FAST ABSTRACT: Seasonality in Vulnerability Discovery in Major Software Systems

HyunChul Joh and Yashwant K. Malaiya
Computer Science Department
Colorado State University, Fort Collins, CO 80523
[*dean2026, malaiya*]*@cs.colostate.edu*

Abstract

Prediction of vulnerability discovery rates can be used to assess security risks and to determine the resources needed to develop patches quickly to handle vulnerabilities discovered. An examination of the vulnerability data suggests a seasonal behavior that has not been modeled by the recently proposed vulnerability discovery models. This seasonality has not been identified or examined so far. This study examines whether vulnerability discovery rates for Windows NT, IIS Server and the Internet Explorer exhibit a significant annual seasonal pattern. Actual data has been analyzed using seasonal index and autocorrelation function approaches to identify seasonality and to evaluate its statistical significance. The results for the three software systems show that there is indeed a significant annual seasonal pattern.

1. Introduction

A large number of vulnerabilities have been discovered in operating systems, web servers and browsers which represent a major security risk. If we can predict the vulnerability discovery pattern expected and the attributes of the vulnerabilities discovered, we can allocate the needed resource at the right time for corrective measures, which can greatly reduce the security risks.

Recently some vulnerability discovery models have been proposed which characterize security vulnerabilities in a quantitative manner [1]. They presume a specific discovery rate behavior during the product lifecycle spanning a few years. However the potential presence of seasonality has not yet been examined which may cause higher discovery rates during some specific durations in a calendar year. Seasonality analysis is a well known statistical approach used by researchers and analysts in life-sciences, finance, etc [2]. Here we examine the potential seasonal effect in the software vulnerability discovery process. If seasonality is actually present, that would require the use of seasonally adjusted data for identifying longer term trends and adjusting projections for seasonality for better accuracy.

2. Analysis of Seasonality

This paper focuses on three major software systems Windows NT, Internet Information Services (IIS) server and Internet Explorer (IE). Vulnerability data sets for these were obtained by mining the database at National institute of Standards and Technology [3].

We will examine the null hypothesis H_0: no seasonality is present. We will evaluate it using the seasonal index measure which states how much the average for a particular period tends to be above (or below) the expected value. The monthly seasonal index values are given by [4]:

$$s_i = \frac{d_i}{d}$$

where, s_i is the seasonal index for i^{th} month, d_i is the mean value of i^{th} month, d is a grand average. Hence, for example, a seasonal index of 1.25 indicates that the expected value for that month is 25% greater than $\frac{1}{12}$ of the overall average where the expected value is 1.

To see whether the seasonal indexes are statistically significant, chi-square (χ^2) test for the null hypothesis H_0 has been done. To be statistically significant, the calculated value of χ^2 statistic value (χ_s^2) must be greater than χ^2 critical value (χ_c^2) with small enough p-value.

The other approach to identify seasonality is to use the autocorrelation function (ACF). With time series values of z_b, z_{b+1}, …, z_n, the ACF at lag k, denoted by r_k , is [5]:

$$r_k = \frac{\sum_{t=b}^{n-k}(z_t-\bar{z})(z_{t+k}-\bar{z})}{\sum_{t=b}^{n}(z_t-\bar{z})^2}\text{, where } \bar{z} = \frac{\sum_{t=b}^{n} z_t}{(n-b+1)}$$

ACF measures the linear relationship between time series observations separated by a lag of k time units. When an ACF value is located outside of chosen upper

1071-9458/08 $25.00 © 2008 IEEE
DOI 10.1109/ISSRE.2008.31

or lower confidence intervals, there is a significant relationship associated with that time lag.

Results of the analysis for the three chosen software systems using seasonal index and the ACF analysis are discussed below.

3. Results and Discussion

Table I gives the number of vulnerabilities for each month with mean values and standard deviations. Table II shows the computed seasonal indexes with p-values. In both tables, it can be easily seen that vulnerabilities tend to be disclosed at a higher rate during mid-year and year-end months. The very small p-values for H_0 indicate there is a strong evidence of non-uniform distributions of vulnerability discovery [3]. Figure 1 gives the percentage of vulnerabilities disclosed each month. It clearly shows that summer and winter rates are above the 1/12 = 8.33% line which is the overall mean.

Figure 2, shows ACF values of the three software systems. Since in Tables I, Table II and Figure 1, mid-year and year-end months have more vulnerabilities, it is expected that the lags corresponding to about six months or its multiples would have the ACF values outside of the confidence intervals. In Figure 2, for Windows NT, lags for 5, 6, 11 and 24 are outside of the confidence interval; in other words, there are strong autocorrelations with lags that are multiples of 6 which is confirming a seasonal pattern. For IIS, lags for 4, 6, 7, 8, 10, 11, 14 and 18 are outside of the confidence interval. For IE, lags for 1, 4, 5, 6, 7, 12, 19 and 24 are significantly different from zero of ACF.

