بسم الله الرحمن الرحيم

تقرير

CyberSecurity

CyberTool: JSON to CSV

الاسم : حسين طلال سعيد الوجيه

التخصص : الامن السيبراني

ID : 2023110222

تاريخ انشاء المشروع : ٢٠٢٥\٩\١٤

❖ الخلاصة(Abstract)

تُقدم هذه الأداة، "محوّل JSON إلى"CSV ، حلاً فعالاً لتحويل البيانات المهيكلة بين تنسيق JSONوتنسيق .CSV تهدف الأداة إلى تبسيط عملية إدارة البيانات التي غالبًا ما تكون ضرورية في سياق الأمن السيبراني، مثل تحليل سجلات النظام (logs) أو بيانات الشبكة. من خلال توفير واجهة بسيطة وواضحة، تُمكن هذه الأداة المحترفين والباحثين من معالجة البيانات بسهولة وفعالية، مما يسهل عمليات التحليل والمراجعة اللاحقة.

❖ المقدمة(Introduction)

في عالم الأمن السيبراني، تعتبر البيانات أصلًا أساسيًا. ومع ذلك، غالبًا ما تأتي هذه البيانات بتنسيقات مختلفة وغير متوافقة، الذي يسهل تحليله في برامج جداول CSVو (APIs) الذي يُستخدم بشكل شائع في واجهات برمجة التطبيقات JSON مثل البيانات. تكمن المشكلة في أن عدم القدرة على تحويل هذه البيانات بكفاءة يمكن أن يعيق عملية التحليل والاستجابة السريعة للتهديدات.

تم تصميم هذه الأداة لمعالجة هذه المشكلة من خلال توفير حل موثوق ."CSV إلى JSON هنا يأتي دور أداة "محوّل ومرن لتحويل البيانات بين هذين التنسيقين. يمكن تطبيق الأداة في سيناريوهات متعددة، مثل تحويل بيانات السجلات من ، أو حتى تحويل التقارير الإحصائية (IDS) من نظم كشف التسلل (incident logs) الخوادم، أو سجلات الحوادث لتسهيل عرضها وتفسيرها. من خلال تبسيط هذه العملية، تتيح الأداة للمحللين التركيز على استنتاج الرؤى الأمنية المهمة بدلاً من قضاء الوقت في التعامل مع تنسيقات البيانات غير المتوافقة.

❖ المنهجية(Methodology)

تم تطوير أداة التحويل بناءً على منهجية منظمة لضمان الكفاءة والوثوقية. تعتمد الأداة بشكل أساسي على **لغة بايثون** واستخدام مكتبات قياسية وموثوقة، مما يقلل من التعقيد ويضمن قابلية التوسع.

1. **المكتبات المستخدمة:**
   - **json:** تُعد هذه المكتبة جزءًا أساسيًا من بايثون وتُستخدم لتحليل وقراءة البيانات بتنسيق JSON، مما يسمح بمعالجة هياكل البيانات المعقدة بفعالية.
   - **csv:** تُستخدم هذه المكتبة لمعالجة ملفات CSV بتنسيق توفر دوالًا قوية لقراءة وكتابة البيانات، حيث جدول بيانات، مما يسهل تحويل البيانات من وإلى هذا التنسيق.
   - **pyfiglet:** تُستخدم هذه المكتبة لإنشاء فن ASCII، مما يُضفي لمسة احترافية وجمالية على واجهة المستخدم في سطر الأوامر.
   - **traceback:** تُستخدم هذه الوحدة لتوفير تتبع مفصل لأخطاء وقت التشغيل، مما يساعد في تصحيح الأخطاء وتقديم رسائل واضحة للمستخدم.

```
import json
import csv
import pyfiglet
import traceback
```

