OPEN SOURCE LAB - REPORT
VULNERABILITY AND MALWARE DETECTION

Name : **Rishi S**
Roll no: **16PT30**
Github link : https://github.com/rishi16pt30/OSS_LAB

Vulnerability Assessment :

A Vulnerability is a weakness or bug that exists in a software, which can be exploited by a hacker or a cybercriminal to gain access into the system. By gaining the access he can execute code or perform unauthorized access in computer systems. Vulnerability also allows hackers to install malware, steal or destroy sensitive data.

The process of identifying and prioritizing the vulnerabilities in a computer system is known as vulnerability assessment. Vulnerability assessment gives us a holistic view of the vulnerabilities and exposures that exist in the computer system system. The main aim of the project is to perform vulnerability assessment.

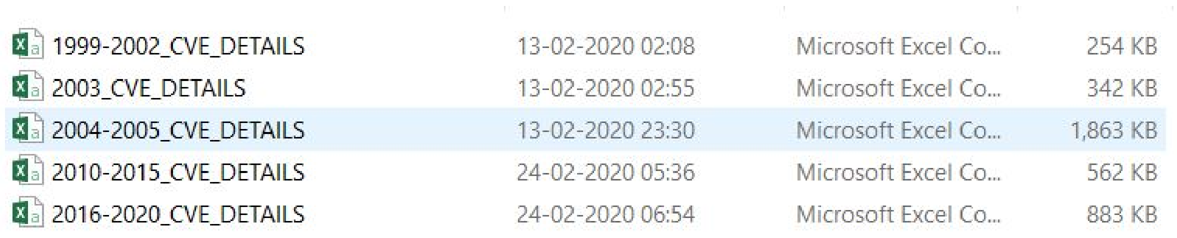
Vulnerability & Exposure Database :

A Vulnerability database is a platform which collects, evaluates and shares information about known or newly discovered vulnerabilities. MITRE runs the largest database known as CVE (Common Vulnerability and Exposure). CVE also assigns a severity score to all vulnerabilities known as CVSS score (Common Vulnerability Scoring System). A CVE listing contains a unique CVE-Id, a brief description and references related to the vulnerability, severity (scoring) of the vulnerability, type of vulnerability or exposure, and many other details. CVE does not contain the risk,impact or fix for the vulnerability.

Vulnerabilities are added to CVE databases by CNAs. CNA stands for CVE Numbering Authority. CNAs are organizations who identity and distribute CVE-Ids to researchers or IT vendors for inclusion of new vulnerabilities or exposures in the CVE database.

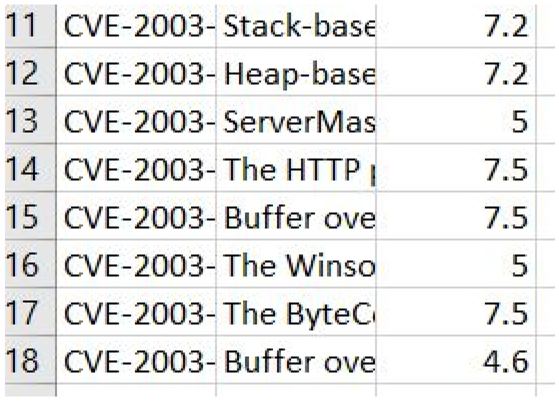
Workflow :
The data from CVE databases are scraped using beautifulsoup library in python. This data is from the year 1999 to 2020. The scraped vulnerability consists of the CVE-ID, description and the severity of the vulnerability known as CVSS scoring. The operating system vulnerabilities are also identified and scraped.

Following screenshots show the snapshot of downloaded cven details



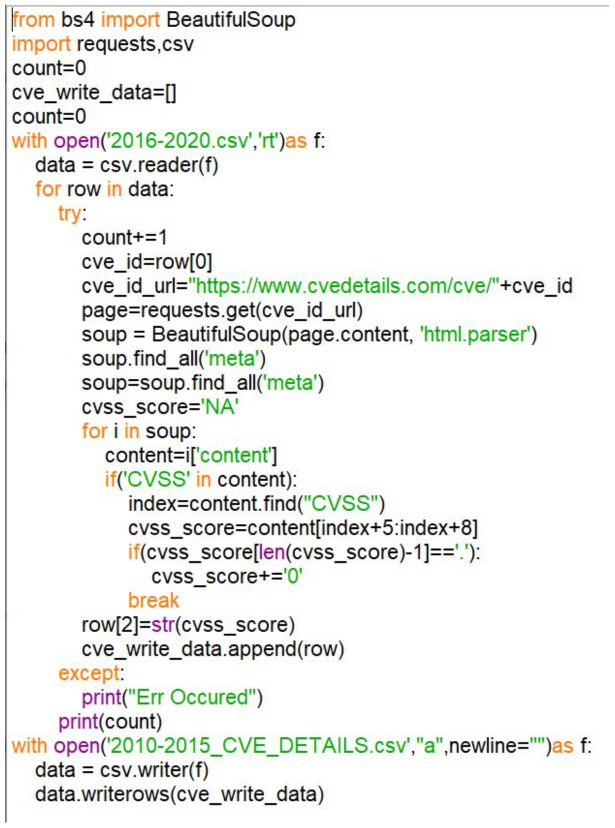
1999-2002_CVE_DETAILS	13-02-2020 02:08	Microsoft Excel Co...	254 KB
2003_CVE_DETAILS	13-02-2020 02:55	Microsoft Excel Co...	342 KB
2004-2005_CVE_DETAILS	13-02-2020 23:30	Microsoft Excel Co...	1,863 KB
2010-2015_CVE_DETAILS	24-02-2020 05:36	Microsoft Excel Co...	562 KB
2016-2020_CVE_DETAILS	24-02-2020 06:54	Microsoft Excel Co...	883 KB

