Password Cracker Factory

Implémentation de techniques de cassage de mots de passe via le patron de conception « Fabrique ».

Objectifs pédagogiques

À l'issue du projet, l'étudiant devra être capable de :

- Implémenter le **Pattern Fabrique** (Factory Method et/ou Abstract Factory).
- Comprendre et simuler deux grandes techniques de cassage de mot de passe (Brute Force et Dictionnaire).
- Concevoir un système modulaire et évolutif.
- Faire la distinction entre une **attaque locale** et une **attaque en ligne**.
- Créer des **clients cibles d'authentification** en local (console) et en ligne (formulaire PHP).
- Présenter de manière claire un projet technique à l'oral.

Livrables attendus

1. Code source complet, structuré, bien commenté (hébergé sur GitHub)
2. Le fichier *Readme* du dépôt GitHub de votre projet devra contenir un rapport succinct contenant :
    - Une description de l'architecture logicielle (diagramme de classes UML)
    - Le choix des patrons de conception utilisés et leur justification.
    - Une explication des variantes implémentées.
    - D'éventuelles pistes d'amélioration.
3. Une vidéo de démonstration et explication technique (max 10 minutes), contenant :
    - Présentation rapide de la structure globale du code
    - Présentation rapide du fonctionnement général
    - Vue d'ensemble de la structure du projet
    - Démonstration concrète des attaques (les 4 variantes)
    - Utilisation du pattern Fabrique

Détails techniques

1. Structure générale du projet

Les étudiants devront concevoir un outil modulaire qui permet de choisir dynamiquement la **méthode d'attaque** (brute force ou dictionnaire) ainsi que **le type de cible** (locale ou en ligne).

Exemple d'usage :

java CrackerApp --type dictionnary --target local --login admin

2. Attaques à implémenter

Pour toute attaque, on supposera que le login est déjà connu.

a. Attaque par Brute Force

- ✓ Générer automatiquement toutes les combinaisons possibles de caractères selon un alphabet (a-z, 0-9, etc.).
- ✓ Support configurable de la longueur maximale des mots de passe.
- ✓ Vérifie chaque mot de passe sur la cible d'authentification.

b. Attaque par Dictionnaire

- ✓ Charger un fichier texte contenant une liste de mots de passe potentiels.
- ✓ Les tester un à un contre la cible.

3. Types de cibles

a. Cible locale

- ✓ Programme Java simple qui contient un login + mot de passe définis en dur.
- ✓ Se lance via la console
- ✓ Prend les paramètres de connexion sous forme d'arguments en ligne de commande
- ✓ Réagit à l'entrée utilisateur avec une notification de succès ou d'échec de la connexion

Exemple d'usage : java LocalAuthenticator admin passer1234

b. Cible en ligne

- ✓ Mini-site PHP avec un formulaire permettant de renseigner un login et un password.
- ✓ Valide les entrées et affiche "Connexion réussie" ou "Échec de la connexion".
- ✓ L'attaque en ligne consiste à envoyer des requêtes HTTP vers la page cible. Le statut de la réponse permettra de vérifier si le mot de passe a été trouvé ou non.

4. Usage d'une Fabrique

Les étudiants doivent implémenter un **patron de type Fabrique**, pour gérer dynamiquement les combinaisons.

Chaque **fabrique concrète** crée une combinaison spécifique de stratégie d'attaque + type de cible.