SURAJ ESWARAN
CS559 QUANTITATIVE SECURITY

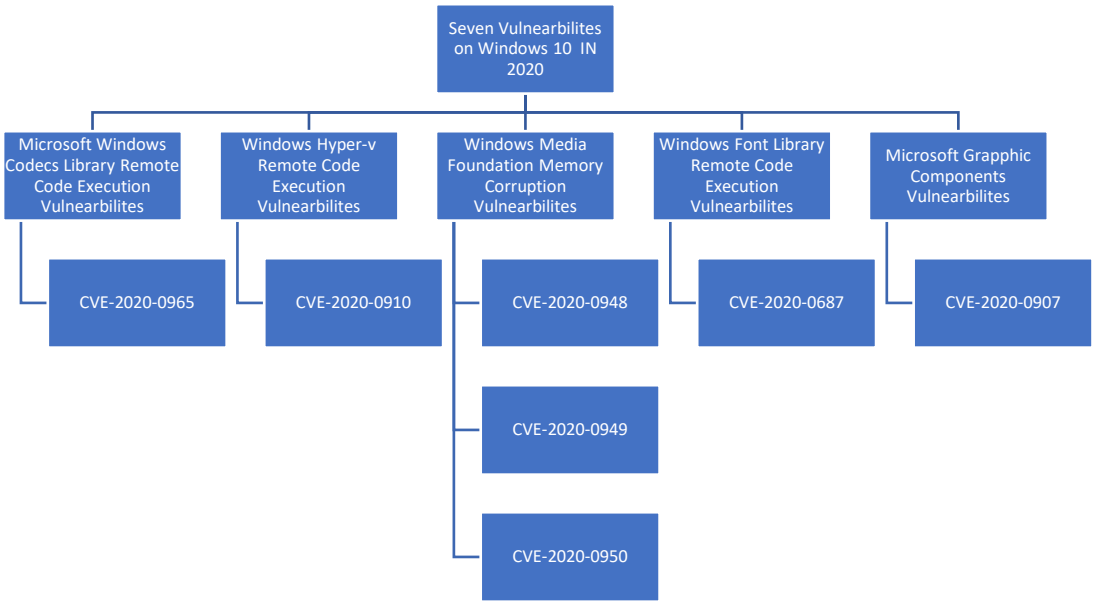
ASSIGNMENT 2: VULNERABILITY DATA EXTRACTION AND ANALYSIS

INTRODUCTION

Vulnerability Analysis involves identifying, analyzing, and prioritizing the vulnerability in a system. For this assignment ,we have to extract the data from the database, preprocess them and perform an analysis. The system that we are planning to analyze were Windows 10 and Apache HTTP Server.

Windows 10 was released by Microsoft after the succession of Windows 8 on July 29th, 2015. It was made available to the public through MSDN, Technet and free update of Windows 8 and Windows RT via Windows stores. Windows 10 became the most popular version of Windows worldwide, crossing Windows 7 on January 2018. Over 57% of all PCs are running on Windows 10, which suppressed other operating systems like Linux and MacOS.

According to Forbes, there are seven critical vulnerabilities that are impacting Windows 10 and its users in Aril 2020. The seven vulnerabilities are as followed:



However, given that there are seven critical vulnerabilities fixed this month, and those two exploits that are being used in active attacks, it might be an idea to jump the gun. After the update has been imposed, force a check for updates to give you the opportunity to download and install immediately, which would help in reducing vulnerabilities.

Let us see how we have analyzed the vulnerability trend from the release of the version to the present situation.

DATABASE USED

The database that we used for analyzing the vulnerabilities are from the tool named "CVE-Search". We preprocessed the entire CVE data and extracted to Windows 10 only.