Below is the snap of an excel sheet (Consists of CVE-Id, Description and score)



11	CVE-2003-	Stack-base	7.2
12	CVE-2003-	Heap-base	7.2
13	CVE-2003-	ServerMas	5
14	CVE-2003-	The HTTP	7.5
15	CVE-2003-	Buffer ove	7.5
16	CVE-2003-	The Winso	5
17	CVE-2003-	The ByteC	7.5
18	CVE-2003-	Buffer ove	4.6

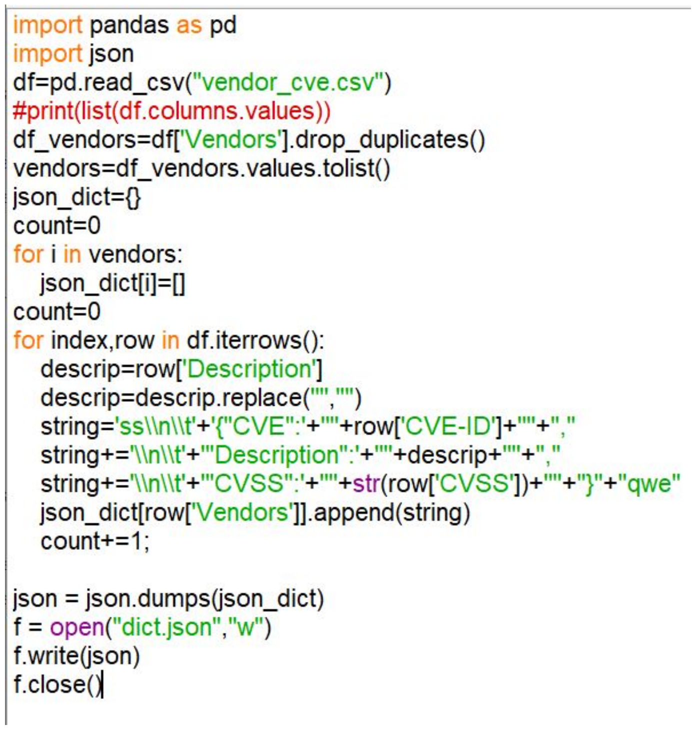
Below is the snap of code for scraping the CVE data



```
from bs4 import BeautifulSoup
import requests,csv
count=0
cve_write_data=[]
count=0
with open('2016-2020.csv','rt')as f:
    data = csv.reader(f)
    for row in data:
        try:
            count+=1
            cve_id=row[0]
            cve_id_url="https://www.cvedetails.com/cve/"+cve_id
            page=requests.get(cve_id_url)
            soup = BeautifulSoup(page.content, 'html.parser')
            soup.find_all('meta')
            soup=soup.find_all('meta')
            cvss_score='NA'
            for i in soup:
                content=i['content']
                if('CVSS' in content):
                    index=content.find("CVSS")
                    cvss_score=content[index+5:index+8]
                    if(cvss_score[len(cvss_score)-1]=='.'):
                        cvss_score+='0'
                    break
            row[2]=str(cvss_score)
            cve_write_data.append(row)
        except:
            print("Err Occured")
        print(count)
with open('2010-2015_CVE_DETAILS.csv',"a",newline="")as f:
    data = csv.writer(f)
    data.writerows(cve_write_data)
```

After downloading the data, the data in the form of spreadsheets are converted to JSON objects (JavaScript Object Notation format). This is because data in Json format is easily understandable and paves the way for faster access.

