PIPR - Dokumentacja projektu p.t.:

AntyWirus

Autor:
Jakub Kowieski

Przykładowe logo programu

Treść projektu

AntyWirus jest to prosty program typu antywirus. Napisany w całości w pythonie oraz korzystający z plików typu json i csv. Posiada on interfejs tekstowy (w terminalu). Program pozwala na następujące czynności:

- Prosty skan katalogu pod kątem wirusów - na podstawie konkretnych reguł
- Stworzenie indeksu plików w konkretnym katalogu - lista opisująca pliki, zawierająca m.in. ścieżki, informacje o stanie pliku i dodatkowe informacje pozwalające na szybkie skanowanie (np. hash)
- Aktualizacja indeksu plików
- Szybki skan katalogu - sprawdzenie tylko nowych lub zmodyfikowanych plików
- Cykliczny szybki skan - jak wyżej, ale w regularnych, konfigurowalnych odstępach czasowych
- Naprawienie zainfekowanych plików - wycięcie wirusa

Opis klas:

File - jest to klasa zajmująca się przechowywaniem danych o plikach. Najważniejszą metodą jest możliwość sprawdzania czy dany plik był modyfikowany.

IndexFile - klasa jest typem prostej “bazy danych”, zawiera listę obiektów typu File, bezwzględną ścieżkę skanowania, oraz bezwzględną scieżkę do folderu w którym znajduje się plik “.index_file”. Pozwala ona na różne operacje na plikach, podmieniania starego na nowy, usuwania czy zwykłe dodawanie do listy. Poprzez moduł antiwirus_io może także tworzyć, czytać oraz nadpisywać plik “.index_file”.

AntiWirus - główna klasa odpowiadające za najważniejsze funkcje programu. Zawiera obiekt IndexFile oraz listę funkcji typu rules (o niej później). Jest jednyną klasą komunikującą się z interfejsem. Posiada metody, które umożliwiają szybkie skanowanie, proste skanowanie, reperowanie - wycinanie wirusów, aktualizacje obiektu IndexFile.

AntiWirusUI - klasa jako jedna ma styczność z użytkownikiem, czyli musi być to interfejs tekstowy. Przechowuje obiekt AntiWirus oraz ostatni sprawdzany stan czasu.Odpowiada ona za poprawną komunikację użytkownika i klasy AntiWirus. Najważniejszą funkcją jest _run(), która obsługuje menu operacji. Ma także zaimplementowane cykliczne wywoływanie szybkiego stanu (modyfikowalny poprzez zmienną `CYCLE_TIME)` używając modułu time.

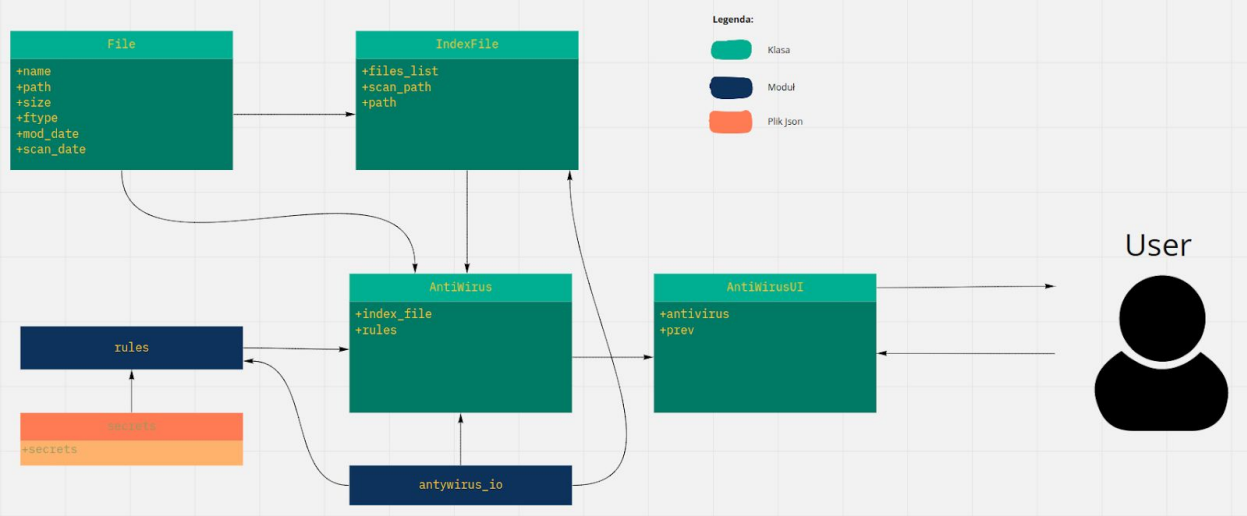
Opis modułów:

antiwirus_io - moduł, który zawiera funkcje działające na plikach. Korzysta z obiektów klay File.

