Competition and Quality Restoration: An Empirical Analysis of Vendor Response to Software Vulnerabilities

Ashish Arora, Chris Forman, Anand Nandkumar[1] and Rahul Telang

{ashish, cforman, anandn, rtelang}@andrew.cmu.edu

Carnegie Mellon University

First Draft: October 2005

This version: June 2006

Abstract:

We examine the effect of competition on one aspect of software quality: time taken by software vendors to release patches to software vulnerabilities. We distinguish between two effects. The first is the direct competition effect: vendors with more competitors have more to lose from tardy patches. However, even vendors that do not compete in the product market but whose products share a software vulnerability may nonetheless compete indirectly: They implicitly increase the threat of disclosure for each other. Our results demonstrate that a one unit increase in the number of competitors lowers expected patching times between 4% and 10%. We further demonstrate that an increase in the number of vendors sharing common software components also lowers patching times: A one unit increase in the number of such vendors lowers expected patching times between 4% and 5% days on an average. Further, firms with larger product sales patch faster: a 10% increase in installed base is associated with an earlier patch release by about 1.4%. Our results support the notion that increased competition, directly and indirectly, leads to faster patching times and improved consumer welfare.

Keywords: Vulnerability disclosure, quality, competition, patching.

---

[1] Corresponding author. Author names are in alphabetical order. We thank the Software Industry Center at Carnegie Mellon University for financial support. We thank Avi Golfarb and seminar participants at Carnegie Mellon University, the International Industrial Organization Conference, 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. Rahul Telang acknowledges generous support of National Science Foundation through the CAREER award CNS-0546009. All errors are our own.

1. Introduction

Costs related to information security have recently had a large and increasing impact on the U.S. economy. A recent study put the annual cost of major software bugs to the U.S. economy at over $60 billion (NIST 2002). Though there are not as yet any official U.S. government statistics on information security, several private groups have demonstrated the growth in security-related incidents and their antecedents. The number of information security incidents reported to CERT/CC, a large federally funded research laboratory that measures and researches Internet security problems, grew from 2412 in 1995 to 137,529 in 2003.[2] Meanwhile, the number of reported software security defects or "vulnerabilities", one leading indicator of security incidents, grew from 171 in 1995 to 5990 in 2005.

The rapid increase in the number of vulnerabilities discovered in software over the past several years has led many to argue that high levels of concentration and significant early mover advantages in software markets lead to an under-provision of security. Since these vulnerabilities are due to defects in software, this is part of a more general issue of market structure and software quality, namely that firms with market power deliberately under-provide quality in an effort to maximize profits. However, others have argued that the link between market structure and provision of software quality has been exaggerated. For one, some studies have found that users are unwilling to pay for software quality because it is difficult for them to value it ex ante: If under provisioning of quality is due to lack of user willingness-to-pay then market structure may have little impact on quality. Moreover, incentives to provide quality will sometimes be influenced by vendors in related markets. Software products sold in different markets often share components. This implies that they will sometimes be affected by the same vulnerability as well. This implies that a firm's patching decisions will not only be affected by competition in its own market, but also by vendor behavior in technologically related markets as well.

In this paper, we examine the relationship between competition and one dimension of software quality: the time taken by the vendor to release a patch for a known vulnerability. We begin by developing our hypotheses in a model of vendor patching behavior. Vendors make investments to lower patching times based on the extent to which they internalize end user losses. End user losses are increasing in the time elapsed between the initial disclosure of the vulnerability and the release of the patch. Increases in market competition increase vendor losses from unpatched vulnerabilities since competition increases the likelihood that vulnerabilities will influence customers to switch vendors. As a result, increases in market competition will lead to lower patching times. We label this direct impact of competition on patching times as the *competition effect.* Further, increases in the total number of vendors—both competitors and noncompetitors—that share the vulnerability will lead (in probability) to earlier initial public disclosure of

[2] CERT/CC statistics are available at http://www.cert.org/stats/cert_stats.html.

the vulnerability. Since end user losses from unpatched software are increasing in the time the vulnerability remains unpatched and disclosed, increases in total vendors that share a vulnerability will lead to lower patching times. We label this relationship the *disclosure effect.*

We formally derive this and other auxiliary hypotheses. We compare these hypotheses with actual data on vendor patching times. To empirically separate the effects of competition and disclosure on vendor patching times, we exploit two sources of variation in our data. First, we utilize the variation across vulnerabilities in the number of rivals and nonrivals affected. Increases in the number of direct rivals to the vendor will influence patching times through both the competition and disclosure effects, while increases in nonrivals will influence patching times only through disclosure. Second, we utilize variation across vulnerabilities in how vendors are informed of vulnerabilities. Vulnerabilities are *publicly disclosed* when a third party or another vendor announces the existence of a vulnerability, and they are *privately disclosed* when CERT/CC informs the vendor of the presence of a vulnerability while the vulnerability remains unknown to the general public. We identify the competition effect by examining how changes in the number of affected vendors influence average patching times when vulnerabilities are publicly disclosed. We identify the disclosure effect by comparing how changes in the number of affected vendors influence average patching times under private and public disclosure.

Addressing our research goals requires detailed data on software vulnerabilities, vendor patching times, market structure, and software characteristics. We examine vendor responses to 241 vulnerabilities reported to CERT/CC from September 2000 to August 2003. These data are among the most complete of their kind that are available. We supplement these data with information on market size obtained using a market survey conducted by Harte Hanks Market Intelligence, a commercial market research firm.

Our results demonstrate that competition and disclosure each have an economically and statistically significant impact on patching times. We show that a one unit increase in the number of rivals lowers expected patching times by between 4% and 10%; this translates to an average decline of 7 to 17 days due to the direct effects of competition. Disclosure also plays a role: a one unit increase in vendors from unrelated markets (non-rivals) that share the same vulnerability will lead to a decrease in expected patching times between 4% and 5%, or a decline of 7 to 8 days. We infer that changes in structure to own and technologically related markets induce changes in quality provision. Last, we also show that changes in market size will also increase quality provision: a 10% increase in vendor installed base will lead to a 1.3% to 1.4% decline in patching times.

Our research is unique in demonstrating how products with common technological inputs can influence output market competition even when buyers perceive these markets as unrelated. Recent work on information technology markets has emphasized strategic interactions among vendors producing products

that are complements in demand or which share a common platform (e.g., Bresnahan and Greenstein 1999; Bresnahan and Yin 2006; Gawer and Henderson 2005). Like this prior literature, we emphasize how firms that share common components have interrelated output market decisions. However, in contrast to this prior work, we do not require these firms to produce in markets that are substitutes or complements in demand.

Our findings also inform the debates on how to best encourage provision of software quality. For one, our research demonstrates that despite high levels of concentration in many software markets, the threat of disclosure from vendors in complementary markets works to reduce patching times almost as much as increases in the number of direct competitors. Further, our research informs recent debates on whether third-party information security agencies such as CERT/CC should inform the public of new vulnerabilities, or whether they should instead disclose them only to affected vendors. Our results show that the threat of potential disclosure provides powerful incentives for vendors to invest in software patching.

2. Related Literature

This paper is related to three streams of research: competition and quality provision; vertical and horizontal differentiation strategies in markets undergoing rapid technological change; as well as the economics of information security.

Competition and Quality Provision. 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.[3] In general, prior work has demonstrated that increases in competition leads to better quality provision. Demberger and Sherr (1989) provide evidence that deregulation in the legal services industry leads to greater customer satisfaction. Dranove and White's (1994) literature survey suggests that higher market concentration leads to lower quality in hospital markets. Borenstein and Netz (1999) note that airlines were less likely to schedule their flights at passengers' most preferred times during the period of price regulation. Hoxby (2000) finds that metropolitan areas with more schools districts produce higher quality measures in terms of student achievements. Mazzeo (2003) provides evidence of longer flight delays in more concentrated airline markets, while Cohen and Mazzeo (2004) find evidence of higher quality when banks face multi-market bank competitors.

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. Bresnahan and Greenstein 1999

[3] 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).

argue that changes in industry structure in related markets can have long run implications for product market competition. In our research we argue that changes in structure to markets that share common inputs will have important implications for vendors' quality decisions. Such changes are likely to be particularly salient in software markets, where vendors in different market segments increasingly share common modules.

Product differentiation in technology markets. Our research also builds upon recent empirical work that has studied quality provision and horizontal and vertical differentiation strategies in markets undergoing rapid technological change. Bresnahan, Stern, and Trajtenberg (1997) study two dimensions of product differentiation among personal computer manufacturers: branding and location on the technological frontier. Greenstein (2000) examines geographic variation in the provision of quality among U.S. Internet Service Providers, highlighting in particular whether these firms offer services such as broadband service, hosting, or web design. Greenstein and Markovich (2006) study business models and the determinants of market value among vendors providing Internet hosting services. While these studies are similar to ours in their study of quality provision in information technology markets, they do not explicitly examine the link between market structure and quality.

Economics of Information Security. One active area of theory research in information security has studied the economic impacts of vulnerability disclosure and the optimal timing of disclosure for society. Schneier (2000) argues that losses from attacks are not only influenced by the intensity of attacks, but also on how long the vulnerability remains unpatched. Arora, Telang and Xu (2004) show that early disclosure of vulnerabilities is not necessarily socially optimal, though it will engender earlier releases of patches. Cavusaglu et al (2005) use a multi-vendor model to examine the socially optimal disclosure of vulnerabilities. Nizovtsev and Thursby (2005) examine the factors that influence a benign identifiers' decision to disclose vulnerabilities to CERT instead of disclosing them publicly. Choi, Fershtman and Gandal (2005) examine how vulnerabilities affect vendor and consumer behavior in the software market. They conclude that vendors are likely to disclose vulnerabilities when the probability of an attacker exploiting a vulnerability is relatively high. Further, they note that vendors may disclose vulnerabilities even when it is socially suboptimal.

Empirical work examining vulnerability disclosure is rarer. Arora, Nandkumar and Telang (2004) provide empirical evidence on the impact of vulnerability publication when disclosure is not accompanied by patches. They find that undisclosed vulnerabilities attract the least number of attempts to breach a host, while vulnerabilities that are disclosed without a patch attract the most number of attempts to breach a host. To the extent that such breaches are correlated with monetary losses, early disclosure could result in substantial economic losses. Arora, Krishnan, Telang and Yang (2005) use a dataset assembled from

CERT/CC's vulnerability notes and SecurityFocus database to show that early disclosure leads to faster patching times. Telang and Wattal (2004) use an event study methodology to show that vulnerability disclosure leads to a loss of market value. Our research is similar to prior work in that we examine the economic outcomes from vulnerability disclosure. However, to our knowledge, no prior work has studied how competition influences vendors' strategic response to vulnerability disclosure.

3.1 Software vulnerabilities and patches

Unlike many physical goods, the problems related to software can be mitigated even after product release. Vendors try to introduce the product relatively early in the product development cycle even though early release might entail greater investments in ex post support (Arora, Caulkins and Telang 2005). This makes both vulnerabilities in software as well as patches that fix vulnerabilities intrinsic to any "shrink wrapped" software. The probability of a malicious attacker exploiting a specific vulnerability to compromise end user computers is positively correlated with the amount of time the vulnerabilities remain 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 the vendor's patch: (1) the extent to which of end user losses affect the future demand for the product and (2) the cost of fixing the vulnerability. Typically, an early fix entails higher costs but also reduce customer losses and, hence also, reduce loss of future sales.

In many cases, a newly discovered vulnerability could affect many different products (for future reference we label these *common vulnerabilities*). For instance, a stack buffer overflow vulnerability in Sendmail (a commonly used mail transfer agent), 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, Wirex. Some of these products potentially compete with the while others are in very distinct markets.[4] For instance, Wirex and Mirapoint produced 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. However, all these products use Sendmail code, and hence, were affected by the vulnerability in it.

