Mini-Projet Python – Analyse de sécurité & scanner réseau

Abilash SELLATHURAI P37 M2IRS

**I) Objectif du projet**

Ce mini-projet avait pour but de développer un outil de cybersécurité en Python. Cet outil devait être capable d’analyser des fichiers de logs système afin de détecter des adresses IP suspectes. Il devait également permettre la visualisation graphique des résultats et offrir la possibilité de tester l’activité de ces adresses IP sur certains ports réseau.

En complément, l’outil devait intégrer une interface en ligne de commande, une interface interactive, ainsi qu’un système d’exportation des résultats dans différents formats. Ce projet met en évidence l’utilisation de Python pour renforcer la sécurité d’un système informatique de manière simple, efficace et accessible.

**II) Étapes du projet**

Le projet a été organisé en trois grandes phases, correspondant chacune à une fonctionnalité essentielle de l’outil de sécurité.

**1. Analyse des logs**

La première étape a consisté à analyser un fichier de logs système Linux de type auth.log. Un script Python a été conçu pour extraire les lignes contenant des tentatives de connexion échouées, identifiées par le message « Failed password ». À l’aide d’une expression régulière, le script a pu isoler les adresses IP associées à ces tentatives.

Ensuite, un dictionnaire a été utilisé pour comptabiliser le nombre d’occurrences de chaque IP. Cette méthode permet d’identifier rapidement les adresses les plus fréquemment impliquées dans des connexions suspectes.

**2. Analyse statistique et visualisation**

Les adresses IP extraites ont été intégrées dans une structure de données tabulaire grâce à la bibliothèque **pandas**. Les données ont été triées afin de faire ressortir les adresses les plus actives, notamment les cinq premières, qui ont fait l’objet d’une analyse plus approfondie.

Un graphique en barres a ensuite été généré avec **matplotlib**, permettant de visualiser ces adresses IP de manière claire et intuitive. Les résultats ont été exportés aux formats **CSV** et **HTML**, facilitant leur archivage et leur réutilisation.

**3. Scanner réseau multithreadé**

La dernière étape du projet consistait à créer un scanner réseau capable de vérifier si certaines adresses IP avaient des ports ouverts parmi une sélection de ports sensibles (tels que 22, 80 et 443).

Un premier scanner fonctionnant en mono-thread a été mis en place, puis optimisé à l’aide de la bibliothèque **threading** pour améliorer les performances grâce au multithreading. Une option --verbose a également été ajoutée pour permettre l’affichage des ports fermés, offrant un diagnostic plus complet.

Les résultats du scan ont été enregistrés dans des fichiers **CSV** et **HTML**, répertoriant les IP analysées et les ports ouverts détectés. Un menu interactif en mode texte a été intégré afin de permettre à l’utilisateur de sélectionner facilement l’action souhaitée (analyse, visualisation ou scan). Par ailleurs, une interface CLI complète a été développée à l’aide du module **argparse**, rendant l’outil encore plus flexible en ligne de commande.

**III) Exemples de résultats**

Parmi les adresses IP identifiées, l’adresse 203.0.113.5 s’est révélée la plus active, avec quatre tentatives de connexion échouées, suivie par 192.168.1.12, impliquée dans trois tentatives. Le graphique en barres généré a mis en évidence ces adresses en tête des activités suspectes.

Le scanner réseau a également permis de détecter que certaines de ces adresses, comme 203.0.113.5, disposaient de ports ouverts (notamment les ports 22 et 80), ce qui pourrait représenter un point d’entrée potentiel pour une attaque.

L’ensemble des résultats a été archivé dans des fichiers CSV et HTML au sein d’un répertoire dédié.

**IV) Conclusion**

Ce projet a permis de concevoir un outil efficace pour détecter des activités suspectes à partir de fichiers de logs système. En y ajoutant des fonctionnalités d’analyse statistique, de visualisation graphique et de scan réseau multithreadé, l’outil devient un véritable assistant pour la surveillance de la sécurité réseau.

Grâce à une architecture modulaire, le code reste facile à maintenir et à améliorer. L’ajout d’une interface en ligne de commande et d’un menu interactif rend l’utilisation de l’outil accessible et flexible.

Ce projet constitue ainsi une excellente base pour développer une solution plus complète de supervision de la sécurité informatique.