ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича***»***

(СПБГУТ)

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Факультет ИКСС

Кафедра ЗСС

Курсовая работа

по дисциплине:

**«Цифровая криминалистика»**

на тему:

**«Определение последовательности действий атакующего путем анализа логов»**

Выполнили:

Студенты гр. ИКТБ-98м

Зеличенок И.Ю.

Щипцов Д.И.

Проверила:

Федорченко Е.В

Санкт-Петербург

2020г.

**Оглавление**

[Задание на курсовую работу 3](#_Toc62062973)

[Подзадачи 3](#_Toc62062974)

[Обзор логов 4](#_Toc62062975)

[Описание программы 7](#_Toc62062976)

[Листинг программы 7](#_Toc62062977)

# **Задание на курсовую работу**

Определить последовательность действий атакующего путем анализа логов событий Windows и оповещений Suricata.

# **Подзадачи**

1. Скачать логи Win Events и Suricata\_alert с сайта: <http://mirrors.rit.edu/cptc/2018/t1/> .
2. Определить последовательность действий команды, сопоставив события Windows и алерты suricata
3. Проанализировать действия команды и предположить уровень ее квалификации.

# **Обзор логов**

Логи предстают перед нами в следующем виде (Рис. 1, 2):

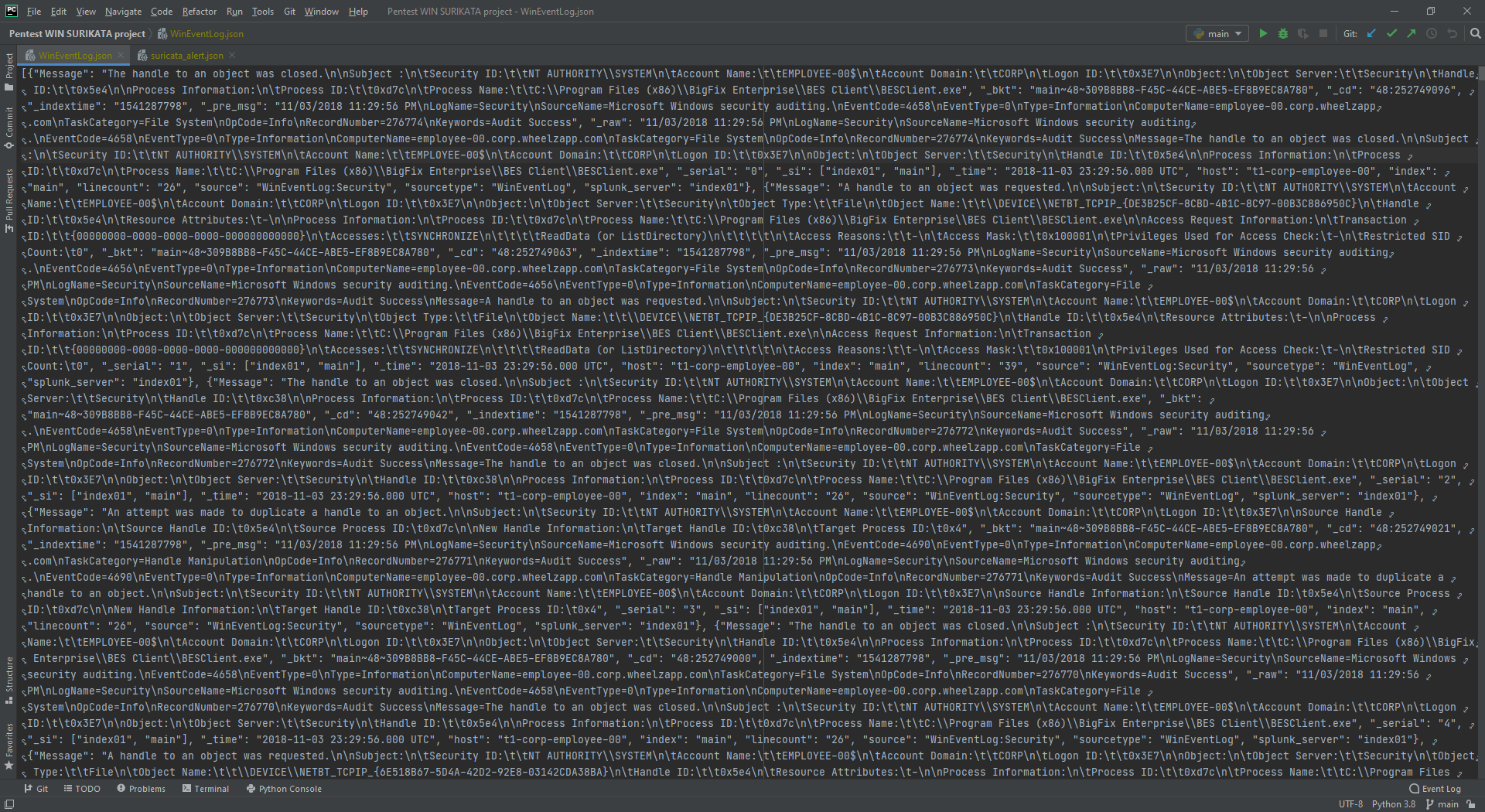


Рис. 1 Данные внутри файла WinEventLog.json.

