

ASSIGNMENT 3: VULNERABILITY ANALYSIS



Product Name: MacOS
Initial Release: January 1984
Current Release: September 2020
Code size: 80 SLOC^[1]

INTRODUCTION

Vulnerability Analysis involves identifying, analyzing, and prioritizing the vulnerability in a system. In this assignment you will extract vulnerability discovery data for two software products from a vulnerability data base (NVD). For extraction of data , we have utilized a script named vul.pull.py, which was written by Katherine Haynes.

MacOS is an operating system that powers every Apple computer. The first version was released in 1984 and completely changed the computer industry. It was the first commercial computer to feature a graphical user interface and a mouse, which made the machines much easier to user and therefore more accessible to non-tech users^[2]. Previously, using a computer meant understanding the abbreviated textual commands needed to interact with the command-line interface of an Apple II or IBM PC. But the introduction of the graphical user interface changed everything.

DATABASE USED

A python file , "vul.pull.py", is given for extracting vulnerabilities number from NVD JSON files. Instead of downloading the NVD JSON file from NIST website, a script named "nvd_csv_json_download.py" written by Mohammed Al Amin, which can download the NVD files and extracts in the JSON files.

TOOLS USED

For this assignment, we do not need to use any tools since we have extracted all the data from nvd_csv_json_download.py script.

SCRIPTS USED

For this analysis, we have to use python programming language on Linux Operating System since it is easy to plot the trends with the analyzing process. We ran a python script “vul.pull.py” which pulls out vulnerabilities from the NVD database. The input involves json file of annual vulnerability data from NVD.

The three steps involves:

- *CVESearch: The main purpose of this process is to read the NVD JSON files, pulls out all entries matching a text string, and returns a dictionary containing the matching CVE IDs and published date.*
- *CVEsave: It first reads in the dictionary from cveSearch and processes it. Then, it returns two numpy arrays, one with the time and one with the associated accumulated vulnerabilities. Finally, it Includes an option to save these two arrays to a file.*
- *CVEPlot: It takes two numpy arrays (time and vulnerabilities) and creates a line plot.*

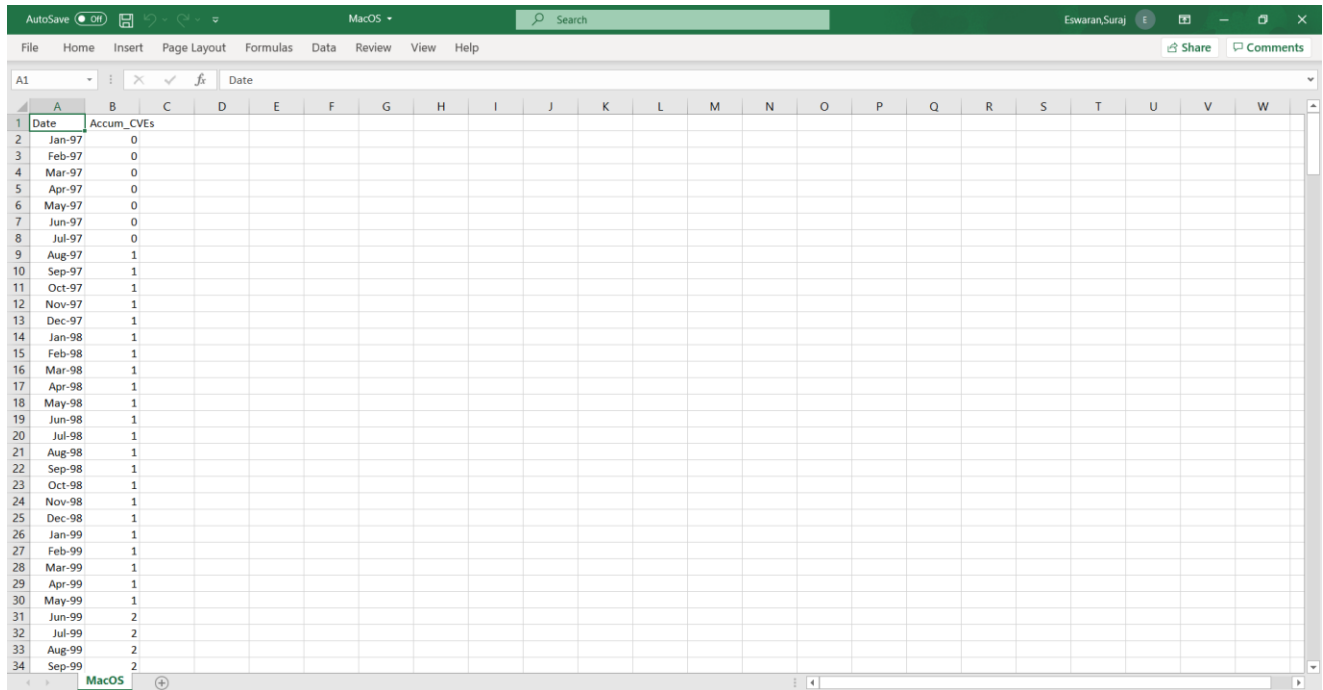
```
# Specify information for the products you wish to pull out:
myText1 = 'apple:mac_os_x'
myJSON1 = 'MacOS.json'
mySave1 = 'MacOS.csv'
myTitle1 = 'MacOS Vulnerabilities'
myFig1 = 'MacOS.png'

myText2 = 'internet_information_server:'
myJSON2 = 'IIS.json'
mySave2 = 'IIS.csv'
myTitle2 = 'Internet Information Server HTTP Server Vulnerabilities'
myFig2 = 'IIS.png'
```