1. **سير عمل الأداة:**
   - **القراءة والتحقق:** تبدأ الأداة بطلب مسار ملف الإدخال من المستخدم. تُستخدم دوال مخصصة مثل `read_json` و`read_csv` للتحقق من وجود الملف وصحة تنسيقه. إذا كان الملف فارغًا أو تالفًا، تُقدم الأداة رسالة خطأ واضحة للمستخدم.
   - **المعالجة الديناميكية:** بعد قراءة البيانات، تقوم الأداة بتحديد رؤوس الأعمدة (headers) ديناميكيًا من مفاتيح البيانات الموجودة في ملف JSON.
   - **الكتابة والتحويل:** تُستخدم دوال الكتابة لكتابة البيانات المعالجة في ملف الإخراج الجديد بالتنسيق المطلوب (CSV أو JSON).

```
if __name__=="__main__":
    PrintFun()
    while True:
        number_transfer = input("Choose a number  : ")

        if number_transfer == '1' :

            file_path = input("Enter the name of the file you want to convert with an extension  :")
            converter = JsonTocsvConverter(file_path)

            if converter.read_json():
                csv_file_path = input("Enter the name of the new csv file (with .csv extension )  :")
                #التحقق من امتداد الملف
                if not csv_file_path.endswith('.csv'):
                    csv_file_path += '.csv'

                converter.convert_to_csv(csv_file_path)
            break

        elif number_transfer == '2' :

            csv_file_path = input("Enter the name of the csv file to convert(with  extension )  :")
            # file_path = input("Enter the name of the new json file (with .json extension )  :")
            converter = JsonTocsvConverter(csv_file_path)
            if converter.read_csv():
                json_file_path = input("Enter the name of the new JSON file (without extension)  :")
                #التحقق من امتداد الملف
                if not json_file_path.endswith('.json'):
                    json_file_path += '.json'

                # استخدام نفس الاوبجكت لتحويل البيانات
                converter.convert_to_json(csv_file_path,json_file_path)
            break

        elif number_transfer == '3' :
            print("Exit .............")
            break
        else :
            print(f"[{number_transfer}]: invalid number.. ")
```

```
def PrintFun():
    text = "                JSON to CSV"
    text2= "            Dr.Sundus              Eng.HUSSIEN"
    ascii_art = pyfiglet.figlet_format(text)
    ascii_art2 = pyfiglet.figlet_format(text2)
    print(ascii_art)
    print(ascii_art2)
    print("-------------------------------------({ Hello My D.Sundus })-------------------------------------")
    print("Select the transfer number : ")
    print("----------------")
    print("[01] Json to Csv |")
    print("----------------")
    print("----------------")
    print("[02] CSV to Json |")
    print("----------------")
    print("----------")
    print("[03] Exit |")
    print("----------")
```

❖ الخوارزمية وتصميم الكود (Algorithm &

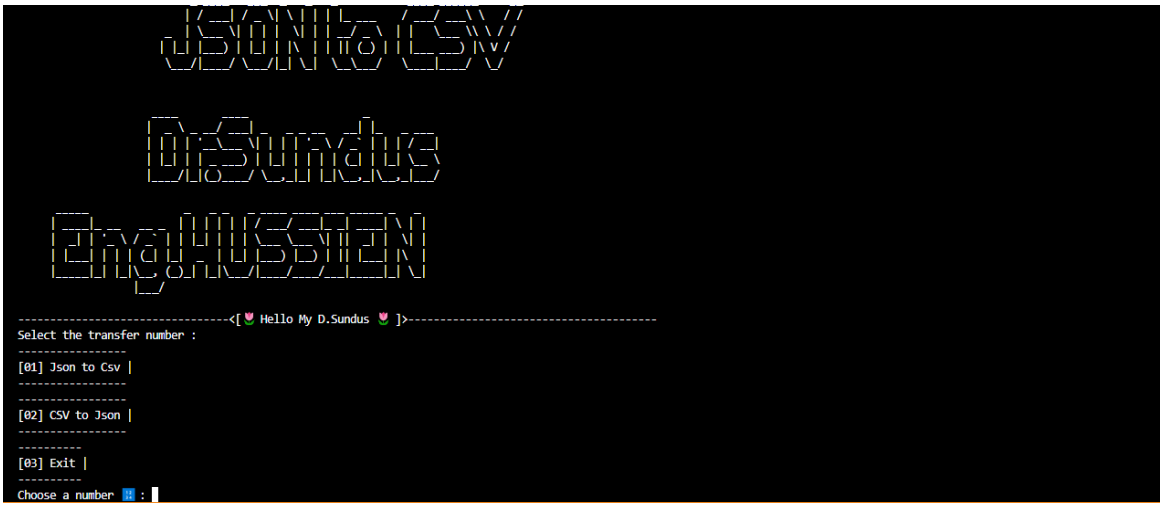
يعتمد تصميم الكود على نهج **البرمجة الشيئية(Object-Oriented Programming - OOP)** ، حيث تم تنظيم الأداة داخل كلاس واحد يسمى `JsonTocsvConverter`. هذا التصميم يجعل الكود منظمًا، قابلًا لإعادة الاستخدام، وسهل الصيانة.

```python
# كلاس الاداة كامل
class JsonTocsvConverter:
# هذه الدالة عرفنا فيها المتغيرات

    def __init__(self,file_path):
        self.file_path = file_path
        self.data = None
```

