Impact of Software Vulnerability Announcements on the Market Value of Software Vendors – an Empirical Investigation[1]

Rahul Telang, Sunil Wattal
{rtelang, swattal}@andrew.cmu.edu

Abstract

Researchers in the area of information security have mainly been concerned with tools, techniques and policies that firms can use to protect themselves against security breaches. However, information security is as much about security software as it is about secure software. Software is not secure when it has defects or flaws which can be exploited by hackers to cause attacks such as unauthorized intrusion or denial of service attacks. Any public announcement about a software defect is termed as 'vulnerability disclosure'.

In this paper, we use the event study methodology to examine the role that financial markets play in determining the impact of vulnerability disclosures on software vendors. We collect data from leading national newspapers and industry sources by searching for reports on published software vulnerabilities. Our main result is that vulnerability disclosures do lead to a negative and significant change in market value for a software vendor. On average, a vendor loses around 0.6% value in stock price when a vulnerability is reported. This is equivalent to a loss in market capitalization values of $0.86 billion per vulnerability announcement. To provide further insight, we use the information content of the disclosure announcement to classify vulnerabilities into various types. This is the first study to measure vendors' incentive to develop secure software and also provides many interesting implications for software vendors as well as policy makers.

Keyword:
information security, software vulnerability, quality, event study, disclosure policy

1. Introduction

Many believe that software vendors typically follow the policy of 'sell today and fix it tomorrow'; or 'I'd rather have it wrong than have it late' (Paulk et al 1994; Arora, Caulkins, Telang 2004) for launching software products in the market. This policy, dictated by the need to launch products quickly before competitors, seemed to work in the past because software errors which escape detection during pre-launch testing appear very infrequently in normal operations (once every '5000' years, as per Adams, 1980). However, Adams' '5000-year error' theory might not hold in the internet age because hundreds (if not thousands) of people are looking for flaws in other vendors software products, drastically increasing the chances that a flaw will be exposed. Various people such as hackers, independent security

[1] The authors would like to thank Ashish Arora, Ramayya Krishnan, Sandra Slaughter, participants at the Security Working Group Meeting at Heinz School of Public Policy, and Workshop on Information Systems and Economics (WISE 2004) for helpful comments and suggestions.

firms and academic researchers are interested in finding flaws in other vendors' software for different reasons. Not only are security software products such as firewalls at risk, but software like operating systems, enterprise software and database software also contain flaws which can be exploited to create security related attacks.

The Organization of Internet Safety (OIS) (www.oisafety.org) defines security vulnerability as[2]: "security vulnerability is a flaw within a software system that can cause it to work contrary to its documented design and could be exploited to cause the system to violate its documented security policy". Software vulnerabilities have widespread impact and can potentially cause billions of dollars in downtime and disruptions to firms (In this paper, the word 'firms' refers to companies which use software products; 'vendors' refers to companies which develop the software products). A study by NIST in 2002 estimates the cost of faulty software at $60 bn per year. Incidents like the Code Red virus (in 2001) and the Melissa virus (in 1999) occurred when hackers exploited flaws in software. The damage due to Code Red was estimated at $2.1 bn and due to Melissa at $1.1bn[3]. The Gartner Group estimates that the system downtime caused by security vulnerabilities would triple from 5% of the total downtime in 2004 to 15% of the total downtime in 2008[4]. Microsoft's $200m campaign for .NET was marred by the discovery of a security flaw in Visual C++ .NET barely a month after Microsoft Chairman Bill Gates directed employees to focus on building more secure software (dubbed the 'Trustworthy Computing' initiative). Moreover, vulnerability disclosure is finding its way into firms' strategy toolkits as it evident from a WSJ report in Feb 2004 that software vendors are spending time and effort in discovering flaws in their rivals' software products in order to influence the rivals' stock prices. For example, security software vendor IDS released a vulnerability alert on rival Checkpoint's firewall software on the day Checkpoint usually holds its annual US investor conference. Some examples of vulnerability announcements reported in popular press are:

- News.com(04/25/2000) *"A computer security firm has discovered a serious vulnerability in Red Hat's newest version of Linux that could let attackers destroy or deface a Web site - or possibly even take over the machine itself……."*
- WSJ(02/11/2004) *"Microsoft Corp. warned customers about serious security problems with its Windows software that let hackers quietly break into their computers to steal files, delete data or eavesdrop on sensitive information……."*

Inspite of all these concerns about software vulnerabilities, not much has been mentioned in literature about the incentive of software vendors to invest in defect-free software. Literature on software risks fails to include any measure for security related risks (Wallace, Keil and Rai, 2004; Barki, Rivard and Talbot, 1993). In traditional literature, software quality is measured in terms of reliability and integrity of the source code (Harter, Krishnan and Slaughter, 2000)–which essentially tests software against specified streams of input from users. However, in today's internet age, software designers must not only think of users, but also malicious adversaries. Some quality models such as ISO9126 fail to include computer security (Pfleegar 1997). Therefore software which has been certified as high quality, based on existing definitions of software quality, can have many security flaws. Researchers in computer science are working on better integration of the two disciplines of software quality and software security while designing software (Wang and Wang, 2003; McGraw 2004). So far, there has only been anecdotal evidence that software vulnerabilities are causing vendors to lose market value. For example, the Wall Street Journal (11/09/2004) reported that Microsoft's Internet Explorer (IE) is losing market share in the web browser market to competitors like Mozilla's Firefox, due to numerous flaws discovered in IE.

