

Optimal Policy for Software Vulnerability Disclosure

Ashish Arora, Rahul Telang, Hao Xu

H. John Heinz III School of Public Policy and Management

Carnegie Mellon University, Pittsburgh PA 15213

Email: {ashish; rtelang; xhao}@andrew.cmu.edu

Abstract

Software vulnerabilities represent a serious threat: most cyber-attacks exploit known vulnerabilities. Unfortunately, there is no agreed-upon policy for their disclosure – white-hats who discover vulnerabilities, security mailing lists and CERT follow different ad-hoc policies. This paper develops a framework to analyze the optimal timing of disclosure policy (time given to vendor to patch the vulnerability). Disclosure policy indirectly affects how the speed and quality of the patch that a vendor develops, and thus CERT and similar bodies acting in the public interest can use it to influence behavior of vendors and reduce social cost. We formulate a game-theoretic model involving a social planner who sets disclosure policy and a vendor who decides on patching. We show that vendors always choose to patch later than a socially optimal disclosure time. The social planner can optimally shrink the time window of disclosure to push vendors to deliver patch in a timely manner. We extend the basic model in a number of directions, most importantly, allowing for the proportion of users implementing patches to depend upon the quality of the patch, which is itself a choice variable for the vendor. Our paper provides a decision framework for understanding how disclosure timing may affect vendor's decision and in turn, what should a policy maker do.

Keyword: Software Vulnerability, Disclosure policy, instant disclosure, patching, game theory, CERT.

*The authors thank the participants at the Third workshop on Economics and Information Security (WEIS 2004), Minneapolis and Ninth INFORM Conference on Information Systems and Technology (CIST) 2004, Denver for their valuable feedback.

“First, the Nation needs a better-defined approach to the disclosure of vulnerabilities. The issue is complex because exposing vulnerabilities both helps speed the development of solutions and also creates opportunities for would be attackers.”

The National Strategy to Secure Cyberspace, (2003: p 33)

1. Introduction

Information security breaches pose a significant and increasing threat to national security and economic well-being. In the Symantec Internet Security Threat Report (2003), each company surveyed experienced on an average about 30 attacks per week. These attacks often exploit software defects or vulnerabilities.¹ Over the last few years, the number of vulnerabilities found and disclosed has exploded. A recent report (Symantec, 2003) documents 2,524 vulnerabilities discovered in 2002, affecting over 2000 distinct products, an 81.5% increase over 2001. The CERT/CC (Computer Emergency Response Team/Coordination Center) received 3,782 reports of vulnerabilities in the year 2003 alone and has reported more than 82,000 incidents involving various cyber attacks. Anecdotal evidence suggests that losses from such cyber-attacks can run in the millions.² Software vendors, including Microsoft, have announced their intention to increase the quality of their products and reduce vulnerabilities. Despite this, it is likely that vulnerabilities will continue to be discovered and disclosed in foreseeable future.

1.1 Vulnerability Disclosure Policies

There is considerable debate not only about the reasons for software vulnerabilities but also about how they should be disclosed. Generally, vulnerability discoverers report (or are expected to report) vulnerabilities to vendors and keep it secret to allow time for vendors to develop a patch³. The argument is that the vendor would come up with a workaround or a patch and then make the vulnerability public, in due course, balancing costs of patching and disclosure with the benefits. However, many discoverers came to believe that often disclosure was excessively delayed or inadequate, and the quality of patches unsatisfactory. This led to the creation of full-disclosure mailing lists in late 90's, such as “Bugtraq”.⁴

Proponents of full disclosure claim that the threat of instant disclosure increases public awareness, makes

¹ The shutting down of the eBay and Yahoo! websites due to hacker attacks and the Code Red virus, which affected more than 300,000 computers are just two well known examples where software defects were exploited.

² For example, CSI (Computer Security Institute) and FBI estimated that the cost per organization across all types of breaches was around \$ 1 million in year 2000.

³ See, <http://www.usenix.org/publications/login/1999-11/features/disclosure.html>.

⁴ Our focus here is on “when” rather than “how much” information is disclosed.

as much information public as needed for users to protect themselves against attacks, puts pressure on the vendors to issue high quality patches quickly, and improves the quality of software over time.

However, many believe that the disclosure of vulnerabilities, especially without a good patch, is dangerous for it leaves users defenseless against attackers. Richard Clarke, President Bush's former special advisor for cyberspace security, criticizing full disclosure said: "It is irresponsible and sometimes extremely damaging to release information before the patch is out."⁵

While Bugraq tends to favor full and quick disclosure, organizations like CERT/CC argue for a more cautious approach. CERT traditionally has played a very important role in disseminating the vulnerability information to public. After learning of a vulnerability, CERT contacts the vendor(s) and provides a time window to patch the vulnerability; de facto but unofficial policy is to give vendors 45 days. After that the vulnerability is publicly disclosed, usually with a patch also being available. Since CERT is a non-profit, government funded entity, it is expected to be acting in the interest of society as a whole.

In the meantime, other organizations are proposing their own different policies. For example, OIS (Organization for Internet Safety)⁶ which represents the consortium of 11 software vendors, has suggested its own guidelines. In addition, firms such as iDefense are proposing market mechanisms where they propose to buy vulnerability information from users and use that information to protect their clients. They also have their own policies for disclosing vulnerabilities. However, it is not clear whether such policies maximize the social benefits. Kannan and Telang (2004) also argue that such market based mechanism for reporting and disclosing vulnerability information can be socially harmful unless proper disclosure guidelines are in place.

1.2 Research Questions

In summary, we lack a conceptual framework that would analyze and develop disclosure policy. As the citations indicate, the public policy problem is real and likely to become ever more important over time.⁷

Thus, the goal of this paper is to develop a theoretical framework to design an optimal policy for

⁵ See [http://www.blackhat.com/html/bh-usa-02/bh-usa-02-speakers.html#Richard Clarke](http://www.blackhat.com/html/bh-usa-02/bh-usa-02-speakers.html#Richard%20Clarke).

⁶ Members include @stake, BindView, Caldera International (The SCO Group), Foundstone, Guardent, ISS, Microsoft, NAI, Oracle, SGI, and Symantec. For a complete document, refer to <http://www.oisafety.org>

⁷ See also the debate between Robert Graham and Bruce Schneier at <http://www.robertgraham.com/diary/disclosure.html>. See also Preston and Lofton (2002).

vulnerability disclosure. In particular, we examine how long a vendor should be allowed to keep a vulnerability secret, in order to balance the need to protect users while providing vendors with incentives to develop a patch expeditiously. Our framework then can be used to analyze various policies proposed by different parties, and to suggest improvements in policies of entities acting on behalf of society at large, such as CERT.

Optimal disclosure policy depends upon the behavior of vendors, of potential attackers, and of users. In this paper, vendors are assumed to minimize costs, and hence their patching behavior depends upon the cost of developing a patch and the fraction of the losses of users that the vendor internalizes. Disclosure policy affects expected user loss, and thus, indirectly affects the vendors patching behavior as well. We do not model user or attacker behavior in detail. Attacker behavior is captured by the probability that potential attackers will discover a vulnerability. We do not explicitly model how users decide whether and when to apply the patch. However, in an extension of the basic model, we do allow the probability that users patch to vary with the quality of the patch. Finally, we focus on only one aspect of disclosure policy, namely the timing of disclosure, and leave for future research the question of how much information is disclosed.⁸

We show that a commitment to early disclosure by a “social planner” is indeed an effective way of prompting vendors for a quicker patch, although it is not always socially beneficial. We then extend our model to include the case when patching time is stochastic. In an important extension, we show that when users are slow to apply patches, social planner is better off increasing the optimal patching time and vendors also delay their patch appropriately. Finally, we explore the tradeoff between patching time and quality of the patch when higher quality increases the rate of patch implementation.

The rest of the paper is organized as follows. In section 2, we review relevant work on issues related to software vulnerability. We present the basic economic model in section 3 and the choice of the socially optimal disclosure window in section 4. Section 5 extends the basic model to allow for uncertainty in patching time and incorporate diffusion of patching such that only a portion of customers apply the patch when it is made available, and the rest gradually apply patch. Concluding remarks and implications of

⁸Implicitly, we assume that enough information is disclosed to enable potential attackers to exploit it.

results are presented in section 6 and 7.

2. Prior Literature

This paper contributes to the nascent literature on the economic and policy aspects of cyber-security, hitherto a near exclusive domain of computer scientists and technologists. For example, Howard (1998) provides a taxonomy of computer attacks and classification of intrusions. Lipson (2002) provides an overview of technical approaches and policy implications for cyber attacks. Related empirical work has been devoted to trend analysis of vulnerabilities. Shimell and Williams (2002) present a framework for trend analysis. Arbaugh et al (2000) propose a life cycle model for vulnerability analysis and show how frequently vulnerability is exploited from the time it is made public.

