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

April 2004

Abstract

Disclosing vulnerabilities in a timely fashion is a real and ever more important policy question. Late disclosure reduces the time window that customers are exposed to attacks, but decreases vendor's willingness to deliver quick patch. Currently, there is little or no guidance with each organization following it own ad-hoc policy. This paper is to demonstrate how through optimal timing of disclosure policy (time given to vendor to patch the vulnerability), policy makers can influence behavior of vendors and reduce social cost. We formulate a game-theoretic model. We show that vendors always choose to patch later than a socially optimal disclosure time. Social planner can optimally shrink the time window of disclosure to push vendors to deliver patch in a timely manner. We show that, in general, neither instant disclosure nor non-disclosure is optimal. We then extend the model to allow uncertainty in developing patch and show that increasing uncertainty incurs more cost and vendor delivers quicker patch. In response to larger uncertainty, social planner should shrink the time window. We further extend the model so that the proportion of users implementing patches depends on both the time elapsed and the quality of the patch as well. The corresponding optimal policy is more flexible-vendors have more time to develop a higher-quality patch. Our paper provides a decision tool in understanding how disclosure timing may affect vendor's decision and in turn, what should a policy maker do.

1. Introduction

Information security breaches pose a significant and increasing threat to national security and economic wellbeing. According to Symantec Internet Security Threat Report (2003), each company surveyed experienced on average 30 attacks per week. These attacks often exploited software defects or vulnerabilities.[1] 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.

Often, vulnerability discoverers report vulnerabilities to vendors and keep it secret to allow time for vendors to develop patch[3]. The argument was that the vendor would come up with a workaround strategy or a patch and make the vulnerability public, in due course, balancing costs of patching and disclosure with the benefits. However, many discoverers came to believe that frequently disclosure was excessively delayed or inadequate, leading to the creation of full-disclosure mailing lists in late 90's, such as "Bugtraq".[4] The proponents of full disclosure claim that the threat of instant disclosure increases public awareness, puts pressure on the vendors to issue high quality patches quickly, and improves the quality of software over time.[5]

But many believe that disclosure of vulnerabilities, especially without a good patch is dangerous, for it leaves users defenseless against attackers. At the 2002 Black Hat Conference of Information Technology, Richard Clarke[6], President Bush's former special advisor for cyber

---

[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. 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) has received over 4000 reports of vulnerabilities in the year 2002 alone and has reported more than 82,000 incidents involving various cyber attacks.

[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] Please refer to Jeremy Rauch's article: 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.

[5] In a recent paper Arora et al (2004) find that such instant disclosures do push vendors into responding earlier.

[6] Refer to: http://www.blackhat.com/html/bh-usa-02/bh-usa-02-speakers.html#Richard Clarke. For more details on this debate see (Farrow 2000; Rauch 1999; Preston and Lofton 2002)

space security, criticizing full disclosure said: "It is irresponsible and sometimes extremely damaging to release information before the patch is out."[7]

Institution like the CERT/CC are also important players in the vulnerability disclosure process, because often the discoverer of a vulnerability will inform CERT. CERT then contacts the vendor and provides them with certain a time window to patch the vulnerability (provide a solution so that the vulnerability could not be exploited). After that time window elapses, the vulnerability (along with a patch, if available) is publicly disclosed.

Currently, there are no guidelines or rules for disclosing vulnerabilities, with some vulnerabilities being disclosed very soon after being discovered.[8] While the appropriate dissemination of vulnerability is valuable because it enables users to protect themselves and improves subsequent versions of software, there is considerable debate about when and how the vulnerabilities should be disclosed. As the citations indicate, the public policy problem is real and likely to become ever more important over time. However, there is little extant research that can inform the development of public policy on vulnerability disclosure.

The major goal of this paper is to develop a theoretical framework to design an optimal policy for vulnerability disclosure, which also enables an analysis of the factors that condition how much time should be given to a vendor to develop a patch before vulnerability is publicly disclosed. For this, we develop a theoretical model of the vendor's decision of when to patch, when it is uncertain about how quickly the vulnerability will be exploited by attackers. (We assume that the vendor will only disclose a vulnerability publicly when it releases the patch.) We formulate vendor's decision of when to patch as a one-time, unalterable decision on when to patch upon the discovery of the vulnerability.[9]

---

[7] See also the debate between Robert Graham and Bruce Schneier http://www.robertgraham.com/diary/disclosure.html

[8] For example, CERT follows a 45 days disclosure policy. It appears that CERT almost never discloses a vulnerability without a patch.

[9] Further extension of the model may consider allowing vendor to make real-time decision (from time to time) on when to patch so that vendor may choose to slow or quicken the patch upon the change in the environment. For example, if attackers unexpectedly find the vulnerability very early and commit attacks, the vendor may want to quicken the patch. This paper is therefore best understood as an analysis of vendor policies on patching, and how they are affected the vulnerability disclosure policies in place.

One major contribution of this research is to demonstrate that how an entity such as CERT, acting on behalf of society at large, can use disclosure policy as leverage to modify the incentives vendors face. Importantly, we show that a commitment to early disclosure policies by a "social planner" is indeed an effective way of prompting vendors for a quicker patch, although it is not always beneficial. Using the same theoretical building blocks, we then extend our model to the case when patching time is stochastic and show that vendors chooses to patch more quickly on average, and the "social planner" chooses earlier disclosure policy (smaller windows). In an important extension, we allow patching to take time (i.e., only a fraction of users install patches) and find that this implies a delay in the vendor's and the social planner's optimal patching times. 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 time '*T*' in section 4. In section 5, we extend the basic model to allow for uncertainty in patching time. In section 6, we extend the model to 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 results are presented in section 7.

2. Prior Literature

There is a rich literature on the technical aspects of software vulnerability research, but our focus here is on the literature that directly link to our model. Krsul, Spafford and Tripunitara (1998) classify common vulnerabilities in four major categories. They discuss the characteristics of vulnerability, violations by its exploitation and approaches to prevent these violations. 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. They discuss factors in implementing

such a framework. Arbaugh et al (2000) propose a life cycle model for vulnerability analysis and show how frequently vulnerability is exploited since the time it is made public.

Only a few papers have analyzed economic issues related to problems in the information security. One of the few papers to discuss markets for vulnerabilities is Camp & Wolfram (2000). They describe a means for creating a market for vulnerabilities to increase the security of systems. They contend that government intervention by issuing a new currency in the form of credits for security vulnerabilities will provide incentives to make systems more secure. Kannan, Telang and Xu (2003) present a paper on the market for software vulnerability and show that generally market based mechanism reduces user welfare.

Gordon *et al.* (2002) discuss how the economic issues in Information Sharing & Analysis Centers (ISACs) created under the Presidential Decision Directive 63 are similar to information sharing issues in trade associations, including the problem of free riding. Other papers have analyzed security investments that software users undertake to protect themselves against potential exploits. Gordon & Loeb (2002) develop an economic model for optimal information security investment decisions. Schechter & Smith (2003) discusses how to security investments have to take into account the intruder's cost of breaking-in.

Arora *et al.* (2003a) develop an economic model to study a software vendor's decision of when to introduce its product and how much to invest in patching bugs and vulnerabilities after introduction. Interestingly, they find that a profit-maximizing vendor delivers a product with fewer vulnerabilities than is socially optimal, once one takes into account the social cost of delays in bringing the product to market. However, the profit-maximizing vendor is less willing to patch than is socially efficient. Varian (2000) points out that a key policy aspect of managing information security is to align legal liability to best suitable party. In our model, the vendor internalizes a part of the customer's losses, which allows for imperfect liability.

