

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°7 : Les listes

BTS CIEL
2 ^{ème} année
Page 1 sur 3

Exercice 1 : Analyse des connexions réseau

Vous avez une liste de paquets réseau, chaque paquet étant représenté par un tuple contenant deux informations :

1. La taille du paquet en octets (premier élément du tuple).
2. Le protocole utilisé (deuxième élément du tuple, par exemple 'TCP', 'UDP', 'ICMP').

Objectifs :

1. **Afficher tous les paquets réseau :**

Affichez chaque paquet de la liste sous la forme (taille, protocole).

2. **Identifier les paquets qui dépassent 1500 octets :**

Parcourez la liste et identifiez les paquets dont la taille est supérieure à 1500 octets. Affichez ces paquets.

3. **Calculer le pourcentage de paquets TCP et UDP :**

Calculez et affichez le pourcentage de paquets utilisant le protocole TCP et le protocole UDP par rapport au nombre total de paquets présents dans la liste.

Code de départ :

```
packets = [
    (500, 'TCP'),
    (1000, 'UDP'),
    (2000, 'TCP'),
    (50, 'ICMP'),
    (1600, 'TCP'),
    (800, 'UDP')
]
```

Résultats attendus :

```
Paquets qui dépassent 1500 octets :
(2000, 'TCP')
(1600, 'TCP')
```

```
Pourcentage de paquets TCP : 50.00%
Pourcentage de paquets UDP : 33.33%
```

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°7 : Les listes

BTS CIEL
2 ^{ème} année
Page 2 sur 3

Exercice 2 : Analyse des journaux de connexion

Vous avez un serveur qui enregistre les connexions des utilisateurs avec leur adresse IP et le moment exact de la connexion. Les informations sont stockées dans une liste de tuples, chaque tuple contenant l'adresse IP et l'heure de connexion.

Écrire un programme en python qui réalise les tâches suivantes :

1. **Identifier les adresses IP qui se sont connectées plus de 3 fois.**

Afficher chaque adresse IP avec le nombre total de ses connexions si elle s'est connectée plus de 3 fois.

2. **Trouver l'adresse IP qui s'est connectée en premier et celle qui s'est connectée en dernier.**

Afficher l'adresse IP avec la première et la dernière connexion basées sur l'horodatage.

3. **Compter le nombre total d'adresses IP uniques qui se sont connectées au serveur.**

Afficher le nombre total d'adresses IP distinctes ayant accédé au serveur.

Code de départ :

```
logs = [
    ('192.168.1.1', '2024-09-12 12:15:10'),
    ('192.168.1.2', '2024-09-12 12:16:15'),
    ('192.168.1.1', '2024-09-12 12:20:20'),
    ('192.168.1.3', '2024-09-12 12:30:25'),
    ('192.168.1.1', '2024-09-12 12:35:30')
]
```

Résultats attendus :

IP connectées plus de 3 fois :

Première connexion : ('192.168.1.1', '2024-09-12 12:15:10')
Dernière connexion : ('192.168.1.1', '2024-09-12 12:35:30')

Nombre total de connexions uniques : 3

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°7 : Les listes

BTS CIEL
2 ^{ème} année
Page 3 sur 3

Exercice 3 : Détection des adresses IP suspectes

Vous travaillez sur un système de surveillance réseau qui identifie les adresses IP suspectes se connectant à un serveur. Vous disposez de deux listes :

1. Une **liste noire** (blacklist) contenant les adresses IP suspectes.
2. Une **liste des connexions** (ips_connected) contenant les adresses IP qui se sont connectées au serveur.

Objectifs :

1. **Détection des adresses IP suspectes :**

Comparez la liste des adresses IP connectées au serveur avec la liste noire des adresses IP suspectes.

Affichez les adresses IP qui se trouvent à la fois dans la liste noire et dans la liste des connexions.

Code de départ :

```
blacklist = ['192.168.1.5', '10.0.0.7', '172.16.0.10']
ips_connected = ['192.168.1.1', '192.168.1.5', '192.168.1.10', '10.0.0.7']
```

Résultats attendus :

```
IP suspectes détectées :
192.168.1.5
10.0.0.7
```

```
Action suggérée : Bloquer ces IPs.
```