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

Afsah Anwar
University of Central Florida
afsahanwar@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:

Afsah Anwar, Aminollah Khormali, and Aziz Mohaisen. 2018. POSTER: Understanding the Hidden Cost of Software Vulnerabilities: Measurements and Predictions. In *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security*, June 4–8, 2018, Incheon, Republic of Korea. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3196494.3201580>

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 re-deploying the fixes; a study by the National Institute of Standards and Technology (NIST) estimated that vulnerabilities costed 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.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ASIA CCS '18, June 4–8, 2018, Incheon, Republic of Korea

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5576-6/18/06.

<https://doi.org/10.1145/3196494.3201580>

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 the 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.

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.

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 (where multiple dates are available). For 1,262 links that prevent automated scraping, we manually visited

Figure 2: General Structure of the NARX Neural Network the links to find the disclosure date. 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 *adjusted 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 approach used in the literature, we sample vulnerabilities reported in the press. We search for "software vulnerabilities in 2017" in *Forbes*, and *ZDNet*, and list four vulnerabilities for comparison.

2.2 Assessing Vulnerability's Impact

To assess the effect of a vulnerability for the day on which the vulnerability was published, we look for the stock value on that particular date. For all dates with disclosed vulnerabilities whereby the stock data is unavailable and the value on the last operating day, d_0 , is x , the market was closed on days d_1, d_2 , and d_3 , and the value on next operating day, d_4 , is y . We approximate the values on days d_i for $i \in \{1, 2, 3\}$ as $d_i = x + \frac{i \times (y - x)}{d}$, where d is the number of days between d_0 and d_4 (here, 3). We predict a stock price for the day when a vulnerability occurs considering previous values and calculate Abnormal Return on day i (AR_i , for $i \in \{1, 2, 3\}$), where $AR_i = R_i - \bar{R}$, such that R_i is the actual stock price on day i , and $\bar{R}$ is the expected stock without vulnerability (predicted). We then calculate the Percentage of Abnormal Return on day i (PAR_i), where $i \in 1, 2, 3$, as $PAR_i = \frac{AR_i \times 100}{R_i}$. Finally, we calculate the Overall (%) Abnormal Return on day i (OAR_i), where $i \in \{1, 2, 3\}$. For vendors $\{V_1, \dots, V_k\}$, the PAR values for a vendor V_j are denoted by PAR_i^j for $i \in \{1, 2, 3\}$. We calculate $OAR_i = \sum_{j=1}^k PAR_i^j$ on day i .

3 PREDICTION

Normalization. Each feature x , open, close, high, low, volume, and adjacent close, is transformed into feature vector $z = \frac{x - \bar{x}}{\sigma}$, where $\bar{x}$ and σ , are the mean and standard deviation of the original feature vector x , respectively. The resulting features are then fed into the nonlinear auto-regressive neural network with exogenous factors (NARX) to predict the stock value of vendors.

Figure 3: Actual vs. Predicted: NARX.

NARX Neural Network. NARX, generally applied for discrete-time nonlinear dynamical systems' behavior prediction, is an efficient forecasting tool [8]. Compared to other architectures, which consider feedback from both hidden and output neurons, NARX is more efficient and accurate [11]. Based on the NARX neural network model, the next value is regressed on previous values of the output and exogenous input, represented using the following:

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

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:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{ti} - y_{pi})^2,$$

where n is the number of samples, and y_t 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

Table 1: Results for each Vendor. Vul. stands for vulnerability count and OAR_1 , OAR_2 , and OAR_3 stand for the average effect at day 1, 2, and 3 (percent), respectively. $\blacktriangle$ indicates that the vulnerabilities had no overall impact on vendor's stock value while $\blacktriangledown$ indicates that the stock of the vendor were impacted, overall.

Vendor	MSE	Vul.	$OAR_1^{(1)}$	$OAR_2^{(1)}$	$OAR_3^{(1)}$
Adobe	5.9E-4	494	$\blacktriangledown$ 0.65	$\blacktriangledown$ 0.37	$\blacktriangledown$ 0.50
Advantech	9.5E-4	9	$\blacktriangle$ 0.61	$\blacktriangle$ 0.89	$\blacktriangle$ 0.96
Apache	9.9E-4	37	$\blacktriangle$ 0.60	$\blacktriangle$ 0.98	$\blacktriangle$ 1.17
Apple	2.8E-4	154	$\blacktriangle$ 0.41	$\blacktriangle$ 0.75	$\blacktriangle$ 1.03
Atlassian	9.7E-3	4	$\blacktriangledown$ 3.85	$\blacktriangledown$ 3.86	$\blacktriangledown$ 3.12
Cisco	2.3E-3	111	$\blacktriangle$ 0.10	$\blacktriangle$ 0.33	$\blacktriangle$ 0.42
Citrix	2.4E-3	9	$\blacktriangle$ 0.14	$\blacktriangle$ 0.01	$\blacktriangle$ 0.57
Facebook	1.1E-3	6	$\blacktriangle$ 0.13	$\blacktriangledown$ 0.33	$\blacktriangle$ 0.45
Fortinet	4.5E-3	7	$\blacktriangle$ 0.37	$\blacktriangle$ 0.19	$\blacktriangle$ 0.92
Ge	5.8E-4	3	$\blacktriangle$ 0.12	$\blacktriangledown$ 0.58	$\blacktriangledown$ 0.39
Google	7.6E-4	410	$\blacktriangledown$ 0.08	$\blacktriangledown$ 0.21	$\blacktriangledown$ 0.08

CRITICAL. Vulnerabilities involving unauthorized accesses have a higher cost, seen in their effect on the stock price. (3) Denial of Service attacks unrelated to confidentiality do not impact stock.