Our approach is in the spirit of Gordon & Loeb (2002), who develop an economic model for (privately) optimal information security investment decisions, whereas we analyze socially optimal disclosure policy that takes into account the privately optimal decision making process of the vendor. Arora, Caulkins and Telang (2003) model how the possibility of patching affects incentives of software vendors to invest in quality. Varian (2000) points out that a key policy aspect of managing information security is to align legal liability to rest with the party that has the lowest cost of preventing a loss. In our model, the vendor internalizes a part of the customer's losses, which allows for imperfect liability.

Only a few papers have analyzed economic issues related to problems in the vulnerability disclosures. Kannan and Telang (2004) show that the market for software vulnerability would have an adverse impact on social welfare if the market does not follow some agreed upon policies for disclosing vulnerabilities. Camp and Wolfram (2000) describe a means for creating a market for vulnerabilities in order to increase the security of systems. Rescorla (2003) argues, using the vulnerability reporting data, that disclosing information about security holes is a bad idea. In another paper, Rescorla (2004) shows that users are generally slow to patch their systems. Cavusaglu et. al. (2003) show that securities breaches negatively affect the market value of firms. Telang and Wattal (2004), on the other hand, show that software vendors in whose products vulnerabilities are disclosed tend to suffer declines in market value. Arora *et al.* (2004) empirically analyze how vendors respond to different disclosure policies and find evidence indicating that

early disclosure prompts vendors to patch more quickly than otherwise.⁹

3. Model

There are four major participants in our model – a “social planner”, a vendor, users (who are customers of the vendors products) and attackers.¹⁰ The social planner chooses a disclosure policy (i.e., the latest a vulnerability must be disclosed) to minimize total social cost. The vendor responds to change in disclosure policy by allocating resources for developing a patch. Users incur loss when the vulnerability in their system is exploited by attackers. Our model deals with the “representative user”, but is easily expanded to allow for different types of users. Attackers exploit the vulnerability when they become aware of it, if customers have not implemented the patch, or if the patch is not available.

We model a situation where a vulnerability is discovered by a benign discoverer (also called a whitehat, and different from the vendor or attackers) and is reported to a social planner (like CERT).¹¹ The social planner passes this information to the vendor and also sets the disclosure time. We allow the vendor to make a one-time decision on when to patch upon the discovery of the vulnerability. This assumption makes sense if the vendor has to commit resources for patching for a period of time. However, implicitly it assumes that the vendor releases a patch as soon as it is ready. The analysis of separating the patch development and release decisions implies extending the model to allow for dynamic decision making. While this extension will significantly complicate the structure of the model, we conjecture no changes in the basic results. Therefore, we choose simplicity. For now, patching time is assumed to be deterministic and quality of patch is assumed fixed. We also assume that customers apply patch immediately upon the delivery of patch. We will relax these assumptions in section 5.

⁹ As this paper was being completed, we became aware of a recent working paper by Cavusoglu et. al. (2004), which also analyses the question of vulnerability disclosure. Although their approach is similar to ours, unlike us, they work with specific functional forms and analyze disclosure policy for different scenarios. As well, they do not provide comparative static results of how disclosure policy varies with factors such as the software life-cycle or the share of customer losses internalized by the vendor. Our paper also differs from theirs in that we also model the vendor decision on patch quality, and the impact on disclosure policy.

¹⁰ In economics, a “social planner” is not just a convenient way of thinking about socially efficient solution, but also of representing policy makers in an idealized form. Our intent is not to suggest Soviet type central planning.

¹¹ The goal of this model is to study how social planner balances between the tradeoff of late and early disclosure. Thus, if the vendor finds the vulnerability, it will act as if the official disclosure time were infinite. If the attacker finds the vulnerability, there is no interesting policy question either. Formally, this is as if the official disclosure time were zero.

We treat the disclosure policy as binary. Either all information is disclosed or none. Hence, a disclosure policy is the choice of a time T , such that during that time vulnerability information is kept *secret* from public and shared with only the vendor to allow it to develop a patch. Once time T elapses, the information is disclosed to the public irrespective of the availability of patch. Instant disclosure policy means $T = 0$ while secrecy policy implies a $T = \infty$.



Figure 1. Software Life Cycle

In figure 1, at time ‘0’ the product is released and used by users.¹² A benign user discovers the vulnerability at t_0 . Social planner allows vendor a time window T that this vulnerability is kept secret no later than time $T + t_0$ and disclosed after that. Vendors provide a patch for this vulnerability at a calendar time $\tau + t_0$, possibly after disclosure. Attackers can potentially discover the vulnerability at some time s . Note that τ , T and s are the time windows of patch development, disclosure by social planner and rediscovery by attacker respectively, measured from the time the vulnerability is first discovered. The product is assumed to become obsolete (life cycle ends) at t_l .

We assume that attackers can exploit an un-patched vulnerability instantly upon its disclosure. Thus, attackers might find and exploit it at time $s + t_0$ or at time $T + t_0$, whichever is earlier. Estimates suggest that about 60% of the documented vulnerabilities can be exploited almost instantly, either because exploit codes are widely available or because no exploit tool is needed (Symantec, 2003). Modifying our model to allow for some period of exploit-tool development is straight-forward and yields little insight.

A key assumption in our analysis is that customers remain unprotected until a patch is released. In other words, in order to focus on the impact of patching, we ignore the real possibility that once a vulnerability is disclosed, users can take independent measures (workarounds such as adding a firewall to the system, shutting a port to defend against a specific threat) to avoid attacks or mitigate their impact. Allowing for this possibility will likely reduce the impact of disclosure policy on vendor patching

¹²We ignore diffusion of the product and assume that all users start using the product at time ‘0’.

behavior. In the extreme case, if customers can avoid any losses by taking precautions at low cost, patching becomes pointless. Similarly, we formally ignore the cost of patching to customers, although in a subsequent section we analyze the case where not all customers install patch right away upon release of the patch.

3.1 Vendor's Cost Function

When the social planner decides to disclose a vulnerability after T units of time elapse, the software vendor decides on allocating resources for patch development. The vendor's objective function (modeled here as a cost function to be minimized) has two terms. The first term is the cost of patch development. The more resources a vendor allocates to patching, the shorter is the time taken to patch. Therefore, the time window of patch development, τ , is a proxy for vendor's resource allocation. Let $C(\tau)$ denote the vendor's patch-developing cost. We assume that all else held constant, the quicker the patch, the higher are the costs, i.e., $\frac{\partial C(\tau)}{\partial \tau} < 0$. Also, marginal value of freed resources should be decreasing, as is commonly assumed. Therefore, we also assume $\frac{\partial^2 C(\tau)}{\partial \tau^2} > 0$.

The second component of cost for a vendor is the proportion of customer loss that vendor internalizes in the form of either a loss in reputation or loss in future sales. We represent this proportion by λ and call it internalization factor. Although vendors do not face any legal liability for the defects in their product, λ may also be interpreted as a legal liability if that changes in future¹³. The exact value of λ is an empirical question and may vary depending on industry and vendors. The expected customer loss, $L(\tau, T)$, a function of the time window of disclosure T and the time window for patching, τ .¹⁴

Hence, vendor's cost is:

$$V = C(\tau) + \lambda L(\tau, T) \tag{1}$$

3.2 Customer Loss Function

At this point, we need to be more specific about customer loss function, $L(\tau, T; X)$. For now we assume

¹³ The Uniform Computer Information Transactions Act (UCITA) says that vendors of software cannot be held liable for defects in the products they produce. Hence, standard defective product laws do not apply. UCITA is now passed in Maryland and Virginia. However, the debate over vendor liability is still going on.

¹⁴ They also obviously depend on vulnerability and customer specific factors, which we ignore her. For example, vulnerabilities in financial software usually cause more damage than those in personal education software. Similarly, vulnerabilities that are easier to exploit may be more dangerous. Finally, the damage also depends on the number of users affected and their size.

that all customers patch immediately after the patch is available, an assumption we shall relax in later sections. Customers suffer loss whenever a vulnerability is exploited by attackers. Thus customers suffer loss when either C1 or C2¹⁵ is true.

C1: An attacker finds the vulnerability on his own before patch is available.

C2: The vulnerability is disclosed without a patch.

Intuitively, we expect that longer a user is exposed to vulnerability without a patch, the greater the losses suffered. If users are exposed to vulnerability for duration x and the expected cumulative customer loss is $l(x)$ then we assume that $l(x)$ is increasing in x . We also assume that $l(x)$ is strictly convex, meaning that the longer the exposure time, the higher the incremental damage from every additional time unit of exposure. Recall that $l(x)$ represents cumulative expected user losses for the users as a whole. Thus, even though a given user, once attacked, may not suffer additional losses even as he continues to be exposed, as the period of exposure increases, so are the number of users who are likely to be attacked because the number of attackers aware of the vulnerability and in possession of the attack scripts increases. As Arbaugh et al (2000) note “intrusions increase once the community discovers a vulnerability and the rate of intrusions accelerates as news of the vulnerability spreads to a wider audience”.

