Preliminary and Incomplete

Internet Security, Vulnerability Disclosure, and Software Provision

Jay Pil Choi, Chaim Fershtman, and Neil Gandal[1]

April 5, 2005

Abstract

In this paper, we examine how software vulnerabilities affect firms that sell software and consumers that purchase software. In particular, we model three decisions of the firm: (I) an upfront investment in the quality of the software to reduce potential vulnerabilities, (II) a policy decision whether to announce vulnerabilities, (III) and a price for the software. We also model two decisions of the consumer: (I) whether to purchase the software and (II) whether to apply a patch.

[1] Choi: MSU, Fershtman: Tel Aviv University and CEPR, Gandal: Tel Aviv University and CEPR. We thank several anonymous reviewers from WEIS 2005 for their helpful comments. A research grant from Microsoft is gratefully acknowledged. Any opinions expressed are those of the authors.

1. Introduction

According to a recent study conducted by America Online and the National Cyber Security Alliance (2004), 80 percent of the computers in the US are infected with spyware and almost 20 percent of the machines have viruses. Although some of the so-called killer viruses turned about to be hoaxes, several real viruses have done significant damage. According to the Economist magazine, the Blaster worm and SoBig.F viruses from the summer of 2003 resulted in $35 Billion in damages.[2]

Additionally, it appears that the time between the announcement of a software vulnerability and the time in which that vulnerability can be exploited has declined significantly. According to the Economist, the time from disclosure to attack was six months for the Slammer worm (January 2003) which infected 90% of all unprotected computers within 10 minutes, while the time for the Blaster worm (August 2003) was only three weeks.[3]

The Slammer, Blaster, and Sobig.F worms exploited security patches that had been released by Microsoft. That is, although the "fix" was widely available, relatively few users applied them. In practice, patches are released only when several bugs have been fixed. If a patch is released for each bug, the vulnerability can easily be "reverse engineered" and exploited by hackers. But when a cumulative patch is released, it is more difficult to reverse engineer and find the individual vulnerabilities.

The high cost of these viruses emphasizes the increasing importance of cyber security. Our focus in this paper is how software vulnerabilities affect the firms that sell the software and the consumers that purchase software. In particular, we model three decisions of the firm: (I) An upfront investment in the quality of the software to reduce potential vulnerabilities, (II) a policy decision whether to announce vulnerabilities, (III) and a price for the software. We also model two decisions of the consumer: (I) whether to purchase the software and (II) whether to apply a patch.

[2] http://www.economist.co.uk/science/displayStory.cfm?story_id=2246018.
[3] Ibid.

1.1 Literature Review

Our paper builds on the budding literature at the "intersection" of computer science/engineering and economics on cyber security. See Anderson (2001) for a nice introduction to the topic.[4] Much of the early work in this area has focused on the lack of incentives for individuals or network operators to take adequate security precautions. This is because there is a security externality; individuals (or network operators) will not adequately protect against viruses on their computer (networks), since a large portion of the cost of the spread of the virus is incurred by others.[5]

Varian (2000) argues that assigning liability to network operators would likely lead to a market for insurance.[6] Since insurance firms typically will insure only those who engage in preventive measures,[7] he argues that the incentive for providing security would be increased, that is, the security externality would be internalized.

In the early days of the Internet the Computer Emergency Response Team (CERT) acted as an intermediary between users who reported vulnerabilities to CERT and vendors who produced the software and the patches. CERT typically didn't release information about the vulnerability until the patch was available. Recently, a private market for vulnerabilities has developed where firms such as iDefense act as intermediaries, paying those who report vulnerabilities and providing the information to software users who have subscribed to the service.

Several papers in the literature examine the effects of creating a market for vulnerabilities. Camp and Wolfram (2004) heuristically discuss this issue. Kannan and Telang (2004) employ a

[4] Another helpful source is Anderson's excellent "Economics and Security Resource Page" page: http://www.cl.cam.ac.uk/users/rja14/econsec.html. For a wealth of articles on computer security, see Bruce Schneir's web page at http://www.schneier.com/essays-comp.html.