rules - zawiera listę funkcji RULES, w której się znajdują funkcję typu rules, czyli jako parametry musi mieć plik, string pliku (zawartość pliku jako tekst) oraz tryb w jakim wywołuje się funkcję, jeżeli mode jest równy ”repair” to funkcja “wycina” wirusa i zwraca plik, w innym wypadku plik jest skanowany, jeżeli znajdzie wirusa zwraca True, lecz jeśli nie znajdzie zwraca dla danej reguły zwraca False.

settings - zawiera CWD, który jest ścieżką bezwzględną do katalogu packages

Diagram komunikacji klas i modułów:



Sposób Uruchamiania:

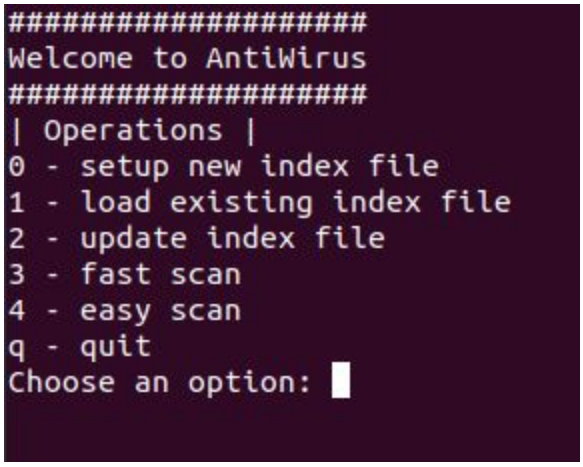
Wymagany jest interpreter Python 3.8.5. Program musi być powinien być uruchamiany na systemie Linux (testowane na Ubuntu 20.04.1 LTS).

Żeby uruchomić trzeba użyć komendy “python3” na pliku “main.py” w głównym folderze programu.

(jeżeli użytkownik znajduje się w głównym folderze wystarczy użyć “python3 main.py”).

Jak korzystać z programu:

Menu AntiWirusa



Wystarczy wpisać numer operacji jaki chcemy wykonać i program dalej nas poprowadzi.
Żeby natychmiast zamknąć program należy nacisnąć CTRL-Z lub CTRL-C, a jak chcemy wyjść z menu wystarczy wcisnąć “q”.
Jeżeli program pyta nas o ścieżkę do danego folderu to należy wpisać BEZWZGLĘDNĄ ŚCIEŻKĘ do niego. Inaczej program potraktuje to jako niewłaściwą ścieżkę.
Gdy pojawiają się nawiasy klamrowe “[Y/n]” to należy wpisać “y”, czyli pozwalamy zrobić to co program proponuje albo “n” jeżeli nie.

Główna struktura plików:

antywirus/

├─── __init__.py

├─── main.py

├─── package/

│ ├─── antiwirus_io.py

│ ├─── antiwirus.py

│ ├─── antiwirus_ui.py

│ ├─── file.py

│ ├─── index_file.py

│ ├─── __init__.py

│ ├─── rules.py

│ ├─── secrets.json

│ └─── settings.py

Testy:

Test stworzenia indeksu plików

```
####################
Welcome to AntiWirus
####################
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: 0
Choose a path for index file: /home/kuba/v2/antiwirus
Choose a path for fast scan: /home/kuba/v2/testy
Success !!! Index file is in: /home/kuba/v2/antiwirus
```

Test załadowanie indeksu plików

```
####################
Welcome to AntiWirus
####################
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: 1
Choose a path where is index_file: /home/kuba/v2/antiwirus
Successfully loaded index file
```

Test aktualizowania indeksu plików

```
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: 2
Successfully updated index file
```

Test szybkiego skanowania

```
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: 3
Successfully updated index file
0 dangerous files found
Successfully repaired all dangerous files
```

Test łatwego skanowania

```
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: 4
Choose a scan path: /home/kuba/v2/testy
0 dangerous files found
```

Test wyjścia z menu

```
| Operations |
0 - setup new index file
1 - load existing index file
2 - update index file
3 - fast scan
4 - easy scan
q - quit
Choose an option: q
```

Dalszy rozwój:

- Dodanie graficznego interfejsu
- Dodanie większej ilości reguł
- Ułatwienie wpisywania ścieżek (np. dokańczanie nazw folderów poprzez kliknięcie Tab)