

LAB 1 : Développer un Keylogger

Introduction

Un Keylogger est un programme informatique qui enregistre chaque frappe effectuée par un utilisateur d'ordinateur, notamment dans le but d'accéder frauduleusement à des mots de passe et à d'autres informations confidentielles. Dans ce TP, vous allez réaliser, à l'aide du langage Python, votre propre keylogger.

Exercice - Coder un Keylogger

1. Expliquer brièvement le concept de keylogger. Et quels sont les dangers encourus si on est infecté pour ce genre de code malicieux ?
2. Y a-t-il une utilisation légitime pour ce genre de programme ? expliquer
3. Commencer par installer la bibliothèque `pynput`. Puis importer `Pynput.keyboard`
4. Créer un nouveau programme nommé **“keylogger.py”**.
5. Créer la méthode `processkeys(key)` qui prend comme paramètre une touche du clavier sur laquelle on a appuyé, affiche sa valeur.
6. Nous allons maintenant utiliser la méthode `pynput.keyboard.listener()` qui va se mettre en écoute, et va capturer les touches sur lesquelles nous allons appuyer.
 - 6.1. Déclarez une variable `keyboard_listener` qui va contenir le résultat de l'appel de la fonction `pynput.keyboard.listener()` avec le paramètre `on_press=processkeys`.
 - 6.2. Quel est le rôle du paramètre `on_press` ?
7. Ajouter les lignes de code suivantes :

```
with keyboard_listener : keyboard_listener.join()
```

8. Quel est le rôle des instructions ci-dessus ?
9. Lancer votre programme, que se passe-t-il lorsque vous tapez sur les touches du clavier ?
10. C'est quoi le problème avec l'affichage ? Essayez d'améliorer l'affichage de telle sorte que :
 - 10.1. Au lieu de faire un simple print, déclarez une variable globale `log` de type string.
 - 10.2. Pour chaque touche pressée, votre programme va concaténer le caractère correspondant dans la variable `log`.
 - 10.3. Votre programme ne doit concaténer que les caractères saisis (utilisez l'attribut `key.char`).
Il faut mettre l'instruction à l'intérieur dans bloc `try ... except`. Pourquoi ?
 - 10.4. Votre programme doit également concaténer les espaces, les retours chariots, et les backspaces.
 - 10.5. Pour tout le reste (ex: `key.right`, `key.left`, ...), votre programme doit les remplacer par la chaîne vide.
11. Vous devez maintenant améliorer votre keylogger, en mettant en place un fichier log, qui va stocker les valeurs saisies. Pour cela :
 - 11.1. Importer les bibliothèques `thread`, et `os`.
 - 11.2. Déclarer une autre variable globale `path` de type `os.path("chemin vers le fichier log.txt")`
 - 11.3. Déclarer une fonction `report()`, celle-ci va se charger de sauvegarder les caractères saisis dans le fichier log. Pour cela :
 - 11.3.1. Déclarer les deux variables globales `log` et `path`
 - 11.3.2. Créer une variable `logfile = open(path, "a")`.
 - 11.3.3. Que fait la méthode `open` ?
 - 11.3.4. Que représente le paramètre "a" ?
 - 11.3.5. Rajouter l'instruction `logfile.close()`. A quoi sert-elle ?

12. Ajouter un appel à la fonction `report()` avant l'instruction `keyboard_listener.join()` (voir tâche numéro 7).
13. Tester votre programme.
14. Quels sont les points faibles de ce keylogger ?

Réflexion d'un(e) ingénieur(e)

15. Proposer une version améliorée de ce keylogger. Justifier le choix des améliorations et détailler les étapes de mise en œuvre.