Efficiency of Vulnerability Disclosure Mechanisms to Disseminate Vulnerability Knowledge

Hasan Cavusoglu, Huseyin Cavusoglu, and Srinivasan Raghunathan

Abstract—Security vulnerabilities in software are one of the primary reasons for security breaches, and an important challenge from knowledge management perspective is to determine how to manage the disclosure of knowledge about those vulnerabilities. The security community has proposed several disclosure mechanisms, such as full vendor, immediate public, and hybrid, and has debated about the merits and demerits of these alternatives. In this paper, we study how vulnerabilities should be disclosed to minimize the social loss. We find that the characteristics of the vulnerability (vulnerability risk before and after disclosure), cost structure of the software user population, and vendor's incentives to develop a patch determine the optimal (responsible) vulnerability disclosure. We show that, unlike some existing vulnerability disclosure mechanisms that fail to motivate the vendor to release its patch, responsible vulnerability disclosure policy always ensures the release of a patch. However, we find that this is not because of the threat of public disclosure, as argued by some security practitioners. In fact, not restricting the vendor with a time constraint can ensure the patch release. This result runs counter to the argument of some that setting a grace period always pushes the vendor to develop a patch. When the vulnerability affects multiple vendors, we show that the responsible disclosure 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. This implies that the coordinator does not necessarily increase the grace period to accommodate more vendors. We then extend our base model to analyze the impact of 1) early discovery and 2) an early warning system that provides privileged vulnerability knowledge to selected users before the release of a patch for the vulnerability on responsible vulnerability disclosure. We show that while early discovery always improves the social welfare, an early warning system does not necessarily improve the social welfare.

Index Terms—Information security, software vulnerabilities, disclosure mechanisms, responsible vulnerability disclosure, economic modeling, game theory.

1 Introduction

The rate of information security breaches has been increasing significantly [7]. One of the main reasons for this increase is security vulnerabilities in software.[1,2] 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. Further, wide availability of these tools on the Internet eliminates the need for specialized knowledge and expertise to exploit vulnerabilities, making even novice hackers capable of launching sophisticated attacks on vulnerable systems.

Software experts often blame software vulnerabilities on poor software development practices, such as improper testing, failure to control common programming errors, and poor understanding of the interactions between different components of complex software [31]. Studies estimate that there can be as many as 20 flaws per thousand lines of code [19]. Although many software vendors have recently started paying more attention to secure software engineering practices,[3] 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

1. Vulnerabilities can be considered as technical errors. Other main reasons for the increase in the number of security breaches include misconfiguration of systems (configuration errors) and user mistakes (human errors).

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*" [12].

3. 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 [9].

- *H. Cavusoglu is with the Sauder School of Business, University of British Columbia, Vancouver, BC V6T 1Z2, Canada. E-mail: cavusoglu@sauder.ubc.ca.*
- *H. Cavusoglu and S. Raghunathan are with the School of Management, University of Texas at Dallas, Richardson, TX 75083. E-mail: {huseyin, sraghu}@utdallas.edu.*

Manuscript received 15 Sept. 2006; revised 14 Dec. 2006; accepted 29 Dec. 2006; published online 26 Jan. 2007.
Recommended for acceptance by A. Rubin.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-0221-0906.

debated issue in information security [28]. 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? 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?

The steps followed to handle dissemination of the knowledge of software vulnerabilities after a benign user identifies them are collectively known as the *vulnerability disclosure process*. *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[4] associated with vulnerabilities. Despite the consensus on the objective of responsible vulnerability disclosure, no such consensus exists on how the vulnerability knowledge should be disseminated to various stakeholders. The lack of consensus is partially because there is no common understanding of the impact of the vulnerability knowledge on different stakeholders. The vulnerability information has relevance to the vendor, software users, and hackers. Therefore, the sequence and timing of announcement of the vulnerability, that is, how vulnerability knowledge is managed, affects the total cost of the vulnerability to the society.

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. 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 [31]. 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 [24]. 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, 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.

4. The social loss is defined as total loss caused by a software vulnerability which comprises the vendor's patch development cost and the damage and workaround costs incurred by vulnerable firms.

Because each of these policies tends to favor one stakeholder, 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 [33]. 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 [33], and the vendor is given motivation without jeopardizing the security of firms using the vulnerable software [30]. Over time, 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 [10]. Instead of working with CERT/CC, an identifier can reveal the vulnerability knowledge directly to the vendor. 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* [21] by the 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 the 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 [32]. Since the US government has emphasized the criticality of a predesigned responsible vulnerability disclosure process [11], and software vendors and security research firms have begun to jointly develop a unified framework for vulnerability disclosure [21], it is expected that these efforts will soon converge. Yet, it is not clear what constitutes the responsible vulnerability disclosure process.

The goals of our research are 1) to understand the impact of vulnerability disclosure mechanisms on the decisions of

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 vendor releases a patch, although it sets a 60-day target period [14]. 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 [1]. BindView follows OIS guidelines for vulnerability disclosure [5]. CYBSEC gives 45 days to the vendor before issuing an advisory to inform the public [13].

various stakeholders and 2) to develop policy guidelines for the responsible vulnerability disclosure.[7] If the one-size-fits-all solution is not right [32], we are specifically interested in 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 are also interested in how much time the vendor should be given to develop a patch.[8]

We develop a game-theoretical model in which the coordinator minimizes the societal loss, which includes both damage to vulnerable firms and the patch development cost to the software vendor. Next, we investigate a case in which the same vulnerability affects multiple software vendors. This is very important given that 1) many software vendors base their proprietary software on open-source codes and 2) a flaw in a common protocol may lead to vulnerabilities in several software applications. Further, we extend our base model in two different directions. First, 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. Second, we study a controversial question of whether an early warning system that provides privileged vulnerability knowledge to selected users before the release of a patch for the vulnerability would improve the social welfare.

