SURAJ ESWARAN
CS559 QUANTITATIVE SECURITY

ASSIGNMENT 3: VULNERABILITY ANALYSIS

Microsoft IIS

Product Name: Microsoft Internet Information Services
Initial Release: May 1995
Current Release: October 2019
Maximum Length of content in a request: 30000000 bytes
Maximum length of the query string: 2048 bytes
Maximum of the URL: 4096 bytes
Code size: 67 SLOC[1]

Internet Information Services is a web server software which was developed by Microsoft on 30th May 1995 in order to use it as a host anything on the Web. It supports HTTP, HTTP/2, File Transfer Protocol, File Transfer Protocol Secure, Simple Mail Transfer Protocol and Network News Transfer Protocol[2]. It runs on Microsoft .NET platform on Windows Operating Systems. But it is possible to run Internet Information Services on Linux and MacOS with the help of Mono.

IIS is useful in hosting ASP.NET web applications and static web sites. It can be considered as a File Transfer Protocol Server which can be extended to host web applications on other platforms too. Another feature is that it can be administrated using CLI or using PowerShell. Due to these factors, IIS is high versatile and stable we server in Windows platforms[3].

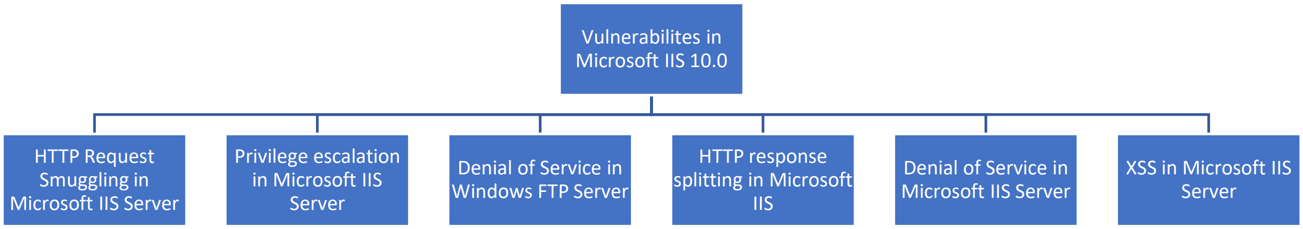

Vulnerabilites in Microsoft IIS 10.0
HTTP Request Smuggling in Microsoft IIS Server
Privilege escalation in Microsoft IIS Server
Denial of Service in Windows FTP Server
HTTP response splitting in Microsoft IIS
Denial of Service in Microsoft IIS Server
XSS in Microsoft IIS Server

Fig 1: Vulnerabilities in Microsoft IIS 10.0

IIS Vulnerabilities

Web Server Survey: Microsoft has been the second-best player in totals for totals as top active servers comparing all the domains. Microsoft begun to reach its gains where large domain hosting system at Network Solutions completing a migration to Windows 2000, because it has far less exposure to the mass hosting companies like Apache. It has been witnessed that Code Red has been the significant change in the month of September. But the togetherness of Code Red worm and IIS systems has improved the security of the system on the internet[4].

According to survey in 2017, researcher have given a statement regarding that the disclosure of zero-day vulnerability and exploit for a flaw in IIS 6.0. Two researchers have posted a proof-of-concept to GitHub. They explored by using a remote attacker using a long header in a PROFFIND request. Thus, the best strategy would be disabling WebDAV(World Wide Web Distributed Authoring and Versioning) which allows remote web users to be collaborated , write and edit on a server[4].

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 my 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.

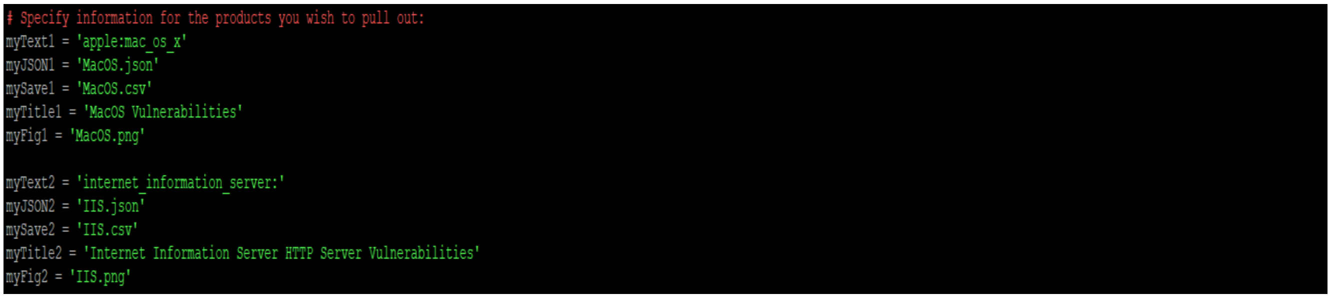
SCRIPTS USED

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: Th 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.*



