EMERGING ISSUES IN RESPONSIBLE VULNERABILITY DISCLOSURE[1]

Hasan Cavusoglu
University of British Columbia
Sauder School of Business
Vancouver, BC CANADA
cavusoglu@sauder.ubc.ca

Huseyin Cavusoglu
Tulane University
A. B. Freeman School of Business
New Orleans, LA USA
huseyin@tulane.edu

Srinivasan Raghunathan
University of Texas at Dallas
School of Management
Richardson, TX USA
sraghu@utdallas.edu

Abstract

Security vulnerability in software is the primary reason for security breaches, and an important challenge for IT professionals is how to manage the disclosure of vulnerability information. The IT security community has proposed several disclosure policies, such as full vendor, immediate public and hybrid, and has debated which of these should be adopted by coordinating agencies such as CERT. Our early study (Cavusoglu et al. 2004a) analyzed the optimal disclosure policy that minimizes social loss when vulnerability affects only one software vendor. In this paper, we extend our early work into three directions in order to sled light on current issues in vulnerability disclosure process. (i) When the vulnerability affects multiple vendors, we show that the coordinator's optimal policy cannot ensure that every vendor will release a patch. However, when the optimal policy does elicit a patch from each vendor, we show that the coordinator's grace period in the multiple vendor case falls between the grace periods that it would set individually for the vendors in the single vendor case. (ii) We analyze the impact of an early discovery, which can be encouraged with proper incentive mechanisms, on the release time of the patch, the grace period, and the social welfare. (iii) We also investigate the impact of an early warning system that provides privileged vulnerability information to selected users before the release of a patch for the vulnerability on the social welfare. Finally, we explore the several policy implications of our results and their relationship with current disclosure practices.

[1] An early version of this paper (Cavusoglu et al. 2004b) was presented at Workshop on Information Technology and Systems (WITS'2004) where it has received best paper nomination.

1. Introduction

The rate of IT security breaches has been increasing significantly. The main reason for this increase is security vulnerabilities in software.[2] Hackers exploit software flaws to cause serious damage to firms, including blocking system resources to authorized users, modifying and corrupting sensitive information, and launching attacks on other organizations from victim systems. The sophistication of attack tools and the interconnected nature of the Internet enable hackers to exploit vulnerabilities on a large scale in a short time.[3] Further, wide availability of these tools on the Internet eliminates the need for specialized knowledge and expertise to exploit vulnerabilities, making even novice users capable of launching attacks on vulnerable systems.

Many software vendors have recently started paying more attention to secure software engineering practices.[4] However, secure software with zero vulnerability is unlikely. When vulnerabilities are identified, software vendors often release patches to fix them. Although some software vulnerabilities are first identified by malicious users or hackers, most of them are discovered by benign users. Because benign users do not exploit vulnerabilities and, in many cases, want to prevent hackers from exploiting them, the question of how benign users should disseminate vulnerability knowledge to others has become a keenly debated issue in IT security (SBQ 2002). The questions raised in the debate include: Should the vulnerability be kept secret, (i.e., not announced to the public at large) until the vendor releases a patch? Or should the public be informed about the vulnerability immediately after its discovery? Or should there be some other mechanisms to disclose vulnerability information to the public?

Responsible vulnerability disclosure addresses how a vulnerability identifier should disclose vulnerability information to appropriate people, at appropriate times, and through appropriate channels in order to minimize the social loss associated with vulnerabilities. Despite the consensus on the objective of responsible vulnerability disclosure, the perception about what constitutes responsible vulnerability disclosure has changed over time. During the early days of software development, the common practice was to inform only the vendor about the discovered vulnerability and to keep it secret from the public until the vendor developed a patch was the

---

[2] Microsoft defines security vulnerability as "*a flaw in a product that makes it infeasible- even when using the product properly- to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, or assuming ungranted trust*" (Culp 2000).

[3] Code Red worm infected 20,000 systems within 10 minutes. The Slammer worm attacked more than 90 percent of vulnerable systems within 10 minutes (Dacey 2003).

[4] In 2002, Microsoft took an unprecedented step of ceasing development of new Windows operating system software for the entire month, and sending the company's 7,000 systems programmers to a special security training program (CBS News 2002).

common practice. This process is called *full vendor disclosure*. Because full vendor disclosure gives the control of handling a vulnerability to the vendor and cannot force the vendor to develop a fix, some reported vulnerabilities have gone unfixed or have been fixed after a long delay (Schneier 2001). It has become clear that unless the vendor is committed to develop a patch as soon as possible, full vendor disclosure may put the public at risk. The tardiness of vendors has caused some users to adopt *immediate public disclosure*, which releases vulnerability knowledge immediately to the public after its identification. Unlike full vendor disclosure, immediate public disclosure gives vendors a strong incentive to fix the vulnerability as soon as possible to prevent a public embarrassment (Pond 2000). Moreover, public disclosure of the vulnerability information allows vulnerable firms to take some intermediate measures to reduce the risk of exploitations until the vendor releases a patch. The opponents of immediate public disclosure point out that disseminating vulnerability knowledge to the public does not improve overall security: If there is no patch to fix the vulnerability, hackers have an opportunity to develop exploits and attack vulnerable systems before the vendor releases a patch to address the problem. Although immediate public disclosure might provide necessary motivations to vendors that might otherwise ignore the vulnerability otherwise, it also punishes other vendors that would make an honest effort to deliver the patch promptly by not providing them adequate time to address the vulnerability. Because each of these policies, full vendor and immediate public disclosure, addresses the management of vulnerability knowledge only from one stakeholder's perspective, the security community has recently realized a need for a middle ground in which both the vendor and benign user can compromise in the vulnerability disclosure process to make the process better for society (Stone 2003). In this new disclosure process, also known as *hybrid disclosure*, the benign user does not announce the vulnerability knowledge to the public immediately, but instead allows the vendor some time to develop a patch. If the vendor does not release its patch before the deadline, the public is informed about the vulnerability. It is argued that with this approach, neither a benign user nor a vendor can accuse the other of being the irresponsible party in handling the process of the vulnerability disclosure (Stone 2003), and the vendor is given motivation without jeopardizing the security of firms using the vulnerable software (Schiller 2002). The Computer Emergency Response Team/Coordination Center (CERT/CC) has emerged as a third-party coordinator to handle the hybrid vulnerability disclosure process. CERT/CC acts as an intermediary between vulnerability identifiers and vendors. An identifier first releases the vulnerability knowledge to CERT/CC, which then verifies the vulnerability and informs the affected vendor(s) about the vulnerability. In

addition, it sets a 45-day grace period for the vendor to come up with a patch. CERT/CC discloses information about vulnerabilities to the public after the deadline regardless of the availability of patches from affected vendors (CERT 2000). However, the benign user can also reveal the vulnerability knowledge directly to the vendor. The vendor and the identifier then follow a fixed set of procedures before the vulnerability is disclosed to the public. In addition to the CERT/CC guidelines, a number of guidelines are currently available to govern this direct relationship between the vendor and the identifier, such as *Guidelines for Vulnerability Reporting and Response* (OIS 2004) by Organization for Internet Safety (OIS).[5] Under this type of hybrid disclosure, the coordinator's role is limited to resolving any disputes between the vendor and vulnerability identifier. Security consulting firms are also actively involved in the vulnerability identification process. They usually have their own guidelines for responsible vulnerability disclosure, and work directly with software vendors. These guidelines typically follow either full vendor disclosure or hybrid disclosure.[6]

The multitude of disclosure mechanisms characterized as responsible vulnerability disclosure creates chaos and confusion on the vendor side (Shepherd 2003). Since the US government has emphasized the criticality of a pre-designed responsible vulnerability disclosure process (Chambers and Thompson 2004), and software vendors and security research firms have begun to jointly develop a unified framework for vulnerability disclosure (OIS 2004), it is expected that those efforts will soon converge. Yet, it is not clear what constitutes the responsible vulnerability disclosure process. Cavusoglu et al. (2004a) analyzed the impact of vulnerability disclosure mechanisms on the decisions of various stakeholders and developed policy guidelines for the management of vulnerability knowledge based on the effect of these disclosure mechanisms on social welfare. We found that the one-size-fits-all solution for software vulnerability disclosure is not right (Shepherd 2003) and specified conditions under which full vendor disclosure, immediate public disclosure or hybrid disclosure qualifies as the responsible vulnerability disclosure process. For cases where the hybrid approach is optimal, we determined how much time the vendor should be given to develop a patch. Arora et al. (2004) also attempted to determine the optimal timing of software vulnerability

[5] OIS's guidelines set a grace period of 30 days, though it leaves the door open to determine the grace period on a case-by-case basis.

[6] For example, eEye Digital Security does not disclose any information to third parties until the manufacturer releases a patch although it sets a 60-day target period (eEyes 2004). Across Security keeps the details of the vulnerability secret until the vendor releases a patch if the vendor is responsive. Otherwise it decides when to release on a case by case basis (Across 2004). BindView follows OIS guidelines for vulnerability disclosure (BindView 2003). CYBSEC gives 45 days to the vendor before issuing an advisory to inform the public (CYBSEC 2004).

disclosure. Analyzing software vulnerability disclosure process is extremely timely as the US Department of Homeland Security reviews vulnerability disclosure mechanisms and considers mandating a centralized vulnerability disclosure policy (Fisher 2003).