The results thus show strong seasonality in the systems examined, with much discovery rates in some months. This needs to be taken into account for making accurate projections.

Further research with diverse software products, both commercial and open-source, is needed to identify the causes of seasonality and possible variation across software systems.

Table I.
Vulnerabilities Disclosed

	WinNT	IIS	IE
Jan	42	15	15
Feb	20	10	32
Mar	12	2	22
Apr	13	11	29
May	18	12	41
Jun	24	17	45
Jul	18	11	53
Aug	17	7	42
Sep	11	6	26
Oct	14	6	20
Nov	18	7	26
Dec	51	28	93
Total	258	132	444
Mean	21.5	11	37
s.d.	12.37	6.78	20.94

Table II.
Seasonal Index Values

	WinNT	IIS	IE
Jan	1.95	1.36	0.41
Feb	0.93	0.91	0.86
Mar	0.56	0.81	0.59
Apr	0.60	1.00	0.78
May	0.84	1.09	1.11
Jun	1.12	1.55	1.22
Jul	0.84	1.00	1.43
Aug	0.79	0.64	1.14
Sep	0.51	0.55	0.70
Oct	0.65	0.55	0.54
Nov	0.84	0.64	0.70
Dec	2.37	2.55	2.51
χ_c^2	19.68	19.68	19.68
χ_s^2	78.37	46	130.43
p-val.	3.04e-12	3.23e-6	1.42e-6

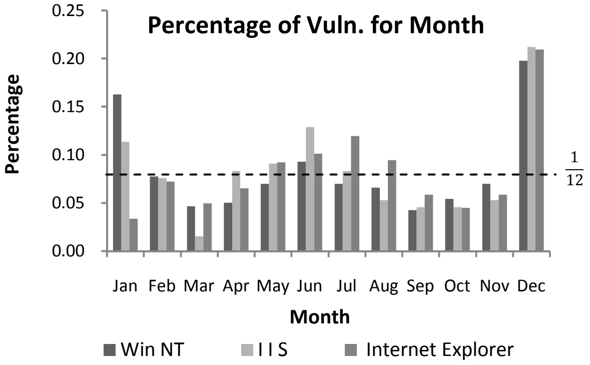

Figure 1. Percentage of vulnerability for each month

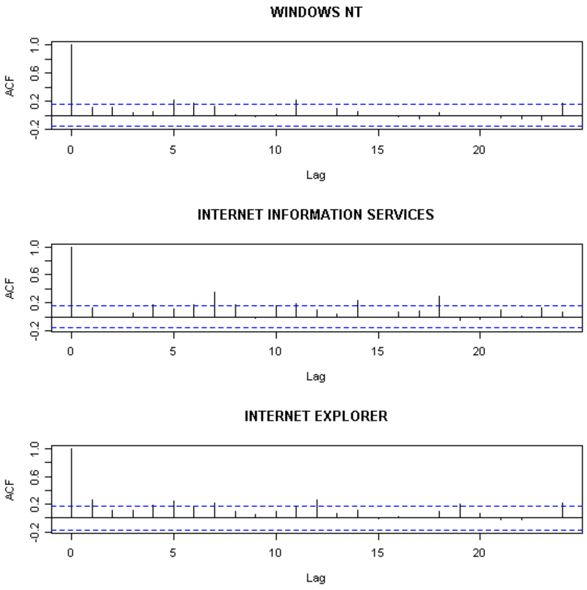

Figure 2. Autocorrelation functions; upper/lower dotted lines represent 95% confidence intervals. An event occurring at time t + k (k > 0) is said to lag behind an event occurring at time t, the extent of the lag being k. Here lags are in month.

References

[1] O. H. Alhazmi and Y. K. Malaiya, "Application of Vulnerability Discovery Models to Major Operating Systems," *IEEE Trans. Reliability*, Mar. 2008, pp. 14-22.

[2] M.Rios, J.M. Garcia, J. A. Sanchez, and D. Perez, "A Statistical Analysis of the Seasonality in Pulmonary Tuberculosis," *European Journal of Epidemiology*, Vol. 16, No. 5. (May, 2000), pp. 483-488.

[3] National Institute of Standards and Technology. National Vulnerability Database. Available:http://nvd.nist.gov/download.cfm, March 31, 2008.

[4] Hossein Arsham. Time-Critical Decision Making for Business Administration. Available: http://home.ubalt.edu/ntsbarsh/Business-stat/stat-data/Forecast.htm#rseasonindx, March 31, 2008.

[5] B. L. Bowerman and R. T. O'connell, *Time Series Forecsting: Unified concepts and computer implementation*. 2nd Ed., Boston: Duxbury Press, 1987.