[5] An interesting question is whether the disclosure of vulnerabilities hurts the market value of the software vendors. If so, this would suggest that software firms have incentives to improve the quality of the software prior to releasing it. Using a data set with 114 vulnerability announcements, Wattal and Telang (2004) show that software firms lose on average 0.76% of their market value when a vulnerability is discovered.

[6] For a formal analysis of system reliability and free riding, see Varian (2002).

[7] Think about the automobile industry – insurers will typically not insure a car against theft unless protective devices such as an alarm and/or an immobilizer has been installed.

formal model to examine whether a market based mechanism is better than the setting in which a public agency (CERT) acted as an intermediary. Schechter (2004) formally models the market for vulnerabilities and Ozment (2004) formally shows how such a market can function as an auction. In these settings, there is no strategic role for a software vendor, which is a main feature of our analysis.

Arora, Telang, and Xu (2004) examine the optimal policy for software vulnerability disclosure. Although they indeed have a strategic software vendor, the vendor strategy is limited to whether it will release a patch and if so when to release the patch. In our paper, we examine incentives for vendors to reduce vulnerabilities and how to price the software, as well as whether to release a patch.[8]

Network effects are prevalent in the computer software industry. However, large networks are also more vulnerable to security breaches, because the success of the network provides hackers with a greater incentive to exploit potential vulnerabilities.[9] Our model incorporates this feature.

We consider a profit maximizing software vendor. Given a fixed level of software quality (security), we first examine whether the software vendor will announce software vulnerabilities and the price that the vendor will charge for the software. We then examine the level of investment in software security.

2. Model

There is a profit maximizing software vendor that decides on price, investment, and disclosure policy. Consumers maximize utility and can either purchase one unit of the software or not purchase at all. If consumers purchase the software and the firm discloses a patch, consumers have to decide whether to install the patch, which is costly. Hackers do not have a formal objective function but there are parameters that describe the outcome of their behavior.

[8] Many of these papers discussed in this section have been presented in workshops on the economics of information security (WEIS). See the references for the web pages of these workshops.
[9] Arora, Krishnan, Nandkumar, Telang, and Yang (2004) find empirical evidence that vulnerability disclosure increases the number of attacks per host and patching decreases the number of attacks per host.

Consumers

Let:

- $\theta \in [1,2]$ – consumer type. We employ a uniform distribution, which means that consumers are "evenly spread" over the interval. Our qualitative results do not depend on this assumption, which is made for simplicity.

- $v_1 + \theta v_2$ – Value of the software to type θ.

- θD – Damage from each security problem to type θ. Hence, both the consumer value and damage are linear functions of consumer type. Our qualitative results do not depend on this feature; we made these assumptions for tractability.

- c – the cost of a downloading/installing a patch to consumers.

We assume that c is a constant and that $c<D$.

Software Vendor

The software vendor maximizes profits and needs to make decisions regarding

- I = the level of investment. This determines the quality of the software. In particular, n(I) = number of security problems. Higher quality software reduces the number of potential bugs, that is, $n'(I)<0$. We also assume that the marginal investment required to find additional vulnerabilities increases in the number of vulnerabilities, or $n''(I)>0$. That is, each additional vulnerability is harder to find and hence requires more investment. Formally, we will assume that $n(I)=1/I$. This functional form emphasizes the fact that it is prohibitively costly to eliminate all vulnerabilities ex ante.

- Whether or not to announce a security problem. $A \in (0,1)$. Announcing a problem means that A=1, while not announcing means that A=0.[10]

- p = price of the software.

[10] With different levels of damage (D), there can be a more sophisticated announcement policy.

Hackers and Technology

Hackers exert effort which is costly. We assume either that (I) they receive monetary rewards for causing damage or (II) that they have an intrinsic motivation that comes from causing damage. In both cases, hackers work harder when they expect to create more damage, i.e., with a larger unprotected network. The following parameters that describe technology and hacker behavior are consistent with either interpretation.

- η-- The probability that the firm will find the problem before the hackers, that is, the percent of the problems that the firm finds first (or the probability that the problem is reported to a firm by a benevolent user). We assume that η is exogenous.