To characterize the expected customer loss function $L(\tau, T)$, consider the following two cases:

C3: Patch is released before time window of disclosure T , i.e., $\tau < T$

C4: Patch is released after T , $\tau > T$.

When the patch is released before disclosure time (C3), customers suffer loss only if attackers find the vulnerability on their own prior to the patch (C1). Referring to Figure 1, $s + t_0$ is when attacker finds the vulnerability and $\tau + t_0$ is when patch is released. Customers are attacked between calendar time $s + t_0$ and $\tau + t_0$. Hence, expected cumulative customer loss is $l(\tau - s)$. On the other hand, if the patch is released after T (i.e., C4), there are two considerations: first, attackers find the vulnerability on their own (C1), and can exploit it for a period equal to $\tau - s$. Alternatively, at time T , attacker learns about the vulnerability when it is disclosed, and exploits it for a period equal to $\tau - T$, because the patch is made available only at τ .

¹⁵ Note that our precondition was that finder report a vulnerability to CERT, which is unknown to attackers. When attacker first finds the vulnerability, it can exploit customers.

To capture the uncertainty about when a vulnerability will also be discovered by an attacker, we assume that the time interval s it takes attackers to discover the vulnerability on their own is stochastic, with a distribution $F(s)$. Therefore, the probability that attacker does not find it within period T is simply $1 - F(T : t_0)$. Clearly, $F(s : t_0)$ is conditional on the vulnerability being *first* discovered by a whitehat at t_0 ¹⁶. We assume that $F(s : t_0)$ increases with t_0 because as attackers learn about the software, they are more likely to find the vulnerability. Thus, the expected customer loss can be written as follows:

$$L(\tau, T) = \begin{cases} \int_0^\tau l(\tau - s) dF(s : t_0), & \text{when } \tau \leq T \\ \int_0^T l(\tau - s) dF(s : t_0) + (1 - F(T : t_0))l(\tau - T), & \text{when } \tau > T \end{cases} \quad (2)$$

The first part of the function is customer loss when patch is released before disclosure time window T but attacker rediscovers the vulnerability at an earlier time s ($s < \tau$) and exposing customers to attacks for the duration $\tau - s$. The second part is when patch is released after disclosure, and attacker can either find it before T and attack for $\tau - s$ or find at time T when it is disclosed by social planner and attack for duration $\tau - T$.

Since l is convex, L is convex in time window of patch development τ (see proof in appendix 2). Moreover, since both C and L are convex in τ , the vendor's total cost V is also convex in patch development time τ . Therefore, for a given T , there always exists an optimal patching time for vendor.

3.3 Social Cost Function

The social cost is simply the sum of patch development cost and loss to customers:

$$S = C(\tau) + L(\tau, T) \quad (3)$$

The only difference between S and V is that the former includes the entire expected user loss, whereas the latter includes only a fraction λ . Thus, the vendor's cost function V converges to S when $\lambda = 1$ because now vendor internalizes the entire loss to customers and therefore interests of the vendor and the social planner are perfectly aligned. It is also immediate that S is convex in τ .

3.4 Social Planner's Decision

When the vendor internalizes only a portion of customer loss, i.e. $\lambda \in (0, 1)$, the vendor's incentives and the social planner's incentives are not aligned. Moreover, the social planner cannot choose τ , but instead

¹⁶ If the attacker is the first to discover the vulnerability then any disclosure policy T is moot.

can only choose a disclosure policy T^* that indirectly affects the vendor's choice of τ . Clearly, the sequence of the decision-making is critical. We assume that the social planner commits to a policy T and the vendor follows.¹⁷ This is the only interesting case because a policy must be announced in advance, and in practice CERT has a de facto policy, which is known to vendors. From equation (3), the first order condition (FOC) for social planner's optimal disclosure policy T^* is

$$\frac{\partial C}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial L}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial L}{\partial T} = 0 \quad (4)$$

Theorem 1 shows that there exists an optimal solution T^* for the social planner and establishes that there exists a unique disclosure policy such that $0 < T^* < \infty$. Also, corollary 1 implies that instant disclosure and secrecy policy are never optimal. All proofs are provided in appendix 2.

Theorem 1: *There exists an optimal solution T^* to equation (4).*

Corollary 1: *Neither instant disclosure nor infinite secrecy is optimal.*

In the following we analyze this solution and how it changes with various factors in more detail.

4. Characterizing optimal disclosure policy

Recall that the optimal disclosure time window T^* depends on vendor's reaction to T^* . Hence we first outline the vendor's reaction function to disclosure policy T .

4.1 Vendor's Reaction to Disclosure Policy T

The vendor chooses τ to minimize its total cost given disclosure time T . Since the vendor's cost is convex in τ , there exists a solution for vendor's cost-minimization problem. The first order condition for cost minimization, which implicitly defines the optimal patching time τ^* as a function of T is:

$$\frac{\partial C}{\partial \tau} + \lambda \frac{\partial L}{\partial \tau} = 0 \quad (5)$$

Let τ_i and τ_s correspond to the optimal patching time given instant disclosure ($T = 0$) and infinite secrecy policy (i.e., $T = \infty$), respectively. We first show, as many full disclosure proponents believe, that reducing T does push vendors to patch more quickly, but only if $T < \tau_s$ as proposition 1 formalizes.

Proposition 1: *The vendor's optimal patching time τ^* is bounded within $[\tau_i, \tau_s]$. For $T \in [0, \tau_s)$, the*

¹⁷ The case when vendor decides first and social planner follows is unrealistic and uninteresting. The case where both the vendor and the social planner move simultaneously has a trivial result, with $T = \tau$ (see appendix I).

vendor always patches after the disclosure time T i.e., $\tau > T$. Early disclosure T pushes the vendor to patch earlier.

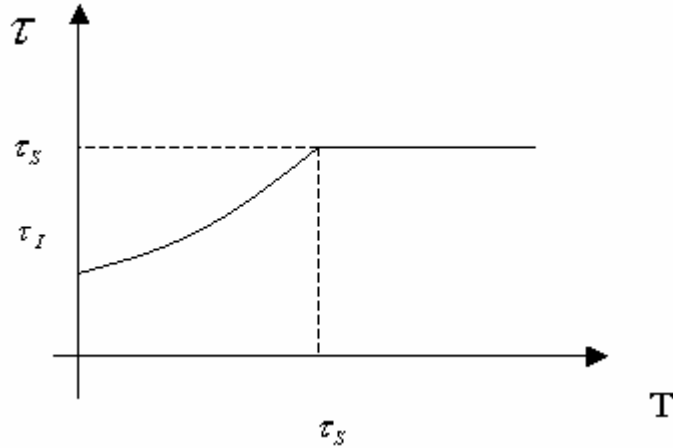


Figure 2 Vendor's Patching Time as Function of T

4.2 Factors affecting the optimal disclosure policy, T^*

The social planner will choose a T^* that will minimize the social cost, taking into account how the vendor will respond, i.e., taking into account that τ^* is a function of T . Figure 2 illustrates the vendor's reaction to disclosure policy T . The vendor's optimal patching time increases in T and is always greater than T until T reaches τ_S . Note that $T = \infty$ is useless because the vendor would anyway patch at τ_S . Also, given a disclosure window T , $T \in [0, \tau_S)$, the vendor will always patch after disclosure, since $\lambda \leq 1$.

4.2.1 The internalization factor λ .

The lower is the value of λ , the fraction of the users' loss internalized by the vendor, the lower is vendor's willingness to patch. Since vendor does not patch aggressively, and disclosure without a patch is socially harmful, social planner also has no choice but to increase the disclosure window T . Therefore, it follows that increase in λ will cause a vendor to patch earlier because it now internalizes a larger fraction of the customer's losses. Hence, social planner will also reduce disclosure time T to increase vendor's pace of patch delivery. Figure 3 shows that as the internalization ratio increases, both the patching time and the disclosure time fall, and the gap between the two diminishes. This is formalized in proposition 2.

