EAI Endorsed Transactions
on Security and Safety
Research Article EAI.EU

Measuring the Cost of Software Vulnerabilities

Afsah Anwar[1,*], Aminollah Khormali[1], Jinchun Choi[1,2], Hisham Alasmary[1,3], Sung J. Choi[1], Saeed Salem[4], DaeHun Nyang[5], David Mohaisen[1]

[1]University of Central Florida, Orlando, FL 32816, USA
[2]Inha University, Incheon, Republic of Korea
[3]King Khalid University, Abha, Saudi Arabia
[4]North Dakota State University, Fargo, ND, USA
[5]Ewha Womans University, Seoul, Republic of Korea

Abstract

Enterprises are increasingly considering security as an added cost, making it necessary for those enterprises to see a tangible incentive in adopting security measures. Despite data breach laws, prior studies have suggested that only 4% of reported data breach incidents have resulted in litigation in federal courts, showing the limited legal ramifications of security breaches and vulnerabilities. In this paper, 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 and to estimate vulnerability disclosure dates as a baseline for estimating the implication of software vulnerabilities. We further build a model for stock price prediction using the nonlinear autoregressive neural network with exogenous factors (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 prediction performance. Our analysis also shows that the effect of vulnerabilities on vendors varies, and greatly depends on the specific software industry. Whereas some industries are shown statistically to be affected negatively by the release of software vulnerabilities, even when those vulnerabilities are not broadly covered by the media, some others were not affected at all.

Received on 24 March 2020; accepted on 06 May 2020; published on 12 May 2020

Keywords: Vulnerability Economics, Stock Return Prediction, NVD.

doi:10.4108/eai.13-7-2018.164551

1. Introduction

Vulnerabilities in a software expose users to unwarranted environments leading to security and privacy issues [1]. These vulnerabilities can be a result of flaws or bugs in the software code base [2]. Defects can be due to limited unit testing, performance testing, or stress testing which render the software to behave in an unintended fashion, exposing the product and the users to risk alike. In an event of such a vulnerability, intuitively, users prefer vendors that take such defects with utmost priority, fix them, report them to their users, and keeping the users susceptibility in check. Failure to do so, can put the vulnerable vendors at risk, whereby the users seek different vendors, causing huge irreparable damage [3].

In practice, vulnerabilities have multiple costs associated with them. For example, a vulnerability leads to loss of trust by users, tarnished brand reputation, and ultimately results in the loss of customer-base. To deal with vulnerabilities, vendors also incur additional costs in the form of developer-hours spent on fixing them and redeploying those fixes. Consequently, vulnerabilities could be a direct cause of a vendor losing a competitive edge in the global market to vendors less prone to them. For example, a study by the National Institute of Standards and Technology (NIST) estimated that the US economy loses about $60 billion USD every year for patches development and redistribution, systems re-deployment, as well as direct productivity loss due to vulnerabilities [4].

*Corresponding author. Email: afsahanwar@knights.ucf.edu

To make matters worse, the number of security incidents and vulnerabilities have been growing at a rapid pace, leading to similar growth in resources required for fixing them. In 2012, for example, Knight Capital, a financial services 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* [5]. Losses from WannaCry (2017), a ransomware attack in over 150 countries affecting more than 100,000 groups, is estimated to be $4 Billion USD [6]. 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 [7]. With the deployment of software in critical infrastructure, vulnerabilities could have an overwhelming impact. For example, defects such as the loss of radio contact between the air traffic controller and the pilots due to unexpected shutdown of the voice communication system and crash of the backup system within a minute of turning it on, could cost lives [8].

The cost of vulnerabilities is a variable that does not depend only on the type of vulnerability, but also on the industry, potential users, and the severity of the vulnerability as seen by those users. For example, users of security or financial software are more likely to lose trust in their product, compared to general e-commerce applications. A more severe vulnerability is also more likely to impact a vendor than a minor software glitch. For example, a vulnerability that can be used to repeatedly launch a Denial of Service (DoS) attack could be viewed more severely by users than, say, an access control misconfiguration (e.g., 1-time access-token exposure). We note that while companies may have cyber insurance, they are still susceptible to losses due to vulnerabilities and the cost of vulnerability could be due to long term impact on brand. Although the immediate cost is bared by the insurance provider, these incidents will result in increased insurance cost.

For publicly-traded drug and auto vendors, Jarrell and Peltzman [9] show that recalling products have an impact on value. Conversely, even if software vendors are experiencing an increase in software vulnerabilities, researches have shown that software vendors may not suffer significant losses due to those vulnerabilities [10], or that revenue and products may increase concurrently. However, there are also underlying costs associated with each software vulnerability, as mentioned above, and those costs maybe invisible [10]. For example, Romanosky et al. [11] studied software-related data breaches in the United States and found that 4% of them resulted in litigation in federal courts, out of which 50% (2% of the original studied cases) were won. Considering no impact of vulnerabilities on vendors, as shown by prior work, the vendors do not seem to face any immediate effect on themselves, unlike the end users. In this work, we work on finding how vendors could be equally impacted inversely by vulnerabilities.

Contributions. We quantitatively analyze the loss faced by software vendors due to software vulnerabilities, through the lenses of stock price and valuation. To this end, this work has the following contributions. (i) An evaluation of all the publicly disclosed vulnerabilities from the National Vulnerability Database (NVD) and their impact on their vendors. (ii) An accurate method for predicting the stock price of the next day using the NARX Neural Network. (iii) Industry-impact correlation analysis, demonstrating that some industries are more prone to stock loss due to vulnerabilities than others. (iv) Vulnerability type analysis, indicating that different types have different powers affecting the stock return of a vendor.

Our work stands out in the following aspects, compared to the prior work (more in section 2). First, unlike the prior work, which is event-based (tracks vulnerabilities that are only reported in the press), we use a comprehensive dataset of disclosed vulnerabilities in the National Vulnerability Database (NVD). Additionally, data breaches are the events where a vulnerability in a vendor is exploited to gain access to its data storage with malicious intent. Per Spanos and Angelis [12], 81.1% of the prior work they surveyed was limited to security breaches, while we focus on all software vulnerabilities. Furthermore, per the same source, 32.4% of the prior work used Lexis/Nexis (database of popular newspapers in the United States) as their source, 24.3% used the Data Loss Archive and Database (data for privacy breach), 13.5% used CNET (technology website), and 13.5% used Factiva (global news database). In this study, we uniquely focus on using NVD. (ii) We design a model to accurately predict stock for the next day to precisely measure the effect of a vulnerability. Our approach outperforms the state-of-the-art approach using linear regression (e.g., while our mean-squared error (MSE) using ANN is below 0.6, using linear regression results in MSE of 6.24). (iii) Unlike the prior work, we did not exclude any vendors, as we considered publicly-traded vendors on nine different stock markets, namely, NASDAQ, NYSE, EPA, ASX, STO, NYSEAMERICA, TYO, CVE, and NSE. Spanos and Angelis [12] in their survey found that 83.8% of the surveyed work used vendors that traded in a US stock market, 13.5% used vendors from different countries and only 2.9% (1 out of 34 works) used firms traded in TYO (the leading stock exchange in Japan) [12].

Organization. The rest of the paper is organized as follows: In section 2, we re-visit the literature. In section 3, we present our approach to the problem. In section 4, we present our prediction model. In section 5, we evaluate the results. In section 6 we further comment on the statistical significance of our results,

followed by discussion, limitations and future work in section 7. We conclude the paper in section 8.

2. Related Work

Our work is an amalgam of different fields, where we connect the vulnerabilities to economic effect on a vendor. Perceptions often relate vulnerabilities to effect on end users. Little has been said and done from the vendor's perspective.

A lot of work has been done on software vulnerabilities and reported to the community. The area has been approached from different fronts making the topic multi-faceted, some of which we review below.

Effect on Vendor's Stock. Hovav and D'Archy [10], and Telang et al. [13] analyzed, in event-based studies, vulnerabilities and their impact on vendors. While Hovav and D'Archy have shown that market exhibit no signs of significant negative reaction due to vulnerabilities, Telang et al. showed that a vendor on average loses 0.6% of its stock value due to vulnerabilities. Goel et al. [14] pointed out that security breaches have an adverse impact of about 1% on the market value of a vendor. Campbell et al. [15] observed a significant negative market reaction to information security breaches involving unauthorized access to confidential data, but no significant reaction to non-confidential breaches. Cavusoglu et al. [16] showed that the announcement of Internet security breaches has a negative impact on vendors' market value.

Anwar et al. [17] analyzed the effect of vulnerabilities on vendors and demonstrated the impact depends on the products' industry sector. Gamero-Garrido et al. [18] characterized the effect of legal threats on vulnerability researchers and observed that 40% of the studied vendors allow academic researchers to evaluate their products, and 25% of security researchers stated they do not do so because they fear legal measures.

Vulnerability Analysis. Li and Paxson [19] outlined a method to approximate public disclosure date by scrapping reference links in NVD, which we use in this study. Nguyen and Massaci [20] pointed out that the vulnerable versions of data in NVD is unreliable. Christey and Martin [21] outlined caveats with the NVD data, also suggesting its unreliability. Romanosky et al. [22] found that data breach disclosure laws, on average, reduce identity theft caused by data breaches by 6.1%. Similarly, Gordon et al. [23] found a significant downward shift in impact post the September 11 attacks. Steinke et al. [24] presented a vulnerability management framework. Stock et al. [25] focused on notifying affected vendors about vulnerabilities.

Zhao et al. [26] conducted an empirical study on data from two web vulnerability discovery ecosystem to analyze their trends. Trinh et al. [27] proposed an algorithm-based string solver to identify vulnerabilities in web applications. Saha [28] extended an attack graph-based vulnerability analysis framework to include complex security policies for efficient vulnerability analysis. Zhang et al. [29] used data from NVD to predict time to next vulnerability and argued that NVD has a poor prediction capability. They also pointed out inconsistencies in NVD, e.g., missing version information, vulnerability release time, and obvious errors.