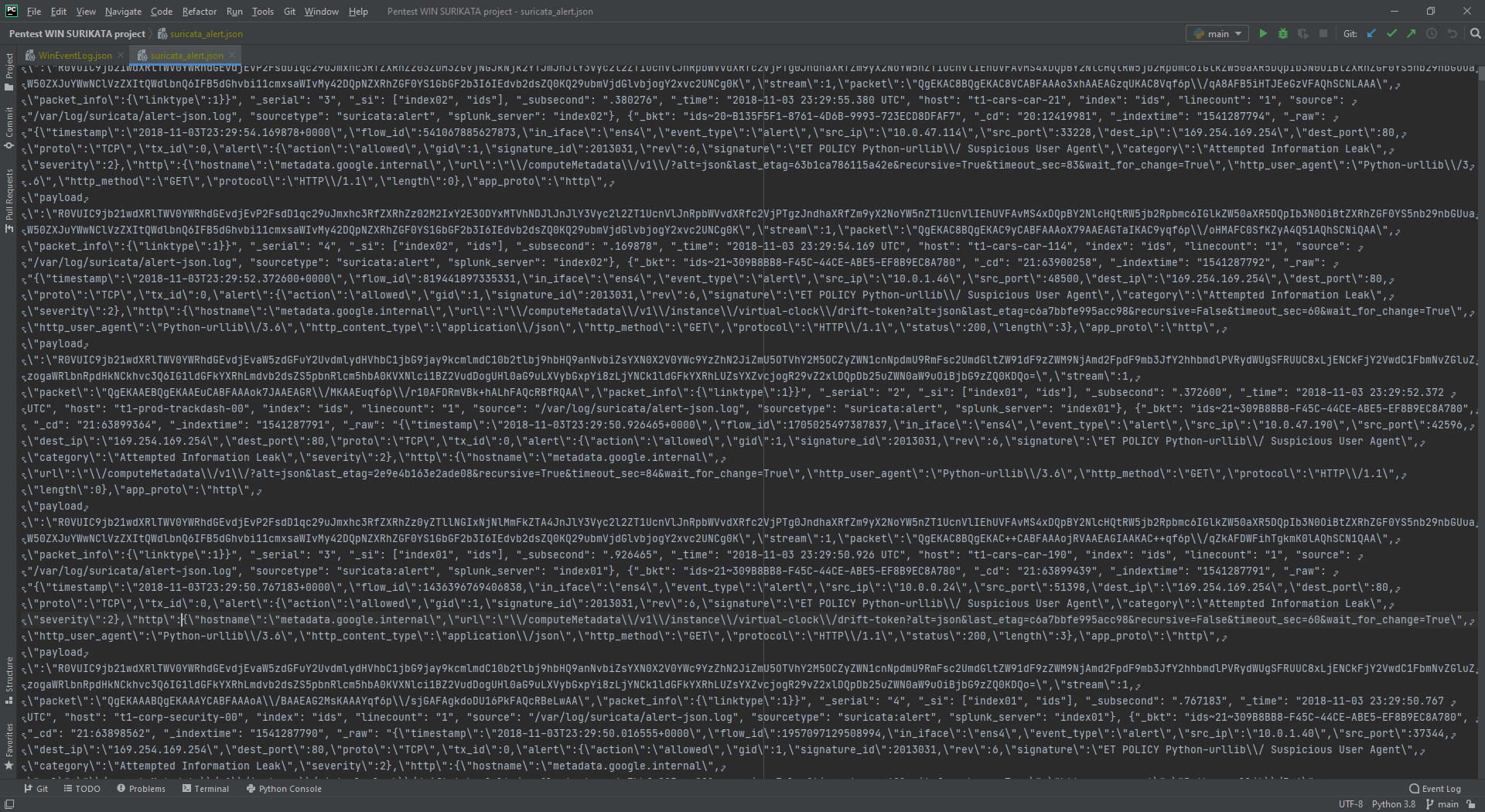


Рис. 2 Данные внутри файла suricata\_alert.json.

Для дальнейшего анализа, данные были приведены в читаемый вид средствами отладки используемого интерпретатора.