TOOLS USED

For this, we have to utilize the tool named "CVE-Search", which is a tool for import CVE (Common Vulnerabilities and Exposures) and CPE (Common Platform Enumeration) into a MongoDB to facilitate search and processing of CVEs. *This tool used to store vulnerabilities and related information of the vulnerabilities.*

The requirements are:

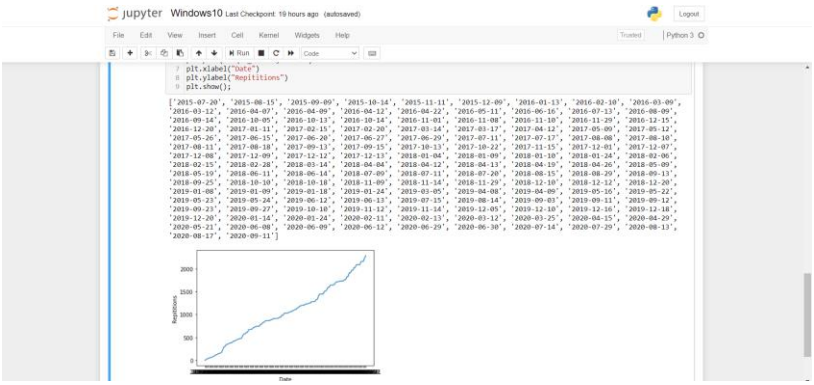
1. *Python 3.6*
2. *MongoDB 2.2*

First, we have to populate the databases by running the files db_mgmt_dictionary.py , and d_mgmt_json.py. The data base will be stored in MongoDB, followed by fetching the JSON file with the help of db_updater.py.

Then we can do the advanced search by running search.py keyword -o json where keyword is the one, we would search for. CVE-search is based on a set of tools, it can be used and combined with standard Unix tools.

SCRIPTS USED

Finally, we would get the necessary JSON file which has to be converted to CSV format for analyzing purpose. For this analysis, we have to use python programming language on Jupyter Notebook since it is easy to plot the trends with the analyzing process.



- *Importing the necessary packages needed for this analysis.*

```
import pandas as pd
import matplotlib.pyplot as plt
```

- *Reading the JSON file and printing the following data.*

```
df = pd.read_json(r'windows10.json', lines = True, orient = str)
data
```

- *Displaying the data of the JSON file.*

	Unnamed: 0	Modified	Published	access	assigner	cvss	cvss-time	cvss-vector	cwe	id	impact	
0	0	2019-05-14 19:38:00	2015-08-15 00:59:00	{'authentication': 'NONE', 'complexity': 'LOW'...	cve@mitre.org	7.2	2019-05-14 19:38:00	AV:L/AC:L/Au:N/C:C/I:C/A:C	CWE-264	CVE-2015-1769	{'availability': 'COMPLETE', 'confidentiality'...	['http://blogs.tec
1	1	2019-05-15 11:53:00	2015-08-15 00:59:00	{'authentication': 'NONE', 'complexity': 'MEDI...	cve@mitre.org	4.3	2019-05-15 11:53:00	AV:N/AC:M/Au:N/C:P/I:N/A:N	CWE-200	CVE-2015-2423	{'availability': 'NONE', 'confidentiality': 'P...	['http://www.sec
2	2	2019-05-15 15:23:00	2015-07-20 18:59:00	{'authentication': 'NONE', 'complexity': 'MEDI...	cve@mitre.org	9.3	2019-05-15 15:23:00	AV:N/AC:M/Au:N/C:C/I:C/A:C	CWE-119	CVE-2015-2426	{'availability': 'COMPLETE', 'confidentiality'...	['http://blog.trer
3	3	2019-05-15 15:01:00	2015-08-15 00:59:00	{'authentication': 'NONE', 'complexity': 'LOW'...	cve@mitre.org	2.1	2019-05-15 15:01:00	AV:L/AC:L/Au:N/C:P/I:N/A:N	CWE-200	CVE-2015-2433	{'availability': 'NONE', 'confidentiality': 'P...	['http://www.sec
4	4	2019-05-15 11:59:00	2015-08-15 00:59:00	{'authentication': 'NONE', 'complexity': 'MEDI...	cve@mitre.org	9.3	2019-05-15 11:59:00	AV:N/AC:M/Au:N/C:C/I:C/A:C	CWE-20	CVE-2015-2435	{'availability': 'COMPLETE', 'confidentiality'...	['http://www.sec
...	...	...	...	...	...	...	...	...	...	...	...	
2278	2278	2020-03-26 21:15:00	2020-02-13 16:15:00	{'authentication': 'NONE', 'complexity': 'MEDI...	cve@mitre.org	9.3	2020-03-26 21:15:00	AV:N/AC:M/Au:N/C:C/I:C/A:C	CWE-843	CVE-2020-3757	{'availability': 'COMPLETE', 'confidentiality'...	['https://a
2279	2279	2020-05-07 18:22:00	2020-04-29 15:15:00	{'authentication': 'SINGLE', 'complexity': 'LO...	cve@mitre.org	6.5	2020-05-07 18:22:00	AV:N/AC:L/Au:S/C:P/I:P/A:P	CWE-78	CVE-2020-7804	{'availability': 'PARTIAL', 'confidentiality':...	['http://www.han
2280	2280	2020-05-22 13:10:00	2020-05-21 19:15:00	{'authentication': 'NONE', 'complexity': 'LOW'...	cve@mitre.org	7.5	2020-05-22 13:10:00	AV:N/AC:L/Au:N/C:P/I:P/A:P	CWE-88	CVE-2020-7808	{'availability': 'PARTIAL', 'confidentiality':...	['https://www.bo