Sabottke et al. [30] proposed a Twitter-based exploit detector to identify which vulnerabilities are likely to be exploited. Homaei and Shahriari [31] analyzed vulnerabilities report between 2008 and 2014, and observed that security professionals can prevent 60% of them by focusing on just seven vulnerability categories. Horvath et al. [32] pointed out that CVSS metrics are more suitable for software products running in an IT-environment than products for personal use. Holm and Afridi [33] studied the reliability of CVSS through a survey of 384 experts, covering more than 3,000 vulnerabilities, and concluded that the outcome depends on the type of vulnerability. Allodi et al. [34] assessed the vulnerabilities by evaluating information cues that increase assessment accuracy. Johnson et al. [35] assessed the credibility of CVSS scoring data using a Bayesian method and found CVSS is quite reliable except for a few dimensions. They argued, by analyzing five vulnerability databases, that NVD is the most reliable for CVSS quality.

Financial Impact of Defects. Jarrell and Peltzman [9] analyzed the impact of recall in the drug and auto industries on vendors' stock value loss. Towards calculating the effect of a vulnerability, it is crucial to predict a hypothetical valuation of the stock in the absence of a vulnerability. Kar [36] suggested using Artificial Neural Network (ANN) as a reliable method for predicting stock value. Farhang et al. [37] suggested higher security investments in Android devices do not result in higher product prices on customers.

3. Methodology

The goal of this study is to determine the impact of publicly disclosed vulnerabilities on their vendors. Our dataset of publicly disclosed vulnerabilities is gathered from the information available on the National Vulnerability Database (NVD). Prior work have shown that product recall have an adverse impact on a vendor's stock [9]. Taking cue from the prior art, we consider the fluctuation in the stock price as a measure of the vulnerabilities' impact. To this end, we calculate the impact on the day a vulnerability was disclosed by a vendor and the days following to it, with respect to the predicted value of the stock on the day of vulnerability disclosure. However, we limit up to the third day of the public disclosure of the vulnerability to reduce the

likelihood of interference with factors that might affect the market value. The rest of this section explains in details the steps taken to achieve the above goal.

3.1. Data and Data Augmentation

The major repository for publicly disclosed vulnerabilities is NVD [38]. Therefore, we use NVD as a source dataset for our analysis. The dataset of publicly disclosed vulnerabilities is then augmented with the stock data from *Alpha Vantage* [39] for analysis of the impact. Fig. 1 summarizes, at a high-level, the flow of data creation, from the source of data to the final dataset. In a nutshell, we extract information from JSON files downloaded from the NVD, scrape through the reference links for each vulnerability provided by NVD to approximate the disclosure date of the vulnerability.

Overall, the data from the NVD ranges from 1998 to 2018, making our dataset range over 20 years. The 17.1K vendors from the NVD are then searched over the internet for their market and code. Using the vendor's stock market and code, we then gather the historical daily stock returns data for each of the vendors. We use the *Alpha Vantage* as the source of historical data. For all the vendors that exist in the *Alpha Vantage*, we analyze the impact of vulnerabilities in them on their stock returns.

National Vulnerability Database (NVD). NVD is a vulnerability database maintained by the National Institute of Standards and Technology (NIST) that serves as a one-stop listing of all the vulnerabilities reported to MITRE [40]. Analysts at NVD further analyze the vulnerabilities before inserting them into the database.

Consequently, NVD enlists the following information for each of the vulnerabilities: the Common Vulnerabilities and Exposures Identifier (CVE-ID), vendor, product, Common Vulnerability Scoring System (CVSS), published date, Common Weakness Enumeration Identifier (CWE-ID) [41], description, reference links, etc.. Additionally, the NVD uses both the versions, version 2 and version 3 [42, 43], of the CVSS (a widely used severity scoring technique).

CVSS version 3, released in the latter half of 2015 labels vulnerabilities as LOW, MEDIUM, HIGH, and CRITICAL, while version 2 classifies them into LOW, MEDIUM, and HIGH. Although version 3 has been adopted by the database, the NVD is yet to accept it throughout the dataset. The *vendor* attribute is the name of the vendors that has the vulnerability in their software, the *product* element is the name of the software that had the vulnerability, *CWE-ID* is the type of the vulnerability or the reason for the vulnerability, *description* contextualizes the vulnerability including the exploit conditions, *published date* is the date when the vulnerability entered the database, and the *reference links* are additional details about the vulnerability, such as, security advisory, vendor advisory, security thread, email thread, patch details, commits (in an event of a vulnerability in open-source vendors). Particularly, the reference links contain vulnerability details, such as date the vulnerability was reported to the vendor, date it was acknowledged, date it was patched, disclosure date, and other information like, products effected, etc.

Algorithm 1: Finding the public disclosure date.

```
function CVE-ID, reference link set maker (f);
  Input  : NVD JSON file
  Output: set, cve_link (key - CVE-ID)
Extract domains from URLs, and find the
  minimum number of domains that cover
  maximum number of vulnerabilities.
Observe the HTML response of the domains to
  build a scraper to extract the probable
  disclosure date.
Group the dates by vulnerabilities, and find
  disclosure date as the minimum of the dates.
function date formatter (date);
  Input  : generic date
  Output: returns a formatted date, as, YYYY-MM-DD
if disclosure date < published date from NVD then
    public disclosure date ← disclosure date
else
    public disclosure date ← published date from NVD;
end
```

Data Preprocessing and Augmentation. The NVD data provides data in XML or JSON format. The data is distributed in multiple files such that the each file represents vulnerabilities in a specific year. Altogether, our dataset, built upon these files, comprise of vulnerabilities reported to the NVD until May 21, 2018. Additionally, we also observe vulnerabilities that were inserted into the database, but were later removed from the database. Such vulnerabilities can be identified by the sight of a description prefixed by "REJECT:". Moreover, the rejected vulnerabilities do not have any other information. For our analysis, we disregard all those vulnerabilities. Finally, our dataset encompasses a total of 101,580 vulnerabilities.

The impact of a vulnerability can be felt on the date a vulnerability is disclosed to the public or on the days in its vicinity. Since the *published date* attribute captured in the NVD is the date when a vulnerability enters into the database and not the date when it was disclosed by a vendor, it is important to find the date when it was publicly disclosed. To do so, we scrape through the links present in the NVD and label the disclosure dates corresponding to each of the links (if present), similar to the approach taken by Li and Paxson [19]. For a vulnerability with multiple URLs, after labelling

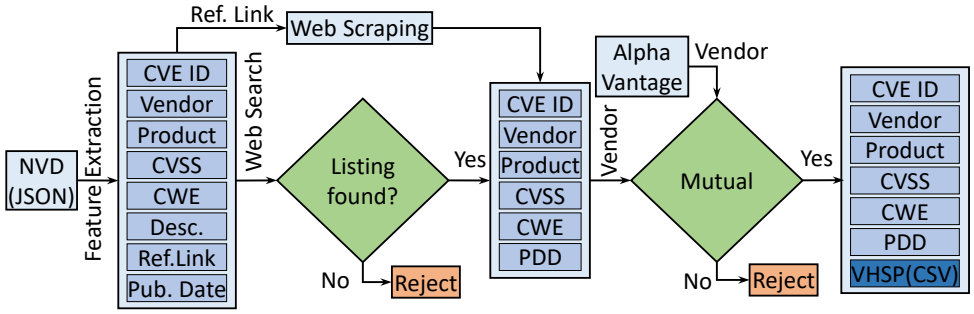

Figure 1. Dataset Creation Flow. *Desc.* is vulnerability description, *Ref. Link* is the link referring to vulnerability details, *Pub. Date* is the Published Date, *CVSS* is Common Vulnerability Scoring System metrics, *CWE* is the Common Weakness Enumeration identifier, *PDD* is the Public Disclosure Date, and *VHSP* is the Vendor Historical Stock Price downloaded of mutual vendors from Yahoo Finance.

all the URLs with corresponding disclosure dates, we consider the earliest date as the public disclosure date of the vulnerability. It should be noted that we ignore the links directing to patches, as the date of patching may differ from the disclosure date (disclosure is done after vulnerability patching), and the market can respond to public disclosure date.

Algorithm 1 summarizes the aforementioned steps towards determining the public disclosure date of a vulnerabilities. In particular it takes the JSON files from the NVD as inputs and extracts the CVE-ID, reference links, and the published date. We then scrape through individual URLs in the reference links and extract the public distribution date corresponding to each of them. We then group the dates by CVE-ID, followed by finding the minimum of the dates as described in steps 4 & 5. Lastly, the older of the dates (date from the links and published date from NVD) is approximated as the public disclosure date as detailed in steps 7 - 10.

We record redundant vendor names in our dataset, e.g., schneider-electric vs. schneider_electric, trendmicro vs. trend-micro, and palo_alto_networks vs. paloaltonetworks. We consolidate the various vendors under a consistent vendor name. For all the vendors in the above dataset, we further augment them by incorporating historical stock return data from Alpha Vantage.

Alpha Vantage. *Alpha Vantage* [39] is a community of researchers, engineers, and professionals, and is the leading provider of real-time and historical data on stocks, physical currencies, and digital currencies for free through an API. We found the market code for every vendor in our dataset, along with the company code. We then search for a list of vendors codes and the market they are traded on. We obtain a list of companies being traded on NYSE and another list of companies being traded on NASDAQ from Alpha Vantage. For the unlabelled vendors, we manually search over the internet for their codes and their markets. After compiling, we use the *Alpha Vantage*'s API to download historical daily stock data for individual vendors as a Comma Separated Values (CSV) file. The file contains five information attributes, namely, date, open, low, high, and close. The *date* attribute corresponds to the date on which the stock's performance is captured. The *open* and the *close* attributes are the stock values of the vendor on the given day at which the market opens and closes, respectively. The *low* and the *high* attributes correspond to low and high values of the vendor's stock price on the given day. Upon careful examination of the vulnerable vendors in our dataset, and successful augmentation with the Alpha Vantage dataset, we generate an overall dataset of 202 vendors.

Predicting Return. Calculating the impact of a vulnerability is dependent on the effect due to non-occurrence of the vulnerability. We, therefore, determine the stock return of a vendor for this hypothetical event by leveraging the machine learning-based prediction models. To this end, we feed the open, low, high, and close of the preceding days as inputs to our prediction model to predict the return for the day a vulnerability occurs. We describe our prediction model in section 4. We use the return (in an event of non-occurrence of vulnerability) as a baseline to compare with the actual return on the day of vulnerability occurrence.