In this paper, we extend our early work (Cavusoglu et al. 2004a) to investigate three critical issues. First, we focus on vulnerability disclosure process when the same vulnerability affects multiple software vendors. This is very important given that (i) many software vendors base their proprietary software on open-source codes and (ii) a flaw in a common protocol may lead to vulnerabilities in several software applications. A casual observation of vulnerability databases such as US-CERT Vulnerability Note Database readily confirms that many vulnerabilities indeed impact products from more than one vendor. Since early studies focused on vulnerability disclosure process when vulnerability affects only one software vendor, there is a need for analyzing vulnerability disclosure process when vulnerability affects more than one vendor. Specifically, we answer the question of whether the coordinator should give a longer or shorter grace period when there are multiple vendors affected by the vulnerability compared to when there is only one vendor affected by the same vulnerability. This provides a guideline to the coordinator dealing with multiple vendors affected by the same vulnerability as to whether it sets a disclosure policy considering the vendor that will patch its product the latest or the earliest. Second, we analyze the impact of an early discovery on the release time of the patch, the grace period, and the social welfare. If an early discovery benefits the society as a whole, this helps justify incentive mechanisms that lead to an early discovery of the vulnerability. This may also provide a motivation for the coordinator to design appropriate incentive mechanism to induce an early discovery. Third, we study a controversial question of whether an early warning system that provides privileged vulnerability information to selected users before the release of a patch for the vulnerability would improve the social welfare. Recently, CERT started to inform members of Internet Security Alliance (ISA) about newly discovered vulnerabilities right after informing the vendor. Although sharing vulnerability information with a limited number of high profile organizations which control, facilitate, or enable and/or rely on critical infrastructure might reduce possible damages to critical infrastructure, it is not clear if the society as a whole is better off with the early warning system. In a blaze of recent publicity over this contentious practice, our research sheds light on social welfare implications of an early warning system which provide privileged vulnerability information to a limited set of users.

The rest of the paper is organized as follows. In the next section, we review the relevant literature. In section 3, the basic model adopted from Cavusoglu et al. (2004a) is presented and their

findings on vulnerability disclosure for the single vendor case are summarized. In section 4, we focus on the multiple vendor case and contrast it with the single vendor case. We analyze early discovery in section 5 and early warning in section 6. Policy implications, limitations, and future work are discussed in the subsequent section. Finally, we end the paper with a summary of our results.

2. Literature Review

The information systems community recognized the significance of IT security more than a decade ago. In one of the earliest papers to focus exclusively on IT security, Straub (1990) discussed the importance of deterrence in reducing computer abuse. Since then, planning for and management of IT security has been a focus of IS literature (Niederman et al. 1991; Loch et al. 1992; Straub and Welke 1998). This literature has considered design of deterrent, preventive, and detection measures – collectively self-protection – and disaster recovery measures to enable an organization to control security risk by reducing both the probability and the severity of loss. Most of the research in IT security management has been primarily qualitative in nature and is based on case studies and conceptual frameworks. In response to questions raised by firms about the return on security investments (ROSI) in the wake of increasing scale and scope of security breaches and increasing IT security budgets (Hulme 2002), researchers have recently initiated work on the economic aspects of IT security. The research in this stream investigates issues such as valuation of security technologies (Cavusoglu et al. 2002; Cavusoglu et al. 2003), assessment of costs from security breaches (Cavusoglu et al. 2004c), and optimal level of IT security investments (Cavusoglu et al. 2004d). None of the above-mentioned work addresses any public policy aspect of IT security through security knowledge management, though IT security is considered a national security issue.

Despite serious discussion among security professionals, to the best of our knowledge, only few academic studies investigate vulnerability disclosure policies from social, organizational, or economics perspectives. These studies generally describe current practices on vulnerability discovery and disclosure process. Laakso et al. (1999) describe the vulnerability handling process with regard to its three main actors, the originator (i.e., identifier of the vulnerability), the coordinator (i.e., CERT), and the repairer (i.e., vendor). They propose a life-cycle model that requires cooperation among these actors for an effective vulnerability handling process. Using this model as a foundation, Havana (2003) analyzes the communication in the software vulnerability reporting process between identifiers/coordinators and vendors. This survey study found the current reporting structure to be ill-defined and in need of improvements in knowledge management,

organizational learning, and ethics and trust relationships between identifiers and vendors. It also revealed that both groups are generally opposed to the immediate public and full vendor disclosure. In their case study analysis, Arbaugh et al. (2000) test the vulnerability life-cycle model using incident data from CERT/CC about three common vulnerabilities. Takanen et al. (2004) identify ethical responsibilities of stakeholders in the vulnerability disclosure process. Due to the gamut of alternative disclosure policies developed to cater to various stakeholders and their diverse interests, it has been argued that standardization of vulnerability disclosure processes is essential (Shepherd 2003). Despite some discussions on governmental regulation (Tain 2002; Palella 2003), National Infrastructure Advisory Board (NIAC), which shapes the US government's information security policies, chose to encourage non-governmental means to establish standardization in vulnerability disclosure practices. Preston and Lofton (2002) approached the vulnerability disclosure from a legal perspective. They argued how and when the disclosure of information about software vulnerabilities can be protected under the first amendment. They also drew upon the issue of liability that may arise as a result of vulnerability disclosure.

Recently, contemporary studies by Cavusoglu et al. (2004a) and Arora et al. (2004) simultaneously addressed the issue of how to disclose vulnerability information to public, assuming that a given vulnerability affects only a single vendor. Arora et al. (2004) find that the vendor always releases the patch after the end of the period set by the coordinator. Even with a casual observation, the main result of Arora et al. (2004) can be nullified. In really, patches almost never come after the public disclosure by the coordinator. Unlike Arora et al. (2004), Cavusoglu et al. (2004a) give a complete characterization of all available disclosure policies. Arora et al. (2004) make two critical assumptions which may explain why they find that the vendor always releases the patch after the end of the period set by the coordinator. First, they ignore the possibility that vulnerability can be exploited by hackers before a benign user discovers it. Second, they don't consider the increased risk of exploitations after the public disclosure. The former assumption implies that hackers only are able to discover the vulnerability after a benign user discovers it, and the latter assumption entails that public disclosure does not affect the exploitation rate. Therefore, they reach a conclusion that the coordinator's grace period is never efficient in terms of forcing the vendor to release a patch before the end of the grace period. In this paper, unlike prior literature on vulnerability disclosure policies, we address how the vulnerability information should be disclosed when the vulnerability affects more than one vendor. We also question the effect of early discovery of the vulnerability and early warning to selective users, on responsible disclosure policy and

resulting social welfare.

Kannan and Telang (2004) look at the competition between benign users and hackers in the vulnerability discovery process. They analyze the impact of monetary incentives provided by a profit-seeking organization on the vulnerability identification. Assuming that monetary incentives increase benign users' effort to find vulnerabilities, they model the competition between hackers and benign users in discovering vulnerabilities and the resulting social welfare. They found that a non-profit-based vulnerability discovery process such as CERT's is almost always better than a profit-seeking discovery process such as IDefense's. Their analysis implicitly ignores the vulnerability disclosure process to focus on the vulnerability discovery process. However, given that a CERT-type mechanism is superior in terms of social welfare, their results provide necessary motivations for the main focus of this study, which is how to disclose a vulnerability identified by a benign user to maximize the social welfare.

While research on software vulnerability disclosure is limited, researchers in the economics and accounting areas have investigated financial disclosure for a long time. Using disclosure as a vehicle, they have tried to establish a link between financial reporting and economic consequences (Grossman and Stiglitz 1980; Demski and Feltham 1994; Fischer and Verrecchia 1999). This stream of research also examines why firms sometimes disclose, sometimes withhold private information whose dissemination is not mandatory (Verrecchia 1983; Dye 1990; Clinch and Verrecchia 1997) and which disclosure arrangements are preferred in the absence of prior knowledge (Verrecchia 1982; Lundholm 1991; Bushman 1991). However, disclosure in an accounting context is different from software vulnerability disclosure. First, unlike vulnerability disclosure, the timing of financial disclosure is generally known to public in advance (e.g., annual reports, quarterly statements). Second, in financial disclosure, the source of information revealed is the firm, yet it is the third party that has discovered the vulnerability information in software vulnerability disclosure. Third, even if the disclosure is mandatory in non-discretionary financial disclosure, information is not revealed to the controlling authority before it is disseminated to the public, unlike software vulnerability disclosure.

3. Basic Model Adopted from Cavusoglu et al. (2004a)

We first describe the basic model of Cavusoglu et al. (2004a) and then summarize their results regarding optimal disclosure policy when the vulnerability affects only a single vendor. There are four stakeholders in the vulnerability disclosure process: software developer (vendor), software deployers (firms), vulnerability identifier (benign user or hacker), and central coordinator (CERT).

We analyze how responsible vulnerability disclosure should take place to minimize the social loss.[7] We assume that the software with a security-related vulnerability is introduced to the market at time zero. Both benign users and hackers may discover the vulnerability. When a benign user discovers the vulnerability at time t_0, he/she discloses this information to the coordinator immediately. The coordinator determines the risk associated with the vulnerability and decides on its disclosure policy. Some vulnerabilities are more likely to be exploited, especially those for which automated attack tools are readily available. We define δ to represent the likelihood of successful exploitation of the vulnerability. The coordinator notifies the vendor about the vulnerability and sets a grace period T. The vendor then decides on the time to release its patch p (hereafter, patch release time). If the vendor releases a patch before the end of the grace period (i.e., $p \le t_0 + T$) the coordinator or the vendor publicly announces the vulnerability along with the patch at time p. However, if the vendor does not come up with a patch before the end of the grace period (i.e., $p > t_0 + T$) the coordinator discloses the vulnerability to the public without any patch for it at time $t_0 + T$. If a patch is not available when the vulnerability is publicly announced, vulnerable firms try to find a quick fix (workaround) that reduces the risk of exploitation by hackers, such as disabling services associated with the vulnerable software, reconfiguring systems to change the flow of information to bypass the vulnerable software, or monitoring the vulnerable systems extensively. Firms incur a cost of s per unit time (workaround cost) until the vendor releases a patch. This effort reduces the likelihood that hackers exploit the vulnerability to $\delta\gamma$, where γ $(0 \le \gamma \le 1)$ is the inefficiency of the workaround.