منطق الأداة Logic)

١_ **الواجهة الرئيسية :** عند تشغيل البرنامج، تُعرض قائمة رئيسية باستخدام دالة `PrintFun()`, مما يسمح للمستخدم باختيار العملية المطلوبة) تحويل من JSON إلى CSV أو العكس.(



٢_التحقق من الملف

- لتحويل JSON إلىCSV ، يتم استدعاء `read_json()`. هذه الدالة تتحقق من أن الملف يحتوي على قائمة من القواميس.

```python
def read_json(self):
    try :
        with open(self.file_path,'r',encoding='utf-8') as file :
            loaded_date = json.load(file)

        if isinstance(loaded_date,list):
            self.data = loaded_date

        elif isinstance(loaded_date,dict):
            list_found = False
            for value in loaded_date.values():
                if isinstance(value,list) and value:
                    self.data = value
                    list_found = True
                    break

            if not list_found:
                print("Error : The json file does not contain a list of data to convert.")
                return False

        else:
            print("Error : The json file format is not a list or dictionary of data .")
            return False

        print("The file has been uploaded successfully ^_^ ")
        return True

    except FileNotFoundError:
        print(f"Error : File not found ==> {self.file_path}")
        return False
    except json.JSONDecodeError:
        print(f"Error : In File formatting ==> {self.file_path}")
        return False
    except Exception as e:
        print("An unexpected Error occurred ==> {e}")
        traceback.print_exc()
        return False
```

- لتحويل CSV إلىJSON ، يتم استدعاء `read_csv()`. هذه الدالة تتحقق من أن الملف يحتوي على بيانات يمكن قراءتها

```python
def read_csv(self):
    try:
        with open(self.file_path, 'r', encoding='utf-8') as file:
            csv_reader = csv.DictReader(file)
            self.data = [row for row in csv_reader]

        if not self.data:
            print("Error: The CSV file is empty or contains no data.")
            return False

        print("The CSV file has been uploaded successfully ^_^")
        return True

    except FileNotFoundError:
        print(f"Error: CSV file not found ==> {self.file_path}")
        return False

    except Exception as e:
        print(f"An unexpected Error occurred while reading the CSV file: {e}")
        traceback.print_exc()
        return False
```

٣_التحويل

- دالة `convert_to_csv()`تستخدم `csv.DictWriter`لكتابة البيانات. تقوم بجمع جميع المفاتيح الموجودة في بيانات JSON لإنشاء صف الرؤوس في ملف.CSV

```python
def convert_to_csv(self,csv_file_path):

    if not self.data:
        print("Error : Empty json file or no data to convert ")
        return False

    try:
        headers = set()
        for row in self.data:
            if isinstance(row, dict):
                headers.update(row.keys())
            else:
                print(f"Skipping non-dictionary data: {row}")

        headers = list(headers)

        with open(csv_file_path,'w',newline='',encoding='utf-8') as file:
            csv_writer = csv.DictWriter(file,fieldnames=headers,restval='')
            csv_writer.writeheader()
            for row in self.data:
                csv_writer.writerow(row)

        print(f"The file has been successfully converted and saved in :{csv_file_path}")
        return True
    except Exception as e:
        print(f"An Error occurred while writing the csv file {e}")
        traceback.print_exc()
        return False
```