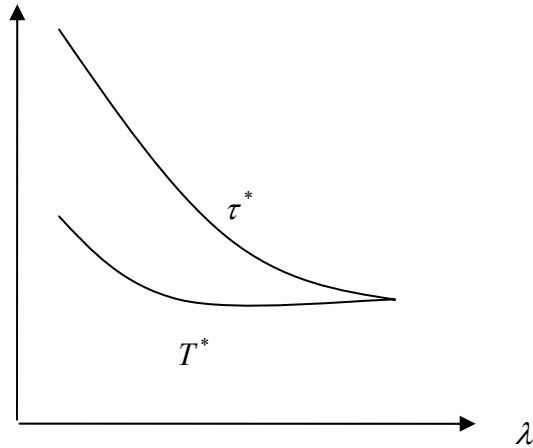


Figure 3: Optimal Disclosure Policy and Optimal Patching Time as Functions of λ

Proposition 2: *A higher internalization ratio, λ , implies a lower socially optimal disclosure window, T .*

Recall that the vendor always patches after disclosure ($\tau > T$). Thus, there is a period where customers are exposed. Setting T implies a tradeoff between reducing patching time and increasing customer exposure during the time between disclosure and the release of the patch. As λ increases, the gap between T and τ falls, and τ becomes more responsive to T . In short, social planner has higher leverage on vendor's actions when λ is higher. This is an important result because it points out that disclosure policy relies upon the sensitivity of the vendor to the losses its customers suffer. If the vendor is sensitive to its customers' losses, i.e., λ is high, then disclosure policy has more bite. Therefore, tighter disclosure windows can result. On the other hand, if the vendor is insensitive to its customers' losses (λ is small), tight windows will be socially costly by exposing users for a longer period. Put differently, if a vendor is not sensitive to customer losses, more direct regulation may be required to compel vendors to patch in a timely manner. Alternatively, legislation clarifying and strengthening vendor liability may be required.

There are two ways to a higher λ . One is when customers care and know about their losses and are willing to punish the vendor for the vulnerability and lack of patch. Preliminary results in Telang and Wattal (2004) indicate that stock prices for vendors do appear to fall when vulnerabilities in their product are reported. Interestingly, the loss in market value is larger when a vulnerability is disclosed without a patch. We conjecture that λ is higher, the greater is competition among vendors. Similarly, larger users are more likely to contract with vendors about the patching support. Thus, vendors whose market base

consists of large users will have higher λ .

4.2.2 Lifecycle of the product t_0 .

How does disclosure policy change when the vulnerability is discovered early in the life cycle of the product? All else equal, the threat of attackers discovering the vulnerability early in the software product lifecycle is smaller. This may be because, early in its life cycle, a product is less well understood, and exploit tools are less likely to be widely available and only sophisticated hackers can find and exploit the vulnerabilities.¹⁸ Therefore, as we show in Proposition 3, the social planner should give vendors more time for developing patch early in the lifecycle of the product. More generally, any factor that causes the probability of discovery, $F(s)$ to decrease will have the same effect. In anticipation of higher (lower) rediscovery probability, social planner should reduce (increase) the disclosure window, T .

Proposition 3: *The earlier a vulnerability is discovered in the product life cycle (lower t_0), the higher is the socially optimal disclosure time (T) and the patching time (τ), i.e., both are decreasing in t_0 . More generally, a decrease in the probability of discovery of the vulnerability by attacker, $F(s)$, increases the socially optimal disclosure window, T and patching time, τ .*

5 Extensions

5.1 Stochastic Patching Time

The basic model assumes that vendor determines exactly when to patch the vulnerability. In reality, vendor can only allocate resource such as people and computing power, but the actual patching time is uncertain. Specifically, if one lets the patch development time, ω , be stochastic, with a distribution function $H(\omega : \tau)$ where $E(\omega) = \tau$. The vendor controls the mean patching time τ . The greater the resources allocated for patching, on average, the earlier the patch is delivered. We assume that vendor chooses the mean patching time, τ to minimize the following cost function:

$$V = C(\tau) + \lambda \int_{t_0}^{t_1} L(\omega, T) dH(\omega : \tau) \quad (6)$$

As before, vendor comes to know about the vulnerability at t_0 and invests resources to release the patch in

¹⁸ A recent study based on honeypot data indicates that the older the Linux distribution, the more likely it was to be compromised if left unpatched. (<http://www.honeynet.org/papers/trends/life-linux.pdf>).

expectation at time $\tau + t_0$, but the actual patch can come at any time from t_0 to t_1 , so the consumer loss function is as given in the second term of (6). Analogously, social cost differs from vendor cost only in how much vendor internalizes the loss to customers:

$$S = C(\tau) + \int_{t_0}^{t_1} L(\omega, T) dH(\omega : \tau) \quad (7)$$

As (6) and (7) suggest, allowing for uncertainty will not affect our results.¹⁹

5.2 Customers do not patch instantaneously:

Thus far we have assumed that all customers would patch immediately after the patch is available. A recent .NET passport vulnerability is a good example. A fix on the server side stops the invasion and customers need no patch. In these cases, the basic model is sufficient.²⁰ However, many vulnerabilities require customers to download and apply the patch. Not all customers apply patches immediately after it is available (Rescorla 2003). It is reported that six months after the DDOS attacks that paralyzed several high-profile Internet sites, more than 100,000 machines were detected still not patched and vulnerable (InternetNews.com, 2000).

There are several reasons why all users may not patch their systems immediately. First, it takes time to disseminate the patching information to all users. Second, some customers lack the requisite computer skills. Patches can also be costly to apply because users may have to reboot their systems suspending many legitimate applications for some time. Finally, many users believe that patches are poor quality which may lead to more attacks. An example of a poor quality patch is the Microsoft patch for a vulnerability identified as CVE-2001-0016 (Beatie, et al, 2002). The initial patch disabled many updates of service pack 2 of Windows NT, making the patched system even more vulnerable to attacks.

Clearly, how quickly customers apply patches is critically dependent on two factors: the time elapsed since the patch is released (denoted by z) and the quality of the patch (denoted by q). We first consider the case when quality is exogenously specified. Later, we shall extend the model to allow the quality of patch to affect the diffusion of patching. For simplicity, we assume that patching time is deterministic.

Recall that we used $l(t)$ to denote the cumulative customer loss if they are exposed for a duration t .

¹⁹The proofs are available upon request.

²⁰See <http://www.internetweek.com/security02/showArticle.jhtml?articleID=10817869>

Before the release of patch, no customer is protected, therefore all the loss materializes. But after the release, some customers apply patches and are protected. The rest still suffer losses. However, since a patch may disclose additional details about the vulnerability, we allow the losses incurred (by unpatched users) post release of the patch to differ from those before the patch. Let z denotes the time since the release of the patch and $p(z)$ denote the cumulative proportion of customers that have applied that patch after z units of time have elapsed since the patch was released, then instantaneous post patch loss is $(1 - p(z))\tilde{l}(z)$, where both functions are increasing in z and $\tilde{l}(z)$ is the post patching customer loss for the users who did not patch. Thus, the customer loss function becomes

$$\begin{aligned}
& \begin{cases} \int_0^\tau l(\tau - s)dF(s : t_0) + \int_0^{t_1 - \tau} \tilde{l}(z)(1 - p(z))dz = L(\tau : T) + \tilde{L}(\tau; T), & \text{when } \tau \leq T \\ \int_0^T l(\tau - s)dF(s : t_0) + (1 - F(T : t_0))l(\tau - T) + \int_0^{t_1 - \tau} \tilde{l}(z)(1 - p(z))dz = L(\tau; T) + \tilde{L}(\tau; T), & \text{when } \tau > T \end{cases} \\
& (8)
\end{aligned}$$

The only difference from the basic model is that there is an additional term which captures post patching losses as well. Since the life-cycle of the product, t_1 is finite, the post patching cost is reduced when the patch is delayed. Hence, if some users delay patching, the vendor will optimally delay the release of the patch and the social planner would also increase the disclosure window. In the extreme case, where no one patches, clearly, the best action for both the vendor and the social planner is to never release the patch (or delay it infinitely).

Formally, one way to express the delay by users in implementing patches is through a downward shift in $p(z; x)$, where x is any factor that affects the willingness of users to patch, such as the quality of the patch. Another interpretation of x includes including technologies for “pushing” patches to hosts on a network, which can lead to quicker patching by users, represented here by a downward shift in $p(z)$.

Proposition 4: *When users do not patch instantly, the vendor slows patch development and social planner allows more time before disclosure.*

It is also plausible that the proportion of post-patching costs internalized by the vendor differ from the fraction of pre-patching costs internalized. For instance, the vendor is even less likely to be liable for losses arising from the failure of the user to patch. Though we do not discuss this in detail here, it is

easy to see that allowing for a different internalization factor will not significantly change the analysis.

The discussion here also points to the fact that slower diffusion of patching is socially costly. Therefore, a key problem for both vendors and social planner is to institute policies (including the use of technology) that increase the rate of adoption of patches. While a detailed analysis of this issue is beyond the scope of this paper, in the following we consider one aspect, namely the quality of patches.