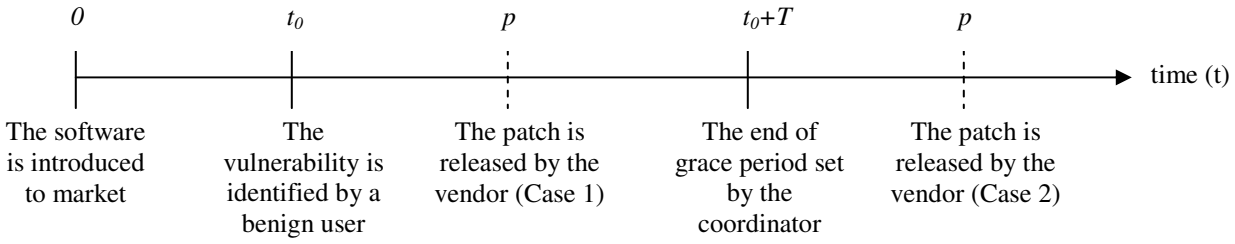

Figure 1: The Timeline for the Vulnerability Discovery and Disclosure Process

Hackers can also discover the vulnerability before the public becomes aware of it. We assume that hackers identify the vulnerability for the first time at time y. In our model, discovery of the vulnerability by hackers is a stochastic process with an instantaneous discovery rate α. After a

[7] Although we consider a central coordinator in the responsible vulnerability disclosure process, as we mentioned in the introduction, this process can work without any coordinator.

hacker discovers the vulnerability, vulnerable firms are attacked at a rate of a. We assume that rate of attacks increases to ka, $k>1$, after the vulnerability is announced to the public. We assume that N firms are using the vulnerable software and are under the risk of attack. The timeline for the vulnerability discovery /disclosure process is depicted in Figure 1.

3.1. The Vendor's Problem

The vendor incurs two types of cost in our model. First, it incurs a patch development cost to fix the vulnerability. This cost depends on the patch release time. We denote the development cost by $\varepsilon_1-\varepsilon_2(p-t_0)$, where ε_1 represents the cost of instantaneous patch development and ε_2 characterizes savings in patch development cost per unit time associated with delaying the release.[8] Second, the vendor incurs a cost in terms of reputation loss. This cost includes the loss in future sales because of a reduction in perceived quality of software and users' trust in software vendor. We use β to denote the expected reputation cost to the vendor per affected firm.

The vendor decides when to release a patch. If the vendor chooses to release the patch before the deadline (i.e., $p\le t_0+T$) then its cost is

$$V=\varepsilon_1-\varepsilon_2(p-t_0)+\beta\left[\int_0^p\int_y^p \alpha N\delta a\,dtdy\right]=\varepsilon_1-\varepsilon_2(p-t_0)+\frac{\alpha\beta N\delta a p^2}{2}\ . \tag{1}$$

If the vendor chooses to release the patch after the deadline (i.e., $p>t_0+T$) its cost is

$$\begin{aligned} V&=\varepsilon_1-\varepsilon_2(p-t_0)+\beta\left[\int_0^{t_0+T}\int_y^{t_0+T} \alpha N\delta a\,dtdy+\int_{t_0+T}^{p}\delta\gamma Nka\,dt\right]\\ &=\varepsilon_1-\varepsilon_2(p-t_0)+\beta\left[\frac{\alpha N\delta a(t_0+T)^2}{2}+\gamma\delta Nka(p-t_0-T)\right]. \end{aligned} \tag{2}$$

3.2. The Coordinator's Problem

The coordinator determines the optimal grace period for a patch release to minimize the sum of the vendor's patch development cost and the damage costs incurred by affected firms. Note that the reputation cost of vendors is simply a transfer of wealth between firms and vendors, and hence does not affect the social welfare. When hackers exploit the vulnerability, the firm incurs a damage cost $D\theta$, where θ, $\theta\sim U(0,1)$, represents the type of the firm and D is the maximum amount of

[8] We assume that $\varepsilon_1\gg\varepsilon_2$ so that the vendor always incurs a positive patch development cost if it decides to release a patch.

damage. If the vendor releases the patch before the end of the grace period (i.e., $p \le t_0 + T$) the social cost is

$$C = \varepsilon_1 - \varepsilon_2(p - t_0) + \int_0^p \int_y^p \int_0^1 \alpha N \delta D \theta a \, d\theta dt dy = \varepsilon_1 - \varepsilon_2(p - t_0) + \frac{\alpha N \delta D a p^2}{4}. \quad (3)$$

If the vendor releases the patch after the end of the grace period ($p > t_0 + T$), the social cost is

$$\begin{aligned} C &= \varepsilon_1 - \varepsilon_2(p - t_0) + \int_0^{t_0+T} \int_y^{t_0+T} \int_0^1 \alpha N \delta D \theta a \, d\theta dt dy + \int_{t_0+T}^p \int_0^1 \delta \gamma N D \theta k a \, d\theta dt + \int_{t_0+T}^p N s \, dt \\ &= \varepsilon_1 - \varepsilon_2(p - t_0) + \frac{\alpha N \delta D a (t_0 + T)^2}{4} + \frac{\gamma N \delta D k a}{2}(p - t_0 - T) + N s (p - t_0 - T). \end{aligned} \quad (4)$$

Cavusoglu et al. (2004a) first solved how the vendor would react to a given T set by the coordinator. Then, given the vendor's reaction, they solve for the optimal T in the coordinator's problem for responsible vulnerability disclosure. Their results are summarized in Table 1.

Table 1: Optimal Disclosure Policy and Patch Release Time in the Single Vendor Case

Equilibrium	Condition(s)	The coordinator's Disclosure Policy (T^*)	The Vendor's Patch Release Time (p^*)
S1	(i) $\gamma k < \Omega$	∞	$\max(\Omega/\alpha, t_0)$
S2	(i) $\gamma k > \Omega$ (ii) $t_0 > \Omega/\alpha$	Irrelevant	t_0
S3	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega/\alpha$ (iii) $\Lambda > \Omega/\alpha$	Any $T \in [(\Omega/\alpha) - t_0, \infty)$	Ω/α
S4	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega/\alpha$ (iii) $\Omega/\alpha \ge \Lambda > t_0$	$\Lambda - t_0$	$t_0 + T^* = \Lambda$
S5	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega/\alpha$ (iii) $\Lambda \le t_0$	0	$t_0 + T^* = t_0$

Cavusoglu et al. (2004a) find that responsible vulnerability disclosure may require choosing not to disclose the vulnerability knowledge to the public (S1), that is, giving the full control of the patch release decision to the vendor. This result presents counterintuitive evidence to those who argue that setting a grace period always forces the vendor to develop its patch. Indeed, not setting any grace period may encourage the release of the patch in cases where the public disclosure of vulnerability does not provide necessary motivation to the vendor to develop its patch.

When $\gamma k > \Omega$, the conflicting objectives of the vendor and the coordinator affect the exact patch release time. If the marginal benefit of delaying the patch release for the vendor is less than the marginal cost of delaying at the time of discovery of the vulnerability ($\Omega/\alpha < t_0$), then the vendor chooses to patch immediately after it is informed about the vulnerability (S2). In this case, the public announcement of the vulnerability (through the release of the patch) is controlled by the vendor. Therefore, the coordinator's optimal disclosure policy is irrelevant. When the vendor releases the patch, but does not have an incentive to do so immediately after it is informed about the vulnerability, the coordinator's disclosure policy affects the vendor's decision and the social cost (S3, S4, and S5). In this case, the coordinator minimizes the social cost by controlling the timing of the vendor's patch release.

Λ/τ represents the *marginal benefit-to-cost ratio of the coordinator* to allow additional time to the vendor at time τ, *where* $t_0 \le \tau \le t_0 + T$. If this ratio is less than one when the vulnerability is identified (i.e., $\Lambda < t_0$), the coordinator chooses the full public disclosure policy (i.e., $T^* = 0$). In this case, the vendor also opts to release the patch immediately because not releasing the patch immediately hurts the vendor (S5). Unlike in S2, the vendor is forced to release the patch in S5. When Λ/t_0 is greater than one, then, the timing of the patch release is controlled by the entity, vendor or coordinator, that is more impatient for the release of the patch. If the vendor is more impatient than the coordinator, then the coordinator will set a grace period that is not shorter than what the vendor will require to develop the patch (S3). If the coordinator is more impatient than the vendor, then the coordinator will set a grace period such that the vendor releases the patch at the time preferred by the coordinator (S4).

