

Evaluation – Python Securite

Exercice 1

Détection de Mot de Passe en Dur dans le Code Source

Objectif : Développer un outil qui scanne des fichiers Python à la recherche de mots de passe en dur, en utilisant **expressions régulières et analyse statique du code**.

Tâches :

1. Écrire un script qui:
 - Parcourt un répertoire donné en paramètre dans le terminal et identifie tous les fichiers .py.
 - Analyse chaque fichier pour repérer les chaînes suspectes, comme :
 - Assignations du type password = "secret"
 - Clés API exposées sous forme de chaîne de caractères
 - Affiche les lignes suspectes et le fichier où elles apparaissent.
2. Ajouter un mode **"strict"**, qui bloque certains types de chaînes (ex : contenant secret, key, pwd, token).

Exercice 2

Création d'un Faux Serveur Piège (Honeypot)

Objectif : Implémenter un **honeypot simple** qui écoute sur des ports sensibles et **enregistre les tentatives de connexion** pour détecter des activités suspectes.

Tâches :

1. Créer un serveur Python qui écoute sur plusieurs ports sensibles (22, 3306, 8080, etc.).
2. Dès qu'une connexion est détectée, enregistrer :
 - L'adresse IP de l'attaquant
 - Le port cible
 - L'heure de la connexion
 - Une éventuelle commande envoyée par l'attaquant
3. Afficher les résultats en temps réel.

Exercice 3

Détection d'URL Malveillantes

Objectif : Écrire un **analyseur d'URL** qui identifie les liens potentiellement malveillants en utilisant des critères de **vérification heuristique**.

Tâches :

1. Charger une liste d'URL depuis un fichier texte.
2. Pour chaque URL, vérifier plusieurs indicateurs de risque :
 - Utilisation de domaines courts et anonymes (bit.ly , tinyurl, etc.).
 - Présence de mots-clés suspects (free, win, prize, etc.).
 - Utilisation d'un grand nombre de **caractères spéciaux** dans l'URL.
 - Vérification avec une liste noire de domaines suspects.
3. Noter chaque URL sur **10** en fonction des critères détectés.

Bonus :

- Ajouter une vérification avec une API comme VirusTotal pour obtenir un score de réputation.