- *Creating a new column called Published for dividing the date time data.*

```
data.head()
dates = data["Published"]
months = []
```

- *Differentiating the date format from date time format using for loop.*

```
for date in dates:
    date, time = date.split()
    months.append(date)
    date = date.split("-")
    time = time.split(":")
```

- *Listing the unique dates from the date with initialization of integer named count.*

```
unique_months = list(set(months))
counts = []
c = 0
```

- Using a for loop for count the number of vulnerabilities found in that particular date.

```
for i in unique_months:
    c+=months.count(i)
    counts.append(c)
print(unique_months)

plt.plot(unique_months, counts)
plt.xlabel("Date")
plt.ylabel("Repititions")
plt.show();
```

- *Printing of the unique dates that were present in JSON file.*

```
['2015-07-20', '2015-08-15', '2015-09-09', '2015-10-14', '2015-11-11', '2015-12-09', '2016-01-13', '2016-02-10', '2016-03-09',
'2016-03-12', '2016-04-07', '2016-04-09', '2016-04-12', '2016-04-22', '2016-05-11', '2016-06-16', '2016-07-13', '2016-08-09',
'2016-09-14', '2016-10-05', '2016-10-13', '2016-10-14', '2016-11-01', '2016-11-08', '2016-11-10', '2016-11-29', '2016-12-15',
'2016-12-20', '2017-01-11', '2017-02-15', '2017-02-20', '2017-03-14', '2017-03-17', '2017-04-12', '2017-05-09', '2017-05-12',
'2017-05-26', '2017-06-15', '2017-06-20', '2017-06-27', '2017-06-29', '2017-07-11', '2017-07-17', '2017-08-08', '2017-08-10',
'2017-08-11', '2017-08-18', '2017-09-13', '2017-09-15', '2017-10-13', '2017-10-22', '2017-11-15', '2017-12-01', '2017-12-07',
'2017-12-08', '2017-12-09', '2017-12-12', '2017-12-13', '2018-01-04', '2018-01-09', '2018-01-10', '2018-01-24', '2018-02-06',
'2018-02-15', '2018-02-28', '2018-03-14', '2018-04-04', '2018-04-12', '2018-04-13', '2018-04-19', '2018-04-26', '2018-05-09',
'2018-05-19', '2018-06-11', '2018-06-14', '2018-07-09', '2018-07-11', '2018-07-20', '2018-08-15', '2018-08-29', '2018-09-13',
'2018-09-25', '2018-10-10', '2018-10-18', '2018-11-09', '2018-11-14', '2018-11-29', '2018-12-10', '2018-12-12', '2018-12-20',
'2019-01-08', '2019-01-09', '2019-01-18', '2019-01-24', '2019-03-05', '2019-04-08', '2019-04-09', '2019-05-16', '2019-05-22',
'2019-05-23', '2019-05-24', '2019-06-12', '2019-06-13', '2019-07-15', '2019-08-14', '2019-09-03', '2019-09-11', '2019-09-12',
'2019-09-23', '2019-09-27', '2019-10-10', '2019-11-12', '2019-11-14', '2019-12-05', '2019-12-10', '2019-12-16', '2019-12-18',
'2019-12-20', '2020-01-14', '2020-01-24', '2020-02-11', '2020-02-13', '2020-03-12', '2020-03-25', '2020-04-15', '2020-04-29',
'2020-05-21', '2020-06-08', '2020-06-09', '2020-06-12', '2020-06-29', '2020-06-30', '2020-07-14', '2020-07-29', '2020-08-13',
'2020-08-17', '2020-09-11']
```

