Mini Projets Python en Cybersécurité avec Visualisation

Projet 0 : Automatisation de la Classification de Dossiers par Extension

Objectif :

Développer un script Python pour automatiser la classification des fichiers dans différents dossiers en fonction de leur extension, et créer un fichier de log pour enregistrer les actions effectuées.

Description du Projet :

1. **Scanner un répertoire source :**
    - Parcourir tous les fichiers dans un répertoire donné par l'utilisateur.
2. **Classer les fichiers :**
    - Identifier l'extension de chaque fichier (ex: .txt, .pdf, .jpg).
    - Créer des sous-dossiers pour chaque type de fichier (s'ils n'existent pas déjà).
    - Déplacer chaque fichier dans le sous-dossier correspondant à son extension.
3. **Créer un log :**
    - Enregistrer chaque action effectuée dans un fichier de log (fichiers déplacés, chemin d'origine, chemin de destination, erreurs éventuelles).
    - Générer un rapport final avec le résumé des opérations.

Étapes Détailées :

1. **Initialisation et Configuration :**
    - Demander à l'utilisateur de spécifier le répertoire source.
    - Créer les sous-dossiers pour chaque type de fichier en fonction des extensions trouvées.
2. **Parcours et Classification des Fichiers :**
    - Parcourir tous les fichiers dans le répertoire source.
    - Pour chaque fichier, déterminer son extension.
    - Déplacer chaque fichier dans le sous-dossier approprié.
3. **Création de Log :**

- Utiliser une bibliothèque de gestion des logs pour enregistrer les actions effectuées.
- Enregistrer les informations suivantes : fichier déplacé, chemin d'origine, chemin de destination, erreurs rencontrées.
- Générer un rapport final avec un résumé des opérations effectuées et des éventuelles erreurs.

Bibliothèques Recommandées :

- `os` pour la gestion des fichiers et des répertoires.
- `shutil` pour déplacer les fichiers.
- `logging` pour créer et gérer le fichier de log.

Compétences Développées :

- Manipulation des fichiers et des répertoires en Python.
- Utilisation des bibliothèques Python pour automatiser des tâches.
- Création et gestion de fichiers de log.
- Structuration d'un projet Python simple pour automatiser des processus courants.

Ressources et Pré-requis :

- Installation de Python 3.x.
- Familiarité de base avec la manipulation des fichiers et des répertoires en Python.
- Connaissance de base des bibliothèques Python `os`, `shutil` et `logging`.

Consignes pour les Étudiants :

1. **Configurer l'Environnement de Développement :**
   - Installer Python et un éditeur de texte ou un IDE (VSCode, PyCharm, Jupyter Notebook).
2. **Développer le Script :**
   - Écrire un script Python qui réalise les tâches décrites.
3. **Tester et Valider le Script :**
   - Tester le script avec différents types de fichiers et répertoires.
   - Vérifier que le script crée correctement les sous-dossiers et déplace les fichiers.
   - Assurer que le fichier de log enregistre toutes les actions et erreurs de manière adéquate.
4. **Présenter les Résultats :**
   - Fournir un rapport final décrivant le fonctionnement du script, les défis rencontrés et les solutions apportées.
   - Inclure des exemples de log générés et des captures d'écran des dossiers avant et après l'exécution du script.

Projet 1 : Analyse de Logs d'Accès Web

- **Objectif :** Écrire un script Python pour analyser les logs d'accès web et détecter des tentatives d'intrusion.
- **Description :**
  - Importer un fichier de logs d'accès web (format CSV).
  - Analyser les logs pour identifier des schémas suspects comme les tentatives de connexion répétées.
  - Visualiser les adresses IP les plus fréquentes et les types de requêtes.
  - Générer un rapport des IP suspectes.
- **Bibliothèques suggérées :** `pandas` pour la manipulation des données, `re` pour la recherche de motifs, `matplotlib` ou `seaborn` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Lecture de fichiers, manipulation de données avec pandas, expressions régulières simples, visualisation de données.