The rest of the paper is organized as follows: In the next section, we review the relevant literature. In Section 3, we model and analyze the vulnerability process to characterize the responsible vulnerability disclosure mechanism, assuming that the vulnerability affects only a single vendor. In Section 4, we study how knowledge of vulnerability should be disseminated if the vulnerability affects multiple vendors. In Section 5, we extend our model to analyze the impact of 1) early discovery of the vulnerability and 2) early warning to selected users on the responsible disclosure mechanism. 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

Despite serious discussion among security professionals, to the best of our knowledge, only few academic studies investigate vulnerability disclosure policies from social, organizational, or economic perspectives. Laakso et al. [18] describe the vulnerability disclosure 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 [16] analyzes the communication in the software vulnerability reporting process between identifiers/coordinators and vendors. This survey study finds 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 reveals that both groups are generally opposed to the immediate public and full vendor disclosure. In their case study analysis, Arbaugh et al. [2] test the vulnerability life-cycle model using incident data from CERT/CC about three common vulnerabilities. Nizovtsev and Thursby [20] study the incentives of individuals to disclose software vulnerabilities. Arora et al. [3] consider how the vendor's patching decision changes for a given grace period. Takanen et al. [34] identify ethical responsibilities of stakeholders in the vulnerability disclosure process. Different from other studies in vulnerability disclosure literature, Preston and Lofton [25] approach the vulnerability disclosure from a legal perspective and argue how and when the disclosure of information about software vulnerabilities can be protected under the First Amendment.

Our study in vulnerability disclosure is related to studies in the vulnerability discovery literature and the patch management literature. In the vulnerability discovery literature, researchers in general question the social value of discovering software vulnerabilities. Schechter [29] proposes a vulnerability knowledge market where software vendors offer rewards for finding vulnerabilities in their products to improve security. Ozment [22] argues that bug auctions can improve the efficiency of such vulnerability market. Building on these arguments, Kannan et al. [17] model the competition between benign users and hackers in the vulnerability discovery process to analyze the impact of monetary incentives provided by a profit-seeking organization on vulnerability identification. Rescorla [27] challenges the assumption that effort invested in finding vulnerabilities in software is socially useful and finds that the likelihood of rediscovery is very low, which conflicts with the key assumption of vulnerability disclosure models that attackers are likely to rediscover any vulnerability found by a benign identifier. Thus, he concludes that there is no value of vulnerability hunting from a social welfare perspective. Using a new and cleaner data set, Ozment [23] later shows that likelihood of rediscovery is not negligible, and therefore provides some support for the value of vulnerability hunting and the need for vulnerability disclosure mechanisms.

In the patch management literature, researchers address the importance of patching for security management. Cavusoglu et al. [8] tackle the question of how patches should be released by vendors and updated by firms in batches. They find that the social welfare is maximized when patch release and update cycles are synchronized. They also show that cost sharing and liability can coordinate the patch management decisions of vendors and firms to achieve social optimality. August and Tunca [4] study the effects of patching costs and negative security externalities in managing network security. They conclude that policies to manage network security should target the incentives of end users.

7. Our study is extremely timely as the US Department of Homeland Security reviews vulnerability disclosure mechanisms and considers mandating a centralized vulnerability disclosure policy [15].

8. Vulnerability disclosure is typically discussed along two dimensions. The first dimension is the quality of information that needs to be disclosed while the second dimension is the time of disclosure to various stakeholders. The debate on quality of information is between those who favor complete disclosure that includes exploits and attack scripts and those who favor limited disclosure that discloses the mere existence of the vulnerability so that firms can take actions to limit exposure to vulnerability. Complete disclosure is very risky since it allows automatic exploitations of vulnerabilities and, therefore, cannot be responsible disclosure. In this paper, we address the second dimension of vulnerability disclosure assuming that any type of disclosure provides only information about vulnerability (limited disclosure), not attack tools that can exploit the vulnerability.

While vulnerability disclosure and patch management are somewhat related topics, they are different in several dimensions. First, in vulnerability disclosure, the issue is how much time should be given to the vendor to develop a patch before releasing vulnerability information to public. In patch management, the issue is how the vendor should release its patches and how vulnerable firms should update their systems with released patches. Second, vulnerability disclosure becomes an issue if the vulnerability is discovered by an identifier who can use disclosure as a threat to get the patch out from the vendor. Patch management is an issue if the identifier is the vendor itself or a third party who does not threaten the vendor with public disclosure. Third, vulnerability disclosure process involves an identifier (or a coordinator like CERT) and the vendor. Although the firm is definitely a stakeholder in vulnerability disclosure, the firm has no action to take in the disclosure process. However, both the vendor and firm are active players in the patch management process. Fourth, the objective in vulnerability disclosure is to minimize the social loss by disseminating vulnerability information to appropriate parties at appropriate times, whereas the objective in patch management is to minimize individual losses by choosing appropriate release and update cycles.

3 Model

There are four stakeholders in the vulnerability knowledge dissemination 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.[9] We model the vulnerability disclosure process as a sequential game. First, the coordinator sets how much time to give to the vendor for patch development. Then, the vendor decides when to release its patch knowing the grace period set by the coordinator.

