

# Competition and Patching of Security Vulnerabilities: An Empirical Analysis<sup>1</sup>

Ashish Arora  
Fuqua School of Business  
Duke University  
1 Towerview Drive  
Durham, NC 27708  
[ashish.arora@duke.edu](mailto:ashish.arora@duke.edu)

Chris Forman  
College of Management  
Georgia Institute of Technology  
800 West Peachtree St. NW  
Atlanta, GA 30308  
[chris.forman@mgt.gatech.edu](mailto:chris.forman@mgt.gatech.edu)

Anand Nandkumar\*  
Indian School of Business  
Gachibowli, Hyderabad 500 032  
India  
[anand\\_nandkumar@isb.edu](mailto:anand_nandkumar@isb.edu)

Rahul Telang  
H. John Heinz III College  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213  
[rtelang@andrew.cmu.edu](mailto:rtelang@andrew.cmu.edu)

## Abstract

We empirically estimate the effect of competition on vendor patching of software defects by exploiting variation in number of vendors that share a common flaw or common vulnerabilities. We distinguish between two effects: the direct competition effect when vendors in the same market share a vulnerability, and the indirect effect, which operates through non-rivals that operate in different markets but nonetheless share the same vulnerability. Using time to patch as our measure of quality, we find empirical support for both direct and indirect effects of competition. Our results show that ex-post product quality in software markets is not only conditioned by rivals that operate in the same product market, but by also non-rivals that share the same common flaw.

Keywords: Information security; Competition; Software Quality; Vulnerabilities  
JEL Classification: L10, L15, L86

---

\* Corresponding author. Author names are in alphabetical order.

<sup>1</sup> We thank Avi Goldfarb, Tomas Roende, and seminar participants at the University of Maryland, Carnegie Mellon University, the International Industrial Organization Conference, the ZEW, and the Workshop on the Economics of Information Security for helpful comments. We further thank CERT/CC for providing essential data. This research was partially supported by a grant from Cylab, Carnegie Mellon University, to Ashish Arora and Rahul Telang. Anand Nandkumar thanks the Software Industry Center at Carnegie Mellon University for financial support. Rahul Telang acknowledges generous support of National Science Foundation through the CAREER award CNS-0546009. Chris Forman acknowledges the support of the Sloan Foundation through an Industry Studies Fellowship. All errors are our own.

## 1. Introduction

Many, if not most, cyber attacks exploit software defects (see for example Arbaugh et al. 2000). It is widely believed that poor software quality is an outcome of the market power that software vendors enjoy. However, there is little, if any, empirical work that examines the relationship between software quality and the degree of competition. One of the main difficulties in undertaking such empirical work is the lack of variation in number of competitors. Almost all software product markets are national, if not global, making it difficult to estimate the effects of competition using regional variation in competition, as is commonly done for other industries. A second key challenge is to measure quality.

In this paper, we use two unique features of our data to overcome these challenges. First, we use the time taken by vendors to release a patch for a software vulnerability as our proxy for quality. Although all concede that it is better that a product not have bugs in the first instance, this is unrealistic and perhaps may even be too costly. Thus, patches are an important component of post-sales product support and timely patch release is an important part of overall information security (Arora, Caulkins and Telang 2006; Beattie et al. 2002).

Second, we use variation in the number of vendors affected by a common vulnerability (a vulnerability that affects products manufactured by different vendors, discussed in detail later) to empirically estimate the effects of competition. In particular, this variation enables us to examine how number of vendors affected by a vulnerability influences patch release behavior of software vendors. A timely patch is vital in limiting losses from cyber-attacks, which are increasing in the time elapsed between the initial disclosure of the vulnerability and the release of the patch. Tardy patching likely reduces customers' willingness to pay for a vendor's current and future products. Therefore, how quickly a vendor will release the patch should depend upon how accurately customers are able to judge whether the vendor's patches were tardy or not, and also the choices available to the customer.

The degree of competition faced by the vendor affects both of these factors. For instance, in deciding how tardy a vendor is, customers may compare it to how quickly other vendors in the same market (henceforth rivals) provided a patch for the vulnerability. Thus, the number of rivals who are also developing a patch is one dimension of the degree of competition, and we label this the direct effect. Customers may also look at the patching performance of vendors affected by the same vulnerability but operating in other markets (henceforth non-rivals) in assessing the timeliness of the patch provided by their vendor. Thus, the number of non-rivals affected by the common vulnerability measures a different dimension of the degree of competition, which we label the indirect effect. Competition has a third dimension as well, which we call the disclosure effect. When a vendor releases a patch, it de facto discloses the vulnerability to all, and attackers can exploit all unpatched machines. This affects both rivals and non-rivals who are working to release their patches.

To test the relationship between competition and quality, we examine responses by 21 vendors to 241 vulnerabilities reported to CERT/CC from September 2000 to August 2003. Our results demonstrate that the direct, indirect, and disclosure effects play a significant role in shaping the speed with which vendors release patches to software.

Our research makes a significant contribution to our understanding of vendor's investment in software quality, in particular recent work in the information security literature that has examined vendor patch release behavior (Arora, Krishnan, Telang, and Yang 2006, Cavusoglu, Cavusoglu, and Raghunathan 2005; Choi, Fershtman, and Gandal 2005; Li and Rao 2007). To our knowledge our research is the first to demonstrate how increases in disclosure threats from rivals and non-rivals influences investments in information security and software quality. Our paper is also unique in demonstrating the relationship between competition and software quality. In particular, our research demonstrates that despite high levels of concentration in many software markets, the competition from vendors in related markets works to reduce patch

release times. This indirect form of competition can be as effective in reducing time to patch as increases in the number of direct competitors.

## **2. Related Literature and Contribution**

This paper is related to four streams of research: economics of information security, software quality and software process, competition and quality provision, and competition in technologically related markets.

There is relatively little work that focuses on managerial or organizational issues in the information security domain (Straub et al. 2008). Only recently have researchers started investigating important economic questions in the areas of information security. Our research is motivated by theoretical models of the relationship between the timing of vulnerability disclosure and the expected losses from attacks (Schneier 2000; Arora, Telang, and Xu 2008; Cavusoglu, Cavusoglu, and Raghunathan 2005) and more broadly research that has studied the factors shaping the timing and nature (public or private) of vulnerability disclosure by firms and third parties (Kannan and Telang 2005, Nizovtsev and Thursby 2007; Choi, Fershtman, and Gandal 2005).

More recently, researchers have also focused on understanding the behavioral aspects of computer security (e.g. Loch et al.1992; Straub and Welke 1998). For example, Straub (1990) showed that investments in countermeasures reduced losses from computer abuse. Tucker and Miller (2008) show that concerns related to privacy and information security can inhibit diffusion of networked IT. Harn, Kai-Lung, Sang-Yong and Ivan (2007) evaluate the effectiveness of various online privacy policies using an information processing theory approach. Recently some empirical work has examined the economic implications of vulnerability disclosure. Arora, Nandkumar, and Telang (2006) find that disclosure of information about vulnerabilities increases frequency of attacks, especially if the patch is not available. Even the release of a patch results in a temporary increase in attacks but a sharp decline thereafter, resulting in a lower average attack

frequency. Arora, Krishnan, Telang and Yang (2008) use a dataset assembled from CERT/CC's vulnerability notes and SecurityFocus database to show that early disclosure leads to faster patch release times. Telang and Wattal (2007) use an event study methodology to show that vulnerability disclosure leads to a loss of market value. Li and Rao (2007) empirically examined the role of private intermediaries on the timing of patch release by vendors and found that the presence of private intermediaries decreases vendors' incentive to deliver timely patches. Our research is similar to prior work in that we examine the economic outcomes from vulnerability disclosure. However, in contrast to the prior work in this area, we study the relationship between competition and vendor patch release times.

The software community has long been concerned with the determinants of software quality. The literature has examined the link between quality and software development process (e.g., Banker, Davis, and Slaughter 1998; Harter, Krishnan, and Slaughter 2000; Agarwal and Chari 2007). These studies conclude that a higher level of software process maturity is associated with better software. Our study is different from this prior work in two important aspects: First, we focus on ex-post quality rather than pre-release software quality. Second, in an advance over the literature, we explicitly examine the link between software quality and competition.