For the vulnerabilities gathered from the press, we followed the same steps and observe that these vulnerabilities have an adverse effect on vendor stock price in almost every case.

Discussion. Prior works show the impact of vulnerabilities using Cumulative Abnormal Rate (CAR), which aggregates AR's on different days. However, CAR does not effectively capture the impact of a vulnerability. For example, CAR would indicate no-effect if the magnitude (upward) of one or more days analyzed negate the magnitude (downward) of other days. Also, our results through a rigorous analysis are statistically significant. One main shortcoming of the prior work, however, is that it overlooks analyzing the cost based on sectors of the software industry. In this study we found a significant effect of vulnerabilities on a given day and limited ourselves to the third day after the release of the vulnerability in order to minimize the impact of other factors. Eliminating effects of other factors and measuring long-term losses is an open question.

5 RELATED WORK

While Hovav and D'Archy [12] have shown that market shows no signs of significant negative reaction due to vulnerabilities, Telang *et al.* [18] show that a vendor on average loses 0.6% of its stock value due to vulnerabilities. Goel *et al.* [10] also pointed out that security breaches have an adverse impact on the market value of a vendor. Bose *et al.* [5] show that each phishing alert leads to a loss of market capitalization that is at least US\$ 411 million for a firm.

Li and Paxson [14] outlined a method to approximate public disclosure date by scrapping reference links in NVD, which we use in this study. Nguyen and Massaci [15] show versions data in NVD to be unreliable. Christey and Martin [7] outlined caveats with the NVD data. Jarrell and Peltzman [13] analyzed the impact of recall in the drug and auto industries on vendors' stock value loss. Farhang *et al.* [9], suggest that higher security investments in Android devices do not impose higher product prices on customers.

Table 2: Per industry stock impact likelihood analysis.

Industry	Likelihood
Software	Highly Likely
Consumer Products	Highly Likely
Finance	Highly Likely
Security	Equally Likely
Electronics & Hardware	Equally Likely
Conglomerate	Less Likely
Device	Less Likely
Networking	Less Likely

6 CONCLUSION AND FUTURE WORK

We perform an empirical analysis on vulnerabilities from NVD and look at their effect on vendor's 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

- [1] . —. The Cost Impact of Major Virus Attacks Since 1995. (—). <https://www.computereconomics.com/article.cfm?id=936>
- [2] . —. CVE. (—). <https://cve.mitre.org/>
- [3] . —. CWE. (—). <https://cwe.mitre.org>
- [4] . —. "WannaCry" ransomware attack losses could reach \$4 billion. (—). <http://cbsn.ws/2yYjif2>
- [5] Indranil Bose and Alvin Chung Man Leung. 2014. Do phishing alerts impact global corporations? A firm value analysis. *Decision Support Systems* 64 (2014), 67–78.
- [6] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 332 (1970), 1509–1526.
- [7] Steve Christey and Brian Martin. 2013. Buying into the bias: Why vulnerability statistics suck. *BlackHat, Las Vegas, USA, Tech. Rep* 1 (2013).
- [8] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.
- [9] Sadegh Farhang, Aron Laszka, and Jens Grossklags. 2017. An Economic Study of the Effect of Android Platform Fragmentation on Security Updates. *arXiv preprint arXiv:1712.08222* (2017).
- [10] Sanjay Goel and Hany A Shawky. 2009. Estimating the market impact of security breach announcements on firm values. *Information & Management* 46, 7 (2009), 404–410.
- [11] Bill G Horne and C Lee Giles. 1995. An experimental comparison of recurrent neural networks. In *Advances in neural information processing systems*. 697–704.
- [12] Anat Hovav and John D'arcy. 2005. Capital market reaction to defective IT products: The case of computer viruses. *Computers & Security* 24, 5 (2005), 409–424.
- [13] Gregg Jarrell and Sam Peltzman. 1985. The impact of product recalls on the wealth of sellers. *Journal of Political Economy* 93, 3 (1985), 512–536.
- [14] Frank Li and Vern Paxson. 2017. A Large-Scale Empirical Study of Security Patches. (2017).
- [15] Viet Hung Nguyen and Fabio Massacci. 2013. The (un) reliability of NVD vulnerable versions data: An empirical experiment on Google Chrome vulnerabilities. In *ACM AsiaCCS*. 493–498.
- [16] Strategic Planning. 2002. The economic impacts of inadequate infrastructure for software testing. (2002).
- [17] Jenny Strasburg and J Bunge. 2012. Loss swamps trading firm, Knight Capital searches for partner as tab for computer glitch hits \$440 million. *Wall Street Journal* (2012).
- [18] Rahul Telang and Sunil Wattal. 2007. An empirical analysis of the impact of software vulnerability announcements on firm stock price. *IEEE Transactions on Software Engineering* 33, 8 (2007), 544–557.