- γN^e = the probability of attack if the problem is not announced, where N^e is the expected number of consumers who purchase the software but do not install a patch. Hence, a larger network (of consumers without patches) increases the probability of attack by hackers.[11] That is, there is a "negative network" effect. $\gamma < 1$ is exogenous and can be thought of as the difficulty of exploiting a vulnerability when reverse engineering (via a patch) is not possible.[12]

- N^e = the probability of attack if the problem is announced. The assumption here is that the release of a patch makes reverse engineering feasible for the hacker and increases the likelihood of attack.[13]

[11] Recall that Arora , Krishnan, Nandkumar, Telang, and Yang, (2004) find empirical evidence that patching decreases the number of attacks per host.

[12] Alternatively, we could assume that (I) γ = the probability of attack if the problem is not announced, (II) the probability of attack if the problem is announced is equal to one, and (III) the expected damage is a function of the number of consumers who do not patch consumers because this provides the incentive for the hackers to attack he network. The formal model is identical under this alternative interpretation.

[13] Arora , Krishnan, Nandkumar, Telang, and Yang, (2004) find empirical evidence that not disclosing vulnerabilities may result in fewer attacks.

Expected Damage

Because of the negative network effect, the expected damage is increasing in the number of consumers on the network that do not have a patch. The expected damage to a consumer of type θ from a vulnerability that is found first by hackers is given by $(1-\eta)\gamma N^e \theta Dn(I)$.

Similarly, the expected damage to a consumer of type θ from a vulnerability that is found first by firm who announces the vulnerability and releases a patch is $\eta N^e \theta Dn(I)$ for consumers that do not have a patch and zero for consumers that have a patch.

Timing

The timing is as follows

- Stage 1: Firms choose the level of investment (I) that determines the number of vulnerabilities, n(I).
- Stage 2: Firms set price (p) and announcement policy (A).
- Stage 3: Consumers make purchasing decisions

Equilibrium

In the second stage, I is given. Hence n*(I) is given as well. For every (n,A,p), we need to define the equilibrium allocation of consumers. Each consumer type θ chooses whether or not to acquire the software, and if so whether to patch. Hence, write

$\Psi(\theta|\ (n,A,p)\ ) \in (\{0,1\},\ \{0,1\})$, where the first $\{0,1\}$ refers to whether to buy the software and the second $\{0,1\}$ refers to whether to patch or not.

(A^*,p^*) and $\Psi(\theta|\ldots)$ is an equilibrium if

(1) $\Psi(\theta|\ldots)$ is the optimal consumer strategy given (A^*,p^*) and $n^*(I)$,

(2) Given $\Psi(\theta|\ldots)$, the firm cannot unilaterally increase profits by changing its strategy.

3. Analysis

In this section, we find the equilibrium defined in the previous section given the investment in security. The game is solved by backwards induction.

3.1 Consumer Adoption Decision

Hence, we begin with stage three, the consumer purchasing decision. Here the quality (I) and the number of vulnerabilities n(I) are given. Similarly the price and announcement policies have been determined.

Let B* be the number of consumer who buy the software and install patches if they become available; similarly let N* be the number of consumer who buy the software, but will not apply patches if they become available. We will assume rational expectations, hence $N^e = N^*$.

We now determine the behavior of consumers with respect of adopting a patch or not and determine the demand for software, given that consumers will react optimally with respect to whether to apply patches. There are three cases and they depend on the cost to consumers of installing a patch (c), the potential damage (D), and the number of software vulnerabilities (n[I]):

Case 1: c>D. No one will ever install a patch.
Case 2: v_2>n(I)D. In such a case, $\theta[v_2\text{-}n(I)D]$ is positive and growing with θ.
Case 3: v_2<n(I)D In this case, $\theta[v_2\text{-}n(I)D]$ declines with θ.

Case 1 is completely uninteresting and we assumed that D>c. In case 2, the value less the damage is increasing in consumer type. In case 3, the opposite is true. At this stage, we focus on case 2, which is in some sense the "natural" case.

We now need to determine the optimal consumer choice under all possible subgames. First consider the case in which the firm commits to announcing vulnerabilities.