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 then 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 then notifies the vendor about the vulnerability and sets a grace period T. The vendor next decides the time p at which it will release its patch (hereafter, patch release time). If the vendor releases a patch before the end of the grace period (i.e., $p \leq t_0 + T$), the coordinator or the vendor publicly announces the vulnerability along with the patch information 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 vulnerability, or monitoring the vulnerable systems extensively. Since a quick fix (workaround) is mostly achieved by reducing some functionality of the software, 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 $(0 \leq \gamma \leq 1)$ is the inefficiency of the workaround.

Hackers can discover the vulnerability anytime after the software is introduced to the market. We assume that hackers identify the vulnerability for the first time at time y. This setup makes it possible that a vulnerability can be identified by a hacker before a benign user. We assume that discovery time of the vulnerability by hackers is uniformly distributed with probability density function α. Typically, α is quite small because not all software vulnerabilities are discovered even after the software has been in use for a long time. After a hacker discovers the vulnerability, vulnerable firms are attacked at a rate of a (i.e., the number of attacks that each firm suffers per time unit is a).[10] We assume that the rate of attacks increases to ka, $k > 1$, after the vulnerability is announced to the public.[11] 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 Fig. 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. Since patch development involves extensive testing to make sure that the patch is working properly and does not conflict with other applications when deployed across various platforms, the vendor incurs more cost if it releases the patch faster. 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.[12] 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

9. Although we consider a central coordinator in the responsible vulnerability disclosure process, as we mentioned in the Section 1, this process can work without any coordinator.

10. Browne et al. [6] show that the linear model in time is a good approximation for modeling the number of security breaches resulting from a specific vulnerability. One might also argue that the higher the rate of the discovery of vulnerability by hackers, the higher the rate of its exploitation. This can be easily incorporated in our model by allowing a to be a function of α. As α is not a decision variable, the equilibria that we find do not change, except that a will be replaced by $a(\alpha)$.

11. Rescorla [27] shows that if hacker identifies the vulnerability first, he and his close associates enjoy exploiting it. During this period of *private exploitation*, the population at large is unaware of the vulnerability. At some point, a benign user discovers the same vulnerability and responsible disclosure guidelines are followed. However, after public disclosure if a patch has not come out yet, hackers attack vulnerable systems at a higher rate than the private exploitation period during the period of *public exploitation*.

12. Since there is no empirical evidence on how time impacts the patch development cost, we assume that development cost decreases with a constant rate in time and that $\varepsilon_1 > \frac{\varepsilon_2^2}{\alpha\beta\delta^2 Na} - \varepsilon_2 t_0$ so that the vendor always incurs a positive patch development cost if it decides to release a patch. However, any monotonically decreasing patch cost function would lead to qualitatively the same results. The reason is that the vendor's overall cost in (1) is always convex with any type of monotonically decreasing patch cost function. We chose the linear patch cost function for mathematical convenience.

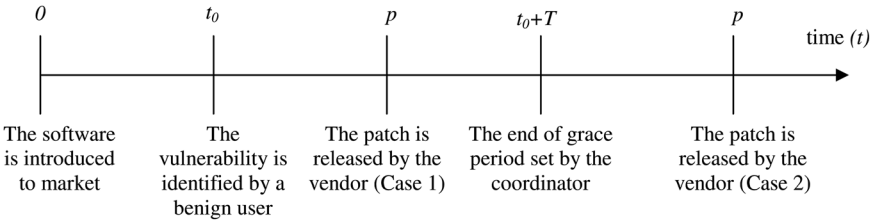

Fig. 1. The timeline for the vulnerability discovery and disclosure process.

expected reputation cost to the vendor per successful attack. 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

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

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 damage.[13] If the vendor releases the patch before the end of the grace period (i.e., $p \le t_0 + T$), the social cost is

$$\begin{aligned} C &= \varepsilon_1 - \varepsilon_2(p - t_0) + \int_o^p \int_y^p \int_0^1 \alpha N\delta D\theta ad\theta dtdy \\ &= \varepsilon_1 - \varepsilon_2(p - t_0) + \frac{\alpha N\delta Dap^2}{4}. \end{aligned} \tag{3}$$

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

$$\begin{aligned} C &= \varepsilon_1 - \varepsilon_2(p - t_0) + \int_0^{t_0+T} \int_y^{t_0+T} \int_0^1 \alpha N\delta D\theta ad\theta dtdy \\ &\quad + \int_{t_0+T}^p \int_0^1 \delta\gamma ND\theta kad\theta dt + \int_{t_0+T}^p Nsdt \\ &= \varepsilon_1 - \varepsilon_2(p - t_0) + \frac{\alpha N\delta Da(t_0+T)^2}{4} + \frac{\gamma N\delta Dka}{2}(p - t_0 - T) \\ &\quad + Ns(p - t_0 - T). \end{aligned} \tag{4}$$

In the following section, we first solve how the vendor would react to a given T. Given the vendor's reaction, we then solve the coordinator's problem to find the optimal T.

3.3 The Vendor's Patch Release Policy

After the coordinator informs the vendor about the vulnerability and its disclosure policy, the vendor decides whether to release a patch, and if so, when to release it. When $p \le t_0 + T$ (hereafter, case 1), the vendor minimizes the cost expression given in (1). The following patch release times capture the vendor's best response:

$$p^{(1)} = \begin{cases} t_0 + T & \text{if } T < (\Omega/\alpha) - t_0 \\ \overline{p} = \Omega/\alpha & \text{if } T \ge (\Omega/\alpha) - t_0 \text{ and } t_0 \le \Omega/\alpha \\ t_0 & \text{if } t_0 > \Omega/\alpha, \end{cases} \tag{5}$$

where $\overline{p}$ is the interior solution, $^{(i)}$ represents case i, and $\Omega = \varepsilon_2/\beta\delta Na$.