A common vulnerability is typically an artifact of a shared code base or design specification or due to a proprietary extension of a widely used software component. When a vulnerability is known to be common to many products, if one vendor releases a patch for its product it effectively publicly discloses

[4] Given the number VU#897604 by CERT, See http://www.kb.cert.org/vuls/id/897604 (accessed 22 Sept, 2006.)

the vulnerability in the rivals' products as well. As public disclosure of the vulnerability provides information to attackers, the end user losses of the rivals are higher after disclosure. In short, increases in the number of vendors sharing a vulnerability potentially leads to earlier disclosure and greater end user losses, other things equal. We label the relationship between such increases and patching times as the *disclosure* effect; all else equal, the disclosure effect should lead to shorter patching times.

Increases in the number of direct competitors will also decrease patching times. As noted above, the literature on product quality and competition suggests that when there are many competing products, end users have more choices, and thus, future sales of a product are likely to be more sensitive to perceived quality. In our context, this implies that end users can compare vendor responses and penalize laggards. Thus a greater number of competitors is also expected to reduce expected patching times. We label the impact of increased direct competition on patching times as the *competition effect.* In the paper, we identify these effects separately and show how competition and disclosure threat influences vendors' time to patch vulnerabilities.

4. Model

We develop a simple model of firm investment in one dimension of product quality, the time to release of patches for software vulnerabilities. We use this model to develop empirically relevant hypotheses. This model builds upon prior work by Arora, Telang, and Xu (2004). However, in contrast to this prior work, we examine how patching behavior is influenced by market competition, size, and disclosure threat. For purposes of illustration, we consider the case in which there are two vulnerable vendors, however proofs are presented for the general case of multiple vulnerable vendors. All proofs are in Appendix 1.

4.1 Model Set-up

There are 5 main players is the model: vendor *i*, vendor *j,* (the other vendor that shares the vulnerability with vendor *i*), an intermediary like CERT (which provides a *protected period* T to the vendors to come up with a patch), the attacker and the end user. We refer to vendor *j* as "other vendor" which is also affected by same vulnerability and the set of all vendors affected the same vulnerability as "vulnerable vendors". Vendor *j* is affected by the same vulnerability as vendor *i* but may or may not be competing in the same market. End users suffer losses from vulnerabilities due to exploits from attackers. Intermediaries inform vulnerable vendors about the presence of a common vulnerability and provides them with a protected period.

The scenario considered by our model is as follows: A vulnerability that affects multiple products is first discovered by an identifier, who informs the intermediary. The intermediary then informs all vulnerable vendors and gives them a protected period by keeping the vulnerability secret until all the vulnerable

vendors fix the vulnerability. The vendor and other vulnerable vendor(s) then commit to a one-time decision on when they will release a patch for the vulnerability[5]. We assume that patches are homogenous and completely remove vulnerabilities. Thus, vendors choose only when to patch and not the quality of the patch that they release.

Some of the other assumptions of the model are as follows: First, we model disclosure as a binary outcome: Either all of the details of a vulnerability are disclosed or none at all. Second, we assume that when a vulnerability is patched for one vendor it is immediately disclosed for the other vendor. Third, end users are exposed to malicious attacks until the patches are released by the vendor. Stated otherwise, customers do not take action to remove the threat from vulnerabilities independent of the patches issued by vendors. Fourth, for simplicity, we assume that upon patch release end users immediately install patches. We also do not model the end user costs to install patches.

4.2 Vendor Objective Function

In this subsection, we develop the objective function for vendor *i*; the objective function for vendor *j* is symmetric. For simplicity, we begin with the case in which patching times are deterministic: Vendors can specify exactly the time it takes to develop a patch. We consider the stochastic case in the next subsection. Vendors use two pieces of information to decide the optimal time to release patch – end user loss and patch development cost. End user loss is a function of loss per customer, internalization factor and the number of product installations, or quantity.

$\theta_i(\tau_i, \tau_j, s)$ denotes the cumulative loss incurred by an end user as a result of being exposed to the vulnerability. Third party disclosure takes place at time *s,* which is the time when an attacker or the intermediary makes vulnerability information public before the vendor releases the patch to the vulnerability. If third party disclosure takes place at time *s* and the patch is released at τ_i, then end users are exposed for the duration τ_i - *s*. The end user loss in such a case is denoted by $L(\tau_i - s)$. $L(.)$ is assumed to be increasing and convex in the end user's exposure and L(0) = 0. Since third party disclosure is an uncertain event, we denote the probability of a third party disclosing the vulnerability before τ_i as a distribution function F(.).[6] When vendor *i* is first to release a patch for the vulnerability, end users of the vendor incur losses only upon third party disclosure. Thus the end user loss function is given by

$$\theta_i = \int_0^{\tau_i} L(\tau_i - s) dF(s) ds \qquad (1)$$

[5] The model does not consider a scenario in which the vendor and competitors make real-time adjustments to their pre-committed patch release dates, which may be an extension to the model.

[6] The protected period is embedded in the distribution of *s*. In other words, if the protected period is T, then in the notation of the model, F(T) = 1. This way of modeling CERT economizes on notation.

However, when vendor *i* patches after the other vulnerable vendor, vendor *j*, there are two possible scenarios. First, *third party disclosure* could happen before vendor *j* releases its patch. In this case the end user losses are the same as in equation (1). Second, vendor *j* may release its patch to the vulnerability prior to third party disclosure (before *s*). Since the act of the other vulnerable vendor releasing its patch implicitly constitutes disclosure of the vulnerability to vendor *i,* end user losses are $L(\tau_i - \tau_j)$. The probability that the other vendor releases patch before third party disclosure is given by $1 - \Pr(s \le \tau_j) = 1 - F(\tau_j)$. Hence the end user loss is this case is simply $(1 - F(\tau_j))L(\tau_i - \tau_j)$.

Thus the total expected loss is given by

$$\theta_i = \int_0^{\tau_i} L(\tau_i - s)dF(s) + \left(1 - F(\tau_j)\right)L(\tau_i - \tau_j) \qquad (2)$$

Vendors do not internalize all customer losses. The proportion of losses internalized by a vendor depends on the extent to which end users penalize the vendor for losses. For example, users may decide not to purchase the vendor's product in the future. We consider two factors that influence the fraction of consumer losses internalized by the vendor. First, some fraction of losses will be internalized, independent of the number of competitors in the market. For example, some users may decide simply to stop using the product. We refer to this factor as ω_i. Second, when there are rivals to the vendor, an additional source of internalization may arise. End users may compare responses and penalize the laggards. For example, slow patching times may cause end users to switch vendors. We denote this factor as λ_i in the model. The proportion of end user losses internalized by the vendor is thus $\omega_i + \lambda_i$. Thus if *j* is a rival of vendor *i* the proportion of end user losses internalized vendor *i* is higher than if *j* is not a rival.

The patch development cost depends on the resources allocated by vendors to produce a patch for the vulnerability. Since earlier patches require the vendor to devote more resources, the patch development cost for vendor $i, C(\tau_i)$ is decreasing in τ_i. We further assume that $C(\tau_i)$ is convex in τ_i because the marginal utility of freed resources is decreasing. Hence marginal cost will be increasing with respect to τ_i. Due to these assumptions, $\frac{\partial C(\tau_i)}{\partial \tau_i} < 0$ and $\frac{\partial^2 C(\tau_i)}{\partial \tau_i^{\,2}} > 0$.

Let q_i denotes the number of customers (quantity) of vendor *i*. A higher q_i would imply greater losses for vendors from the vulnerability.

We are now in a position to characterize the vendor's objective function. It is given by

$$V_i = C(\tau_i) + q_i\theta(\lambda_i + \omega_i)_i \qquad (3)$$

The vendor chooses τ_i to minimize V_i given τ_j. Both firms simultaneously choose their patching times.

Since, the strategic case under certainty is not particularly insightful, especially to derive empirically testable hypotheses, we modify the model to incorporate uncertainty (with respect to patch release date) and derive testable hypotheses.

4.3 Strategic interaction under uncertainty:

In practice, it is unlikely that vendors will be able to determine precisely when a patch will be ready for the software. In order to accommodate this reality, we allow for stochastic patching times. Let x_i denote the random variable that denotes the actual patch release date of the vendor. The expected patch date of the vendor is given by $E[x_i] = \tau_i$. We assume that *x* has the distribution function G(x; τ). The amount of resources allotted by vendors towards patching the vulnerability determines the expected patching time. Higher amount of resources are associated with earlier patching times. Vendors choose an optimal τ^*, given the distribution of others vendors' patching times. We look for a symmetric Nash Equilibrium in which vendors simultaneously commit to an expected patch release date τ..

Under uncertainty the vendor's cost function is given by:

$$\tilde{V}_i = \tilde{C}_i + q_i(\lambda_i + \omega_i)\tilde{\theta}_i \qquad (4)$$

The expected cost function under uncertainty is given by

$$\tilde{C}_i \equiv \int_0^R C(x_i) dG(x_i : \tau_i) \qquad (5)$$

The end user loss under uncertainty for vendor *i* is given by

$$\tilde{\theta}_i \equiv \int_0^R \int_0^{x_i} \left(L(x_i - z) d\Phi(z : \tau_j, s, N)\right) dG(x_i : \tau_i) \qquad (6)$$

where $z \equiv \min\{x_1, x_2, ..., x_j, s\}$ and $\Phi(.)$ is the distribution of *z*.

Intuitively, a commitment to early patching by a rival will increase the payoff to the firm from patching quickly as well. This suggests lemma 1 below that the reaction functions are upward sloping i.e., the patch release times are strategic complements (Milgrom and Roberts, 1994?). (All proofs are in the Web Appendix.)

Lemma: *The reaction functions are upward sloping. Or* $\frac{\partial \tau_i^*}{\partial \tau_j} > 0 \ \ \forall j \neq i$.

Theorem 1: There exists a unique Nash equilibrium in pure strategy.

A direct consequence of the upward sloping reaction functions and the symmetric nature of the equilibrium is that when there are more vulnerable vendors, the threat of rival disclosure is greater and as a consequence the vendor's expected loss is higher. This induces the vendor to allocate more resources to patching the vulnerability which results in an earlier patch release.

The model yields three empirically testable propositions.

Result 1: *An increase in the number of vendors that share a common vulnerability increases the disclosure threat and leads to declines in the optimal expected patching time* τ^*.

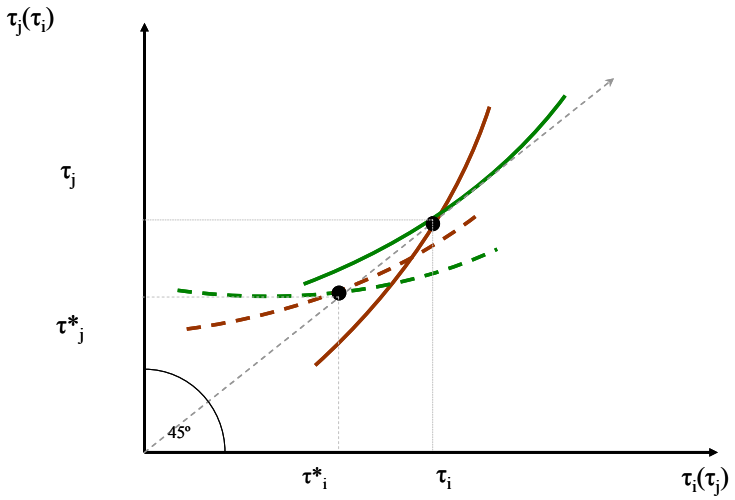


Figure 1: The effect of an increase in disclosure threat

Figure 1 shows the effect of disclosure threat. The solid line represents the upward sloping reaction functions when there are *N-1* vendors. The reaction functions intersect at (τ_i, τ_j) which is the equilibrium at low disclosure threat. An increase in the number of vendors affected from *N-1* to *N* increases disclosure threat makes the reaction functions shift down, and as a result the new equilibrium patch release times are much earlier at (τ_i^*, τ_j^*). Thus the heightened disclosure threats results in an earlier patch release by both vendors affected by the vulnerability.