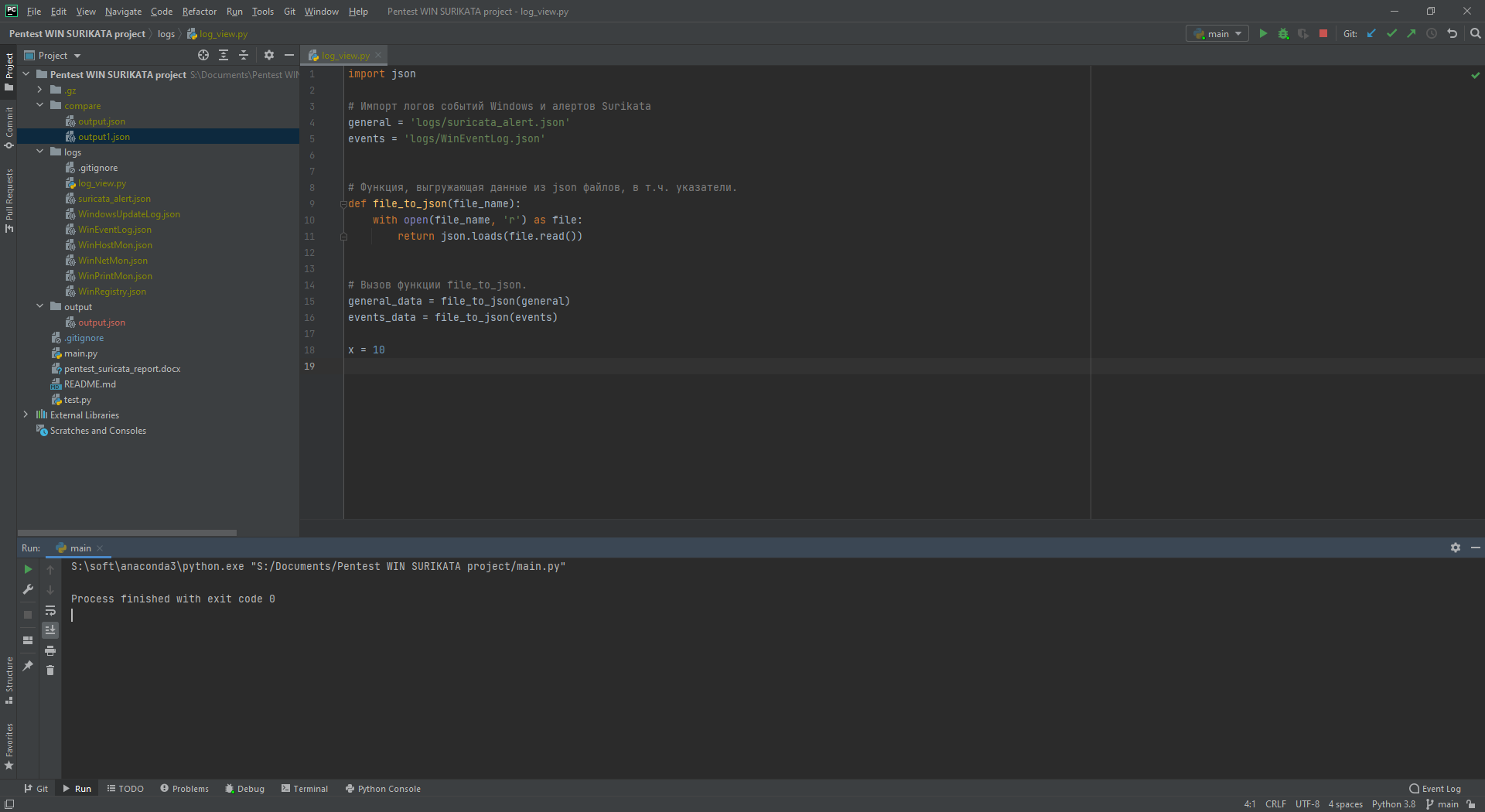


Рис. 3 Код для первичного анализа логов.