3. Model

There are four major participants in our model – a "social planner", vendor, customer and

attacker.[10] The social planner chooses a disclosure policy (i.e., the latest a vulnerability must be disclosed) to minimize total social cost. Vendor responds to change in disclosure policy by allocating capital in patching vulnerability to minimize his cost. Customers incur loss when the vulnerability in their system is exploited by attackers.

We model a situation where a vulnerability is discovered by a benign discoverer (other than 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 vendor to make a one-time, committed decision on when to patch upon the discovery of the vulnerability. One argument is that once the vendor has allocated the resources to develop patch, it is hard to make real-time adjustment. Further extension of the model may consider allowing vendor to make real-time decision (from time to time) on when to patch so that vendor may choose to slow or quicken the patch in response to changes in the environment. This extension will significantly complicate the structure of the model. However, 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 4 and 5.

We treat the disclosure policy as binary. Either full 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 $T = \infty$.

[10] In economics, a "social planner" is a convenient way of thinking about the 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. Formally, this is as if the official disclosure time were zero.

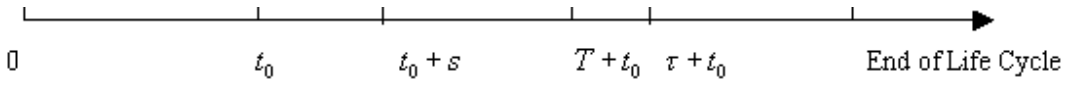

Figure 1. Software Life Cycle

In figure 1, at time '0' the product is released and used by users.[12] A benign user discovers the vulnerability at calendar time t_0. Disclosure policy *T* requires 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. Note that τ, *T* and *s* are simply the time windows of patch-developing, disclosure by social planner and discovery by attacker respectively, measured from the calendar time t_0, which is the time when the vulnerability is first known.

We assume that attackers can exploit an unpatched 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. According to a recent report (Symantec, 2003), approximately 60% of the documented vulnerabilities can be exploited almost instantly either because exploit codes are widely available for free downloading or because no exploit tool is needed. Modifying our model to allow for some period of exploit tool development is straight-forward and yields little insight. Accordingly, we assume that an unpatched vulnerability is exploited instantly upon disclosure.

A key assumption here, that can be relaxed in further extensions, 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 to avoid attacks or mitigate their impact. Allowing for this possibility will likely reduce the impact of disclosure policy on vendor patching 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

[12] We do not consider the diffusion of the product. We assume that all users start using the product at time '0'.

the case where not all customers install patch right away upon release of the patch.

3.1 Vendor's Cost Function

Given a disclosure policy *T*, the software vendor makes decision on allocating its resources in making the patch available. The vendor's objective function (modeled here as a cost function to be minimized) has two terms. The first term is the cost of developing the patch. Recall that τ is the time window of patch developing. In this model, it is used as a proxy of vendor's resource allocation. $C(\tau)$ denotes 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, since marginal utility of freed resources should be decreasing, as commonly assumed. Hence, with respect to τ, marginal cost should also be increasing. Therefore, we also assume $\frac{\partial^2 C(\tau)}{\partial \tau^2} > 0$.

The second cost is a proportion of customer loss that vendor internalizes (via a loss in reputation, loss of future sales). We represent this proportion by λ and call it internalization factor. Currently vendors do not face any legal liability from losses arising due to vulnerabilities in their products but this may change in the future. The expected customer loss is $\theta(\tau, T : X)$, a function of the disclosure policy *T* and the time window for patching, τ. It obviously also depends on customer specific or vulnerability specific factors, which we ignore for simplicity.[13]

Hence, vendor' cost is:

$$V = C(\tau) + \lambda\theta(\tau, T : X) \qquad (1)$$

where λ is the internalization factor.

3.2 Customer Loss Function

At this point, we need to be more specific about $\theta(\tau, T : X)$. We first illustrate under what conditions attacker may exploit customers. Customers suffer loss when either C1 or C2 is true.

[13] 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.

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

C2: Vulnerability is disclosed without a patch by social planner.

We first define $D(t)$ as the cumulative customer loss if they are exposed for a duration t.[14] Intuitively, $D(t;)$ should increase in exposure time t, because the longer the exposure, the greater the chances that an attacker will also develop an exploit, and also because the longer the exposure, the larger the number of malevolent attackers who learn about the vulnerability and get access to the exploit. We also assume that D is strictly convex in t, meaning that the longer the exposure time, the higher the incremental damage from every additional time unit of exposure. As Arbaugh et al (2000) note "Intrusions increase once the community discovers a vulnerability with the rate of intrusions accelerating as news of the vulnerability spreads to a wider audience." The reason of the increasing rate of attacks (at least at the early stage) is that time allows for the spread of vulnerability information to more attackers, the marginal number of attacks increases in sync with the increase in the number of attackers.

Now we can characterize the specific structure of $\theta(\tau, T)$. It is clear that θ will critically depend on when the patch is made available (τ) and when the vulnerability is disclosed (T). Consider the following two cases:

C3: Patch is released before *T;*

C4: Patch is released after *T.*

When patch is released before disclosure time (C3), customers suffer loss only if attackers finds the vulnerability on its own and 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, customer loss is $D(\tau - s)$. On the other hand, if the patch is released after T (i.e. case C4), there are two considerations: first, attacker can find the vulnerability on its own (C1), and have $\tau - s$ [15] of time to exploit. Alternatively, at time

[14] We assume that $D(t)$ is only a function of duration and does not depend on point in software lifecycle the exploitation occurs.

[15] Note that here we omit customer loss after patch is available. In reality, patched vulnerability still causes damage

T, attacker learns about the vulnerability when it is disclosed, and has $\tau - T$ time to exploit it, because the patch is made available only at τ.

To capture the uncertainty about when a vulnerability will also be discovered by an attacker, we assume that the time that attacker finds the vulnerability (*s*) is stochastic, with a distribution *F*(*s*). Therefore, the probability that attacker does not find it within period *T* is simply $1 - F(T : t_0)$, where t_0 is the calendar time when the vulnerability was first discovered. Note that $F(s : t_0)$ is conditional on the vulnerability not being discovered by the attacker before t_0 [16]. We assume that $F(s : t_0)$ increases with t_0 because as attackers accumulate experience and knowledge about the software, they are more likely to find the vulnerability.

Thus, the expected customer loss can be written as follows:

$$\theta(\tau, T; X) = \begin{cases} \int_0^{\tau} D(\tau - s) dF(s : t_0), \text{ when } \tau \le T \\ \int_0^{T} D(\tau - s) dF(s : t_0) + (1 - F(T : t_0)) D(\tau - T), \text{ when } \tau > T \end{cases} \quad (2)$$

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

If *D* is convex, θ is convex in τ (see proof in appendix 2). Moreover, since both *C* and θ are convex in τ, vendor's cost *V* (equation 1) is also convex in τ. Therefore, for 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-developing cost and loss to customers:

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

---

to customers due to not patching. We will study this issue in later section. Furthermore, since the introduction of self-patching or self-updating software, software may automatically patch itself.