When $p > t_0 + T$ (hereafter, case 2), minimizing the cost expression in (2) gives

$$p^{(2)} = \begin{cases} t_0 + T & \text{if } \gamma\delta\beta Nka > \varepsilon_2 \\ \infty & \text{if } \gamma\delta\beta Nka < \varepsilon_2. \end{cases} \tag{6}$$

To find the optimal p (i.e., p^*) for a given T, the vendor compares its best reactions for case 1 and case 2, given in (5) and (6), respectively. This comparison reveals that the vendor does not release any patch for the vulnerability when $\gamma k < \Omega$ if the coordinator provides a finite grace period. The reason is that, after the public disclosure, savings in patch development cost outweighs the reputation loss associated with postponing the patch release per unit time. Therefore, the threat of public announcement of the vulnerability is not effective. However, the vendor releases its patch when $\gamma k > \Omega$. Specifically, the term $\Omega/\gamma k$ represents the marginal benefit-to-cost ratio of postponing the release of the patch for the vendor after the public disclosure of the vulnerability.

13. Total damage cost is the sum of each firm's expected damage cost due to a successful exploitation, calculated over the discovery time of the vulnerability by a hacker y, attack time t, and the firm type θ. Since the cumulative number of attacks increases in time until a patch or workaround is applied, total damage cost becomes quadratic in time after working out the integrations (see (3) and (4)). Quadratic total damage cost implies that the longer the exposure time, the higher the incremental damage from each additional unit of exposure to the vulnerability.

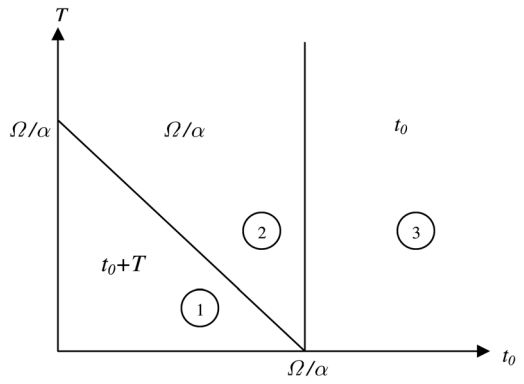

Fig. 2. Vendor's best response function when $\gamma k > \Omega$.

Proposition 1. *When $\gamma k < \Omega$, the vendor disregards the vulnerability when given a finite grace period.*

The proofs of our results are available in the online supplement, which can be found in the Computer Society Digital Library at http://computer.org/tse/archives.htm. If the vendor releases a patch (i.e., $\gamma k > \Omega$), there are three possibilities for the patch release time. The vendor can release its patch 1) at the end of the grace period, 2) before the end of the grace period, or 3) immediately. Fig. 2 shows these equilibrium regions for all possible values of T and t_0. $\Omega/\alpha\tau$ represents the marginal benefit-to-cost ratio of the vendor to postpone the release of the patch before the public disclosure, where $t_0 \le \tau \le t_0 + T$. Hence, we can interpret Ω/α as the *tolerance level of the vendor* to delay the release of a patch; that is, the higher the value of Ω/α, the longer the vendor waits to release its patch. When $t_0 > (\Omega/\alpha)$, the vendor releases the patch immediately (see region 3 in Fig. 2). In this case, the vendor's self-interest, rather than the coordinator's threat of public disclosure, determines the vendor's patch release time.

When $t_0 \le \Omega/\alpha$, the grace period affects the patch release time decision made by the vendor. If the vulnerability is discovered early in the product life-cycle, the vendor utilizes the grace period fully as long as the grace period is not too long (see region 1 in Fig. 2). In this case, since the likelihood that hackers have already found the vulnerability is low, the expected reputation cost is low, and the vendor can afford to delay the release of the patch to save in patch development cost. In region 2, the vendor patches the vulnerability before the end of the grace period because waiting until the end of the grace period is not beneficial for the vendor.

Proposition 2. *If the vendor releases a patch when it is given a finite grace period ($\gamma k > \Omega$), it intends to wait to release the patch if the discovery time of the vulnerability (t_0) is smaller than the tolerance level of the vendor (Ω/α), and release the patch immediately, otherwise.*

3.4 The Coordinator's Optimal Disclosure Policy

After anticipating the vendor's best response for a given T, the coordinator decides on the optimal T (i.e., T^*) which minimizes the social loss. First, assume that $\gamma k > \Omega$ holds. When $t_0 > \Omega/\alpha$, the vendor releases the patch immediately. Therefore, the coordinator's disclosure policy is irrelevant.

When $t_0 \le \Omega/\alpha$, there are two alternative regions in the vendor's reaction function. In region 2, the social cost, $C(p = \Omega/\alpha)$, is independent of T. Therefore, the coordinator can choose any grace period as long as the vendor stays in that region (i.e., $T \in ((\Omega/\alpha) - t_0, \infty)$). In region 1, the vendor releases its patch at $t_0 + T$. Since $C(p = t_0 + T)$ is convex (i.e., $\partial^2 C(p = t_0 + T)/\partial T^2 > 0$, assuming that $T \in [0, (\Omega/\alpha) - t_0]$, the coordinator chooses its disclosure policy as follows:

$$T = \begin{cases} (\Omega/\alpha) - t_0 & \text{if } \Lambda > \Omega/\alpha \text{ and } t_0 \le \Omega/\alpha \\ \Lambda - t_0 & \text{if } \Omega/\alpha \ge \Lambda > t_0 \text{ and } t_0 \le \Omega/\alpha \\ 0 & \text{if } \Lambda \le t_0 \text{ and } t_0 \le \Omega/\alpha, \end{cases} \quad (7)$$

where $\Lambda = 2\varepsilon_2/\alpha N \delta a D$.

