Rapport projet Forensic / SEC-FOR promo 2025 Groupe 12:

zabihullah.nasir@ecole2600.com
ramzi.rouabah@ecole2600.com
baptiste.cheynel@ecole2600.com
arthur.teig@ecole2600.com
hugo.boilloux@ecole2600.com
ludovic.jeanvoine@ecole2600.com

Présentation du projet :

Le projet a comme finalité de réaliser un outil d'extraction automatique de données forensique à partir d'une image disque.
L'objectif de ce projet est de :

- Réaliser un outil fonctionnel en Python utile pour une analyse Forensique
- Travailler en équipe organisée
- Produire un outil lisible, évolutif et maintenable, pas uniquement pour vos pires ennemis
- Savoir rédiger de la documentation
- Montrer des exemples d'utilisation
- Mettre en oeuvre les connaissances du cours

Argumentation de l'architecture du code :

Classe ForensicExtractor :

- Cette classe encapsule l'ensemble des fonctionnalités d'extraction de données.
- Elle contient des méthodes pour exécuter des commandes système, lister les partitions, extraire des fichiers spécifiques, et plus encore.
- La classe est conçue de manière à ce que chaque méthode ait une responsabilité claire et spécifique dans le processus d'analyse forensique.

Méthodes de Traitement de Données :

- Le code utilise des méthodes dédiées pour traiter différentes étapes de l'analyse forensique, telles que l'extraction de partitions, l'extraction de fichiers Windows, et l'extraction de fichiers spécifiques.
- Les retours des utilitaires The Sleuth Kit sont gérés par des expressions régulières permettant de cibler les données voulues en conservant un temps de traitement relativement faible
- Chaque méthode est conçue pour être réutilisable et modulaire, permettant ainsi une extensibilité facile pour prendre en charge de nouvelles fonctionnalités ou types d'analyses.

Utilisation d'Outils Externes :

- Le code interagit avec des outils externes tels que mmls et fls pour obtenir des informations sur les partitions et les fichiers.
- Il utilise également des expressions régulières pour extraire des données spécifiques à partir de la sortie de ces outils, ce qui ajoute une flexibilité dans le processus d'extraction.

Gestion de Configuration :

- Le code utilise un fichier de configuration YAML pour spécifier les fichiers d'intérêt à extraire. Cette approche permet une personnalisation facile du processus d'extraction en fonction des besoins spécifiques de l'utilisateur.

Test implémenté / possible :

Test d'Extraction de Partitions :

- Vérifier si le code peut correctement extraire les informations sur les partitions à partir de l'image de disque en utilisant des données d'entrée connues.

Test d'Extraction de Fichiers Windows :

- Vérifier si le code peut extraire avec succès les hives de registre Windows (SYSTEM, SECURITY, SOFTWARE, SAM) à partir de l'image de disque.

Test d'Extraction de Fichiers Spécifiques :

- Vérifier si le code peut extraire des fichiers spécifiques à partir d'une partition donnée en utilisant des chemins de fichiers valides.

Test de Gestion d'Erreurs :

- Vérifier la robustesse du code en simulant des erreurs telles que des partitions manquantes ou des fichiers introuvables, et vérifier si le code gère ces situations de manière appropriée.

Test de Performance :

- Mesurer le temps nécessaire pour extraire différents types de données à partir de l'image de disque et évaluer la performance globale du code.

Test de Configuration YAML :

- Vérifier si le code charge correctement les configurations à partir du fichier YAML, et si les fichiers spécifiés dans la configuration sont extraits avec succès.

Test de Réutilisabilité :

- Réaliser plusieurs tests en utilisant différentes images de disque et configurations pour évaluer la capacité du code à s'adapter à différentes situations.

Test de Documentation :

- Vérifier si la documentation du code est claire et complète, et si elle fournit des informations suffisantes sur l'utilisation et les fonctionnalités du code.

Test de Fonctionnalités Supplémentaires :

- Explorer d'autres fonctionnalités potentielles du code, telles que l'extraction d'informations de navigateur ou d'autres types de fichiers système, et vérifier si le code peut les prendre en charge correctement.

Cas non supporté :

Bugs connu (et si on saurait les résoudre ou non):