

WISE 2004 Extended Abstract

Monopoly, Software Quality and Liability

Byung Cho Kim (bckim@andrew.cmu.edu)

Pei-Yu Chen (pychen@andrew.cmu.edu)

Tridas Mukhopadhyay (tridas@andrew.cmu.edu)

Carnegie Mellon University

Introduction

Software has become an indispensable part of our daily lives. Not only large-scale operating systems in businesses but also home electric appliances such as television sets and telephones are powered by hundreds of program instructions. As our society depends more on software, its malfunction becomes more disastrous. For example, the task force investigating the cause of the August blackout that crippled much of the Northeast and parts of Canada concluded that a software failure at Akron, Ohio-based FirstEnergy Corp. may have contributed significantly to the outage (Computer World, 2003). The explosion of software usage has increased the vulnerability of computer systems, highlighting security concerns.

Security experts argue that low quality of software is the major contributor to poor security. In September 2003, Steven Adler, senior security strategist for Microsoft Corp. apologized for the damage and losses caused by the onslaught of computer viruses that have attacked his company's software. As an incentive mechanism for software quality improvement, software liability has been intensely discussed among computer scientists and jurists for decades. Ryan (2003) argues that it is not practical for consumers to create their own security software and that it is reasonable to assume that manufacturers of such products should ensure the reliability of these products. Although the software vendors have not faced liability for security failures yet, some companies are demanding liability clauses in contracts with vendors, holding the vendors responsible for any security breach connected to their software (Fisher, 2002). Due to potential liability as well as increasing customer security awareness, some leading software vendors started making efforts to develop secure software.

In this paper, we analyze the software market where a monopolistic vendor dominates the market. We present an economic model to investigate whether imposing liability on the vendor leads to better security. We consider patching cost as well as liability to be possible incentives for the vendor to improve the initial quality of the software product. Our model is applied in both the fully covered market and the partially covered market. Interestingly, our preliminary results show that liability mechanism is an effective way to improve software quality under certain conditions. This paper contributes to the literature in that it not only gives a clear picture of liability in the software market from economic perspective but also provides implications to managers and policy makers.

Model

Our model is built on the models of vertical quality differentiation (Mussa and Rosen, 1978). We assume that both the vendor and the customers are aware of security issues. In other words, they are capable of predicting loss due to security breaches of the software product. $a(1-k)(1-s)$ represents the expected loss when patching quality is k and the level of security quality is s . Patching quality means how well patch management will be done by the vendor and we assume that it is exogenous, either because patching technology is mainly influenced by the technology available in the marketplace or because this decision is made before product launch. This is a reasonable assumption in that budget for patching is likely to be estimated before the vendor introduces the product to the market. Security quality measures vulnerability of the software to attacks. Bug-free software can be considered to be of perfect security quality. Both parameters are scaled between 0 and 1. The loss is set up in a way that it is entirely preventable if either patching or security quality is perfect. In early 2002, Microsoft stopped all Windows feature development and focused only on analysis of design, code, test plans and documentation. Our model reflects the current phenomenon by analyzing the market where the vendor emphasizes on security development given a certain level of functionality V , which means the features that the customers enjoy with the software.

Without Liability:

In the case where the entire risk is on the customer side, the expected utility of a customer is defined as follows:

$$EU = \theta(V - a(1-k)(1-s)) - p$$

V is functionality and $a(1-k)(1-s)$ represents the expected loss when patching quality is k and the level of security quality is s . p is the price and θ captures customer heterogeneity indicating how much value a customer derives from the software. Without loss of generality, we assume that θ is uniformly distributed on $[\bar{\theta} - 1, \bar{\theta}]$.

A security software vendor's expected profit is

$$E\pi = px - ts^2 - k^2(1-s)$$

where x is the demand for the product, ts^2 represents production cost of software with quality level s . We assume quadratic cost function, so that cost increases as quality level rises at a growing rate. $k^2(1-s)$ is patching cost, which is a quadratic function of patching quality, k . For software with higher initial quality at the product launch, the vendor is likely to achieve a certain level of patching at lower cost. This is captured by having the factor $(1-s)$.