Fig 1: Change in code

The necessary changes that were made to run the code is to change the specified information for the products you wish to pull out. In the code, it was mentioned Windows 10, so we need to change it to Apple MacOS.

We have obtained a record of total 4947 vulnerabilities from January 1997 to October 2020. The CVS file and JSON file has been created after running the script.



Date	Accum_CVEs
Jan-97	0
Feb-97	0
Mar-97	0
Apr-97	0
May-97	0
Jun-97	0
Jul-97	0
Aug-97	1
Sep-97	1
Oct-97	1
Nov-97	1
Dec-97	1
Jan-98	1
Feb-98	1
Mar-98	1
Apr-98	1
May-98	1
Jun-98	1
Jul-98	1
Aug-98	1
Sep-98	1
Oct-98	1
Nov-98	1
Dec-98	1
Jan-99	1
Feb-99	1
Mar-99	1
Apr-99	1
May-99	1
Jun-99	2
Jul-99	2
Aug-99	2
Sep-99	2

Fig 2: Resultant Excel file

GRAPH

1. Linear Model

Linear model is an equation that relates a relationship between two quantities with a constant rate of unit. It is usually described by two parameters: slope(or growth rate) and y intercept which is known as initial value. Thus , the linear model is represented as $y = mx + c$ where m =slope and c = constant rate^[3].

Below , we can see the graph of Vulnerabilities of MacOS from January 1997 to October 2020. The sign of a regression coefficient shows us whether there is a positive or negative correlation between independent variable and dependent variable. Positive coefficient results as independent variable value increases, mean of dependent variable also increase, whereas negative coefficient shows us that as independent variable increases , dependent variables decreases^[4].

If you move to the right along the x-axis by one thousand vulnerabilities, the line increases by 2059.7. It is only safe to interpret regression results within the observation space of your data. Let us consider that the regression line was flat, which tends to a coefficient of

zero. In this case, the mean would not change regardless of how far along the line you move. Thus, there is no effect, and you can witness a high p -value to go along with it.