Press. We contrast the impact of the vulnerabilities reported in the NVD with the impact of the vulnerabilities that capture the media attention. Towards this, we collect four vulnerabilities that were reported in the media. Specifically, we look for news with relating to *"software vulnerabilities"* in media outlets, such as *Forbes* and *ZDNet*, and capture four vulnerabilities for comparison, namely *Alteryx*, *Dow Jones*, *Viacom*, and *Equifax*.

3.2. Assessing Vulnerability's Impact

To assess the impact of vulnerabilities, we cluster our dataset by vendors. Additionally, there can be multiple

vulnerability disclosures on the same day. Moreover, we have historical daily-stock return data for each vendor. Consecutively, we find the impact of vulnerabilities on a particular day — which means that, in an event of multiple vulnerabilities on a day, it is impossible to calculate the impact of individual vulnerabilities. Therefore, for all such days, we determine the overall impact of vulnerabilities.

With this distinction in place, we further narrow down the vulnerabilities by disclosure date. At this point it is also important to remember that, while a vulnerability can be declared on any day of the week, except for weekends and holidays. Therefore, for every date with corresponding disclosed vulnerability that does not have stock information, the effect of the vulnerability can only be observed on the next operational day. Thus, we approximate the vulnerability to have occurred on the next operating day of the market.

Towards analyzing the cost of vulnerability against its corresponding vendor, we perform an *event-based* study. To do so, we realize a hypothetical event of non-occurrence of vulnerability, as described earlier. We call this event realization as the *Normal Return*. Additionally, the actual performance of the stock market, i.e., the stock return at the end of the day depending on the actual stock performance is called as the *Actual Return*. The comparison between the *Normal Return* and the *Actual Return* reflects on the abnormality in return for an event. To quantify the abnormality on a day due to vulnerability disclosure, we compare and contrast the *Normal Return* (R) and the *Actual Return* ($\bar{R}$). Moreover, the impact of a vulnerability can also be delayed, upon considering the reaction time of the consumers. To this end, we find the impact on the days following the vulnerability disclosure. Finally, we limit the impact calculation to the third day from disclosure, to limit the influence of external factors on market returns. In particular, we define Abnormal Return (AR) as the deviation of *Actual Return* from the *Normal Return*. Mathematically, AR on day i, AR_i, for $i \in \{1, 2, 3\}$, is determined, such that

$$\text{AR}_i = R_i - \bar{R},$$

where R_i is the *Actual Return* on day i, and $\bar{R}$ is the *Normal Return* on day i. We then calculate the percentage (%) of Abnormal Return on day i (PAR_i), where $i \in \{1, 2, 3\}$, as

$$\text{PAR}_i = \frac{\text{AR}_i \times 100}{R_i}$$

Finally, we calculate the Overall (%) Abnormal Return on day i (OAR_i), where $i \in \{1, 2, 3\}$. For vendor $\{V_1, \ldots, V_m\}$ with vulnerability $\{v_1, \ldots, v_n\}$, the PAR values for a vulnerability v_j are denoted by PAR_i^j for $i \in \{1, 2, 3\}$. We calculate OAR_i on day i for a vendor V_k as,

$$\text{OAR}_i^k = \sum_{j=1}^{n} \text{PAR}_i^j$$

Algorithm 2 shows the method used for calculating the impact of vulnerabilities by mapping the market-codes with vendor names then identifying public disclosure dates of the vulnerabilities followed by calculating the impact of the vulnerabilities on a disclosure date.

Algorithm 2: Impact calculation from prediction results. pdd here means public disclosure date

```
function vendor, code set maker (f);
  Input  : CSV file with vendor and its code
  Output: set, code_vendor (key - vendor code)
function vendor, pdd set maker (f);
  Input  : CSV file with vulnerability details from
           NVD
  Output: set, vendor_pdd (key - vendor name)
directory = path to the directory where
  prediction results of all the vendors is stored
For all the vendors with stock data
Create 3 lists each for date, ActualReturn, and
  NormalReturn
if date[i] == pdd then
    Impacts can be calculated as:
    day0 ← (ActualReturn[i] −
      NormalReturn[i]) ∗ 100/ActualReturn[i]
    day1 ← (ActualReturn[i] − NormalReturn[i +
      1]) ∗ 100/ActualReturn[i]
    day2 ← (ActualReturn[i] − NormalReturn[i +
      2]) ∗ 100/ActualReturn[i]
else
    if pdd not in date[i] then
        Compute j such that date[j] < pdd and
          date[j + 1] > pdd
        Impacts can be calculated as:
        day0 ←
          (ActualReturn[j + 1] − NormalReturn[j +
          1]) ∗ 100/ActualReturn[j + 1]
        day1 ←
          (ActualReturn[j + 1] − NormalReturn[j +
          2]) ∗ 100/ActualReturn[j + 1]
        day2 ←
          (ActualReturn[j + 1] − NormalReturn[j +
          3]) ∗ 100/ActualReturn[j + 1]
    else
        continue;
    end
end
```

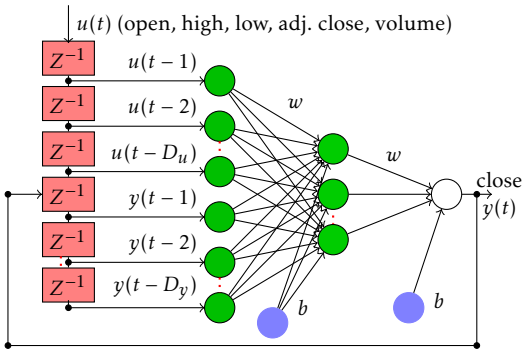

Figure 2. General Structure of the NARX Neural Network. u is the activation function, w is the weight, and b is the bias factor.

4. Prediction

To quantify the impact a vulnerability, we perform an event-based study. In particular, an event-based study compares and contrasts the Normal Return and the Actual Return. Our historical stock return dataset contains the Actual Return on a day. Defined as the return for non-occurrence of an occurred event and considering the trends in the past and ignoring the event, we determine the Normal Return (as explained in the previous section). We leverage the machine learning-based algorithms towards this determination. As aforementioned, our historic stock return dataset contains the following attributes for every vendor: date, open, close, high, and low. These attributes are then fed as features to our prediction models.

Recognizing the nonlinear behavior of the returns, therefore, we make use of nonlinear prediction techniques to analyze and predict the behavior [44]. Additionally, we perform data preprocessing to improve the performance of the machine learning algorithms.

We begin by normalizing the feature set for standardization. In other words, feature standardization projects the raw data into a new vector-space where each feature in the data has a mean and a standard deviation of zero and unit, respectively, i.e., every feature is represented in a space with more specific and richer realization. A widely accepted method for feature standardization is taking into account the mean and the standard deviation of features. Mathematically, the mapping transforms the feature vector x into z, where $z = \frac{x-\bar{x}}{\sigma}$, where $\bar{x}$ and σ, are the mean and standard deviation of the original feature vector x, respectively. These features are used as input to our machine learning-based prediction model to predict the Normal stock returns. In particular, we use the NARX Neural Network. To draw a parallel with the prior work and for comparison, we also use a linear regression model for prediction.

Table 1. NARX parameter settings.

Parameter	Value
Number of input neurons	Five
Number of output neurons	One
Transfer functions	tansig (hidden layer) purelin (output layer)
Training, validation, testing	70%, 15%, and 15%
Evaluation function	Mean squared error
Learning Algorithm	Levenberg-Marquardt

4.1. NARX Neural Network

The NARX neural network, generally applied for prediction of the behavior of discrete-time nonlinear dynamic systems, is one of the most efficient tools for predicting the behaviour [44]. Among the unique characteristics of NARX is its ability to provide accurate forecasts of the stock values by exploiting an architecture of recurrent neural network with limited feedback from the output neuron. When compared to other architectures which consider feedback from both hidden and output neurons, NARX is shown to be more efficient and also yields better results [45]. Extrapolating the NARX neural network model as per our needs, we determine the Normal Return, $y(t)$, on the day t, where t is the day on which a vulnerability occurred in the vendor. The Normal Return, $y(t)$, is regressed on previous values of the output and exogenous input, and is represented as following model:

$$y(t) = f[u(t-1), ..., u(t-d_n); y(t-1), ..., y(t-d_n)],$$

where $u(t-i)$, where $i \in [1, d_n]$, i.e., the input (low, high, open) values from historical return value, d_n is the number of days before the day the vulnerability occurs that is considered as input to the model for training. Additionally, $y(t-i)$ is the Actual Return for the corresponding input $u(t-i)$. The lag d_n is the exogenous inputs and output of the system, and the function f is multi-layer feed forward network. The general architecture of the NARX neural network is shown in Fig. 2.

For every vendor in the dataset, we divide the historical return data into training, validation and test subsets. In particular, the training, the validation, and the test subsets are selected as 70%, 15%, and 15% of the dataset respectively. The training data is used to train a predictive model. Additionally, the Mean Squared Error (MSE) is used as a parameter to evaluate the performance of the models. The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{t_i} - y_{p_i})^2,$$

where n is the number of samples. y_t and y_p represent the Actual Return and the corresponding Normal Return value on the day of vulnerability, respectively. A feed-forward neural network with one hidden layer has been used as a predictor function of the NARX. Levenberg-Marquardt (LM) back-propagation learning algorithm [46] is employed to tune the weights of the neural network. The specifications of the proposed NARX neural network are presented in Table 1.

4.2. Baseline for Comparison: ARIMA

In addition to the NARX neural network model, and to build a parallel with prior work, we use linear regression to determine the Normal Return. Towards this, we use one of the most popular time series prediction models, the Autoregressive Integrated Moving Average (ARIMA) model [47]. Particularly,

using linear regression we conduct the prediction for the stock return of one vendor, namely, Adobe, to determine the Normal Return. Conceptually, the AR portion of the ARIMA signifies that the variable to be predicted is regressed on its past values. Additionally, the MA portion in the ARIMA model indicates that the error in the regression model is a linear combination of error values in the past. The ARIMA model with external regressors, x, for one-step ahead prediction is represented by

$$y_p(t) - \phi_1 y_t(t-1) = \mu - \theta_1 e(t-1) + \beta(x(t) - \phi_1 x(t-1)),$$

where y_p and y_t are the Normal and Actual stock return, respectively. μ is a constant, while the θ, and the ϕ are the MA coefficient and the AR coefficient values, respectively.