An increase in λ_i increases the expected losses to vendor *i* and hence results in an earlier patch release by both vendors. Higher λ (which happens when vendor *j* is also a rival) results in a pre-commitment to patch earlier. Given a commitment by one of the vendors to release patch early, all the other vendors that are affected by the vulnerability are also likely to pre-commit to an earlier patch release date. In this paper, we refer to the effect of λ as the effect of competition.

Result 2: *An increase in the number of rivals that share a common vulnerability increases* λ *(in addition to also increasing the disclosure threat) and leads to declines in the optimal expected patching time* τ^*.

Also the expected losses that vendors face is higher when faced with larger market share, and thus, such vendors will patch quicker.

Result 3: τ^* *for a vendor is decreasing in quantity.*

We now use the data on software vulnerabilities to empirically test these results. Before describing our results, we provide a detailed description of our sample and the variables contained therein.

5. Data and variables:

We assembled our data set of vulnerabilities from publications of the CERT Coordination Center (CERT/CC), for the period September 2000 to August 2003. Not all vulnerabilities reported to CERT/CC are eventually published and included in our sample. When a vulnerability is reported to CERT/CC, it researches the vulnerability and contacts the vendor if the vulnerability is authentic and exceeds CERT/CC's minimum threshold value for severity, as measured by the CERT METRIC (which is described later). Vendors then decide whether to respond to CERT/CC's notification of the vulnerability. If the vendor decides to respond to CERT/CC's notification, the typical response takes one of three forms. First, the vendor may acknowledge the presence of the vulnerability. In this case, CERT/CC lists the product's status as "vulnerable". Second, the vendor may report that the product is not vulnerable, in which case CERT/CC lists the vendor's status as "not vulnerable". Third, the vendor may choose not to respond to the vulnerability; in this case, CERT/CC records the vendor's status as "unknown". CERT/CC is widely regarded as the premier source of information on software vulnerabilities and patches.[7] On an average, in a year, about 3000 vulnerabilities get reported to CERT/CC of which only about 10%, those deemed technically or economically significant, are published.

Our unit of observation is a vendor – vulnerability pair. CERT/CC published 526 vulnerability notes over our sample period, with 622 different vendors affected by these vulnerabilities. Each vulnerability note may contain information on multiple vulnerabilities. In all, these vulnerability notes included 4659 observations (vendor-vulnerability pairs). Of these, 762 were listed as "not vulnerable", 2182 were "unknown," while 1714 were listed as "vulnerable". We retained only observations with "status vulnerable" for the purpose of empirical analysis.

We additionally drop observations that would introduce heterogeneity into the sample and obscure our efforts to identify the relationship between market structure and patching times. We dropped observations from non commercial, vendors (such as universities), typically offering open-source products, since these vendors may not adhere to traditional notions of profit maximization. We further dropped vendors that are

[7] Other data sources such as online forums commonly include only information on vulnerabilities that were instantly disclosed. Other sources also do not verify vulnerabilities in the same way that CERT does.

not headquartered in US, as they may compete against other non-US firms for which we have no data, influencing our measures of competition and market share. We also removed protocol vulnerabilities from the data, as patches to these vulnerabilities typically involved protocol changes whose scope extends beyond a particular product. Further, we dropped observations wherein the vendors discovered and disclosed the vulnerability to CERT/CC of its own accord along with a patch. (Our results are robust to inclusion of these observations). This left us with a sample consisting of 241 distinct vulnerabilities and 461 observations. However, the dropped vendors were included on our independent variable that proxies for competition and disclosure threats.

One way that we separately identify the effects of competition and disclosure is by examining how the marginal effects of increasing direct rivals and non-rivals (other vendors with the vulnerability in common) influences patching times under different disclosure regimes. We label vulnerabilities as *instantly disclosed* when the security agency CERT/CC informs the vendor of the presence of the vulnerability even though the existence of the vulnerability had been publicly disclosed (by some third party). We label vulnerabilities as *non-instantly disclosed* when CERT/CC discloses a vulnerability that had previously not been publicly disclosed. In the next two sections, we provide descriptive statistics for dependent and independent variables under both types of disclosure.

5.1 Independent Variables

In this section we discuss the construction of our independent variables. A description of all variables is included in Table 1.

Competition. To determine how threats from competition and disclosure influence patching times, we construct two variables. *VENDORS* is equal to the total number of vendors listed as "vulnerable" by CERT for a specific vulnerability. *RIVALS* is equal to the number of vendors that CERT lists as vulnerable and that operate in the same market as the vendor in the vendor-vulnerability pair. *NONRIVALS* is equal to the number of vendors that are vulnerable but which operate in different markets. We determined rivals and nonrivals using market definitions in the Harte Hanks CI Technology database (see next section). In those cases where the product was obscure and not included in the CI Technology database, we examined product manuals to identify product classification. As an example, suppose the vendor-vulnerability pair in an observation was Microsoft-Windows XP and the vulnerability influenced products produced by Red Hat and Oracle. In this case, total *RIVALS* would include Red Hat but not Oracle, *NONRIVALS* would include Oracle, while total *VENDORS* would include both Red Hat and Oracle. We plot histograms of number of vendors, rivals and nonrivals for vulnerabilities in appendix-I.

Quantity. Data on installations of software was collected using the Harte-Hanks CI Technology Database (hereafter CI database). From 2000-2002, the survey had responses from about 58,094 organizations in the United States. Our data contains the stock of IT hardware and software reported by establishments, with over 100 employees, in December of each year. The CI database is one of the richest sources of data on U.S. business IT investment available. However, establishments in the sample report only binary decisions of software use: details on number of licenses are not reported. To develop a measure of the total installed base of the software, we use the number of establishments that indicated use of the product weighted 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 establishments. This puts more weight on products used in larger organizations, and arguably provides us a more accurate proxy for quantity. Since the CI database oversamples certain industry sectors we follow Forman, Goldfarb, and Greenstein (2005) and weight our data using County Business Patterns data from the U.S. Census. For details on this procedure, see Appendix III. To compute our final measure of quantity, we multiply the number of employees by the establishment weights and sum across establishments. Because the distribution of quantity is highly skewed, we take the log of installed base for our analysis. We label this variable *LOGQUANTITY*.

Other variables. In order to account for differences in severity of vulnerabilities we use the log of (one plus) CERT/CC's severity measure, which is a number between 0 and 180.[8] We label this variable *LOGSEVERITY*. Anecdotal evidence from industry sources suggest 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. In addition, we control for market fixed effects to address unobserved differences across product markets in factors such as intensity of competition, ability of customers to change suppliers and ease of developing patches. Finally, we also include firm dummies for 7 leading vendors, which jointly account for about 74% of the observations in our sample.

Descriptive statistics for all of the independent variables are included in Table 2. Table 2 also shows how these summary statistics vary by disclosure type. Vulnerabilities that are instantly disclosed have overall a

[8] The set of criteria that determines the measure includes whether (i)information about the vulnerability is widely available (ii)the vulnerability being exploited in the incidents reported to US-CERT? (iii) Is the Internet Infrastructure at risk because of this vulnerability? (iv) How many systems on the Internet are at risk from this vulnerability? (v) What is the impact of exploiting the vulnerability? (vi) How easy is it to exploit the vulnerability? www.kb.cert.org/vuls/html/fieldhelp

slightly lower number of vendors than non-instantly disclosed vulnerabilities (7.87 v. 11.65) as well as lower rivals (5.38 v. 7.31) and nonrivals (2.47 v. 4.30).

Table 1: Variable Descriptions

Variable	*Description*
DURATION	Time taken by vendors to patch vulnerabilities
LOGDURATION	Log of DURATION
VENDOR	Total number of vulnerable vendors
RIVAL	The number of vulnerable rivals
NONRIVAL	The number of vulnerable non-rivals
INSTANT	Instant disclosure
NONINSTANT	Non-instant disclosure
LOGQUANTITY	Log(1+ Total # of employees at customers (those that used the software) sites)
LOGVERSIONS	Log of number of versions
LOGSEVERITY	1+Log of CERT/CC metric.
LEADER	First vendor(s) to patch the vulnerability.
HARDWARE	Vendors that also manufacture computer hardware

Table 2: Descriptive statistics

Variable	*N*	*Mean*	*Std. Dev*
Full Sample			
LOGDURATION	461	3.43	2.13
VENDORS	461	9.02	8.04
RIVALS	461	5.96	5.87
NONRIVALS	461	3.03	3.65
LOGQUANTITY	461	13.95	2.26
LOGVERSIONS	461	0.22	0.50
LOGSEVERITY	461	22.52	20.34
Instant Disclosure Sample			
LOGDURATION	321	3.60	2.30
VENDORS	321	7.88	7.32
RIVALS	321	5.38	5.54
NONRIVALS	321	2.47	2.81
LOGQUANTITY	321	14.07	2.19
LOGVERSIONS	321	0.19	0.48
SEVERITY	321	22.67	21.18
Non-instant disclosure Sample			
LOGDURATION	182	3.05	1.63
VENDORS	140	11.65	8.97
RIVALS	140	7.31	6.37
NONRIVALS	140	4.30	4.87
LOGQUANTITY	140	13.68	2.41
LOGVERSIONS	140	0.31	0.51
SEVERITY	140	22.18	18.35

5.2 Dependent variable

Our dependent variable is *DURATION*, the elapsed days from the date when the vendor came to know of the vulnerability until the vendor released a patch. *DURATION* depends on the regime of disclosure – *instant or non-instant disclosure*. If the vulnerability is *instantly disclosed*, *DURATION* is the elapsed time in days between when the vulnerability was publicly disclosed and the time the vulnerability was patched by the vendor. If the vulnerability was *non-instantly disclosed*, *DURATION* is the elapsed time between when CERT/CC informed the vendor of the existence of the vulnerability and when the vendor issued a patch. For the empirical analysis we use the log of one plus the number of elapsed days as our dependent variable. We label this variable *LOGDURATION*.

The final sample comprises 461 observations relating to 241 distinct vulnerabilities. Of these, 4.2%, or about 20 observations, had no patch. For these unpatched observations, we assign the maximum value of the dependent variable (8.27). As we will show below, our results are unchanged when we redo the analysis by using a tobit model that treats these observations as right censored. From Table 2 average *LOGDURATION* is higher under instant disclosure than under noninstant disclosure (3.60 v. 3.05).

5.3 Statistical Method and Identification

In this section, we describe our method for identifying how competition and disclosure influence vendor patching times. Our goal is to examine how the log of duration of patching times for vendor *i* in market *m* facing vulnerability *v* varies with changes in competition, disclosure, and quantity. To do this, one may estimate the linear model

$$LOGDURATION_{imv} = \beta_0 + \beta_1 COMPETITION_{imv} + \beta_2 DISCLOSURE_v + \beta_3 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \varepsilon_{iv} \quad (7)$$

Where X_i is a vector of vendor controls that includes vendor and market fixed effects and Z_v is a vector of vulnerability controls that includes severity metric. Our interest is in identifying the parameters β_1 through β_3 which reflect the effects of competition, disclosure, and market size, respectively. In practice, we use $RIVALS_{iv}$ to proxy for $COMPETITION_{iv}$ and $RIVALS_{iv} + NONRIVALS_{iv}$ to proxy for $DISCLOSURE_{iv}$, giving us

$$LOGDURATION_{imv} = \beta_0 + \beta_1 RIVALS_{imv} + \beta_2 (RIVALS_{imv} + NONRIVALS_{imv}) + \beta_3 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \varepsilon_{iv} \quad (8)$$

In the following three sections, we discuss alternative ways of estimating β_1 through β_3. Each of these models will make slightly different identification assumptions. By employing alternative identification assumptions, we hope to explore the robustness of our results.

