

## Solution à : Forensic\_et\_Mat\_1\_2

Une alerte n'a pas été déclenchée sur activité suspecte : un élément a dû se glisser dans le fichier des événements.

Analyse du déroulement passé : l'attaquant a utilisé une stratégie de dissimulation dans un nom de tâche planifiée d'un journal d'événements, insérant un flag en direct dans le champ *TaskName*, au lieu de l'exécutable ou d'un log.

Un fichier journal Windows EVT\_X est analysé à la recherche d'un flag au format 404CTF{...}, qui est extrait automatiquement s'il est trouvé, en le loguant dans un fichier généré : flag\_detection\_log.txt

Étape 1 : Ouverture du fichier log, un fichier texte pour écrire le compte rendu de l'analyse.

Étape 2 : Lecture du fichier EVT\_X, utilisant la bibliothèque python-evtx pour parcourir les événements du journal.

Étape 3 : Parsing de type XML puis extraction, en transformant chacun des événements en un objet XML utilisable.

Étape 4 : Récupération des champs utiles, en retrouvant l'ID d'événement, la date/heure et les données associées.

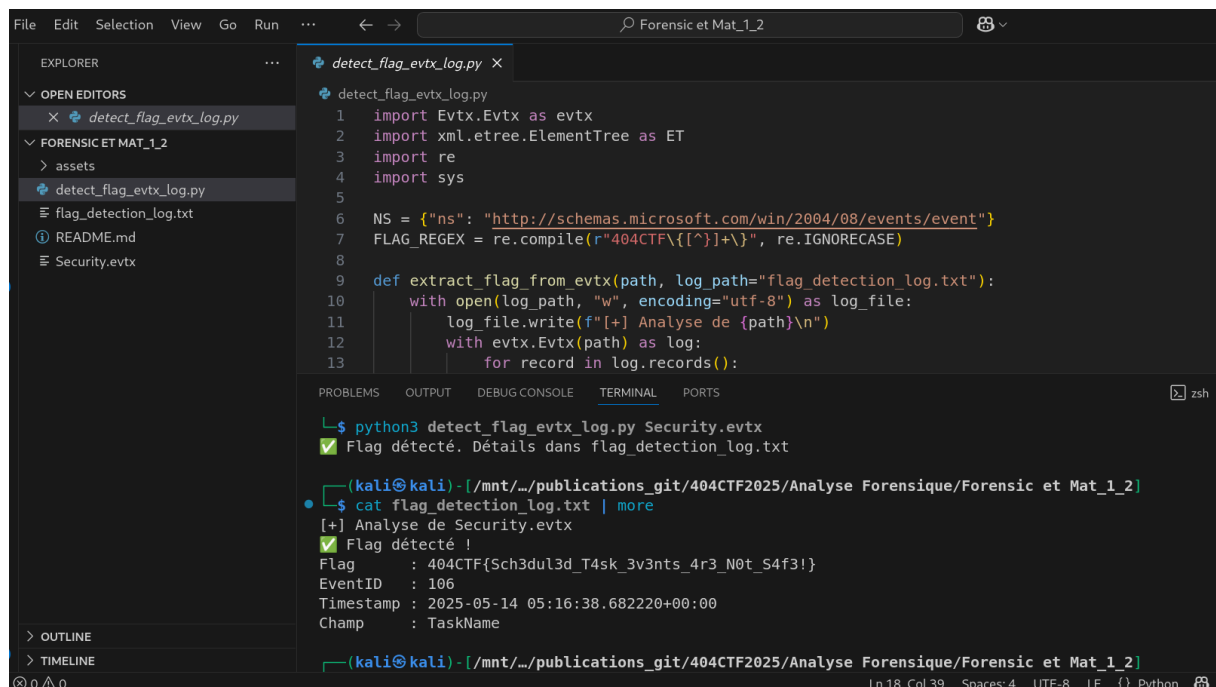
Étape 5 : Recherche de flag, parcourant tous les champs de EventData, en cherchant une chaîne qui y correspond.

Si le flag est trouvé :

- Écrire les infos dans le log (flag, EventID, Timestamp, champ concerné)
- Afficher un message en console
- Arrêter le script immédiatement (exit(0))

Si aucun flag n'est trouvé :

- Après avoir analysé tous les événements : indiquer qu'il n'a rien été trouvé.



```
File Edit Selection View Go Run ... < -> Forensic et Mat_1_2

EXPLORER
  OPEN EDITORS
    X detect_flag_evtx_log.py
  FORENSIC ET MAT_1_2
    > assets
    + detect_flag_evtx_log.py
    flag_detection_log.txt
    README.md
    Security.evtx

detect_flag_evtx_log.py
1 import Evtx.EvtX as evtx
2 import xml.etree.ElementTree as ET
3 import re
4 import sys
5
6 NS = {"ns": "http://schemas.microsoft.com/win/2004/08/events/event"}
7 FLAG_REGEX = re.compile(r"404CTF\[^\]\+\}", re.IGNORECASE)
8
9 def extract_flag_from_evtx(path, log_path="flag_detection_log.txt"):
10     with open(log_path, "w", encoding="utf-8") as log_file:
11         log_file.write(f"[+] Analyse de {path}\n")
12         with evtx.EvtX(path) as log:
13             for record in log.records():
```

```
zsh
└─$ python3 detect_flag_evtx_log.py Security.evtx
[+] Flag détecté. Détails dans flag_detection_log.txt

(kali@kali) - [/mnt/.../publications_git/404CTF2025/Analyse Forensique/Forensic et Mat_1_2]
└─$ cat flag_detection_log.txt | more
[+] Analyse de Security.evtx
[+] Flag détecté !
Flag      : 404CTF{Sch3dul3d_T4sk_3v3nts_4r3_N0t_S4f3!}
EventID   : 106
Timestamp : 2025-05-14 05:16:38.682220+00:00
Champ     : TaskName
```

L'issue est validée avec ce flag : 404CTF{Sch3dul3d\_T4sk\_3v3nts\_4r3\_N0t\_S4f3!}