5.3 Patch Quality and patching by users

It is plausible that patch quality is a critical factor in determining how quickly customers will apply the patch. Hence, we extend the model to allow vendor to determine both patching time τ and quality of patch, q . Quality has two opposing effects on the vendor's cost function; given by $V = C(\tau, q) + L(\tau, q; T) + \tilde{L}(\tau, q; T)$; and hence also, the vendor's behavior. Higher patch quality q implies higher patch development costs C . Further, it is plausible increasing patch quality should be more costly the shorter is the time allocated for the patch: *shortening* the patch development cycle and increasing the quality of the

patch draw upon scarce resources, and thus it is natural to conjecture that $\frac{\partial C}{\partial q} > 0$ & $\frac{\partial^2 C}{\partial \tau \partial q} \leq 0$. It is also

plausible that at any time z , the better the patch quality, the proportion, $p(z, q)$, of customers that have patched is also greater, i.e., $p(z, q)$ is increasing in q . On the other hand, it is clear from (8) that the marginal customer loss from a small delay in patching is greater, the greater is the quality. Thus,

$\frac{\partial^2 V}{\partial \tau \partial q}$ is of indeterminate sign. Note that $\frac{\partial^2 V}{\partial \tau \partial q} = \frac{\partial^2 C}{\partial \tau \partial q} + \lambda \frac{\partial^2 l}{\partial \tau \partial q} + \lambda \frac{\partial^2 \tilde{l}}{\partial \tau \partial q}$. It is easy to show from (8)

that $\frac{\partial^2 l}{\partial \tau \partial q} = 0$ & $\frac{\partial^2 \tilde{l}}{\partial \tau \partial q} > 0$. Since $\frac{\partial^2 C}{\partial \tau \partial q} < 0$, if the patch development cost effect, $\frac{\partial^2 C}{\partial \tau \partial q}$, dominates,

$\frac{\partial^2 V}{\partial \tau \partial q} \leq 0$. However, on the other hand, if the post-patch loss effect, $\frac{\partial^2 \tilde{l}}{\partial \tau \partial q}$, dominates, $\frac{\partial^2 V}{\partial \tau \partial q} \geq 0$. Note

that the higher λ is, the more likely it is that the post-patch loss effect dominates.

To understand how endogenous quality affects the results, consider first the impact of increasing the disclosure window, T , on τ and q . Since the disclosure window does not affect post-patching losses directly, increasing it will lead to an increase in τ . The impact of higher τ on q depends on the sign of

$\frac{\partial^2 V}{\partial \tau \partial q}$. If the post-patch loss effect dominates, then $\frac{\partial^2 V}{\partial \tau \partial q} > 0$ and the optimal strategy for the vendor is

to reduce q (Recall that vendor is minimizing cost). On the other hand, if the patching cost effect

dominates then $\frac{\partial^2 V}{\partial \tau \partial q} < 0$, the vendor will increase q .

Proposition 5: *When patch quality is endogenous, an increase in T will increase τ . If $\frac{\partial^2 V}{\partial \tau \partial q} \geq 0$, quality*

will decrease and, conversely, quality will increase if $\frac{\partial^2 V}{\partial \tau \partial q} \leq 0$.

Proposition 5 appears counter-intuitive. It is commonly believed is that providing more time to vendors should lead to increase in the quality of the patch. However, from proposition 5, it is clear that this view is only partially true. If the patch development cost is the major cost, then giving vendor more time does lead to higher quality. On the other hand, if the post-patch customer loss is the more significant cost for a vendor, then giving vendor more time leads to lower and not higher quality. The intuition is as follows. An increase in τ (due to a longer disclosure window) has two opposing effects on the marginal benefit from increasing quality. On the one hand, it reduces the marginal cost of quality (i.e., $\frac{\partial C}{\partial q}$ is smaller the higher is τ) but on the other hand, it also reduces the marginal benefit of quality (i.e., the reduction in post-patching loss due to higher quality is also smaller when the patch is delayed). When the post-patching loss is the dominant factor, increasing the disclosure window will reduce patch quality.

Given how vendor behavior changes with change in T , we can examine what should be optimal disclosure policy when endogenous quality choice is also a decision variable. Let $\tau(T)$ and $q(T)$ be the vendor's choice of τ and q for a given T , and let T^* be the socially optimal T , then T^* is implicitly given by the social planner's first order condition

$$dS/dT = \{C_\tau + [L_\tau + \tilde{L}_\tau]\} d\tau/dT + (C_q + \tilde{L}_q) dq/dT + L_T = 0 \quad (9)$$

We want to understand how optimal policy is different when q is a decision variable compared to the case when q is exogenous (as in the previous sections). To compare to the exogenous case, let q be exogenously fixed at $q(T^*)$. If $T = T^*$, $\tau = \tau(T^*)$. However, since q is no longer endogenous, the derivative of S with respect to T is given by

$$\left. \frac{dS}{dT} \right|_{q \text{ is exog set} = q(T^*)} = \{C_\tau + [L_\tau + \tilde{L}_\tau]\} \frac{d\tau}{dT} + L_T \quad (10)$$

From the vendor's first order condition for q we know that $C_q + \tilde{L}_q = (1 - \lambda)\tilde{L}_q < 0$. Evaluated at T^* , $\tau(T^*)$ and $q(T^*)$, (10) is equal to $-(C_q + \tilde{L}_q) \frac{dq}{dT}$ which is negative if the post-patch loss effect dominates and positive if the patching cost effect dominates (Recall from proposition 5 that $dq/dT > 0$ if patching cost dominates and $dq/dT < 0$ if post-patching loss dominates). Since S is convex in T , the optimal T for the exogenous case will be greater than T^* if the post-patch loss effect dominates but less than T^* if the patching-cost effect dominates. This leads to the following proposition

Proposition 6: *The socially optimal disclosure window is longer with endogenous quality than when quality is exogenous if patching cost dominates the post-patching customer loss (or $\frac{\partial^2 V}{\partial \tau \partial q} < 0$), and conversely if the post-patching customer loss dominates patching cost (or $\frac{\partial^2 V}{\partial \tau \partial q} > 0$), the optimal disclosure window is smaller with endogenous quality.*

Intuitively, if increasing quality has a bigger impact on the cost of developing the patch rather than on the post patch user loss internalized by the vendor, it makes sense to encourage the vendor to improve quality by allowing the vendor more time. Conversely, if the post-patch user loss effect dominates, then quality will increase as the disclosure window shrinks. In this case, it makes sense to shrink the disclosure window to encourage higher quality. Note also that which impact dominates depends critically upon λ , the portion of the user loss internalized by the vendor. The smaller is λ , the more likely that the patching-cost effect will dominate. This is in line with the often heard view that a key problem with patching is the post-patching cost of customers. It also suggests that the current poor status of patch quality may be a reflection that vendors tend to only internalize a small proportion of customer loss. Proposition 6 suggests that, if λ is small, endogenous quality of patch implies an additional reason for allowing vendors more time to patch.

6. Implications of empirical studies and directions for further research

The goal of this paper is to develop a framework for policy makers and entities like CERT to determine

the optimal time to disclose a software vulnerability to the public. Any disclosure policy must balance the need to protect users against attackers and the need to prod vendors to develop patches expeditiously. Thus, disclosure policy depends upon the behavior of vendors, of potential attackers, and of users. In the following we comment on some empirical studies that have implications on our results and analyze how these studies fit into our framework.

Rescorla (2003) finds that the rate of patching can be alarmingly low. In his case study of the OpenSSL Remote Buffer Overflow vulnerability (exploited by the notorious *slapper* worm), two weeks after the release of the patch, 60% of the investigated servers did not patch. In the following weeks, administrators only made minimal progress in applying patches. In terms of our model, Rescorla (2003) finds that $p(z, q)$ is small. Thus our results imply instant disclosure would appear to be a bad idea. Rather, vendors should be given ample time to develop and test patches, which would also improve the uptake of patches by users. However, Rescorla's study does not capture the heterogeneity among the users. It is conceivable that all users who cared about the vulnerability in fact patch in time but the rest do not face significant consequence and hence choose not to patch.

In another interesting paper, Rescorla (2004) suggests that the probability that a vulnerability will be rediscovered by an attacker is very low. This means that $F(s)$ is very small. As suggested in the discussion of proposition 3, when $F(s)$ shifts downward (meaning reduction in the overall probability of discovery by blackhats) CERT should allow more time for patch development. Consider the extreme case when $F(s) = 0$. There is no risk that attacker is ever going to exploit the vulnerability unless the vulnerability itself is disclosed. In this case, the best policy for CERT is obviously never to disclose. However, though interesting, Rescorla's results are derived in an indirect fashion and are subject to a variety of qualifications. Regardless, these results and the implications of our model are clear: additional empirical research is urgently needed to provide reasonable estimates of $p(z, q)$ and $F(s)$. In addition, our model has pointed to the crucial role of λ , the proportion of user losses internalized by the vendor, and this remains an additional important area for further empirical research.

7. Conclusions

How and when vulnerabilities should be disclosed is an important policy issue. In this paper, we develop

a model that analyzes how a policy maker should go about setting a disclosure policy. Our framework analyzes the vendor behavior and the optimal policy in the shadow of what attackers and consumers are likely to do. Since the policy maker can only indirectly affect the patching behavior of the vendors, our model outlines how policy maker can optimally influence vendor behavior. An important objective in this paper is to formulate a general model, without restrictive functional form assumptions.