5.3.1 Estimation using RIVALS and NONRIVALS

Our first approach estimates a variant of equation (8) that includes market-level and vendor fixed effects fixed effects, and vulnerability random effects, giving us the estimating equation

$$LOGDURATION_{imv} = \beta_0 + \alpha_1 RIVALS_{imv} + \beta_2 NONRIVALS_{imv} + \beta_3 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \mu_v + \varepsilon_{iv} \quad (9)$$

If $RIVALS_{iv}$ and $NONRIVALS_{iv}$ influence $LOGDURATION_{iv}$ linearly as assumed, then β_2 identifies the effects of disclosure, α_1 identifies the combined effects of disclosure and competition, and $\alpha_1 - \beta_2$ identifies the effect of competition. β_3 identifies how market size influences patching speed. X_i includes a vector of vendor fixed effects,[9] while Z_v includes controls for *LOGSEVERITY, LOGVERSIONS,* and a set of market fixed effects.[10] μ_v is a vulnerability random effect.

Identification of this model rests on several assumptions. First, identification of the effects of competition assumes linearity of *LOGDURATION* with respect to *RIVALS* and *NONRIVALS*. Prior empirical research in industrial organization has demonstrated that the impact of the marginal entrant on price competition is declining in the number of entrants (Bresnahan and Reiss 1991, Mazzeo 2002). If the influence of entry of quality competition is similarly nonlinear, then our estimates of the marginal effect of competition and disclosure will be inconsistent. Estimates of the model that take logs of *RIVALS* and *NONRIVALS* yield qualitatively similar results; however these alternative estimates do not allow us to recover the structural estimates of the marginal effect of increasing competition. Our estimates also depend on measurement of *RIVALS* and *NONRIVALS*. That is, they require us to accurately measure market boundaries. If *RIVALS* and *NONRIVALS* are measured inaccurately this will likely produce a bias towards zero in the estimates.

Identification in the model arises from variation in *RIVALS* and *NONRIVALS* within vendors that participate in multiple markets, and within markets that have multiple vendors. For example, consider a vulnerability in Util-linux package. This utility is not packaged in all LINUX variants but only in Red Hat, SCO and Sun. But then this package is also packaged in products by Juniper and Cisco. Hence for this vulnerability, the vendors listed vulnerable are - Red Hat, SCO, Sun, Juniper and Cisco. So for Sun,

[9] These are Apple, HP (includes HP, Compaq, and Digital), Microsoft, Sun, SCO, RedHat, and IBM (includes Lotus, iPlanet, and IBM). The omitted category includes a number of smaller vendors for which we have insufficient observations to identify a separate fixed effect. Appendix-I, Table A1 displays the distribution of proportions for vendor fixed effects. 26% of observations are from vendors that do not have a separate fixed effect. Vendors that do not have fixed effects are Adobe, Allaire, Compaq, Macromedia, Netscape, Network Associates, Novell, Oracle, SGI, Symantec, Trend Micro, and Veritas.

[10] These are for operating systems, web browsers, application development software, database management, groupware software, and web server software. Each category includes a minimum of 15 observations. The omitted category includes small markets for which we have insufficient observations to identify a separate fixed effect. Appendix Table 1 includes the distribution of proportions. 5.09% of observations are from markets that do not have a separate fixed effect.

Red Hat and SCO, *RIVALS* =2 while For Juniper and Cisco *RIVALS* =1. *NONRIVALS* for Cisco and Juniper is 3 and *NONRIVALS* for Red Hat, Sun and SCO is 2.

So with Rivals and NonRivals the variation is between and within vulnerabilities. Moreover, a percentage (26%) of observations are for vendors appears infrequently in our sample and so do not have a separate fixed effect, so our estimates will also reflect a small amount of cross-vendor variation. We retain these observations to maintain a sample that reflects the distribution of vendor sizes across the population, however as a robustness check we re-estimate the model using only vendors for which we can estimate separate fixed effects for each vendor and show that the results are qualitatively similar.

Our model also assumes 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, we recognize that *LOGQUANTITY* may reflect in part recent demand for software products. If so, then this would lead to a downward bias on our estimate of β_3 ; that is, it would lead us to overstate the relationship between market size and quality provision. Unreported estimates, which exclude *LOGQUANTITY* , yield very similar estimates for other variables, indicating that the bias, if any, does not extend to other estimates.

5.3.2 Estimation using instant disclosure

While our first model relied on measurement of *RIVALS* and *NONRIVALS* and a linearity assumption to identify competition and disclosure, our second model utilizes variation in the vulnerability disclosure regime to to place bounds on the structural parameters β_1 and β_2. The effect of the marginal vendor on disclosure is equal to zero under instant disclosure since the vulnerability has already been disclosed. Using this information, we can decompose the effect of number of vendors on patching times as follows:

$$LOGDURATION_{imv} = \beta_0 + \beta_1 VENDORS_{iv} + \beta_2 (1 - INSTANT_v) * VENDORS_{iv} + \beta_3 INSTANT_v + \beta_4 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \mu_v + \varepsilon_{iv} \quad (10)$$

where *INSTANT* is a binary variable that is equal to one when a vulnerability is instantly disclosed. Note that the effect of the marginal vendor on disclosure is equal to zero when *INSTANT*=1 since the vulnerability has already been disclosed. We estimate the following model:

$$LOGDURATION_{imv} = \beta_0 + \gamma_1 VENDORS_{iv} + \gamma_2 INSTANT_v * VENDORS_{iv} + \gamma_3 INSTANT_v + \beta_3 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \mu_v + \varepsilon_{iv} \quad (11)$$

where $\gamma_1 = \beta_1 + \beta_2$ identifies how increases in the number of vendors will lower patching times through increases in competition, while $\gamma_2 = -\beta_2$ identifies how increases in the number of vendors leads to lower patching times through disclosure. Because some vendors will not be rivals, *VENDORS* overestimates the

number of rivals so that $\gamma_1 = \beta_1 + \beta_2$ is an underestimate of the competition effect, while $-\gamma_2$ is an unbiased estimate of the disclosure effect.

We next contrast the identification assumptions of model (11) with model (9). In contrast to model (9), model (11) does not rely on accurate identification of *RIVALS* and *NONRIVALS*. Moreover, while we do assume that *LOGDURATION* is linear in *VENDORS,* the identification strategy is robust to alternate functional form assumptions, e.g., using the log of vendors yields similar results. However, this flexibility comes at some cost. First, as noted above, we are able only to place a lower bound on the competition effect using this model. Second, this model introduces the possibility that *INSTANT* is endogenous: instantly disclosed vulnerabilities may differ in some unobservable way that influences patching times.

To control for this potential source of endogeneity, we present the results of instrumental variable (IV) regressions that use data on theB identity of the identifier of the vulnerability as instruments for *INSTANT*. Our results suggest that the endogeneity, if any, is minor and the IV estimates are similar to the featured estimates.

The effect of *LOGQUANTITY* on patching times may be different for software vendors that also sell hardware than for other firms since 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 patching times as 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 (like IBM, HP, Sun)

5.3.3 Estimation using rivals, nonrivals, and instant disclosure

Our third model combines both approaches, identifying the competition and disclosure effects using rivals and nonrivals. We used equation (9) to demonstrate how competition could be identified using variations in the number of rivals and nonrivals. Since the effects of disclosure will only be felt under noninstant disclosure, we can rewrite equation (9) as

$$LOGDURATION_{imv} = \begin{array}{c}\beta_0 + \alpha_1 RIVALS_{imv} + \beta_2 (1 - INSTANT) * (RIVALS_{imv} + NONRIVALS_{imv}) + \\ \beta_3 LOGQUANTITY_{imv} + \theta_1 X + \theta_2 Z_v + \mu_v + \varepsilon_{iv}\end{array} \quad (12)$$

Collecting terms gives us the following estimation equation

$$\begin{array}{l}LOGDURATION_{imv} = \beta_0 + \gamma_1 RIVALS_{imv} + \beta_2 NONRIVALS_{imv} - \beta_2 INSTANT_v * RIVALS_{imv} - \\ \qquad \beta_2 INSTANT_v * NONRIVALS_{imv} + \beta_3 LOGQUANTITY_{im} + \theta_1 X_i + \theta_2 Z_v + \mu_v + \varepsilon_{iv}\end{array} \quad (13)$$

where $\gamma_1 = \beta_1 + \beta_2$. Equation (13) suggests that the model is over-identified. We test for and are unable to reject the over-identification restrictions that the coefficient on *NONRIVALS* is equal to that of *INSTANT*NONRIVALS* and that the coefficient on *INSTANT*RIVALS* is equal to that of *INSTANT*NONRIVALS*. Thus, we estimate two sets of models: one with no over-identification restrictions and one in which we constrain the coefficients of *NONRIVALS, INSTANT*NONRIVALS,* and *INSTANT*RIVALS* to be equivalent to one another.

6. Result and Discussion

We begin with some simple comparison of conditional means and then proceed with the regression analysis.

6.1 Analysis of conditional means

In table 3, we provide some preliminary evidence on the effects of competition and disclosure through an examination of conditional means. We categorize *VENDORS* as "high" if the natural log of number of VENDORS for a vulnerability was above the median and "low" otherwise. As noted above, increases in vendors under instant disclosure will influence patching times only through their effect on competition. Thus, under instant disclosure the difference in the sample means of *LOGDURATION* between categories identifies the effect of increasing competition on patching times: an increase in the number of vendors from below the median to above the median lowers *LOGDURATION* by a statistically significant 0.58. Changes in disclosure regime will lower patching times due to the disclosure effect;. thus, differences in *LOGDURATION* across the disclosure regimes provide an estimate of the disclosure threat effect. Moreover, the disclosure threat increases with the number of vendors. Column (3) shows that the data support these hypotheses: instant disclosure leads to a reduction in DURATION by about 0.93 when VENDORS is high and by about 0.60 when VENDORS is low. All of these differences are statistically significant.

Table 3: Comparison of conditional mean of *LOGDURATION*

Disclosure / No of Vendors	*Instant disclosure (1)*	*Non instant disclosure (2)*	*Disclosure effect* (3)
High (Above Median)	3.42*** (0.21)	2.49*** (0.16)	-0.93***(0.29)
Low (Below Median)	4.00*** (0.17)	3.40*** (0.24)	-0.60** (0.29)
Competition effect	-0.58***(0.27)	*Disclosure+ competition effect = -0 .81*	

Notes: Cells demonstrate mean of DURATION conditional on different combinations of disclosure and number of vendors. Standard errors in parentheses. Sample median of vulnerable vendors=6. * *Significant at 90% confidence level.* ** *Significant at 95% confidence level.* *** *Significance at 99% confidence level.*

6.2 Regression results using rivals and nonrivals

We begin by estimating an OLS regression with clustering in standard errors by vendor. We then estimate a GLS model with random effects at the vulnerability level. Since our results are qualitatively similar across the two models, we use the random effects model as our baseline in this section and throughout the rest of the paper. As a robustness check to our treatment of right-censored observations, we also estimate a random effects tobit model in which unpatched observations are treated as right-censored. Last, we estimate a vendor fixed effects model with only the top 7 vendors.

Table 4: Identification using Rivals and Nonrivals - Model (9)

Variable	*OLS with cluster correction (1)*	*Random effects GLS (2)*	*Random effects Tobit (3)*	*Random effects GLS, Sample of "fixed effects" vendors (4)*
RIVALS $\alpha_1 = \beta_1 + \beta_2$	-0.07*** (0.04)	-0.08*** (0.03)	-0.08*** (0.02)	-0.07*** (0.03)
NONRIVALS (β_2)	-0.06 (0.04)	-0.08*** (0.02)	-0.08*** (0.03)	-0.05** (0.03)
LOGQUANTITY (β_3)	-0.18** (0.06)	-0.13*** (0.05)	-0.14*** (0.05)	-0.07 (0.08)
LOGVERSIONS	0.65*** (0.20)	0.29** (0.13)	0.31** (0.18)	0.16 (0.21)
LOGSEVERITY	-0.27 (0.23)	-0.12 (0.13)	-0.15 (0.14)	-0.12 (0.15)
HARDWARE*LOGQUANTITY	0.31*** (0.15)	0.27*** (0.11)	0.28** (0.13)	0.21* (0.12)
Constant	8.01*** (1.09)	6.86*** (0.89)	7.28*** (0.85)	6.86*** (0.89)
N	461	461	461	340
R-squared	0.17	0.15	-	0.09
Log Likelihood	-	-	-905.75	-
R-squared (between)	-	0.13	-	0.10
#vulnerabilities	241	241	241	213
Market fixed effects	2	2	2	2
Vendor Fixed effects	7	7	7	7
σ_u	-	1.71	1.76	1.66

Standard errors in parentheses. * *Significant at 90% confidence level.* ** *Significant at 95% confidence level.* *** *Significance at 99% confidence level.*

Table 4 shows that, the number of nonrivals has a significantly negative effect (at the 1% level) on patching times across all specifications, except the OLS model. One additional nonrival will lower patching times by between 8 and 13 days, or by about 5-8%, depending on the specification. The direct effects of competition appear to be small. The coefficient estimate on *RIVALS* is negative and statistically significant (at the 1% level) in all models. However, the coefficient estimates are similar in magnitude to those for *NONRIVALS.* As noted above the structural parameter estimate for competition is equal to β_1, the difference between the coefficient for rivals and non-rivals. This estimate of β_1 is small in magnitude and statistically insignificant, ranging from -0.01 to -0.02. These estimates imply that although one additional rival is associated with between a 5% and 8% decline in duration times, the contribution from the competition effect is modest and insignificant.

