FORENSIC | Fouine <3
Forensic Software Tool

Master 1 Cybersécurité

Groupe 2

Nissim ABEHCERA
Télio GUIGNARD
Arthur ALEXANDRE
Félix BLANC
Victor MARCY
Youenn TILLAR

Table des matières

Description du sujet d’étude

La forensic consiste à collecter, analyser et interpréter des données numériques dans un contexte juridique. L'objectif est d’enquêter et de trouver des preuves électroniques.

Pour ce faire, on utilise des outils d'analyse forensic, tels que des logiciels spécialisés, pour récupérer des données à partir d'ordinateurs, de disques durs, de smartphones et d'autres appareils électroniques. Ces outils peuvent extraire des informations à partir de fichiers, de bases de données, de registres système et de nombreux autres types de données. Les résultats de l'analyse sont ensuite utilisés pour établir des preuves numériques, telles que des fichiers de journalisation, des captures d'écran et des vidéos.

Contextualisation

A l'occasion de notre Master 1 de Cybersécurité d'École 2600, première école française 100% dédiée à l’enseignement de la cybersécurité, les étudiants de 2ème année (promotion 2024),
Il nous a été proposé de réaliser un outil forensic de collecte de données numériques en Python afin d’extraire des fichiers d’une image disque au format EWF.

Objectifs

- **Réaliser un outil fonctionnel en Python, utile pour une analyse forensique ;**
- **Travailler en équipe organisée ;**
- **Produire un outil lisible, évolutif et maintenable ;**
- **Savoir rédiger de la documentation ;**
- **Réaliser une démonstration des capacités de l’outil ;**
- **Mettre en oeuvre les connaissances du cours de forensics ;**

Gestion de projet

Répartition des tâches

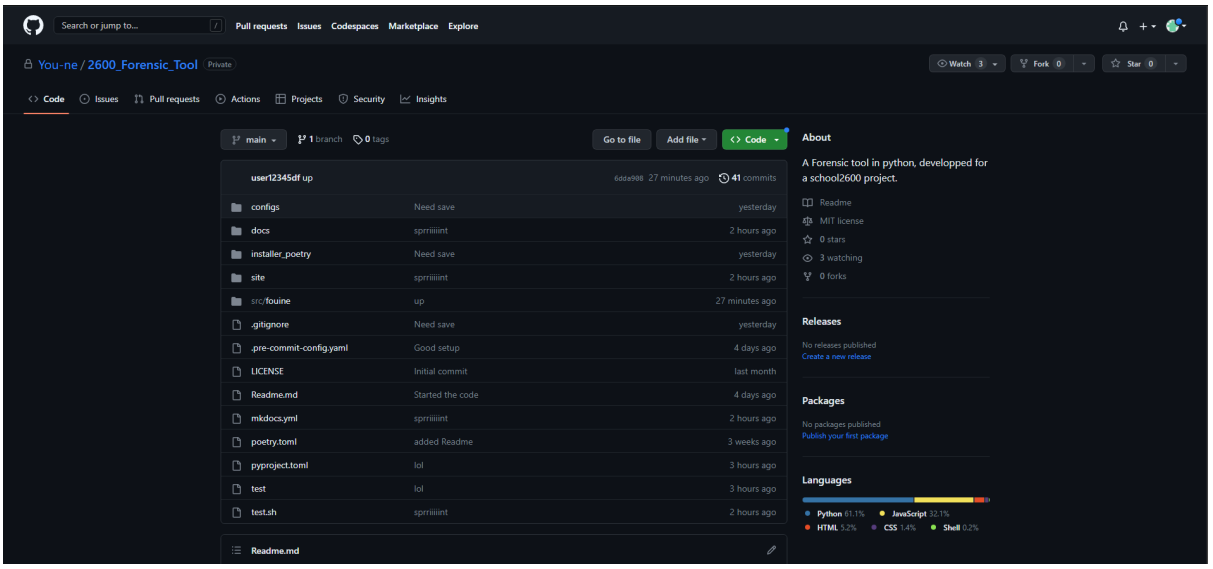
Pour ce projet impliquant 6 étudiants travaillant en groupe aléatoire, les tâches ont été malheureusement réparties comme suit :

Responsables	Tâches
Youenn Tillard	Mise en place du projet DevOps | Lafouine.py | Les configurations | Le Parser
Félix Blanc	Core.py | Event Logs | Documentation
Télio Guignard	Documentation, Testing de l'outil | Rapport de projet

Repo - GitHub

Nous avons utilisé le dépôt Github https://github.com/You-ne/2600_Forensic_Tool pour le développement, ce qui nous a permis de suivre une approche classique de gestion de projet Open-Source.