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

As explained before, *C* is cost of patching to the vendor and θ is the loss to the customers. Clearly, that vendor's cost function *V,* converges to S when $\lambda = 1$ because then the 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

For $\lambda \in (0,1)$, vendor's incentives and social planner's incentives are not aligned. However, the social planner cannot choose τ, but instead can only choose a disclosure policy *T** and indirectly affect the vendor's choice of τ. Clearly, the sequence of the decision-making is critical. This game can be played in three different ways:

1) Social planner and vendor choose their optimal strategies simultaneously;
2) Vendor decides first and social planner follows;
3) Social planner makes decision first and vendor follows;

It is easy to see that the first two games lead to rather trivial outcomes (See Appendix 1). Moreover the social planner can announce (and CERT does have a de facto policy) and commit to a disclosure policy *T*. Therefore, we focus on the third structure where policy maker announces a time *T* and vendor reacts to it optimally

Recall that from equation (3) first order condition (FOC) for social planner's optimal disclosure policy T^* is

$$\frac{\partial C}{\partial \tau}\frac{\partial \tau}{\partial T} + \frac{\partial \theta}{\partial \tau}\frac{\partial \tau}{\partial T} + \frac{\partial \theta}{\partial T} = 0 \qquad (4)$$

Theorem 1 shows that there exists an optimal solution *T** for social planner. Also, we show that, in corollary 1, instant disclosure and secrecy policy are never optimal disclosure policy. Proofs of all theorems and propositions 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.*

4 Insights and Policy Implications

Also note that, as we expect, *T** depends on vendor's reaction to *T*. In other words, *T** is

dependent on $\frac{\partial \tau}{\partial T}$. Hence, in the following section, we first outline the vendor's reaction function to disclosure policy *T*. Now the setup of model is complete and we are positioned to draw implications from the model.

4.1 How Vendor Reacts to Disclosure Policy *T*

Vendor chooses to minimize its total cost given disclosure time *T*. We have shown that vendor's cost is convex in patching time (τ), hence there exists a solution for vendor's cost-minimization problem. The first order optimization condition, which implicitly defines the optimal patching time τ* as a function of *T* and other variables is:

$$\frac{\partial C}{\partial \tau} + \lambda \frac{\partial \theta}{\partial \tau} = 0 \qquad (5)$$

Let τ_I and τ_S correspond to the optimal patching time given instant disclosure (T = 0) and infinite secrecy policy (i.e., T = ∞), respectively. The optimal patching time τ^* is bounded in a range $[\tau_I, \tau_S]$ (See appendix 2 for the proof.). We first show that, as many full disclosure proponents believe, reducing *T* is indeed effective in pushing vendors to patch more quickly, but only if T < τ_S as proposition 1 formalizes.

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

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 the threshold point τ_S.

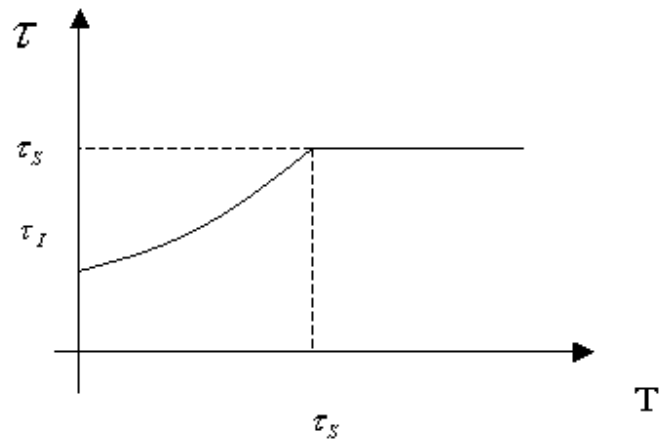

Figure 2 Vendor's Patching Time as Function of *T*

4.2 Characterizing optimal disclosure policy

4.2.1 Impact of λ. It is straightforward to see that increases in λ will cause a vendor to patch earlier because he internalizes a larger fraction of the customer's losses. 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 also suggests that patching time becomes more responsive to the disclosure policy. In turn, that points to proposition 2, which shows that when the vendor internalizes a larger fraction of the loss to customers (larger λ) the optimal disclosure window is smaller (smaller *T*). Note first 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*. This proposition also implies that instituting some type of liability, which in our model implies an increase in λ, would imply earlier patches by the vendor, as well as more aggressive disclosure policies.

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

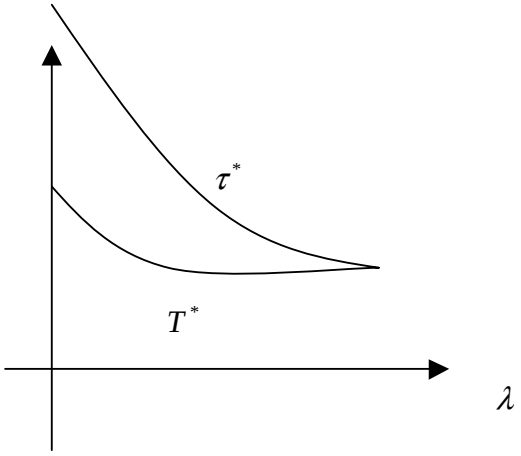

Figure 3*: Optimal Disclosure Policy and Optimal Patching Time as Functions of* λ

4.2.2 Impact of t_0. Proposition 3 indicates that social planner should give vendors more time for developing patch early in the lifecycle of the product. The intuition is as follows: early in the software product lifecycle, the threat of attackers finding the vulnerability is smaller, all else held constant. If the vulnerability is discovered early, the social planner can optimally allow the vendor more time to patch, which also implies lower social cost.

Proposition 3: *The earlier a vulnerability is discovered in the product life cycle (smaller* t_0*) the greater are the socially optimal disclosure time (T) and the patching time (*τ*).*

5. Stochastic Patching Time

The basic model assumes that vendor determines 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. Extending the basic model to allow for stochastic patching time leaves our results unchanged, as formally shown in appendix 3. In addition, under some additional assumptions, we find that increases in uncertainty cause the vendor to patch earlier but also the social planner to reduce *T*.

Let τ and σ denote the mean patching time and variance of patching time, respectively. The actual patching time is stochastic, denoted by ω, such that $E(\omega) = \tau$. We allow vendor to determine on the mean patching time (τ), which is the outcome of resource allocated by vendor. The more resource is allocated for patching, the earlier on average the patch is delivered. We assume that vendor knows the distribution of actual patching time $\omega : \Phi(\omega : \tau, \sigma)$. In other words, variance (σ) is an exogenous variable, as long as the mean (τ) is chosen, the distribution of actual patching time (ω) is predetermined and known to vendor. Hence, the vendor chooses the mean patching time (τ) to minimize the following cost function:

$$V = C(\tau) + \lambda \int_0^e \theta(\omega, T) d\Phi(\omega : \tau) \qquad (6)$$

where $e + t_0$ is the calendar time of the end of software lifecycle. As before, social cost differs from vendor cost only in how much vendor internalizes the loss to customers:

$$S = C(\tau) + \int_0^e \theta(\omega, T) d\Phi(\omega : \tau) \qquad (7)$$

How does the introduction of uncertainty per se affect the disclosure policy and patching? To accommodate uncertainty, we use the concept of stochastic dominance. First-order stochastic dominance says that when one random variable first-order stochastic dominates the second, it is more likely larger than the other. It is also sufficient for the mean of the first variable to be larger than that of the second variable (Rothschild and Stiglitz, 1970). Second order stochastic dominance captures risk. among two choice alternatives, if the first is second-order stochastic dominated by the second, the first choice is more risky. This also implies a smaller variance for the second distribution. For notational simplicity, we assume that a higher mean is equivalent to first order stochastic dominance and a smaller variance is equivalent to second order stochastic dominance i.e.,[17]

---

[17] Note that $\omega_1 \prec_{F.S.D} \omega_2$ is a sufficient condition for $\tau_1 < \tau_2$ while the opposite is not true. Similarly, second order stochastic dominance implies smaller variances but the opposite is not true. Our assumptions would be satisfied for any distribution characterized completely by the mean and the variance, such as the Normal distribution.