Columns (1) through (3) show that increase in *LOGQUANTITY* has a statistically significant negative effect on patching times for non-hardware vendors. The effect *LOGQUANTITY* is lower for the subsample of large vendors. Large vendors in our sample are disproportionately hardware vendors. Thus, the different estimate in column (4) may reflect in part the smaller sample[11] from which we are estimating this parameter. Columns (1) through (3) imply that a 10% increase in quantity is associated with between a 1.1% and 1.8% decline in duration. In contrast, for hardware vendors, increases in *LOGQUANTITY* have no significant effect on patching times, and may even modestly increase patching times.

6.3 Regression results using instant disclosure

In this section we identify the effects of competition, disclosure, and installed base using variation in disclosure (Equation (11)). The results are presented in Table 5.

Table 5 - Results for identification using Instant Disclosure – Model (11)

Variable	Random effects GLS (1)	Random effects Tobit (2)	Random effects IV (3)	Sample of "fixed effects" vendors (4)
INSTANT	-0.20 (0.33)	-0.16 (0.34)	0.38 (1.44)	-0.09 (0.39)
VENDORS ($\gamma_1 = \beta_1 + \beta_2$)	-0.10*** (0.03)	-0.11*** (0.03)	-0.09* (0.04)	-0.09*** (0.03)
INSTANT*VENDORS ($\gamma_2 = -\beta_2$)	0.06** (0.03)	0.05* (0.03)	0.04 (0.06)	0.07** (0.03)
LOGQUANTITY (β_3)	-0.14*** (0.05)	-0.16*** (0.05)	-0.14*** (0.06)	-0.08 (0.08)
LOGVERSIONS	0.30* (0.17)	0.33* (0.13)	0.35* (0.20)	0.21 (0.26)
LOGSEVERITY	-0.13 (0.13)	-0.15 (0.13)	-0.13 (0.13)	-0.14 (0.14)
HARDWARE*LOGQUANTITY	0.27** (0.12)	0.28** (0.12)	0.29** (0.13)	0.20 (0.15)
Constant	7.07*** (0.82)	7.46*** (0.85)	6.71*** (1.19)	5.87*** (1.19)
N	461	461	461	340
R-squared	0.16	-	0.17	0.12
Log Likelihood	-	-903.13	-	-
R-squared (between)	0.14	-	0.15	0.11
#vulnerabilities	241	241	241	213
Market fixed effects	2	2	2	2
Vendor Fixed effects	7	7	7	7
σ_u	1.71	1.76	1.57	1.64

Standard errors in parentheses. ** Significant at 90% confidence level. ** Significant at 95% confidence level. *** Significance at 99% confidence level.*

Columns (1) and (2) show that estimates of the disclosure effect using random effects GLS and Tobit models yield results that are qualitatively similar to the estimates in section 6.2. One additional vendor is associated with between a 4% and 7% decline in duration times[12] due to disclosure threat. The estimate of the competition effect (β_1) can be recovered by adding the coefficient estimate of *INSTANT*VENDORS* to that of *VENDORS*. The estimates for β_1 are marginally significant and suggest that one additional vendor

[11] There are 112 observations comprising of hardware vendors.

[12] Recall that the parameter estimate for the disclosure effect is $\gamma_2 = -\beta_2$, so these estimates are equal to -0.06 and -0.05 in columns 1 and 2.

is associated with between a 2% and 6% decline in duration times due to the competition effect. The competition effect as estimated here is much larger (in absolute value) than that in section 6.1. This is surprising since our identification strategy in section 5.2 suggested that these estimates should overestimate the competition effect, although measurement error in measures of rivals and non-rivals may have biased the estimated competition effect towards zero. Estimates of *QUANTITY* variables are qualitatively similar to those in section 6.2.

For the remainder of Table 5, we use instrumental variable techniques to examine the assumption of exogenous capabilities (column 3 and 4). We employ random effects instrumental variable models. We define eight instruments. First, we instrument for instant disclosure using four dummy variables indicating the source of disclosure: *THIRD-PARTY-CONSULTANT*, *INDIVIDUAL, USER*[13], and *UNIVERSITY*. These groups have different incentive to publicly disclose vulnerabilities. For instance, a *THIRD-PARTY-CONSULTANT* is in general more likely to publicly disclose vulnerabilities as opposed to *INDIVIDUAL* identifiers who are more likely to work with either vendors or CERT. However, they are unlikely to be correlated with duration of patching times, conditional on our other right hand side variables. Next, we instrument for *INSTANT*VENDORS* by interacting the previous four instruments with *VENDORS*. We therefore have eight instruments for two endogenous variables (*INSTANT*, and *INSTANT*VENDORS)*.

The results of the instrumental variable regressions are in column (3). Overall, they are qualitatively similar to the results in columns (1) and (2), however, the parameter estimate of γ_2 is smaller and no longer statistically significant suggesting that we are not able to measure any significant change in duration times due to the disclosure effect. Estimate on γ_1 imply that one additional vendor is associated with about a 6% decline in duration times due to the competition effect. Finally, column (4) reports the results for the sample of top 7 vendors, and shows similar results.

6.4 Regression results using rivals, nonrivals, and instant disclosure

In this section we combine the identification strategies from the prior two subsections and estimate Equation (13). They are reported in Table 6. In columns (1) and (2) we regress *LOGDURATION* on the independent variables in equation (13) while allowing all parameter estimates to be driven by the data; that is, we do not place any restrictions on the parameter estimates. Estimation of the unconstrained model yields several estimates of the disclosure effect. Only one of these estimates is significant: the coefficient estimate on the number of nonrivals. Using these coefficient estimates, a one unit increase in the number of nonrivals is associated with between 9% and 12% decline in duration times. To state it another way, an

[13] An *INDIVIDUAL* is an identifier that is not a security consulting firm; A *USER* is an end-user of the software.

increase in one nonrival will lower patching times by about 15 to 20 days. Other estimates of the disclosure effect were small and generally statistically insignificant. The effect of *QUANTITY* is very similar to the earlier estimates; about 1.4% to 1.5% decrease in *DURATION* for a 10% increase in quantity. The competition effect (estimate on rivals) is between 8% to 12% or about 13 to 20 days.

Table 6: Results for identification using Rivals, Nonrivals, and Instant Disclosure

Variable	Random effects unconstrained model (1)	Random effects IV unconstrained model (2)	Random Effects constrained model (3)	Random Effects IV constrained model (4)
INSTANT	-0.13 (0.38)	0.05 (0.92)	-0.21 (0.37)	0.75 (0.64)
INSTANT*RIVALS ($-\beta_2$)	0.01 (0.04)	0.06 (0.07)	-0.05* (0.03)	-0.04*** (0.01)
INSTANT*NONRIVALS ($-\beta_2$)	0.10* (0.06)	0.04 (0.09)	-0.05* (0.03)	-0.04*** (0.01)
RIVALS (β_1)	-0.08** (0.04)	-0.11** (0.05)	-0.10*** (0.03)	-0.04* (0.03)
NONRIVALS(β_2)	-0.12*** (0.04)	-0.09* (0.05)	-0.05* (0.03)	-0.04*** (0.01)
LOGQUANTITY($-\beta_3$)	-0.15*** (0.05)	-0.14*** (0.05)	-0.13*** (0.05)	-0.14*** (0.06)
LOGVERSIONS	-0.27 (0.17)	-0.35* (0.19)	-0.27 (0.17)	-0.32** (0.13)
LOGSEVERITY	-0.13 (0.13)	-0.13 (0.13)	-0.15 (0.13)	-0.10 (0.14)
HARDWARE*LOGQUANTITY	0.23** (0.13)	0.27*** (0.12)	0.28** (0.12)	0.34*** (0.11)
Constant	7.06*** (0.86)	6.91*** (1.15)	6.81*** (0.85)	6.33*** (0.87)
N	461	461	461	461
R-squared	0.15	0.15		
Log Likelihood	-	-		
R-squared (between)	0.13	0.13		
#vulnerabilities	241	241	241	241
Market fixed effects	2	2	2	2
Vendor Fixed effects	7	7	7	7
σ_u	1.73	1.73		

* *Significant at 90% confidence level.* ** *Significant at 95% confidence level.* *** *Significance at 99% confidence level.* § *In columns (3) and (4) the coefficient of INSTANT*RIVALS and INSTANT*NONRIVALS are constrained to be equal to that of NONRIVALS. R-squared not reported for constrained models.*

As noted in section 5.3 the unconstrained model may be over-identified, and the small and generally insignificant estimates of *INSTANT*RIVALS* and *INSTANT*NONRIVALS* in columns (1) and (2) suggest that over-identification may be a problem. We conducted a series of hypothesis tests and failed to reject the constraints that the coefficient on *NONRIVALS* is equal to that of *INSTANT*NONRIVALS* (χ^2 - 0.12; p-value 0.73 under Random Effects GLS and χ^2 - 0.82; p-value 0.36 under random effects IV) and the coefficient on *INSTANT*RIVALS* is equal to that of *INSTANT*NONRIVALS* (χ^2 – 1.39; p-value 0.24 under random effects GLS and χ^2 - 0.03; p-value 0.86 under random effects IV) We then estimated random effects GLS specification as well as a random effects IV specification constraining *INSTANT*NONRIVALS* to be equal to *NONRIVALS* and *INSTANT*RIVALS* to be equal to *INSTANT*NONRIVALS*.

Columns (3) and (4) provide the parameter estimates of the constrained models. These results are qualitatively similar to those in columns (1) and (2). A unit increase in the number of nonrivals is associated with a 4% to 5% decline in patch release times which is equivalent an expected decline of 7 to 8 days due to the disclosure effect. A one unit increase in rivals is associated with a 4% to 10% decrease in duration due to competition effect. This would lead to a decline of 7 to 17 days per rival. The results of increases in quantity are qualitatively similar to those shown in other sections. A 10% increase in *LOGQUANTITY* is associated with between a 1.3% and 1.4% decline in duration.

6.5 Actual Disclosure and Threat of Disclosure

The model in section 3 assumes that the vendors make one-time investment decisions to patch vulnerable software. Our results have demonstrated that competition and disclosure each influence patching times. Thus far, we have interpreted our disclosure results as reflecting the threat of potential disclosure: that is, increases in the number of vendors sharing a common vulnerability reduce patching times due to the expectation of earlier public disclosure. However, one alternative hypothesis is that our results reflect the impact of actual disclosure. That is, vendors may increase their investments in patching vulnerabilities once disclosure takes place. This alternative interpretation of our results is consistent with our primary hypothesis—that investment in software quality is influenced by changes in market structure—however it offers an explanation for this relationship that is different from that posited by our theory model.