3.2 The Firm Announces Vulnerabilities

Let W_p be the net consumer value from buying the software and installing the patch

(1) $W_p(\theta, N^e)= [v_1 + \theta v_2] -\gamma(1-\eta)N^e\theta Dn(I)- \eta n(I)c.$

The first term, $[v_1 + \theta v_2]$, is the consumer valuation which is increasing in type θ; the second term, $\gamma(1-\eta)N^e\theta Dn(I)$, is the expected damage when the hackers find the vulnerabilities before the firm. The expected damage increases in γ, where higher values of γ mean that reverse engineering is easier. The expected damage also increases in $(1-\eta)$ which is the probability that the hacker discovers the vulnerabilities before the firm. The expect damage also increases in N^e, the size of the unprotected consumer network, consumer type θ, and $n(I)$, the number of software vulnerabilities. The third term is the overall expected cost of the patches (to consumers who patch) if the firm finds the problems first.

Let W_{np} be the net consumer value from buying the software, but not installing the patch.

(2) $W_{np}(\theta, N^e)= [v_1 + \theta v_2] -\gamma(1-\eta)N^e\theta Dn(I)- \eta N^e\theta Dn(I),$

The second term is again the damage when the hackers find the vulnerabilities before the firm, while the third term is the expected damage when the firm finds the vulnerabilities before the hackers. There is potential damage in the latter case to consumers do not employ a patch because the release of the patch facilitates reverses engineering.

When the firm sets it price, it will either sell to two sets of consumers (those who patch and those who don't) or it will just sell to those who patch. If we compare equations (1) and (2), the only difference between W_p and W_{np} is the last term in each of the equations. If everybody patches, $N^e = 0$, but in that case "not patching" is a better option for any individual consumer.

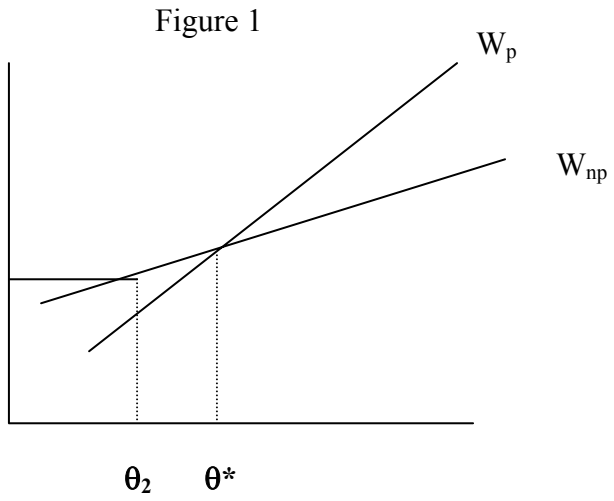
Hence, there cannot be an equilibrium in which all consumers patch. This illustrates that problems with vulnerabilities cannot be solved (exclusively) "ex post" by having everyone patch; because of the incentive to be free rider, such an equilibrium cannot exist.

Thus the firm has two options: (I) announce vulnerabilities and sell the software both to consumers who will apply patches and to consumers who will not apply patches and (II) not announce vulnerabilities. Of course, in the latter case no one patches.

3.3 Sell both to consumers who patch and consumers who do not patch

From equations (1) and (2) above, given D,c and N^e, there is a marginal consumer – denoted $\theta^*(N^e)$ -- such that for $\theta>\theta^*(N^e)$, a consumers installs the patch and for $\theta<\theta^*(N^e)$, a consumer does not install the patch. From these equations, the equilibrium is characterized by $W_{np}(\theta^*, N^*)= W_p(\theta^*, N^*)$, which implies $c=\theta^*DN^*$. That is the expected damage to the marginal consumer who does not install a patch when the firm announces a vulnerability is equal to the cost of the patch. Hence, $B=2-\theta^*$, and $N^*=\theta^*-\theta_1$, where θ_1 is the lowest value consumer that buys the software. (See figure 1 below.)

Figure 1



Since the firm extracts all of the surplus from the marginal consumer,

$p_1=W_{np}(\theta_1, N^*)=[v_1 + \theta v_2] -\gamma(1-\eta)N^*\theta Dn(I)- \eta N^*\theta Dn(I)$.