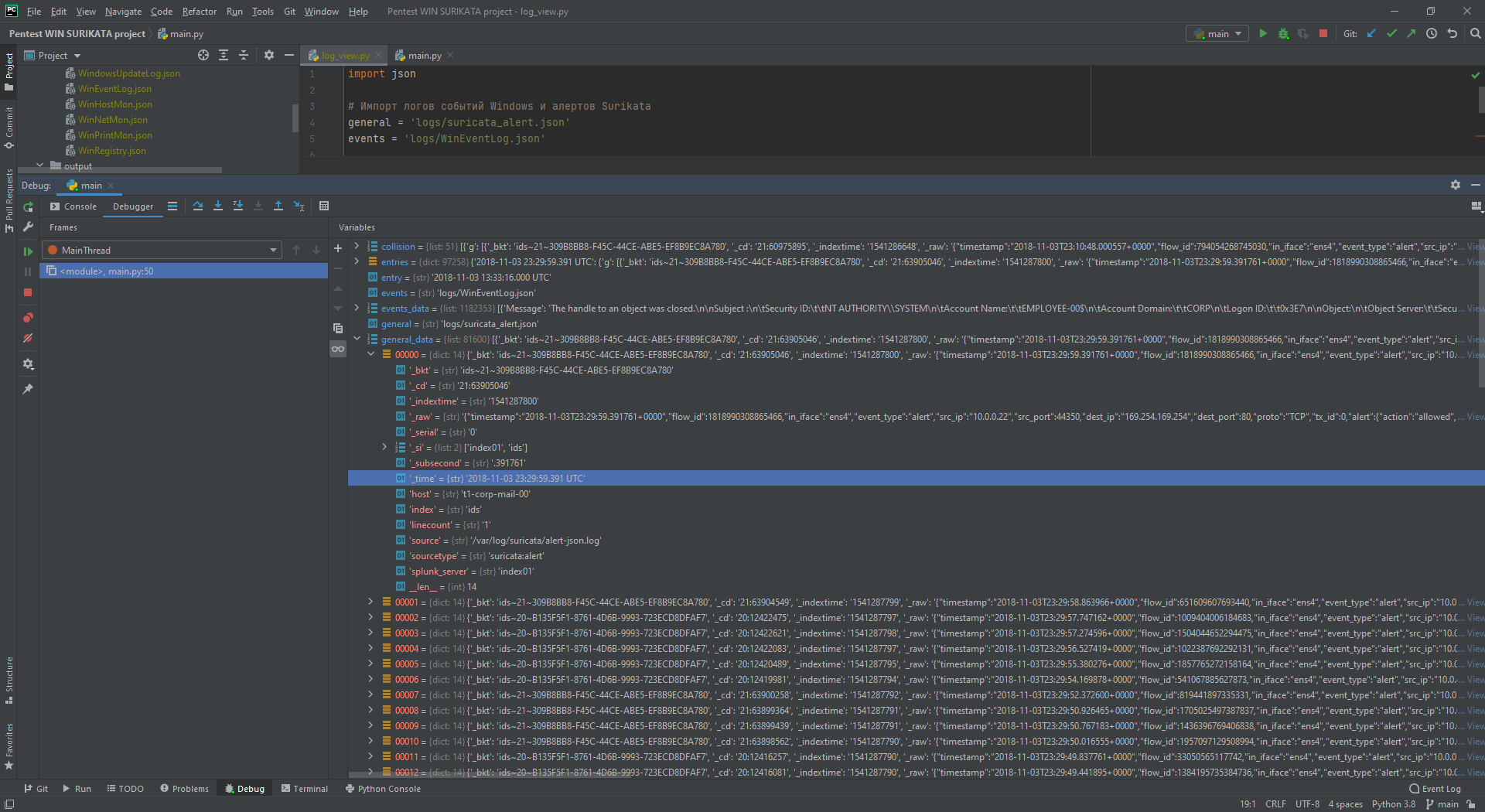


Рис. 4 Выявление ключа, за которым закреплено значение времени сообщения suricata.