Prior literature(Jarrell and Peltzman 1985, Davidson and Worrell 1992) predicts that product defect announcements are associated with loss in market value of a firm. However the results of these prior studies cannot be directly applied to product defects in the software industry because of the following characteristics of software products: One, software products generally come with a click-wrap agreement (***EULA***–End User License Agreement) which limits the vendors' liability. Two, the general philosophy held by software vendors, software customers and the US courts is that software is a uniquely complex product that will probably have some defects (Cusumano 2004). Three, popular software like those from Microsoft are constantly subject to malicious and non-malicious attacks and have a greater number of flaws reported in them as compared to software by Apple, where the user base is comparatively smaller. Therefore, the presence of vulnerabilities may not always signal a lower quality product; it may infact signal superiority over

---

[2] In this paper, we use the terms 'software vulnerability', 'security vulnerability', 'bug', 'flaw' interchangeably. Any other type of vulnerability such as a non-security related vulnerability explicit mentioned by name.

[3] Source : www.cisco.com/warp/public/cc/so/neso/sqso/roi1_wp.pdf

[4] http://www.tekrati.com/T2/Analyst_Research/ResearchAnnouncementsDetails.asp?Newsid=3608

competition. E.g. John Thomson, CEO of Symantec, predicts that the flaws in Linux will increase as the installed base increases. Motivated by these observations, in this research we try to quantify the losses that software vendors bear when a vulnerability is disclosed in their product. The research questions that we seek to answer are:

1. What is the impact of vulnerability disclosures on the market value of a software vendor?
2. How do the characteristics of the vulnerability impact this change in market value?

Most prior research on information security discusses the economics of such investments from a customer perspective, rather than from a software vendor perspective (E.g. Anderson 2001). Gordon and Loeb (2002) show that firms should make investments in information security far less than the expected loss from a security breach. Gordon et al (2002), Gal-Or and Ghose (2003) discuss the economics of information sharing among firms on security related issues. Prior event study analyses on information security have focused on the change in market value of firms whose systems are breached (Cavusoglu et al (2004). This study shows that announcements of a security breach negatively impact the CAR (Cumulative Abnormal Return) of firms whose information systems have been breached. Campbell et al (2003) conduct a similar event study and find that only the impact of confidentiality related security breaches is negative and significant; the impact of non-confidentiality related security breaches is not significantly different from zero.

Disclosure of vulnerabilities has been one contentious area and some recent academic work is examining this issue more formally. Arora, Telang and Xu(2004) study the optimal timing of vulnerability disclosure and show how disclosure can force vendors to release patches quickly. Arora, Caulkins and Telang(2003) find that larger software vendors find it optimal to rush product into market and then invest in post launch patching. Arora, Telang and Xu (2004) study the optimal timing of vulnerability disclosure and show how disclosure can force vendors to release patches quickly. Kannan and Telang (2004) explore the welfare implications of a market mechanism for software vulnerabilities and report that a market based mechanism for software vulnerabilities always underperforms a CERT-type mechanism.

However none of these studies measures the impact of disclosure on vendor's market value or profitability. While one major goal of disclosure is to eventually force vendors to develop secure software, empirically, there is no evidence that suggests that disclosure indeed creates such incentives. Our paper provides an understanding of whether such disclosures create incentives for the vendors to produce secure software in the first place.

Our methodology follows closely from prior event study analysis(Campbell, Andrew and MacKinley 1997). We collect data on about 146 vulnerability disclosure announcement over the period of over 5 years for 18 publicly traded vendors. Our results confirm that vulnerability disclosure adversely and significantly affects the stock performance of a software vendor. We show that, on average, a software vendor loses around 0.63% of market value on the day of the vulnerability announcement. This translates to a dollar amount of $0.86 billion loss in market value. We also show that markets do not penalize a vendor any more if the vulnerability is discovered by a third party than by the vendor itself.

The main contribution of this research is that this is one of the first studies, to our knowledge, that has tried to measure the impact of vulnerabilities on software vendors. Thus, we extend prior literature on product defects and confirm that software vendors too suffer a loss in market value when a flaw is discovered in their product. This is inspite of the fact that software vulnerabilities are prevalent among software of almost all major vendors and that vendors face no legal liability if clients suffer losses due to the software flaw. This has important implications in terms of vendors making investment in software quality as well as policy and legal issues which govern vulnerability disclosures.

This rest of the paper is organized as follows: We develop our hypotheses in Section 2. In Section 3, we discuss the methodology of data collection and also describe the event study methodology. In Section 4, we present an outline of our multivariate regression analysis and highlight the ongoing research. Finally, we present the concluding remarks in Section 5.