The results are shown only for Adobe and for the rest of the vendors only the MSE is shown in Table 3. Fig. 3 depicts the Actual and Normal stock returns. The low value of the error strongly suggests that the NARX model can forecast the stock values with high accuracy. In addition, the error histogram, as shown in Fig. 5, represents the performance of the predictor. We observe that the majority of the instances are forecasted precisely, and with very small prediction error. In Fig. 4, although visual representation suggests a weakness of fit with ARIMA in prediction the stock values, the difference in the value of MSE for these two models, 6.42 for ARIMA and 0.59 for NARX, quantitatively justifying the goodness of the proposed method over the existing methods in the literature.

5. Results

We perform our experiment over a large dataset of publicly available vulnerabilities, encompassing all possible publicly traded vendors. To begin with, we augment the dataset, label the extracted vendors with their market and their code, and eliminate non-public companies. We used Google and Yahoo Finance to label the vendors. Among the publicly available companies, we use the Alpha Vantage's API to collect historical data corresponding to them. While predicting the Normal Return at the closure of the market for the day when a vulnerability occurred, we consider the last 50 days the stock market was active (this excludes holidays). Thus, we reject all those that have less than 50 active entries. Additionally, we exclude vendors for which we can not gather historical data.

We then determine the impact of vulnerabilities on a vendor after grouping them by date, meaning that multiple vulnerabilities could correspond to a single date. Therefore, the effect we see in Table 3 could be due to one or more vulnerabilities. For every vulnerability disclosure date and vendor, we calculate % Abnormal Return on days 0, 1, and 2 (AR_1, AR_2, and AR_3 respectively as described above).

We present the results in Table 3, including the normalized MSE, count of the vulnerabilities, and Abnormal Return on days 1, 2, and 3 for every vendor. We observe that 155 out of the 202 vendors suffer an adverse impact of vulnerabilities on their returns on either of the days.

Table 2 represents a breakdown of vendors by industry and the likelihood of their stock being impacted by vulnerabilities. To do so, we divide the analyzed vendors into 8 categories: software, device, networking, security, consumer product, conglomerate, electronics & hardware, and finance industry. In particular, the software industry contains software vendors such as Adobe, Atlassian, Google, VMware, Sap, Oracle, Redhat, etc. The device industry includes Advantech and Apple. The networking industry includes Cisco, Citrix, Netgear, etc. The security industry includes Fortinet, Juniper, Paloalto Networks, Symantec, etc. The consumer product industry includes Rockwell Automation, Baidu, Osram, Splunk, Schneider, Teradata, Facebook, Netapp, etc. The electronics & hardware industry includes Lenovo, and Nvidia. Finally, the finance industry includes Bank of America, Equifax, Dow Jones etc.

Table 2. Per industry stock impact likelihood analysis.

Industry	Likeliness
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

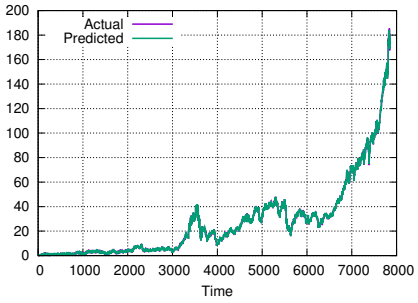

Figure 3. Actual vs. Predicted: NARX.

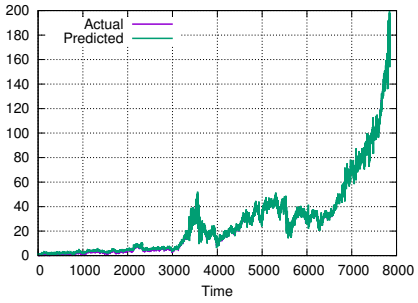

Figure 4. Actual vs. Predicted: ARIMA.

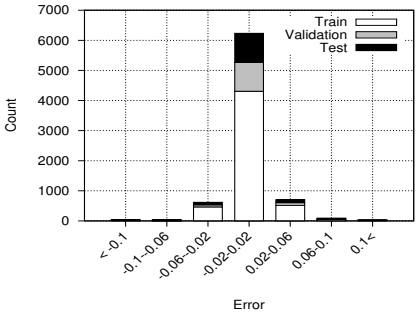

Figure 5. Error Histogram of Adobe Stock.

To assign a likelihood of an industry's returns being impacted by vulnerabilities, we use Highly-Likely when the number of vendors with stocks affected negatively by the vulnerabilities in the given industry is larger than those not affected, Less-Likely otherwise; we use Equally-Likely when the number of vendors affected equals the number of vendors not affected.

To investigate the industries that are less likely to be affected by vulnerabilities at the vendor level, we examine vulnerabilities from 10 such vendors. For every vendor, we observed that there are a few dates which have a vulnerability, where that vulnerability does not have any visible impact on the return. In other words, these dates see a surge in the vendors' stock returns, despite vulnerability occurrence, thereby nullifying the impact of vulnerabilities on the other days. For a better understanding, we then examine the description of the vulnerabilities leading to the following observations:

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 returns negatively have a combination of version 3 label of HIGH or CRITICAL, and a description containing phrases such as "denial of service", "allows remote attacker to read/execute", "allows context-dependent attackers to conduct XML External Entity XXE attacks via a crafted PDF", and "allows context-dependent attackers to have unspecified impact via an invalid character". Additionally, vulnerabilities description such as "allows authenticated remote attacker to read/execute", "remote attackers to cause a denial of service", and "allows remote attackers to write to files of arbitrary types via unspecified vectors" have little (on days 0, 1, and 2) to no effect on the returns. Therefore, we can conclude that vulnerabilities involving unauthorized accesses have a higher cost, seen in their detrimental effect on the stocks.

3. Vulnerabilities with phrases such as "local users with access to" and "denial of service" in the description have no impact on the stock. Therefore, DoS attacks lacking confidentiality factor lead to no impact on stock value.

Severity effect. To study the significance of our results, we evaluate the impact of vulnerabilities and their severity. To do so, we first conduct a correlation (Pearson correlation coefficient) analysis between the impact and severity of the vulnerabilities. As stated earlier, a public distribution date can have multiple vulnerabilities corresponding to it, therefore, the impact of a particular vulnerability is impossible to quantify. Our prior manual effort hints at the higher impact of more severe vulnerabilities.

We start by assigning a severity index to every public distribution date. In doing so, we prioritize the CVSS version 3 scoring system over version 2. For vulnerabilities that do not have version 3 labels, we consider the version 2 label as their tag. Moreover, we prioritize a more severe vulnerability over less severe vulnerability. More precisely, if a public distribution date has a critical, high, medium, or low vulnerability, we consider the date to contain a critical severity vulnerability, and label it as critical.

Having labeled the public distribution dates with severity labels, we examine the correlation between impact and the severity labels. We observe a low positive correlation between the severity and the impact on the individual days. In particular, the correlation coefficient between severity and the impact on the day a vulnerability surfaces and the following two days is 0.119, 0.115, and 0.11, respectively. Although, we observed a positive correlation, the low magnitude of the correlation indicates that the we cannot consider the the severity of vulnerabilities as an indicator of impact on vendors. This shows the change in the role of severity labels to the overall impact of vulnerabilities on its vendors.

Table 3. Results for each Vendor. Vul.: vulnerability count, OAR_1, OAR_2, and OAR_3 stand for the average effect at day 1, 2, and 3 (percent), respectively. [(2)] Vendor names are abbreviated as follows: ls.=Lattice Semiconductor, ra.=rockwellautomation, akt.=Akamai Technologies, en.=Extreme Networks, johnson.=johnsoncontrols. ▲: vulnerabilities had no overall impact on vendor's stock value. ▼: the stock of the vendor were impacted.