GRAPH

We have taken a graph of the amount of CVE reports concerning the Windows from 20th July 2015 to 11th September 2020. From the graph, we can infer that there is a steady increase in the vulnerability trend. The trendline shows the uptrend from 2015 to 2017 and can be thought of as support when entering a position. Since the introduction of the platform started, discovery of the vulnerabilities were less. As years went on, the number of vulnerabilities tend to increase as new updates begun to function. On 11th September 2020, it tend to have a total vulnerability of 2233 in total. Uptrend lines is considered as a support and indicate that net-demand is increasing even

as the vulnerabilities rises. As long as vulnerability remain above the trend line, the uptrend is considered solid and intact.

GRAPH 1: DATE VS NUMBER OF VULNERABILITIES

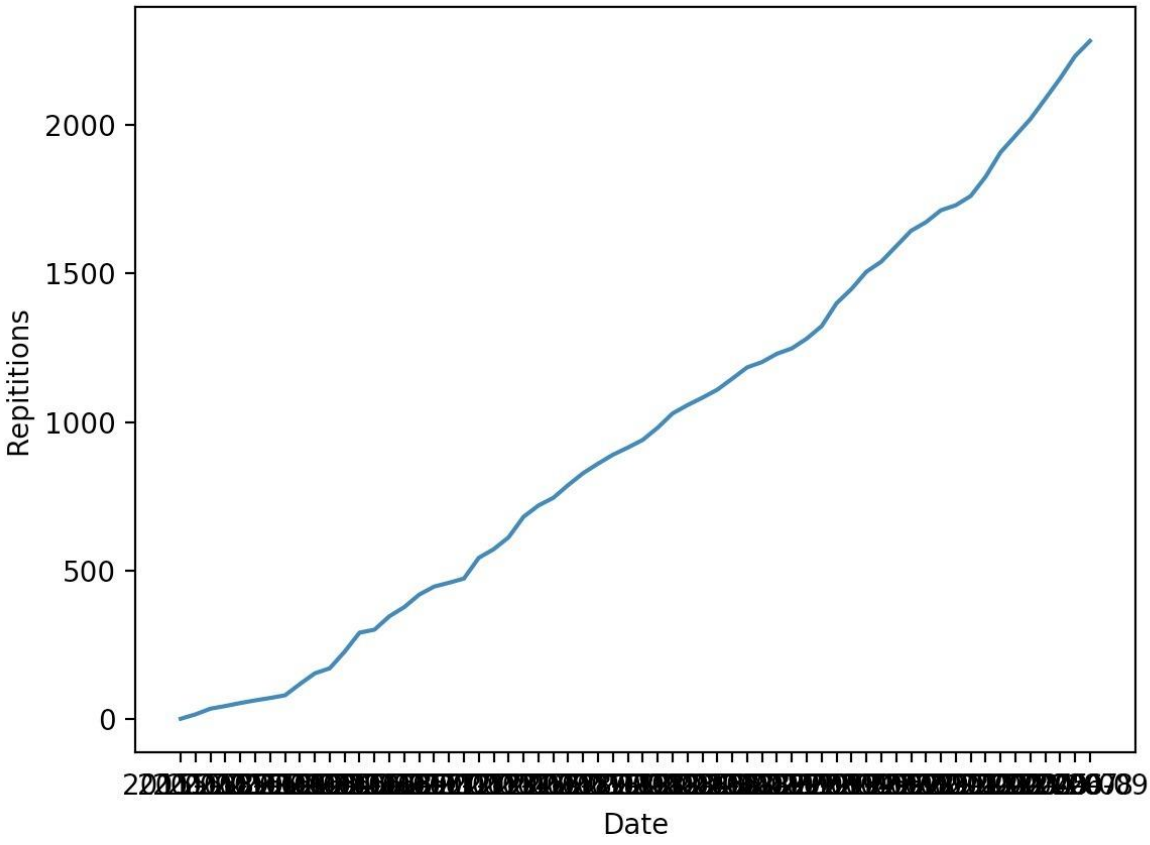
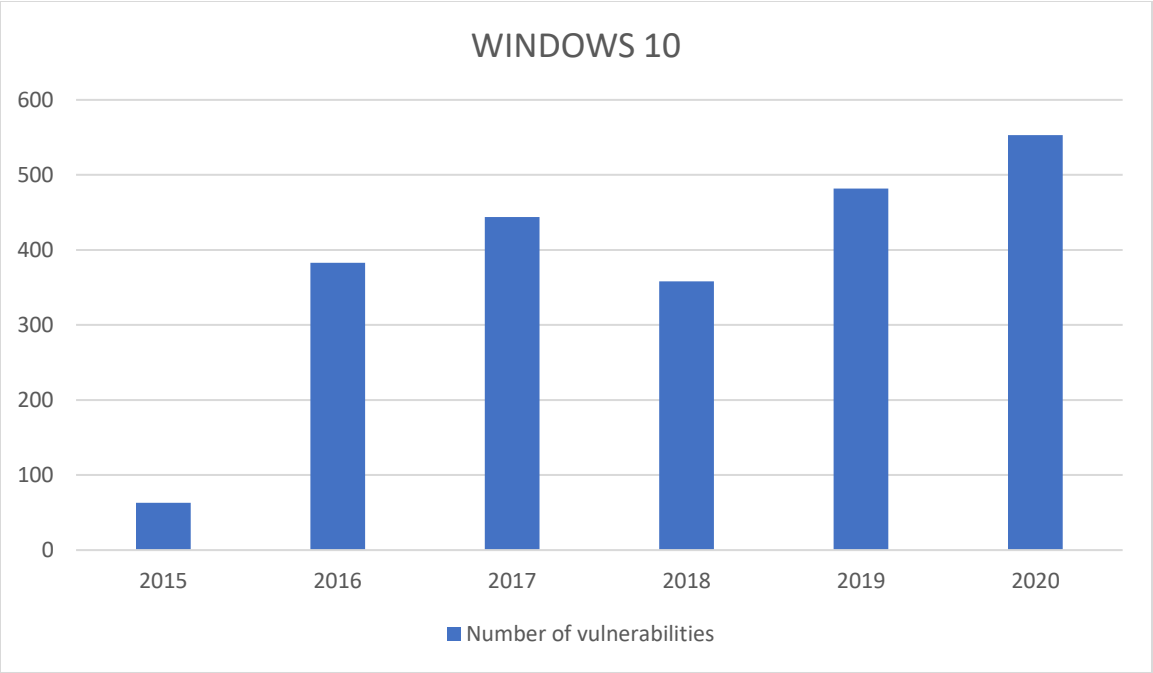