2. Hypotheses

Much of the prior literature on product defects (Jarrell and Peltzman (1985), Davidson and Worrell (1992)) shows that generally defective product and recall announcements are associated with loss in market value of a firm. Banker and Slaughter (1998) find that unplanned and critical maintenance activities increase software maintenance costs. Thus, fixing bugs entails cost to firms. For example, the security fixes may cost about $2000-$9000 when done during testing phase. However, they may cost more than 4-8 times when fixed after the application has been shipped[5]. Slaughter, Harter and Krishnan (1998) and Westland (2003) suggest that software defects are harder and costlier to fix if discovered later in the software development cycle (e.g. when the product has been shipped to the customer). Moreover, security breach announcements by the user organizations have been known to have a negative impact on the

[5] http://www.sbq.com/sbq/rosi/sbq_rosi_software_engineering.pdf

share value for firms. Cavusoglu et al (2004) show that the market capitalization values of firms decreases, on average, by $2.1 billion within two days of a security breach. Cyber insurance firm J.S. Wurzler charges an additional premium to firms for using Windows NT due to the number of security breaches in the software (Gordon, Loeb and Sohail, 2003). Clearly, poor security costs the users of the software. Thus, the cost to the vendor can be written as

Cost of vulnerability disclosure to vendor = Cost of patching the vulnerability + λ * (Cost to the users of the software due to exploitation, and/or cost to patch the system)

Here, λ is the internalization factor. That is the user loss that is internalized by the vendor due to lost sale or reputation loss (or liability if imposed in future). Clearly, λ depends on how willing users are to "punish" the vendor, how competitive the market is and the characteristics of vulnerability (See Arora, Telang and Xu 2004). For example, this is a typical customer reaction.

"We are extremely concerned by the high amount of vulnerabilities and patches from Microsoft. This goes against the credibility of what they have been saying,"
Michael Kamens, global security director at Thermo Electron Corp[6].

Given the fact the competitors seem to be using vulnerability disclosure as a strategic way to undermine the rivals' reputations, it seems that vulnerability disclosure signals a potential loss in future cash flows for a software vendor owing to customer dissatisfaction because customers suffer a loss if their systems get breached. It also signals an increase in product related costs due to the time and effort that the vendor spends in developing a patch or a fix for the flaw. Therefore, we hypothesize that

H1: *A software vendor suffers a loss in market value when a security related vulnerability is announced in its products.*

Our second hypothesis pertains to whether the software vendor releases a patch for the product at the time of the vulnerability announcement. As per the popular convention followed in the vulnerabilities market, vendors are given some time to work on a patch for the vulnerability before it is made public. Vendors may also provide a workaround (such as disabling features of the software) when a vulnerability is disclosed and choose to address the vulnerability in a future upgrade. Presence of the patch is also likely to reduce customers' loss if they apply the patch. Since presence of patch also reflects vendor's commitment to its customers we expect that vulnerabilities disclosed with a patch will compensate, to an extent, the negative signal due to vulnerability disclosure. Vendors are also pushing for limited disclosures so that they can release the remedial patch in time. This also suggests that patches play a critical role. Therefore our second hypothesis is:

H2: *CAR (Cumulative Abnormal Return) [negative] of a stock is greater for vulnerabilities where the software vendor does not release a patch at the time of the vulnerability disclosure.*

Campbell et al(2003) further show that the loss in market value for firms is more for confidentiality related breaches than for non-confidentiality related breaches. Confidentiality related breaches involve attacks where an intruder can gain access into a system and can steal sensitive information. Non-confidentiality related breaches include attacks where the most likely scenario is a disruption and/or a downtime. Therefore, we would expect that the vendor loses more market value if the vulnerability in its software causes a confidentiality related breach. Hence next hypothesis is:

H3: *CAR is greater for a vulnerability which can potentially cause a breach in confidentiality as compared to non-confidentiality related breaches.*

The impact of a software flaw on a vendor also depends on how severe the vulnerability is. Davidson and Worrell (1992) conduct an event study with product defect announcements in the tire industry and showed that the impact of severe flaws (which involve a recall) is more than that of less severe flaws (which involve repairs but nor recall). Therefore, we propose our next set of hypotheses as:

H4: *The loss in market value of a software vendor is greater if the announced vulnerability has a higher severity.*

A recent article in the Wall Street Journal hinted that firms are using vulnerability disclosure as a strategic weapon against competitors. E.g. ISS disclosed a vulnerability in rival Checkpoint's flagship firewall product just ahead of Checkpoint's investor summit. Vendors themselves disclose vulnerability information in their products routinely. In fact, many believe that vendors would prefer not to disclose information at all but they fear that someone else would do it. Generally vendors are likely to be more careful about the disclosure as opposed to third party. Moreover, disclosure by vendors would signal their commitment to providing secure software, we hypothesize that

H5: The loss in market value for a software vendor is lower in case the security vulnerability is discovered by the vendor itself rather than by rivals or third party security firms.