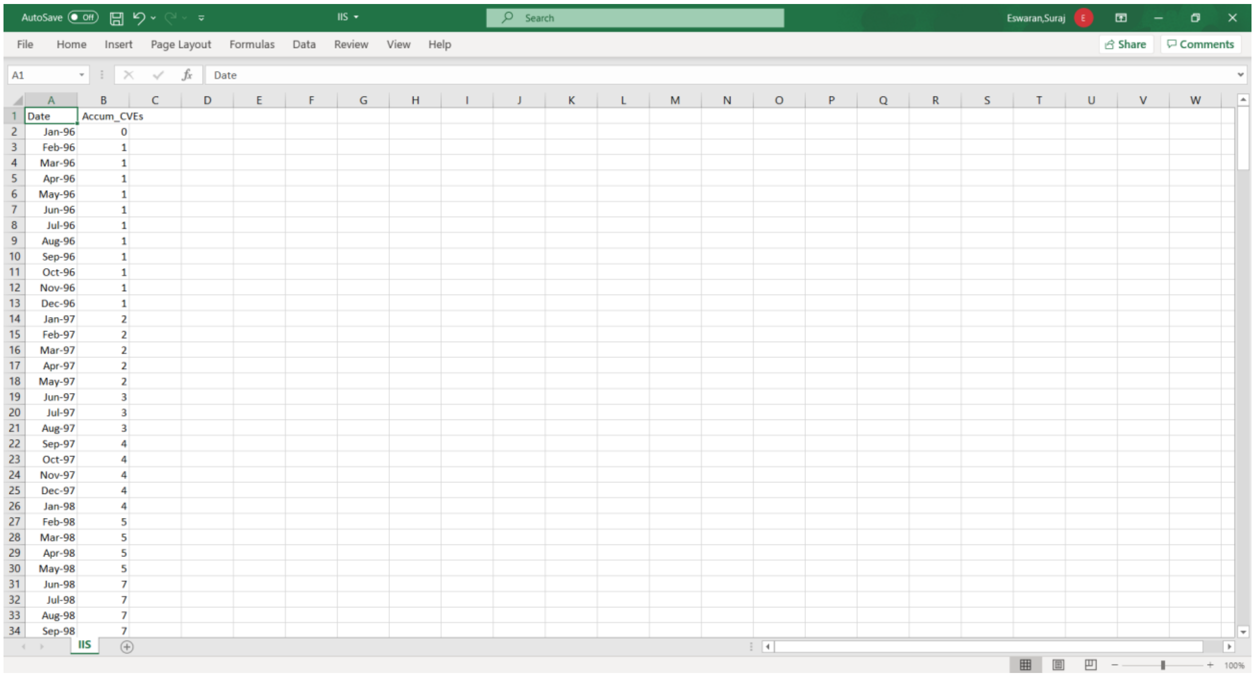
```
# Specify information for the products you wish to pull out:
myText1 = 'apple:mac_os_x'
myJSON1 = 'MacOS.json'
mySave1 = 'MacOS.csv'
myTitle1 = 'MacOS Vulnerabilities'
myFig1 = 'MacOS.png'

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

Fig 2: 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 135 vulnerabilities from January 1997 to October 2020. The CVS file and JSON file has been created after running the script.



Date	Accum_CVEs
Jan-96	0
Feb-96	1
Mar-96	1
Apr-96	1
May-96	1
Jun-96	1
Jul-96	1
Aug-96	1
Sep-96	1
Oct-96	1
Nov-96	1
Dec-96	1
Jan-97	2
Feb-97	2
Mar-97	2
Apr-97	2
May-97	2
Jun-97	3
Jul-97	3
Aug-97	3
Sep-97	4
Oct-97	4
Nov-97	4
Dec-97	4
Jan-98	4
Feb-98	5
Mar-98	5
Apr-98	5
May-98	5
Jun-98	7
Jul-98	7
Aug-98	7
Sep-98	7

Fig 3: Resultant Excel file

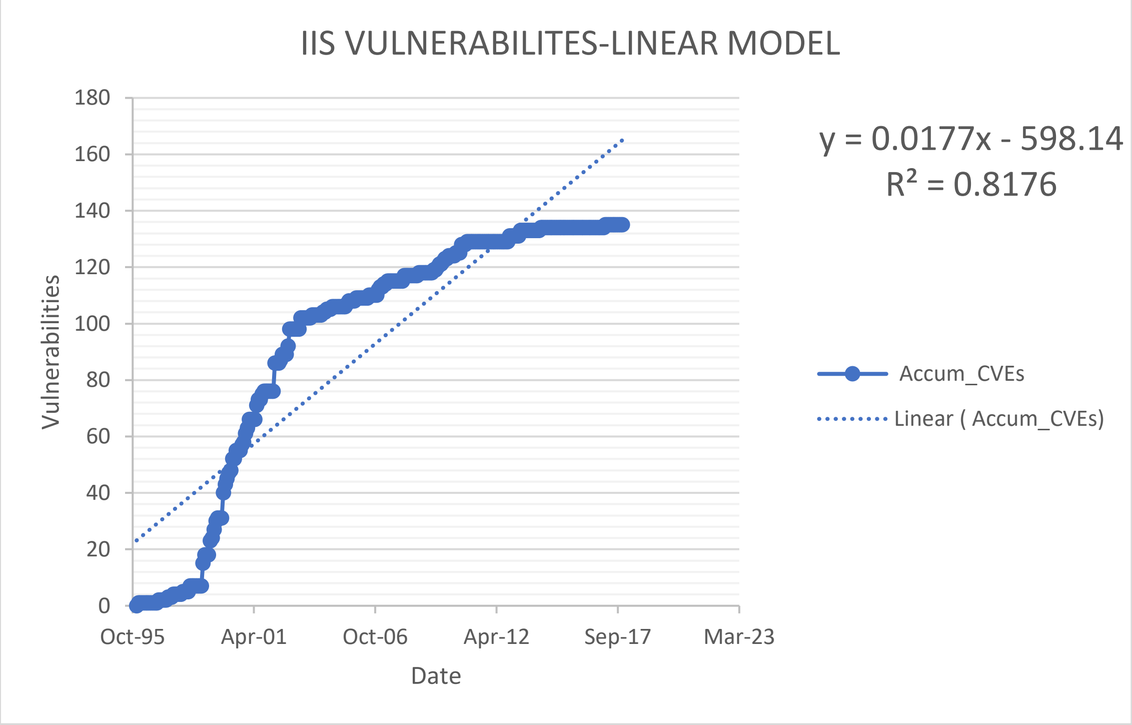
GRAPH

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[5].*

Below , we can see the graph of Vulnerabilities of IIS from May 1995 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.