4. Analysis of the Multiple Vendor Case

Cavusoglu et al. (2004a) and Arora et al. (2004) assumed that the vulnerability affects only a single vendor. However, the same vulnerability can be associated with products of more than one vendor. For example, the vulnerability in an underlying standard poses a risk to products of different vendors that use the same standard. A similar case occurs when a vulnerability is discovered in open-source software, which is incorporated into more than one software product. Since vendors may face different cost structures and serve diverse customer bases, they may have different preferences as to when to release a patch to address the vulnerability. At the same time, the patch release time decision of one vendor can put other vendors at a disadvantage. The release of a patch from a vendor can create a negative externality for the vendors whose products are still exposed to

the common vulnerability, because releasing a patch makes the vulnerability public. Hence, vendors must take into account not only the disclosure policy set by the coordinator but also the patch release decisions made by other vendors when deciding on a patch release time. The coordinator must consider these interactions among vendors while deciding on a disclosure policy. The issue is whether the coordinator should give a longer or shorter grace period when there are multiple vendors affected by the vulnerability compared to when there is only one vendor affected by the same vulnerability. Should it set a disclosure policy considering the vendor that will patch its product the latest or the earliest? Setting a longer grace period in a multiple vendor case may induce some vendors to choose a patch release time later than what they would choose otherwise, causing additional risk to their customers. However, setting a shorter grace period can leave customers of some vendors without a patch, leading to increased exposure to hacker attacks (Schiller 2002).

We extend our single vendor model (Cavusoglu et al. 2004a) to incorporate multiple vendors. We assume that the common vulnerability affects software from two different vendors. We order vendors based on their decisions of patch release time. We assume that vendor 1 releases its patch before vendor 2.[9] Vendor *i,* where $i \in \{1,2\}$, incurs patch development cost of $\varepsilon_{1i} - \varepsilon_{2i}(p_i - t_0)$ and reputation cost of β_i for each affected customer using its product. N_i represents the total number of customers who are using vendor *i*'s product. We signify patch release times with p_is and the coordinator's disclosure policy with T_m. All other parameters remain the same as in the single vendor model summarized in section 3.

Two cases can occur in the multiple-vendor model. In the first case, vendor 1 releases its patch before the end of the grace period allowed by the coordinator. In this case, vendor 1's decision determines when the public becomes aware of the common vulnerability present in products of both vendors. In the second case, vendor 1 releases its patch after the end of the grace period set by the coordinator. Here, the coordinator determines when the public becomes aware of the common vulnerability. As vendor 1 releases its patch after the public disclosure by the coordinator, vendor 1

[9] Thus, we assume that $\Omega_1 < \Omega_2$ holds, where $\Omega_i = \varepsilon_{2i} / \beta_i N_i \delta a$. Note that Ω_i / α is vendor *i* th's tolerance level to delay the release of a patch. Therefore, the vendor with a higher value of Ω_i prefers to wait more before releasing its patch.

does not affect the customers of vendor 2 directly, unlike in the first case.[10] Analysis of these cases leads to the following optimal disclosure policy for the coordinator, as shown in table 2.[11]

Table 2: Optimal Disclosure Policy and Patch Release Times in Two-Vendor Case

Equilibrium	Condition(s)	The Coordinator's Disclosure Policy (T_m^*)	Vendor 1's Patch Release Time (p_1^*)	Vendor 2's Patch Release Time (p_2^*)
M1	(i) $\gamma k < \Omega_1$	∞	$\max\left(\Omega_1/\alpha,\, t_0\right)$	never
M2	(i) $\gamma k > \Omega_2$ (ii) $t_0 > \Omega_1/\alpha$	irrelevant	t_0	t_0
M3	(i) $\gamma k > \Omega_2$ (ii) $t_0 \le \Omega_1/\alpha$ (iii) $\bar{\Lambda} > \Omega_1/\alpha$	Any $T_m \in \left[\frac{\Omega_1}{\alpha} - t_0, \infty\right)$	$\frac{\Omega_1}{\alpha}$	$\frac{\Omega_1}{\alpha}$
M4	(i) $\gamma k > \Omega_2$ (ii) $t_0 \le \Omega_1/\alpha$ (iii) $t_0 < \bar{\Lambda} < \Omega_1/\alpha$	$\bar{\Lambda} - t_0$	$t_0 + T_m^* = \bar{\Lambda}$	$t_0 + T_m^* = \bar{\Lambda}$
M5	(i) $\gamma k > \Omega_2$ (ii) $t_0 \le \Omega_1/\alpha$ (iii) $\bar{\Lambda} \le t_0$	0	t_0	t_0
M6	(i) $\gamma k > \Omega_1$ (ii) $\gamma k < \Omega_2$ (iii) $t_0 > \Omega_1/\alpha$	irrelevant	t_0	never
M7	(i) $\gamma k > \Omega_1$ (ii) $\gamma k < \Omega_2$ (iii) $t_0 \le \Omega_1/\alpha$ (iv) $\hat{\Lambda} > \Omega_1/\alpha$	Any $T_m \in \left[\frac{\Omega_1}{\alpha} - t_0, \infty\right)$	$\frac{\Omega_1}{\alpha}$	never
M8	(i) $\gamma k > \Omega_1$ (ii) $\gamma k < \Omega_2$ (iii) $t_0 \le \Omega_1/\alpha$ (iv) $t_0 < \hat{\Lambda} < \Omega_1/\alpha$	$\hat{\Lambda} - t_0$	$t_0 + T_m^* = \hat{\Lambda}$	never
M9	(i) $\gamma k > \Omega_1$ (ii) $\gamma k < \Omega_2$ (iii) $t_0 \le \Omega_1/\alpha$ (iv) $\hat{\Lambda} \le t_0$	0	t_0	never

[10] However, as vendor 1's decision influences the disclosure policy of the coordinator, vendor 1's decision indirectly affects the customer base of vendor 2.

[11] We provide the derivations of our results in the appendix.

where $\hat{\Lambda} = \frac{2\varepsilon_{21} + \gamma\delta N_2 kaD + 2N_2 s}{\alpha(N_1 + N_2)\delta aD}$ and $\bar{\Lambda} = \frac{2\varepsilon_{21} + 2\varepsilon_{22}}{\alpha(N_1 + N_2)\delta aD}$.

As can be seen from Table 2 and Table 1, the equilibria in the multiple vendor case are very similar to those in the single vendor case. As in the single vendor case, we can characterize the equilibriums in the multiple vendor case into three groups depending on the value of the marginal cost to marginal benefit of postponing the patch release for each vendor at the time of public disclosure. The first group (only M1) occurs when the marginal cost is less than the marginal benefit of postponing the patch release for both vendors right after the public disclosure (i.e., $\gamma k < \Omega_1$).[12] In such a situation, the coordinator knows that setting a finite grace period will not induce the release of a patch by either vendor. Therefore, it chooses not to set a finite grace period, which in turn encourages vendor 1 to release its patch at $\max(\Omega_1/\alpha, t_0)$. However, even this is not enough to motivate vendor 2 since the release of a patch by vendor 1 effectively means the public disclosure of the vulnerability knowledge, and vendor 2's marginal benefit is less than the marginal cost of postponing the release at the time when the public becomes aware of the vulnerability. This case is analogous to S1 in the single vendor case.

The second group occurs (M2 through M5) when the marginal cost is greater than the marginal benefit of postponing the patch release for both vendors right after the public disclosure (i.e., $\gamma k > \Omega_2$).[13] In these cases, each vendor releases a patch to fix the common vulnerability. Equilibria M2 through M5 in the multiple vendor case are qualitatively similar to equilibria S2 through S5 in the single vendor case. The only difference is that the coordinator sets its disclosure policy based on its marginal benefit to marginal cost ratio of allowing additional time to vendors ($\bar{\Lambda}/\tau$) and vendor 1's marginal benefit to marginal cost ratio of postponing the patch release ($\Omega_1/\alpha\tau$), where $t_0 \le \tau \le t_0 + T_m$. Unlike the single vendor case, the coordinator's benefit to cost ratio includes costs and benefits associated with both vendors.[14]

The third group occurs (M6 through M9) when the marginal benefit-to-cost ratio of postponing the patch release is less than one for vendor 1 (i.e., $\gamma k > \Omega_1$), and greater than one for vendor 2 (i.e., $\gamma k < \Omega_2$) at the time when the public gets the vulnerability knowledge. These equilibria in the

[12] Recall that we assumed $\Omega_1 < \Omega_2$. Therefore when $\gamma k < \Omega_1$ holds, $\gamma k < \Omega_2$ also holds.

[13] Recall again that when $\gamma k > \Omega_2$ is true, $\gamma k > \Omega_1$ is also true since $\Omega_1 < \Omega_2$.

[14] In the multiple vendor case, the coordinator's marginal benefit is $(\varepsilon_{21} + \varepsilon_{22})$ and its marginal cost at time τ is $\alpha(N_1 + N_2)\delta aD\tau/2$. This gives rise to marginal benefit-to-cost ratio of $\bar{\Lambda}/\tau$.

multiple vendor case are also qualitatively similar to equilibria S2 through S5 in the single vendor case, except in the multiple vendor case vendor 2 does not release any patch at all. Again, the coordinator's disclosure policy is determined by its marginal benefit-to-cost ratio of allowing additional time to vendors ($\hat{\Lambda}/\tau$) and vendor 1's marginal benefit-to-cost ratio of postponing the patch release ($\Omega_1/\alpha\tau$), where $t_0 \le \tau \le t_0 + T_m$. However, since vendor 2 will never release a patch, the coordinator's marginal benefit of allowing additional time to vendors includes savings per unit time from damage cost to customers of vendor 2 (i.e., $\gamma\delta N_2 kaD/2$) and from workaround cost to customers of vendor 2 (i.e., $N_2 s$).