We would also expect that vulnerabilities reported would have a more widespread (negative) impact on the market value of a firm than vulnerabilities reported in industry sources such as CERT. Therefore, we hypothesize that

H6: *The magnitude of CAR is more when the vulnerability is reported in popular press than in industry sources.*

---

[6] http://www.computerworld.com/softwaretopics/os/windows/story/0,10801,92349,00.html

3. Data Description & Methodology

3.1 Vulnerability Disclosure Process

The typical process of vulnerability disclosure takes place as shown in Figure 1.

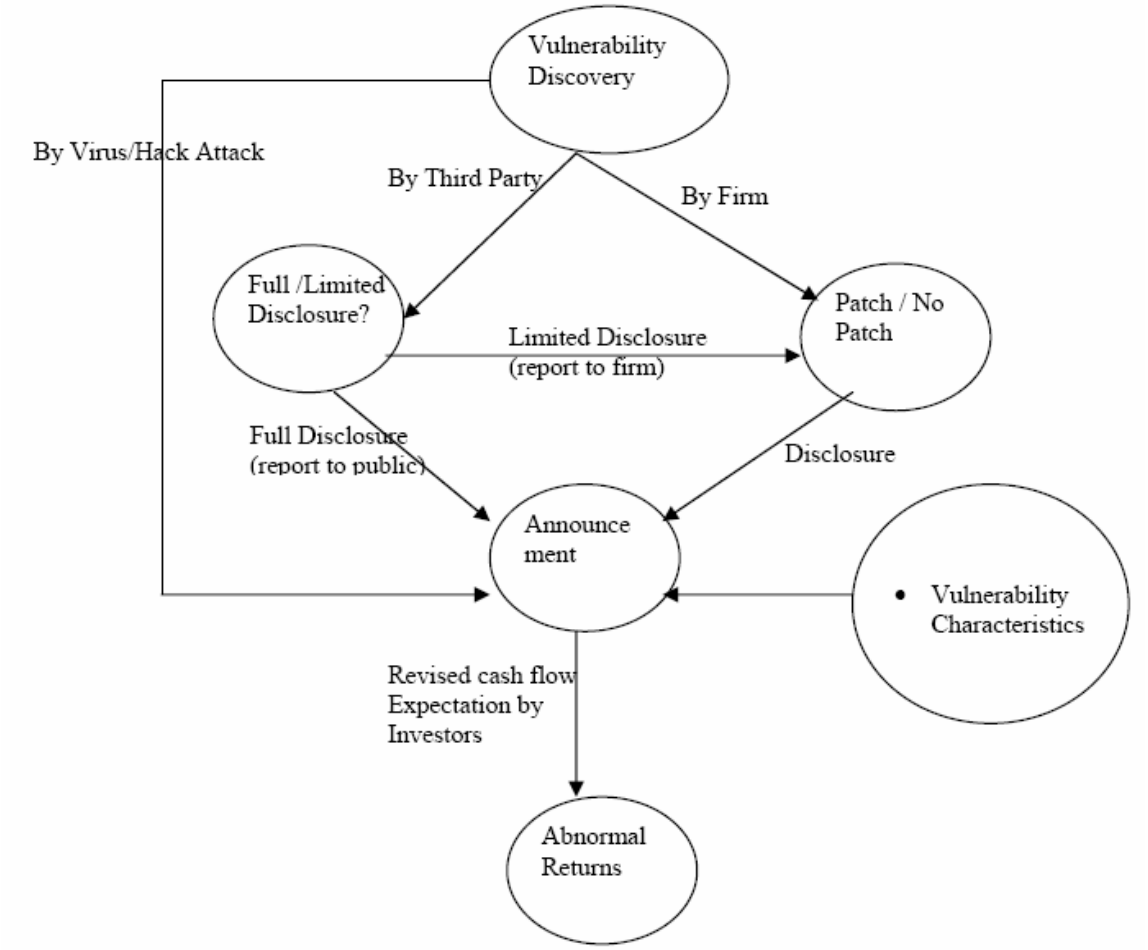

Figure 1: Vulnerability Discovery and Disclosure

The process starts with the vulnerability discovery. There are various sources for vulnerability discovery. Some vulnerabilities are discovered by amateur researchers or by independent security analysts (ISA) like eEye Security.

If the ISA chooses to follow 'limited disclosure', it reports the vulnerability to the concerned vendor or to an independent body like CERT who disclose the vulnerability to public later (generally after the fix has been made available) in a limited way[7]. Sometimes, the vendor itself might discover a vulnerability in its products. On the other hand, if the ISA chooses to follow 'full disclosure' or if a hacker discovers the vulnerability, the vulnerability may get announced directly to public. Forums such as Bugtraq allow for full disclosure of vulnerability information. In any case, after the vulnerability has been announced, investors might re-evaluate their projections on the software vendors' profitability based on the vulnerability characteristics. Consequentially, the stock price of the vendor might show an abnormal return due to the vulnerability announcement.

[7] CERT then reports the vulnerability to the respective vendor and gives it time (typically 45 days) to come out with a patch.