F.O.S.D (First Order Stochastic Dominance): If $\tau_1 < \tau_2$, then $\omega_1 \prec_{F.S.D} \omega_2$, where $E(\omega_i) = \tau_i$, for $i = 1,2$

S.O.S.D: If $\sigma_1 < \sigma_2$, and $\tau_1 = \tau_2$ we have $\omega_1 \succ_{S.S.D} \omega_2$.

Under these assumptions, we can show that if $\frac{\partial V}{\partial \tau}$ is convex in τ [18] then the vendor chooses to patch more quickly if it perceives greater uncertainty (captured here as greater variance) in patching time. The intuition is that larger uncertainty increases expected customer cost (and hence also the part that the vendor internalizes) and therefore vendor is willing to invest more in reducing the average patching time, for any given *T*. However, larger variation in patching time also incurs more loss to social planner, and hence, the social planner will also reduce disclosure time, implying a further reduction in τ.

Proposition 4: *With higher uncertainty, vendor reduces their mean time to patch and also, the social planner reduces disclosure time. Therefore,* $\frac{d\tau^*}{d\sigma} < 0$ *and* $\frac{dT^*}{d\sigma} < 0$

6. Patch Quality & Diffusion of Patching: Implications for Disclosure Policy

Until now we assumed that all customers would patch immediately after the patch is available. The 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.

However, many vulnerabilities require customers to download and apply the patch. Not all customers apply patches immediately after it is available. It is reported that six months after the DOS 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 at least three reasons why not all users patch the minute the patch is released. First, it takes time to disseminate the patching information to all users. Second, some customers lack

[18]Without further assumption about the functional form of vendor cost, these signs are undetermined. Note that many functional forms (such as polunomial, expoential function, and so on) satisfy this assumption.

the requisite computer skills. This is sometimes also used as an evidence of poor quality of patch. Consider that the most recent service pack of Windows 2000 Server, which is as large as 27.4 MB and takes a customer an estimated 70 minutes to download through dial-up connection. Large size may well be the reason that many Windows home users do not apply patch. Third, some users are aware of the patch, but would wait to be sure that the patch is more likely to prevent damage than it may cause. An example of a poor quality patch is the Microsoft patch for 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.

Obviously, how quickly customers apply patches is critically dependent on two factors: the time elapsed since the patch is released[19] (denoted by x) and the quality of the patch (denoted by q). We first consider that vendor only determine when to deliver patch (τ). Later we extend to allow the quality of patch to affect the diffusion of patching.

Recall that we used $D(t)$ to denote the cumulative customer loss if they are exposed for a duration t. Before the release of patch (τ), no customer is protected, therefore all the loss materializes. After the release, at any time a proportion of customers are protected through application of patches. Let $p(x)$ denote the cumulative proportion of customers that applied patch after it is released for time x. We assume that $p(x)$ increases with x. At time x, the marginal loss to customers is $(1-p(x))\frac{dD(x+\tau)}{dx}$. Note that $D(x+\tau)$ measures the cumulative attacks. Hence, the total post-patching loss to customers is[20]:

$$\tilde{\theta}(\tau) = \int_0^{\infty} (1-p(x))dD(x+\tau) \qquad (8)$$

If $\tilde{\lambda}$ is the proportion of the post patch release cost that vendor internalizes, the vendor's

[19] One may argue that the time that customers exposed to attacks determines how quickly customers apply patches. The rationale is that the longer customers have been at risk, the more likely they apply patch quickly. Note that exposure time may be different from (usually longer than) the released time. We have developed a model to allow the patching ratio dependent on the exposure time (cf. http://www.andrew.cmu.edu/~xhao/workingpaper.) The setup is more complicated since the patching ratio at any time depends on disclosure policy (T) and when attackers find the vulnerability (s) but yields similar results.

[20] Note that if we allow D() to differ pre and post patch, we can accommodate costs of implementing patches.

expected cost is

$$V(\tau) = C(\tau) + \lambda.\theta(\tau, T) + \tilde{\lambda}.\tilde{\theta}(\tau) \qquad (9)$$

Extending the basic model to allow for diffusion of patching leaves our results unchanged. Additionally, we found that when vendor internalizes more post-patching cost (increase $\tilde{\lambda}$), vendor would like to slow down the release of patch. The intuition is that we now distinguish post-patching loss and loss prior to patching. Late patch (larger τ) increases the loss prior to patching and reduces the post-patching loss. When vendor internalizes more post-patching loss, it is natural for vendor to slow the patch-developing.

Proposition 5: *With diffusion of patching, vendor slows patch-developing and social planner allows more time before disclosure. (i.e.* $\frac{d\tau^*}{d\tilde{\lambda}} > 0$ and $\frac{dT^*}{d\tilde{\lambda}} > 0$*)*

Various factors, including technologies for "pushing" patches to hosts on a network can lead to quicker diffusion of patching, represented here by an upward shift in p(x).. As expected, an upward shift in p(x) will cause the vendor to quicken the delivery of the patch. The social planner will also reduce the time of disclosure in response, as illustrated in proposition 6. The intuition is that shift in *p(x)* has an same effect as a decrease in the internalization factor (smaller $\tilde{\lambda}$), in that both reduce the post patching costs of the vendor. We provide proof in the appendix.

Proposition 6: *With quicker diffusion of patching vendor delivers patch more quickly..*

Differences in patch quality considered:

Since patch quality is a critical factor in determining how quickly customers will apply patch, we extend the model to allow vendor to determine: patching time τ and quality of patch q. We assume higher patch quality *q* implies higher costs, represented by $C(\tau, q)$. At any time *x*, the

proportion of customers that applied patches ($p(x,q)$) increases in the quality of patch such that customers would like to apply patch more quickly given the patch of better quality.

$$V(\tau,q) = C(\tau,q) + \lambda.\theta(\tau,T) + \tilde{\lambda}.\tilde{\theta}(\tau,q) \qquad (10)$$

We show that the vendor improves patch quality if 1) The vendor internalizes less loss to customers; 2) Social planners allows more time for disclosure; 3) The vulnerability is discovered early in the life cycle, as summarized in proposition 7. Also, the vendor slows the delivery of patch simultaneously.

Proposition 7: *Vendor chooses to improve patch quality if the internalization ratio is smaller or social planner enlarges disclosure time window or the vulnerability is discovered in the early stage of software life cycle.*

7. Conclusions

How and when vulnerabilities should be disclosed is an important question. In this paper, we develop a model for analyzing that focuses on the impact of disclosure policy upon vendor behavior. Both vendor behavior and the optimal policy take place in the shadow of what attackers are likely to do. As well, both are conditioned by a variety of factors, such as the behavior of users when the vulnerability is disclosed, and after a patch is released.