While a rich theory literature has examined the link between competition and quality, empirical work has been limited due to the inherent challenges of measuring product quality.<sup>2</sup> In general, prior work has demonstrated that increases in competition lead to better quality provision (e.g., Domberger and Sherr 1989; Dranove and White 1994; Borenstein and Netz 1999; Hoxby 2000; Mazzeo 2003; Cohen and Mazzeo 2004). However, most prior work in this literature has focused upon services industries such as banking, legal or health services in which markets are local and empirical estimates are identified using cross sectional variation across geographic markets. In contrast, we examine this relationship within the context of a major product market,

---

<sup>2</sup> Prior theory work has demonstrated that increases in concentration can lead to an increase or decrease in product quality. For examples, see Gal-Or (1983), Levhari and Peles (1973), Schmalensee (1979), Swan (1970), and Spence (1975).

software, and obtain identification using variation in the number of products affected by software vulnerabilities.

While prior work has demonstrated a link between competition and product quality, it has not studied the interaction between firms in technologically related markets as we do. Recent work has highlighted the impact of firm strategic decisions in technologically related markets (e.g., Bresnahan and Greenstein 1999; Bresnahan and Yin 2006; Kretschmer 2005; West and Dedrick 2000). However, this research has focused on markets that are complements in demand. We argue that vendors who share common inputs will have important implications for vendors' quality decisions. To our knowledge, ours is one of the first papers to demonstrate empirically the interrelationships of strategic decisions among firms that share common inputs. Such interrelationships are likely to be particularly salient in software markets, where vendors in different market segments increasingly share common modules (e.g., Banker and Kauffman 1991; Brown and Booch 2002).

### **3. Conceptual Framework and Hypotheses**

Unlike defects in physical goods, software defects can be mitigated even after product release via patch release (Arora, Caulkins and Telang 2006). This makes both vulnerabilities in software, as well as patches that fix vulnerabilities, common among software products. The probability of a malicious attacker exploiting a specific vulnerability to compromise end user computers is positively related to the time the vulnerability remains without a fix. Thus, the timing of patches critically determines the extent of end user losses, and patches are perceived as a very important part of ex-post customer support.

Two considerations drive the timing of a vendor's patch: (1) how the cost of developing the patch increases as patch development is accelerated and (2) the extent of user losses and the fraction of these losses that the vendor internalizes. Typically, an early patch entails higher costs but also reduces customer losses. Two main factors condition the extent of end user losses

internalized by a vendor: the extent of market competition and the number of end users (or market size).

The first factor that influences the timeliness of patch release is the degree of competition. Prior literature from other industries has shown that increases in competition are associated with greater product quality due to firm efforts to vertically differentiate themselves to win customers over from their competitors (Domberger and Sherr 1989; Dranove and White 1994; Mazzeo 2003; Cohen and Mazzeo 2004). In our research, we measure the effects of competition on quality by examining how increases in the number of other firms affected by a vulnerability influence patching times. Greater competition, especially from rivals, implies that end users are more likely to penalize lagging vendors due to the availability of a number of alternatives, or greater loss in reputation for the lagging vendor.

*Hypothesis 1: Vendors that face more rivals affected by the same vulnerability are more likely to release a quicker patch.*

In many cases, a newly discovered vulnerability could affect many different products (for future reference we label these common vulnerabilities). A common vulnerability is typically due to a shared code base or design specification, or due to a proprietary extension of a widely used software component. For instance, a stack buffer overflow vulnerability in Sendmail (a commonly used mail transfer agent)<sup>3</sup>, disclosed in 2003, affected the following vendors: Apple, Conectiva, Debian, FreeBSD, Fujitsu, Gentoo Linux, Hewlett-Packard, IBM, MandrakeSoft, Mirapoint, NetBSD, Nortel Networks, OpenBSD, OpenPKG, Red Hat, SCO, Sendmail Inc., Sequent (IBM), SGI, Slackware, Sun Microsystems, SuSE, The Sendmail Consortium, Wind River Systems, and Wirex. Some of the products produced by these vendors potentially compete with one another while others are in very distinct markets. For example, Wirex and Mirapoint

---

<sup>3</sup> Vulnerability number VU#897604 by CERT/CC classification. See <http://www.kb.cert.org/vuls/id/897604> (accessed 09/22/2006).

produce email products, Wind River produces embedded software, while many of the other products are operating systems. Even among the latter, there is considerable variation in the hardware platforms used. However, all these products use Sendmail code, and hence were affected by the vulnerability in it.

When a vulnerability is common to many products, customers can compare how quickly the vendor releases the patch relative to vendors affected by the vulnerability but operating in different markets (non-rivals henceforth; we will refer to such competition as “indirect” competition). Thus the number of vendors operating in a different market is also likely to influence the timing of patch release by a vendor. While such comparisons have been a recurring theme in the literature on competition and product quality, our setting is unique in that comparisons may occur among firms in different market segments.

*Hypothesis 2: Vendors that face a larger number of indirect competitors affected by the same vulnerability are more likely to release a quicker patch.*

In addition to letting customers judge more precisely whether their vendor is releasing patches in a timely manner, the number of other firms affected by the same vulnerability affects patching behavior through another route that we label the disclosure effect. Users’ expected losses from software vulnerabilities will be higher when these vulnerabilities have been publicly disclosed, because disclosure makes it easier for attackers to find vulnerabilities (Arbaugh et al 2000; Arora, Nandkumar, and Telang 2006). Because software vendors internalize some fraction of users’ losses, vulnerability disclosure is associated with briefer times to patch release (Arora, Krishnan, Telang, and Yang 2006). Prior work has focused on how public disclosure of vulnerabilities by third party intermediaries shapes patch release behavior (Schneier 2000; Arora, Telang, and Xu 2008; Cavusoglu et al 2005). While disclosure of vulnerabilities by third parties is important, a vulnerability can also be disclosed when someone issues a patch for it.

In deciding how expeditiously to develop and release a patch, a firm must consider how quickly the vulnerability is likely to be disclosed. The greater the number of firms (rivals or non-rivals) affected by the vulnerability, the more likely that someone will release a patch disclosing the vulnerability. In other words, the greater the number of vendors affected by a vulnerability, the greater the threat of disclosure, and hence, the more expeditiously each vendor will try to develop and release a patch.

Thus, increases in the number of affected vendors will also influence patch release times indirectly through disclosure. We label this the disclosure effect. Below, we describe in detail how this is identified separately from the effects of direct and indirect competition.

*Hypothesis 3: Vendors facing a greater threat of disclosure are likely to release a patch sooner.*

In focusing on these three dimensions of the degree of competition, we are mindful that we are neglecting the most obvious dimension, namely the sheer number of sellers in the market, whether or not they are affected by a given vulnerability. When there are many competing products, end users have more choices, and thus, future sales of a product may be more sensitive to perceived quality (Levhari and Peles 1973; Spence 1975; Schmalensee 1979). This neglect reflects our data: There is no variation in the total number of sellers within a market. In order to account for this and other market specific effects, we use product market dummies in our empirical specification.<sup>4</sup>

A second factor that determines the total customer losses incurred due to the vulnerability is the number of end users (or market size). Roughly speaking, a greater number of end users increases the total losses from the vulnerability and may also increase the attractiveness of the vulnerability to a malicious attacker. In general, attackers prefer exploiting popular products relative to obscure ones. Thus the likelihood that the vulnerability is exploited by an attacker is

---

<sup>4</sup> As a robustness check, we also re-estimated our results using a single market (operating systems), with very similar results, indicating that any potential bias is very small.

likely to be higher for popular products (Honeynet Project 2004, Symantec 2004). Moreover, the sheer number of end users also implies greater monetary losses that a vendor internalizes from a vulnerability. While other research has explored how vendor size influences the speed with which vendors release patches (Arora, Krishnan, Telang, and Yang 2006), to our knowledge we are the first to investigate how market size influences time to patch release.

*Hypothesis 4: Vendors with larger market size are likely to release a patch sooner.*

#### **4. Data and Variables**

We assembled our data set of vulnerabilities from notes published by CERT/CC.<sup>5</sup> The vulnerabilities analyzed in this study were published by CERT between September 2000 and August 2003. On average, about 3000 vulnerabilities are reported to CERT/CC in a year, of which only about 10% are deemed legitimate and significant enough to be published. After determining if a reported vulnerability is authentic and exceeds CERT/CC's minimum threshold value for severity, as measured by the CERT METRIC (described later), CERT/CC staff contact vendors that in CERT's view, may be affected by the vulnerability. CERT tends to contact as many vendors as possible even those who it suspects may be remotely affected by the vulnerability. Vendors then respond back to CERT whether they are vulnerable or not and in many cases with a list of products that are affected by the vulnerability. A vendor's response can typically be one of the following. The vendor may acknowledge the vulnerability in its product(s). In this case, CERT/CC lists the product's status as "vulnerable." The vendor may report that the product is not vulnerable, in which case CERT/CC lists the vendor's status as "not vulnerable." The vendor may also choose not to respond: In this case, CERT/CC records the vendor's status as "unknown."