To determine the optimal disclosure policy, the coordinator then compares the social cost in region 1 and region 2. We can show that $C(p = \Omega/\alpha) = C(p = t_0 + T)|_{T=(\Omega/\alpha)-t_0}$, $C(p = \Omega/\alpha) > C(p = t_0 + T)|_{T=\Lambda-t_0}$ and

$$C(p = \Omega/\alpha) > C(p = t_0 + T)|_{T=0}.$$

Therefore, given $\gamma k > \Omega$, the following policies are optimal:

$$T^* = \begin{cases} \text{Irrelevant (i.e., } T \in [0, \infty)) & \text{if } t_0 > \Omega/\alpha \\ \text{Any } T \in [(\Omega/\alpha) - t_0, \infty) & \text{if } \Lambda > \Omega/\alpha \text{ and } t_0 \le \Omega/\alpha \\ \Lambda - t_0 & \text{if } \Omega/\alpha \ge \Lambda > t_0 \\ & \quad \text{and } t_0 \le \Omega/\alpha \\ 0 & \text{if } \Lambda \le t_0 \text{ and } t_0 \le \Omega/\alpha. \end{cases} \quad (8)$$

Second, assume that $\gamma k < \Omega$ holds. The coordinator knows that the vendor will not release a patch if it is given a finite grace period. However, if the coordinator gives an infinite grace period to the vendor, then the vendor releases its patch. The reason is that by not constraining the vendor with a finite grace period, the coordinator actually forces the vendor to stay in case 1, where it is always beneficial for the vendor to release a patch at some finite time, which is $\max(\Omega/\alpha, t_0)$. When $\gamma k < \Omega$, the coordinator may either publicly disclose the vulnerability ($T < \infty$) or choose to keep the vulnerability secret from the public and implement full vendor disclosure ($T = \infty$). Therefore, the coordinator has to compare the resulting social loss from these two alternatives to find the responsible vulnerability disclosure. The social loss is confined to a positive value in the second alternative since the patch will be released eventually, but not in the first alternative where the vendor does not release its patch at all. Therefore, the coordinator chooses not to restrict the vendor with a finite grace period. Based on these findings, the coordinator's optimal disclosure policy is summarized in Table 1.

We 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.

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) $t_0 \leq \Omega/\alpha$ (iii) $\Lambda > \Omega/\alpha$	Any $T \in [(\Omega/\alpha) - t_0, \infty)$	Ω/α
S4	(i) $\gamma k > \Omega$ (ii) $t_0 \leq \Omega/\alpha$ (iii) $\Omega/\alpha \geq \Lambda > t_0$	$\Lambda - t_0$	$t_0 + T^* = \Lambda$
S5	(i) $\gamma k > \Omega$ (ii) $t_0 \leq \Omega/\alpha$ (iii) $\Lambda \leq t_0$	0	$t_0 + T^* = t_0$

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). 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). Λ/τ represents the marginal benefit-to-cost ratio of the coordinator to allow additional time to the vendor at time τ, where $t_0 \leq \tau \leq t_0 + T$. 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).

Our results show that none of the disclosure mechanisms can be used for the responsible vulnerability disclosure exclusively. Each one can be a vehicle to motivate the vendor and ensure that the identifier acts responsibly depending on the vendor's and the coordinator's incentives. The responsible mechanism for the vulnerability disclosure is *full vendor disclosure* in S1. S3 and S4 are examples of *hybrid disclosure* in which the coordinator sets a finite grace period. In S5, the coordinator chooses *immediate public disclosure* as the responsible vulnerability disclosure mechanism. There may not even be a need for a disclosure mechanism to motivate the vendor if the vendor is internally motivated (S2).

4 Analysis of the Multiple Vendor Case

The previous analysis 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.[14] 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 a 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

14. CERT Vulnerability Notes Database reveals that each firm sharing a common vulnerability develops a patch for its product independent of other affected vendors. A positive externality in which a patch released by a vendor fixes a similar vulnerability in another vendor's product is not common.

TABLE 2
Optimal Disclosure Policy and Patch Release Times in the 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) $\gamma k < \Omega_1$	∞	$\max(\Omega_1/\alpha, t_0)$	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) $\overline{\Lambda} > \Omega_1/\alpha$	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) $t_0 < \overline{\Lambda} < \Omega_1/\alpha$	$\overline{\Lambda} - t_0$	$t_0 + T_m^* = \overline{\Lambda}$	$t_0 + T_m^* = \overline{\Lambda}$
M5	(i) $\gamma k > \Omega_2$ (ii) $t_0 \le \Omega_1/\alpha$ (iii) $\overline{\Lambda} \le t_0$	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

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

leave customers of some vendors without a patch, leading to increased exposure to hacker attacks [30].

We extend our single-vendor model 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.[15] 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 succesful attack. 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.

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 does not affect the customers of vendor 2 directly, unlike in the first case.[16] Analysis of these cases leads to the following optimal disclosure policy for the coordinator, as shown in Table 2.

15. 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's tolerance level to delay the release of a patch. Therefore, a vendor with the a higher value of Ω_i prefers to wait more before releasing its patch.

16. 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.