With Liability:

Consider the case where liability is imposed on the vendor side. Then the expected loss does not affect customer utility:

$$EU = \theta V - p$$

Although variable cost of production is assumed to be zero, the expected loss plays a role of variable cost in this case:

$$E\pi = (p - a(1-k)(1-s))x - ts^2 - k^2(1-s)$$

Full Market Coverage

We analyze a three-stage game. In the first stage, the monopolistic vendor decides quality level s given functionality V and patching quality k and in the second, the vendor sets up price p . Then the consumers decide whether or not to buy the product. Denote 1 be the case without liability and 2 be the one with liability. F in the subscript stands for full market coverage. We find that the optimal security quality in each case:

$$s_{1F}^* = \frac{a(\bar{\theta} - 1)(1 - k) + k^2}{2t} \quad \text{and} \quad s_{2F}^* = \frac{a(1 - k) + k^2}{2t}$$

Proposition 1: *When customer valuation is low in the fully covered market, imposing liability on the monopolist gives an incentive for quality improvement. However, when customer valuation is high, liability mechanism may lower security quality.*

This is interesting in that liability does not always guarantee better security. Up to a certain level of customer valuation, imposing liability leads to quality improvement as what has been expected by the practitioners. However, in the market with liability, customers do not appreciate the vendor's effort for security development. Thus, security quality does not directly affect demand. In this market structure, the vendor maximizes profit by minimizing cost. Consequently, in this case, imposing liability on the vendor may discourage the vendor to develop security.

Proposition 2: *In the market without liability, security quality is underprovided by the monopolist compared to social optimum. However, once liability is imposed, the monopolist offers socially optimal level of security:*

$$s_{1F}^* = \frac{a(\bar{\theta} - 1)(1 - k) + k^2}{2t} < \frac{a(\bar{\theta} - \frac{1}{2})(1 - k) + k^2}{2t} = s_{1SP}^*$$

$$s_{2F}^* = \frac{a(1 - k) + k^2}{2t} = s_{2SP}^*$$

Proposition 2 reflects the current market with underprovided security quality. It shows that imposing liability on the vendor may give an incentive for the monopolist to offer socially optimal level of security. This result has a policy implication in that liability should be imposed on the vendor to achieve socially optimal security when the market is fully covered by the monopolist.

Partial Market Coverage

We relax the full market coverage assumption and let the demand be decided by the market. When partial market coverage is allowed, the customers who have higher willingness to pay than $\hat{\theta}$, willingness to pay of the marginal customer, will buy the software product. P in the subscript means partial market coverage. The optimal levels of security quality under both market structures are as follows:

$$s_{1P}^* = \frac{a\bar{\theta}^2(1-k) + 4k^2}{8t} \quad \text{and} \quad s_{2P}^* = \frac{a^2(1-k)^2 - a(1-k)\bar{\theta}V - 2Vk^2}{a^2(1-k)^2 - 4Vt}$$

Proposition 3: *In the market without liability, security quality increases as the customer valuation increases. When patching quality is high, the monopolist offers higher security quality as patching quality gets higher. When patching quality is low, the monopolist lowers security quality as patching quality increases.*

Our result shows that higher customer valuation leads to higher security quality. Interestingly, we show that in the partially covered market without liability, patching does not always compensate initial security development although it may be likely. Under the condition where patching quality is high, offering more secure product is a best strategy for the monopolist as patching quality gets higher. This may be due to the convex nature of the patching cost. The monopolist can reduce the patching cost by offering higher initial quality at the product launch. Thus, in case that patching is costly, security quality increases as patching quality increases.

Proposition 4: *In the market with liability, security quality does not always increase as the customer valuation increases. When functionality is high, high customer valuation leads to better security.*

In the market where the customers are responsible for the entire risk, the customers should value the software product more to get the highly secure software. However, once liability is imposed on the vendor, the customers will not care about the possible loss due to the security breaches. Only thing that the customers appreciate is functionality. Our results show that given a fairly high level of functionality, high valuation leads to better security.