---

<sup>5</sup> Other data sources such as online forums do not usually give a "protected period" to vendors to patch vulnerabilities before disclosing them publicly. Also, other sources also do not verify vulnerabilities in the same way that CERT does.

Our unit of observation is a vendor – vulnerability pair. Our goal is to estimate the influence of competition on how long an affected vendor takes to provide a fix for the vulnerability. Given CERT’s strategy of contacting many vendors, even those who may not be affected, we only considered vendors that acknowledged that their product(s) were vulnerable as those that were affected by the focal vulnerability. It is nonetheless quite plausible that a vendor might in fact be affected by the focal vulnerability even when CERT lists the status for a vendor-vulnerability pair as “unknown.” However, we have no practical way of ensuring of whether the vendor was actually affected or not in such cases. Hence, the resulting bias, if any, cannot be determined.<sup>6</sup> Our sample consists of 1714 vendor-vulnerability pairs that were listed as “vulnerable” by CERT/CC. From this set, we dropped observations that relate to non-commercial entities (such as universities and not-for-profit vendors) and foreign vendors (vendors that do not have significant sales in the US).<sup>7</sup> Non-commercial entities may have other objectives other than just maximizing profits and hence may not be subjected to the same pressures as for-profit vendors. Moreover, we are also unable to measure market size for non-commercial entities or foreign vendors reliably.<sup>8</sup> Many of the non-commercial entities have very few corporate customers, which, as we will explain later, is our measure of quantity. Also many of the foreign vendors typically have customer bases that mainly consist of non-US customers, while our measure of quantity is based on US corporate customers. We hence do not include observations that relate to such vendors in our empirical analysis. However, as we will explain in detail later, we include both non-commercial as well as foreign vendors in our measures of competition.

---

<sup>6</sup> Arora *et al.* (2006) suggest that these many of the vulnerabilities that were not acknowledged by vendors may not be genuine. However, in cases where the vulnerability may have been genuine, but were not acknowledged by the focal vendor, the focal vendor was unlikely to fix the vulnerability.

<sup>7</sup> The list of eliminated vendors and non-commercial entities consists of Apache, BSD (FreeBSD, OpenBSD), Debian, GNU, Gentoo, ISC, KDE, MIT Kerberos, OpenAFS.org, OpenLDAP project group, OpenBSD that makes OpenSSH, OpenSSL project group, Openwall GNU Linux group, Samba Team, Sendmail Inc., Slackware, Sorcerer Linux, Stunnel, Tcpdump.Org, The Linux Kernel Archives, Trustix, University of Washington, XFree86, Xpdf, Yellow Dog Linux, mod ssl and zlib.org.

<sup>8</sup> Foreign vendors include Mandrake Linux that is headquartered in France and Turbo Linux, headquartered in Japan. The results are qualitatively unchanged even if we include foreign vendors.

We also removed protocol vulnerabilities from the sample, as patches to these vulnerabilities typically involve protocol changes whose scope extends beyond a particular product. In many cases, even if a vendor knew the existence of such vulnerabilities, fixing it involves changes not just to the product but also the underlying protocol, which may involve cooperation of other involved parties. Protocol vulnerabilities thus typically do not confirm to the phenomenon considered in this paper. Finally, we dropped observations wherein the vendors discovered and disclosed the vulnerability to CERT/CC of its own accord along with a patch. In such cases, since we cannot reliably measure when the vendor came to know of the existence of the vulnerability, we cannot also reliably determine how long the focal vendor took to release a patch. Our final sample includes 241 distinct vulnerabilities and 461 observations.

We use variance in the manner with which vulnerabilities are disclosed to identify the competition and disclosure effects. From CERT/CC data (and discussions with CERT/CC staff), we know the date when a vendor is notified of the vulnerability. CERT/CC also records if and when the vulnerability was publicly disclosed. Thus, we label vulnerabilities as instantly disclosed if the existence of the vulnerability had been publicly disclosed (by some third party) prior to CERT/CC's notification to the vendor. We label vulnerabilities as non-instantly disclosed when CERT/CC discloses a vulnerability that had previously not been publicly disclosed.

#### **4.1 Dependent Variable**

Our dependent variable is DURATION, a measure of the number of days a vendor takes to release the patch. Measurement of DURATION depends on the regime of disclosure – instant or non-instant. If the vulnerability is instantly disclosed, DURATION is the elapsed time in days between the date when the vulnerability was publicly disclosed and the date when the vendor released the patch. If the vulnerability is non-instantly disclosed, DURATION is the elapsed time between the CERT/CC notification to the vendor and the date when the vendor released the patch. For the empirical analysis we use the log of  $(1 + \text{DURATION})$  as our dependent variable. We

label this variable LOGDURATION. Of the 461 observations in our sample, 4.3%, or about 20 observations, had no patch. For these unpatched observations, we assign the maximum value of LOGDURATION that we observed in our sample (8.27). As we will show, our results are unchanged when we use a tobit model that treats these observations as right censored. Table 2 provides the descriptive statistics for LOGDURATION.

## 4.2 Independent Variables

A description of all independent variables is included in Table 1, while descriptive statistics are included in Table 2.

< TABLE 1 ABOUT HERE >

*Competition:* To measure how the different dimensions of the degree of competition influence patch release times, we construct three variables. RIVALS is the number of vendors that CERT lists as vulnerable and that operate in the same product market. NON-RIVALS is the number of vendors that are vulnerable but operate in a different market. VENDORS is the sum of RIVALS and NON-RIVALS and is the total number of vendors listed as “vulnerable” by CERT for a specific vulnerability. We determined rivals and non-rivals using market definitions in the Harte-Hanks CI Technology database (hereafter CI database).<sup>9</sup> As an example, suppose the focal vendor-vulnerability pair was Microsoft-Windows XP vulnerability and the vulnerability was shared by products produced by Red Hat and Oracle. In this case, RIVALS consists of Red Hat (since both Red Hat and Microsoft are in the operating system market). NON-RIVALS consist of Oracle, while VENDORS consists of both Red Hat and Oracle. As explained earlier, although we exclude the observations that relate to non-commercial entities and foreign vendors, we include both groups in our measures of competition. As a robustness check, we have re-estimated our regressions using measures of competition excluding non-commercial and foreign vendors and the results are qualitatively similar.

---

<sup>9</sup> In those cases where the product was not included in the database, we examined product manuals to classify the product.

*Quantity:* Data on cumulative sales quantity for a product was collected using 2002 data from the CI database. The database reports only binary decisions of software use in a firm or establishment: details on number of copies of a software product are not reported. To develop a measure of the total installed base of a software product, we use the number of firms that indicated use of the product and weighted it by the number of employees in the organization. For instance if 1000 establishments own at least 1 licensed copy of Red Hat Linux, and each establishment has 500 employees, our measure for quantity would be 500,000, which is the aggregate number of employees in those firms. This puts more weight on products used in larger firms, and arguably provides us a more accurate proxy for quantity. Finally, we follow Forman, Goldfarb, and Greenstein (2005) and weight our data using County Business Patterns data from the U.S. Census, because the CI database oversamples certain industry sectors. In sum, to compute our final measure of quantity, we multiply the binary measure of software use for each firm by the number of firm employees and by firm weights. We then sum across firms. As the distribution of quantity is highly skewed, we take the log of quantity (LOGQUANTITY) for our analysis.

*Other variables:* In order to account for differences in severity of vulnerabilities we use the log of (one plus) CERT's severity metric, which we label LOGSEVERITY. The CERT severity metric is a number between 0 and 180, and is a comprehensive measure of the severity of vulnerabilities. We use this as our principal control for unobserved vulnerability characteristics. CERT uses an extensive set of criteria, including (i) whether information about the vulnerability is widely available; (ii) whether the vulnerability being exploited in the incidents has been reported to US-CERT; (iii) whether the Internet infrastructure is at risk because of this vulnerability; (iv) the number of systems on the Internet that are at risk from this vulnerability;

(v) the impact on users of an exploit; and (vi) the ease with which the vulnerability can be exploited, e.g., whether the vulnerability can be exploited remotely or not.<sup>10</sup>

Anecdotal evidence from industry sources suggests that quality testing of patches on multiple versions consumes additional time in the patch development process. Thus, we also control for the log of the number of software versions that have been produced (LOGVERSIONS). Descriptive statistics for all of the independent variables are included in Table 2.