An important objective in this paper is to formulate a general model, without narrow function form assumptions, that can characterize the problem. Second, using as few assumptions as possible, we derive a number of 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, therefore, is to disclose the vulnerability sooner than the vendor would like, in order to push the vendor to release the patch sooner. 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 these results are robust to a number of extensions, including uncertainty in patching time, endogenous variations in the quality of the patch, and imperfect compliance by users to the patch.

Even so, 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 the point is that our model can be tailored to reflect those differences without changes to the basic structure of the model. 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. (February 2003), "Provision of Software Quality in the Presence of Patching Technology," Carnegie Mellon University, working paper

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*.

Du, W. & Mathur, A.P. (1998), "Categorization of Software Errors that led to Security Breaches", 21ST

NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE, CRYSTAL CITY, VA

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/

Krsul, I., Spafford, E. & Tripunitara, M. (1998). "Computer vulnerability analysis", Purdue University.

Lipson, H. (2002), "Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues", *CERT/CC special report*

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", *24 Whittier Law Review, 71-142.*

Reinganum, J. (1982). A Dynamic Game of R&D: Patent Protection and Competitive Behavior. *Econometrica*, **48**, 671–688.

Rothschild, M. and J.E. Stiglitz, 1970, "Increasing Risk I: A Definition," *Journal of Economic Theory, II, 225-243*

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

Varian, H.R. (2000), "Managing Online Security Risks," The New York Times, http://www.nytimes.com/library/financial/columns/060100econ-scene.html

CERT Technical report, "Overview of Attack Trends", http://www.cert.org/archive/pdf/attack_trends. pdf

Symantec Inc., 2003, "Symantec Internet Security Threat Report". http://www.symantec.com

NetworkMagazine.com, 2000, "The Pros and Cons of Posting Vulnerabilities".
http://www.networkmagazine.com/article/NMG20001003S0001

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 = ∞.

If 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, in a simultaneous move game, both players choose at (τ_S, τ_S).

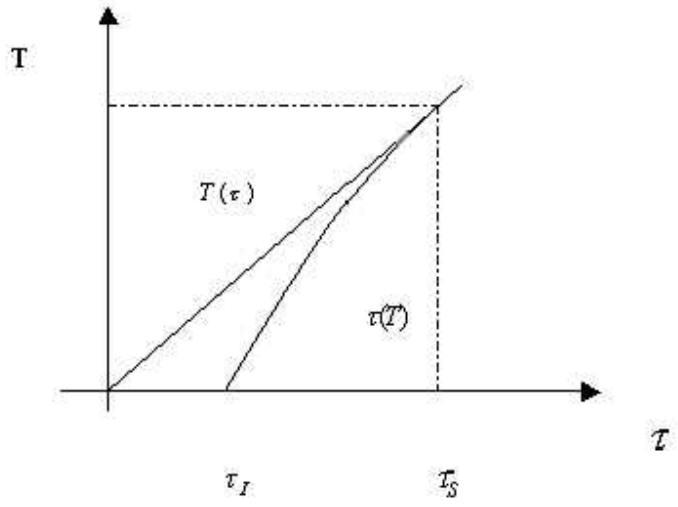

Figure A 1: Social planner and vendor's reaction function

Appendix 2: The Model and Its Extensions

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

Proof: From equation (2),

when $\tau > T$, $\frac{\partial\theta}{\partial\tau} = D(\tau-\tau)\frac{dF(\tau:t_0)}{d\tau} + \int_0^\tau \frac{dD(\tau-s)}{d\tau}dF(s:t_0) = \int_0^\tau \frac{dD(\tau-s)}{d\tau}dF(s:t_0)$

Hence, $\frac{\partial^2\theta}{\partial\tau^2} = \frac{d(D(\tau-\tau))}{d\tau} + \int_0^\tau \frac{d^2D(\tau-s)}{d\tau^2}dF(s:t_0) = D'(0) + \int_0^\tau \frac{d^2D(\tau-s)}{d\tau^2}dF(s:t_0)$

Since *D* is increasing and convex in τ, $\frac{d^2D(\tau-s)}{d\tau^2} \geq 0$ and $D'(0)$, hence we have $\frac{\partial^2\theta}{\partial\tau^2} > 0$. Similarly, one can show that when $\tau \leq T$, $\theta(\tau,T)$ is convex in patching time τ. **QED**

Proof of 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.

$$\frac{dS}{dT} = \frac{\partial C}{\partial\tau}\frac{d\tau}{\partial T} + \frac{\partial\theta}{\partial\tau}\frac{d\tau}{\partial T} + \frac{\partial\theta}{\partial T} \quad (11)$$

Here τ is vendor's optimal decision given *T*. Thus, it must satisfy the following equation

$\frac{\partial C}{\partial\tau} + \lambda\frac{\partial\theta}{\partial\tau} = 0$ (F.O.C)

Putting them together, $\frac{dS}{dT} = (1-\lambda)\frac{\partial\theta}{\partial\tau}\frac{d\tau}{\partial T} + \frac{\partial\theta}{\partial T}$. Also note that, $1 > \frac{d\tau^*}{dT} > 0$ (see proposition 1).

Therefore $\frac{dS}{dT} = (1-\lambda)\frac{\partial\theta}{\partial\tau}\frac{d\tau}{\partial T} + \frac{\partial\theta}{\partial T} < (1-\lambda)\frac{\partial\theta}{\partial\tau} + \frac{\partial\theta}{\partial T}$ (12)

We now show that $\frac{dS}{dT}$ is negative when *T=0* and positive when $T = \infty$, which is sufficient condition that there exists a point that makes $\frac{dS}{dT}$=0. Also at this point, *S* is locally convex in *T*.

1) When *T=0*, $F(T:t_0) = 0$ by definition.

Since $\tau > T$,

$$\frac{\partial\theta}{\partial\tau} = \int_0^T \frac{D(\tau-s)}{d\tau}dF(s:t_0) + (1-F(T:t_0))D'(\tau-T) = D'(\tau)$$

$$\frac{\partial\theta}{\partial T} = (F(T)-1)D'(\tau-T) = -D'(\tau)$$

Putting together, for any $\lambda \neq 1$, we have $\frac{dS}{dT} < (1-\lambda)\frac{\partial \theta}{\partial \tau} + \frac{\partial \theta}{\partial T} = -\lambda D^{'}(\tau) < 0$.

2) When $T = \infty$, $\theta(\tau, T) = \int_0^{\tau} D(\tau - s) dF(s : t_0)$.

Therefore, we have $\frac{\partial \theta}{\partial T} = 0$.

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

The proposition is therefore proved. **QED**

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. **QED**

Proof of Proposition 1: For ease of notation, from now we define

$$\theta_1(\tau) = \int_0^{\tau} D(\tau - s) dF(s : t_0) \text{ and } \theta_2(\tau) = \int_0^{T} D(\tau - s) dF(s : t_0) + (1 - F(T : t_0)) D(\tau - T)$$

$$\text{so that } \theta(\tau, T) = \begin{cases} \theta_1(\tau), when\ \tau \leq T \\ \theta_2(\tau, T), when\ \tau > T \end{cases} \quad (13)$$

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

1) *For* $T \in [0 \quad \tau_s)$, *the vendor always patch 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 $\theta(\tau, T) = \theta_1(\tau)$, the same as that under secrecy policy when $T = \infty$. Hence, $\tau^* = \tau_S$, which contradicts the precondition. Hence, $\tau^* > T$.