We derive a number of interesting results. We find, first and foremost, that as long as the vendor does not internalize all the losses suffered by users, the vendor will release the patch later than socially optimal. Further, optimal disclosure policy is to disclose the vulnerability sooner than the vendor would like in order to push the vendor to release the patch earlier. The optimal disclosure policy therefore trades off some loss from the exploitation of the vulnerability from disclosure against a delay in the release of the patch (which itself increases the risk of the vulnerability being discovered and exploited by malicious attackers). We find that, in general, both instant disclosure and secrecy policy are sub-optimal. We find that these results are robust to a number of extensions, including uncertainty in patching time, imperfect compliance by users to the patch, and endogenous variations in the quality of the patch. Interestingly, we identify conditions under which the quality of patches actually decreases if vendors are given more time to patch.

Our results are subject to a variety of qualifications. First, we do not allow patch release policy to vary with time. Thus, our model is best thought of relating to policy rather than a patch release decision support system. Second, we assume certain patterns of exploit behavior, and how these change with vulnerability disclosure. Third, we ignore defensive measures by users when informed of a vulnerability without a patch. It is entirely possible that different assumptions may lead to different conclusions about optimal disclosure policy, but our model can be tailored to reflect those differences without changes to the basic structure. Early disclosure proponents also argue that public disclosure, in the long run, would force vendors to provide secure software in the first place. Recent work by Telang and Wattal (2004) does indicate that vendors lose stock value when vulnerability is disclosed. This, in turn, should force vendors to release better product. In the current set-up we do not consider such strategic choices.

In this sense, our model highlights the key areas where additional empirical evidence is required, by

bringing out the key implications of the assumptions we have made. The contribution of this paper, therefore, lies not only in the specific results obtained but also in the framework developed that allows for stochastic discovery of vulnerabilities, uncertainty in patching time, and uncertainty in the installation of patches by users, and highlights the possibilities and limits of social disclosure policy.

References

- Arbaugh, W.A., Fithen, W. L. & McHugh, J. (2000), "Windows of Vulnerability: A Case Study Analysis", *IEEE Computer*.
- Arbaugh, W.A., Browne, H., McHugh, J & Fithen, W.L. (2001), "A Trend Analysis of Exploitations". *IEEE Symposium on Security and Privacy*. Oakland, California, USA.
- Arora, A., Caulkins, J.P., & Telang R. (2003), "Sell first, fix later: Impact of patching on software quality," Carnegie Mellon University, working paper, February.
- Arora, A., Krishnan R., Telang R., and Yang Y. (2004), "How quickly to they Patch? An Empirical Analysis of Vendor Response to Disclosure Policies", Carnegie Mellon University, working paper, December.
- Beattie, S., Arnold, S., Cowan, C., Wagle, P. & Wright, C. (2002), "Timing the Application of Security Patches for Optimal Uptime", *Proceedings of LISA '02: Sixteenth Systems Administration Conference*
- Camp, L. & Wolfram, C. (2000). "Pricing Security" In *Proceedings of the CERT Information Survivability Workshop*, Boston, MA Oct. 24-26. pp-31-39.
- Cavusoglu H., Cavusoglu H., Raghunathan S. (2004), "How should we disclose Software vulnerabilities?", *14th Annual Workshop on Information Technologies and Systems*, Washington D.C.
- Cavusoglu H., B. Mishra, Raghunathan S. (2004), "The Effect of Internet Security Breach Announcements on Market Value of Breached firms and Internet Security Developers", *International Journal of Electronic Commerce*, 9(1), 69-104.
- Gordon, L.A. & Loeb, M.P. (2002). "The Economics of Information Security Investment". *ACM Transactions on Information and System Security*, 5.
- Howard, J. (1998), "An Analysis of Security Incidents On the Internet," thesis, <http://www.cert.org/research/JHThesis/Word6/> (Accessed Oct 13, 2004.)
- Kannan, Karthik and Telang Rahul (2005), "Market for Software Vulnerabilities? Think Again", *Management Science (Forthcoming)*.
- Lipson, H. (2002), "Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues", *CERT/CC special report*
- NetworkMagazine.com (2000), "The Pros and Cons of Posting Vulnerabilities". <http://www.networkmagazine.com/article/NMG20001003S0001> (Accessed, Oct 13, 2004)
- National Strategy to Secure Cyberspace, 2003, <http://www.whitehouse.gov/pcipb/> (Accessed 17 Jan, 2005)
- Polk, T. (1993), "Automated Tools for Testing Computer System Vulnerability", *Technical Report NIST SP 800-6*, National Institute of Standards and Technology
- Preston, E. and Lofton, J. (2002). "Computer security publications: information economics, shifting liability and the first amendment", *Whittier Law Review*, 24, 71-142.
- Rescorla, E. (2003), "Security holes... Who cares?", *Proceedings of the 12th USENIX Security Conference*, August.

- Rescorla, E. (2004), "Is finding security holes a good idea?", *The Third Workshop on Economics and Information Security*. Minneapolis MN.
- Schechter, S.E. & Smith, M.D. (2003), "How Much Security is Enough to Stop a Thief?", *The Seventh International Financial Cryptography Conference*, Gosier, Guadeloupe, January.
- Shimeall, T. & Williams, P. (2002), "Models of Information Security Trend Analysis", *CERT/CC*
- Symantec Inc (2003), "Symantec Internet Security Threat Report". <http://www.symantec.com>
- Telang Rahul and Wattal Sunil (2004), "Impact on Software Vulnerability Announcements on the Market Value of Software Vendors – an Empirical Investigation", *Workshop on Economics of Information Systems*, Washington D.C.
- Varian, H.R. (2000), "Managing Online Security Risks," *The New York Times*,
<http://www.nytimes.com/library/financial/columns/060100econ-scene.html>. (Accessed Oct 13, 2004)

Appendix 1: Sequence of Actions: Vendor and Social Planner's Decision Game

The game between vendor and social planner involves three possible orders of moves. Here we show that if both move simultaneously or if the vendor moves first, the outcome is simply for the vendor to patch as if there were no disclosure policy at all. Let τ_s be the time a vendor would patch if $T = \infty$.

If the vendor leads, for any τ , social planner's best reaction is $T^* = \tau$. Note that any T less than τ is not optimal because customers incur more loss while T^* has no effect on τ ; any T larger than τ is not optimal either because after the availability of patch, social needs not to keep it a secret, on the contrary, social planner should inform the customers right away. Hence the equilibrium is (τ_s, τ_s) . Using the same logic, one can show that the optimal response functions will be as shown in figure A1 below. For any τ , social planner's best reaction is $T^* = \tau$. For any given any $T^* = \tau_s$, the vendor's best response is $\tau^* > T$ as we show in appendix 2. Hence, the equilibrium in a simultaneous move is (τ_s, τ_s) .

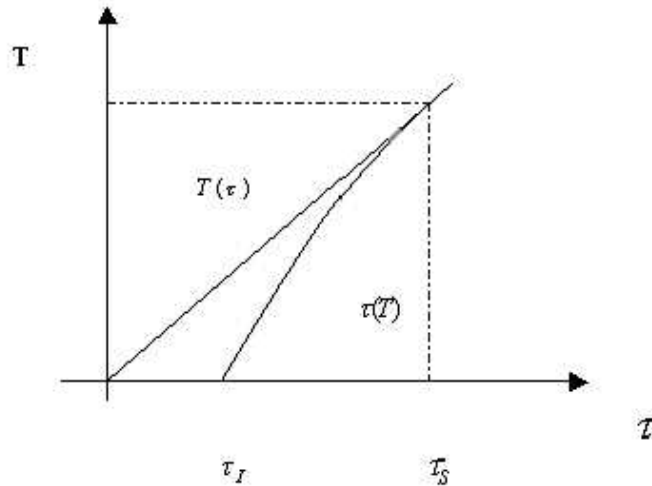


Figure A 1: Social planner and vendor's reaction functions

Appendix 2: The Model and Its Extensions

Customer loss function $L(\tau, T)$ is convex in patching time τ .

Proof: For ease of notation, define $l' = \frac{dl}{d\tau}$ & $l'' = \frac{d^2l}{d\tau^2}$. From (2), when $\tau > T$,

$$\frac{\partial L}{\partial \tau} = (1 - F(s : t_0)) \frac{dl(\tau - T)}{d\tau} + \int_0^T \frac{dl(\tau - s)}{d\tau} dF(s : t_0).$$

Hence, $\frac{\partial^2 L}{\partial \tau^2} = (1 - F(s : t_0)) \frac{d^2l(\tau - T)}{d\tau^2} + \int_0^T \frac{d^2l(\tau - s)}{d\tau^2} dF(s : t_0)$. Since l is increasing and convex in τ , we

have $\frac{\partial^2 L}{\partial \tau^2} > 0$. Similarly, one can show that when $\tau \leq T$, $L(\tau, T)$ is convex in patching time τ . **QED**

Proposition 1: Vendor's optimal patching time τ^* is bounded within $[\tau_l, \tau_s]$. For $T \in [0, \tau_s)$, the vendor always patch after the disclosure time T i.e., $\tau^* > T$. Early disclosure T pushes vendor to patch earlier.