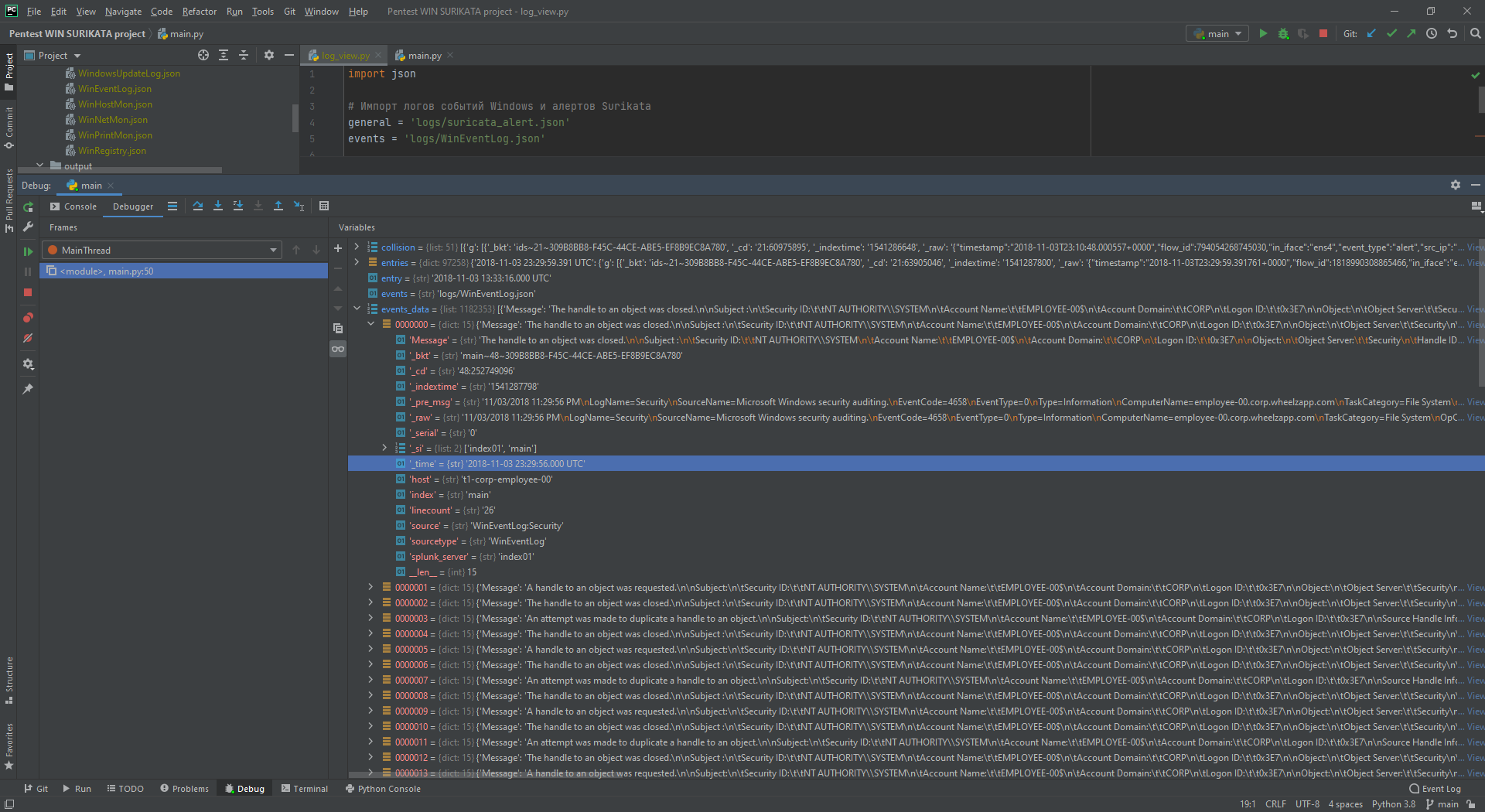


Рис. 5 Выявление ключа, за которым закреплено значение времени сообщения событий Windows.

На Рис. 4 и Рис. 5 видно, что ключом, отвечающим за передачу времени в сообщениях Suricata и Windows events является один и тот же идентификатор \_time, по которому и был проведен анализ.

# **Описание программы**

Для дальнейшей работы был разработан скрипт, который сканировал каждый лог на соответствие событий Windows и оповещений Suricata по одинаковой временной метке. Затем эти данные были выведены в отдельный файл под названием output.json.

Скрипт был написан на Python 3.8, использована среда разработки от Jet Brains под названием PyCharm. Из дополнительных библиотек: JSON.

# **Листинг программы**

Далее приведен исходный код программы и структура файла output.json (Рис. 6). В выводе за парами сообщений закреплены ключи ‘g’ и ‘e’, где ‘g’ – сообщение Suricata, а ‘e’ – соответствующие этому сообщению события Windows.

import json  
  
# Импорт логов событий Windows и алертов Surikata  
general = 'logs/suricata\_alert.json'  
events = 'logs/WinEventLog.json'  
  
  