If you move to the right along the x-axis by twenty vulnerabilities, the line increases by 58.14. It is only safe to interpret regression results within the observation space of your data. Let us consider that if our 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.

Linear model: slope = 0.0177/day, which is 6.4605 per year
Intercept =-598.14
R^2 = 0.9785

2. <u>*AML Model*</u>

This model describes the rate of vulnerability discovery(w(t)) or cumulative number of vulnerabilities(ò(t)). ò(t) can be found out by integrating w(t) with respect to time.

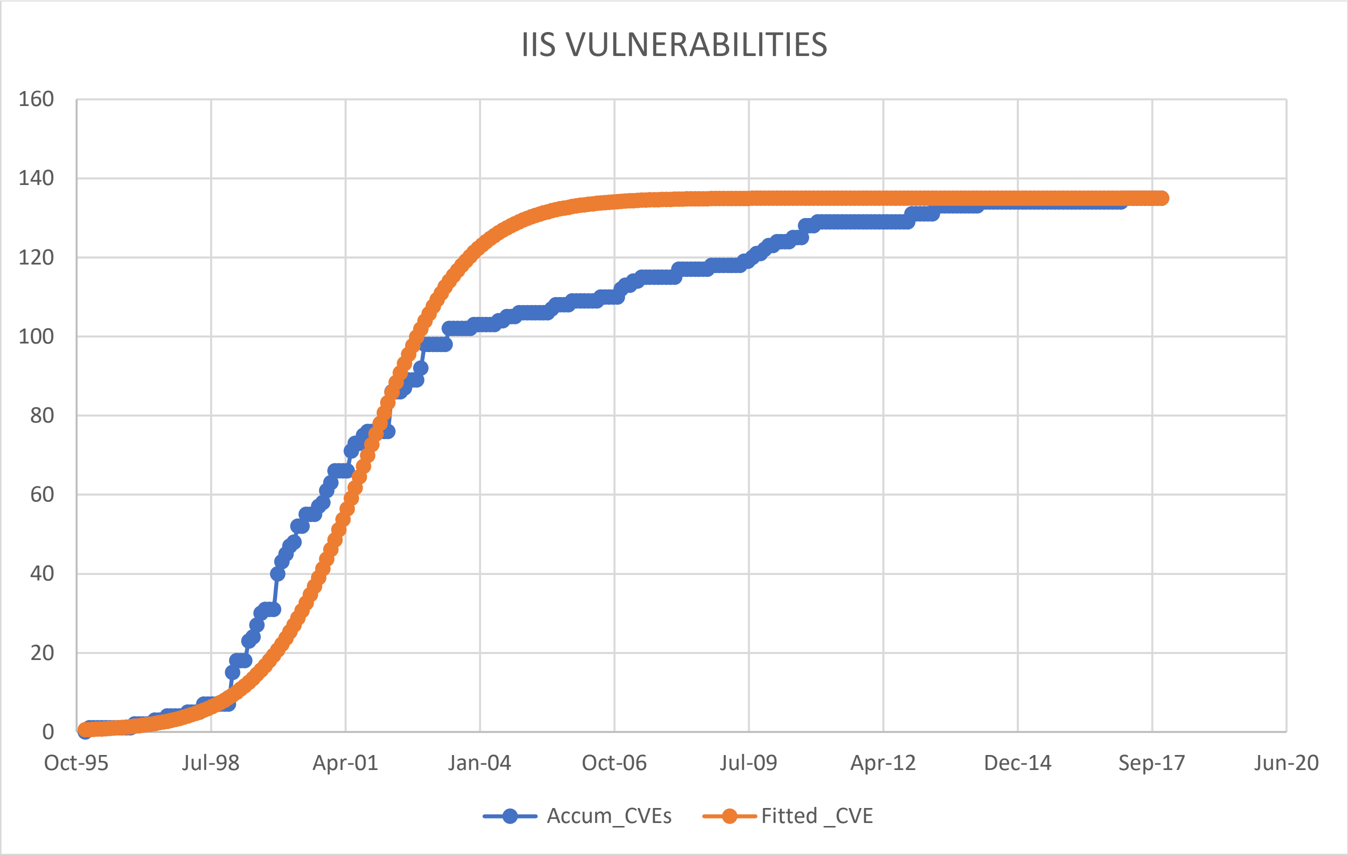
$$\frac{dy}{dt} = Ay\,(B - y)$$

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[6].

Below , we can see the graph of Vulnerabilities of MacOS from May 1995 to October 2020. From the solver, we got the value of A=0.0006, B=135 and C=2. 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 135.



AML MODEL: A = 0.0006, B = 135, C=2 ,R^2 = 0.958754

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:

While seeing the graph, there seems to be a discontinuity from 2017 and 2020.This is could be due to inactive release of latest versions, which tend to be stagnant. The biggest aspect is that we could found out the large number of vulnerabilities to be on the month of January 1999 due to exploitation of CVE-1999-0384 which deals with Forms 2.0 ActiveX control. Thus, there has not been any additional version after 2017, which involves a count of 135.

REFERENCES:

1. https://forums.iis.net/t/1251652.aspx
2. Schaefer, K., Cochran, J., Forsyth, S., Glendenning, D., & Perkins, B. (2012). Professional Microsoft IIS 8. John Wiley & Sons.
3. https://stackify.com/iis-web-server/
4. https://www.dnsstuff.com/windows-iis-server-tools
5. Graybill, Franklin A. Theory and application of the linear model. Vol. 183. North Scituate, MA: Duxbury press, 1976.
6. 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.