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

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 and can be classified into three groups. 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., $\gamma k < \Omega_1$).[17] 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 (M2 through M5) occurs 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., $\gamma k > \Omega_2$). 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 ($\overline{\Lambda}/\tau$) and vendor 1's marginal benefit to marginal cost ratio of postponing the patch release ($\Omega_1/\alpha\tau$), where $t_0 < \tau \leq t_0 + T_m$. Unlike the single-vendor case, the coordinator's benefit-to-cost ratio includes costs and benefits associated with both vendors.[18] The third group (M6 through M9) occurs 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 that the public gets the vulnerability knowledge. Equilibria M6 through M9 are qualitatively similar to equilibria S2 through S5 in the single-vendor case, except that 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 $t_0 \leq \tau \leq t_0 + T_m$. 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 work-around cost to customers of vendor 2 (i.e., $N_2 s$).

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

18. 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 $\overline{\Lambda}/\tau$.

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

Proposition 3. *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 3 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 the multiple-vendor case when an additional vendor is introduced.

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 4. *1) If a vendor in the single-vendor case is to share the vulnerability with a vendor that has less incentive than itself, then the disclosure policy of the coordinator in the multiple-vendor case cannot prevent the original vendor from releasing its patch.*

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

Next, we address how the coordinator's decision changes when the 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 the multiple-vendor case compared to the single-vendor case.[19] In the first scenario, both vendors

19. 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).

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 $[2\varepsilon_{2i} + 2\varepsilon_{2j}]/[\alpha(N_i + N_j)\delta a D\tau]$ 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 5. *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 ($\gamma k > \Omega_2 > \Omega_1$) 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 1) more incentive if $\frac{\varepsilon_{22}}{N_2} < (>) \frac{\varepsilon_{21}}{N_1}$, and 2) 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 one 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 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 ($\Lambda < \overline{\Lambda}$), it does not shorten the grace period. Corollary 1 follows directly from Proposition 5.

Corollary 1. *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 ($\gamma k > \Omega_2 > \Omega_1$) 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, $T_i \leq T_m \leq T_j$, where $i \neq j \in \{1, 2\}$.*

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 $[2\varepsilon_{21} + \gamma\delta N_2 k a D + 2N_2 s]/[\alpha(N_1 + N_2)\delta a D\tau]$. 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 6. *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 $t_0 \leq \Omega_1/\alpha < \Omega_2/\alpha$), 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 $\frac{\gamma}{2}\delta k a D + s < (>) \frac{\varepsilon_{21}}{N_1}$.*

Similar to Proposition 5, we show in Proposition 6, 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 ($(\gamma\delta k a D/2) + s$). 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 Extensions to the Basic Model

5.1 Early Discovery of Vulnerabilities

There is an inherent relationship between the vulnerability disclosure process and the vulnerability discovery process. Although the focus of our analysis is on vulnerability disclosure, we can extend the basic model 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. Our basic model 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

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	∞	$\max(\Omega/\alpha, t_0')$
S2	S2	Irrelevant	t_0'
	S3	Any $T' \in [(\Omega/\alpha) - t_0', \infty)$	Ω/α
	S4	$\Lambda - t_0'$	Λ
	S5	0	t_0'
S3	S3	Any $T' \in [(\Omega/\alpha) - t_0', \infty)$	Ω/α
S4	S4	$\Lambda - t_0'$	Λ
S5	S4	$\Lambda - t_0'$	Λ
	S5	0	t_0'

security firm, peer recognition, and favorable press coverage [24]. 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 may drive discovery of vulnerabilities faster. 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. Table 4 shows how the optimal disclosure policy changes when the vulnerability is discovered earlier. A comparison between patch release times in each row reveals an interesting result summarized in the following proposition:

Proposition 7. *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 7 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 optimal disclosure policies given in Table 1 and Table 4:[20]

Proposition 8. *If the discovery of vulnerability occurs earlier ($t_0' < t_0$), the coordinator does not shorten the grace period (i.e., $T'^* \geq T^*$).*

20. When there are several grace periods that are optimal, we assume that the coordinator chooses the shortest one.

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 9. *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 8 and 9 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 nonmonetary incentive schemes 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.

5.2 Early Warning System to Selected Firms

In this extension, we investigate an early warning system that provides the vulnerability knowledge 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 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 from developing a patch promptly, knowing that

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) $\gamma k < \Omega$	∞	$\max(\Omega_v/\alpha, t_0)$
V2	(i) $\gamma k > \Omega$ (ii) $t_0 > \Omega_v/\alpha$	Irrelevant	t_0
V3	(i) $\gamma k > \Omega$ (ii) $t_0 \leq \Omega_v/\alpha$ (iii) $\Lambda_v > \Omega_v/\alpha$	Any $T \in [(\Omega_v/\alpha) - t_0, \infty)$	Ω_v/α
V4	(i) $\gamma k > \Omega$ (ii) $t_0 \leq \Omega_v/\alpha$ (iii) $\Omega_v/\alpha \geq \Lambda_v > t_0$	$\Lambda_v - t_0$	$t_0 + T^* = \Lambda_v$
V5	(i) $\gamma k > \Omega$ (ii) $t_0 \leq \Omega_v/\alpha$ (iii) $\Lambda_v \leq t_0$	0	$t_0 + T^* = t_0$

where $\Lambda_v = (2\varepsilon_2 - 2\upsilon Ns)/(\alpha N\delta aD(1-(1-\gamma)\upsilon))$ *and* $\Omega_v = \epsilon_2/(\beta\delta Na(1-(1-\gamma)\upsilon))$.

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 prenotification 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.[21] 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. With an early warning system, we obtain equilibria given in Table 5.

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 $t_0 \leq \tau \leq t_0 + T$. The change in these values is summarized below:

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

21. Not every firm can join this prenotification 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 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 nondisclosure agreements. We follow one strike and you're out policy."