2) *For* $T \in [0 \quad \tau_s)$, *Early disclosure T pushes vendor to patch earlier.*

Proof: We want to show that for $T \in [0 \quad \tau_s)$, $\frac{d\tau^*}{dT} > 0$

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:

$$\frac{\partial^2 V}{\partial \tau^2}\frac{d\tau}{dT}+\frac{\partial^2 V}{\partial \tau \partial T}=0 \qquad \text{Thus,} \quad \frac{d\tau^*}{dT}=\frac{-\dfrac{\partial^2 V}{\partial \tau \partial T}}{\dfrac{\partial^2 V}{\partial \tau^2}} \tag{14}$$

Differentiating V w.r.t τ and T and applying integration by parts, we have that $\frac{\partial^2 V}{\partial \tau \partial T}=\frac{\partial^2 \theta}{\partial \tau \partial T}=\lambda\big(F(T)-1\big)D''(\tau-T)<0$.

Thus, we have $\frac{d\tau^*}{dT}>0$

And it is also true that

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

3) *Vendor's optimal patching time is bounded.*
Proof:

Note that 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 $\tau^* \geq \tau_I$.

Also when $T = \tau_S$, $\tau^* = \tau_S$. Thus, it follows that $\tau^* < \tau_S$

To summarize, τ^* is bounded. **QED**

Proof of Proposition 2:

1) First, we prove that $\frac{d\tau^*}{d\lambda}<0$

First of all, τ^* must satisfy $\frac{\partial V}{\partial \tau}=0$. Differentiate both sides with respect to λ :

$$\frac{\partial^2 V}{\partial \tau^2}\frac{d\tau}{d\lambda}+\frac{\partial^2 V}{\partial \tau \partial \lambda}=0.$$

Thus $$\frac{d\tau^*}{d\lambda}=\frac{-\dfrac{\partial^2 V}{\partial \tau \partial \lambda}}{\dfrac{\partial^2 V}{\partial \tau^2}}$$

We only need to show that $\dfrac{\partial^2 V}{\partial \tau \partial \lambda}=\dfrac{\partial^2 \theta}{\partial \tau}\int_0^T D'(\tau-s)dF(s:t_0)+\big(1-F(T:t_0)\big)D'(\tau-T)>0$. Thus,

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

2) We now prove that $\dfrac{dT^*}{d\lambda}<0$

Let $G(T)=\dfrac{\partial S}{\partial \tau}\dfrac{\partial \tau}{\partial T}+\dfrac{\partial S}{\partial T}$.

T^* must satisfy $G(T)=0$ (15)

which is the F.O.C of social planner's optimal decision on T. Differentiate both sides with respect to λ: $\dfrac{\partial G}{\partial \tau}.\left(\dfrac{\partial \tau}{\partial T}.\dfrac{dT}{d\lambda}+\dfrac{\partial \tau}{\partial \lambda}\right)+\dfrac{\partial G}{\partial T}\dfrac{dT}{d\lambda}+\dfrac{\partial G}{\partial \lambda}=0$

Arrange terms and combine them

$$\left(\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial T}+\frac{\partial G}{\partial T}\right)\frac{dT}{d\lambda}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \lambda}+\frac{\partial G}{\partial \lambda}=0 \quad \frac{d^2 S}{dT^2}\frac{dT}{d\lambda}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \lambda}+\frac{\partial G}{\partial \lambda}=0.$$

Thus, $$\frac{dT^*}{d\lambda}=-\frac{\dfrac{\partial G}{\partial \tau}\dfrac{\partial \tau}{\partial \lambda}+\dfrac{\partial G}{\partial \lambda}}{\dfrac{d^2 S}{dT^2}}$$ (16)

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

i) We now show that $\dfrac{\partial G}{\partial \tau}<0$.

Recall that $\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}$.

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

$1 - \frac{\lambda.\frac{\partial^2 S}{\partial \tau^2}}{\frac{\partial^2 V}{\partial \tau^2}} = 1 - \frac{\lambda C'' + \lambda \theta''}{C'' + \lambda \theta''} > 0$ and $\frac{\partial^2 S}{\partial \tau \partial T} = \frac{\partial^2 \theta}{\partial \tau \partial T} = (F(T-1)).D''(\tau - T) < 0$ Thus, we have

$\frac{\partial G}{\partial \tau} < 0$.

ii) We show that $\frac{\partial G}{\partial \lambda} > 0$.

$\frac{\partial G}{\partial \lambda} = \frac{\partial S}{\partial \tau}\frac{\partial^2 \tau}{\partial T \partial \lambda}$. Recall that $\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}$.

Differentiate both sides w.r.t λ

$$\frac{\partial^2 \tau}{\partial T \partial \lambda} = \frac{-\frac{\partial^2 S}{\partial T \partial \tau}.\frac{\partial^2 V}{\partial \tau^2} + \frac{\partial^2 S}{\partial T \partial \tau}.\lambda.\left(\frac{\partial^2 \theta}{\partial \tau^2}\right)}{\left(\frac{\partial^2 V}{\partial \tau^2}\right)^2} > \frac{-\frac{\partial^2 S}{\partial T \partial \tau}.\frac{\partial^2 V}{\partial \tau^2} + \frac{\partial^2 S}{\partial T \partial \tau}.\left(\frac{\partial^2 V}{\partial \tau^2}\right)}{\left(\frac{\partial^2 V}{\partial \tau^2}\right)^2} = 0$$

Hence, we have $\frac{\partial G}{\partial \lambda} > 0$.

We also know that $\frac{\partial \tau}{\partial \lambda} < 0$. Together with i) and ii), we proved that the numerator is positive. The proposition is proved. **QED**

We conjectured that when time elapses attackers gain more knowledge about the software and therefore more likely to find the vulnerability earlier. We formally formulate this assumption as follows:

F.S.D Assumption: If $t_0 > \tilde{t}_0$, all else held constant, we have $s \prec_{F.S.D} \tilde{s}$.

Lemma 1: For any *m*, we have $\frac{\partial F(m:t_0)}{\partial t_0} > 0$

Proof: By the definition of F.S.D , for any *m*, we have $\Pr(s > m) < \Pr(\tilde{s} > m)$.

i.e. $F(m:t_0) > F(m:\tilde{t}_0)$. Hence, it is immediate that $\frac{\partial F(m:t_0)}{\partial t_0} > 0$. **QED**

Proof of Proposition 3: As in the proof to proposition 2, we differentiate both sides of equation (15) w.r.t t_0 :

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

$$\left(\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial T}+\frac{\partial G}{\partial T}\right)\frac{dT}{dt_0}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial t_0}+\frac{\partial G}{\partial t_0}=0$$

$$\frac{d^2S}{dT^2}\frac{dT}{dt_0}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial t_0}+\frac{\partial G}{\partial t_0}=0$$

Thus, $\frac{dT^*}{dt_0}=-\frac{\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial t_0}+\frac{\partial G}{\partial t_0}}{\frac{d^2S}{dT^2}}$ (17)

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

$$\frac{\partial G}{\partial t_0}=\frac{\partial^2 S}{\partial \tau \partial t_0}.\frac{\partial \tau}{\partial T}+\frac{\partial^2 S}{\partial T \partial t_0}$$

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

$$\frac{\partial^2 S}{\partial T \partial t_0} = \frac{\partial^2 \theta}{\partial T \partial t_0} = \frac{\partial\left((F(T:t_0)-1)D'(\tau - T)\right)}{\partial t_0} = \frac{\partial F(T:t_0)}{\partial t_0}.D'(\tau - T)$$

From Lemma 1, we have $\frac{\partial^2 S}{\partial T \partial t_0} > 0$.