# Функция, выгружающая данные из json файлов, в т.ч. указатели.  
def file\_to\_json(file\_name):  
 with open(file\_name, 'r') as file:  
 return json.loads(file.read())  
  
  
# Скан файлов по времени на наличие соответствий.  
def map\_to\_entries(output, json, sublist\_name):  
 for item in json:  
 time = item['\_time']  
  
 if time not in output:  
 output[time] = {}  
  
 if sublist\_name not in output[time]:  
 output[time][sublist\_name] = []  
  
 output[time][sublist\_name].append(item)  
  
  
# Вызов функции file\_to\_json.  
general\_data = file\_to\_json(general)  
events\_data = file\_to\_json(events)  
  
  
# Вызов функции map\_to\_entries, запись всех событий в единый словарь entries по меткам 'g' и 'e'.  
entries = {}  
map\_to\_entries(entries, general\_data, 'g')  
map\_to\_entries(entries, events\_data, 'e')  
  
  
# Выявление пар алертов и событий, внесение их в массив collision.  
collision = []  
  
for entry in entries:  
 item = entries[entry]  
  
 if 'g' in item and 'e' in item:  
 collision.append(item)  
  
  
# Запись в файл вывода.  
with open('output/output.json', 'w') as out\_file:  
 out\_file.write(json.dumps(collision, indent=2))

