December 2007

A Reputation-Based Mechanism for Software Vulnerability Disclosure

Xia Zhao
University of Texas, Austin

A Reputation-Based Mechanism for Software Vulnerability Disclosure

Xia Zhao **Jianqing Chen** **Andrew B. Whinston**

Department of Information, Risk, and Operations Management

The University of Texas at Austin, Austin, TX, 78712

xia.zhao@phd.mccombs.utexas.edu; chenjq@mail.utexas.edu; abw@uts.cc.utexas.edu

Abstract

Whether and how to disclose software vulnerability information has been debated intensely. An optimal disclosure policy should balance the tradeoff between its impact on software vendors' incentives and the potential risks imposed on customers. Previous research on software vulnerability primarily focused on the timing aspect of the disclosure policy. In this paper, we investigate another dimension -- the reputation aspect -- of the disclosure policy. We propose a disclosure mechanism integrated with a reputation system which reflects software security level. Reputation by itself can effectively provide an incentive for software vendors to fix vulnerabilities. Furthermore, the reputation operator can partially release vulnerability details to optimize social welfare.

Keywords: Software vulnerability disclosure, patch, reputation systems, information security

Introduction

As computers and the Internet exert its influence in all realms of human endeavor, the rising security concerns have been eroding its value. Software vulnerabilities or flaws are the primary reason for security breaches. Currently software vulnerability information is disclosed through many sources, such as CERT/CC (Computer Security Incident Response Team/Coordination Center),[1] Bugtraq mailing list,[2] Secunia,[3] and iDefence.[4] There is a contentious debate on whether and

[1] CERT/CC is a research center associated to Carnegie Mellow University and the Department of Homeland Security (DHS).

how to disclose software vulnerabilities. Proponents of vulnerability disclosure claim that vulnerability disclosure can help users beware of potential security risks and take precautions, as well as provide software vendors incentives to develop and release patches in a timely manner. Opponents argue that vulnerability disclosure, especially without a patch, will expose customers to security risks as crackers (black-hat hackers) can easily identify and exploit the vulnerabilities. This paper studies the optimal vulnerability disclosure policy from an economic perspective.

Current vulnerability disclosure sources follow different disclosure policies. CERT/CC and iDefense notify software vendors before they make the vulnerability information public. In contrast, Bugtrap and Secunia mailing lists instantaneously disclose vulnerability details, even links to exploit codes. The common characteristic of the above mechanisms is that vulnerability information is publicly disclosed, sooner or later.

Vulnerability disclosure effectively expedites vendors' response. It, however, may cause dramatic damage to consumers. Crackers may devise attacks targeting at public vulnerabilities, especially unpatched ones. Even disclosing patched vulnerabilities may cause significant economic loss since customers may not apply the patch in a timely manner.

How to disclose vulnerability information is a challenging issue. The disclosure source should balance the tradeoff between incentive effect and the potential risks associated with public vulnerabilities. Previous research on vulnerability disclosure primarily focused on one dimension -- the timing aspect -- of the disclosure policy. In this paper, we consider another dimension -- the reputation aspect -- of vulnerability disclosure.

We propose a disclosure mechanism integrated with a reputation system for software security quality. The reputation system is operated by an infomediary, who is an independent third-party institution. Reputation scores are publicly observable and calculated according to the numbers of identified vulnerabilities and available patches. In each period, a vulnerability may be identified and reported to the infomediary by users. Once a vulnerability is reported, the infomediary informs the vendor. The vendor will determine whether to fix the vulnerability at a cost. At the end of each period, the infomediary updates the software's reputation score according to the vendor's response. In particular, when a vulnerability is identified and the vendor fails to releases a patch, the infomediary decreases the reputation score. When no vulnerability is identified and a patch is released for any previous vulnerability, the infomediary increases the reputation score. Otherwise, the reputation score stays constant. In addition, the infomediary may partially release vulnerability information.

Customers can infer the security level of the software from the software's reputation. Therefore, the reputation score affects customers' willingness to pay, which in turn influences the vendor's incentive to develop patches. In addition, the level

---

http://www.cert.org

[2] BugTraq is a full disclosure mailing list among security professionals. It is supported by SecurityFocus, a vendor-neutral site that provides security information to everyone at no charge. www.securityfocus.com/about

[3] Secunia is a privately held, financially profitable company which provides security solutions for businesses and governmental institutions. http://secunia.com

[4] iDefense Security & Vulnerability Research Labs is for-profit security company which provides security advisory to its subscribers. http://labs.idefense.com

of vulnerability details disclosed will moderate the impact of the reputation. When a higher degree of vulnerability details is disclosed, customers will suffer more heavily from unpatched vulnerabilities.

By establishing a reputation system and controlling the disclosure level, our mechanism has the potential to improve software security from two aspects. First, the reputation effect can effectively provide software vendors incentives to fix vulnerabilities. Second, controlling the degree of vulnerability details disclosed potentially optimizes the security risks faced by customers.

The paper is organized as follows. In section 2, we review the literature on software vulnerability disclosure and reputation systems. In section 3, we formulate a model and derive some preliminary results. Section 4 describes the research plan. Finally we conclude the paper.