2) We show that $\frac{\partial^2 S}{\partial \tau \partial t_0}.\frac{\partial \tau}{\partial T} > 0$.

$$\frac{\partial S}{\partial \tau} = C(\tau) + D(\tau - T) + \int_0^T \left(D(\tau - s) - D(\tau - T)\right) dF(s)$$

$D(\tau - s) - D(\tau - T)$ is monotonically decreasing in *s*.

We assumed that $t_0 > \tilde{t}_0$, $s \prec_{F.S.D} \tilde{s}$

From F.S.D theorem, we know that $\frac{\partial S}{\partial \tau} > \frac{\partial \tilde{S}}{\partial \tau}$.

Thus $\frac{\partial^2 S(t_0)}{\partial \tau \partial t_0} > 0.$, $\frac{\partial^2 S}{\partial \tau \partial t_0}.\frac{\partial \tau}{\partial T} > 0$.

Combining 1) and 2), we proved that $\frac{\partial G}{\partial t_0} > 0$. The proposition is thus proved. **QED**

Proof of Proposition 1 (under uncertainty):

$\frac{\partial V}{\partial \tau} = 0$ is the F.O.C of vendor's optimal decision given T.

Differentiate both sides with respect to T: $\frac{\partial^2 V}{\partial \tau^2}\frac{d\tau}{dT} + \frac{\partial^2 V}{\partial \tau \partial T} = 0$

Thus, we have $\frac{d\tau^*}{dT} = \frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}}$

V is convex in τ, i.e. $\frac{\partial^2 V}{\partial \tau^2} > 0$.

$$V = C(\tau) + \lambda.\left(\int_0^T \int_0^\tau D(\omega - s)dF(s:t_0)d\Phi(\omega:\tau) + \int_T^e \left(\int_0^T D(\omega - s)dF(s:t_0) + (1 - F(T:t_0)D(\omega - T)\right)d\Phi(\omega:\tau)\right)$$

Integrate by parts:

$$\frac{\partial V}{\partial T} = \lambda.\left(\int_T^e (F(T:t_0) - 1)D(\omega - T)d\Phi(w:\tau)\right) < 0$$

Let $K(\omega) = (F(T:t_0) - 1)D(\omega - T)$ and $\frac{\partial V}{\partial T} = \lambda.\left(\int_T^e K(\omega)d\Phi(w:\tau)\right)$

$K(\omega)$ is decreasing in ω. According to F.S.D assumption, for any $\tau_1 < \tau_2$, then $\omega_1 \prec_{F.S.D} \omega_2$.

Hence, according to F.S.D theorem, we have $\frac{\partial V}{\partial T}_{|\tau=\tau_1} > \frac{\partial V}{\partial T}_{|\tau=\tau_2}$ i.e. $\frac{\partial^2 V}{\partial \tau \partial T} < 0$ Thus, $\frac{d\tau^*}{dT} > 0$

QED

Proof of Proposition 2 (under Uncertainty): $\frac{\partial V}{\partial \tau} = 0$ is the vendor's F.O.C given T.

Differentiate both sides with respect to T: $\frac{\partial^2 V}{\partial \tau^2}\frac{d\tau}{d\lambda} + \frac{\partial^2 V}{\partial \tau \partial \lambda} = 0$

$\frac{d\tau^*}{d\lambda} = \frac{-\frac{\partial^2 V}{\partial \tau \partial \lambda}}{\frac{\partial^2 V}{\partial \tau^2}}$. V is convex, i.e. $\frac{\partial^2 V}{\partial \tau^2} > 0$.

$$V = C(\tau) + \lambda.\left(\int_0^T \int_0^\tau D(\omega - s)dF(s:t_0)d\Phi(\omega:\tau) + \int_T^e \left(\int_0^T D(\omega - s)dF(s:t_0) + (1 - F(T:t_0)D(\omega - T)\right)d\Phi(\omega:\tau)\right)$$

$$\frac{\partial V}{\partial \lambda} = \int_0^T \int_0^\tau D(\omega - s)dF(s:t_0)d\Phi(\omega:\tau) + \int_T^e \left(\int_0^T D(\omega - s)dF(s:t_0) + (1 - F(T:t_0)D(\omega - T)\right)d\Phi(\omega:\tau)$$

As in the proof of proposition 1 under uncertainty, $\frac{\partial^2 V}{\partial \tau \partial \lambda} > 0$. Thus, we have $\frac{d\tau}{d\lambda} < 0$ [21] **QED**

Proof of Proposition 4:

$G(T) = 0$ is the F.O.C of social planner's optimal decision on T. Differentiate both sides with respect to σ:

[21] Since proofs for proposition 3 under uncertainty are similar as those of deterministic case, we skip the proofs.

$$\frac{\partial G}{\partial \tau}.\left(\frac{\partial \tau}{\partial T}.\frac{dT}{d\sigma}+\frac{\partial \tau}{\partial \sigma}\right)+\frac{\partial G}{\partial T}\frac{dT}{d\sigma}+\frac{\partial G}{\partial \sigma}=0$$

Rearrange and combine terms

$$\left(\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial T}+\frac{\partial G}{\partial T}\right)\frac{dT}{d\sigma}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \sigma}+\frac{\partial G}{\partial \sigma}=0$$

$$\frac{d^2S}{dT^2}\frac{dT}{d\sigma}+\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \sigma}+\frac{\partial G}{\partial \sigma}=0$$

As in the proof to proposition 2, we have $\frac{\partial G}{\partial \tau}<0$. We also know that $\frac{\partial \tau}{\partial \sigma}<0$. Hence, we only need to prove that $\frac{\partial G}{\partial \sigma}>0$. $\frac{\partial G}{\partial \sigma}=\frac{\partial^2 S}{\partial \tau \partial \sigma}.\frac{\partial \tau}{\partial T}+\frac{\partial^2 S}{\partial T \partial \sigma}$

1) First, according to second-order stochastic dominance theorem, we have $\frac{\partial^2 S}{\partial \tau \partial \sigma}>0$

2) $$\frac{\partial^2 S}{\partial T \partial \sigma}=.\frac{\partial \int_0^e (F(T)-1)D'(\omega-T)d\Phi(\omega:\tau,\sigma)}{\partial \sigma}>0$$

Hence, we have $\frac{\partial G}{\partial \sigma}>0$. which implies $\frac{dT^*}{d\sigma}<0$. **QED**

Proof of Proposition 5:

1) We first prove that $\frac{d\tau^*}{d\tilde{\lambda}}>0$

From equation (8) and (9): $\tilde{\theta}(\tau)=\int_0^\infty (1-p(x))dD(x+\tau)$ and $V(\tau)=C(\tau)+\lambda.\theta(\tau,T)+\tilde{\lambda}.\tilde{\theta}(\tau)$

Hence, we have $\frac{\partial^2 V}{\partial \tau \partial \tilde{\lambda}}=\frac{d\tilde{\theta}}{d\tau}=(p(0)-1)D'(\tau)<0$. Since τ^* satisfies F.O.C: $\frac{\partial V}{\partial \tau}=0$.