<TABLE 2 ABOUT HERE>

## 5. Empirical Models and Results

In this section, we describe our method for identifying how competition and disclosure influence vendors' patch release times. We also discuss the results of our baseline empirical analysis.

### 5.1 Empirical Model

Our goal is to examine how our proxy for ex post quality – namely the duration of patch release time for vendor  $i$  in market  $m$  facing vulnerability  $v$  – varies with changes in competition. If  $DIRECTCOMP_{iv}$  represents the effects of direct competition (competition from rivals),  $INDIRECTCOMP_{iv}$  represents that of indirect competition (competition from non-rivals), while  $DISCLOSURE_{iv}$  represents the effects of increased disclosure arising from a greater number of affected vendors, one may estimate the following linear model:

$$LOGDURATION_{imv} = \beta_0 + \beta_1 DIRECTCOMP_{iv} + \beta_2 INDIRECTCOMP_{iv} + \beta_3 DISCLOSURE_{mv} + \beta_4 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \theta_3 K_m + \varepsilon_{imv} \quad (1)$$

---

<sup>10</sup> See [www.kb.cert.org/vuls/html/fieldhelp](http://www.kb.cert.org/vuls/html/fieldhelp). (Last accessed on January 12, 2007)

where  $X_i$  is a vector of vendor characteristics that include vendor fixed effects for large vendors,  $K_m$  a vector of market fixed effects and  $Z_v$  is a vector of vulnerability characteristics that includes the severity metric. Our interest is in identifying the parameters  $\beta_1$  through  $\beta_4$  which correspond to hypotheses 1-4 and reflect the effects of direct competition, indirect competition, disclosure, and market scale respectively.

Competition enables users to compare and benchmark the performance of their vendor relative to others that share the same vulnerability. These effects, captured by  $\beta_1$  and  $\beta_2$  in equation 1, could arise either from rivals or non-rivals. Competition also allows users to switch to a vendor providing a superior mix of price and quality. This type of competition can only be provided by other sellers in the same product market, whether or not they share the vulnerability. Recall that we use market fixed effects to control for this and other unobserved factors that vary across markets.

We use dummies for the three largest markets, which account for 88% of the sample.<sup>11</sup> A small percentage (12%) of observations is from small markets that have insufficient observations to identify an individual market dummy. However, our results are robust to their exclusion. We also include firm dummies for the eight leading vendors, who jointly account for about 85% of the observations in our sample.<sup>12</sup> Estimates using only the top eight vendors with a full set of vendor fixed effects yield results similar to those reported.

We assume that LOGQUANTITY is statistically exogenous. In support of this assumption we note that LOGQUANTITY reflects the stock of installations in the CI database in 2002, rather than the purchase quantity in any particular year. However, LOGQUANTITY may reflect recent demand, which may be correlated with unobservables that influence patch release times. If so, our estimates would overstate the relationship between cumulative sales and quality

---

<sup>11</sup> These include dummies for the operating system, application server and web browser markets

<sup>12</sup> These are Apple, HP (includes HP, Compaq, and Digital), Microsoft, Sun, SCO, RedHat, IBM (includes Lotus, iPlanet, and IBM) and Oracle. The omitted category consists of smaller vendors, that do not have sufficient observation to identify an individual vendor dummy. These are Adobe, SGI, Allaire, Macromedia, Netscape, Network Associates, Novell, Symantec, Trend Micro, and Veritas.

provision, and potentially bias other estimates as well. However, re-estimating the models after excluding LOGQUANTITY yields very similar estimates for other variables, indicating that the bias, if any, does not extend to other estimates.

The effect of LOGQUANTITY on patch release times may be different for software vendors that also sell hardware: Such firms may also internalize the effect of vulnerable software on related hardware sales. For example, vulnerabilities in Sun's Solaris operating system may influence sales of its workstations too, shifting the relationship between installed base of Solaris and patch release times compared to other software firms. To capture these potential differences, we interact LOGQUANTITY with a vendor hardware dummy that is equal to one when a software vendor also sells hardware (HARDWARE).<sup>13</sup> Re-estimation of models without including HARDWARE and HARDWARE\*LOGQUANTITY as covariates yields similar estimates of  $\beta_1$  and  $\beta_2$ , although it yields different estimates of  $\beta_4$ .

A key issue in estimation is controlling for heterogeneity across vulnerabilities. Arguably, LOGSEVERITY is a very good summary measure, but perhaps not perfect. For instance, it may not account for differences in the complexity of fixing vulnerabilities. Vulnerability fixed effects are infeasible since our sample consists of 461 vendor-vulnerability pairs with 241 distinct vulnerabilities. Hence, we estimate a random effects GLS specification.<sup>14</sup>

## 5.2 Identification using variation in rivals

We use several approaches to identify the coefficients  $\beta_1$  through  $\beta_4$  to improve confidence in our estimates. We begin with a simple comparison of sample means and then proceed with discussing the results of the regressions. In Table 3, we provide some preliminary evidence on the effects of competition through an examination of conditional means.

<TABLE 3 ABOUT HERE>

---

<sup>13</sup> In the dataset the hardware vendors are HP (includes Compaq and Digital Equipment Corporation.), Sun Microsystems and IBM

<sup>14</sup> We test for, and are unable to reject, the presence of unobserved heterogeneity

We categorize RIVALS as “high” if the number of affected RIVALS for a vulnerability was above the median and “low” otherwise. An increase in the number of RIVALS from below the median to above the median lowers LOGDURATION by a statistically significant 0.92.

Next we present regression results where we simply estimate how variation in the number of affected rivals affects patching time. Thus, we only capture one dimension of competition, from the RIVALS.

$$LOGDURATION_{imv} = \beta_0 + \beta_1 RIVALS_{iv} + \beta_4 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \theta_3 K_m + \varepsilon_{imv} \quad (2)$$

We estimated equation (2) using OLS. The Breusch-Pagan test overwhelmingly rejects the assumption of homoskedasticity ( $\chi^2$  145.68; p-value=0.00). One significant source of heteroskedasticity could be the presence of unobserved differences between vulnerabilities. As noted above, we estimate random effects models to allow for the presence of such vulnerability-specific unobservables.

<TABLE 4 ABOUT HERE>

In table 4 we present three sets of estimates. In column (1), we estimate equation 2 using a sub-sample comprising of observations from the operating system market (OS sample henceforth). In columns (2) and (3) we present estimates of the full sample with and without market dummies. These results suggest that an increase in the number of rivals decreases patch release times by about 7-9% or between 12-15 days per rival. The estimated effect is stable to the addition of market dummies. Quantity also decreases patching times: A 10% increase in quantity is associated with about a 1.3% decrease in patch release times or about 2 days. Thus, our analysis supports hypotheses 1 and 4.

A point to note is that the results are similar if we only use the sample of vendors in the operating system (OS) market. In part, this reflects the dominance of operating systems

vulnerabilities in our sample. It does imply that the (unaccounted for) variation in the number of sellers across different product markets is not a major source of concern.

### 5.3 Identification using variation in rivals, non-rivals and disclosure

The specification laid out in equation (2) ignores non-rivals. It also ignores the disclosure threat that might arise from both rivals and non-rivals. In this section, we expand the specification to explore both of these additional dimensions of competition.