Literature Review

There is a stream of literature exploring various issues of software vulnerability disclosure. Arora et al. (2005) empirically show that the instant disclosure policy leads to earlier patch delivery. In another paper, Arora, et al (2005) use a game-theoretic model to illustrate the incentive misalignment between software vendors and a social planner. They show that vendors always choose to patch less expeditiously than socially optimality and the social planner can shrink the protected period so as to push vendors to deliver patches in a timely manner. Kannan and Telang (2005) theoretically compare different vulnerability disclosure mechanisms, a CERT-type mechanism and a market-based mechanism (e.g., iDefense), and demonstrate that the latter, if not regulated, always underperforms the former. Our paper studies to what degree an infomediary should disclose vulnerability information, considering the vendor's incentive and customers' loss associated with the disclosure.

Our work is also related to the literature on reputation systems. Reputation mechanisms have been extensively studied to deal with moral hazard and/or adverse selection issues (Friedman and Resnick 2001, Fudenberg and Levine 1992, Mailath and Samuelson 2001). Recently online reputation systems have been widely used in the electronic trading environment, such as online auction (e.g., eBay and eLance), online shopping search engines (e.g., BizRate and Pricegrabber), and online forum (e.g., Slashdot). Dellarocas (2005) discusses the issues regarding the design, evaluation and use of the Internet-based reputation mechanism from an economic perspective. Different from the consumer-feedback based reputation, Xu et al. (2007) introduce an audited reputation systems, where an auditor checks the quality of products and updates firms' reputations accordingly. Ekmekci (2005) designs a rating system to address the moral hazard issues in an adverse selection setting. The rating system can sustain the reputation effect and alleviate the moral hazard problem by censoring the feedback information (e.g., only releasing summary statistics, only showing the most recent data, refining the performance data into a binary form). Our paper develops a reputation system to address the moral hazard issue of a long-

lived software vendor. Differing from the aforementioned reputation literature, our paper concerns a product with cumulative quality.

The Model

We build up a game-theoretic model to examine the impact of the proposed mechanism on a vendor's behavior. We consider a setup with a long-lived software vendor who sells a software product to short-lived consumers in an infinite horizon.

At the beginning of period t, $t = 1, 2, ..., \infty$, a vulnerability associated with the software may be discovered with probability s, $0 \le s \le 1$. We use b to indicate whether a vulnerability is discovered, with $b = 1$ representing a vulnerability discovered and $b = 0$ representing no vulnerability identified. The identified vulnerability will be reported to an infomediary. The infomediary, which aims at maximizing social welfare, informs the vendor and gives it one period to develop a patch. The vendor chooses whether or not to develop a patch at a cost c. If the vender invests in working on a patch, it can successfully develop a patch for the vulnerability with probability γ. We use e_t to indicate whether the investment is made, $e_t \in \{0, 1\}$, with $e_t = 1$ representing investing. Denote p_t as the price of the software charged by the vendor at period t. The software vendor maximizes its long-term profit by the choice of p_t and e_t.

The infomediary employs a reputation score to evaluate the security level of the software. We use $r \in R$ to represent the score, where $R = \{-\infty, ..., -1, 0\}$. At the end of period t, if a newly identified vulnerability is unpatched, the infomediary will decrease the reputation score by 1. If no vulnerability is discovered but a patch is released for a previous vulnerability, the infomediary adds 1 to the reputation score. Otherwise, the reputation score stays constant. Besides updating the reputation score in each period, the infomediary can publish the unpatched vulnerability at different levels of details. Let $y_t \in \{0, 1\}$ represent the degree of vulnerability details disclosed. y_t is the infomediary's decision variable.

The timing of the events in each period is represented in Figure 1 .

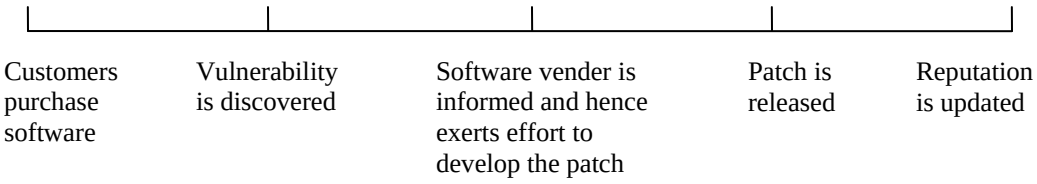

Figure 1: Timing of the events in each period

At the beginning of each period, a continuum of consumers enters the software product market. The customer's value for the software is v. Assume customers are heterogeneous in their value of the software and v follows a distribution

$F(v)$. The unpatched vulnerability is subject to crackers' attack. A customer losses *z* from an attack.

Even though the infomediary does not release any detail about previous vulnerabilities, crackers can still discover and exploit the undisclosed vulnerabilities with probability *h*. It is plausible that the more information disclosed by the infomediary, the more likely crackers exploit the vulnerability. We model this information effect by assuming that the likelihood of getting attacked is increased to $h+y$, provided that *y* degree of vulnerability detail is disclosed.