Proof: For ease of notation, from now we define

$$L_1(\tau) = \int_0^\tau l(\tau - s) dF(s : t_0) \text{ and } L_2(\tau) = \int_0^T l(\tau - s) dF(s : t_0) + (1 - F(T : t_0))l(\tau - T) \text{ so that}$$

$$L(\tau, T) = \begin{cases} L_1(\tau), & \text{when } \tau \leq T \\ L_2(\tau, T), & \text{when } \tau > T \end{cases}$$

Proposition 1 has three major results. We will prove them one by one.

i) For $T \in [0, \tau_s)$, the vendor always patches after the disclosure time T i.e., $\tau^* > T$.

Proof: Suppose that $\tau^* \leq T$, recall from equation (2) that when $\tau^* \leq T$, loss to customers, $L(\tau, T) = L_1(\tau)$,

the same as that under secrecy policy when $T = \infty$. Hence, $\tau^* = \tau_s$, which contradicts the precondition.

Thus it follows that $\tau^* > T$.

ii) For $T \in [0, \tau_s)$, early disclosure T pushes vendor to patch earlier.

Proof: We want to show that for $T \in [0, \tau_s)$, $\frac{d\tau^*}{dT} > 0$. First, τ^* must satisfy the F.O.C of vendor's optimal

decision: $\frac{\partial V}{\partial \tau} = 0$. Differentiate both sides with respect to T :

$$\frac{\partial^2 V}{\partial \tau^2} \frac{d\tau}{dT} + \frac{\partial^2 V}{\partial \tau \partial T} = 0 \quad \text{Thus,} \quad \frac{d\tau^*}{dT} = - \frac{\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}. \quad \text{Differentiating } V \text{ w.r.t } \tau \text{ and } T \text{ and applying integration by}$$

parts, we have that $\frac{\partial^2 V}{\partial \tau \partial T} = \frac{\partial^2 L}{\partial \tau \partial T} = \lambda(F(T) - 1)l''(\tau - T) < 0$. Thus, we have $\frac{d\tau^*}{dT} > 0$. It follows that

$$\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}} = \frac{\lambda(1 - F(T))l''(\tau - T)}{\int_0^T l''(\tau - T)dF(s) + \lambda(1 - F(T))l''(\tau - T)} < 1$$

iii) *Vendor's optimal patching time is bounded between $[\tau_I, \tau_S]$*

Proof: When $T \geq \tau_S$, we have $\tau^* = \tau_S$. For $T < \tau_S$, from (2) we know that τ^* is increasing in T . Recall

that τ_I is optimal patching time when $T=0$. Thus, it follows that $\tau^* \geq \tau_I$. Thus, τ^* is bounded. **QED**

Theorem 1: We wish to show that there exists a point that satisfies the first-order condition for social optimality and is convex locally in T .

Proof $\frac{dS}{dT} = \frac{\partial C}{\partial \tau} \frac{d\tau}{dT} + \frac{\partial L}{\partial \tau} \frac{d\tau}{dT} + \frac{\partial L}{\partial T}$. The vendor's optimal τ , given T satisfies $\frac{\partial C}{\partial \tau} + \lambda \frac{\partial L}{\partial \tau} = 0$.

Substituting we get, $\frac{dS}{dT} = (1 - \lambda) \frac{\partial L}{\partial \tau} \frac{d\tau}{dT} + \frac{\partial L}{\partial T}$. We now show that $\frac{dS}{dT}$ is negative when $T=0$ and positive when $T = \infty$, which, assuming that S is smoothly differentiable, ensures that there exists at least one point such that $\frac{dS}{dT} = 0$. If this point is unique, S must be locally convex in T . Moreover, it is also easy to see

that if there are multiple points such that $T=0$, at least one such point S is locally convex in T .

1. For $\tau > T$, $\frac{\partial L}{\partial \tau} = (1 - F(T : t_0))l'(\tau - T)$ and $\frac{\partial L}{\partial T} = -(1 - F(T : t_0))l'(\tau - T)$. When $T=0$, $F(T : t_0) = 0$.

Further, from proposition 1.ii, we know that $\frac{d\tau^*}{dT} < 1$. Thus, for any $\lambda \neq 1$, we have

$$\frac{dS}{dT} = (1 - \lambda) \frac{\partial L}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial L}{\partial T} < (1 - \lambda) \frac{\partial L}{\partial \tau} + \frac{\partial L}{\partial T} = -\lambda l'(\tau - T) < 0.$$

2) When $T = \infty$, $L(\tau, T) = \int_0^\tau l(\tau - s)dF(s : t_0)$. Therefore, we have $\frac{\partial L}{\partial T} = 0$.

$$\frac{dS}{dT} = (1 - \lambda) \frac{\partial L}{\partial \tau} \frac{d\tau}{dT} + \frac{\partial L}{\partial T} > \frac{\partial L}{\partial T} = 0.$$

Proof of Corollary 1: Since $\frac{dS}{dT}$ is never 0 at neither $\tau = 0$ nor $\tau = \infty$. Hence, neither instant disclosure nor secrecy policy is optimal.

Proposition 2: *An increase in the internalization ratio, λ , reduces the socially optimal disclosure window, T .*

1) First, we prove that holding T constant, $\frac{d\tau^*}{d\lambda} < 0$. The optimal τ^* must satisfy $\frac{\partial V}{\partial \tau} = 0$. Differentiate

both sides with respect to λ we get $\frac{d\tau^*}{d\lambda} = \frac{-\frac{\partial^2 V}{\partial \tau \partial \lambda}}{\frac{\partial^2 V}{\partial \tau^2}}$. The denominator is positive at any interior

minimum. Thus, if the numerator is negative, the result follows. $\frac{\partial^2 V}{\partial \tau \partial \lambda} = \int l'(\tau - s) dF(s : t_0) > 0$. Thus,

$$\frac{d\tau^*}{d\lambda} < 0$$

2) We now prove that $\frac{dT^*}{d\lambda} < 0$. Let $G(T) = \frac{\partial S}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial S}{\partial T}$. The optimum T^* must satisfy $G(T) = 0$.

Differentiate both sides with respect to λ : $\frac{\partial G}{\partial \tau} \left(\frac{\partial \tau}{\partial T} \frac{dT}{d\lambda} + \frac{\partial \tau}{\partial \lambda} \right) + \frac{\partial G}{\partial T} \frac{dT}{d\lambda} + \frac{\partial G}{\partial \lambda} = 0$. Arrange terms and combine them

$$\left(\frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial G}{\partial T} \right) \frac{dT}{d\lambda} + \frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial \lambda} + \frac{\partial G}{\partial \lambda} = 0 \Rightarrow \frac{d^2 S}{dT^2} \frac{dT}{d\lambda} + \frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial \lambda} + \frac{\partial G}{\partial \lambda} = 0. \quad \text{Thus, } \frac{dT^*}{d\lambda} = -\frac{\frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial \lambda} + \frac{\partial G}{\partial \lambda}}{\frac{d^2 S}{dT^2}}. \quad \text{From}$$

proposition 1, $\frac{d^2 S}{dT^2} > 0$. Therefore, we only need to show that the numerator is positive.

i) We now show that $\frac{\partial G}{\partial \tau} < 0$. Recall that $\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}$.

$$\frac{\partial G}{\partial \tau} = \frac{\partial^2 S}{\partial \tau^2} \cdot \frac{\partial \tau}{\partial T} + \frac{\partial^2 S}{\partial \tau \partial T} = \frac{\partial^2 S}{\partial \tau^2} \cdot \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}} + \frac{\partial^2 S}{\partial \tau \partial T} = \lambda \cdot \frac{\partial^2 S}{\partial \tau^2} \cdot \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}} + \frac{\partial^2 S}{\partial \tau \partial T} = \frac{\partial^2 S}{\partial \tau \partial T} \left(1 - \lambda \frac{\frac{\partial^2 S}{\partial \tau^2}}{\frac{\partial^2 V}{\partial \tau^2}} \right).$$

$$1 - \frac{\lambda \frac{\partial^2 S}{\partial \tau^2}}{\frac{\partial^2 V}{\partial \tau^2}} = 1 - \frac{\lambda C'' + \lambda L''}{C'' + \lambda L''} > 0 \quad \text{and} \quad \frac{\partial^2 S}{\partial \tau \partial T} = \frac{\partial^2 L}{\partial \tau \partial T} = -(1 - F(T : t_0)) l''(\tau - T) < 0 \quad \text{Thus, we have } \frac{\partial G}{\partial \tau} < 0.$$

ii) We show that $\frac{\partial G}{\partial \lambda} > 0$. $\frac{\partial G}{\partial \lambda} = \frac{\partial S}{\partial \tau} \frac{\partial^2 \tau}{\partial T \partial \lambda}$. Thus, it is sufficient that $\frac{\partial^2 \tau}{\partial T \partial \lambda} > 0$. Differentiate both sides

w.r.t λ and noting that $\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}$.