Vendor	MSE	Vul.	OAR_1 [(1)]	OAR_2 [(1)]	OAR_3 [(1)]	Vendor	MSE	Vul.	OAR_1 [(1)]	OAR_2 [(1)]	OAR_3 [(1)]
microsoft	2.50E-03	5540	▲3.75	▲0.98	▼7.83	baidu	1.55E-02	9	▲0.65	▲2.03	▲2.21
oracle	2.62E-03	4547	▲6.65	▲11.45	▲34.82	belden	1.95E-03	9	▼0.06	▼0.15	▼0.13
apple	3.06E-03	3982	▲1.51	▼7.12	▲0.08	cherokee	2.46E-03	9	▼0.27	▲0.28	▼0.35
cisco	1.38E-02	3388	▼2.59	▼183.50	▼248.42	imperva	1.46E-02	9	▼0.42	▼0.63	▼1.28
adobe	2.44E-03	2533	▲4.90	▼2.72	▼7.30	natus	2.14E-03	9	▼0.02	▲0.05	▲0.01
hp	3.24E-03	1527	▲11.01	▲16.55	▲20.77	canon	7.22E-03	8	▲0.40	▲0.31	▲0.37
apache	6.28E-03	1018	▲15.50	▲7.41	▼8.57	eaton	1.31E-02	8	▲0.46	▼0.17	▲0.40
redhat	9.08E-04	743	▲3.86	▲5.17	▲3.57	eclipse	1.56E-03	8	▲0.03	▲0.04	▼0.08
symantec	8.54E-03	435	▲0.70	▼1.54	▼2.37	en.	1.21E-02	8	▲0.25	▲0.15	▲0.12
sap	2.34E-03	431	▲1.70	▼0.60	▼0.70	mitel	5.31E-03	8	▼0.26	▼0.49	▼0.49
qualcomm	8.47E-03	334	▲0.65	▲5.29	▲3.14	unisys	6.01E-03	8	▼1.18	▼0.30	▼0.80
vmware	6.08E-03	307	▲11.55	▼4.02	▲3.64	idera	3.28E-03	7	▲0.33	▲1.13	▲2.29
juniper	5.59E-03	255	▼1.34	▼1.82	▼1.22	collector	2.44E-03	6	0.00	▲0.01	▲0.09
ca	7.69E-03	216	▼5.18	▼16.72	▼24.09	lantronix	5.99E-03	6	▼0.16	▼0.16	▼0.07
realnetworks	1.73E-03	203	▲1.35	▲6.55	▲5.41	netease	1.25E-03	6	▲0.06	▲0.12	▲0.23
citrix	2.41E-03	196	▲0.83	▲0.61	▲0.33	supermicro	5.21E-03	6	▲0.08	▼0.02	▲0.14
f5	2.29E-03	144	▲0.87	▲0.63	▼0.01	activision	2.59E-03	5	▼0.02	▼0.02	▼0.86
nvidia	2.72E-03	130	▼0.71	▼1.03	▼0.85	ada	7.55E-03	5	▼0.32	▼0.82	▼1.18
fortinet	8.89E-03	125	▼0.25	▼20.49	▼13.50	adtran	7.71E-03	5	▲3.24	▲0.17	▲14.43
intel	4.99E-03	100	▼11.08	▼11.20	▼3.76	akt.	2.75E-03	5	▼1.11	▼1.32	▼2.23
checkpoint	6.15E-03	75	▼0.17	▲0.73	▼0.11	electronicarts	3.76E-03	5	▼0.93	▲1.81	▲1.25
dell	1.12E-02	74	▲0.67	▲1.31	▼0.04	ericsson	2.74E-03	5	▼0.22	▼0.16	0.00
netgear	4.11E-03	70	▲0.36	▲0.81	▲1.78	netcomm	5.39E-03	5	▲0.16	▲0.23	▲0.22
xerox	5.94E-03	62	▼0.41	▼0.32	▼1.54	tcp	3.81E-03	5	▼0.06	▼0.08	▼0.04
netapp	8.89E-03	49	▼2.06	▼4.78	▼0.09	verisign	3.77E-03	5	▼0.03	▼0.06	▼0.21
splunk	8.43E-03	38	▼5.51	▼5.32	▼3.14	vonage	2.72E-03	5	▲0.07	▼0.08	▼0.11
nokia	9.85E-03	36	▲2.52	▼0.25	▲4.03	3ds	4.67E-03	4	▲0.09	▲0.13	▲0.13
opentext	9.02E-03	34	▼0.33	▼0.70	▼0.95	atom	2.55E-02	4	▼0.22	▼0.15	▼0.23
bmc	2.10E-02	32	▼0.27	▲2.09	▲1.59	bottomline	4.47E-03	4	▼0.05	▼0.03	▼0.18
quest	9.40E-03	31	▲0.27	▲0.19	▲0.11	bt	2.00E-03	4	▼0.09	▼0.41	▼1.51
sony	2.20E-03	30	▼4.19	▼1.29	▼1.07	carbonblack	5.65E-02	4	▼0.11	▼0.11	▼0.11
motorola	1.94E-03	29	▼0.06	▼0.04	▼0.01	gree	4.30E-03	4	▲0.01	▼0.02	0.00
autodesk	2.82E-03	25	▼0.04	▲0.03	▼0.36	johnson.	7.42E-03	4	▼0.02	▼0.03	▼0.08
ez	6.49E-03	21	▼1.31	▼0.77	▼2.36	linecorp	4.54E-02	4	▲0.42	▼0.05	▲0.06
cvs	1.65E-03	20	▼0.57	▼0.42	▲0.02	tylertech	4.65E-04	4	▼0.01	▼0.01	▲0.02
emerson	9.77E-03	20	▼0.76	▲1.16	▲2.54	arista	3.69E-03	3	▲0.23	▲0.23	▲0.23
honeywell	8.04E-04	20	▼0.06	▲0.29	▲0.42	commvault	2.83E-03	3	▼0.42	▼0.04	▼0.27
philips	8.55E-03	18	▲0.04	▼0.54	▲0.94	kingston	1.59E-03	3	▼0.16	▼0.10	▼0.09
rapid7	1.14E-02	16	▼1.83	▼0.86	▲0.69	ptc	1.77E-03	3	▲0.02	▼0.16	▲0.06
sierrawireless	5.33E-03	16	▼0.73	▲0.64	▲0.40	redwood	1.96E-03	3	0.00	▼0.01	▼0.02
facebook	1.17E-03	14	▲0.15	▲0.12	▲0.05	a10networks	8.79E-03	2	▲0.01	▼0.27	▼0.71
rpm	8.81E-04	14	▼0.15	▲0.02	▲1.32	associated	2.97E-03	2	▲0.02	▲0.10	▲0.05
amazon	4.56E-04	13	▼0.55	▼0.42	▼0.25	cavium	3.06E-03	2	▼0.08	▼1.25	▼1.58
intuit	1.44E-03	13	▲1.52	▲1.37	▲1.69	counterpath	1.92E-02	2	▼0.01	▼0.01	▲0.02
paypal	2.81E-03	13	▼0.04	▲0.03	▲0.08	dteenergy	9.36E-04	2	▲0.02	▲0.01	▲0.02
seagate	2.32E-03	12	▼0.37	▼1.61	▼2.31	flowers	1.52E-02	2	0.00	▲0.05	▲0.13
yandex	8.58E-03	12	▲0.38	▲0.42	▲0.52	fsi	1.13E-02	2	▼0.07	▼0.07	▼0.10
technicolor	3.76E-01	11	▲3.60	▼1.56	▲0.75	gopro	7.79E-03	2	▲0.05	▲0.03	▼0.08
broadcom	8.87E-04	10	0.00	▼0.04	▼0.08	ipass	2.43E-03	2	▼0.01	▼0.02	▼0.02

Size effect. From Table 3 it is clear that vendors with higher number of vulnerabilities are less impacted by the vulnerabilities. However, as we go down the table and to Table 4, i.e., vendors with fewer vulnerabilities, we find that the vendors appear more susceptible to vulnerabilities. Moreover, the vendors up in the table are large and well-known companies with many products, while the vendors lower in the table are less known. This trend can be explained by the fact that the companies appearing earlier in the table have multiple products, thus the impact of vulnerability on a product does not have a significant effect on the company (vendor) as a whole.

Table 4. Results for vendors with low vulnerability count. Symbols mean the same as in Table 3. [(2)] Vendor names are abbreviated as follows: bankofam.=Bank of America, digii.=Digi International Inc, dent.=Dentsply Sirona, tableau.=Tableau Software, agilent.=Agilent Technologies, persist.=Persistent Systems. ▲: vulnerabilities had no overall impact on vendor's stock value. ▼: the stock of the vendor were impacted.

Vendor	MSE	Vul.	OAR_1	OAR_2	OAR_3	Vendor	MSE	Vul.	OAR_1	OAR_2	OAR_3
marvell	6.11E-03	2	▲0.05	▲0.02	▼0.02	fusion	8.15E-03	1	▼0.03	▼0.06	▼0.04
mckesson	7.27E-04	2	▼0.07	▼0.04	▼0.03	garmin	5.86E-03	1	▼0.01	▼0.05	▼0.11
microchip	4.22E-03	2	0.00	▼0.04	▼0.05	halliburton	6.43E-03	1	▼0.22	▼0.26	▼0.19
micronet	1.27E-02	2	▼0.05	▼0.05	▼0.05	honda	5.42E-03	1	▼0.01	▲0.01	▲0.05
nationalinstruments	1.68E-02	2	▼0.07	▲0.15	▲0.12	ironmountain	1.71E-02	1	▼0.36	▼0.22	▼0.10
newrelic	4.54E-03	2	▼0.36	▲0.12	▲0.99	kirby	2.73E-03	1	▲0.01	▲0.01	▲0.06
nice	2.51E-03	2	▲0.46	▲0.24	▲0.69	kronos	6.18E-03	1	0.00	▲0.09	▲0.07
nxp	1.73E-03	2	▼0.01	▼0.02	▼0.01	logit	5.81E-03	1	▼0.04	▼0.04	▲0.02
persistentsystems	2.67E-01	2	▼0.40	▼0.44	▼0.41	magic	8.87E-04	1	0.00	▼0.03	▼0.01
radware	5.50E-03	2	▼0.05	▼0.03	▼0.34	maximus	6.46E-03	1	▼0.01	▼0.03	▼0.02
realpage	4.04E-03	2	▼0.14	▲0.21	▲0.49	medtronic	3.42E-03	1	▼0.03	▼0.03	▲0.03
renren	1.06E-02	2	▼0.59	▼0.38	▼0.50	mercadolibre	2.56E-03	1	▼0.07	▼0.01	▲0.30
sina	3.58E-03	2	▲0.02	▲0.15	▲0.31	merit	4.14E-03	1	▲0.24	▲0.34	▲0.92
sprint	1.33E-03	2	▼0.03	▼0.05	▼0.09	mobileiron	9.66E-03	1	▲0.02	▼0.05	▼0.03
square	3.71E-03	2	▲0.01	▲0.03	▲0.04	navis	9.85E-03	1	0.00	▲0.02	0.00
summerinfant	4.64E-03	2	▼0.03	▼0.02	▼0.05	pebble	3.39E-03	1	▲0.02	▼0.02	▼0.07
thomsonreuters	8.16E-03	2	▼0.06	0.00	▲0.02	pegasystems	2.26E-03	1	▼0.06	▼0.02	▲0.04
tivo	9.20E-03	2	▼0.02	▼0.03	▲0.08	pico	2.97E-03	1	▲0.02	▼0.15	▼0.47
tmobile	1.32E-03	2	▼0.29	▼0.34	▼0.27	pnc	6.96E-03	1	▲0.04	▼0.01	▼0.01
trimble	1.01E-02	2	▼0.34	▼0.45	▼0.10	purestorage	1.47E-02	1	▼0.20	▼0.20	▼0.22
twitter	7.15E-03	2	▼0.04	▲0.05	▼0.05	raytheon	5.25E-04	1	0.00	▲0.05	▼0.01
vasco	5.96E-03	2	▲0.04	▼0.28	▼0.32	salesforce	5.08E-03	1	▲0.02	▲0.03	0.00
vodafone	6.56E-03	2	▼0.10	▼0.08	▼0.02	sears	1.91E-03	1	▲0.03	▲0.02	0.00
webster	2.12E-03	2	▲0.06	▼0.50	▼0.42	smithmicro	5.10E-03	1	▲0.03	▲0.03	▲0.02
westerndigital	1.11E-03	2	▼0.04	▼0.07	▼0.07	southwest	1.30E-03	1	▼0.07	▼0.08	▼0.19
aerohive	1.20E-02	1	▼0.03	▼0.03	▼0.02	starbucks	3.37E-03	1	▲0.02	▲0.09	▲0.08
agilenttechnologies	5.64E-03	1	▲0.01	▼0.12	0.00	streamline	6.93E-03	1	▼0.60	▲0.84	▲0.50
bankofamerica	1.82E-03	1	▼0.01	0.00	▼0.01	synacor	9.42E-03	1	0.00	▼0.03	0.00
bb&t	8.69E-03	1	▼0.19	▼0.22	▼0.20	tableausoftware	1.42E-02	1	▼0.17	▲0.06	0.00
big	2.68E-03	1	▼0.19	▼0.03	▼0.12	tellurian	4.65E-03	1	▼0.03	▼0.08	▼0.12
broadvision	1.79E-03	1	▲2.30	▲4.65	▲4.89	tesla	2.05E-03	1	▲0.07	▼0.21	▼0.18
cern	9.84E-03	1	▲0.02	▼0.02	▼0.14	titan	3.66E-03	1	0.00	▼0.05	▼0.01
cgi	2.11E-03	1	▲0.04	▲0.04	▲0.04	tucows	7.67E-04	1	0.00	0.00	0.00
dasanzhone	3.94E-03	1	▲0.04	▼0.01	▲0.05	ubiquitinetworks	5.15E-03	1	▲0.03	▲0.01	▲0.02
dentsplysirona	6.00E-03	1	▼0.02	▼0.03	▲0.01	urban	4.47E-03	1	▲0.01	▲0.03	▲0.02
digiinternationalinc	1.11E-02	1	▲0.14	▲0.12	▲0.12	verifone	6.73E-03	1	0.00	▲0.18	▼0.21
dish	6.32E-03	1	▲0.01	▲0.02	0.00	vivint	7.40E-03	1	▼8.23	▼9.36	▼7.94
dolby	4.71E-03	1	▲0.05	▲0.01	▲0.03	vivo	5.81E-02	1	▲0.39	▲0.39	▲0.39
energizer	8.19E-03	1	▼0.05	▼0.06	▲0.29	wellsfargo	5.33E-03	1	▼0.07	▼0.04	▼0.06
ford	2.83E-03	1	▼0.06	▼0.09	▼0.02	westpac	3.79E-03	1	0.00	▼0.01	0.00
Alteryx	4.80E-02	1	▼0.61	▼2.18	▼7.70	Viacom	2.30E-03	1	▼1.60	▲0.60	▼0.62
Dow Jones	3.50E-04	1	▼0.08	▼0.34	▼0.03	Equifax	4.90E-04	1	▲1.52	▼14.02	▼24.19