As in the single vendor case, we have equilibria (M2 and M6) in the multiple vendor case in which the disclosure policy is irrelevant. Except for those equilibria, the optimal vulnerability disclosure mechanism for the multiple vendor case is composed of three disclosure policies identified in the literature. M1 represents the *full vendor disclosure* as the coordinator allows the vendors an infinite amount of time to release their patches. M5 and M9 are examples of *immediate public disclosure*. In M3, M4, M7, and M8 the responsible disclosure policy is *hybrid disclosure*.

4.1. Comparison of Disclosure Policies in Single and Multiple Vendor Cases

Proposition 1: *Optimal disclosure policy of the coordinator may not guarantee the release of a patch from both vendors in the multiple vendor case.*

The coordinator's vulnerability disclosure policy ensures that a patch will be eventually released when there is only one affected vendor. Yet, proposition 1 reveals that this may not be achieved when the vulnerability affects more than one vendor. The characteristics of optimal disclosure policy may behave differently when more than one vendor experiences the same vulnerability. Table 3 presents the possible transitions in optimal disclosure policies from the single vendor case to multiple vendor case when an additional vendor is introduced.

Table 3: Possible Transitions among Equilibria When Multiple Vendors are Affected

Equilibrium in the single vendor case	Equilibria in the multiple vendor case (When the vendor in the single vendor case has ***more incentive*** to patch compared to the new vendor introduced)	Equilibria in the multiple vendor case (When the vendor in the single vendor case has ***less incentive*** to patch compared to the new vendor introduced)
S1	M1	M1, M6, M7, M8, M9
S2	M2, M6	M2
S3	M3, M4, M5, M7, M8, M9	M2, M3, M4, M5
S4	M3, M4, M5, M7, M8, M9	M2, M3, M4, M5
S5	M3, M4, M5, M7, M8, M9	M2, M3, M4, M5

Although many different transitions are possible as shown in Table 3, an analysis of these transitions sheds significant light on whether the coordinator's disclosure policy in the multiple vendor case has different influence on vendors' decisions to patch or not, compared to the single vendor case. We summarize this result in the following proposition.

Proposition 2: ***(i)*** *If a vendor in the single vendor case is to share the vulnerability with a vendor that has less incentive than itself, the disclosure policy of the coordinator in the multiple vendor case cannot prevent the original vendor from releasing its patch.*

(ii) *If a vendor that can only be motivated to release a patch under full vendor disclosure mechanism in the single vendor case is to share the vulnerability with a vendor that has more incentive than itself, the disclosure policy of the coordinator in the multiple vendor case forces the original vendor not to release its patch.*

Proposition 2 highlights the effect of optimal disclosure policy of the coordinator on patch release decisions of vendors in single and multiple vendor cases. Next we address how the coordinator's decision on the optimal disclosure policy changes when a disclosure policy is set for more than one vendor. We consider two different scenarios to answer the question of whether the coordinator should allow more time or less time in multiple vendor case compared to single vendor case.[15] In the first scenario, both vendors release their patches. In the second scenario, only one vendor releases its patch.

4.1.1. Scenario 1: When Both Vendors Release Patches

Assume that the coordinator gives a grace period of T_i to vendor i if vendor i is the only vendor affected by the vulnerability and a grace period of T_m to both vendors in the multiple vendor case. The coordinator's marginal benefit-to-cost ratio of allowing additional time to vendor i in the single vendor case is $2\varepsilon_{2i} / \alpha N_i \delta a D \tau$ at time τ. However, its marginal benefit-to-cost ratio of allowing additional time to both vendors i and j in the multiple vendor case is $\left[2\varepsilon_{2i} + 2\varepsilon_{2j}\right] / \left[\alpha(N_i + N_j)\delta a D \tau\right]$ at time τ. These ratios represent the extent to which the coordinator is willing to give more time to the vendor(s). By reconciling this observation with possible transitions given in Table 3, we obtain the following result.

Proposition 3: *Assuming that both vendors have incentives to wait before releasing a patch* $(t_0 \leq \Omega_1 / \alpha < \Omega_2 / \alpha)$ *and release the patch when they are given a finite grace period*

[15] Since our focus is to analyze change in grace periods, we consider cases in which the coordinator sets a finite grace period and the coordinator's policy is relevant (i.e., S3, S4, and S5).

($\gamma k > \Omega_2 > \Omega_1$) *in the multiple vendor case, the grace period determined in the multiple vendor case is not longer (shorter) than the grace period provided in the single vendor case to the vendor with*

(i) *more incentive if* $\frac{\varepsilon_{22}}{N_2} < (>) \frac{\varepsilon_{21}}{N_1}$, and

(ii) *less incentive if* $\frac{\varepsilon_{21}}{N_1} < (>) \frac{\varepsilon_{22}}{N_2}$.

Based on these results, we can conclude that in the multiple vendor case, while setting a grace period, the coordinator does not consider the incentive type of the vendor (i.e., it does not consider how tolerant each vendor is). Having a high- (low-) incentive vendor share the same vulnerability with a low- (high-) incentive vendor does not necessarily lead to a longer (shorter) grace period to accommodate the second vendor. This result is quite interesting given that you would expect the coordinator to consider the incentives of the vendors in setting a grace period. Indeed, whether the coordinator gives a longer grace period is solely determined based on a comparison of savings in the patch development cost of each vendor per its customer (ε_{2i} / N_i). Depending on this comparison, the coordinator will have more incentive or less incentive to extend the grace period compared to the single vendor case when another vendor shares the same vulnerability. If sharing the vulnerability with another vendor makes the coordinator more willing to extend the grace period ($\Lambda < \bar{\Lambda}$), it does not shorten the grace period. Otherwise, it does not extend the grace period. Corollary 1 follows directly from Proposition 3.

Corollary 1: *Assuming that both vendors have incentives to wait before releasing a patch* ($t_0 \le \Omega_1 / \alpha < \Omega_2 / \alpha$) *and release the patch when they are given a finite grace period* ($\gamma k > \Omega_2 > \Omega_1$) *in the multiple vendor case, the grace period in the multiple vendor case is within the range of two grace periods provided to vendors individually when they are the only vendor affected. In other words,* $T_i \le T_m \le T_j$, *where* $i \ne j \in \{1, 2\}$.

This result shows that when setting a policy in the multiple vendor case the coordinator makes a compromise and sets a grace period such that it is equal to or larger than the minimum grace period and equal to or smaller than the maximum grace period that it would set, if the coordinator dealt with vendors individually.

4.1.2. Scenario 2: When Only One Vendor Releases a Patch

In this section, we investigate the impact of having a second vendor, which ignores the vulnerability, on the coordinator's disclosure policy. The question that we address is whether the coordinator extends the grace period provided to the vendor with higher incentive if this vendor shares the same vulnerability with a low-incentive vendor that will never patch its vulnerability in the multiple vendor case.

When the vendor with more incentive is the only vendor in the single vendor case, the coordinator's marginal benefit-to-cost ratio of allowing additional time to the vendor at time τ is $2\varepsilon_{21} / \alpha N_1 \delta a D \tau$. When another vendor which has less incentive and will not release its patch shares the same vulnerability, the coordinator's marginal benefit-to-cost of allowing additional time to both vendors at time τ is $\left[2\varepsilon_{21} + \gamma\delta N_2 kaD + 2N_2 s\right] / \left[\alpha(N_1 + N_2)\delta a D\tau\right]$. A comparison of these ratios reveals how the coordinator's policy changes from the single vendor case if the high-incentive vendor shares its vulnerability with the low-incentive vendor that will never patch.

Proposition 4: *Assuming that only the high-incentive vendor releases a patch in the multiple vendor case when the vendors are given a finite grace period (i.e.* $\Omega_2 > \gamma k > \Omega_1$ *and* $t_0 \le \Omega_1 / \alpha < \Omega_2 / \alpha$*), the grace period in the multiple vendor case is not longer (shorter) than the grace period provided to the vendor with more incentive in the single vendor case if* $\frac{\gamma}{2}\delta kaD + s < (>) \frac{\varepsilon_{21}}{N_1}$.

In Proposition 3, we found that when a high- (low-) incentive vendor shares the same vulnerability with a low- (high-) incentive vendor, the coordinator does not necessarily give a longer (shorter) grace period to accommodate the second vendor given that both vendors release their patches. Similarly, we show in Proposition 4, that when the high-incentive vendor shares the same vulnerability with the low-incentive vendor that ignores the vulnerability, the coordinator does not necessarily give a longer or shorter grace period. The coordinator allows the vendor with more incentive no less than the grace period that it would allow if the vendor is the only vendor affected by the vulnerability only if the patch development cost of vendor 1 per its customer (ε_{21} / N_1) is smaller than the damage and workaround cost per customer of vendor 2 ($(\gamma\delta kaD / 2) + s$). This implies that the coordinator may extend the grace period given to the high-incentive vendor in the multiple vendor case even if the coordinator knows that the low-incentive vendor will not release any patch at all. The reason is that the coordinator minimizes the social loss and considers savings in damage and workaround cost for customers of vendor 2.