One challenge we face is to separately identify the direct and indirect effects of competition from the disclosure effect. The answer lies in another source of variation in our data. The conceptual framework outlined in section 3 has the following implications: the threat of disclosure arises from increases in the number of rivals and non-rivals affected by the same vulnerability and arises only under non-instant disclosure. Put another way, when many vendors are affected by the vulnerability and the vulnerability has not yet been publicly disclosed, affected vendors face a disclosure threat (over and above direct and indirect competition threat). In contrast, when the vulnerability is already disclosed (instant disclosure), vendors begin to develop patches knowing that attackers are also aware of the vulnerability, the direct and indirect competition effects operate but the disclosure threat is absent. Since some vulnerabilities in the sample are instantly disclosed and others are not, this provides the basis for the identification strategy shown below:

$$\begin{aligned}
 LOGDURATION_{imv} = & \beta_0 + \beta_1 RIVAL S_{iv} + \beta_2 NONRIVAL S_{iv} + \\
 & \beta_3 (1-INSTANT_v) * (RIVAL S_{iv} + NONRIVAL S_{iv}) + \beta_4 LOGQUANTITY_{im} + \\
 & \theta_1 X_i + \theta_2 Z_v + \theta_3 K_m + \varepsilon_{imv}
 \end{aligned} \tag{3}$$

Collecting terms gives us the following estimating equation:

$$\begin{aligned}
 LOGDURATION_{imv} = & \beta_0 + \gamma_1 RIVAL S_{iv} + \gamma_2 NONRIVAL S_{iv} + \\
 & \gamma_3 INSTANT_v * (RIVAL S_{iv} + NONRIVAL S_{iv}) + \beta_3 INSTANT_v + \beta_4 LOGQUANTITY_{im} \\
 & + \theta_1 X_i + \theta_2 Z_v + \theta_3 K_m + \varepsilon_{imv}
 \end{aligned} \tag{4}$$

where  $\gamma_1 = \beta_1 + \beta_3$  represents the combined effects of competition and disclosure that arises from increases in the number of rivals and  $\gamma_2 = \beta_2 + \beta_3$  is the combined effects of indirect competition and disclosure from non-rivals that operate in related markets.  $\gamma_3 = -\beta_3$  represents the effect of disclosure threats that arise from rivals and non-rivals alike. Variation in the number of rivals affected by the vulnerability identifies  $\beta_1$ . Likewise, identification of  $\beta_2$  arises from variation in the number of non-rivals affected by the vulnerability. Identification of  $\beta_3$  utilizes variation between vulnerabilities in the mode of disclosure as well as variation in the number of rivals and non rivals affected by a vulnerability. One concern with this identification strategy is possibility that INSTANT may be endogenous: instantly disclosed vulnerabilities may differ in some unobservable way that influences patch release times. We later relax the assumption that INSTANT is exogenous by instrumenting for it, and also explore an alternative identification strategy that does not rely upon INSTANT.

The results are presented in Table 5. Note that equation (4) implicitly constrains the disclosure threat from rivals and non-rivals to be similar. The  $\chi^2$  test fails to reject this constraint ( $\chi^2(1)=1.21$ ; p-value = 0.27). Column (1) shows that the combined effect of direct competition and disclosure from rivals,  $(\beta_1 + \beta_3)$ , is about a 9% decrease in patch release time per rival or about 15 days. Likewise the combined effect of non-rivals,  $(\beta_2 + \beta_3)$ , is about 10% per non-rival or about 17 days. Both these coefficients are significant at the 1% level. The effect of disclosure is about 4%, (or about 7 days) which is statistically significant at the 10% level. The coefficient of  $\beta_4$  suggests that a 10% increase in quantity is associated with a 1.5% decrease in patch release

times, or about 3 days. The estimates of  $\beta_1$  and  $\beta_2$  are about 5% (8 days) per rival and about 6% (10 days) per non-rival respectively. These estimates are also significant at the 10% level.

<TABLE 5 ABOUT HERE>

Also, the estimates of the market dummies are consistent with the notion that vendors release patches earlier in more competitive markets. For instance, in column (1), the estimate of the dummy for the Operating System market, with a total of 23 vendors, is -1.75 (0.70). The corresponding estimate for the application server market, with three vendors, is -0.26 (0.37).

To summarize, both rivals and non-rivals have an economically significant effect on a vendor's decision of when to release a patch for a vulnerability. Also, the effects of rivals and non-rivals are similar in magnitude; the combined effect of competition and disclosure effects both from rivals and non-rivals is about 15-17 days. Further, increases in quantity also reduce duration by about 3 days. In short, our results provide support for hypotheses 1 through 4.

## **6. Robustness Checks**

In this section, we outline some of the additional analyses we undertook to examine the robustness of our estimates to various assumptions. First we examine the robustness of our results to an alternative strategy for treating right-censored observations. Second, we check the robustness of our results to the assumption that INSTANT is exogenous by presenting the results of instrumental variable regressions. Third, since our baseline estimates require accurate measurement of RIVALS and NON-RIVALS, we show the results of another model that uses an identification strategy that does not rely on such measurement. Finally, we examine whether our results were driven by knowledge spillovers of how to patch the vulnerability rather than the effects of competition or disclosure.

### **6.1 Censoring**

We re-estimated equation (4) using a Tobit model in which unpatched observations are treated as right-censored. As with our baseline random effects GLS specification, the constraint

that the disclosure threat from rivals and non-rivals is identical cannot be rejected ( $\chi^2(1) = 1.96$ ; p-value = 0.16). In column (2) of table 5, we show the results of estimating equation (4) using a random effects Tobit specification. The point estimates of the effect of rivals and non-rivals ( $\gamma_1$  and  $\gamma_2$  respectively) are similar to that of the random effects GLS estimates and statistically significant at the 5% level. The combined effect from rivals is about 9% per rival or about 15 days while that from non-rivals is about 10% per non-rival or about 17 days. Both these coefficients are significant at the 5% level. The effect of disclosure from both rivals and non-rivals is also similar to that of the random effects GLS estimates – a statistically significant (at the 10% level) 4% or about 7 days. The coefficient of  $\beta_2$ , recovered using  $\gamma_2 + \gamma_3$ , is statistically significant at the 10% level and implies the effect of competition from one more non-rival is about 10 days. While  $\beta_1$  implies that the effect of competition arising from one more rival is 8 days, it is not statistically significant. The estimate of LOGQUANTITY ( $\beta_4$ ) suggests that a 10% increase in quantity is associated with a 1.9% decrease in patch release times or about 3 days.

We also conducted a Hausman test comparing the estimates of our baseline model in column (1) with that of the constrained Tobit model. The test rejects any systematic differences between the two specifications ( $\chi^2(21) = 5.97$ ; p-value 0.99). Hence we conclude that assigning the maximum value of LOGDURATION to observations for which the vendor did not release a patch does not result in biased estimates of competition and disclosure.

## 6.2 Potential endogeneity of instant disclosure

As noted above, identification in equation (4) is based on the assumption that INSTANT is exogenous. However, it is plausible that instantly disclosed vulnerabilities may differ in some unobservable way that influences patch release times. We present results that suggest that such endogeneity, if any, does not bias our estimates. We estimated an instrumental variables (IV) specification that uses instruments for INSTANT, INSTANT\*RIVALS and INSTANT\*NON-RIVALS which yields quantitatively similar estimates of  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ .

We use data on the identity of the identifier of the vulnerability as instruments for INSTANT. A vulnerability can be discovered by any of the following parties: end users, vendors, CERT/CC, universities or information security consultants. Identifiers of vulnerabilities have different incentives to publicly disclose vulnerabilities. For example, a consultant may be more likely to publicly disclose vulnerabilities relative to other identifiers since such disclosure may signal the consultant's technical ability. However, end users may be more likely to work with either vendors or CERT/CC since they are primarily concerned with minimizing losses from security incidents. Hence, the sources of discovery of vulnerabilities are likely to be correlated with type of disclosure but are unlikely to be correlated with duration of patching times. We use whether the vulnerability was discovered by a consulting firm (CONSULTANT), university (UNIVERSITY) or end user (USER) to predict INSTANT. We implemented the method outlined in Wooldridge (2002). We first estimated a probit regression in which we used the sources of discovery described above and other exogenous covariates in equation (4) to predict instant disclosure. We used the predicted value of the probit regression as an instrument for INSTANT, and interactions of the predicted value with RIVALS and NON-RIVALS as instruments for INSTANT\*RIVALS and INSTANT\*NON-RIVALS respectively and implemented the procedure suggested in Baltagi (1995).

Column (3) of Table 5 shows the results of the random effects IV model used to estimate equation 4. Tests for the power of the instruments suggest that the instruments are adequate ( $\chi^2$  statistic for INSTANT is 10.04; INSTANT\*RIVALS is 64.23; INSTANT\*NON-RIVALS is 64.44).<sup>15</sup> The coefficient estimate of RIVALS is statistically significant at the 10% level and suggests that one additional rival is associated with a 8% decline in duration times, or about 13 days. While the effect of disclosure ( $\beta_3$ ) is no longer statistically significant, the point estimate implies that one additional non-rival or rival is associated with a decrease in vendor patch release

---

<sup>15</sup> As before, we were unable to reject the constraint imposed by equation 4 that the disclosure effect of rivals is the same as non-rivals ( $\chi^2 = 0.39$ ; p-value 0.53).

times by 4% or about 6 days. Although the implied competition effects ( $\beta_1$  and  $\beta_2$ ) are also not statistically significant, the point estimates are similar to that of the random effects specification. The coefficient of LOGQUANTITY implies that a 10% increase in quantity is associated with a 1.7% decrease in patching time. A Hausman test comparing the coefficients of the baseline random effects model with that of the IV regression rejects any systematic differences between the estimates of the random effects and IV models ( $\chi^2$  1.72; p-value = 1.00). Hence the IV results suggest that endogeneity of instant disclosure, if any, does not affect our estimates of  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ .

