# Sécurité informatique INF26307

# Travail Pratique #2

## SESSION HIVER 2023

| | |
|---|---|
| Date limite de remise du TP | 14 mars 2023 à 19h00 |
| Équipe | Individuel ou en équipe de 2 ou 3 étudiants. |
| Pondération | 15% |

## Mise en contexte

Comme nous avons vu lors du cours d'introduction, il existe des sites qui qualifient la robustesse d'un mot de passe. Ce TP vous permettra de casser des mots de passe encryptés avec l'algorithme MD5. Cela pourrait s'avérer utile si un jour vous vous retrouvez sur une ile déserte et que vous deviez trouver le mot de passe du compte « root » de la machine à téléporter.

## Développement à faire

Le présent travail pratique se segmente en deux parties, soit la première qui consiste à réaliser une application générant un dictionnaire de mots de passe à partir de paramètres passés par l'utilisateur afin de faire une attaque par brute force.

La deuxième partie consiste à créer une application permettant de tester des hachages MD5 qui sont remis par l'enseignant afin de trouver le mot de passe caché dans les hachages à partir de deux dictionnaires de mot de passe :

- Celui créé par vous sur votre première application;
- Celui disponibilisé par l'enseignant;

**Application - Dictionnaire :**

Cette application devra générer un dictionnaire de mot de passe en fonction des paramètres qui sont saisis dans l'interface de départ par l'utilisateur. L'interface au départ de l'application est simple, elle doit demander 4 informations avant de générer le dictionnaire :

- La longueur minimale des mots de passe du dictionnaire (ex : 3)
- La longueur maximale des mots de passe du dictionnaire (ex : 8)
- Les caractères permis dans le dictionnaire (Ex : abcdefghijklmnopqrstuvwxyz1234567890)
    - Cette fonction peut se faire à partir d'un choix à cocher par l'utilisateur (alphabet minuscule, alphabet majuscule, chiffre 0 à 9, caractères spéciaux avec des choix individuels : # $ % ? & * etc.) ou par un champ de saisie textuel;
- Le dossier et le nom de fichier de sortie dans lequel seront déposés les mots de passe (ex : c:\dico.txt).
    - Pour cet item, il vous est possible d'utiliser les modules déjà en place dans votre environnement de développement;
    - Le dictionnaire doit être peuplé à raison d'un mot par ligne;

Ainsi, votre code doit générer l'ensemble des combinaisons de mot de passe possible selon les paramètres qui lui sont introduits.

Voici un exemple :

- La longueur minimale des mots de passe du dictionnaire : **1**
- La longueur maximale des mots de passe du dictionnaire : 3
- Les caractères permis dans le dictionnaire : **abc**
- Le dossier et fichier de sortie : **c:\pass.txt**

Nous devrions retrouver les 39 éléments suivants à raison d'un item par ligne dans le fichier ***pass.txt*** à la racine du répertoire **c:** :

**a b c aa ab ac ba bb bc ca cb cc aaa aab aac aba abb abc aca acb acc**
**baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc**

Vous comprendrez que mon exemple est très simple, mais que cela peut devenir d'autant plus complexe de générer des mots de passe lorsque l'on inclut l'ensemble des lettres de l'alphabet (majuscule et minuscule) de même que les 10 chiffres de base et des caractères spéciaux.

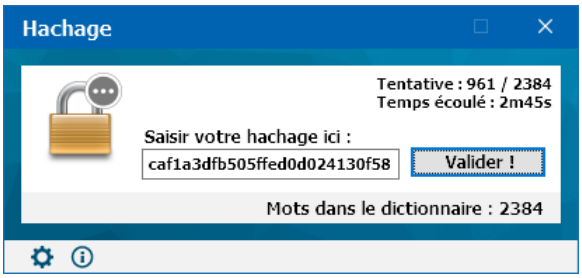
**Application – Hachage :**

Cette application devra demander à l'utilisateur de saisir un hachage à trouver. Elle ira récupérer le contenu du dictionnaire que vous devrez sélectionner à partir d'un fichier. Pour chacune des entrées présentes dans le dictionnaire, elle devra générer un hachage MD5 du mot contenu dans le dictionnaire et le comparer au hachage saisi par l'utilisateur au départ de l'application. Si la correspondance fonctionne, elle devra aviser l'utilisateur du mot de passe correspondant au hash trouvé. Si la correspondance ne fonctionne pas, elle devra incrémenter de 1 un compteur affiché à l'utilisateur et tenter refaire le même processus de hachage/comparaison avec le prochain mot du dictionnaire passe jusqu'à temps d'avoir passé à travers l'ensemble des mots de passe présent dans le dictionnaire.