5. Early Discovery of Vulnerabilities

There is an inherent relationship between the vulnerability disclosure process and the vulnerability discovery process. In this section, we extend the basic model of Cavusoglu et al. (2004a) to investigate the impact of a change in the vulnerability discovery process on vulnerability disclosure policy of the coordinator and patch release decision of the vendor. Cavusoglu et al. (2004a) assumes that benign users report vulnerabilities to the coordinator without any monetary incentives. However, there might be other incentives that motivate benign users to reveal this information, such as reputation gain as an identifier or a security firm, peer recognition, and favorable press coverage (Ranum 2004). For instance, CERT acknowledges identifiers in its advisories. Similarly, Microsoft Security Bulletins give credits to people or organizations who barter the vulnerability knowledge to Microsoft. Although pure altruism can be a motivation for some identifiers, self-serving interests drive discovery of most vulnerabilities. The social planner may establish a mechanism that stimulates benign users to exert more effort to discover the vulnerabilities earlier. However, early discovery can change the disclosure policy of the coordinator and resulting patch release decision of the vendor. Thus it is not clear if the social welfare improves when vulnerabilities are discovered earlier.

In this section, we assume that some incentives are provided to benign users which lead to an early discovery of vulnerability at time t_0' such that $t_0' < t_0$, where t_0 is the time of the discovery without additional incentives. The following table shows how the optimal disclosure policy changes when the vulnerability is discovered earlier.

Table 4: Changes in Equilibria with Early Discovery

Discovered at t_0	Discovered at $t_0' < t_0$		
Equilibrium	Equilibrium	Disclosure Policy (T'^*)	Patch Release Time (p'^*)
S1	S1	∞	$\max\left(\Omega/\alpha,\, t_0'\right)$
S2	S2	irrelevant	t_0'
	S3	Any $T' \in \left[(\Omega/\alpha) - t_0', \infty\right)$	Ω/α
	S4	$\Lambda - t_0'$	Λ
	S5	0	t_0'
S3	S3	Any $T' \in \left[(\Omega/\alpha) - t_0', \infty\right)$	Ω/α
S4	S4	$\Lambda - t_0'$	Λ
S5	S4	$\Lambda - t_0'$	Λ
	S5	0	t_0'

A comparison between patch release times in each row reveals an interesting result summarized in the following proposition.

Proposition 5: *If the discovery of vulnerability occurs earlier ($t_0' < t_0$), the vendor releases the patch no later than when it would release otherwise (i.e., $p'^* \leq p^*$).*

Proposition 5 reveals that an early discovery does not worsen the exposure window during which the society is susceptible to the vulnerability from its inception. Vulnerabilities are fixed earlier if they are identified earlier. The early disclosure also influences the grace period set by the coordinator. The following result is obtained by comparing the disclosure policies given in Table 1 and Table 4. [16]

Proposition 6: *If the discovery of vulnerability occurs earlier ($t_0' < t_0$), the coordinator does not shorten the grace period (i.e., $T'^* \geq T^*$).*

Since optimal disclosure policy and patch release time change when there is an early discovery, the social welfare might also change as a result. The difference between social welfare when the vulnerability is discovered at t_0 and when it is discovered at t_0' is

$$C|_{t_0} - C|_{t_0'} = \varepsilon_2(p' - p + t_0 - t_0') + \frac{\alpha N \delta D a}{4}(p^2 - p'^2)$$

When the change in welfare is calculated for every pair of equilibria, we find the impact of early discovery on the social welfare, as summarized in the following proposition.

Proposition 7: *The society is always better off with an early discovery of the vulnerability.*

The proposition shows that an early discovery does not degrade the social welfare. The results of propositions 6 and 7 imply that the social welfare can be improved without reducing the grace period if vulnerabilities are discovered earlier. Therefore, the social planner should consider both monetary and non-monetary incentive schemas to get benign users to exert more effort to identify vulnerabilities earlier. For instance, the coordinator may compensate the identifier with a certain portion of the social welfare gain and retain some gain such that even with a compensation scheme society is better off with an early discovery.

6. Early Warning System to Selected Firms

In this section, we investigate an early warning system that provides the vulnerability information to selected software users. Currently CERT informs members of Internet Security Alliance (ISA) about newly discovered vulnerabilities right after informing the vendor. The members of ISA

[16] When there are several grace periods that are optimal, we assume that the coordinator chooses the smallest one.

consist of high profile firms which control, facilitate, or enable critical infrastructure and/or rely on it. From the central planner perspective, it may be compelling to believe that, to alleviate possible damage to critical infrastructure, it would be beneficial to share vulnerability details with these organizations since they may take existing precautions until a fix becomes available from the vendor. However this practice might discourage some vendors to develop a patch promptly, knowing that some affected firms are partially protected. Moreover, an early warning system can cause more damage than benefit if the vulnerability knowledge is leaked prematurely to the public by members that receive the early warning. Even if the leakage of vulnerability knowledge is fully prevented through strict written nondisclosure agreements, it is not clear whether the whole society gains from such a pre-notification mechanism for vulnerability disclosure.

We assume that the coordinator provides early vulnerability knowledge to υ fraction of software users which do not leak or misuse it.[17] When they receive the vulnerability knowledge, they apply available workarounds to reduce their risk of being attacked to $\gamma\delta$. The rest of the analysis remains the same. After solving the game between the coordinator and the vendor, we find the coordinator's disclosure policy and the vendor's patch release time as follows.

Table 5: Optimal Disclosure Policy and Patch Release Time with an Early Warning System

Equilibrium	Conditions	The Coordinator's Disclosure Policy (T^*)	The Vendor's Patching Time (p^*)
V1	(i) $\gamma k < \Omega$	∞	$\max\left(\Omega_\upsilon / \alpha, t_0\right)$
V2	(i) $\gamma k > \Omega$ (ii) $t_0 > \Omega_\upsilon / \alpha$	irrelevant	t_0
V3	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega_\upsilon / \alpha$ (iii) $\Lambda_\upsilon > \Omega_\upsilon / \alpha$	Any $T \in \left[(\Omega_\upsilon / \alpha) - t_0, \infty\right)$	Ω_υ / α
V4	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega_\upsilon / \alpha$ (iii) $\Omega_\upsilon / \alpha \ge \Lambda_\upsilon > t_0$	$\Lambda_\upsilon - t_0$	$t_0 + T^* = \Lambda_\upsilon$
V5	(i) $\gamma k > \Omega$ (ii) $t_0 \le \Omega_\upsilon / \alpha$	0	$t_0 + T^* = t_0$

[17] Not every firm can join this pre-notification service. For example, ISA has strict rules and guidelines that define who can be a member. A firm's willingness to pay necessary membership fees to join to such an alliance cannot provide the firms with access to an early warning service. Through a private communication with Larry Clinton, who is the membership manager of ISA, we learned that "the applicant must be approved by the board of directors which consists of most respected leaders in the cyber security field including former chairman of US Congress Committee on Intelligence and the director of CERT." He also pointed out that "to guard against misuse and leakage of data all ISA members must sign written non-disclosure agreements. We follow one strike and you're out policy."

	(iii) $\Lambda_v \le t_0$		

where $\Lambda_v = (2\varepsilon_2 - 2\upsilon Ns)/\alpha N\delta aD(1-(1-\gamma)\upsilon)$ and $\Omega_v = \varepsilon_2/\beta\delta Na(1-(1-\gamma)\upsilon)$

Note that the general structure of the optimal disclosure policies does not change by offering early warnings to selected firms. Both the grace period and patch release time with an early warning system can be different than those without one. Similarly to the basic model, $\Omega_v/\alpha\tau$ denotes the marginal benefit-to-cost ratio of the vendor to postpone the patch release and Λ_v/τ denotes the marginal benefit-to-cost ratio of the coordinator to allow additional time to the vendor in the early warning system at time τ, where $t_0 \le \tau \le t_0 + T$. The change in these values is summarized below.

Corollary 2: *When the coordinator implements an early warning system to selected users, (i) the vendor has more incentive to postpone the release of its patch and (ii) the coordinator has more (less) incentive to allow more time for the release of the vendor's patch if* $1-\gamma > (<) Ns/\varepsilon_2$.

Those customers who subscribe to such a system will apply workaround to reduce their chance of being attacked. In turn, this leads to a reduction in the vendor's reputation cost. Hence the marginal benefit-to-cost ratio for the vendor to delay the release of its patch increases. On the other hand, the coordinator's incentive to allow additional time to the vendor to develop its patch shows different characteristics. If the workaround is expensive (i.e., $s > (1-\gamma)\varepsilon_2/N$), the coordinator has less incentive to extend the grace period.

Since the vendor's and the coordinator's incentives to postpone the release of the patch can be conflicting, it may not be straightforward to draw conclusions on whether the patch is actually released earlier in the case of an early warning system compared to no early warning system. The following proposition sheds light on this issue.

Proposition 8: *Given that full vendor disclosure is not optimal disclosure policy (i.e.,* $\gamma k > \Omega$*)* ***(i)*** *If the vendor has incentive to release its patch immediately when there is no early warning system, the vendor releases its patch at* $p_v^* \ge p^* = t_0$ *with an early warning system.* ***(ii)*** *If the vendor does not have incentive to release its patch immediately when there is no early warning system, the vendor releases it patch at* $p_v^* > (\le) p^*$ *when* $\Lambda_v > (\le) p^*$ *with an early warning system.*

Proposition 8 shows that the adoption of an early warning system may cause an increase or decrease in the exposure window to the vulnerability. The intuition behind this result can be seen as follows. Corollary 2 states that the vendor always prefers to release its patch later when an early

warning system is in place. From the coordinator perspective, whether it should extend its grace period beyond the optimal time when the patch is released without an early warning system is purely based on cost and benefit of such an action. Λ_v / p^* represents the marginal benefit-to-cost ratio of extending the grace period beyond p^*. If this ratio is greater than one, the marginal benefit is higher than the marginal cost. Therefore, the coordinator sets the grace period such that the release in the presence of an early warning system is later than the release in absence of such an early warning system.