### 6.3 Measurement error of rivals and non-rivals

The model outlined in equation (4) relied on an accurate definition of product markets. Any measurement error in rivals or non-rivals could potentially bias the estimates of direct and indirect competition. In this section, we present the results of estimates of  $\beta_1$ ,  $\beta_2$  and  $\beta_3$  obtained by solely exploiting variation in the mode of disclosure of vulnerabilities; we assume in this section that the effect of the marginal rival and non-rival are the same. This strategy does not require accurate measurement of rivals and non-rivals. However it constrains  $\beta_1$  to be equal to  $\beta_2$ . Recognizing that VENDORS is the sum of RIVALS and NON-RIVALS, the estimating equation can be written as

$$\begin{aligned} LOGDURATION_{imv} = & \beta_0 + \alpha_1 VENDOR_{S_v} + \alpha_2 INSTANT_{V_v} * VENDOR_{S_v} + \\ & \beta_4 LOGQUANTITY_{im} + \beta_5 INSTANT_{V_v} + \theta_1 X_i + \theta_2 Z_v + \theta_3 K_m + \varepsilon_{iv} \quad (5) \end{aligned}$$

where  $\alpha_1 = \beta_1 + \beta_3$  and  $\alpha_2 = -\beta_3$ . Under instant disclosure, only the direct and indirect competition effects are operative. When the vulnerability is not instantly disclosed, the disclosure effect is also operative. Hence,  $\alpha_2 = -\beta_3$  identifies how increases in the number of vendors will lower patch release times through disclosure. The separate effect of competition is thus  $\alpha_1 + \alpha_2$ .

Column (4) of Table 5 shows the results of random effects GLS estimation for equation (5). The estimate of  $\alpha_2$  is a statistically significant (at the 10% level) 0.06, and implies that one additional vendor is associated with a 6% decline in patch release times due to disclosure threat, or about 10 days. The coefficient of  $\alpha_1$  implies that one additional vendor is associated with a 10%, or about 17 days, decline in patch release times due to the combined effect of competition and disclosure. The separate effect of competition (recovered using  $\alpha_1 + \alpha_2$ ) suggests that one additional vendor leads to a 4% decline in duration, or about 7 days, due to the effects of competition. However this estimate is not statistically significant. Estimates of the effect of quantity ( $\beta_4$ ) suggest that a 10% increase in installed base is associated with 1.5% decline in duration. Since the results of this specification are qualitatively similar to our earlier estimates, we conclude that possible mis-measurement of RIVALS and NON-RIVALS does not bias our estimates of  $\beta_1$ ,  $\beta_2$  and  $\beta_3$ .

#### **6.4 Spillovers**

One potential alternative interpretation of our results is that they reflect spillovers across vendors on how to fix vulnerabilities: The greater the number of vendors affected by a vulnerability, the greater the knowledge spillovers, and the lower the patch release time. Note that the spillover hypothesis does not predict that the effects of rivals and non-rivals would be greater under non-instant than instant disclosure, as we find above. In other words, the presence of spillovers cannot be used to explain our disclosure results. To demonstrate that the presence of spillovers is unlikely to be the reason for our competition results, we re-estimate equation 5 after excluding products in our sample that are very similar to one another (and for which patches are likely to be most similar). More specifically we re-estimate equation 5 after excluding observations that relate to UNIX and UNIX clones.

Column (5) of Table 5 shows these results.<sup>16</sup> The results suggest that the direct and indirect effects are qualitatively similar to those shown in table 5, column (4). If anything, the point estimates of direct and indirect competition are marginally higher (13% per vendor) without the UNIX related products. Hence we conclude that our results are not driven by spillovers between vendors for a vulnerability.

## **7. Discussion and Conclusion**

We study how direct competition, indirect competition, disclosure, and market size influence decisions by software vendors to invest in the patching of software vulnerabilities, which remains key to cyber security. Our baseline results indicate that one additional non-rival has a similar impact on patching times as an additional rival – a reduction of 8-10 days. In addition, each additional rival or non-rival increases the likelihood of the vulnerability being disclosed, resulting in a reduction in patching time by about 7 days. Further, our results show that vulnerabilities for software products in larger markets are associated with faster patch release times: a 10% increase in the installed base leads to a 1.5% decline in patching times, or about 3 days. Our results fully support all of our main hypotheses.

### **7.1 Limitations**

As with any empirical work, our conclusions are limited by our data. First, we are able only able to identify how one facet of competition influences patch release times: competition from vendors who are also affected by the same vulnerability. We are not able to separately identify the overall impact of competition on software quality. As a result, our results identify a lower bound for the effects of competition. Further, though we have shown that increases in rivals and non-rivals will reduce patching times due to the effects of disclosure, we are unable to identify whether the disclosure effect works through actual early disclosure or through the threat of early disclosure. That is, we are unable to identify whether vendors increase their investments

---

<sup>16</sup> Observations relating to the following operating systems were removed – SUSE linux, SCO Linux, SCO UNIX, IBM AIX, Redhat linux, Sun Solaris, HP-UX, Apple (BSD based operating system only) and Digital Unix.

in software patching in response to actual disclosure of a vulnerability, or increase their investments in response to expected early disclosure. However, these limitations do not influence the primary findings of this paper: that increases in the number of vendors affected by vulnerabilities influence ex post quality provision through the effects of competition and disclosure.

Each of our models required differing identification assumptions regarding the measurement of rivals and non-rivals and their relationship with vendor patch release times. However, by estimating a variety of different models that provide very similar estimates we are able to improve the confidence in our results. Moreover, we explore the robustness of our estimates in a variety of ways, which indicates that various possible sources of bias are not significant, and alternative explanations unlikely.

## **7.2 Implications for research**

Our research provides direct evidence on a question of considerable importance to academics, managers, and policymakers: the relationship between competition and software security.

We demonstrated that the threat of early disclosure has an economically significant effect on vendors patching behavior. Thus this research in part supports the results of theoretical models that argue that threat of early disclosure affects the timing of patch release for vulnerabilities.

We also advance prior empirical research on the relationship between competition and quality. We showed that quality provision is influenced not only by the number of rivals competing with the firm, but also by the number of non-rivals that were affected by the same vulnerability. This suggests that future research on the determinants of quality provision in software markets should focus in particular on the effect of changes in the number of firms that share common code bases. More broadly, we show that in software markets, vendors in one market can influence strategic behavior in another market, due to shared code. We believe that this phenomenon is more important than is widely appreciated. Recent trends in programming

such as object-oriented programming and open source have emphasized software reuse. Research on software engineering has long recognized the promises and challenges of software reuse in the design and development of software. However, there has been relatively little research on how this practice influences strategic decision-making in firms. Shared code bases have the potential to influence product market strategies in unrelated markets through mechanisms other than the ones we consider: for example, other dimensions of software quality and exposure to intellectual property litigation. For example, in March 1993 SCO Group filed a well known lawsuit against IBM for allegedly contributing proprietary SCO code to open-source Linux. This lawsuit had implications for Linux users and software developers across a variety of industries (e.g., Foley 2003). In short, more research is needed to understand the implications of shared code base for the strategic behavior of software firms.

### **7.3 Implications for managers and policymakers**

Understanding the relationship between competition and software quality is important for managers and policymakers. For managers of software buyers, it informs the decision when to invest in temporary countermeasures in anticipation of a patch. Understanding this relationship may also shape software purchase decisions: other things equal, changes in quality provision induced by competition may influence vendor or product choice. For producers of software, an understanding of this relationship will provide clues to competitor behavior. In particular, our results provide a better understanding of how disclosure threat influences vendor's response to vulnerabilities.

These findings also have implications for how vendors build their products. Vendors who use code shared by rivals and non-rivals should be aware of the future implications of the competition and disclosure effects for investments in ex post quality provision. With increasing mergers amongst software vendors and code reuse, the vendors have to appropriately alter their software testing regime. Recent data from Symantec shows that close to a third of the vulnerabilities come from shared code (Higaki, 2008). While prior research on software

engineering economics has attempted to measure how software reuse influences development costs (e.g., Banker and Kauffman 1991; Poulin et al 1993), our research shows that reuse will increase interdependence across firms in security countermeasures.