3.2 Data

Our data comes from the vulnerability disclosures in popular press as well as from the advisory reports in CERT. We include articles published by news networks like Businesswire, Newswire and daily articles in popular press like WSJ, NY Times, Washington Post and LA Times. We used the following terms in our search: 'vulnerability AND disclosure', 'software AND vulnerability', 'software AND flaw', 'virus AND vulnerability'& 'vulnerability AND patch'. In accordance with prior event study literature (Hendricks & Singhal 1996), we exclude certain types of announcements from our sample.

Our dataset contains 146 vulnerability announcements pertaining to around 18 firms in the time period January 1999–May 2004. The descriptive statistics are as follows:

Number of firms	18
Number of announcements	146
%age of vulnerabilities announcements in popular press	35
%age of vulnerabilities for which vendor has patch available at the time of the announcement.	24
%age of vulnerabilities discovered by the vendor itself	36
% of vulnerabilities that could potentially result in a security breach related to confidentiality	39

Table 1 : Descriptive Statistics

3.3 Methodology

We use the standard event study methodology for this analysis. An event study assumes that returns on a stock are significantly impacted by an event of interest (in our case, the event of interest is the vulnerability disclosure announcement). The period of interest for which we observe the event is known as the event window. In practice, researchers use different values of the event window. In our study, we define a 1-day event window (day of the announcement or **day** '0'), since we can exactly determine the date of the event[8]. A shorter event window permits a better estimation of the effects of information on stock prices since it reduces the possibility of other confounding factors not related to the announcement. It also increases the power of the statistical tests. We use two main methods followed in the event study methodology:

1. The Market Model

In the market model, the abnormal returns are estimated as follows:

$$AR_{it} = R_{it} - \alpha - \beta_{it} \cdot R_{mt}$$ [9] (1)

where *i* denotes the event(*i*=1,2……N) and *t* denotes the day of the event. *ARit* denotes the abnormal return of event *i* at time *t*, *Rit* denotes the actual return and *Rmt* denotes the market return at time period t[10]. $\alpha + \beta_{it} \cdot R_{mt}$ denotes the normal return of the firm due to the market–wide movement. The abnormal return is defined as the difference between the actual return and the normal return. This is the part of the actual return that cannot be explained by market movements and captures the effect of the event. Since most of the tech stocks are listed on NASDAQ, we use this as our indicator for market returns. We use ordinary least squares regression to estimate the coefficients α and β for the above model over the estimation window. The estimation window, of size between 120-200 days used in most studies, is the period immediately before the event window. In our case, we use an estimation window of size 160 days, from day -175 to day -16.

---

[8] We also highlight our results using different values of the event window.

[9] R_{it} for a stock is the percent change in the stock price at time t, $(=P_{it} - P_{it-1}) / P_{it-1}$

[10] We obtain the data on the stock and market returns from Yahoo Finance(http://finance.yahoo.com)

2. The Market Adjusted Model

In this case, the abnormal returns are given as

$$AR_{it} = R_{it} - R_{mt} \quad (2)$$

where the terms have the usual meaning as in the Market Model.

The mean abnormal return across all observations on day *t* of the event is given as $\overline{A}_t = \sum_{i=1}^{N} AR_{it}$. The cumulative abnormal return $= \sum_{event} \overline{A}_t$ *CAR* for the event is defined as the sum of the abnormal returns over the event window. Since vulnerabilities were disclosed by more than one vendor on a given day, our test statistic should allow for event day clustering. The following t-statistic proposed by Brown & Warner(1985) takes into account event day clustering as well as cross-sectional dependence in the security specific excess returns.

$$t = \frac{\overline{A}_t}{\sqrt{S^2_{\overline{A}}}} \quad (4)$$

Where $S^2_{\overline{A}} = \frac{1}{T-1}\left(\sum_{s=1}^{T}(\overline{A}_s - \overline{\overline{A}})\right)$ and $\overline{\overline{A}} = \frac{1}{T}\left(\sum_{s=1}^{T}\overline{A}_s\right)$

The null hypothesis is that the abnormal returns are not significantly different from zero. Under the null hypothesis, the abnormal returns are independent and identically distributed and normal with a mean of zero and the variance given by the variance of abnormal returns over the estimation period.

Results

Table 2 summarizes the results of our event study and quantifies the effect of vulnerability disclosures on the stock prices of software vendors (p-values are in parenthesis)

Day 0 CAR	Market Model	Market Adjusted Model
Mean Abnormal Return	-0.63 *(0.01)*	-0.67 *(0.01)*
Percent Less than Zero	64% *(0.00)*	63.5% *(0.001)*

Table 2: Cumulative Abnormal Return

We calculate CARs under two different models (Market Model and Market Adjusted Model). The Mean Abnormal Return Test (equations (1) to (4)) is parametric in nature and makes assumptions about the distribution of abnormal returns. To strengthen our results, we also use the Sign Test, which is a non-parametric test. The Sign Test is based on the sign rather than the magnitude of the abnormal returns and requires that under the null hypothesis, the proportion of abnormal returns greater than (or less than) zero is 50%.