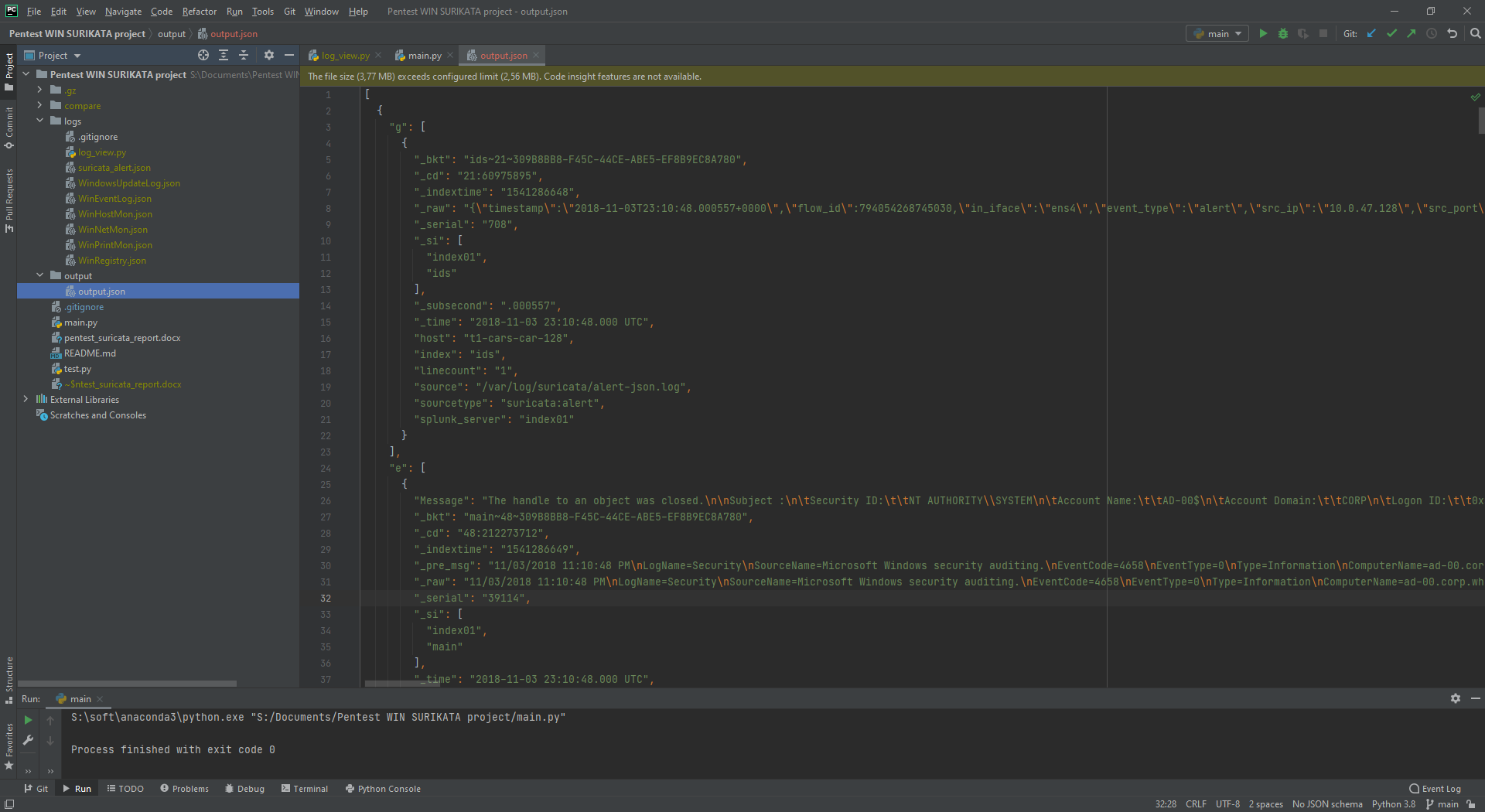


Рис. 6 Структура файла output.json.

# **Анализ логов**