En utilisant un dépôt Github, il est possible de fournir aux utilisateurs des informations plus techniques sur le code, ainsi que des instructions d'installation et d'utilisation.



DevOps Infrastructure

Poetry

Manage les dépendances et les configurations des autres outils du projet.

Black

Reformate les fichiers pour appliquer un style de code.

FlakeHeaven

Vérifie la conformité du code au **PEP8** et d'autres conseils syntaxiques.

Mypy

Vérifie l'application du type-hinting.

Pre-Commit

Run Black, FlakeHeaven and Mypy avant chaque commit.

MK-Docs

Fabrication d'une documentation complète avec doc API.

GIT …

Description du logiciel

Notre logiciel d'extraction de données appelé "Fouine" a pour but d'extraire des artefacts d'une image awf, selon une liste d'artefact désirés, spécifiés dans un fichier.yml en suivant la même structure des KapesFile

Architecture globale

Matrice des flux entre modules

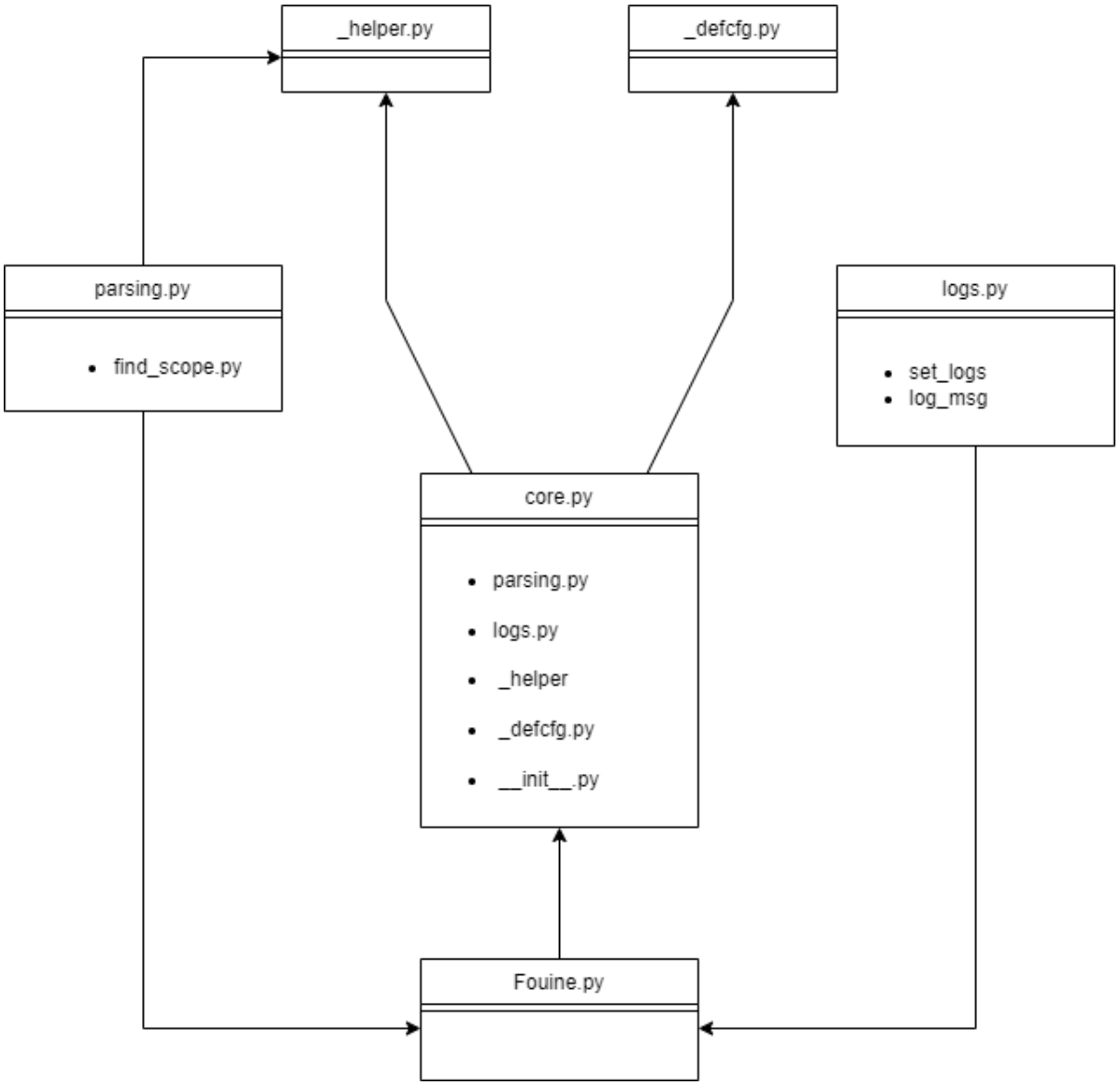
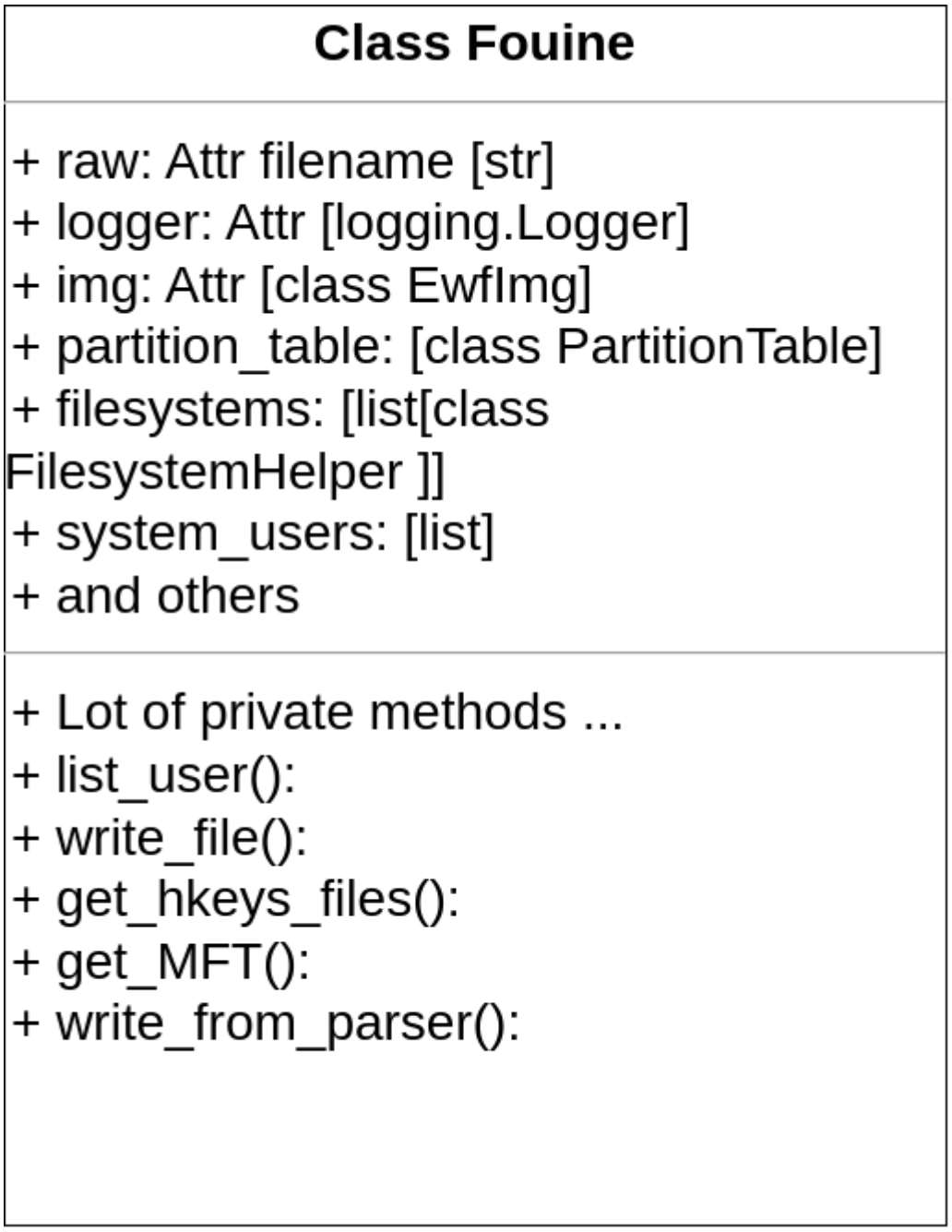


Diagramme de classe - “Fouine”



Scénario de test

Prérequis

Les tests doivent être conduits sous environnement Windows 7 à Windows 11. La librairie The Sleuth Kit doit avoir été installée au préalable.

