

Générateur de mot de passe : Un générateur de mot de passe est un outil qui génère des mots de passe complexes et aléatoires pour améliorer la sécurité des comptes en ligne. Ces générateurs peuvent souvent inclure des options telles que la longueur du mot de passe, l'inclusion de caractères spéciaux, de chiffres, de lettres majuscules et minuscules, ainsi que des options pour éviter des mots de passe facilement devinables.

Fonctionnement

- La fonction `generer_mot_recuratif(x, prefixe)` utilise une récursion pour créer toutes les combinaisons possibles.
- L'utilisateur saisit une taille, et le programme retourne toutes les combinaisons.

Exemple

Pour une taille de 2 : ['aa', 'ab', 'ac', 'ba', 'bb', 'bc', 'ca', 'cb', 'cc']

Outil d'attaque par brut force : Un outil d'attaque par brute force est utilisé pour deviner un mot de passe en essayant toutes les combinaisons possibles jusqu'à ce que le mot de passe correct soit trouvé. Cela peut prendre un certain temps, selon la longueur et la complexité du mot de passe. Les outils de brute force automatisent ce processus en utilisant des algorithmes pour tester rapidement une grande variété de combinaisons.

Objectif

Ce script effectue une attaque brute force en générant toutes les combinaisons possibles de lettres minuscules (a-z) pour une taille donnée.

Fonctionnement

- Utilisation de la même fonction récursive mais avec l'alphabet complet (`string.ascii_lowercase`).
- Mesure du temps que prend la génération des mots de passe.

Exemple

Pour une taille de 3, les mots générés incluent aaa, aab, aac, ... jusqu'à zzz.

Outil d'attaque par dictionnaire : Une attaque par dictionnaire consiste à tester une liste de mots de passe préexistants qui sont souvent utilisés par les

utilisateurs (mots courants, combinaisons simples, etc.). Un outil d'attaque par dictionnaire prend généralement une liste de mots et tente chaque mot comme mot de passe. Cette méthode est plus rapide que la brute force, mais elle est limitée par la qualité du dictionnaire utilisé.

Objectif

Tester une liste de mots de passe courants (contenus dans un fichier dictionnaire) contre un hash pour retrouver le mot de passe d'origine.

Fonctionnement

- Lecture ligne par ligne d'un fichier dictionnaire.
- Chaque mot est haché et comparé avec la cible.

Avantage

Beaucoup plus rapide qu'une brute force, mais nécessite un bon dictionnaire.

Outil de scanner de ports

Explication du code :

1. **Scanner de ports (scanner_ports) :** Cette fonction prend une adresse IP et une plage de ports à scanner. Elle tente de se connecter à chaque port dans la plage spécifiée en utilisant des sockets. Si la connexion réussit (ce qui signifie que le port est ouvert), le port est ajouté à la liste des ports ouverts.
2. **Affichage de l'heure :** Le scanner affiche l'heure de début et de fin du scan pour fournir une idée du temps écoulé pendant l'exécution.
3. **Paramètres de connexion :**
 - **socket.socket(socket.AF_INET, socket.SOCK_STREAM) :** Crée une connexion réseau de type TCP/IP (IPv4).
 - **s.connect_ex() :** Tente de se connecter à l'adresse IP et au port spécifié. Si la connexion réussit, le code retourne 0, indiquant que le port est ouvert.

Exemple d'utilisation :

1. Entrez une adresse IP (par exemple 192.168.1.1) pour scanner les ports de la machine distante.
2. La fonction scanne les ports dans la plage donnée (par exemple, de 20 à 100) et affiche les ports ouverts.

Améliorations possibles :

- Ajouter la possibilité de scanner des plages de ports plus larges.
- Gérer les exceptions et erreurs réseau plus efficacement.
- Ajouter une option pour scanner des ports spécifiques ou toute la plage des ports connus.

Outil de vérification de la force du mot de passe

1. Évaluation de la force du mot de passe (`evaluer_force_mot_de_passe`) :

- Le mot de passe est évalué selon plusieurs critères :
 - Longueur minimale de 8 caractères.
 - Présence de lettres majuscules.
 - Présence de lettres minuscules.
 - Présence de chiffres.
 - Présence de caractères spéciaux.
- Le score est augmenté en fonction de ces critères, et le mot de passe est classé comme faible, moyen, fort ou très fort en fonction du nombre de critères remplis.

2. Interface graphique avec Tkinter :

- L'utilisateur entre un mot de passe dans un champ de texte, puis clique sur un bouton pour évaluer la force du mot de passe.
- Un message s'affiche avec la sécurité du mot de passe (très faible, moyen, fort, très fort).

Utilisation :

1. L'utilisateur entre un mot de passe dans le champ de texte.
2. En cliquant sur le bouton "Vérifier la force", l'outil affiche une fenêtre avec l'évaluation de la force du mot de passe.

Exemple de classification :

- **Mot de passe très faible** : "12345", "password".
- **Mot de passe moyen** : "Password1", "abc123".
- **Mot de passe fort** : "P@ssw0rd!", "M0nT3pa\$\$".

- **Mot de passe très fort** : "2X!mVqz7@tP8", "1G\$tkL&!qW9z".

Améliorations possibles :

- Ajouter un système de recommandation pour renforcer le mot de passe si nécessaire (ex. : "Essayez d'ajouter des caractères spéciaux ou des chiffres").
- Ajouter un indicateur de force visuel (par exemple, une barre de progression ou des couleurs).
- Permettre à l'utilisateur de tester plusieurs mots de passe d'affilée.

Cet outil peut être très utile pour encourager les utilisateurs à choisir des mots de passe plus sécurisés.