To explore the salience of this alternative interpretation, we examine the patching behavior of vendors that were the first to release patches for vulnerabilities. We label these vendors as "leaders." Because we are explicitly selecting on vendors that were faster to patch than their competitors, estimates in this sample may overstate the competition effect if vendors in this sample are systematically more sensitive to competition. Further, if potential disclosure influences patching behavior, estimates from this sample may also overstate the disclosure effect. However, if potential disclosure does not influence patching behavior, then we should expect to see zero disclosure effect. Thus, these estimates will enable us to identify directionally of whether a disclosure effect exists, however the parameter estimates of its magnitude will in general be inconsistent.

Recall that our sample had 241 vulnerabilities and 461 observations. Of the 241 vulnerabilities in our sample, 68 vulnerabilities had non-commercial/foreign vendors as leaders and hence we do not include them. 14 vulnerabilities had not patched at all. This led to a sample of 155 unique vulnerabilities and 179 observations. The number of observations is greater than the number of vulnerabilities because some vulnerabilities have multiple vendors that patch first on the same day.

In an analysis parallel to Table 3, we first split the number of nonrivals in this sample into "low" and "high" groups based on whether they were above or below the median. We then examined if *LEADER* were sensitive to the number of *NONRIVALS* for a vulnerability. Although the effect of disclosure threat in general is associated with an earlier patch release, the effect of disclosure threat among *LEADERS* is not statistically significant.

Table 7 Impact of number of NONRIVALS on LEADERS' DURATION

Severity	High	Low	Ave. Disclosure threat
NONRIVALS	3.33*** (0.22)	3.59*** (0.47)	-0.26 (0.52)

*** $p < 0.01$ ** $p < 0.05$ * $p<0.10$.

We further estimated multivariate regressions on this sample to test if leaders were sensitive to disclosure threat. About 110 vulnerabilities had no other affected vendors (only 1 vendor), and 45 consisted of vendors that faced more than one rival. Because of the small number of observations with a common vendor, we were unable to estimate random effects models and instead use OLS with standard errors corrected for clustering on vendors. In addition, there was relatively little variation in the number of vendors affected by the vulnerability. Thus, in this limited sample, we were only able to utilize our second identification strategy—instant disclosure using total vendors.

Table 8- OLS with for the LEADER sample

Variable	*OLS with cluster corrected standard errors (1)*	*Tobit (2)*	*IV with cluster corrected std. errors (3)*
INSTANT	0.72 (0.51)	0.91* (0.51)	0.66 (1.01)
INSTANT*VENDORS ($\gamma_2 = -\beta_2$)	0.05 (0.09)	0.03 (0.09)	0.02 (0.10)
VENDORS ($\gamma_1 = \beta_1 + \beta_2$)	-0.07*** (0.03)	-0.07* (0.04)	-0.06*** (0.02)
LOGQUANTITY (β_3)	-0.21*** (0.08)	-0.26*** (0.10)	-0.22*** (0.09)
LOGVERSIONS	1.09*** (0.38)	1.25*** (0.39)	1.09*** (0.39)
LOGSEVERITY	-0.06 (0.19)	-0.09 (0.18)	-0.04 (0.19)
HARDWARE*LOGQUANTITY	0.44** (0.16)	0.52*** (0.19)	0.43*** (0.17)
Constant	7.13*** (1.24)	8.00*** (1.40)	7.35 (1.29)
N	179	179	179
#vulnerabilities	155	155	155
Market fixed effects	2	2	2
Vendor Fixed effects	*3*	*3*	*3*

*** $p < 0.01$ ** $p < 0.05$ * $p<0.10$. +*Cluster corrected on vulnerability.*

These results suggest that disclosure influenced the speed with which leaders in our sample patched vulnerabilities, although the parameter estimate is not statistically significant (estimate γ_2). A unit increase in vendors is associated with a 2% to 5% decline in duration times due to disclosure threat. Stated otherwise the effect of disclosure among *LEADERS* is about 3 to 8 days per *VENDOR*. Thus, these results are suggestive that potential disclosure plays an important role in influence vendor patching behavior.

7. Discussion and conclusion

In this study, we show how competition, disclosure, and market size influence decisions by software vendors to invest in one key area of product quality: the patching of software vulnerabilities. Table 8 provides a summary of key findings from three models that use alternate identification strategies. We show that a one unit increase in the number of rivals reduce patching times between 7 to 17 days due to increasing internalization of customer losses. Perhaps more importantly, we demonstrate that changes in the number of nonrivals lower patching times through their impact on disclosure. The average effect of the presence of nonrivals lowers expected patching time from 7 to 8 days. Last, we demonstrate that increases in market size leads to lower patching times: a 10% increase in quantity leads to a 1.3% to 1.4% decline in patching times.

Table 8: Average effect of competition and disclosure threat

	Random Effects Model, Identification using Rivals and Nonrivals (2)	Random Effects IV Model, Identification using Instant Disclosure (3)	Random Effects IV Model, Identification using Rivals, Nonrivals, and Instant Disclosure (4)
Competition Effect (A unit increase in number of competitors)	7%	1%	4%
Disclosure Effect (A unit increase in number of nonrivals)	1%	6%	4%
Disclosure Effect + Competition Effect	8%	7%	8%
Quantity Effect (10% increase in the quality sold)	-1.3%	-1.4%	-1.4%

Column (2) uses baseline GLS estimates in column (2) of Table 4, columns (3) and (4) use instrumental variable estimates in column (5) of Tables 5 and 6.

This research provides evidence on how competition influences quality provision in information technology markets. More generally, we further recent efforts to understand how product market decisions are influenced by changes in market structure in technologically related markets (Bresnahan and Greenstein 1999). Empirical research on this topic remains relatively rare because of the difficulty in obtaining data sets with systematic variation in same and related markets.[14] We provide a framework for understanding how vendors in one market may influence quality provision in another. Further, in contrast to prior research, we show that such linkages can be important even when vendors operate in unrelated output markets.

[14] As a result, leading research in this area often uses a case study approach to collect evidence supporting or refuting hypotheses (e.g., Bresnahan and Greenstein 1999; Gawer and Henderson 2005; Gawer and Cusumano 2002).

These results also 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 type of regulation that would minimize social losses from vulnerabilities critically depends upon 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, threat of disclosure from vendors in complementary markets works to reduce patching times almost as much as increases in the number of competitors. Our research also points to the fact that making markets more competitive by removing entry barrier does increase internalization factor λ (The alternative being to make vendors explicitly liable by regulating).

By demonstrating that disclosure threat can be used as a tool to induce vendors to patch vulnerabilities faster, our results inform the debate on software quality in another way. Our results suggest that *non-instant* disclosure could be more welfare-enhancing than *instant disclosure.* In particular, our results suggest that for policy markets like CERT/CC, any disclosure policy should influence judicious use of disclosure threat to elicit faster vendor responses to vulnerabilities.

References:

Arora A., Caulkins J., Telang R. (2005) “Sell First, Fix Later: Impact of Patching on Software Quality", Management Science (Forthcoming)

Arora A., Krishnan R, Telang R. & Yang Y. (2005) "An Empirical Analysis of Vendor Response to Disclosure Policy," Workshop on Economics of Information Security (WEIS05), Kennedy School of Government, Harvard University, 2005.

Arora A., Nandkumar A. & Telang R. (2004) "Impact of patches and software vulnerability information on frequency of security attacks - An empirical analysis, Working paper," in: *H.John Heinz III school of Public Policy and Management*, Carnegie Mellon University, Pittsburgh, PA, 2004.

Arora A., Telang R. & Xu H. (2004)"Optimal Policy for Software Vulnerability Disclosure," The Third Annual Workshop on Economics and Information Security (WEIS04), University of Minnesota, 2004.

Borenstein S. and 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, 20(3):344-365.

Bresnahan, T., and S. Greenstein, (1996), Technical Progress in Computing and in the Uses of Computers. Brookings Papers on Economic Activity, Microeconomics, 1-78.

Bresnahan, T. and P-Y Lin, (2006), Economic and Technical Drivers of Technology Choice: Browsers. Working Paper, Harvard Business School, Harvard University.

Bresnahan, T. and P. Reiss, (1991), Entry and Competition in Concentrated Markets. *Journal of Political Economy* 99: 977-1009.

Bresnahan, T., S. Stern, and M. Trajtenberg (1997), Market Segmentation and the Sources of Rents from Innovation: Personal Computers in the late 1980s” *RAND Journal of Economics* 28(Special Issue): S17-S44.

Cavusoglu H., H. Cavusoglu, S. Raghunathan (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. and 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.

Dranove D. and W.White (1994), “Recent Theory and Evidence on Competition in Hospital Markets,” Journal of Economics and Management Strategy, 3(1):169-209.

Domberger S. and A. Sherr (1989), “The impact of competition on pricing and Quality of Legal Services,” International Review of Law and Economics, 9:41-56.

Dixit A.K (1986), “Comparative statics in Oligopoly.” *International Economic Review*, 27(1):107-122

Forman C., Goldfarb A., and Greenstein S. (2005), “How did location affect adoption of the commercial Internet? Global village vs. urban leadership” Journal of Urban Economics 58: 389-420.

Gal-Or E., (1983), “Quality and quantity competition” The Bell Journal of Economics, 14(2):590-600.

Gawer, A. and M. Cusumano (2002), Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation. Boston: Harvard Business School Press.

Gawer, A. and R. Henderson (2006), Platform Owner Entry and Innovation in Complementary Markets: Evidence from Intel. NBER Working Paper #11852.

Greenstein, S. (2000), "Building and Developing the Virtual World: The Commercial Internet Access Market," *Journal of Industrial Economics* 48(4): 391-411.

Greenstein, S. and S. Markovich (2006), "Pricing in a Seemingly Homogenous Technology: Electronic Business Service Providers" Working Paper, Kellogg School of Management, Northwestern University.

Hoxby C. (2000), "Does Competition among Public Scools benefit Students or Taxpayers?," American Economic Review, 90(5):1209-1238.

Lerner, J. and J. Tirole (2005) "The Economics of Technology Sharing: Open Source and Beyond" *Journal of Economic Perspectives* 19(Spring): 99-120.

Levhari D. and Peles Y., (1973), "Market Structure, Quality and Durability." The Bell Journal of Economics and Management Science, 4(1): 235-248

Mazzeo, M. (2002), "Product choice and oligopoly market structure" *RAND Journal of Economics* 33(2): 1-22.

Mazzeo M. (2003), "Competition and Service Quality in the U.S. Airline Industry," Review of industrial Organization, 22: 275-296

Schmalensee R. (1979), "Market Structure, durability, and Quality: A Selective Survey," Economic Inquiry, 17: 177-196

Schneier B. (2000) "Full Disclosure and the Window of Exposure," in: *CRYPTO-GRAM*, 2000.

Nizovtsev, D.T., M. "Economic Analysis of Incentives to Disclose Software Vulnerabilities," Workshop on Economics and Information Security (WEIS05), Kennedy School of Government, Harvard University, 2005.

Spence A.M., (1975), "Monopoly, Quality and Regulation" The Bell Journal of Economics 6(2): 417-429

Swan P.L., (1970), "Durability of Consumer Goods," American Economic Review, 60: 884-894

Telang R. and Wattal S. (2005) "Impact of Software Vulnerability Announcements on the Market Value of Software Vendors – an Empirical Investigation," Workshop on Economics of Information Security (WEIS05), Kennedy School of Government, Harvard University, 2005.