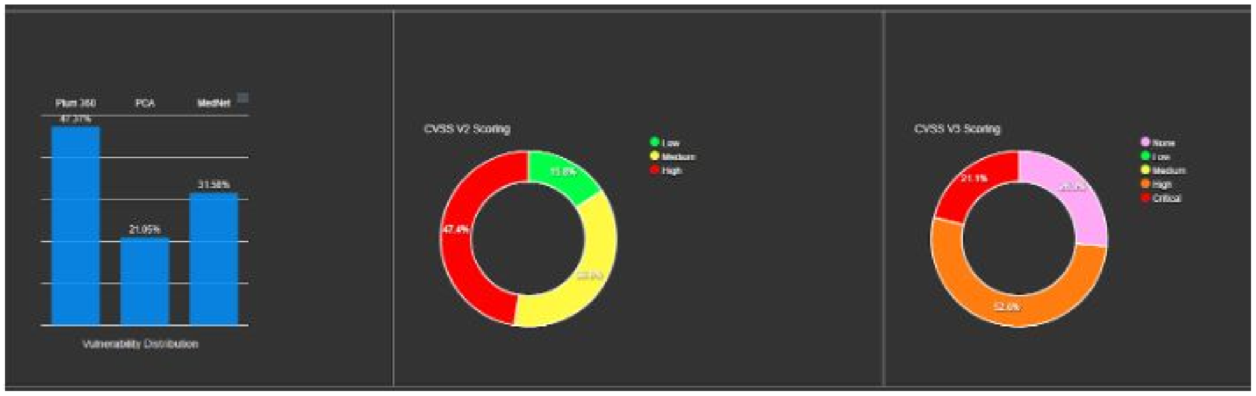
Another python script file converts the data into respective JSON format. These data are stored in NoSql database MongoDB. This is done with the help of PyMongo library in python.

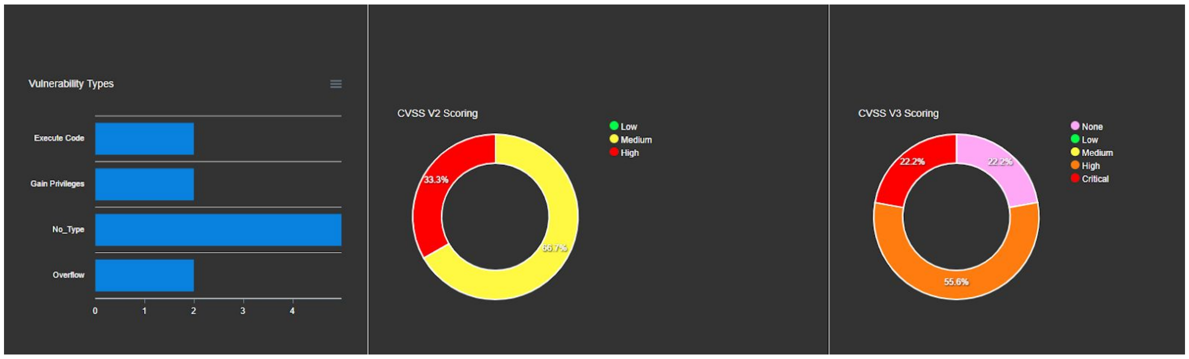


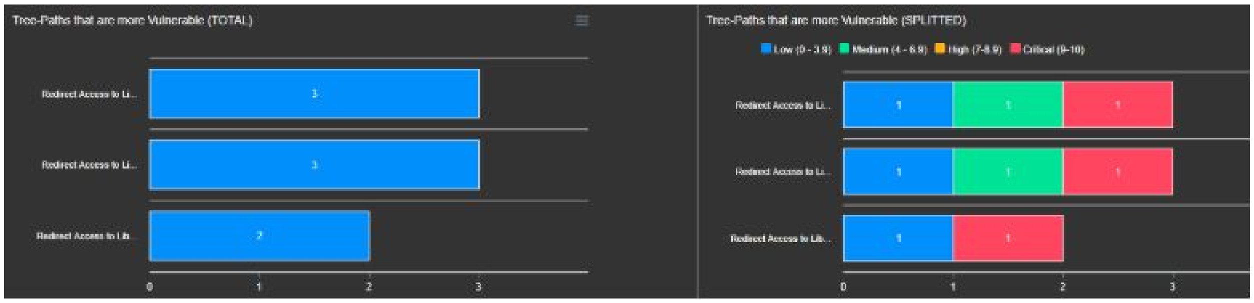
```
import pandas as pd
import json
df=pd.read_csv("vendor_cve.csv")
#print(list(df.columns.values))
df_vendors=df['Vendors'].drop_duplicates()
vendors=df_vendors.values.tolist()
json_dict={}
count=0
for i in vendors:
   json_dict[i]=[]
count=0
for index,row in df.iterrows():
   descrip=row['Description']
   descrip=descrip.replace('"',"")
   string='ss\\n\\t'+'{"CVE":'+'"'+row['CVE-ID']+'"'+","
   string+='\\n\\t'+'"Description":'+'"'+descrip+'"'+","
   string+='\\n\\t'+'"CVSS":'+'"'+str(row['CVSS'])+'"'+"}"+"qwe"
   json_dict[row['Vendors']].append(string)
   count+=1;

json = json.dumps(json_dict)
f = open("dict.json","w")
f.write(json)
f.close()
```

The Final View :

Finally the data is collected once the user executes the program, i.e., the program searches for all the installed applications and the vulnerabilities associated with them are retrieved from the database. The program also shows a data visualization of the vulnerabilities and threats in the system.







Conclusion :

The above visualizations help the user to understand the security of his/her system in a better way. This program helps the user to preserve his system from data theft or ransomware from the hackers.