Those customers who subscribe to such a system will apply a workaround to reduce the chance of successful exploitation of the vulnerability in their system. This, in turn, 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 10. *Given that full vendor disclosure is not the optimal disclosure policy (i.e.,* $\gamma k > \Omega$*), 1) if the vendor has incentive to release its patch immediately when there is no early warning system, the vendor releases its patch at* $p_v^* \geq p^* = t_0$ *with an early warning system, and 2) if the vendor does not have incentive to release its patch immediately when there is no early warning system, the vendor releases its patch at* $p_v^* > (\leq)\, p^*$ *when* $\Lambda_v > (\leq)\, p^*$ *with an early warning system.*

Proposition 10 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 grace period 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 the absence of it.

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

Proposition 11. *The use of an early warning system by a coordinator does not necessarily improve the social welfare.*

Contrary to 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.

6 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" [34]. This is the main reason why the government [11] and a consortium of software vendors and security research firms [21] have attempted to consolidate the multitude of "loosely organized" vulnerability disclosure policies [32]. Our results contribute to those efforts to resolve the most controversial issues surrounding the responsible vulnerability disclosure. First, none of the disclosure practices, immediate public, full vendor, or hybrid, is optimal all the time. This finding justifies why proponents of each practice can find cases to defend their arguments. However, we clearly show that only one disclosure practice is optimal in a given scenario. Second, the grace period provided to the affected vendor cannot be the same for each vulnerability. It depends on several factors. This partly explains why different disclosure policies allow different amounts of time, yet fail to choose the optimal grace period in all cases. Third, the optimal grace period can increase or decrease when the vulnerability is shared by another vendor. Although highlighted as an important issue [21, 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 guidelines 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 [10], others do not provide any guideline to carry out this process effectively [21, 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 have 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 [36]. Our result indicates that an early warning system might not be beneficial even if leakage of information is prevented.

In spite of our best efforts, our model has 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. On the other hand, some parameters need to be collected from the vendors and, hence, may lead to an incentive compatibility issue. Although the issue is important, it is beyond the focus of this study and an avenue for the future research.

Although we model the vulnerability identification as a stochastic event, 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 nonzero vulnerability handling cost to the coordinator can be easily incorporated into our model without changing our results.

Our model does not explicitly capture the incentives of attackers, such as their desires and skill levels, or desirability of targets. We excluded these factors to focus on the dynamic interactions among the identifier, the coordinator, and the vendor in handling vulnerability knowledge.

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 [31], [35]. Since product liability laws imply a higher β in the 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, the coordinator's disclosure policies are crucial in promoting responsible behavior on the 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 always 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 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 also has a profound implication on open-source software development. As the software industry moves toward 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 the proprietary patch development cost. This benefit can be used as another justification for open-source software.

7 Conclusions

In the midst of various disclosure policies for the vulnerability disclosure process proposed by different stakeholders, this research sought to identify the optimal disclosure policy that minimizes the social loss. We found that none of the disclosure policies, immediate public, full vendor, or hybrid, is always optimal. Characteristics of the vulnerability—risk, user population—and vendor's incentives determine the optimal vulnerability disclosure policy. Unlike some existing vulnerability disclosure mechanisms which fail to motivate vendors to release a patch in some cases, responsible vulnerability disclosure that we identify ensures the release of a patch for the vulnerability all the time. In contrast to a widely held belief that allowing the vendor a certain time to develop its patch and making public disclosure if a patch is not released by the end of the grace period effectively motivates and gives incentive to the vendor to develop its patch, we show that restricting the vendor with a time constraint may not ensure the release. In such cases, placing no time restriction actually guarantees the release of a patch. On the other hand, we showed that the coordinator's policy is irrelevant when the vendor cannot delay the release of a patch for its own benefit. Except in these two cases, we show that the coordinator guarantees the release of a patch by setting a finite grace period. Overall, the immediate public disclosure, full vendor disclosure, and hybrid disclosure can all be the equilibria in responsible vulnerability disclosure.

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 can ignore 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.

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

We find that with an early warning 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 an early warning system.

References

[1] "ASPR Notification and Publishing Policy," Across, www.acrossecurity.com/asprNotificationAndPublishingPolicy.htm, 2004.

[2] W.A. Arbaugh, W.L. Fithen, and J. McHugh, "Windows of Vulnerability: A Case Study Analysis," *Computer,* vol. 33, no. 12, pp. 52-59, Dec. 2002.

[3] A. Arora, A.R. Telang, and H. Xu, "Optimal Policy for Software Vulnerability Disclosure," *Proc. Workshop Economics of Information Security,* 2004.

[4] T. August and T.I. Tunca, "Network Software Security and User Incentives," *Management Science,* vol. 52, no. 11, pp. 1703-1720, 2006.

[5] "Organization for Internet Safety Issues a Public Comment Draft for Security Vulnerability Reporting and Response Guide,"