From Table 2, we note that the CAR for day 0 is negative across the two different models. E.g. the Mean Abnormal Return varies between 0.5%-0.67% depending on the model used. Further the Market Model and the Market Adjusted Model are statistically significant at $p<0.01$ Finally, the percent observations less than zero range between 57.8%-64%($p<0.01$). It is clear that CAR is negative and statistically significant for both the models and both the tests.

Thus, our results suggest that software vendors do tend to lose market value when a vulnerability is announced in their product. This provides support for hypothesis *H1* that vulnerability announcements are associated with a loss in market value of software vendors. We also calculate the abnormal returns using different event windows using the Market Model (Table 3).

Day	-1	0	0 to 1	0 to 2	0 to 5	0 to 10
CAR	0.25	-0.63	-0.65	-0.47	-0.25	-0.8
(p-value)	(0.4)	(0.01)	(0.07)	(0.35)	(0.7)	(0.36)

Table 3: CAR for various time periods

From the table, it is clear that the CAR on day 0 is negative and significant at the 0.05 level. However, CAR for day 0 and day 1 combined is significant only at the 0.07 level. This suggests that the stock market is efficient in the sense that the effect of a software vulnerability announcement is quickly incorporated into a vendors' stock price. The p-values for day -1 is not statistically significant. A possible explanation for this is that the effect of news leakage through forums like SecurityFocus is not significant. The CARs in columns 3-6 are negative but not statistically significant. However, it is interesting to note that the CARs are negative for even a 10 day window.

Our result corroborates prior work on defective products (Jarrell et al 1985, Davidson et al 1992) by showing that product defects lead to a loss in market value of a firm. Our study analyzes returns on stocks of software vendors and we find that defective software which compromises the security of customers' information systems leads to a negative impact on the market value of the software vendor.

Market Capitalization

We also calculated the abnormal change in market capitalization values of the software vendor due to the vulnerability announcement[11]. For each firm, the day 0 change in market capitalization value was calculated by multiplying the day -1 market capitalization value with the abnormal returns on day 0. On average, we calculate that the software vendors in our sample lost $0.86 billion in market capitalization value on the day of the vulnerability disclosure.

4. Effect of Vulnerability Characteristics

We first compare the mean CARs of different sub-samples based on vulnerability characteristics to understand how various factors affect market value of the firms. We look at three specific vulnerability characteristics – source of discovery, source of disclosure and whether Microsoft product(Table 4).

Vulnerability Characteristic	Variable	CAR	p-value
Source of Discovery	Discovered by Vendor(**36%**)	***-0.95***	***0.1***
	Discovered by Third Party(**64%**)	-0.47	0.2
Source of Disclosure	CERT(**34%**)	-0.47	0.4
	Press(**35%**)	***-0.98***	***0.04***
Microsoft vs. Non Microsoft	Microsoft(**46%**)	-0.28	0.4
	Non-Microsoft(**54%**)	-0.91	0.13

Table 4: Sub-Sample Method

[11] We obtain the market capitalization values from the CSRP database by multiplying the share price with the number of shares outstanding.

Surprisingly, when vendors disclose the information, investors seem to perceive it more negatively than when some third party releases it. The source of disclosure is also relevant. Investors seem to pay more attention to vulnerabilities published in mainstream newspapers than to CERT. Finally, we do not find an evidence of Microsoft effect though, non-Microsoft firms seem to lose more value.

Ongoing Research

A drawback of the sub-sample method is that this could give spurious results because the effects of a sub-sample, such as press vs. CERT, can be explained by a relationship between press and other independent variables. Therefore we also use a regression model to explain the effect of various vulnerability characteristics on abnormal returns. The regression method has the advantage over the sub-sample method that the regression method captures the effect of all the independent variables simultaneously. However, ordinary least squares model might not be appropriate because it does not account for heterogeneity among firms. The issue of heterogeneity is an important consideration in analyzing panel data. For example, the level of abnormal returns could differ across firms if investors use different valuation models across firms. Estimating aggregate parameters while ignoring heterogeneity could lead to biased and inconsistent estimates(Hsiao 1986).

To incorporate the impact of firm specific heterogeneity in our data, we propose a fixed effects model. The fixed effects model controls for unobservable firm specific variables that are constant over time. This is equivalent to generating dummy variables for each firm and including them in an ordinary linear regression to control for firm specific effects. The model can be specified as:

$$y_{it} = X_{it} \cdot \beta + \mu_i + \varepsilon_{it} \qquad (5)$$

where *i*=1…..*N*(*N* is the total number of firms) and *t*=1……..*T*(*T* is the total number of events). *yit* is the Abnormal Return(*ARit*) for firm *i* at event *t* as calculated according to the market model in equation(1). *Xit* are the independent variables which capture the various vulnerability characteristics, *ui* is the firm specific dummy variable.
So far, our analysis tests for hypothesis *H1* and *H5* and *H6*. Our results from the regression would test hypothesis *H2*, *H3* and *H4*. The regression analysis produces several interesting results, which we do not discuss here due to space constraints. However, we expect to present the analysis on this section at the conference.