Last, these results have implications for the debate of how to improve software quality. Given the rapid increase in the number of reported software vulnerabilities and the consequent economic damages to end users, the factors that contribute to the timing of vendors' patch release has been a matter of great interest among members of the software community. Many members of the security community have recommended regulation aimed at providing incentives for software vendors to minimize the time window of exposure to end users. However, the optimal regulation to minimize social losses from vulnerabilities critically depends upon a proper understanding of factors that condition the timing of patch release to vulnerabilities. Our research demonstrates that despite high levels of concentration in many software markets, indirect competition and threat of disclosure from vendors in complementary markets works to reduce patching times almost as much as increases in the number of competitors.

Finally, for policy makers this work underscores the importance of disclosure threat as a valid policy instrument. In particular, our work demonstrates that disclosure threat can be effectively used to influence the timing of patch release for a vulnerability. Our results suggest that non-instant disclosure could be more welfare-enhancing than instant disclosure, and that for policy markets like CERT/CC any disclosure policy should include judicious use of disclosure threat to elicit faster vendor responses to vulnerabilities.

## References

- Agarwal, M. and K. Chari (2007), "Software effort, quality, and cycle time: A study of CMM level 5 projects," *IEEE Transactions on Software Engineering*, (33), pp. 145–156.
- Arbaugh, W. A., Fithen, W.L., & McHugh, J. (2000), "Windows of vulnerability: A case study analysis," *IEEE Computer*, (33:12), pp. 52-59.
- Arora, A., Caulkins J. & Telang R. (2006), "Sell First, Fix Later: Impact of Patching on Software Quality," *Management Science* (52:3), pp. 465-471.
- Arora, A., Krishnan R., Telang R. & Yang Y. (2006), "An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure," *Information Systems Research* (forthcoming).
- Arora, A., Nandkumar A. & Telang R. (2006), "Impact of patches and software vulnerability information on frequency of security attacks - An empirical analysis," *Information Systems Frontiers* (8:5), pp. 350-362.
- Arora, A., Telang R. & Xu H. (2008), "Optimal Policy for Software Vulnerability Disclosure," *Management Science*, 54(4), pp. 642-656
- Baltagi, B. H. (1995), "Econometric Analysis of Panel Data," Wiley, New York.
- Banker, R. & Kauffman R. (1991), "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Quarterly*, (15:3), pp. 375-401.
- Banker, R., Davis G. & Slaughter S. (1998), "Software Development Practices, Software Complexity, and Maintenance Performance," *Management Science*, (44:4), pp.433-450.
- Beattie, S., Arnold, S., Cowan, C., Wagle, C., Wright, C., and Shostack, A. (2002), "Timing the Application of Security Patches for Optimal Uptime," *Proceedings of LISA XVI* (2002).
- Borenstein, S. & Netz J. (1999), "Why do All the Flights Leave at 8 am?: Competition and Departure-Time Differentiation in airline markets," *International Journal of Industrial Organization*, (17:5), pp.611-640.
- Bresnahan, T. & Greenstein, S. (1999), "Technological Competition and the Structure of the Computer Industry," *Journal of Industrial Economics*, (47:1), pp. 1-40.
- Bresnahan, T. & Yin P. (2006), "Economic and Technical Drivers of Technology Choice: Browsers," Working Paper, Harvard Business School, Harvard University.
- Brown, A. & Booch G. (2002), "Reusing Open Source Software and Practices: The Impact of Open-Source on Commercial Vendors," in *Software Reuse: Methods, Techniques, and Tools: 7th International Conference, ICSR-7 Proceedings*, (ed.) C. Gacek, p. 123-136.

- Cavusoglu, A., Cavusoglu H. & Raghunathan S. (2005), "Recent Issues in Responsible Vulnerability Disclosure," Workshop on Economics and Information Security (WEIS), Boston, MA, June.
- Choi, J.P., Fershtman C. & Gandal N. (2005), "Internet Security, Vulnerability Disclosure, and Software Provision," Workshop on Economics of Information Security (WEIS05), Kennedy School of Government, Harvard University, 2005.
- Cohen, A. & Mazzeo M. (2004), "Competition, Product Differentiation and Quality Provision: An Empirical Equilibrium Analysis of Bank Branching Decisions," Finance and Economics Discussion Series 2004-46. Washington: Board of Governors of Federal Reserve System, 2004.
- Domberger, S. & Sherr A. (1989), "The impact of competition on pricing and Quality of Legal Services," *International Review of Law and Economics*, (9), pp. 41-56.
- Dranove, D. & White W. (1994), "Recent Theory and Evidence on Competition in Hospital Markets," *Journal of Economics and Management Strategy* (3:1), pp.169-209.
- Foley, J. (2003), "You May Be Next," Information Week, November 23, available at <http://www.informationweek.com/shared/printableArticle.jhtml?articleID=16400348>.
- Forman, C., Goldfarb A., & Greenstein S. (2005), "How did location affect adoption of the commercial Internet? Global village vs. urban leadership," *Journal of Urban Economics*, (58), pp.389-420.
- Gal-Or, E. (1983), "Quality and quantity competition," *The Bell Journal of Economics*, (14:2), pp. 590-600.
- Il-Horn Hann, Hui Kai-Lung, Tom Lee Sang-Yong, P.L. Ivan (2007), "Overcoming Online Information Privacy Concerns: An Information-Processing Theory Approach," *Journal of Management Information Systems*, (24:2), pp. 13-42.
- Harter, D.E., Krishnan M.S., & Slaughter S. (2000), "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science* (46:4), pp.451-466.
- Higaki Wesley (2008), "What Are Vendors Doing To Make Software Secure?," Cylab Seminar, <http://www.cylab.cmu.edu/default.aspx?id=2434>.
- Honeynet Project (2004). "Know Your Enemy: Trends," available at <http://www.honeynet.org/papers/trends/life-linux.pdf>
- Hoxby, C. (2000), "Does Competition among Public Schools benefit Students or Taxpayers?" *American Economic Review*, (90:5), pp. 1209-1238.
- Kannan K. & Telang R. (2005), "Market For Software vulnerabilities? Think Again," *Management Science*, 51(5), pp. 726-740.
- Kretschmer, T. (2005), "Competing Technologies in the Database Management Systems Market," NET Institute Working Paper #05-17.

- Levhari, D. & Peles Y. (1973), "Market Structure, Quality and Durability," *The Bell Journal of Economics and Management Science*, (4:1), pp. 235-248.
- Li P. and Rao, H.R. (2007), "An examination of private intermediaries' roles in software vulnerabilities disclosure," *Information Systems Frontiers*, (9), pp. 531-539.
- Loch, K. D., Carr, H. H., and Warkentin, M. E. (1992), "Today's reality, yesterday's understanding," *MIS Quarterly*, (17: 2), pp. 173-186.
- Mazzeo, M. (2003), "Competition and Service Quality in the U.S. Airline Industry," *Review of Industrial Organization*, (22), pp. 275-296.
- National Institute of Standards and Technology, (2002), "The Economic Impacts of Inadequate Infrastructure for Software Testing," NIST Planning Report 02-03.
- Nizovtsev, D.T. & Thursby M. (2007), "To Disclose or Not? An Analysis of Software User Behavior," *Information Economics and Policy*, (19:1), pp. 43-64.
- Poulin, J.S., Caruso J.M., & Hancock D.R. (1993), "The business case for software reuse," *IBM Systems Journal* (32:4), pp. 567-594.
- Schmalensee, R. (1979), "Market Structure, durability, and Quality: A Selective Survey," *Economic Inquiry*, (17), pp. 177-196.
- Schneier, B. (2000), "Full Disclosure and the Window of Exposure," in: CRYPTO-GRAM, 2000.
- Spence, A.M. (1975), "Monopoly, Quality and Regulation," *The Bell Journal of Economics*, (6:2), pp. 417-429.
- Straub, D. W. (1990), "Effective IS security: An empirical study," *Information Systems Research*, (1:3), 255-276.
- Straub, D. W. and Welke, R. J. (1998), "Coping with systems risk: Security planning models for management decision making," *MIS Quarterly*, (23:4) pp. 441-469.
- Straub, D., Goodman, S., and Baskerville, R. 2008. "Framing of Information Security Policies and Practices," in Information Security Policies, Processes, and Practices, D. Straub, S. Goodman and R. Baskerville (eds.), Armonk, NY: M. E. Sharpe.
- Swan, P.L. (1970), "Durability of Consumer Goods," *American Economic Review*, (60), pp. 884-894.
- Symantec (2004), "Symantec Internet Security Threat Report 2004" available at [http://eval.symantec.com/mktginfo/enterprise/white\\_papers/ent-whitepaper\\_symantec\\_internet\\_security\\_threat\\_report\\_vi.pdf](http://eval.symantec.com/mktginfo/enterprise/white_papers/ent-whitepaper_symantec_internet_security_threat_report_vi.pdf)
- Telang, R. & Wattal S. (2007), "Impact of Software Vulnerability Announcements on the Market Value of Software Vendors – an Empirical Investigation," *IEEE Transactions on Software Engineering*, (33:8), pp. 544-557.