- دالة `convert_to_json()`تقرأ البيانات من ملف CSV باستخدام `csv.DictReader`ثم تستخدم `json.dump`لكتابة البيانات في ملف JSONمع تنسيق مناسب (indentation) لسهولة القراءة.

```python
def convert_to_json(self,csv_file_path,json_file_path):
    data = []
    try:
        with open(csv_file_path,'r',encoding='utf-8')as file:
            csv_reader = csv.DictReader(file)
            for row in csv_reader:
                data.append(row)

        if not data:
            print("Error: The CSV file is empty or contains no data.")
            return False


        with open(json_file_path,'w',encoding='utf-8')as file:
            json.dump(data,file,indent=4)

        print(f"The file has been successfully converted and saved in :{json_file_path}")
        return True

    except FileNotFoundError:
        print(f"Error : csv file not found ==> {csv_file_path}")
        return False

    except csv.Error as e:
        print(f"Error: An error occurred while reading the CSV file: {e}")
        return False

    except Exception as e:
        print(f"An unexpected Error occurred : {e}")
        traceback.print_exc()
        return False
```

❖ التعامل مع المدخلات والمخرجات (Input and

تم تصميم الأداة لتكون سهلة الاستخدام وتعتمد على سطر الأوامر (terminal) لإدخال البيانات. يتم التعامل مع المدخلات والمخرجات على النحو التالي:

- **إدخال البيانات :**يزود المستخدم الأداة بأسماء ملفات الإدخال والإخراج عبر سطر الأوامر. على سبيل المثال، يطلب البرنامج من المستخدم إدخال اسم ملف JSON للتحويل، ثم يطلب اسم ملف CSV للحفظ.
- **معالجة الملفات :**الأداة قادرة على قراءة الملفات من نوع **JSON**و **CSV**، مما يسمح لها بالتعامل مع البيانات المهيكلة التي غالبًا ما تُستخدم في سيناريوهات الأمن السيبراني مثل سجلات الأحداث أو بيانات الشبكة.
- **عرض المخرجات :**بالإضافة إلى حفظ البيانات في ملفات الإخراج، تُقدم الأداة رسائل توضيحية للمستخدم عبر سطر الأوامر لإبلاغه بحالة العملية. على سبيل المثال، تُعرض رسالة "The file has been uploaded successfully" عند قراءة الملف بنجاح، ورسالة "The file has been successfully converted"عند إتمام عملية التحويل.

❖ معالجة الأخطاء (Error

١. **التحقق من صحة المدخلات:(Input Validation)**

- **وجود الملف :**تتحقق الأداة مما إذا كان الملف المدخل من قبل المستخدم موجودًا. إذا لم يتم العثور عليه، تُعرض رسالة خطأ مثل : `Error: File not found.`
- **صيغة الملف :**تتحقق الأداة من تنسيق ملفات JSON و CSV قبل البدء في عملية التحويل. إذا كانت البيانات غير مهيكلة بشكل صحيح، تُعرض رسالة خطأ مثل.`Error: In File formatting` :

٢. **أخطاء التشغيل:(Runtime Errors)**

- تُستخدم كتل `try-except`للتعامل مع الأخطاء التي قد تحدث أثناء تشغيل البرنامج، مثل الأخطاء في الكتابة إلى ملف أو الأخطاء المتعلقة بالترميز.(encoding)

٣. **أخطاء الإخراج:(Output Errors)**

- تتحقق الأداة من وجود بيانات قابلة للتحويل. إذا كان ملف الإدخال فارغًا، تُعرض رسالة خطأ مثل `Error: Empty json file` : `or no data to convert.`

٤. يتم استخدام رسائل خطأ مفصلة ومُوجهة للمستخدم، مما يساعده في تحديد المشكلة وحلها.