$$\frac{\partial^2 \tau}{\partial T \partial \lambda} = \frac{-\frac{\partial^2 S}{\partial T \partial \tau} \cdot \frac{\partial^2 V}{\partial \tau^2} + \frac{\partial^2 S}{\partial T \partial \tau} \cdot \lambda \cdot \left(\frac{\partial^2 L}{\partial \tau^2} \right)}{\left(\frac{\partial^2 V}{\partial \tau^2} \right)^2} > \frac{-\frac{\partial^2 S}{\partial T \partial \tau} \cdot \frac{\partial^2 V}{\partial \tau^2} + \frac{\partial^2 S}{\partial T \partial \tau} \left(\frac{\partial^2 V}{\partial \tau^2} \right)}{\left(\frac{\partial^2 V}{\partial \tau^2} \right)^2} = 0. \quad \text{Hence, we have } \frac{\partial G}{\partial \lambda} > 0. \quad \text{We}$$

also know that $\frac{\partial \tau}{\partial \lambda} < 0$. Together with i) and ii), it shows that the numerator is positive.

Proposition 3: *Both the patching time and the optimal disclosure window are decreasing in t_0 .*

Proof: We conjectured that when time elapses attackers gain more knowledge about the software and therefore more likely to find the vulnerability earlier. This can be formally stated

$$\text{as } t_0 > \tilde{t}_0 \Rightarrow F(s:t_0) > F(s:\tilde{t}_0).$$

1. We first show that $\frac{\partial \tau}{\partial t_0} < 0$. Following the lines of proposition 2, this is true if $\frac{\partial^2 V}{\partial \tau \partial t_0} > 0$. Note that,

from proposition 1, $\tau \geq T$. Thus, the relevant part of the expected user loss is $L(\tau, T) = L_2(\tau)$.

Differentiating one gets

$$\frac{d^2 V}{d\tau dt_0} = \int_0^T \frac{dl(\tau-s)}{d\tau} \frac{df(s:t_0)}{dt_0} ds - \frac{dF(T:t_0)}{dt_0} \frac{dl(\tau-T)}{d\tau}, \text{ where } f(s:t_0) = \frac{dF(s:t_0)}{ds}. \quad \text{Integrating by parts the}$$

RHS

$$\frac{d^2 V}{d\tau dt_0} = l'(\tau-T) \frac{dF(s:t_0)}{dt_0} - \int_0^T l''(\tau-T) \frac{dF(s:t_0)}{dt_0} ds - \frac{dF(T:t_0)}{dt_0} l'(\tau-T) = - \int_0^T l''(\tau-T) \frac{dF(s:t_0)}{dt_0} ds < 0.$$

Thus $\frac{\partial \tau}{\partial t_0} < 0$.

2. Next we show that T is decreasing in t_0 . For this, a necessary and sufficient condition is that

$$\frac{\partial^2 S}{\partial T \partial t_0} > 0. \quad \text{As in the proof to proposition 2, we use } G(T) = 0 \text{ and differentiate w.r.t } t_0:$$

$$\frac{\partial G}{\partial \tau} \left(\frac{\partial \tau}{\partial T} \cdot \frac{dT}{dt_0} + \frac{\partial \tau}{\partial t_0} \right) + \frac{\partial G}{\partial T} \frac{dT}{dt_0} + \frac{\partial G}{\partial t_0} = 0 \quad \text{Rearrange and combine terms, we have}$$

$$\left(\frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial T} + \frac{\partial G}{\partial T} \right) \frac{dT}{dt_0} + \frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial t_0} + \frac{\partial G}{\partial t_0} = 0, \quad \text{Or } \frac{d^2 S}{dT^2} \frac{dT}{dt_0} + \frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial t_0} + \frac{\partial G}{\partial t_0} = 0 \quad \text{Thus, } \frac{dT^*}{dt_0} = - \frac{\frac{\partial G}{\partial \tau} \frac{\partial \tau}{\partial t_0} + \frac{\partial G}{\partial t_0}}{\frac{d^2 S}{dT^2}}.$$

From proof of proposition 2, we know that $\frac{\partial G}{\partial \tau} < 0$. We also know that $\frac{\partial \tau}{\partial t_0} < 0$. Hence, we only need to

prove that $\frac{\partial G}{\partial t_0} > 0$. Note that $\frac{\partial G}{\partial t_0} = \frac{\partial^2 S}{\partial \tau \partial t_0} \cdot \frac{\partial \tau}{\partial T} + \frac{\partial^2 S}{\partial T \partial t_0}$.

i) First, we prove that $\frac{\partial^2 S}{\partial T \partial t_0} > 0$.

$$\frac{\partial^2 S}{\partial T \partial t_0} = \frac{\partial^2 L}{\partial T \partial t_0} = \frac{\partial((F(T:t_0)-1)l'(\tau-T))}{\partial t_0} = \frac{\partial F(T:t_0)}{\partial t_0} l'(\tau-T) > 0$$

ii) Next we show that $\frac{\partial^2 S}{\partial \tau \partial t_0} > 0$. The proof follows along very similar lines to that for $\frac{\partial^2 V}{\partial \tau \partial t_0}$.

Combining (1) and (2), and noting that $\frac{\partial \tau}{\partial T} > 0$ we get that $\frac{\partial G}{\partial t_0} > 0$. Since τ is increasing in T , it follows that both the direct and indirect impact of an increase in t_0 , the stage in the life cycle when the vulnerability is discovered, is to delay patch delivery.

Proposition 4 *When users do not patch instantly, the vendor slows patch development and social planner allows more time before disclosure.*

Proof: Let x denote any factor that increases the rate of patching i.e., $\frac{\partial p(z;x)}{\partial x} > 0$. If V is convex, the

sign of $\frac{\partial \tau}{\partial x}$ is the same as the sign of $-\frac{\partial^2 V}{\partial \tau \partial x}$. Note that $\text{sign}(-\frac{\partial^2 V}{\partial \tau \partial x}) = \text{sign}(-\frac{\partial p(t_1-\tau;x)}{\partial x} \tilde{l}'(t_1-\tau)) < 0$,

where t_1 is the life-cycle of the product. Thus, τ is decreasing in x . The proof that T is also decreasing in x follows along similar lines to those in propositions 2 and 3 and is omitted.

Proposition 5: *When patch quality, q , is endogenous an increase in T will increase τ . If $\frac{\partial^2 V}{\partial \tau \partial q} \geq 0$,*

quality will decrease and conversely, q will decrease if $\frac{\partial^2 V}{\partial \tau \partial q} \leq 0$.

Proof: We have to show that $\frac{d\tau^*}{dT} > 0$; and $\frac{dq^*}{dT} \geq 0 \Leftrightarrow \frac{\partial^2 V}{\partial q \partial \tau} \leq 0$. Let $H(\tau, q)$ represent the Hessian

formed from V . If V is convex, $H(\tau, q) > 0$. By Cramer's Rule,

$$\frac{d\tau}{dT} = \frac{\begin{vmatrix} \frac{\partial^2 V}{\partial \tau \partial T} & \frac{\partial^2 V}{\partial \tau \partial q} \\ \frac{\partial^2 V}{\partial q \partial T} & \frac{\partial^2 V}{\partial q^2} \end{vmatrix}}{H(\tau, q)} \quad \text{Note that } \frac{\partial^2 V}{\partial \tau \partial T} = \frac{\partial^2 L}{\partial \tau \partial T} = (F(T)-1)l''(\tau-T) < 0 \quad \text{and} \quad \frac{\partial^2 V}{\partial q \partial T} = 0$$

Hence, $\begin{vmatrix} \frac{\partial^2 V}{\partial \tau \partial T} & \frac{\partial^2 V}{\partial \tau \partial q} \\ \frac{\partial^2 V}{\partial q \partial T} & \frac{\partial^2 V}{\partial q^2} \end{vmatrix} > 0$. Therefore, $\frac{d\tau^*}{dT} > 0$.

Similarly, we have that $\text{sign}\left(\frac{dq^*}{dT}\right) = \text{sign}\left(\frac{\partial^2 V}{\partial \tau \partial T} \frac{\partial^2 V}{\partial q \partial \tau}\right)$ so that $\frac{dq^*}{dT} \geq 0 \Leftrightarrow \frac{\partial^2 V}{\partial q \partial \tau} \leq 0$.