L'application doit indiquer les détails suivants dans son interface :

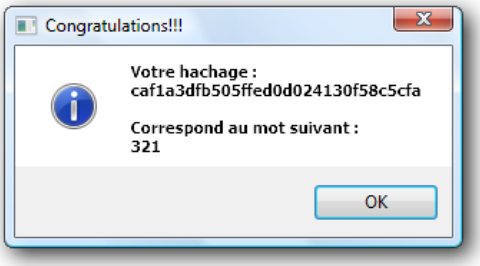
- Quantité de mots dans le dictionnaire
- Le nombre de tentatives réalisées depuis de lancement de la validation
- Le temps cumulé depuis le lancement de la validation

Voici un exemple d'interface graphique en guise d'exemple :




Dans cet exemple, la roue dentelée dans le coin inférieur gauche de l'interface permet de sélectionner le fichier de dictionnaire et d'ajuster d'autres paramètres pour l'application. La mise à jour des tentatives et le temps écoulé se mettent à jour en temps réel après chaque tentative.

Lorsque le hachage est trouvé, une fenêtre apparait mentionnant la correspondance trouvée :



## Hachages à solutionner

Vous devez trouver les mots de passe cachés dans les hachages suivants :

| Hachage | Longueur | Type de caractères |
|---|---|---|
| 937557fac5cffc250ccf72031474078b | 6 | a b G M N O 0 1 2 |
| 2414766fb5121dfebdf220d4b8a550a0 | 6 | a b M N r s 0 1 2 |
| 973d2d342378f637aeeb9ec96f5a4b46 | 6 | a b A B m n 0 1 2 |
| 6e63c6bf94b66e04ff2f48be546f0110 | 10 | a b |
| 90b2360c704e0a805cee6dd6fd71eeaa | 10 | x y z |
| 154e5cae63af27012426896ab9da2ac0 | * | * |
| 139230ff1ef21bb767bb9b475cf99873 | | |
| 7746fcc0b507887866f8bd227b37122c | | |
| db50a77debe1141e236bfefb468440ab | | |

* Utilisez le dictionnaire de mots français fournis dans le TP.

## Conditions de réalisation

Voici les conditions de réalisation dans lesquelles vous devez produire votre travail et vos deux applications :

- Langage de programmation : Libre! Vous pouvez prendre celui de votre choix. Cependant, l'application exécutable (Dictionnaire et Hachage) doit fonctionner dans Windows sans plug-in ou framework supplémentaire;
- Algorithme : Vous pouvez reprendre les librairies de code présenté dans le RFC-1321 donné en référence pour la conception de l'application Hachage, ou encore utiliser les classes de fonction probablement déjà disponibles dans votre environnement de développement;
- Idéalement, les applications « Dictionnaire » et « Hachage » doivent être portables dans Windows sans nécessiter l'installation d'application, plug-in ou encore de framework particulier;
- Vous devez faire preuve d'une grande autonomie dans le développement de vos applications. Il existe une multitude d'informations à votre portée sur Internet pour vous aider et vous devez en faire usage.

## Livrables pour l'évaluation du travail pratique

Pour que votre travail pratique puisse être évalué par l'enseignant, vous devez déposer les 4 éléments suivants :

1. Un rapport écrit d'un **maximum de 10 pages (sans compter les annexes)** avec les sections suivantes :
    - Page de présentation identifiant le nom des étudiants, le titre du TP et le sigle du cours;
    - Présentation de votre implémentation de l'algorithme MD5 :
        - Expliquez comment vous avez implémenté l'algorithme MD5 dans votre application;
    - Présentation de votre application Dictionnaire;
    - Présentation de votre application Hachage;
    - Solution aux hachages :
        - Identification des mots de passe solutionnés avec votre application;
    - Analyse de vos résultats :
        - Site et/ou sources qui vous ont aidé;
        - Problèmes rencontrés, erreurs, bons coups, etc.
    - Conclusion;
    - Annexes :
        - Capture d'écran des deux applications;
        - Références / Bibliographie;
2. Le code source de votre application client;
3. Le code source de votre application serveur;
4. Un fichier ZIP contenant les applications Client et Serveur exécutables sur Windows (version portable ne nécessitant pas l'installation de Framework ou de Plug-In);

## Obligations à respecter

- Vos deux applications doivent fonctionner pour être évaluées.
- Tout comme bon programmeur, votre code doit être bien documenté.
- Une grande partie de votre code pour le hachage est inspiré du RFC ou encore de classes. Si d'autres bouts de code sont inspirés de sources externes (internet, etc.), vous devez documenter vos sources dans l'annexe (références) et faire mention de la portion inspirée.

## Barème de correction

Le barème suivant sera respecté pour l'attribution des points sur le travail pratique (total 15 points).

- Applications dictionnaire et hachage fonctionnelles sur mon poste Windows 11 (SandBox) : **5 points**
- Rapport complet, clair et bien étoffé : **5 points**
- Identification des hachages : **5 points**

| **!! Note importante !!** |
| --- |
| Vous obtiendrez un échec à ce TP si vos applications contiennent un virus, trojan, malware, ransomware ou autres trucs du genre! 😉 |

**Bonne chance!**