

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

Sécurité Informatique Devoir 1 — Énoncé

SIGLE : INF36207
TITRE : Sécurité Informatique
GROUPE : 06
PROFESSEUR : Steven Pigeon
K-212
steven_pigeon@uqar.ca

DATE DE REMISE : 12 Février 2019, avant minuit

— 1 —

0. Modalités. Vous devez faire le devoir en équipes de deux. Le devoir devra être remis par courriel, lequel contiendra une archive comprenant un document texte (.docx ou autre) qui contient les réponses (discussions, numéro bonus et mots de passes récupérés) et les programmes que vous devrez réaliser au n° 2. Les programmes devront être fonctionnels, évidemment, et l'archive devra contenir tout ce qui est nécessaire pour les faire fonctionner (on doit pouvoir les compiler et les exécuter). Le nom de l'archive doit contenir le nom des deux coéquipiers et le numéro du formulaire : 1 . Les versions électroniques (pdf) seront aussi sur Moodle.

Notez que chaque devoir est *unique*. Le numéro 1 ne peut être utilisé que par une seule équipe. Si deux équipes utilisent le même formulaire, les deux seront disqualifiées — un euphémisme pour « auront zéro ».

1. Casser des mots de passe. 15 pts. Supposez que vous ayez capturé le fichier de mots de passe d'un système dont vous tentez de prendre le contrôle. Vous voulez obtenir les mots de passe originaux : vous devez monter une attaque contre ces mots de passe qui sont camouflés grâce à l'algorithme MD5. Les *hashes* que vous avez capturés sont les suivants :

```
6f31ce91f67bf9246ea237850686cc7a
4c85f8da3fac9521dfba156c34564e41
06861a5d5ca1a2e490daa0d64cc1a0b7
cc35f894c9219239ebb858692a86be1b
1f7cf588b860862a2c5ad696462523b8
ab4f6b1c2b3a66b5bf316e03bbd43e3f
cb8d48705c534ae480bda3efd5619571
d7d7c6ad7c0147cb380f28566296efc3
711253e85ce0dbc86cdabb27ec9285c8d
ce44a7233d15bf399a09708e86039511
8650659a907283a65f164327feda276c
badb37f791ded4a9928c86d68b931311
8a2663e9748ae78a256152dc7f52f02c
e8394805e934ce2e004622b9810159c2
02ea8c79deeb6436c40b8e50d45b5ee5
```

Pour ce qui suit, vous avez le choix du langage de programmation : Java, C, C++, C#, Python, Windows-Power Shell, Bash, etc. Cependant, pour le numéro **1 b)**, considérez un langage capable de générer du code *rapide*. De plus, les bibliothèques qui calculent le hash MD5 sont disponibles sur toutes les plateformes et il en existe des *bindings* pour la plupart des langages, vous n'aurez donc pas à l'implémenter vous même. Quelque soit le langage que vous aurez choisi, vous devez remettre un projet complet qui montre que le code fonctionne — *pas de code, pas de points*.

a) Attaques par dictionnaire. 3 pts. Utilisez le dictionnaire `mots-8-et-moins.txt` (que vous trouverez sur Moodle) pour monter une attaque par dictionnaire contre les *hashes* ci-dessus. Donnez les mots de passe en clair récupérés.

b) Attaques par énumération. 10 pts. Vous avez remarqué que certains des *hashes* ne sont pas résolus par l'attaque par dictionnaire. Dans ce cas, vous devrez monter une attaque par énumération où vous allez générer toutes les chaînes de longueur 1, toutes les chaînes de longueur 2, toutes les chaînes de longueur 3, etc. jusqu'à ce que vous trouviez une chaîne dont le *hash* corresponde. Puisque le but n'est pas de vous faire passer de longues heures de temps de calcul, supposez que les mots de passe n'ont pas plus de 8 caractères de long, et que les caractères sont tirés de l'alphabet suivant :

```
abcdefghijklmnopqrstuvwxyz0123456789!@#%&*
```

Donnez les mots de passe récupérés en clair.

2. Sur la complexité de casser les mots de passe. 2 pts. À la lumière du n° 2, qu'est-ce que vous pouvez dire sur le choix d'un mot de passe? Qu'est-ce que vous suggéreriez pour déjouer à la fois les attaques par dictionnaire et les attaques par énumération? Quelle stratégie *simple* utiliseriez vous pour vous choisir un mot de passe sécuritaire (sans pour autant être impossible à taper)?