les nouveaux hash, et chiffrés. Les mots de passes seront lu, déchiffré, chiffré puis stocké dans le nouveau fichier.
En partant de l'idée que l'utilisateur ne changera le master password uniquement lorsque celui-ci aura fuité, il est donc nécessaire d'empêcher celui qui a volé le master password de déchiffré tout les mots de passes. Il y a une solution qui consiste à ne pas

Mémoire protégée Tout mot de passe saisi ou affiché en clair, devra être protégé en étant stocké dans un espace mémoire alloué par void *sodium_malloc(size_t size); une fois que ces valeurs ne sont plus utilisé, il faut absolument les libérer avec void sodium_free(void *ptr);, d'après la documentations de libsodium memory management, ces fonctions permettent d'avoir un fragment mémoire qui sera vidé à la fin du programme.

2.1 Algorithme utilisé

Génération des nombres aléatoire Selon la documentations de libsodium generating random data la fonction void randombytes_buf(void * const buf, const size_t size); permet de remplir un tableau avec des bits aléatoire uniformément distribué.

Hash du master password Argon2 met à disposition deux fonctions permettant chacune de hacher des mots de passe et de les vérifier avec crypto_pwhash_str() et crypto_pwhash_str_verify() respectivement. Les paramètres de mémoire et d'opération ont été réglé à **SENSITIVE**.

Dérivation de la clé Argon2 permet également de générer des clé de chiffrement avec la méthode crypto_pwhash, j'utilise l'algorithme par défaut qui lui se base sur l'algorithme recommandé par libsodium. Les paramètres de mémoire et d'opération ont été réglé à **MODERATE**.

Chiffrement des mots de passe Pour finir, les mots de passes sont chiffré de manière authentifiée en utilisant les fonctions crypto_secretbox_easy() et crypto_secretbox_open_easy() de libsodium secretbox. Les algorithme utilisé sont **XSalsa20 stream cipher** pour le chiffrement et **Poly1305 MAC** pour l'authentification.

3 Manuel d'utilisateur

Compiler le programme g++ main.cpp base64.cpp -lsodium -o passwordManager

Lancer l'application ./passwordManager.

L'état Locked Une fois l'application lancée, celle-ci est dans l'état **Locked**, si les fichiers de mots de passe existent, il sera demandé à l'utilisateur de saisir le mot du master password sinon, il lui sera demandé d'en choisir un nouveau. Ensuite, l'application passe dans l'état **Unlocked**.

L'état Unlocked L'utilisateur a accès un menu qui lui demande ce qu'il souhaite faire. Il a la possibilité d'ajouter des mots de passe, afficher des mots de passe, changer le mot du master password, verrouillé l'application ou la quitter. Ces choix se font en utilisant le pavé numérique du clavier.

Ajout de mot de passe Il est demandé à l'utilisateur de saisir, dans un premier temps le site, puis le mot de passe. Une fois ceci fait le mot de passe sera chiffré

Afficher un mot de passe Un menu s'affiche, listant la totalité des site, chaque site est lié à un numéro, idéalement une pagination sera mise en place pour ne pas surchargé le terminal. L'utilisateur saisi ensuite le numéro correspondant au site dont il veut lire le mot de passe. Le programme va ensuite déchiffré le mot de passe en question et l'afficher dans le terminal.
Attention aux personnes se trouvant derrière vous à ce moment

Changement de mot de passe Une fois l'identité de l'utilisateur vérifiée, il lui sera demandé de choisir son nouveau mot du master password. À ce moment, il faudra patienter le temps que l'application déchiffre avec l'ancienne clé tout les mots de passe et les chiffre avec la nouvelle clé. ceci peut prendre un certain temps.

Verrouiller ou quitter lorsque l'utilisateur décide de Verrouiller ou quitter l'application toute la mémoire est nettoyée, et libérée.