Finally we show the impact of an early warning system on welfare in the next proposition.

Proposition 9: *The use of an early warning system by a coordinator does not necessarily improve the social welfare.*

Unlike expectations, the social welfare does not always improve with an early warning system. This result is quite interesting given that we assumed that there is no leakage of vulnerability knowledge. We can speculate that an early warning system may not be beneficial at all if the leakage cannot be prevented.

7. Discussion, Limitations, and Future Work

Despite a consensus on balancing the need to motivate the vendor and the need to reduce the impact of premature dissemination of vulnerability knowledge, there is no standard method to disclose vulnerabilities responsibly. Absence of a common practice often results in miscommunication, leading to "uncontrollable vulnerability handling, confused or angry customers and unnecessary windows of opportunity for malicious actions" (Takanen et al. 2004). This is the main reason why the government (Chambers and Thompson 2004) and a consortium of software vendors and security research firms (OIS 2004) have attempted to consolidate the multitude of "loosely organized" vulnerability disclosure policies (Shepherd 2003). Our results contribute to those efforts to resolve the most controversial issues surrounding the responsible vulnerability disclosure. First, like the optimal disclosure policy in the single vendor case (Cavusoglu et al. 2004a), none of the disclosure practices, immediate public, full vendor, or hybrid, is optimal all the time in the extensions that analyzed in this paper. However, we clearly show only one disclosure practice is optimal in a given scenario. Second, the grace period provided to the affected vendor(s) cannot be the same for different vulnerabilities. It depends on several factors. Third, the optimal grace period can increase or decrease when the vulnerability is shared by another vendor. Although highlighted as an important issue (OIS 2004, section 6.3), how vulnerability knowledge should be disclosed if

multiple vendors are affected by a common vulnerability has not been discussed in any disclosure guideline in practice. Our results prove useful in providing guidelines on how vulnerability disclosure policies should be modified to accommodate cases in which vulnerabilities affect more than one vendor. Fourth, we show that an early discovery improves the social welfare. An interpretation of this result is that appropriate incentive mechanisms that encourage the early vulnerability identification can greatly reduce the global impact of a vulnerability. Last, although some disclosure practices enforce advance notification of selected users (CERT/CC 2000), others do not provide any guideline to carry out this process effectively (OIS 2004, section 8.1). We find that the society might not always be better off with an early warning system that disseminates the vulnerability knowledge to a selected set of users. Some practitioners expressed their concern over an early warning system on the basis of a possibility of a leakage in the process that could increase the risk to the general population (Vijayam 2003), our result indicates that an early warning system might not be beneficial even if leakage of information is prevented.

Recently, we started seeing security organizations (e.g., iDefense, ISS Inc.) who are actively involved in discovering vulnerabilities. In fact, some of these organizations have programs in place (e.g, Vulnerability Contributor Program by iDefense) that encourage other people to submit vulnerability information to them to get paid in return. These firms then pass the vulnerability knowledge to their clients that are ready to pay for advance notifications. Although our interest is in the disclosure process which aims to maximize the well-being of the society, it will be useful to contrast the disclosure process analyzed in this paper (non-profit-based mechanism) with the disclosure process facilitated by a profit-seeking organization (often called market-based mechanism or profit-based mechanism). Kannan and Telang (2004) find that a profit-based mechanism almost always performs worse than a non-profit-based mechanism in terms of social welfare due to the possibility of information leakage in profit-based mechanism. Even if the leakage is prevented, they show that a profit-based mechanism performs better than a non-profit-based mechanism only under certain conditions. However their analysis ignores an important stakeholder in vulnerability disclosure process, which is the vendor. They assume that firms that subscribe the service provided by the profit-seeking coordinator are fully protected from the adverse effects of the vulnerability, which is hardly the reality. In fact, both iDefense and ISS Inc. inform the affected vendor about the reported vulnerability at the same time or before releasing the vulnerability knowledge to their subscribers. Further, firms that subscribe to early notification services can only get information about workarounds, which provides partial protection against vulnerabilities. The

ultimate solution to fix a vulnerability still remains as patching, which can be done only after the vendor releases a patch. Recognizing this fact, we suspect that a non-profit-based mechanism to handle vulnerabilities would always yield a higher social welfare than a profit-based mechanism. Future work in this area should investigate coexistence of a profit-based mechanism and a non-profit-based mechanism without ignoring the effect of the vulnerability handling process on the vendor.

Since Kannan and Telang (2004) ignore the vendor, they neither define the responsible way of handling vulnerabilities nor model it. They assume that once a hacker identifies a vulnerability before a benign user, the hacker attacks all vulnerable systems instantaneously. This assumption is unrealistic and implicitly makes the timing issue irrelevant. Unlike them, we assume that the vulnerability can only be exploited at some rate per unit time. If a hacker first identifies the vulnerability, he and his close associates enjoy exploiting it. During this period of *private exploitation* (Rescorla 2004), population at large is unaware of the vulnerability. At some point a benign user discovers the same vulnerability and responsible disclosure guidelines are followed. If a benign user first identifies the vulnerability, a hacker can still discover the same vulnerability until the public disclosure is made. However, we assume that the rate of exploitation increases after the vulnerability knowledge is disseminated to the public without an appropriate patch. Moreover, we assume that the vulnerability knowledge can be used to take intermediate precautions in order to reduce the likelihood of a successful exploitation. We supplement it by assuming that firms incur some cost by applying a workaround. All these assumptions improve our model's ability to simulate the reality.

In spite of our best efforts, our model has some limitations. The model uses various parameters. Those parameters have to be estimated accurately to obtain decision variables. However, the main contribution of the paper does not lie in obtaining exact values of those variables to define the grace period, but rather in understanding the intricate relationships between stakeholders in the vulnerability disclosure process and in determining the dynamics of optimal disclosure. Further, most of our results can be explained in terms of ratios of model parameters. So long as these ratios can be estimated with reasonable accuracy, our results can be used to characterize the responsible disclosure. Coordinators such as CERT, software security firms such as eEye, and software vendors such as Microsoft provide estimations on the impact of the vulnerability, the risk of exploitability, and the number of users that might be affected. As the vulnerability disclosure processes mature and coordinator(s), identifiers, and vendors gain more experience, we expect that the estimation

techniques will improve and deliver accurate predictions. We assumed constant attack rates over time before and after public disclosure. In fact, attack rate can increase over time since automated attack scripts can be developed as time passes. Our attempt was to create a parsimonious model that is tractable yet realistic.

Although we model the vulnerability identification and vulnerability exploitation as stochastic events, we assume that patch development is a deterministic event. Our reasoning is that the required work to develop a patch after the vulnerability knowledge is transferred to the vendor is generally composed of fairly deterministic steps. We also believe that our model can be extended to allow randomness in the patch development process. We anticipate that this will not change qualitative nature of our results.

We implicitly assume that there is a constant cost of handling the vulnerability. Since it is constant, we simply assume that it is zero throughout our analyses. Introducing non-zero vulnerability handling cost to the coordinator can be easily incorporated into our model without changing our results.

An important question in security is whether the software industry should be subject to product liability laws like other industries in commerce. This issue has been a topic of discussion among security practitioners and researchers (Schneier 2001; Varian 2000). Since product liability laws imply a higher β in vendor's cost function, and the vendor has more incentive to patch its vulnerability as β increases, we can say that product liability law is another mechanism to motivate vendors to act responsibly. Hence, coordinator's disclosure policies are crucial in promoting responsible behavior on vendor side until the liability laws are in effect in information security.

Currently CERT gives vendors 45 days to address vulnerabilities in their products. Although it seems that CERT is trying to please both sides by setting a hybrid policy, our results show that the *one-size-fits-all* kind of a policy is not an optimal solution. CERT should assess the risk associated with a vulnerability before setting a disclosure policy for that vulnerability. Our results imply that CERT should give less time to vendors if the vulnerability affects many firms and/or the risk associated with the vulnerability is high. In other words, for critical vulnerabilities CERT should be less patient to prevent serious harm to firms.

Contrary to some popular claims, full public disclosure may not be the best solution for vulnerability announcements, as shown in this paper. However, full public disclosure may push vendors to pay more attention to security and lead to better quality software in the long run. This indirect effect of disclosure policy, which is not captured in our model, can be a reason why some

people support full public disclosure. Future research should definitely address this interesting question.

The coordinator can encourage vendors that share a vulnerability to work together to develop a patch for the common vulnerability. It can develop an incentive mechanism to ensure the joint development of a patch for the common vulnerability to eliminate redundancy in patch development efforts of all affected vendors. This interpretation of our result has also a profound implication on open-source software development. As the software industry moves toward the open-source software development, common vulnerabilities will be seen more often. A joint development of a patch for a common vulnerability will become much more needed. Since the cost of patch development is shared by the open-source community, the individual patch cost contribution of each vendor can be significantly less than proprietary patch development cost. This benefit can be used as another justification for the open-source software.