Differentiating both sides with respect to $\tilde{\lambda}$, we get $\frac{d\tau^*}{d\tilde{\lambda}} = \frac{-\frac{\partial^2 V}{\partial \tau \partial \tilde{\lambda}}}{\frac{\partial^2 V}{\partial \tau^2}} > 0$

2) We now prove that $\frac{dT^*}{d\tilde{\lambda}} > 0$

$G(\tau, T) = \frac{\partial S}{\partial \tau}\frac{\partial \tau}{\partial T} + \frac{\partial S}{\partial T} = 0$ is the F.O.C of social planner's optimal decision on T.

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

$\Rightarrow \frac{d^2 S}{dT^2}\frac{dT}{d\tilde{\lambda}} + \frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \tilde{\lambda}} + \frac{\partial G}{\partial \tilde{\lambda}} = 0$. $\Rightarrow \frac{dT^*}{d\tilde{\lambda}} = -\frac{\frac{\partial G}{\partial \tau}\frac{\partial \tau}{\partial \tilde{\lambda}} + \frac{\partial G}{\partial \tilde{\lambda}}}{\frac{d^2 S}{dT^2}}$. Here $\frac{d^2 S}{dT^2} > 0$ since social cost *S* is convex in *T*. From the first step, we have $\frac{\partial \tau}{\partial \tilde{\lambda}} > 0$. Therefore, as long as $\frac{\partial G}{\partial \tau} < 0$ *and* $\frac{\partial G}{\partial \tilde{\lambda}} < 0$, we have $\frac{dT^*}{d\tilde{\lambda}} > 0$.

i) $\frac{\partial G}{\partial \tau} = \frac{\partial^2 S}{\partial \tau^2}.\frac{\partial \tau}{\partial T} + \frac{\partial^2 S}{\partial \tau \partial T} = \frac{\partial^2 S}{\partial \tau^2}.\frac{-\frac{\partial^2 V}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}} + \frac{\partial^2 S}{\partial \tau \partial T} = \lambda.\frac{\partial^2 S}{\partial \tau^2}.\frac{-\frac{\partial^2 S}{\partial \tau \partial T}}{\frac{\partial^2 V}{\partial \tau^2}} + \frac{\partial^2 S}{\partial \tau \partial T}$

$\frac{\lambda.\frac{\partial^2 S}{\partial \tau^2}}{\frac{\partial^2 V}{\partial \tau^2}} = \frac{\lambda C_P'' + \lambda \theta''}{C_P'' + \lambda \theta''} < 1$ Or, $-\frac{\partial^2 S}{\partial \tau \partial T} = (1 - F(T)).D''(\tau - T) > 0 \Rightarrow \frac{\partial G}{\partial \tau} < 0$.

ii) $\frac{\partial G}{\partial \tilde{\lambda}} = \frac{\partial S}{\partial \tau}\frac{\partial^2 \tau}{\partial T \partial \tilde{\lambda}}$.

$$\frac{\partial^2 \tau}{\partial T \partial \tilde{\lambda}} = \frac{-\frac{\partial^2 S}{\partial T \partial \tau} \cdot \left(\frac{\partial^2 \tilde{\theta}}{\partial \tau^2} \right)}{\left(\frac{\partial^2 V}{\partial \tau^2} \right)^2} < 0$$

Note that here $\frac{\partial^2 \tilde{\theta}}{\partial \tau^2} = (p(0)-1)D''(\tau) < 0$ **QED**

Proof of Proposition 6:

If for any x, one has $\tilde{p}(x) > p(x)$, then $\tilde{\tau}^* < \tau^*$ (Here $\tilde{\tau}^*$ and τ^* are vendor's optimal decisions corresponding to $\tilde{p}(x)$ and $p(x)$, respectively.)

Proof:

Let V and $\tilde{V}$ vendor cost functions corresponding to $p(x)$ and $\tilde{p}(x)$, respectively. Since V and $\tilde{V}$ are only different in $p(x)$ and $\tilde{p}(x)$, V and $\tilde{V}$ are only different in $\tilde{\theta}$.

$$\frac{\partial \tilde{\theta}}{\partial \tau} = (p(0)-1)D'(\tau).$$

Hence, one has that $\frac{d\tilde{V}}{d\tau} - \frac{dV}{d\tau} = (\tilde{p}(0)-1)D'(\tau) - (p(0)-1)D'(\tau) = (\tilde{p}(0) - p(0))D'(\tau) > 0$, for any τ, i.e. $\frac{d\tilde{V}}{d\tau} > \frac{dV}{d\tau}$. $\left.\frac{dV}{d\tau}\right|_{\tau^*} = 0$, $\left.\frac{d\tilde{V}}{d\tau}\right|_{\tau^*} > 0$. Since $\tilde{V}$ is convex, $\frac{d\tilde{V}}{d\tau}$ is increasing in τ. Thus, for $\frac{d\tilde{V}}{d\tau} = 0$, τ has to decrease. Hence, one has that $\tilde{\tau}^* < \tau^*$. **QED**

Proof of Proposition 7:

We want to show the following:

$$\frac{d\tau^*}{d\lambda} < 0, \frac{d\tau^*}{dT} > 0 \; and \; \frac{d\tau^*}{dt_0} < 0 \quad \& \quad \frac{dq^*}{d\lambda} < 0, \frac{dq^*}{dT} > 0 \; and \; \frac{dq^*}{dt_0} < 0$$

To avoid redundancy due to the similarity in proofs, we only show $\frac{d\tau^*}{dT} > 0$ and $\frac{dq^*}{dT} > 0$.

Proof: We start with vendor's first order optimization condition:

$$\frac{\partial V}{\partial \tau} = 0$$
$$\frac{\partial V}{\partial q} = 0$$

Taking the total derivative of both equations

$$\frac{\partial^2 V}{\partial \tau^2} d\tau + \frac{\partial^2 V}{\partial \tau \partial q} dq = -\frac{\partial^2 V}{\partial \tau \partial T} dT$$
$$\frac{\partial^2 V}{\partial \tau \partial q} d\tau + \frac{\partial^2 V}{\partial q^2} dq = -\frac{\partial^2 V}{\partial q \partial T} dT$$

By Crammer Rule,

$$\frac{d\tau}{dT} = \frac{\begin{vmatrix} -\frac{\partial^2 V}{\partial \tau \partial T} & \frac{\partial^2 V}{\partial \tau \partial q} \\ -\frac{\partial^2 V}{\partial q \partial T} & \frac{\partial^2 V}{\partial q^2} \end{vmatrix}}{H(\tau, q)}$$

By assumption, the determinant of the Hessian matrix $H(\tau, q)$ is positive.

Note that $\frac{\partial^2 V}{\partial \tau \partial T} = \frac{\partial^2 \theta}{\partial \tau \partial T} = (F(T) - 1)D''(\tau - T) < 0$ and $\frac{\partial^2 V}{\partial q \partial T} = 0$

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 $\frac{dq^*}{dT} = \frac{\begin{vmatrix} \frac{\partial^2 V}{\partial \tau^2} & -\frac{\partial^2 V}{\partial \tau \partial T} \\ \frac{\partial^2 V}{\partial q \partial \tau} & -\frac{\partial^2 V}{\partial q \partial T} \end{vmatrix}}{H(\tau, q)} > 0$ **QED**