Objectifs

- **Extraction des fichiers du registre système & les ruches utilisateurs ;**
- **Extraction des données de navigateurs Internet Edge, Internet Explorer, Firefox & Chrome ;**
- **Extraction des journaux Windows Security & System au minimum ;**
- **Extraction de la MFT ;**

Conduit de test

Phase	Nom du test	Description du test	Attendus	Résultats obtenus
1	TKAPE PARSER	The tool should correctly parse .tkape file and retrieve the file in the EWF image if it exist.	Extraction of artefacts stored in .tkape format.	Valid
1.1	Parsing of file	Run fouine with –config	While loading tkape file it should produce Target artefacts for the core to	Valid

			process	
1.2	Processing of Target artefacts	There should be no error in handling target Artefacs	No path related errors during processing	Valid
2	Default Config	The tool should correctly load from itself to achieve a basic scan stored in a default config	Extraction of artefacs stored in a default config in the form of artifacts	Valid
2.1	Parsing of Artefacts	While loading DEFAULT_CONFIG the core should get correct Artefacts	No error while parsing	Valid
2.2	Extraction	Extract specified targets to host	Retrieve files on disk	Valid

Cas non supportés

Systèmes de fichiers différents de : TFS, NFTS, FAT12-16-32, EXT2-3-4, HFS, ISO, YAFFS, SWAP, FFS

Chemins de fichiers exotiques potentiellement non traites ou cause d’erreurs

Limites et feature inachevées

Par manque de temps les tests n’auront été conduits que sur l’image ewf fournie en classe.

Traçabilité, pas de rapport final sur les fichier attendus et extraits, afin d’examiner les fichiers non trouvés

Prompt tool kit (Mini shell) le shell de commande qui permet d’explorer un filesystem afin d’y jeter un coup d'oeil humain

Exporter les artefacts récursivement par leur attribut de catégorie.

Bonus

Parser de .tkape

Extraction dynamique en fonction des users

Choix si plusieurs systèmes de fichiers si possible

Génération d’une doc API avec mkdocs

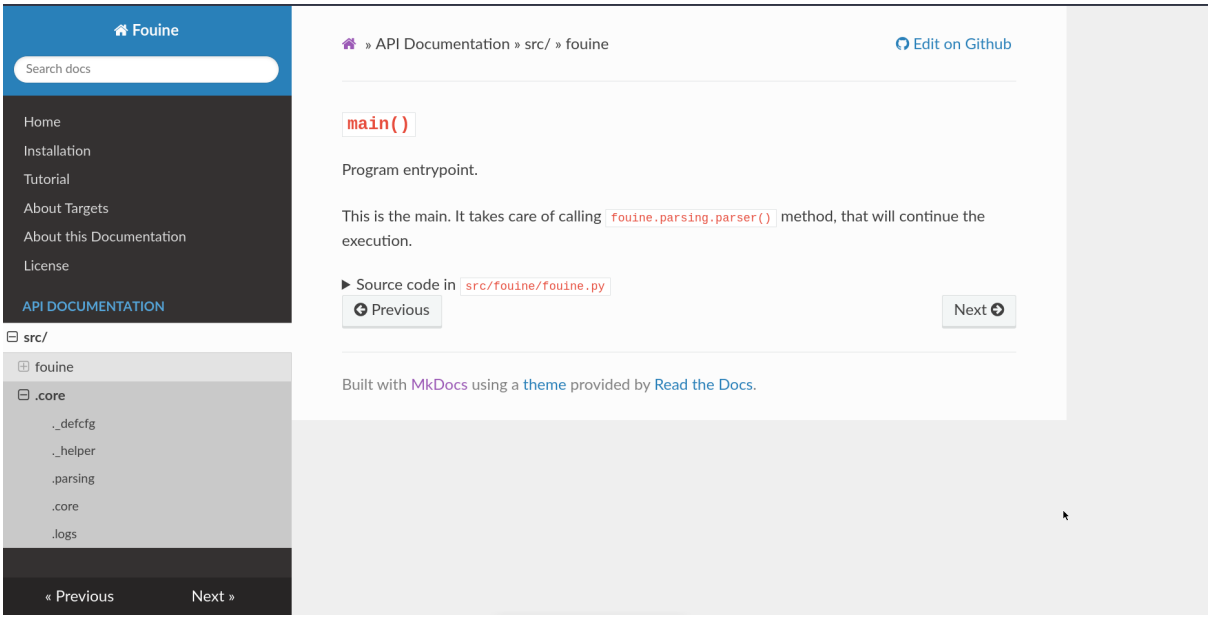
Annexes

```
(fouine/3.10) njord@NJEFS78 ~/Desktop/Devs/2600/FOR/code/2600_Forensic_Tool/src
 $ find . -type f -name "*.py" -exec cat {} + | wc -l
2860
```

- 2860 lignes
- +10 classes
- Des logs en couleurs
- Tests: +300 artefacts d'intérêt extrait

Github - https://github.com/You-ne/2600_Forensic_Tool

Capture d'écran d'une page de documentation



Fouiny Babe after finding some crispy artifact