BindView, http://www.bindview.com/News/display.cfm?Release=2003/0604b.txt/, 2003.
[6] H.K. Browne, W.A. Arbaugh, J. McHugh, and W.L. Fithen, "A Trend Analysis of Exploitations," *Proc. IEEE Symp. Security and Privacy,* pp. 214-229, May 2001.
[7] H. Cavusoglu, H. Cavusoglu, and S. Raghunathan, "Economics of IT Security Management: Four Improvements to Current Security Practices," *Comm. AIS,* vol. 14, article 3, July 2004.
[8] H. Cavusoglu, H. Cavusoglu, and J. Zhang, "Security Patch Management: Can't Live with It, Can't Live without It," *Proc. Workshop Information Technology and Systems,* Dec. 2004.
[9] "Locking Windows," *CBS News.com,* 16 Jan. 2002.
[10] "Vulnerability Disclosure Policy,"CERT Coordination Center, 2000.
[11] J.C. Chambers and J.W. Thompson, "Vulnerability Disclosure Framework: Final Report and Recommendations by the Council," US Nat'l Infrastructure Advisory Council, 13 Jan. 2004.
[12] S. Culp, "Definition of a Security Vulnerability," MicrosoftTechNet, 2000.
[13] "CYBSEC Security Vulnerability Disclosure Policy," http://www.cybsec.com/vulnerability_policy.pdf, 2004.
[14] "Upcoming Advisories," eEyes, http://www.eeye.com/html/research/upcoming/index.html, 2004.
[15] D. Fisher, "CERT, Feds Consider New Reporting Process," *eWeek,* 24 Mar. 2003.
[16] T. Havana, "Communication in the Software Vulnerability Reporting Process," MA thesis, Univ. Jyväskylä, 2003.
[17] K. Kannan,, R. Telang, and H. Xu, "Economic Analysis of the Market for Software Vulnerability Disclosure, *Proc. 37th Hawaii Int'l Conf. System Sciences,* 2004.
[18] M. Laakso, A. Takanen, and J. Röning, "The Vulnerability Process: A Tiger Team Approach to Resolving Vulnerability Cases," *Proc. 11th FIRST Conf. Computer Security Incident Handling and Response,* 1999.
[19] "Procedures for Handling Security Patches," NIST 800-40, US Nat'l Inst. Standards and Technology, 2002.
[20] D. Nizovtsev and M. Thursby, "Economic Analysis of Incentives to Disclose Software Vulnerabilities," *Proc. Workshop Economics of Information Security,* 2005.
[21] "Guidelines for Security Vulnerability Reporting and Response," version 2.0, Organization for Internet Safety, 1 Sept. 2004.
[22] A. Ozment, " Bug Auctions: Vulnerability Markets Reconsidered," *Proc. Workshop Economics of Information Security,* 2004.
[23] A. Ozment, "The Likelihood of Vulnerability Rediscovery and the Social Utility of Vulnerability Hunting," *Proc. Workshop Economics of Information Security,* 2005.
[24] W. Pond, "Do Security Holes Demand Full Disclosure?" *ZDNet,* 15 Aug. 2000.
[25] E. Preston and J. Lofton, "Computer Security Publications: Information Economics, Shifting Liability and the First Amendment," *Whittier Law Rev.,* vol. 24, pp. 71-142, 2002.
[26] M.J. Ranum, "Vulnerability Disclosure—Let's Be Honest about Motives Shall We?" http://www.ranum.com/security/computer_security/index.html 2004.
[27] E. Rescorla, "Is Finding Security Holes a Good Idea?" *Proc. Workshop Economics of Information Security,* 2004.
[28] *Secure Business Quarterly,* special issue vulnerability disclosure, vol. 2, 2002.
[29] S. Schechter, "How to Buy Better Testing: Using Competition to Get the Most Security and Robustness for Your Dollar," *Proc. Infrastructure Security Conf.,* 2002.
[30] J. Schiller, "Responsible Vulnerability Disclosure: A Hard Problem," *Secure Business Quarterly,* vol. 2, no. 1-5, 2002.
[31] B. Schneier, "Bug Secrecy vs. Full Disclosure," *ZDNet TechUpdate,* 14 Nov. 2001.
[32] S. Shepherd, "Vulnerability Disclosure: How Do We Define Responsible Disclosure?" GIAC SEC Practical Repository, SANS Inst., 2003
[33] A. Stone, "Software Flaws: To Tell or not to Tell?" *Computer,* vol. 20, no. 1, pp. 70-73, 2003.
[34] A. Takanen, P. Vuorijarvi, M. Laakso, and J. Roning, "Agents of Responsibility in Software Vulnerability Processes," *Ethics and Information Technology,* vol. 6, no. 2, pp. 93-110, 2004.
[35] H.R. Varian, "Managing Online Security Risks," *New York Times,* 1 June 2000.
[36] J. Vijayam, "Bug Disclosure, Fix Process Improving," *Computerworld,* 10 Mar. 2003.

Hasan Cavusoglu received the PhD degree in management science with a specialization in MIS from the University of Texas at Dallas. He is currently an assistant professor of management information systems at Sauder School of Business, University of British Columbia. His current research interests are economics of information systems, economics of information security, and management of technology and information. He has published in the *IEEE Transactions on Engineering Management* and the *Communications of the AIS*. He is a member of AIS.

Huseyin Cavusoglu received the PhD degree in management science with a specialization in management information systems from the University of Texas at Dallas. He is currently an assistant professor of information systems and operations management at the School of Management at the University of Texas at Dallas. He has published in *Information Systems Research*, the *Communications of the ACM*, the *IEEE Transactions on Engineering Management*, *INFORMS Decision Analysis*, the *International Journal of Electronic Commerce*, and the *Communications of the AIS*. His major research interests are in the areas of information economics, assessment of the value of IT security, and IT security management. He is a member of AIS and INFORMS.

Srinivasan Raghunathan received the PhD degree in business from the University of Pittsburgh. He is currently an associate professor of management information systems in the School of Management at the University of Texas at Dallas. He has published in *Management Science*, *Information Systems Research*, the *Journal of Management Information Systems*, the *IEEE Transactions on Engineering Management*, the *IEEE Transactions on Knowledge and Data Engineering*, and the *IEEE Transactions on Systems, Man, and Cybernetics*. His research interests concern the economics of information systems.