Appendix I - Proofs

Suppose, there are a total of *N* vendors affected by the vulnerability. There are *N-1* other vendors affected by the vulnerability. Let X_i be the random variable that denotes the actual patch release date with x_i its realization. Under uncertainty the vendor cost function is given by

$$\tilde{V}_i = \tilde{C}_i + q_i(\lambda_i + \omega_i)\tilde{\theta}_i$$

Let j denote the other vendors affected by the vulnerability. $j \in \{1,2,3\ldots n\}$ and $j \neq i$. Let $z \equiv \min\{x_1, x_2, ..., x_j, s\}$. Let *G(.)* be the distribution of x_i and let $\Phi(.)$ be the distribution of z

$$\tilde{C}_i \equiv \int_0^R C(x_i) dG(x_i : \tau_i) \text{ and } \tilde{\theta}_i \equiv \int_0^R \int_0^{x_i} \left(L(x_i - z) d\Phi(z : \tau_j, s, N)\right) dG(x_i : \tau_i)$$

a) Derivation of $\Phi(.)$:

$$\Phi(y) \equiv \Pr(z \leq y) = 1 - \Pr(z > y) = \Pr(x_1 > y)...\Pr(x_n > y)\Pr(s > y)$$

$$\Phi(y) \equiv 1 - [1 - \Pr(x_1 \leq y)]...[1 - \Pr(x_n \leq y)][1 - \Pr(s \leq y)]$$

Let $H_j(.)$ be the distribution of x_j

$$\Phi(y) \equiv 1 - [1 - H_j(y)]^N [1 - F(y)]$$

A1: On an average earlier patch release entails higher cost. Also, on an average the marginal cost to patch a vulnerability is increasing in τ (because the opportunity cost of freed resources is decreasing on an average). $\frac{\partial \tilde{C}_i}{\partial \tau_i} < 0; \frac{\partial^2 \tilde{C}_i}{\partial \tau_i^2} > 0$

A2: On an average later patch release results in higher end-user losses. Further, the marginal end user losses are increasing in τ. $\frac{\partial \tilde{\theta}_i}{\partial \tau_i} < 0; \frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i^2} > 0$

$\tilde{V}_i$ is convex in τ_i because $\frac{\partial^2 \tilde{V}_i}{\partial \tau_i^2} = \frac{\partial^2 \tilde{C}_i}{\partial \tau_i^2} + q_i(\lambda_i + \omega_i)\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i^2} > 0$ (due to assumptions A1 and A2).

b) $\frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial \tau_j} < 0 \ \forall j \neq i$

Proof: $\frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial \tau_j} = q_i(\lambda_i + \omega_i)\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial \tau_j}$

$$\tilde{\theta}_i \equiv \int_0^R \int_0^{x_i} \left(L(x_i - z) d\Phi(z : \tau_j, s, N)\right) dG(x_i : \tau_i) \equiv \int_0^R K dG(x_i : \tau_i)$$

Integrate by parts,

$$\tilde{\theta}_i \equiv KG(x_i : \tau_i)\Big|_0^R - \int_0^R K' G(x_i : \tau_i), \text{ where } K' \equiv \int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i} d\Phi(z : \tau_j, s, N) > 0$$

$$\tilde{\theta}_i \equiv K(R) - \int_0^R K' G(x_i : \tau_i) \text{ (Because } G(0)=0 \text{ and } G(R)=1)$$

$$\frac{\partial \tilde{\theta}_i}{\partial \tau_i} \equiv -\int_0^R K' \frac{\partial G(x_i : \tau_i)}{\partial \tau_i} > 0 \text{ by assumption A2}$$

$$\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial \tau_j} \equiv -\int_0^R \frac{\partial K'}{\partial \tau_j} . \frac{\partial G(x_i : \tau_i)}{\partial \tau_i}$$

Sign of $\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial \tau_j}$ depends of sign of $\frac{\partial K'}{\partial \tau_j}$

Integrate *K'* by parts,

$$K' = \Phi(z : \tau_j, s, N) \frac{\partial L(x_i - z)}{\partial x_i}\bigg|_0^{x_i} - \int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial z} \Phi(z : \tau_j, s, N)$$

$$K' = \Phi(z : x_i, s, N) \frac{\partial L(x_i)}{\partial x_i} - \int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial z} \Phi(z : \tau_j, s, N)$$

$$\frac{\partial K'}{\partial \tau_j} = -\int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial z} . \frac{\partial \Phi(z : \tau_j, s, N)}{\partial \tau_j}$$

$$\frac{\partial \Phi(z : \tau_j, s, N)}{\partial \tau_j} = N(1 - H_j(x_i))^{N-1} \frac{\partial H_j}{\partial \tau_j} \big(1 - F(x_i)\big) > 0$$

Thus $\frac{\partial K'}{\partial \tau_j} > 0$ given that $\int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial x_j} < 0$. Hence $\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial \tau_j}, \frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial \tau_j} < 0$

c) Strategies are complimentary between firms. The reaction functions re upward sloping

Proof:

We prove that for any vendor *j* the reaction curves are upward sloping, $\frac{d\tau_i}{d\tau_j} > 0$

Let $\tilde{V}_i^{\,i}$ denote the FOC for vendor *i* and likewise for all other vendors. Let $\tilde{V}_{ii}^{\,i}$ denote the SOC for vendor *i* with respect to τ_i.

From the FOC $\tilde{V}_i^i = 0$ and $\tilde{V}_j^j = 0 \;\; \forall j \neq i$

Totally differentiating the FOC

$$\tilde{V}_{ii}^i \, d\tau_i + \tilde{V}_{i1}^i \, d\tau_1 + . \cdots + \tilde{V}_{ij}^i \, d\tau_j = 0$$

.

.

.

$$\tilde{V}_{ji}^j \, d\tau_i + \tilde{V}_{j1}^j \, d\tau_1 + \cdots + \tilde{V}_{jj}^j \, d\tau_j = 0$$

Since we are interested in the reaction function with respect to vendor *j,* we set all $d\tau_x, x \neq i, j$ to 0. Thus,

$$\tilde{V}_{ii}^{i}\, d\tau_i + \tilde{V}_{ij}^{i}\, d\tau_j = 0$$
$$\tilde{V}_{11}^{1}\, d\tau_1 + \tilde{V}_{1j}^{1}\, d\tau_j = 0$$
$$\vdots$$
$$\tilde{V}_{ji}^{j}\, d\tau_i + \tilde{V}_{jj}^{j}\, d\tau_j = 0$$

Hence,

$$\frac{d\tau_i}{d\tau_j} = -\frac{\tilde{V}_{ij}^{i}}{\tilde{V}_{ii}^{i}} > 0; \frac{d\tau_1}{d\tau_j} = -\frac{\tilde{V}_{1j}^{1}}{\tilde{V}_{11}^{1}} > 0; \ \ldots\ldots \ \frac{d\tau_j}{d\tau_i} = -\frac{V_{ji}^{j}}{\tilde{V}_{jj}^{j}} > 0$$

Diagonal dominance implies that $\left|\tilde{V}_{ij}^{i}\right| < \left|\tilde{V}_{ii}^{i}\right|$ (Dixit, 1986). This implies that $\left|\frac{d\tau_j}{d\tau_i}\right|$<1

d) There exists a unique Nash equilibrium in pure strategy.

Proof:

i. The number of players in the game N is finite.

ii. $\tilde{C}_i(\tau_i)$ and $\tilde{C}_j(\tau_j)\ \forall j \neq i, j \in J$ are continuous for all values of τ_i and τ_j respectively and is bounded above and below. For all vendors $C(0) = \bar{C}$ and $C(R) = \underline{C}$. Hence $C(\tau)$ is bound in the closed interval $[\bar{C}, \underline{C}]$

iii. For all vendors $\tilde{\theta}(0) = 0$ and $\tilde{\theta}(R) = \bar{L}$. Hence $\tilde{\theta}$ is bound in the closed interval $[0, \bar{L}]$ $\tilde{\theta}_i(\tau_i), \tilde{\theta}_j(\tau_j)\ \forall j \neq i, j \in J$ are continuous for all values of τ_i and τ_j respectively and is bounded above and below.

iv. Given (2) and (3) the payoff function $\tilde{V}$ for all vendors is continuous and bounded above and below in the interval $[\bar{C}, \bar{L} + \underline{C}]$.

v. $\frac{\partial \tau_i^*}{\partial \tau_j} > 0$. For $\tau_i > \tau_j$, $\tau_i \leq R$. For $\tau_i \leq \tau_j, 0 \leq \tau_i \leq \tau_j^*$. Hence τ_j is bounded in the interval $[0, R]$. Thus τ_i is bounded.

vi. We know $\frac{\partial \tau_j^*}{\partial \tau_i} > 0$. For $\tau_j > \tau_i$, $\tau_j \leq R$. For $\tau_j \leq \tau_i, 0 \leq \tau_j \leq \tau_i$. Hence τ_j is bounded in the interval $[0, R]$. Thus τ_i and τ_j are bounded.

vii. Hence it is reasonable to regard τ_i and τ_j to fall in the closed interval $[0, R]$

viii. The set $S_i^{\geq} \equiv \{\tau_i \mid \tilde{V}_i(\tau_i, \tau_j) \geq k\}$ for any real number k is a convex set. Similarly $\forall j \neq i, j \in J$, $S_j^{\geq} \equiv \{\tau_j \mid V_j(\tau_j, \tau_i) \geq k\}$ Hence the payoff functions are quasi-concave.

ix. For each $n \in N$ let $\beta_n(\tau) = \{\tau_n \in P \mid V(\tau_{-n}, \tau_n) \geq V(\tau_{-n}, \tau_n') for \ all \tau_n' \in P\}$ be the best response of player *n.* Using (1) through (5), $\beta_n(\tau)$ is non-empty, compact and upper hemi-continuous. Then $\beta(\tau) = \beta_1(\tau) X ... X \beta_N(\tau)$ be the Cartesian mapping is non-empty, compact and upper hemi-continuous. By Kakutani fixed point theorem, the best response correspondence is the Nash equilibrium.

x. Since $\frac{d\tau_i^*}{d\tau_j^*}, \ldots, \frac{d\tau_j^*}{d\tau_i^*} < 1$, The Nash equilibrium is unique.

e) Increase in number of other vendors affected by the vulnerability increases disclosure threat and hence expected end-user losses.
Proof:

If *N* is the number of other vendors, we prove that $\frac{\partial \tau_i^*}{\partial N} < 0$ for vendor *i*

$\frac{\partial \tau_i^*}{\partial N} = -\frac{\frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial N}}{\frac{\partial^2 \tilde{V}_i}{\partial \tau_i^2}}$. We know that the denominator is >0. So now we only need to show that

that $\frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial N} > 0$.

FOC implies $\frac{\partial \tilde{V}_i}{\partial \tau_i} = \frac{\partial \tilde{C}_i}{\partial \tau_i} + q_i(\lambda_i + \omega_i)\frac{\partial \tilde{\theta}_i}{\partial \tau_i} = 0$, where $\frac{\partial \tilde{\theta}_i}{\partial \tau_i} \equiv -\int_0^R K' \frac{\partial G(x_i : \tau_i)}{\partial \tau_i}$

Now, $K' = \Phi(z : \tau_j, s, N)\frac{\partial L(x_i)}{\partial x_i} - \int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial z}\Phi(z : \tau_j, s, N)$. Since *K'* >0 and *K'* is increasing

in *N*, $\frac{\partial K'}{\partial N} > 0$. Hence $\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial N} > 0$. Thus, $\frac{\partial^2 \tilde{V}_i}{\partial \tau_i \partial N} = q_i(\lambda_i + \omega_i)\frac{\partial^2 \tilde{\theta}_i}{\partial \tau_i \partial N} > 0.$

Thus $\frac{\partial \tau_i^*}{\partial N} < 0$

f) Increase in disclosure threat results in vendors patching earlier.
Proof:
Let the $N+1^{th}$ vendor be denoted by subscript *x.* There are now *N* other vendors affected by the vulnerability. We show that $\frac{d\tau^*_i}{d\tau^*_x}, \ldots, \frac{d\tau^*_j}{d\tau^*_x} < 0$

With *N* other vendors, from the FOC $\tilde{V}_i^i = 0$, $\tilde{V}_j^j = 0 \ \ \forall j \neq i$ and $\tilde{V}_x^x = 0$, $\forall x \neq i, x \neq j$