We followed the same steps for the vulnerabilities gathered from the press. We found that these vulnerabilities have an adverse effect on vendor stock in almost every case.

6. Statistical Significance

To understand the statistical significance of our results, we use the confidence interval of the observations as a guideline. Particularly, we measure the statistical confidence of the overall effect of vulnerabilities corresponding to a vendor on days 1, 2, and 3, respectively. Table 5 shows the confidence intervals (lower and upper limit) on days 1, 2, and 3, measured with 95% confidence.

95% Confidence Interval. 95% Confidence Interval (CI) is a range that contains the true mean of a population with 95% certainty. For a smaller population, the CI is almost similar to the range of the data, while only a tiny sample of data lies within the confidence interval for a large population. In our study, we have noticed that our data populations are diverse. While some vendors have a small number of samples, others have a larger number of samples. For example, Fig. 6 – Fig 8 show the distribution of observations of effect for multiple example vendors

Table 5. Statistical confidence for each Vendor. OAR_1, OAR_2, and OAR_3 stand for the average effect at day 1, 2, and 3 (percent), respectively. CI_i is the confidence interval for day_i, where i $\epsilon\{1, 2, 3\}$. [(2)] Vendor names are abbreviated as in Table 3 and Table 4.

Vendor	CI_1		CI_2		CI_3		Vendor	CI_1		CI_2		CI_3	
	Low	High	Low	High	Low	High		Low	High	Low	High	Low	High
oracle	-0.01	0.07	-0.03	0.13	-0.13	0.45	mitel	-0.31	0.05	-0.28	-0.21	-0.27	-0.22
apple	-0.15	0.03	-0.15	0.02	-0.28	0.07	unisys	-0.43	0.09	-0.36	0.28	-0.54	0.31
cisco	-1.35	4.41	-1.16	2.62	-0.90	1.61	idera	-0.06	0.29	-0.26	1.02	-0.47	2.00
adobe	-0.01	0.04	-0.03	0.01	-0.07	0.03	collector	0.00	0.00	0.01	0.01	0.09	0.09
hp	-0.01	0.09	-0.02	0.15	-0.06	0.32	lantronix	-0.08	0.00	-0.10	0.02	-0.06	0.02
apache	-0.02	0.08	-0.04	0.07	-0.06	0.03	netease	0.06	0.06	0.12	0.12	0.23	0.23
redhat	0.00	0.02	0.00	0.03	-0.01	0.03	supermicro	0.02	0.06	-0.09	0.07	0.06	0.08
symantec	-0.01	0.02	-0.03	0.01	-0.03	0.01	activision	-0.03	0.02	-0.07	0.06	-0.55	0.12
sap	0.00	0.02	-0.02	0.01	-0.04	0.03	ada	-0.26	0.05	-0.40	-0.15	-0.60	-0.19
qualcomm	-0.07	0.10	0.01	0.26	-0.03	0.20	adtran	-0.71	2.87	-0.07	0.18	-2.62	12.24
vmware	-0.01	0.20	-0.17	0.10	-0.08	0.14	akt.[(2)]	-0.59	0.04	-0.78	0.12	-1.43	0.32
juniper	-0.03	0.00	-0.05	0.00	-0.04	0.02	cray	0.00	0.00	0.04	0.04	0.03	0.03
ca	-0.43	0.11	-0.90	0.14	-1.69	0.29	ea.[(2)]	-0.80	0.43	-0.18	0.91	-0.06	0.56
realnetworks	-0.06	0.09	0.01	0.15	-0.03	0.16	ericsson	-0.18	0.03	-0.12	0.02	-0.04	0.04
citrix	0.00	0.02	-0.01	0.02	-0.02	0.02	gsi	-0.09	-0.09	-0.03	-0.03	0.20	0.20
f5	-0.01	0.03	-0.01	0.02	-0.01	0.01	netcomm	0.03	0.08	0.06	0.10	0.06	0.09
nvidia	-0.05	0.00	-0.07	0.00	-0.07	0.01	tcp	-0.10	0.06	-0.08	0.03	-0.08	0.05
fortinet	-0.26	0.25	-1.18	0.53	-0.79	0.36	verisign	-0.02	0.01	-0.02	0.00	-0.07	-0.02
intel	-0.37	0.04	-0.36	0.03	-0.12	0.01	verizon	-0.03	0.09	-0.06	0.10	-0.08	0.06
atlassian	-0.07	0.18	-0.20	0.69	-0.20	0.83	vonage	-0.01	0.04	-0.05	0.02	-0.16	0.11
checkpoint	-0.03	0.02	-0.03	0.06	-0.06	0.05	3ds	-0.03	0.08	-0.06	0.15	-0.05	0.14
dell	-0.12	0.27	-0.03	0.33	-0.22	0.21	atom	-0.22	-0.22	-0.15	-0.15	-0.23	-0.23
ra.[(2)]	-0.03	0.06	-0.04	0.09	-0.05	0.10	bottomline	-0.04	0.02	-0.06	0.05	-0.11	0.02
netgear	-0.04	0.06	-0.05	0.10	-0.07	0.18	bt	-0.18	0.14	-0.21	0.01	-0.74	-0.02
xerox	-0.07	0.03	-0.08	0.05	-0.21	0.05	carbonblack	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
netapp	-6.40	1.92	-8.06	2.31	-9.07	2.86	ceragon	-0.03	-0.02	-0.05	-0.03	-0.10	-0.03
splunk	-0.96	0.38	-1.29	0.73	-1.11	0.78	gree	-0.01	0.02	-0.03	0.01	-0.02	0.02
nokia	-0.28	0.48	-0.39	0.37	-0.48	0.79	johnsoncontrols	-0.02	-0.02	-0.03	-0.03	-0.08	-0.08
quest	0.01	0.13	0.01	0.08	0.01	0.05	tylertech	-0.01	-0.01	-0.01	-0.01	0.02	0.02
sony	-0.54	0.16	-0.21	0.09	-0.29	0.19	arista	-0.08	0.24	-0.09	0.25	-0.08	0.24
motorola	-0.01	0.00	-0.01	0.01	-0.01	0.01	kingston	-0.16	-0.16	-0.10	-0.10	-0.09	-0.09
autodesk	-0.03	0.03	-0.05	0.05	-0.12	0.07	kyocera	-0.16	-0.16	0.01	0.01	-0.14	-0.14
ez	-0.37	0.08	-0.21	0.03	-0.52	0.00	nuance	-0.11	0.00	-0.11	-0.08	-0.11	-0.08
arris	-0.20	0.65	-1.50	0.39	-3.93	1.14	ptc	-0.02	0.03	-0.14	0.03	-0.03	0.07
cvs	-0.11	0.03	-0.11	0.06	-0.07	0.07	redwood	0.00	0.00	-0.01	-0.01	-0.02	-0.02
honeywell	-0.02	0.01	-0.02	0.07	-0.01	0.08	associated	0.00	0.02	-0.05	0.15	-0.05	0.10
philips	-0.04	0.05	-0.30	0.12	-0.14	0.45	changyou	-0.26	0.15	-0.13	0.17	-0.08	0.19
rapid7	-0.49	0.03	-0.35	0.14	-0.19	0.37	flowers	0.00	0.00	0.05	0.05	0.13	0.13
sierrawireless	-0.31	0.06	-0.03	0.24	-0.07	0.21	fsi	-0.07	-0.07	-0.07	-0.07	-0.10	-0.10
facebook	-0.03	0.07	-0.03	0.06	-0.03	0.05	gopro	0.05	0.05	0.03	0.03	-0.08	-0.08
amazon	-0.26	0.08	-0.19	0.05	-0.13	0.05	ls.[(2)]	0.00	0.03	0.00	0.01	-0.02	0.01
intuit	-0.10	0.86	-0.05	0.74	-0.01	0.85	mckesson	-0.09	0.02	-0.05	0.02	-0.05	0.01
paypal	-0.04	-0.04	0.03	0.03	0.08	0.08	microchip	-0.04	0.05	-0.09	0.05	-0.13	0.08
seagate	-0.12	0.03	-0.47	0.12	-0.70	0.18	micronet	-0.06	0.01	-0.06	0.01	-0.06	0.01
yandex	-0.06	0.24	0.00	0.21	-0.02	0.28	ni.[(2)]	-0.04	-0.03	0.01	0.14	-0.05	0.16
technicolor	-1.13	2.57	-0.52	-0.10	-0.48	0.78	newrelic	-0.36	-0.36	0.12	0.12	0.99	0.99
broadcom	-0.01	0.01	-0.03	0.01	-0.04	0.01	nice	0.46	0.46	0.24	0.24	0.69	0.69
baidu	-0.08	0.24	-0.11	0.62	-0.13	0.69	persist.[(2)]	-0.40	-0.40	-0.44	-0.44	-0.41	-0.41
microsoft	-0.01	0.01	-0.01	0.01	-0.02	0.01	tivo	-0.02	-0.02	-0.03	-0.03	0.08	0.08