Note that the equilibrium price is decreasing in the number of vulnerabilities. This will give software firms incentives to invest in security; since this will increase the equilibrium price of the software.

In summary, the equilibrium in this case is characterized by the following four equations:

(I) $N^*=\theta^*-\theta_1$,

(II) $B^*=2-\theta^*$,

(III) $c=\theta^*DN^* \Leftrightarrow W_{np}(\theta^*, N^*)= W_p(\theta^*, N^*)$,

(IV) $p_2=W_{np}(\theta_1, N^*)$.

Condition (II) is redundant, but needed for computing profits, which are

(3) $\pi_1= p_1 (N^*+ B^*) = [v_1 + \theta_1 v_2-\gamma(1-\eta)N^*\theta_1 Dn(I)- \eta N^*\theta_1 Dn(I)] [2-\theta_1]$.

In order to the optimal find N*, note from equations (I) and (III), $N^*=\theta^*-\theta_1=c/[DN^*] - \theta_1$. Thus $\theta_1 = c/[DN^*] – N^*$. This can be substituted into (3). Then the expression can be maximized to find the optimal N* (and hence B*), p_1 and π_1.

3.4 The firm does not announce vulnerabilities

Now we need to analyze what happens if the firm doesn't announce vulnerabilities. For a given network size of unprotected consumers, this is better for consumers who do not patch since it reduces the probability that they will suffer damage. But this is not necessarily more profitable for the firm or better for consumers, since the number of unprotected consumers will be higher under this strategy and hence the probability of hacker damage will be higher as well. This, of course lowers profits and consumer willingness to pay.

The value to the consumer of type θ from no announcement (W_{na}) is given by

$$W_{na}(\theta, N^e)= v_1 + \theta v_2 - \gamma N^* \theta Dn(I)$$

Hence, if there is no announcement, the firm will set the price, $p_2 = W_{na}(\theta_2, N^*)$ where $N^*=2-\theta_2$. (Again note that the equilibrium price of software is decreasing in the number of vulnerabilities.) We can find the optimal θ_2, by maximizing profits:

$$(4) \qquad \pi_2= p_2 N^* = [v_1 + \theta_2 v_2 - \gamma N^* \theta_2 Dn(I)] [2-\theta_2] = [v_1 + \theta_2 v_2 - \gamma(2-\theta_2)\theta_2 Dn(I)] [2-\theta_2],$$

where we have substitute $N^*=2-\theta_2$. Maximizing (4) with respect to θ_2, will give the optimal price (p_2) and network size ($2-\theta_2$). It can be shown that

$$(5) \qquad \theta_2^*[n(I)]= [-b + (b-4ac)^{1/2}] / 2a,$$

where $a[n(I)]=3\gamma Dn(I)$, $b[n(I)]=2v_2 - 8\gamma Dn(I)$, and $c[n(I)]=4\gamma Dn(I)+ v_1-2v_2$.[14]

3.5 Firm choice of price and vulnerability announcement policy

We need to compare π_1 and π_2, in order to determine the firm's optimal choice in stage 2 and the resulting equilibrium. Obviously the results will depend on the exogenous parameters (v_1,v_2,c,D, γ, and η) as well as the investment in the first stage. We now provide intuition for how the optimal firm strategy changes when the exogenous parameters change.[15]

For small γ and large η, the firm will not announce vulnerabilities. This makes sense since when γ is very small, it is very difficult for the hacker to reverse engineer without an announcement. This makes consumers willing to pay a relatively high price when the firm doesn't announce

[14] Differencing (4) with respect to)θ_2 yields the following first order condition: $- v_1 + 2v_2 - 4\gamma Dn(I) - 2v_2\theta_2 + 8\gamma Dn(I)\theta_2 - 3\gamma Dn(I)\theta_2^2=0$.

[15] We confirmed this intuition by using numerical analysis.

vulnerabilties. Indeed, in such cases, the equilibrium price will be higher when the firm does not announce vulnerabilities, despite the larger unprotected network. The intuition is when γ is small, it is very difficult for hackers to find vulnerabilities when the firm does not provide patches. Announcing a vulnerability in this case significantly increases the probability that there will be hacker damage; in such a case consumer willingness to pay is higher when vulnerabilities are not announced.