Projet 2 : Scanner de Ports Simple

- **Objectif :** Développer un outil de scanner de ports simple pour détecter les ports ouverts sur une machine cible.
- **Description :**
  - Écrire un script qui teste une liste de ports sur une adresse IP spécifiée.
  - Afficher les ports ouverts.
  - Visualiser les résultats sous forme de graphique (barres des ports ouverts).
- **Bibliothèques suggérées :** `socket` pour la communication réseau, `matplotlib` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Utilisation de la bibliothèque socket, boucles pour tester plusieurs ports, visualisation de données.

Projet 3 : Surveillance de Fichiers

- **Objectif :** Écrire un script pour surveiller les modifications de fichiers dans un répertoire.
- **Description :**
  - Utiliser une bibliothèque pour surveiller les changements dans un répertoire spécifié.
  - Lorsqu'une modification est détectée, enregistrer l'événement dans un fichier de log.
  - Visualiser les événements de modification de fichiers sous forme de graphique temporel.
- **Bibliothèques suggérées :** `watchdog` pour la surveillance de fichiers, `matplotlib` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Utilisation de bibliothèques externes, gestion des événements, visualisation de données.

Projet 4 : Détection de Phishing

- **Objectif :** Créer un script pour analyser des échantillons d'emails et détecter des signes de phishing.
- **Description :**
  - Analyser le contenu des emails pour des liens suspects ou des expéditeurs non vérifiés.
  - Générer un rapport des emails suspectés de phishing.
  - Visualiser les statistiques des emails analysés (ex: pourcentage d'emails de phishing détectés).
- **Bibliothèques suggérées :** `email`, `re` pour les expressions régulières, `matplotlib` ou `seaborn` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Analyse de texte, regex pour détecter les motifs de phishing, visualisation de données.

Projet 5 : Outil Bruteforce pour les Mots de Passe

- **Objectif :** Développer un script pour effectuer des attaques de brute-force simples sur des mots de passe.
- **Description :**
  - Utiliser une liste de mots de passe courants pour essayer de se connecter à un service simulé.
  - Enregistrer les tentatives réussies et échouées.
  - Visualiser les résultats sous forme de graphique (tentatives réussies vs échouées).
- **Bibliothèques suggérées :** `itertools` pour la génération de combinaisons, `time` pour les délais entre les tentatives, `matplotlib` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Boucles, manipulation de listes, compréhension des attaques par force brute, visualisation de données.

Projet 6 : Analyse des Logs de Sécurité Windows

- **Objectif :** Analyser les logs de sécurité Windows pour identifier des événements suspects.
- **Description :**
  - Importer des logs de sécurité Windows (format CSV).
  - Rechercher des événements spécifiques comme les tentatives de connexion échouées.
  - Visualiser les types d'événements et leur fréquence.
  - Générer un rapport des activités suspectes.
- **Bibliothèques suggérées :** `pandas` pour la manipulation des données, `matplotlib` ou `seaborn` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Lecture et manipulation de fichiers CSV, analyse de données, visualisation de données.

Projet 7 : Vérification de Certificats SSL

- **Objectif :** Écrire un script pour vérifier l'état et la validité des certificats SSL des sites web.
- **Description :**
  - Créer un script qui extrait les informations SSL des sites web.
  - Vérifier la date d'expiration et générer des alertes pour les certificats expirés.
  - Visualiser les certificats valides et expirés sous forme de graphique.
- **Bibliothèques suggérées :** `ssl`, `socket`, `matplotlib` pour la visualisation.
- **Difficulté :** Basique
- **Compétences :** Utilisation de bibliothèques réseau, extraction d'informations SSL, visualisation de données.

Conseils Pédagogiques

- **Encadrement :** Fournissez des tutoriels et des exemples de code pour chaque projet.
- **Assistance :** Organisez des sessions de support pour aider les étudiants à surmonter les difficultés.
- **Ressources :** Proposez des jeux de données et des environnements de test simples.

Ces projets permettent aux débutants d'appliquer des concepts de cybersécurité à travers des exercices pratiques en Python, avec des visualisations pour mieux comprendre et présenter les résultats.