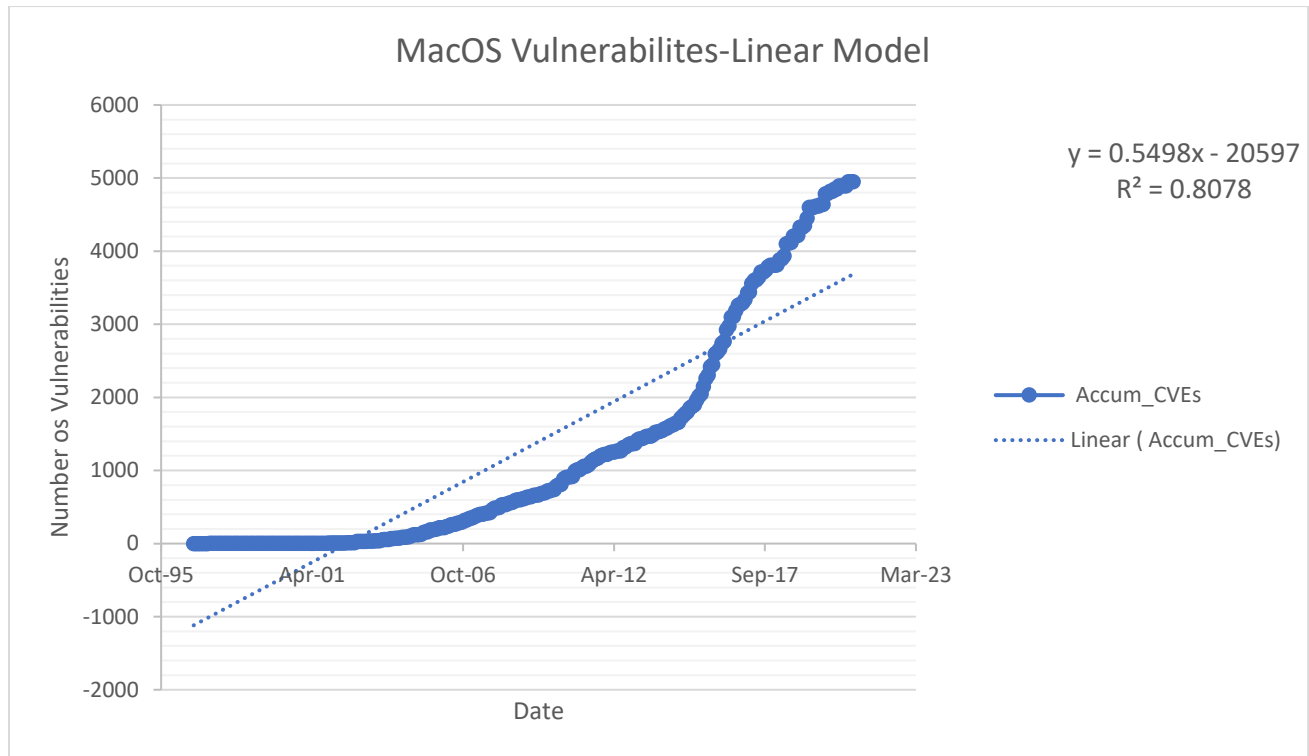


Fig 3: Linear Model for MacOS vulnerabilities

Linear model: slope = 0.5498/day, which is 200.677 per year
Intercept = -20597
 $R^2 = 0.8078$

2. AML Model

This model describes the rate of vulnerability discovery($w(t)$) or cumulative number of vulnerabilities($\bar{v}(t)$). $\bar{v}(t)$ can be found out by integrating $w(t)$ with respect to time.

$$\frac{dy}{dt} = Ay (B - y)$$

where y is called cumulative number of vulnerabilities, t is time. From the paper, A and B are considered to be empirical constant. By solving a differential equation, we get three parameter model, which is called

$$y = \frac{B}{BCe^{-ABt} + 1}$$

C is called as an another constant^[4]

Below , we can see the graph of Vulnerabilities of MacOS from January 1997 to October 2020. From the solver, we got the value of $A=0.00001$, $B=5020$ and $C=5$. As y value increases, the value of B tend to increase . So, it is necessary to find the proper B value in order to fit the values properly which should not be overfitting. Thus, the value is to be 5020.