Let $u_i = u(r=i)$ denote a consumer' willingness to pay for a product with reputation *i*.

$$u_i = v + z(h+y)i$$

At period *t*, consumers purchase the software if and only if $u_i \geq p_t$.

For simplicity, we assume the vendor has a steady potential demand across periods. Therefore, the optimal price the vendor charges only depends on its reputation, and is independent of *t*. Similarly, the patching decision is also independent of *t*. We thus omit the index *t*, and instead use reputation *i* as a state variable.

Preliminary Analysis

Denote $V_i^j \equiv V(r=i,\ b=j)$, the value function, and $\pi_i \equiv \pi(r=i)$, the current revenue. For a period with reputation score i, if a vulnerability is discovered,

$$V_i^1 = \max_{e,\ p} \pi_i - ce + \beta\left[\gamma e V_i + (1-\gamma e)V_{i-1}\right],\ \ i \leq 0 \tag{2}$$

$$V_i^0 = \max_{e,\ p} \pi_i - ce + \beta\left[\gamma e V_{i+1} + (1-\gamma e)V_i\right],\ \ i < 0 \tag{3}$$

$$V_0^0 = \max_{e,\ p} \pi_0 - ce + \beta V_i \tag{4}$$

where β is the discount factor.

Denote $V_i \equiv V(r=i)$, the expected value function.

$$V_i = sV_i^1 + (1-s)V_i^0$$

Notice that patching does not affect the current revenue. Therefore, the pricing problem can be separated from the patching decision.

We can determine the marginal type (consumer) when the reputation and the price of the software is i and p respectively using (1) and the equation $u_i = p$.

$$\overline{v}_i = p - i(h+y)z \tag{5}$$

Therefore, the software vendor's revenue from current period is $\pi_i = p(1-\overline{v}_i) = p(1-p+i(h+\alpha)z)$. The optimal

price and the maximum revenue can be derived as follows.

$$p_i^* = \frac{1+i(h+y)z}{2} \tag{6}$$

$$\pi_i^* = \left(\frac{1+i(h+y)z}{2}\right)^2 \tag{7}$$

Based on the vendor's value function and profit function, we can analyze the equilibrium price, the equilibrium patching decision**,** and the design of the optimal reputation system.

Research Plan and Conclusion

The study aims to demonstrate the feasibility of using a reputation mechanism to improve software security. We use a game-theoretic model to illustrate that the reputation mechanism with a simple and proper measure can substitute direct vulnerability disclosure and improve software security at a lower social cost.

We will extend the baseline model by incorporating relevant elements in the next step. First, we will consider the negative externality of security risks. Given a good reputation, there are two countervailing forces influencing a software vendor's profit. On one hand, a good reputation reflects high expected quality of the software, which in turn leads to higher willingness-to-pay and a larger number of customers. On the other hand, negative externality exacerbates the security problems, and subsequently lowers the willingness to pay and the number of customers. How the negative externality influences the design of the reputation system is unclear. Second, we will consider consumers who live for multiple periods. The customers purchase decisions are essentially durable good problem: they may purchase today or delay to the future, anticipating lower prices. An overlapping generation model will be implemented to explore that insight.

Our paper could generate important implications. We will provide guidance for policy makers on the design of a reputation system for software security and the writing of optimal vulnerability policy. Given the diverse practice of infomediaries on software security, our paper is of considerable significance by offering economic rationale. In addition, this study provides valuable managerial implication to software vendors in making optimal patching strategies in a long run.

Reference

Arora, A., Krishman, R., Telang, R. and Yang, Y. “An Empirical Analysis of Vendor Response to Software Vulnerability Disclosure. ” Working Paper. 2005

Arora, A., Nandkumar, A. and Telang, R. “Does Information Security Attack frequency increase with Vulnerability Disclosure? An Empirical Analysis.” (8:5), 2006, pp. 350-362.

Arora, A., Telang, R. and Xu H. “Optimal Policy for Software Vulnerability Disclosure.” Working Paper, 2005.

August, R. and Tunca, T. "Network Software Security and User Incentives." Management Science. (52:11), November 2006, pp. 1703-1720.

Dellarocas, C. "Reputation Mechanism." In Hendershott, T. ed. "Handbook on Economics and Information Systems," 2006.

Ekmekci, M. "Sustainable Reputations with Rating Systems." Working paper, 2005.

Friedman, E. and Resnick, P. 2001. "The Social Cost of Cheap Pseudonyms". Journal of Economics and Management Strategy, (10:2), 2001, pp. 173-199.

Fudenberg, D. and Levine, D.K. "Maintaining a Reputation When Strategies Are Imperfectly Observed." The Review of Economic Studies, (59:3), 1992, pp. 561-579.

Kannan, K. and Telang, R. "Market for Software Vulnerabilities? Think Again." Management Science. (51:5), May 2005, pp. 726-740.

Mailath, G. J. and Samuelson, L. "Who wants a good reputation?" Review of Economic Studies, (68), 2001, pp. 415-441.

Xu, H., Chen, J. and Whinston, A. B. "Audited Reputation." Working paper, 2007.