Tucker, C. and A. Miller (2008), "Privacy Protection and Technology Diffusion: The case of Electronic Medical Records," Working Paper, MIT Sloan School of Management.

West, J. & Dedrick J., (2000), "Innovation and Control in Standards Architectures: The Rise and Fall of Japan's PC-98," *Information Systems Research*, (11:2) pp. 197-216.

Wooldridge, J. (2002), *Econometric Analysis of Cross Section and Panel Data*, MIT Press.

**Table 1: Variable Descriptions**

<i>Variable</i>	<i>Description</i>
DURATION	Time taken by vendors to issue a patch for a vulnerability
LOGDURATION	Log of DURATION
VENDOR	Total number of vulnerable vendors affected
RIVAL	Number of vulnerable sellers in the same market.
NONRIVAL	Number of vulnerable sellers in other markets
INSTANT	1 if Instant disclosure, 0 otherwise
NONINSTANT	1 if Non-instant disclosure, 0 otherwise
LOGQUANTITY	log(1+total # of employees at customer sites (sites that use the software)).
LOGVERSIONS	log of number of versions
LOGSEVERITY	log(1+ CERT severity metric)

**Table 2: Descriptive statistics**

<i>Variable</i>	<i>Mean</i>	<i>Min.</i>	<i>Max.</i>	<i>Std. Dev</i>
<i>Full Sample N=461</i>				
DURATION (Days)	168	1	3904	558
LOGDURATION	3.52	0.69	8.27	1.92
VENDORS	9.02	1	1.37	8.04
RIVALS	5.96	0	19	5.87
NONRIVALS	3.03	0	24	3.65
LOGQUANTITY	13.95	6.22	17.41	2.26
LOGVERSIONS	0.22	0	3.14	1.63
LOGSEVERITY	2.73	0	4.69	20.34

**Table 3: Comparison of conditional mean of LOGDURATION**

RIVALS	HIGH	LOW	COMPETITION EFFECT
LOGDURATION	2.81 <sup>***</sup> (0.21)	3.73 <sup>***</sup> (0.12)	-0.92 <sup>***</sup> (0.24)

Notes: Cells contain mean of LOGDURATION conditional on RIVALS. Standard errors in parentheses. Sample median of RIVALS = 5.96. \* Significant at 90% confidence level. \*\* Significant at 95% confidence level. \*\*\* Significance at 99% confidence level.

**Table 4: Estimates of Random Effects GLS Regressions of Equation 2 – Dependent Variable LOGDURATION**

Variable	<i>OS Sample</i>	<i>Full Sample Without Market Fixed Effects</i>	<i>Full Sample with Market Fixed Effects</i>
	(1)	(2)	(3)
RIVALS ( $\beta_1$ )	-0.09 <sup>***</sup> (0.03)	-0.09 <sup>***</sup> (0.03)	-0.07 <sup>***</sup> (0.03)
LOGQUANTITY ( $\beta_4$ )	-0.03 (0.17)	-0.14 <sup>**</sup> (0.06)	-0.13 <sup>*</sup> (0.08)
HARDWARE	-2.75 (3.74)	-3.90 <sup>**</sup> (1.66)	-4.20 <sup>***</sup> (1.69)
HARDWARE*LOGQUANTITY	0.21 (0.25)	0.29 <sup>**</sup> (0.11)	0.24 (0.31)
LOGVERSIONS	0.18 (0.21)	0.22 (0.17)	0.24 (0.17)
LOGSEVERITY	-0.14 (0.17)	-0.13 (0.13)	-0.14 (0.13)
Constant	6.15 <sup>***</sup> (0.88)	6.52 <sup>***</sup> (1.01)	7.07 <sup>***</sup> (1.08)
N	366	461	461
R <sup>2</sup> (between)	0.12	0.11	0.13
R <sup>2</sup> (Within)	0.06	0.06	0.06
R <sup>2</sup> (Overall)	0.12	0.14	0.15
#vulnerabilities	159	241	241
Market fixed effects	0	0	3
Vendor Fixed effects	7 <sup>b</sup>	8	8
$\sigma_u$	1.71	1.73	1.71

Notes: Standard errors in parenthesis. \* Significant at 90% level. \*\* Significant at 95% level. \*\*\* Significance at 99% level. <sup>b</sup>Oracle vendor fixed effect cannot be estimated as it is not an OS vendor.

**Table 5: Estimates of Direct and Indirect Competition and Disclosure Effect – Dependent Variable LOGDURATION**

Variable	Equation 4 GLS (1)	Equation 4 Tobit (2)	Equation 4 IV (3)	Equation 5 GLS (4)	Equation 5 GLS (5)
INSTANT	-0.00 (0.37)	0.02 (0.37)	-0.05 (0.92)	-0.08 (0.35)	0.06 (0.44)
RIVALS ( $\gamma_1 = \beta_1 + \beta_3$ )	-0.09*** (0.03)	-0.09*** (0.04)	-0.08* (0.04)		
NON-RIVALS ( $\gamma_2 = \beta_2 + \beta_3$ )	-0.10*** (0.03)	-0.10** (0.04)	-0.09* (0.05)		
INSTANT*RIVALS ( $\gamma_3 = -\beta_3$ )	0.04** (0.02)	0.04** (0.02)	0.04 (0.05)		
LOGQUANTITY ( $\beta_4$ )	-0.15** (0.08)	-0.19*** (0.08)	-0.17** (0.08)	-0.15** (0.08)	-0.15* (0.09)
HARDWARE	-3.77*** (1.57)	-4.58*** (1.74)	-4.23*** (1.74)	-3.38* (1.98)	-2.03 (2.62)
HARDWARE*LOGQUANTITY	0.28*** (0.11)	0.34*** (0.12)	0.31*** (0.12)	0.26** (0.13)	-0.01 (0.25)
LOGVERSIONS	0.26 (0.16)	0.31* (0.18)	0.40** (0.19)	0.26 (0.18)	0.45* (0.25)
LOGSEVERITY	-0.14 (0.14)	-0.15 (0.14)	-0.17 (0.13)	-0.13 (0.13)	-0.19 (0.17)
VENDORS ( $\alpha_1 = \beta_1 + \beta_3$ )				-0.10*** (0.03)	-0.13** (0.03)
INSTANT*VENDORS ( $\alpha_2 = -\beta_3$ )				0.06** (0.03)	0.03 (0.03)
Constant	7.13*** (1.08)	7.85*** (1.15)	7.57*** (1.20)	7.10*** (1.14)	7.37*** (1.18)
Implied estimate of $\beta_1$	-0.05* (0.03)	-0.05 (0.03)	-0.04 (0.04)	-0.04 (0.04)	-0.10** (0.04)
Implied estimate of $\beta_2$	-0.06* (0.03)	-0.06* (0.03)	-0.05 (0.04)	-0.04 (0.04)	-0.10** (0.04)
N	461	461	461	461	195
R <sup>2</sup> (between)	0.14	-	0.14	0.15	0.18
R <sup>2</sup> (Within)	0.08		0.07	0.07	0.12
R <sup>2</sup> (Overall)	0.15		0.15	0.17	0.20
# vulnerabilities	241	241	241	241	158
Log Likelihood	-	-902.02	-	-	-
$\sigma_u$	1.73	1.76	1.69	1.72	1.86

Notes: Standard errors in parenthesis. \* Sig. at 90% level. \*\* Sig. at 95% level. \*\*\*Sig. at 99% level. In columns (1) through (4) estimates include 8 vendor fixed effects and 3 market fixed effects. In column (5) Redhat and SCO vendor fixed effects cannot be estimated as both have now been removed from the sample.