❖ البرمجة الشيئية وتصميم الكلاس (OOP and

تعتمد أداة التحويل على مبادئ البرمجة الشيئية لتنظيم الكود وجعله أكثر فعالية وقابلية لإعادة الاستخدام .تم تحقيق ذلك من خلال إنشاء كلاس رئيسي يسمى

`JsonTocsvConverter`.

- **الكلاس :** **`JsonTocsvConverter`**هذا الكلاس هو القلب النابض للأداة. يحتوي على جميع الخصائص والدوال اللازمة لإدارة عملية التحويل بالكامل.

```
class JsonTocsvConverter:
```

الخصائص:(Attributes)

- `file_path`:لتخزين مسار الملف الذي يتم العمل عليه.
- `data`:لتخزين البيانات التي يتم قراءتها من الملف المصدر

```
def __init__(self,file_path):
    self.file_path = file_path
    self.data = None
```

- **الدوال:(Methods)**
  - `read_json()`: تقوم بقراءة ملف JSON وتخزين محتوياته.
  - `read_csv()`: تقوم بقراءة ملف CSV وتخزين محتوياته.
  - `convert_to_csv()`: تقوم بتحويل البيانات من JSON إلى CSV.
  - `convert_to_json()`: تقوم بتحويل البيانات من CSV إلى JSON.

```
def read_json(self):
```

```
def read_csv(self):
```

```
def convert_to_csv(self,csv_file_path):
```

```
def convert_to_json(self,csv_file_path,json_file_path):
```

فوائد تصميم الكلاس:

- **الكبسلة (Encapsulation):** يتم تجميع جميع الخصائص والدوال المتعلقة بالتحويل في كيان واحد (الكلاس)، مما يحمي البيانات ويجعل الكود أكثر تنظيمًا.
- **سهولة الصيانة:** يمكن تعديل أو تحسين أي دالة داخل الكلاس دون التأثير على الأجزاء الأخرى من البرنامج

❖ العمل المستقبلي والتوصيات (Future Work and

على الرغم من أن أداة "محوّل JSON إلى "CSV تؤدي وظيفتها الأساسية بفعالية، إلا أن هناك العديد من التحسينات التي يمكن إضافتها في المستقبل لزيادة فائدتها وقوتها. من التوصيات المقترحة:

١. **دعم تنسيقات إضافية:** يمكن توسيع الأداة لدعم تنسيقات بيانات أخرى شائعة في الأمن السيبراني مثل XML أو YAML.
٢. **واجهة رسومية للمستخدم (GUI):** يمكن تطوير واجهة رسومية باستخدام مكتبات مثل `Tkinter` أو `PyQt` لجعل الأداة أسهل في الاستخدام للمستخدمين غير التقنيين.
٣. **معالجة البيانات غير المتجانسة:** يمكن تطوير منطق الأداة للتعامل مع ملفات JSON التي تحتوي على بيانات غير متجانسة أو هياكل متداخلة.(nested structures)
٤. **تحسين الأداء:** يمكن تحسين أداء الأداة للتعامل مع الملفات الكبيرة جدًا من خلال معالجة البيانات على دفعات (batches) بدلاً من تحميلها بالكامل في الذاكرة.
٥.

❖ الخاتمة(Conclusion)

يُمثل هذا المشروع نقطة تحول في فهمي لتطبيقات البرمجة الشيئية في حل المشكلات الواقعية. من خلال تطوير أداة "محوّل JSON إلى"CSV ، تمكنت من تطبيق المفاهيم النظرية في الأمان السيبراني وتحويلها إلى أداة عملية. لقد أثبتت الأداة قدرتها على التعامل مع تحديات تنسيق البيانات بكفاءة، مما يوفر حلاً موثوقًا لتحويل البيانات بين JSON و .CSV هذا الإنجاز لا يعكس فقط الفهم التقني، بل أيضًا القدرة على تصميم أداة سهلة الاستخدام وقادرة على التعامل مع الأخطاء، مما يجعلها إضافة قيمة لأي بيئة عمل.

❖ المراجع(References)

- [google.com]: [موقع الكتروني]،
- Python Official Documentation: For modules like `json`, `csv`, and `traceback`.
- [GitHub]