Travail pratique #01 - Crack'n patch

Objectifs

1. Concevoir une application qui fait une validation sécuritaire
2. Utiliser un déboggeur bas niveau pour suivre la logique d'un programme
3. Acquérir une compréhension minimale de l'assembleur x86
4. Être en mesure de tester soi-même la sécurité d'une application

Instructions

Ce travail se divise en 2 parties

Partie 1 : Création d'un programme de validation sécuritaire

Vous devez créer un programme de type console en C++ qui valide qu'un mot de passe de 12 caractère entré au clavier est valide.

1. Votre programme doit demander d'entrer un mot de passe
2. Une fois le mot de passe entré, vous devez afficher un message indiquant si le mot de passe est valide ou non
3. Un seul mot de passe est valide, et il doit toujours être le même
4. Il **ne doit pas** être possible de retrouver votre mot de passe en clair dans votre exécutable
5. Votre mot de passe ne doit contenir que des caractères affichable et saisissable au clavier
6. Efforcez-vous de rendre le code de validation de mot de passe difficile à comprendre même une fois désassemblé.
7. La validation de votre mot de passe doit se faire localement seulement et ne doit pas tenter d'accéder à un service réseau distant
8. Vous pouvez rechercher et utiliser diverses techniques que vous jugez pertinentes pour empêcher que quelqu'un trouve votre mot de passe (hint : exe packer, anti-debugger, encryption, etc)

Vous devez expliquer en détail dans le haut de votre fichier de code source principal (celui contenant votre *main*) pourquoi votre code est sécuritaire, les techniques que vous avez employées, ainsi que votre mot de passe (pour que je puisse tester).

IMPORTANT : Après la remise, votre exécutable sera mit à l'épreuve par l'ensemble de la classe, dans une compétition amicale ou chacun essaiera de briser la sécurité des autres. Il est fortement suggéré de

tenter de *cracker* votre propre programme pour vous assurer qu'il n'a pas de lacunes. Tous les programmes qui respectent les règles de ce TP seront compilés en mode *Release*. Si vous utilisez des techniques spéciales (pre ou post build-steps, etc), assurez-vous que celles-ci s'appliquent aussi en release.

Pour cette partie, vous devez me remettre votre solution complète **une fois nettoyée**. Chacuns des points demandés font partie de la grille de correction et vous serez évalués sur ceux-ci.

Partie 2 : Création d'un *patcher*

Veuillez télécharger l'exécutable distribué avec cet énoncé et créez un programme *patcher* capable de modifier le comportement de celui-ci pour qu'il affiche toujours que le mot de passe est bon peu importe quel mot de passe est entré.

Votre patcher doit modifier le programme fourni.

Pour cette partie, vous devez me remettre votre solution complète **une fois nettoyée**.

Remise

À remettre sur vortex : vos **deux** solutions visual studio **une fois nettoyées**. Veuillez indiquer clairement dans le nom de chacune des 2 remise de quelle partie il s'agit (1 ou 2).