TABLE 1: YEARS VS NUMBER OF VULNERABILITIES

Years	*Number of vulnerabilities*
2015	*63*
2016	*383*
2017	*444*
2018	*358*
2019	*482*
2020	*553*

GRAPH 2: YEAR VS NUMBER OF VULNERABILITIES



We took over the year from the Published date and counted the number of vulnerabilities per year. From the number of vulnerabilities vs years, we can infer that more updates tend to be producing additional vulnerabilities, even though it suppressed the older vulnerabilities. There was a decrease in number of vulnerabilities in the year 2018 ,as there were also major vulnerabilities like Internet Explorer Vulnerability , Abode Flash Vulnerability. With the help of the dataset , we can see how many vulnerabilities are existing under the radar in the years 2015, 2016,2017 and 2018. Following on from this, there are higher rise in the count as it lies above the radar. From these facts, there might be lot of vulnerabilities at the end of 2020 even though the year is not ended.

PROCEDURE

- *Extracted data from CVE-search tool and filtered it with Windows 10.*
- *Downloaded the JSON file of Windows 10.*
- *Converted the JSON file to CSV file.*
- *Performed an analysis using Python and Jupyter Notebook.*
- *Plotted the graph for number of vulnerabilities with the following dates.*

REFERENCES:

1. *https://www.maketecheasier.com/latest-windows-10-security-threats-vulnerabilities/*
2. *https://github.com/cve-search/cve-search*
3. *https://school.stockcharts.com/doku.php?id=chart_analysis:trend_lines*