Proposition 5: *When perfect security tool is available in the secondary market, liability mechanism does not make any difference and the monopolist offers product with no security. When perfect patching is available on vendor cost, liability does not matter but the monopolist offers positive security quality to minimize the patching cost. The optimal level is $s_{1P}^* = s_{2P}^* = \frac{1}{2t}$.*

When there is no patching available, liability mechanism leads to better security under the following conditions:

$$\begin{aligned} a^2 - 4Vt < 0 &\Rightarrow s_{1P}^* > s_{2P}^* \\ a^2 - 4Vt > 0, \quad 1 < \bar{\theta} < \bar{\theta}^+ &\Rightarrow s_{1P}^* < s_{2P}^* \\ a^2 - 4Vt > 0, \quad \bar{\theta} > \bar{\theta}^+ &\Rightarrow s_{1P}^* > s_{2P}^* \end{aligned}$$

$$\text{where } \bar{\theta}^+ = \frac{4Vt}{a^2 - 4Vt} + 4t \sqrt{\left(\frac{V}{a^2 - 4Vt}\right)^2 + \frac{a}{2t(a^2 - 4Vt)}}.$$

When the expected loss is small, security quality is higher in the current market where customers are responsible for the risk. When the expected loss is large, the situation is same as in the case

of the fully covered market. Imposing liability on the vendor leads to quality improvement when the customer valuation is small. However, in the market with high customer valuation, liability mechanism may lead to lower quality. This result has a policy implication in the sense that the conditions for the liability mechanism to be an effective way for quality improvement are identified.

Conclusion

In this paper, we analyze the software market where a monopolistic vendor dominates the market. We investigate whether imposing liability on the vendor leads to better security with an economic model. We show that liability does not always improve security level and find conditions under which liability mechanism can be an effective way to improve security. Our paper contributes to the literature in the following ways. We analyze the impact of liability on security level from economic perspective where not much economic research has been done on this issue although it has been intensely argued for decades by computer scientists and jurists. We consider how imposing liability may change a software producer's incentive in security investment. We also provide some guideline to policy makers by showing the conditions where imposing liability leads to better security. In addition, our paper explains the huge security investment of the monopolistic software vendors by identifying incentives for them to do so such as patching cost and potential threat to be liable for their products.

References:

1. Fisher D., "Contracts Getting Tough on Security", *eWeek*, April 15, 2002.
2. Fisk M., "Causes & Remedies for Social Acceptance of Network Insecurity", in *Proceeding of the Workshop on Economics and Information Security*, University of California, Berkeley, May 16-17, 2002.
3. Heckman C., "Two Views on Security Software Liability: Using the Right Legal Tools", *IEEE Security & Privacy* 1(1): 73-75, 2003.
4. Krishnan M. S., C.H. Kriebel, S. Kekre and T. Mukhopadhyay, "An Empirical Analysis of Productivity and Quality in Software Products", *Management Science* 46(6): 745-759, 2000.
5. Mussaic M. and S. Rosen, "Monopoly and Product Quality", *Journal of Economic Theory* 18: 301-317, 1978.
6. Ronnen U., "Minimum Quality Standards, Fixed Costs, and Competition", *RAND Journal of Economics* 22: 490-504, 1991.
7. Ryan D. J., "Two Views on Security Software Liability: Let the Legal System Decide", *IEEE Security & Privacy* 1(1): 70-72, 2003.
8. Sager I. and J. Green, "The Best Way to Make Software Secure: Liability", *Business Week* 3774: 61, March 18, 2003.
9. Spence M., "Monopoly, Quality, and Regulation", *Bell Journal of Economics* 18: 417-429, 1975.
10. Verton D., "Software Failure Cited in Blackout Investigation", *Computer World*, November 24, 2003.