7. Conclusions

When the vulnerability affects two vendors, we show that if they choose to develop a patch, they release their patches at the same time. On the other hand, there is a possibility that only a single vendor releases its patch and the other vendor does not release its patch. Unlike the single vendor case, responsible vulnerability disclosure policy cannot always guarantee the release of patches from both vendors in the multiple vendor case. Even though the coordinator does not impose any time constraint in some cases, the vendor with less incentive ignores the vulnerability. When both vendors release a patch, the grace period in the multiple vendor case is no shorter than minimum of the grace periods that it would set for the vendors in the single vendor case and no longer than the maximum of the grace periods that it would set for the vendors in the single vendor case. That is, a firm with higher (lower) patching cost benefit per its customer will be given less (more) time compared to the grace period in the single vendor case when it shares the vulnerability with a vendor with lower (higher) patching cost benefit per its customer. However, if the vendor with low incentive ignores the vulnerability, we find that the coordinator's provision to extend or shorten the grace period depends on the patching cost of the high incentive vendor relative to the damage and workaround savings per customer of the vendor that ignores the vulnerability.

There is an inherent relationship between the vulnerability disclosure process and the vulnerability discovery process. Our analysis sheds light on the importance of the vulnerability discovery process. As the social loss increases with the time of vulnerability discovery (t_0), the social planner may establish a mechanism that encourages benign users to discover the

vulnerabilities earlier. If vulnerability is discovered earlier as a result, we find that the vendor will release its patch earlier. The vendor does so even if coordinator does not shorten the grace period. With an early discovery of the vulnerability, we show that the society is always better off.

Lastly, we analyze the impact of an early warning system that provides privileged vulnerability knowledge to selected users before any patch is released. We find that with such a system the vendor has more incentive to postpone the release of its patch, yet the coordinator may increase/decrease the grace period that it will set. Hence, the patch can be delivered earlier or later than the time at which it would be released without an early warning system. Although informing a limited set of users seems to be plausible, we show that the social welfare does not necessarily improve with the use of early warning system.

We believe that the results of this study will be relevant to industry practitioners, including software vendors, software users, and coordination authorities like CERT as the study provides a deeper understanding of forces that shape the optimum policy for the vulnerability disclosure process. We also believe that the results of this study will serve as a guideline for policy makers, who aim to constitute a legal framework for vulnerability disclosure in order to minimize malevolent uses of disclosures and maximize the incentive of software vendors to fix their problems.

REFERENCES

Across. 2004. *ASPR notification and publishing policy*. Available at http://www.acrossecurity.com/aspr NotificationAndPublishingPolicy.htm

Arbaugh, W. A., W. L. Fithen, J. McHugh. 2000. Windows of vulnerability: A case study analysis. *IEEE Computer* **33** 52-59.

Arora, A., R. Telang, and H. Xu. 2004. Optimal Policy for Software Vulnerability Disclosure. *The Third Annual Workshop on Economics and Information Security (WEIS04)*. Minneapolis, MN, May 2004.

BindView. 2003. *Organization for internet safety issues a public comment draft for security vulnerability reporting and response guide*. Available at http://www.bindview.com/News/display.cfm?Release=2003/0604b.txt/

Bushman, R. M. 1991. Public disclosure of the structure of private information markets. *Journal of Accounting Research* **29** 261-276.

Cavusoglu, Hasan, Huseyin Cavusoglu, and S. Raghunathan. 2004a. Analysis of Software Vulnerability Disclosure Policies. *CORS/INFORMS Joint International Meeting*, Banff, Alberta, Canada, May 2004.

Cavusoglu, Hasan, Huseyin Cavusoglu, and S. Raghunathan. 2004b. How Should We Disclose Software Vulnerabilities. *Workshop on Information Technology and Systems (WITS)*, Washington, DC, December 2004.

Cavusoglu, H., B. Mishra, S. Raghunathan. 2002. Assessing the value of detective control in IT security, Proceedings of the 8^{th} *AMCIS*. 1910-1918, Dallas, TX.

Cavusoglu, H., B. Mishra, S. Raghunathan. 2003. Quantifying the value of IT security mechanisms and setting up an effective security architecture. 2^{nd} *Annual Workshop on Economics and Information Security*, College Park, MA.

Cavusoglu, H., B. Mishra, S. Raghunathan. 2004c. The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce* **9**(1) 69-104.

Cavusoglu, H., B. Mishra, S. Raghunathan. 2004d. A model for evaluating IT security investments. *Communications of the ACM* **47**(7) 87-92.

CBS News. 2002. *Locking windows*, CBS News.com. January 16.

CERT/CC. 2000. *Vulnerability disclosure policy*. CERT Coordination Center.

Chambers, J. C. and J. W. Thompson. 2004. *Vulnerability disclosure framework: final report and recommendations by the council*. National Infrastructure Advisory Council. January 13.

Clinch, G. R. E. Verrecchia. 1997. Competitive disadvantage and discretionary disclosures in industries. *Australian Journal of Management* **22** 125-137

Culp, S. 2000. *Definition of a security vulnerability*. MicrosoftTechNet.

CYBSEC. 2004. *CYBSEC security vulnerability disclosure policy*. Available at http://www.cybsec.com/vulnerability_policy.pdf

Dacey, R. F. 2003. *Information security: effective patch management is critical to mitigating software vulnerabilities*. GAO-03-1138T.

Demski, J., Feltham, G. 1994. Market response to financial reports. *Journal of Accounting and Economics* **17** 3-40.

Dye, R. A. 1990. Mandatory versus voluntary disclosure: the case of financial and real externaties. *The Accounting Review* **65** 1-24.

eEyes. 2004. *Upcoming advisories*. http://www.eeye.com/html/research/upcoming/index.html

Fischer, P. E., R. E.Verrecchia. 1999. Public information and heuristic trade. *Journal of Accounting and Economics* **27** 89-124.

Fisher, D. 2003. *CERT, Feds Consider New Reporting Process*. eWeek. March 24.

Gongloff, M. 2003. *Worm rot or not*. CNN Money. August 23.

Grossman, S. J., J. E. Stiglitz. 1980. On the possibility of informationally efficient markets. *American Economic Review* **70** 393-408.

Havana, T. 2003. Communication in the software vulnerability reporting process. *M.A. Thesis*. The University of Jyväskylä.

Hulme, H. 2002. *Businesses keep spending on security*, InformationWeek. 96, January.

Kannan, K. and R. Telang. 2004. Market for vulnerabilities? Think again. *Working Paper*. October.

Laakso, M., A. Takanen, J. Röning. 1999. The vulnerability process: a tiger team approach to resolving vulnerability cases. Proceedings of *the 11th FIRST Conference on Computer Security Incident Handling and Response*. Brisbane.

Loch, K. D., H. H. Carr and M. E. Warkentin. 1992. Threats to information systems: today is reality yesterday is understanding. *MIS Quarterly* **17**(2) 173-186.

Lundholm, R. J. 1991. Public signals and the equilibrium allocation of private information. *Journal of Accounting Research* **29** 322-349.

Niederman, F., J. C. Brancheau and J. C. Wetherbe. 1991. Information systems management issues for the 1990s. *MIS Quarterly* **15**(4) 475-495.

NIST 800-40. 2002. *Procedures for handling security patches*. NIST special publication.

OIS 2004. *Guidelines for security vulnerability reporting and response*. Organization for Internet Safety, Version 2.0. September 1.

Palella, M. 2003. *Vulnerability disclosure, double edged sword*. GIAC SEC Practical Repository, SANS Institute.

Pond, W. 2000. Do security holes demand full disclosure? *ZDNet*. August 15.

Preston, E., J. Lofton. 2002. Computer security publications: information economics, shifting liability and the first amendment. *Whittier Law Review* **24** 71-142.

Ranum, M. J. 2004. *Vulnerability disclosure- let's be honest about motives shall we?* Editorials on Computer Security. http://www.ranum.com/security/computer_security/index.html

Rescorla, E. 2004. Is finding security holes a good idea? *The Third Annual Workshop on Economics and Information Security*, Minneapolis MN. May 13-14.

SBQ. 2002. Special Issue: Vulnerability disclosure. *Secure Business Quarterly* **2**.

Schiller, J. 2002. Responsible vulnerability disclosure: a hard problem. *Secure Business Quarterly* **2** 1-5.

Schneier, B. 2001. *Bug secrecy vs. full disclosure*. ZDNet TechUpdate, November 14.

Shepherd, S. A. 2003. *Vulnerability disclosure: how do we define responsible disclosure?* GIAC SEC Practical Repository, SANS Institute.

Stone, A. 2003. Software flaws: to tell or not to tell? *IEEE Computer* **20** (1) 70-73.

Straub, D.W. 1990. Effective IS security: an empirical study. *Information Systems Research* **1**(3) 255-276.

Straub, D.W. and R. J. Welke. 1998. Coping with systems risk: security planning models for management decision making. *MIS Quarterly* **22**(4) 441-469.

Tain, Y. 2002. Regulation for reporting security flaws, *Working Paper*, Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology.

Takanen, A., P. Vuorijarvi, M. Laakso, J. Roning. 2004. Agents of responsibility in software vulnerability processes. *Ethics and Information Technology*, **6**(2) 93 – 110.

Vamosi, R. 2003. *Digital pearl harbor: it's already happened*. ZDNet. December 22.

Varian, H. R. 2000. *Managing online security risks*. The New York Times, June 1.

Verrecchia, R. E. 1982. The use of mathematical models in financial accounting. *Journal of Accounting Research* **20** 1-42.

Verrecchia, R. E. 1983. Discretionary disclosure. *Journal of Accounting and Economics* **5** 365-380.

Vijayam, J. 2003. Bug disclosure, fix process improving. *Computerworld*. March 10.

APPENDIX: Available from authors upon request.