
Examen

Consignes

Au plus tard à la fin de l'heure prévue d'examen, vous devez avoir déposé un dossier de réponses correctement nommé (voir consigne partie 1) dans le dossier correspondant à votre session-examen dans le dossier correspondant à votre local dans le dossier Exa_HomeDir.

Par exemple pour Bob Smith en IR1A passant l'examen au L001 sur la session 42, un dossier « TIR121_IR1A_BobSmith » se trouvera dans « Exa_HomeDir > L001 > IE-L001-42 ».

Sans un dossier correctement nommé au bon endroit et dans les temps, l'examen sera considéré comme non présenté.

Lors de l'examen, vous pouvez vous munir d'une clé USB contenant votre projet de base (et rien d'autre), d'une (et une seule) feuille A4 de notes manuscrites (recto-verso) ainsi que de feuilles de brouillon (vierges).

Grille de cotation

1) Pytest projet de base :

- KO : note finale = 0/20 (le reste n'est pas corrigé)
- OK : note finale = 3/20 + points examen

2) Pylint tout :

- Fautes graves : note finale = 1/20 (le reste n'est pas corrigé)
- Fautes légères : -0.5/ fautes (minimum 1/20)
- Aucune faute : +1

3) Pytest examen :

- Q1 OK : +8
- Q2 OK : +5
- Qbonus OK : +3

Syslog et auth.log

Pour le projet de base, nous utilisons un fichier de log appelé « syslog ». Pour l'examen, nous allons en utiliser un second : « auth.log ». Sa structure est identique à syslog. Pour rappel, voici à quoi peut ressembler une ligne de ce fichier :

Oct 28 10:31:27 kali systemd[1]: Starting Login Service...

Plusieurs informations sont présentes sur cette ligne :

- La **date** (avec le mois écrit en 3 lettres (en anglais) et le jour en 2 chiffres)
- L'**heure**,
- Le **hostname**,
- Le **programme**,
- L'**id du processus** (pas toujours présent)
- Un **message explicatif** (peut être composé de sous messages séparés par des « : »)

Pour la suite, nous parlerons parfois de ligne de log « bien formée », il s'agit d'une ligne respectant cette syntaxe.

Ajoutez le fichier « auth.log » à votre dossier de log.

Questions

1. Les mises en garde du NetworkManager

Un des programmes émettant des logs dans le fichier « syslog » est le Network Manager. Comme son nom l'indique, il gère le réseau. Il lui arrive d'émettre des mises en garde dans le message et on voudrait pouvoir savoir quand c'est arrivé.

Écrivez donc la fonction `network_message_times` qui retourne une liste (sans doublon) des moments (date et heure complète) où le « NetworkManager » a émis un « <warn> ».

```
def network_message_times(logs):
    """ Pre :
        - logs est une liste où chaque élément est une ligne de log bien formée
    Post :
        - retourne une liste des moments (date et heure) sans doublon
          où le programme NetworkManager a émis un "<warn>"
    """
```

2. Les programmes avec process id

Écrivez la fonction `programs_with_process_id` qui prend en paramètre une liste de logs bien formés et retourne la liste des programmes qui ont des process id.

```
def programs_with_process_id(logs):
    """ Pre :
        - logs est une liste où chaque élément est une ligne de log bien formée
    Post :
        - retourne une liste (sans doublon) des programmes avec process id
          qui sont présents dans logs
    """
```

Bonus : les sessions de kali

Le fichier « auth.log » prend entre autres note des sessions qui sont ouvertes (via le message « session opened for user ... ») et lesquelles sont fermées (via le message « session closed for user ... »). En vous aidant des différentes fonctions que vous avez écrites pour le projet, trouvez combien de session ROOT ont été ouvertes avec un sudo et n'ont pas été fermées. Ecrivez la fonction « bonus » qui ne prend aucun paramètre et retourne simplement la réponse.

```
def bonus():
    """ retourne la réponse à la question bonus """
```

PS : évidemment, aucun test unitaire n'est fourni pour cette question 😊