POSTER: Understanding the Hidden Cost of Software Vulnerabilities: Measurements and Predictions

Afsah Anwar  
University of Central Florida  
afshanwar@knights.ucf.edu

Aminollah Khormali  
University of Central Florida  
aminkhormali@knights.ucf.edu

Aziz Mohaisen  
University of Central Florida  
mohaisen@ucf.edu

ABSTRACT

In this work, we study the hidden cost of software vulnerabilities reported in the National Vulnerability Database (NVD) through stock price analysis. We perform a high-fidelity data augmentation to ensure data reliability for estimating vulnerability disclosure dates as a baseline for assessing software vulnerabilities’ implication. We further build a model for stock price prediction using the NARX Neural Network model to estimate the effect of vulnerability disclosure on the stock price. Compared to prior work, which relies on linear regression models, our approach is shown to provide better accuracy. Our analysis shows that the effect of vulnerabilities on vendors varies, and greatly depends on the specific industry.

KEYWORDS

Vulnerability Economics; Prediction; NVD

ACM Reference Format:


1 INTRODUCTION

Vulnerabilities are defects in software products, exposing them and users to risk alike. To deal with vulnerabilities, vendors incur costs in the form of developer-hours spent fixing them and redeploying the fixes; a study by the National Institute of Standards and Technology (NIST) estimated that the US economy about $60 Billion USD every year for patches development and redistribution, systems re-deployment, and direct productivity loss [16]. In 2012, Knight Capital, a fintech company, lost $400 Million USD because of a bug in their code; the company bought shares at the ask price and sold them at the bid price [17]. Losses from WannaCry (2017), a ransomware attack in over 150 countries affecting more than 100,000 groups, is estimated at $4 Billion USD [4]. Virus attacks, such as Love Bug (2000), SirCam (2001), Nimda (2001), and CodeRed (2001), have had an impact of $8.75 Billion, $1.25 Billion, $1.5 Billion and $2.75 Billion USD, respectively [1]. The direct losses, however, are not the only cost of vulnerabilities, and companies incur additional hidden costs that we attempt to assess in this work.

2 METHODOLOGY

Using the information available on the NVD, the goal of this study is to track the public disclosure date of vulnerabilities and capture their impact on vendors stock market valuation.

2.1 Data and Data Augmentation

Figure 1 summarizes, at a high-level, the flow of data creation, from the source of data to the final dataset. The National Vulnerability Database (NVD), is a vulnerability database maintained by the National Institute of Standards and Technology (NIST) and contains the vulnerabilities reported to MITRE [2]. The NVD includes the following elements for each vulnerability: the CVE-ID, vendor, product, Common Vulnerability Scoring System (CVSS) label, published date, Common Weakness Enumeration Identifier (CWE-ID) [3], description, and reference links.

Data Preprocessing and Augmentation. We use the vulnerabilities reported in the year 2016, and limit our analysis to the severe ones. To assess the impact of intrinsic vulnerabilities in software, rather than inherited vulnerabilities due to third-party libraries, we discard vulnerabilities with “library” in their description. We use the links from the NVD to scrape through the web and label public disclosure date for each of the vulnerabilities by calculating the minimum of the dates gathered from the links corresponding to a vulnerability, and VHSP is the Vendor Historical Stock Price.

Figure 1: Dataset Creation Flow. Desc. stands for vulnerability description, Ref. Link is the link referring to details of the vulnerability, Pub. Date is the Published Date, CVSS is Common Vulnerability Scoring System metrics, CWE is Common Weakness Enumeration identifier, PDD is the Public Disclosure Date, approximated as the minimum of the dates gathered from the links corresponding to a vulnerability, and VHSP is the Vendor Historical Stock Price.

Contributions. We quantitatively analyze the loss by software vendors due to software vulnerabilities through stock price analysis. Particularly, we make the following contribution. (i) An evaluation of vulnerabilities, disclosed in the year 2016, from the National Vulnerability Database (NVD) and their impact on their vendors. (ii) An accurate method for predicting stock price of the next day using NARX Neural Network. (iii) Industry-impact correlation analysis. (iv) Vulnerability type analysis, indicating that different types have different powers of affecting the stock price of a vendor.
For vendors with redundant vendor names, e.g., schneider-electric vs. schneider_electric, trend-micro vs. trend-micro, etc., we consolidated the various vendors under a consistent name, also through manual inspection.

**Yahoo Finance.** The *date* attribute in the historical stock data for each vendor corresponds to the date on which the stock’s performance is captured. The *open* and *close* attributes are the stock value of the vendor on the given day at the opening and closing of the market, respectively. The *low* and *high* are the lowest and highest value of the vendor’s stock achieved on the given day. The *adjacent close* attribute corresponds to the dividends and splits since that day. The *volume* is the number of shares traded on the given day.

**Press.** As a baseline for comparison with our results based on the work model, the next value is regressed on previous values of the output and exogenous input, represented using the following:

\[ y(t) = f[u(t-1),..,u(t-d_u); y(t-1),..,y(t-d_y)], \]

where \( u(t) \) and \( y(t) \) are the input and output of the network at time \( t \). \( d_u \) and \( d_y \) are the lags of exogenous inputs and output of the system, and the function \( f \) is multi-layer feed forward network.

For each vendor, we divide the dataset into training, validation and test subsets (with 70%, 15%, and 15%, respectively). MSE, used to evaluate the performance of the models, is defined as:

\[ \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_{pi} - y_{pi})^2, \]

where \( n \) is the number of samples, and \( y_i \) and \( y_p \) are the actual stock price and the corresponding predicted value, respectively.

**Baseline for Comparison.** We also predicted the stock price of vendors using the Auto-Regressive Integrated Moving Average (ARIMA) model [6], a popular time series prediction model, for comparison. The difference in the value of MSE for these to models, 6.42 for ARIMA and 0.59 for NARX (for Adobe), quantitatively justifies using the proposed method over methods used in the literature.

## 4 RESULTS
Table 1 shows the normalized MSE, count of the vulnerabilities, and AR on days 1, 2, and 3 for every vendor (as described above). We observe that vulnerabilities had an adverse impact on the stock price of 17 out of the 36 vendors. The effect, however, could be due to one or more vulnerabilities on the given day.

Table 2 represents a breakdown of vendors by industry and their likelihood of their stock being impacted by vulnerabilities.

We look at vulnerabilities from 10 vendors to find the reason for the nearly no-effect of vulnerabilities in some industries. From the description of the vulnerabilities, we observe the following. (1) Vulnerabilities affecting vendors’ stock negatively are of critical severity (vulnerabilities with CVSS version 3 label of CRITICAL) while the rest were less severe (vulnerabilities with CVSS labels of HIGH or MEDIUM). (2) Vulnerabilities affecting vendors’ stock price negatively have a combination of version 3 label of HIGH or
We perform an empirical analysis on vulnerabilities from NVD and look at their effect on stock price. Our results show that the effect is industry-specific, and depends on the severity of the reported vulnerabilities. We also compare the results with the vulnerabilities found in popular press: while both vulnerabilities affect the vendor’s stock, vulnerabilities reported in the media have a much more adverse effect. En route, we also design a model to predict the stock price with high accuracy.

**Acknowledgement.** This work is supported in part by NSF grant CNS-1809000 and NRF grant NRF-2016K1A1A2912757.

**REFERENCES**