and several vulnerabilities associated with each vendor. The shown histogram captures counts of the effect of vulnerabilities; the x-axis includes brackets of the effect (measured by OAR) and the y-axis captures the count for the given effect. The diversity of the effect is well-captured by the count distribution. High severity impact is seen in a vendor, where the counts are focused in negative side of the interval; whereas, lower (or no) impact is seen where the count focus is in the positive side. The confidence interval with 95% confidence for a

given population (distribution) can be calculated as,

$$\text{CI} = \left(\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}\right), \quad (1)$$

where $\bar{x}$ is the mean of the population, σ is the standard deviation, and n is the number of samples.

Putting it into perspective, while OAR_i, where $i \in \{1, 2, 3\}$, captures the overall effect of vulnerabilities corresponding to a vendor, the Confidence Interval (CI_i, where $i \in \{1, 2, 3\}$) gives the confidence for the effect to lie within its upper and lower bound. In Table 5, and by considering the data associated with Adobe, for example, we can say with 95% confidence that the confidence interval for the population, CI_i, contains the true mean, OAR_i. We also observe that:

1. OAR_i for the vendors in Table 3 and Table 4 are within their respective confidence intervals, which means that our results reported earlier are statistically significant.

2. The confidence intervals depict that chances of the overall impact of a vendor due to vulnerabilities to lie within their respective confidence intervals is 95%.

3. The true mean for the vendors with their confidence intervals bounded in negative intervals is likely to be negative. Thus, the probability for a vulnerability having a negative impact on days succeeding the day a vulnerability is disclosed on the vendor's stock returns is highly likely.

4. The confidence intervals reveal that 19 of the 202 vendors in Table 3 and Table 4 are non-negative bounded, i.e., they have not been impacted due to vulnerabilities corresponding to them.

5. The true mean for most of the vendors on the three days is bounded from below by negative value. Although the confidence intervals do not say anything about the percentage of the population that would fall in the negative side of the interval, the lower bound indicate a likelihood that the population would have samples with the negative effect on the vendor's stock. Thus, given the various vulnerabilities on a specific vendor, it is likely that some of those vulnerabilities would have a negative effect on the vendor's stock value, even though the overall effect (measured by the mean) would be nullified. This, as well, is well captured in our analysis.

6. Vendors with a large number of vulnerabilities have varying effects due to every vulnerability. This divergence of impact may push the true mean out of the confidence intervals, e.g., Microsoft, Cisco, etc.

7. Discussion and Comparison

Prior art have made concentrated effort on exploring and comprehending the influence of data breaches on a victim's stock returns. Additionally, data breaches can be a result of abuse of known vulnerabilities in the victim's software due to the use of vulnerable software. Intuitively, while data breaches will have an impact on the victim, vulnerabilities, however, will not have a direct impact on the victim. In this work, we focus our effort on understanding their effect on the victim, and our results prove otherwise which is also strengthened by the statistical significance of our results. Given the varying severity of the vulnerabilities, we study the role of severity on the impact on vendor's stock returns. Moreover, we compare our results with prior work. We discuss the results further in the rest of the section.

7.1. Comparison with Prior Works

Studies in the past have made varied conclusions in light of vulnerabilities. In particular the studies reflect the associations with certain aspects of those vulnerabilities, including correlation with type of vulnerability, effect of publicity, etc.. In the following, we compare our findings with the prior work across multiple aspects, e.g., vulnerability type, publicity, data source, methodology, and sector.

Confidentiality vs. non-confidentiality. Campbell et al. [15] observed a negative market reaction for information security breaches involving unauthorized access to confidential data, and reported no significant reaction to non-confidentiality related breaches. Through our analysis, we reached into similar conclusion. Specifically, we found that vulnerabilities that have negative affect on vendor's stock have descriptions containing phrases indicating confidentiality breaches, such as "denial of service", "allows remote attacker to read/execute", "allows context-dependent attackers to conduct XML External Entity XXE attacks via a crafted PDF", and "allows context-dependent attackers to have unspecified impact via an invalid character". This observation is inline with prior work.

Effect of publicity. There have been several works in the literature that attempt to understand how the coverage by media and other forms of publicity for viruses and data breaches affect the returns of a given vendor associated with such vulnerabilities. For example, Hovav and D'Arcy [10] demonstrated that virus-related announcements do not impact the stock returns of vendors. Our results partly contradict their claims, as we show that vulnerabilities impact a vendor's stock value, sometimes significantly (negatively), regardless of whether such vulnerabilities are announced or not. That is, we do not make any claim as we did not specifically study virus-related announcements.

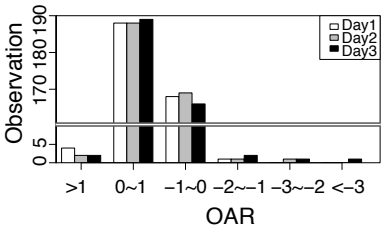

Figure 6. Effect histogram: Adobe

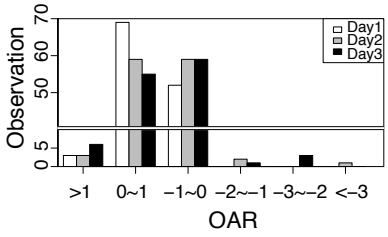

Figure 7. Effect histogram: Vmware

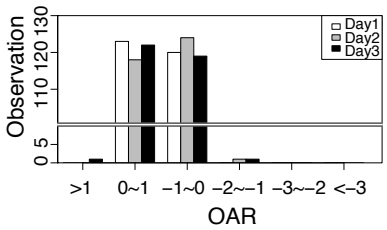

Figure 8. Effect histogram: Symantec

Data source and effect (broadening scopes). Goel et al. [14] and Telang and Wattal [13] estimated the impact of vulnerabilities on the stock value of a given vendor by calculating a Cumulative Abnormal Rate (CAR) and using a linear regression model. Their results are based on security incidents: while both gather data from the press, Telang and Wattal [13] also used a few incidents from Computer Emergency Response Team (CERT) reports. On the other hand, we consider a wide range of vulnerabilities regardless of being reported by the press. Our results show various trends and indicate the dynamic and wide spectrum of the effect of vulnerabilities on the vendors' returns. Additionally, we discuss the caveats of the methodology used by Telang and Wattal [13] below.

Methodology (Addressing caveats of prior work). The prior work have utilized CAR to measure the impact of vulnerabilities [13], which aggregates AR's on different days. However, we design a methodology that captures the impact of vulnerabilities with more precision. In particular, our method performs better due to multiple reasons. First, CAR does not effectively capture the impact of a vulnerability, due to information loss by aggregation: 1) CAR would indicate no-effect if the magnitude (upward) of one or more days analyzed negate the magnitude (downward) of other days. 2) We consider a vulnerability as having had an impact if the stock shows a downward trend on d_1, d_2, or d_3, irrespective of the magnitude. 3) Our results, through a rigorous analysis, are statistically significant.

Second, we demonstrate the caveats of CAR and show the advantages of our approach in capturing a better state of the effect of vulnerabilities on the return, we consider both Samsung and Equifax in Table 3. On one hand, the impact of vulnerability on Equifax on days 2 and 3 was significant (-14.02 and -24.09 vs. +1.52 on day 1), where CAR would capture the effect. On the other hand, such an effect would not be captured by CAR with Samsung (-0.08 and -0.08 on days 1 and 2 vs. +2.95 on day 3). Our approach considers the effect of vulnerabilities on return over different days individually, and thereby preserving the information, rather than losing it due to aggregation.

We also compare the performance of our predictor by contrasting it with the linear regression-based models in the literature. Although Fig. 4 and Fig. 3 visually suggest a similar performance in predicting the stock values, the difference in the value of MSE for these two models, 6.42 for ARIMA and 0.59 for NARX, quantitatively shows the improvement of the proposed method over the existing methods in the literature.

Sector-based analysis. Although it is intuitive that the cost of vulnerabilities on vendors is sector-dependent, a major shortcoming of the literature is that it fails to demonstrate it through analysis. By clustering vendors based on the industry they belong to, our results show the likelihood of effect to be high in software and consumer products' industry, and to be less in the device, networking or conglomerate industries as shown in Table 2. While Table 3 shows that a vulnerability may or may not have an effect on its vendor's stocks, Table 5 shows that individual vulnerabilities may affect the returns.

Shortcomings. In this study we find a significant effect of vulnerabilities on a given day and limited ourselves to the second day after the release of the vulnerability in order to minimize the impact of other factors. However, other factors may affect the stock value than the vulnerability, making the results unreliable, and highlight the correlational-nature of our study (as opposed to causational). Eliminating the effect of those factors, once known, is an open question.

For impact estimation, this study utilizes two datasets, the NVD and the stock data obtained from Alpha Vantage. As such, this study is limited to the vendors that are publicly traded. Moreover, among the publicly traded vendors, we are also limited by the vendors the data of which are captured by Alpha Vantage. For example, we notice that ATI/AMD is a publicly traded vendor, but is not captured by the service during the study period. However, we acknowledge that given our tool, this shortcoming is not difficult to address, although requires ingestion of those additional vendors for analysis.