Totally differentiating the FOC

$$\tilde{V}^i_{ii}\, d\tau^*_i + \tilde{V}^i_{i1}\, d\tau^*_1 + \cdots + \tilde{V}^i_{ij}\, d\tau^*_j + \tilde{V}^i_{ix}\, d\tau^*_x = 0$$

$$\vdots$$

$$\tilde{V}^j_{ji}\, d\tau^*_i + \tilde{V}^j_{j1}\, d\tau^*_1 + \cdots + \tilde{V}^j_{jj}\, d\tau^*_j + \tilde{V}^j_{jx}\, d\tau^*_x = 0$$

$$\tilde{V}^x_{xi}\, d\tau^*_i + \tilde{V}^x_{x1}\, d\tau^*_1 + \cdots + \tilde{V}^x_{xj}\, d\tau^*_j + \tilde{V}^x_{xx}\, d\tau^*_x = 0$$

Symmetry implies $d\tau^*_i = d\tau^*_1 = \ldots = d\tau^*_j = d\tau^*_x$ and $\tilde{V}^i_{i1} = \tilde{V}^i_{i2} = \ldots = \tilde{V}^i_{ij} = \tilde{V}^i_{ix}$. Thus

$$\left(\tilde{V}^i_{ii} + \sum_{k=1}^{j-1} \tilde{V}^i_{ik}\right) d\tau^*_i = -\tilde{V}^i_{ix}\, d\tau^*_x$$

$$\vdots$$

$$\left(\tilde{V}^j_{jj} + \sum_{k=1}^{j-1} \tilde{V}^j_{jk}\right) d\tau^*_j = -\tilde{V}^j_{jx}\, d\tau^*_x$$

$\forall j \neq i, k \neq i, k \neq j$. Thus,

$$\frac{d\tau^*_i}{d\tau^*_x} = \frac{-\tilde{V}^i_{ix}}{\left(\tilde{V}^i_{ii} + \sum_{k=1}^{j-1} \tilde{V}^i_{ik}\right)}; \ldots\ldots; \frac{d\tau^*_j}{d\tau^*_x} = \frac{-\tilde{V}^j_{ji}}{\left(\tilde{V}^j_{jj} + \sum_{k=1}^{j-1} \tilde{V}^j_{jk}\right)} \quad \forall j \neq i, k \neq i, k \neq j \text{ .Or,}$$

Where $\tilde{V}^i_{ix} \equiv -\int_0^R \frac{\partial K'}{\partial \tau_x} . \frac{\partial G(x_i : \tau_i)}{\partial \tau_i}$ with $\frac{\partial K'}{\partial \tau_x} = -\int_0^{x_i} \frac{\partial L(x_i - z)}{\partial x_i \partial z} . \frac{\partial \Phi}{\partial \tau_x}$ being the effect of disclosure threat of vendor *x* on vendor *i's* marginal loss. Likewise for other *j* vendors, $\forall j \neq i$. Stability requires diagonal dominance in the co-efficient matrix. (Dixit,1986). Hence the denominators > 0. Hence

$$\frac{d\tau^*_i}{d\tau^*_x}, \ldots, \frac{d\tau^*_j}{d\tau^*_x} < 0$$

g) A higher λ results in vendors releasing patch earlier.

Proof:

We prove that increase in λ results in an earlier τ for all firms in the case of a symmetric equilibrium. In other words, we show comparative static for a case where all vendors are confronted with a higher λ.

We know that $\frac{\partial \tilde{V}_i}{\partial \tau_i} = \frac{\partial \tilde{C}_i}{\partial \tau_i} + q_i(\lambda_i + \omega_i)\frac{\partial \tilde{\theta}_i}{\partial \tau_i} = 0$

$$\tilde{V}^i_{i\lambda} \equiv \frac{\partial \tilde{V}_i}{\partial \tau_i \partial \lambda_i} = q_i \frac{\partial \tilde{\theta}_i}{\partial \tau_i} > 0$$

$$\frac{\partial \tau^*_i}{\partial \lambda} = -\frac{\tilde{V}^i_{i\lambda}}{\tilde{V}^i_{ii}} < 0$$

h) A higher q results in vendors releasing patch earlier.

Proof:

As with λ, we prove that increase in q results in an earlier τ for all firms in the case of a symmetric equilibrium. In other words, we show comparative static for a case where all vendors are confronted with a higher q. We know that $\frac{\partial \tilde{V}_i}{\partial \tau_i} = \frac{\partial \tilde{C}_i}{\partial \tau_i} + q_i(\lambda_i + \omega_i)\frac{\partial \tilde{\theta}_i}{\partial \tau_i} = 0$

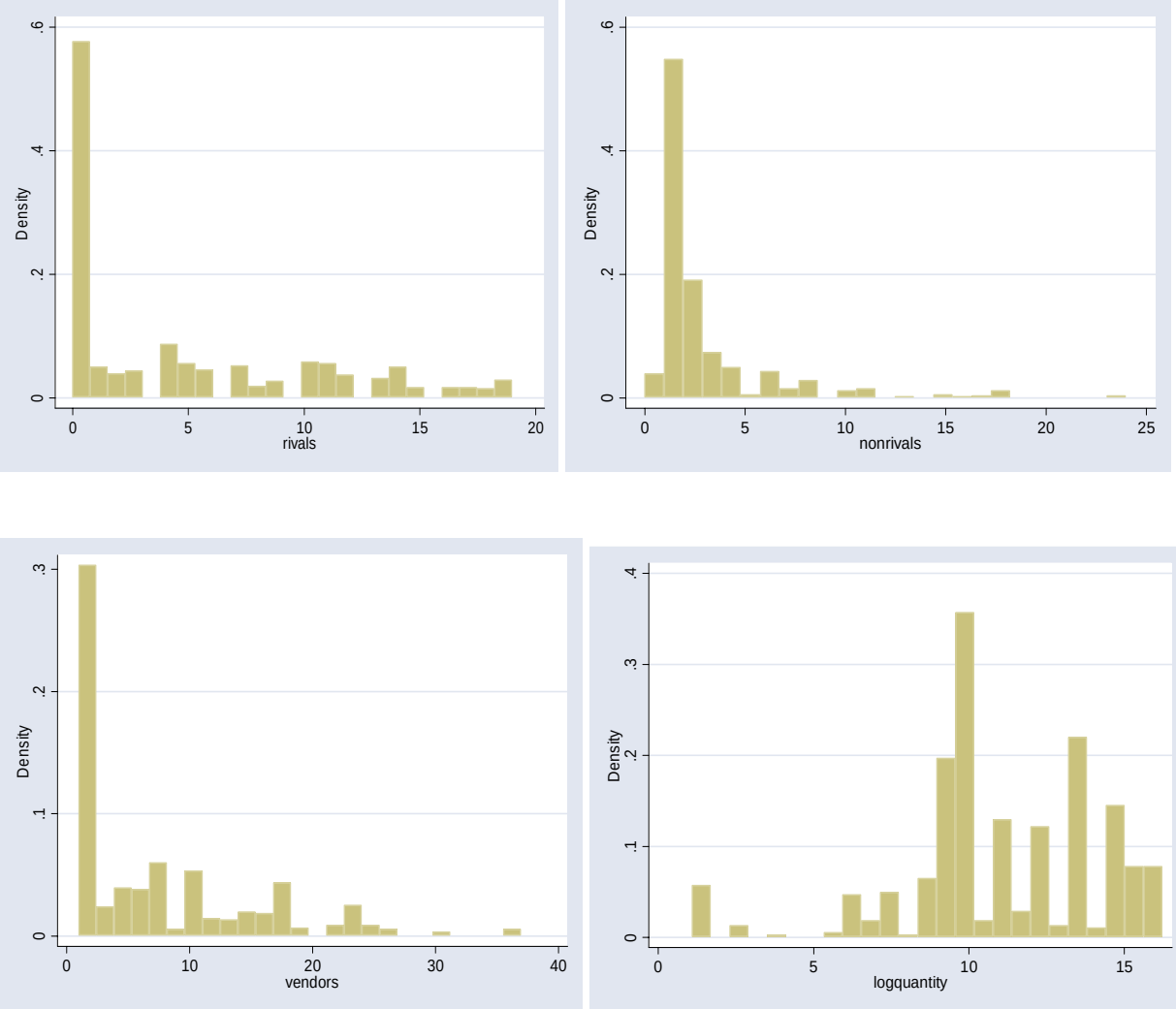
$$\tilde{V}^i_{iq} \equiv \frac{\partial \tilde{V}_i}{\partial \tau_i \partial q_i} = (\lambda_i + \omega_i)\frac{\partial \tilde{\theta}_i}{\partial \tau_i} > 0$$

$$\frac{\partial \tau^*_i}{\partial q} = -\frac{\tilde{V}^i_{iq}}{\tilde{V}^i_{ii}} < 0$$

Appendix II - Table 1: Additional Descriptive statistics

Variable	*N*	*Proportion*
Distribution of Markets		
Proportion of Anti-Virus vulnerabilities	5	1.08
Proportion of Application Development vulnerabilities	3	0.65
Proportion of Application Server Software vulnerabilities	12	2.60
Proportion of Backup And Recovery vulnerabilities	1	0.22
Proportion of Data Base Management vulnerabilities	11	2.39
Proportion of Electronic Mail vulnerabilities	4	0.87
Proportion of Groupware Software vulnerabilities	11	2.39
Proportion of LAN Operating System vulnerabilities	2	0.43
Proportion of Operating System vulnerabilities	365	79.18
Proportion of Suites vulnerabilities	4	0.87
Proportion of System Utilities vulnerabilities	1	0.22
Proportion of System/Software Management vulnerabilities	1	0.22
Proportion of Web Browser vulnerabilities	26	5.64
Proportion of Web Design Tools vulnerabilities	1	0.22
Proportion of Web Development Tools vulnerabilities	4	0.87
Proportion of Web Server Software vulnerabilities	11	2.39
Distribution of Vendors		
Proportion Apple	26	5.64
Proportion Hewlett Packard (includes DEC)	45	9.76
Proportion IBM (includes Lotus & iPlanet)	39	8.45
Proportion Microsoft	72	15.62
Proportion SCO	55	11.93
Proportion Sun Microsystems	43	9.33
Proportion Redhat	60	13.02
Distribution of Disclosure Types		
Proportion of vulnerabilities identified by CERT	18	3.90
Proportion of vulnerabilities identified by University	27	5.86
Proportion of vulnerabilities identified by Consulting company	147	33.41
Proportion of vulnerabilities identified by end user	35	7.59
Proportion of vulnerabilities identified by Vendor	85	18.44
Proportion of vulnerabilities identified by individual	142	30.80
Vulnerability statistics		
Total Vulnerabilities	241	
Vulnerabilities that have no other vendor affected by the same vulnerability	110	
Vulnerabilities that have a private vendor as the *LEADER*	155	
Vulnerabilities with non-private firms as *LEADERS*	54	
Vulnerabilities that have more than 1 *LEADER*	4	
Unpatched observations (vuln.-vendor pairs)	20	
Unpatched vulnerabilities	14	

Histogram of *RIVALS, NONRIVALS, VENDORS* and *LOGQUANTITY*



Appendix III:

Weighting establishment employees using Census county Business patterns (CBP) data:

To obtain a representative sample for the number of employees in establishments we weighted the number of employees in establishments, using a weight that was calculated by comparing the number of census employees for a 3-digit NAICS code with the number of employees for the 3-digit NAICS code in our sample. If *i* represents an establishment that had a positive quantity of the vulnerable product and *HH total employees* denotes the number of employees in the CI database, the weight for the employees in establishments was calculated as:

$$\text{Weight}_i = \left[\frac{\text{Census total employees} - \text{NAICS total (Census)}}{\text{Census total employees}}\right] \times \left[\frac{\text{HH total employees.}}{\text{HH total employees} - \text{NAICS total (HH)}}\right]$$

$$\text{Weighted\_employees}_i = \text{Weight}_i * \text{employees}_i$$

If *p* the vulnerable product for a vendor and Z_{ip} an indicator variable that takes a value of 1 if the establishment had positive quantity of the product, our weighted measure that proxies for quantity (*QUANTITY*), is given by

$$QUANTITY_i = \sum_p Z_{ip} * \text{weighted employees}_i$$

Adjusting the number of establishments and quantity using these weights enables us to account for an over-sampling or under-sampling of specific industries in the HH data.