5. Conclusions and Discussion

To the best of our knowledge, this is the first study to analyze the impact of product defects on software vendors. We also analyze the information content of the vulnerability disclosure announcement and classify vulnerabilities into various sub-types based on the following characteristics: the source of vulnerability disclosure, severity of the vulnerability, availability of a fix, whether an exploit was publicly available at the time of discovery, the type of security breach caused by the vulnerability and the source of vulnerability discovery. Our results show that vulnerability disclosure leads to a significant loss of market value for software vendors. This indicates that the stock markets react negatively to the news of a vulnerability disclosure, because the discovery of a vulnerability could suggest a loss in future cash flow for the software vendors. Software vulnerabilities affect the cash flows of a vendor due to two reasons: one, the vendor has to spend time and effort in providing a patch for the vulnerability and two, the customer dissatisfaction due to product defects could lead to lost sales in future. This has implications for software vendors to invest in improving the quality of their software. While vendors would like to launch software products as soon as possible, our study shows that vendors need to focus testing in areas that can potentially contain greater number of security vulnerabilities. We also show that the effect of a vulnerability announcement is quickly incorporated into the stock price and after the second day, there is no significant impact on the stock prices.

Our study also provides preliminary evidence that firms should integrate security into software quality practices. In a firm with limited resources, this would mean focusing testing efforts in areas that have a greater number of security vulnerabilities. Although researchers in computer science have stressed on this fact (Mc Graw 2004), there hardly exists any literature in software engineering economics which measures the return on investment of incorporating security based metrics in software quality or software risk assessment. While software quality traditionally deals with functional testing, complete security testing would incorporate non-functional testing we well, i.e. subjecting the software to misspecified input streams (Potter and Mc Graw 2004).

Comparison with prior event studies:

It is interesting to compare how the abnormal returns in our event study compare with results in prior event studies. Specifically, we compare our results with event studies in the following categories: security breach related announcements, IT investment related announcements and product defect related announcements. It is especially interesting to note that the loss in market value that vendors suffer due to a security vulnerability is much less than that suffered by firms during a security breach.

Classification of Event Study	Authors	Time Period	CAR
Impact of Vulnerability Disclosures on Software Vendors	Telang R and S Wattal (2004)	1999-2004	-0.63%
Impact of Security Breaches on Firms	Campbell K, Gordon LA, Loeb MP and L Zhou (2003)	1995-2000	-2.0%*
	Cavusoglu H, Mishra B and S Raghunathan (2004)	1998-2000	-2.1%
Impact of Product Recall Announcements	Jarrell G and S Peltzman (1985)	1967-1981	-0.81% (for auto)
	Davidson WL III and DL Worrell (1992)	1968-1987	-0.36% (day -1)
Impact of IT Investment Announcements	Chatterjee D, Richardson VJ and RW Zmud (2001)	1987-1998	1.16%
	Subramani M and E Walden (2001)	Oct 1998-Dec 1998	7.5%
	Dos Santos BL, Peffers K and DC Mauer (1993)	1981-1988	1%

*Not Significant at the 10% level

Table 5: Summary of previous event studies

Implications for Software Quality and Disclosure Policy

As we noted in introduction, one major argument given by the full disclosure group is that it will eventually force the vendors to improve the quality of their product. From our analysis, there seems to be some support for this argument. Disclosure adversely affects the market valuation of the vendors and hence clearly creates some incentives for vendors to produce better quality software. However, market value is only one metric to capture the impact of disclosure. A more interesting and comprehensive work would be to measure the impact of disclosure on profit or market share of these firms. But our paper does provide a starting point for why we should analyze this issue in more detail. Another potential area of future research would be to capture and test the link between security based risks and the quality of software systems.

Our study points that vendors are not necessarily better off disclosing information themselves. Generally, an argument could be made that vendors should release the information themselves, for if not, someone else will and it will lead to worse consequences. However, we do not find any evidence of this. In our sample, none of the vulnerabilities was discovered by hackers. Hackers however exploit vulnerabilities once they are made public by searching for un-patched systems. Vendors are probably better off keeping quiet and integrate their fixes as either service packs(which do not give micro-details on what it fixes) or newer versions and announce the patch only if someone else has disclosed it.

Some vulnerabilities are posted on a public listing such as Bugtraq before these are announced in popular press or CERT. In that case, the actual vulnerability announcement may have little surprise value. Therefore our results are a lower bound for the actual decrease in stock prices experienced by the software vendor if a flaw is reported in its product. We do not include the vulnerabilities reported on Bugtraq since most of these are not confirmed vulnerabilities at the time they are posted online.

A limitation of our study is that most of the data points in our sample are announcements regarding off-the-shelf software products. Our analysis does not cover software development projects where a security flaw can cause millions of dollars worth of damage. The main reason for excluding them was the lack of availability of data on software

failures in such cases. We also reiterate that further analysis in terms of software quality, market share or profitability is needed to fully understand how vulnerability disclosure signals poorer quality and how it affects the vendors' incentives to provide better quality software.