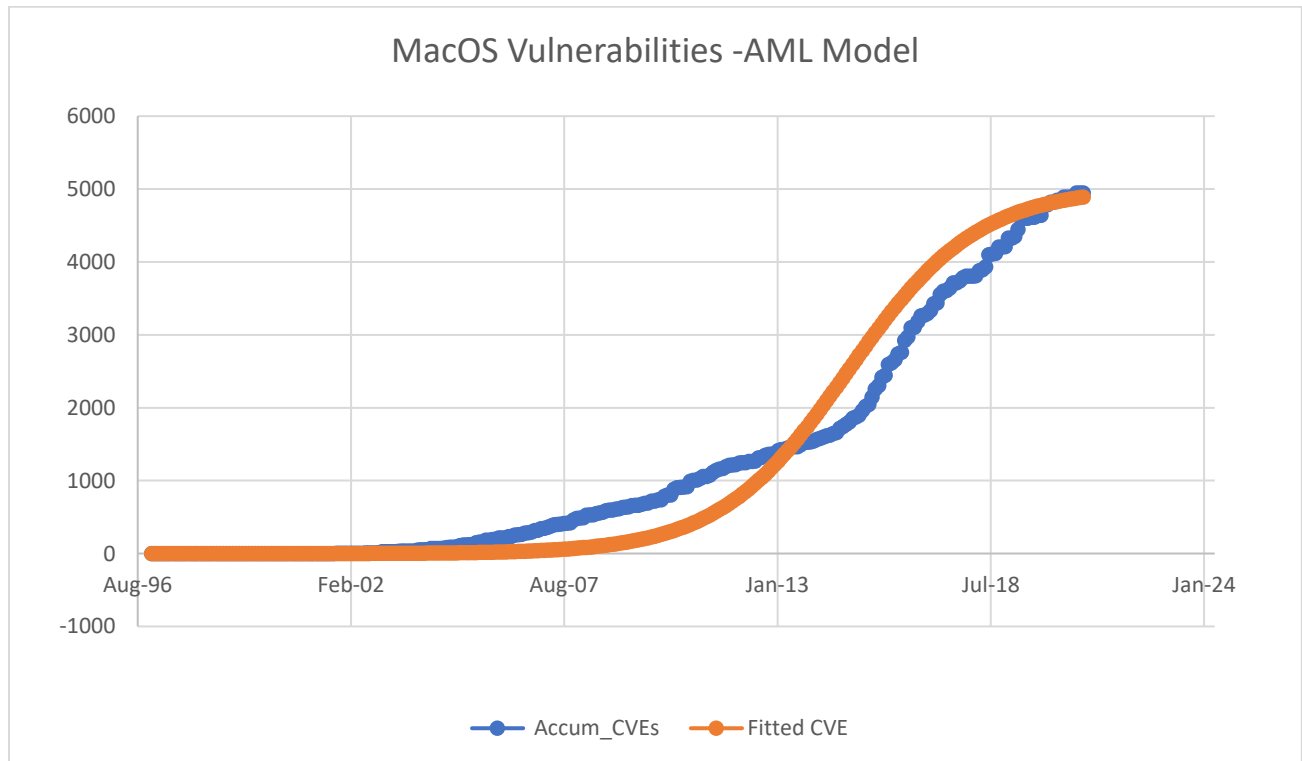


Fig 4: AML Model for MacOS Vulnerabilities

AML MODEL: $A = 0.00001$, $B = 5020$, $C=10$, $R^2 = 0.966767$

PROCEDURE

1. Run the `vul.pull.py` script.
2. Run the `nvd_cve_json_download.py`.
3. Plot the graph for date and accumulated vulnerabilities.
4. Create a linear and AML model.

OBSERVATION

There has been a steady increase in number of vulnerabilities increases , which shows that every version of MacOS developed tend to create a vulnerability even though there were patches formed. There is an increase between the year 2006 to 2007 due to evolution of malware like IM-Worm.OSX.Leap.a and Worm.OSX.Inqtana.A which has made a result saying MacOS doe also have security flaws which have the ability to compress the system. According to the scientist Luca Todesco, there are possibility of two new vulnerabilities which can be a security hole to the system^[6].

YEAR	VULNERABILITIES
1999	1
2000	0
2001	5
2002	20
2003	25
2004	54
2005	96
2006	120
2007	110
2008	95
2009	81
2010	97
2011	74
2012	37
2013	72
2014	150
2015	144
2016	117
2017	100
2018	110
2019	99
2020	50

Table 1: Number of vulnerabilities per year

REFERENCES

1. http://www.projectcodemeter.com/cost_estimation/help/GL_sloc.htm
2. <https://www.macworld.co.uk/feature/os-x-macos-versions-3662757/>
3. Graybill, Franklin A. Theory and application of the linear model. Vol. 183. North Scituate, MA: Duxbury press, 1976.
4. Alhazmi, O. H., & Malaiya, Y. K. (2005, November). Modeling the vulnerability discovery process. In 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05) (pp. 10-pp). IEEE.
5. <https://us.norton.com/internetsecurity-emerging-threats-two-new-vulnerabilities-found-in-mac-os-x.html>