Moreover, apart from the effect on stock, a vendor may sustain other hidden and long-term losses, such as consumers churn (switching to other products or vendors), loss of reputation, and internal losses (such as man-hour for developing remedies), which we do not consider in our evaluation, and open various directions

for future work. Furthermore, much of the effort depends upon the automated gathering of historical stock data, in this study Alpha Vantage is used as a source. Lack of a source encompassing stock exchanges worldwide further limits the study.

7.2. Vulnerabilities and Disclosure

Our analysis of the vulnerabilities shows that while vulnerabilities may or may not have an impact on the stocks, a vulnerability reported by the press is highly likely to impact the stock return. The diverse results for the vulnerabilities collected from NVD are explained by the severity of the vulnerabilities, where 1) the press may report on highly critical vulnerabilities that are more likely to result in loss, or 2) the reported vulnerabilities in the press may create a negative perception of the vendor leading to loss in their stock value. This, as a result, led many vendors to not disclose vulnerabilities in order to cope with bad publicity. For example, Microsoft did not disclose an attack on its bug tracking system in 2013 [48], demonstrating such behavior in vendors when dealing with vulnerabilities [49]. Recent reports also indicate a similar behavior by Yahoo when their online accounts were compromised, and by Uber when their employees' and users' personal information were leaked. More broadly, a recent survey of 343 security professionals worldwide indicates that the management of 20% of the respondents considered cyber-security issues a low priority, alluding to the possibility of not disclosing vulnerabilities even when they affect their systems [50].

8. Conclusion and Future Work

We perform an empirical analysis on vulnerabilities from NVD and look at their effect on the vendor's return. 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 the 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 return with high accuracy. Our work is limited in the sense that we do not consider other external factors affecting the stock or internal factors affecting long term user behavior and deriving vulnerabilities cost. Exploring those factors along with regional differences and cascade effect of vulnerabilities in effect will be our future work.

References

[1] Mohaisen, A., Alrawi, O. and Mohaisen, M. (2015) AMAL: high-fidelity, behavior-based automated malware analysis and classification. *Comput. Secur.* **52**: 251–266.

[2] Mohaisen, A. and Alrawi, O. (2014) AV-meter: An evaluation of antivirus scans and labels. In Dietrich, S. [ed.] *Proceedings of 11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA*, **8550**: 112–131.

[3] Kwon, H., Mohaisen, A., Woo, J., Kim, Y., Lee, E. and Kim, H.K. (2017) Crime scene reconstruction: Online gold farming network analysis. *IEEE Trans. Information Forensics and Security* **12**(3): 544–556.

[4] Tassey, G. (2002) The economic impacts of inadequate infrastructure for software testing. *National Institute of Standards and Technology, RTI Project* **7007**(011).

[5] Strasburg, J. and Bunge, J. (2012) Loss swamps trading firm, knight capital searches for partner as tab for computer glitch hits $440 million. *Wall Street Journal (Online). Retrieved from http://search. proquest. com/docview/1033163975* .

[6] Berr, J. (2017), 'WannaCry' ransomware attack losses could reach $4 billion. URL http://cbsn.ws/2yYjif2.

[7] The cost impact of major virus attacks since 1995. URL https://tinyurl.com/crdekj.

[8] Geppert, L. (2004) Lost radio contact leaves pilots on their own. *IEEE spectrum* **41**(11): 16–17.

[9] Jarrell, G. and Peltzman, S. (1985) The impact of product recalls on the wealth of sellers. *Journal of Political Economy* **93**(3): 512–536.

[10] Hovav, A. and D'arcy, J. (2005) Capital market reaction to defective it products: The case of computer viruses. *Computers & Security* **24**(5): 409–424.

[11] Romanosky, S., Hoffman, D. and Acquisti, A. (2014) Empirical analysis of data breach litigation. *Journal of Empirical Legal Studies* **11**(1): 74–104.

[12] Spanos, G. and Angelis, L. (2016) The impact of information security events to the stock market: A systematic literature review. *Computers & Security* **58**: 216–229.

[13] Telang, R. and Wattal, S. (2007) An empirical analysis of the impact of software vulnerability announcements on firm stock price. *IEEE Transactions on Software Engineering* **33**(8): 544–557.

[14] Goel, S. and Shawky, H.A. (2009) Estimating the market impact of security breach announcements on firm values. *Information & Management* **46**(7): 404–410.

[15] Campbell, K., Gordon, L.A., Loeb, M.P. and Zhou, L. (2003) The economic cost of publicly announced information security breaches: empirical evidence from the stock market. *Journal of Computer Security* **11**(3): 431–448.

[16] Cavusoglu, H., Mishra, B. and Raghunathan, S. (2004) The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce* **9**(1): 70–104.

[17] Anwar, A., Khormali, A., Nyang, D. and Mohaisen, A. (2018) Understanding the hidden cost of software vulnerabilities: Measurements and predictions. In *Proceedings of the 14th EAI International Conference on Security and Privacy in Communication Networks, SecureComm 2018* (Singapore, Singapore).

[18] Gamero-Garrido, A., Savage, S., Levchenko, K. and Snoeren, A.C. (2017) Quantifying the pressure of

legal risks on third-party vulnerability research. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)* (Dallas, TX): 1501–1513.

[19] Li, F. and Paxson, V. (2017) A large-scale empirical study of security patches. In *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)* (Dallas, TX): 2201–2215.

[20] Nguyen, V.H. and Massacci, F. (2013) The (un) reliability of nvd vulnerable versions data: An empirical experiment on google chrome vulnerabilities. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS)* (Sydney, Australia): 493–498.

[21] Christey, S. and Martin, B. (2013) Buying into the bias: Why vulnerability statistics suck. *BlackHat, Las Vegas, USA, Technical Report* **1**.

[22] Romanosky, S., Telang, R. and Acquisti, A. (2011) Do data breach disclosure laws reduce identity theft? *Journal of Policy Analysis and Management* **30**(2): 256–286.

[23] Gordon, L.A., Loeb, M.P. and Zhou, L. (2011) The impact of information security breaches: Has there been a downward shift in costs? *Journal of Computer Security* **19**(1): 33–56.

[24] Steinke, M., Metzger, S. and Hommel, W. (2016) POSTER: VUDEC: A framework for vulnerability management in decentralized communication networks. In *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS)* (Vienna, Austria): 1829–1831.

[25] Stock, B., Pellegrino, G., Rossow, C., Johns, M. and Backes, M. (2016) POSTER: mapping the landscape of large-scale vulnerability notifications. In *Proceedings of the 23rd ACM Conference on Computer and Communications Security (CCS)* (Vienna, Austria): 1787–1789.

[26] Zhao, M., Grossklags, J. and Liu, P. (2015) An empirical study of web vulnerability discovery ecosystems. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security (CCS)* (Denver, Colorado): 1105–1117.

[27] Trinh, M., Chu, D. and Jaffar, J. (2014) S3: A symbolic string solver for vulnerability detection in web applications. In *Proceedings of the 21st ACM Conference on Computer and Communications Security (CCS)* (Scottsdale, Arizona): 1232–1243.

[28] Saha, D. (2008) Extending logical attack graphs for efficient vulnerability analysis. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)* (Alexandria, VA): 63–74.

[29] Zhang, S., Caragea, D. and Ou, X. (2011) An empirical study on using the national vulnerability database to predict software vulnerabilities. In *Proceedings of the 22nd International Conference on Database and Expert Systems Applications DEXA, Part I*: 217–231.

[30] Sabottke, C., Suciu, O. and Dumitras, T. (2015) Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits. In *Proceedings of the 24th USENIX Security Symposium (Security)* (Washington, DC): 1041–1056.

[31] Homaei, H. and Shahriari, H.R. (2017) Seven years of software vulnerabilities: The ebb and flow. *IEEE Security & Privacy* **15**(1): 58–65.

[32] Attila, H., Erdősi, P.M. and Kiss, F. (2016) The common vulnerability scoring system (cvss) generations–usefulness and deficiencies .

[33] Holm, H. and Afridi, K.K. (2015) An expert-based investigation of the common vulnerability scoring system. *Computers & Security* **53**: 18–30.

[34] Allodi, L., Banescu, S., Femmer, H. and Beckers, K. (2018) Identifying relevant information cues for vulnerability assessment using CVSS. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy CODASPY*: 119–126.

[35] Johnson, P., Lagerstrom, R., Ekstedt, M. and Franke, U. (2016) Can the common vulnerability scoring system be trusted? a bayesian analysis. *IEEE Transactions on Dependable and Secure Computing* .

[36] Kar, A. (1990) Stock prediction using artificial neural networks. *Department of Computer Science and Engineering, IIT Kanpur* .

[37] Farhang, S., Laszka, A. and Grossklags, J. (2017) An economic study of the effect of android platform fragmentation on security updates. *arXiv preprint arXiv:1712.08222* .

[38] National vulnerability database (NVD), url=https://nvd.nist.gov/.

[39] Vantage, A., Alpha vantage. URL https://www.alphavantage.co/.

[40] CVE - common vulnerabilities and exposures (CVE). URL https://cve.mitre.org/.

[41] Common weakness enumeration. URL https://cwe.mitre.org/.

[42] Common vulnerability scoring system SIG. URL https://www.first.org/cvss/.

[43] CVSS version 3. URL https://www.first.org/cvss/cvss-v30-user_guide_v1.1.pdf.

[44] Elman, J.L. (1990) Finding structure in time. *Cognitive science* **14**(2): 179–211.

[45] Horne, B.G. and Giles, C.L. (1994) An experimental comparison of recurrent neural networks. In *Proceedings of the Advances in Neural Information Processing Systems 7, [NIPS Conference]*: 697–704.

[46] Moré, J.J. (1978) The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, 105–116.

[47] Box, G.E. and Pierce, D.A. (1970) Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association* **65**(332): 1509–1526.

[48] Menn, J. (2017), Exclusive: Microsoft responded quietly after detecting secret database hack in 2013. URL http://reut.rs/2ysNpw2.

[49] A social science approach to information security. URL http://bit.ly/2l7IefL.

[50] Violino, B. (2017), Data breaches rising because of lack of cybersecurity acumen. URL https://tinyurl.com/y8fcvafc.