DEV – Rapport du projet

Bousard Alexandre

Oger Baptiste

Binôme 69

Développement

HENALLUX, IR

Deuxième année

Groupe E

Année académique 2023-2024

1 Table des matières

2 Introduction

Dans le cadre de notre cours de développement, les professeurs nous ont donnés pour consigne de réaliser un projet en binôme.

En effet, nous devons développer un hash breaker en C. C'est un outil ou un programme conçu pour décrypter ou casser des fonctions de hachage. Les fonctions de hachage sont des algorithmes qui transforment des données en une chaîne de caractères unique et apparemment aléatoire, appelée "hash". Ces hash sont souvent utilisés pour stocker de manière sécurisée des mots de passe ou vérifier l'intégrité des données.

La valeur d'un hash breaker réside dans sa capacité à compromettre la sécurité de ces systèmes en inversant le processus de hachage, révélant ainsi le contenu original des données. Cela peut être utilisé de manière éthique, par exemple, pour tester la robustesse des systèmes de sécurité, mais peut aussi être exploité de manière malveillante pour accéder à des informations sensibles.

En somme, un hash breaker souligne l'importance de choisir des algorithmes de hachage robustes et de mettre en place des mesures de sécurité supplémentaires pour protéger les données sensibles.

3 Structure du projet

```
+-- LICENSE.md
+-- Makefile
+-- README.md
+-- client
|   \-- client.c
+-- docs
|   \-- Ennonce.pdf
|   \-- Rapport.pdf
+-- lib
|   +-- comlib.c
|   +-- comlib.h
|   +-- hashlib.c
|   +-- hashlib.h
|   +-- loglib.c
|   +-- loglib.h
|   +-- stringslib.c
|   \-- stringslib.h
\-- server
    \-- server.c
```

Comme vous pouvez le voir, nous avons décidé de séparer notre code en plusieurs fichiers afin de le rendre plus agréable à lire.

Toutes les parties communes au serveur et au client sont dans les fichiers « comlib ». Les fonctions liées à la gestion des hash sont regroupées dans les fichiers « hashlib »

Et enfin, toutes celles en rapport avec la génération des chaines de caractères sont situées dans les fichiers « stringslib »

4 Compilation

Afin de faciliter la compilation de notre projet, nous avons décidé d'utiliser un makefile. Celui-ci comporte plusieurs commandes :

- All : compile le serveur et le client
- Rebuild : supprimes les anciens fichiers binaires et compile le client et le serveur
- Server : compile le serveur
- Client : compile le client
- Clean : supprime les fichiers binaires

5 Utilisation

Une fois les fichiers compilés, nous pouvons à présent utiliser le hash breaker.
Il est impératif de lancer le serveur avant le client !

Pour exécuter le serveur, il faut préciser plusieurs arguments :

- -i : spécifie l'IP du serveur
- -p : spécifie le port du serveur
- -h : spécifie le chemin d'accès de la liste de hash

Une fois ce-dernier lancé, vous pouvez exécuter le client en précisant ces deux paramètres :

- -i : spécifie l'IP du serveur
- -p : spécifie le port du serveur

Le hash breaker est capable de traiter les hash dans ces formats :

- MD5
- SHA1
- SHA256
- SHA512

Il ne faut pas préciser de quel type il s'agit, le programme le reconnait tout seul !

Une fois que tous les hash présents dans la liste ont été crackés, le serveur se ferme automatiquement. Il enregistre dans un fichier « result » les mots de passe et leur hash correspondant.

6 GitHub

Nous avons décidé de publier notre projet sur GitHub pour plusieurs raisons :

- Facilite le travail de groupe
- Permet de partager le projet avec la communauté (open-source)
- Apport de nouvelles nouveautés

Sur GitHub, les branches permettent de travailler sur différentes versions du projet sans altérer la version principale. Elles isolent les modifications, facilitent le travail collaboratif en permettant à plusieurs personnes de contribuer simultanément, et offrent un espace pour tester les changements avant de les fusionner avec la version principale.

Nous avons décidé de mettre notre projet sous licence MIT. Elle est souvent choisie pour sa simplicité et sa permissivité. Elle permet une utilisation, modification et distribution libre du logiciel tout en imposant peu de restrictions, favorisant ainsi l'intégration dans d'autres projets et l'utilisation commerciale, avec une responsabilité limitée pour le créateur du logiciel.

De ce fait, quiconque voulant contribuer au projet en a le droit !

Le projet est disponible ici : https://github.com/0xb4b0u/hashbreaker