References

- Adams EN III (1980) 'Minimizing Cost Impacts of Software Defects' *IBM Research Report,* RC 8228 April
- Anderson R (2001) 'Why Information Security is Hard – an Economic Perspective' *Proceeding of 17th Annual Computer Security Applications Conference,* New Orleans, Louisiana
- Arora, A., Caulkins, J.P. and R Telang, (2004). Provision of Software Quality in the Presence of Patching Technology, Carnegie Mellon University, working paper.
- Arora, A., Telang, R. and H Xu, (2004). 'Optimal Policy for Software Vulnerability Disclosure', Carnegie Mellon University, working paper.
- Banker RD and SA Slaughter (1997) 'A Field Study of Scale Economies in Software Maintenance', *Management Science*, **43(12),** 1709-1725
- Barki H, Rivard S and J Talbot 'Toward an Assesment of Software Development Risk', *Journal of Management Information Systems,* **10(2),** 203-225
- Brown SJ and JB Warner (1985) 'Using Daily Stock Returns: The Case of Event Studies', *Journal of Financial Economics,* **14(1985),** 3-31
- Campbell JY, Andrew WL and AC MacKinlay (1997) 'The Econometrics of Financial Markets' Princeton University Press
- Campbell K, LA Gordon LA, Loeb MP and L Zhou (2003) 'The Economic Cost of Publicly Announced Information Security Breaches: Empirical Evidence from the Stock Market', *Journal of Computer Security,* **11(3),** 431-448
- Cavusoglu H, Mishra B and S Raghunathan (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),** 69
- Chatterjee D, Richardson VJ and RW Zmud (2001) 'Examining the Shareholder Wealth Effects of Announcements of Newly Created CIO Positions', *MIS Quarterly,* **25(1),** 43-70
- Cusumano, MA (2004) 'Who is Liable for Bugs and Security Flaws in Software?' *Communications of the ACM,* **47(3),** 25-27
- Davidson WL III, DL Worrell(1992) 'The Effect of Product Recall Announcements on Shareholder Wealth' *Strategic Management Journal,* **13(6),** 467-473
- Dos Santos BL, Peffers K and D Mauer (1993) 'The Impact of Information Technology on the Market Value of the Firm', *Information Systems Research,* **4 (March),** 1-23
- Gal-Or, E. & A Ghose (2003). 'The Economic Consequences of Sharing Security Information', In *2nd Workshop on Economics and Information Security*, May 29-30.
- Gordon LA & MP Loeb(2002) 'The Economics of Information Security Investments' *ACM Transactions on Information and Systems Security,***5(4),**438-457
- Gordon, L.A., Loeb, M.P. & Lucyshyn, W. (2002) 'An Economic Perspective on the Sharing of Information Related to Security Breaches: Concepts and Empirical Evidence', In *The 1stWorkshop on Economics and Information Security*, May16-17.
- Gordon LA and MP Loeb and T Sohail (2003) 'A Framework for Using Insurance for Cyber Risk Management' *Communications of the ACM,* **46(3),** 81-85
- Harter DE, Krishnan MS and Slaughter SA (2000) 'Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development' *Management Science,* **46(4),** 451-466
- Hendricks KB and Singhal VR (1996) 'Quality Awards and the Market Value of the Firm: An Empirical Investigation' *Management Science,* **42(2),** 415-436
- Hsiao, C (2002) 'Analysis of Panel Data' Cambridge University Press
- Jarrell G and S Peltzman (1985) 'The Impact of Product Recalls on the Wealth of Sellers' *The Journal of Political Economy,* **93(1),** 512-536
- Kannan K and R Telang (2004) 'Market for Software Vulnerabilities? Think Again.' *Management Science* (Forthcoming).
- McGraw G (2004) 'Software Security' *IEEE Security and Privacy,* **2(2),** 80-83
- Paulk M Weber C, Curtis W and Chrissis M (1994) 'The Capability Maturity Model: Guidelines for Improving the Software Process' *Carnegie Mellon University : Software Engineering Institute*
- Pfleegar CP (1997) 'The Fundamentals of Information Security' *IEEE Software*, **14(1),** 15-17
- Potter B and G McGraw (2004) 'Software Security Testing' *IEEE Security and Privacy,* **2(5),** 81-85
- Slaughter SA, DE Harter and MS Krishnan (1998) 'Evaluating the Cost of Software Quality' *Communications of the ACM*, **41(8),** 67-73
- Wallace L, Keil M and A Rai (2004) 'How Software Project Risk Affects Project Performance: An Investigation

of the Dimensions of Risk and An Exploratory Model', *Decision Scineces,* **35(2),** 289-321
• Wang H and C Wang (2003) 'Taxonomy of Security Considerations and Software Quality' *Communications of the ACM,* **46(6),** 75-78
• Westland, JC (2003) 'The Cost Behavior of Software Defects', *Decision Sciences,* **37,** 229-238