When η is large, there is a high probability that the firm will find the problem first and, hence, when the firm announces this significantly increases the probability of hacking and lowers the price (W_{np}) that unprotected consumers are willing to pay. Hence, when γ is small and η is large the firm will not announce vulnerabilities. Otherwise the firm will announce vulnerabilities.

An increase in the cost of the patch (c) and/or a decrease in the damage (D) reduce the probability that the software vendor will announce a vulnerability. Additionally, if the vulnerability is announced, an increase in c and a decrease in D mean that fewer consumers will apply patches.

The parameters v_1 and v_2 affect the affect consumer valuations, pricing, and profits, but have little effect on the optimal disclosure choice of the firm.

4. Firm Choice of Investment

We now examine the level of investment, the first stage decision of the software vendor. Reducing the number of vulnerabilities increases the profitability of both strategies in the second stage (announcing and not announcing vulnerabilities) because if hackers indeed find the vulnerabilities, there will be less damage. As we saw, this raises the willingness of consumers to pay for the software. It will also typically increase the number of consumers who purchase software.

Recall that n(I)=1/I. Hence in the first stage the firm maximizes π= p* [n(I)] N* [n(I)] – I.

In the case in which the firm announces vulnerabilities, an analytical solution for p* and N* is not possible. Hence we solve this numerically. In the case in which the firm does not announce vulnerabilities, we substitute (5) into (4) and differentiate the profit expression.

We consider examples in section 6.

5. Efficiency

First suppose that the firm chooses to announce the vulnerabilities and provide patches.

For the consumers that patch ($\theta \in (\theta^*, 2)$), consumer surplus (CS) is

$$CS\ (patch) = \{v_1 - \eta n(I)c\}[2-\theta^*] + \{v_2 - \gamma(1-\eta)N^*Dn(I)\}(4-\theta^{*2})/2 - p_1(2-\theta^*)$$

For the consumers that do not apply patches ($\theta \in (\theta_1, \theta^*)$), consumer surplus is

$$CS\ (don't\ patch) = v_1[\theta^*-\theta_1] + \{v_2 - \gamma(1-\eta)N^*Dn(I) - \eta N^*Dn(I)\}(\theta^{*2}-\theta_1^2)/2 - p_1(\theta^*-\theta_1)$$

Profits are $\pi = p_1(2-\theta_1) - I$.

$$\text{Hence total surplus (TS)} = \{v_1 - \eta n(I)c\}[2-\theta^*] + \{v_2 - \gamma(1-\eta)N^*Dn(I)\}(4-\theta^{*2})/2 + v_1[\theta^*-\theta_1] + \{v_2 - \gamma(1-\eta)N^*Dn(I) - \eta N^*Dn(I)\}(\theta^{*2}-\theta_1^2)/2 - I$$

Now suppose that the firm chooses not to announce the vulnerabilities.

$CS = v_1(2-\theta_2) + v_2(4-\theta_2^2)/2 - \gamma Dn(I)(4-\theta_2^2)/2 - p_2(2-\theta_2)$, and
$\pi = p_2(2-\theta_2) - I$.

Hence, total surplus (TS) is

$$TS = v_1(2-\theta_2) + v_2(4-\theta_2^2)/2 - \gamma Dn(I)(4-\theta_2^2)/2 - I.$$

6. Examples

In order to illustrate the some of the key results, we consider several examples. The choice of v_1, v_2, c, and D are not critical for our qualitative results. As discussed in section 3.5, γ and η are the key parameters.[16] In these examples, we show how the equilibrium and total surplus change when γ changes. Although, we do not change η in these examples, as we discussed in section 3.5, higher values of η make it more likely that the firm will not announce vulnerabilities.[17]

Example 1: The exogenous parameters are: v_1=0.1, v_2=20, D=8, γ=0.5, η=0.5, c=2

First suppose that the firm chooses to announce the vulnerabilities and provide patches. It can be shown that the firm maximizes profits by choosing I=1.1. The equilibrium is such that

$$I=1.1,\ p=19.39,\ \theta_1=1.021,\ N^*=0.204,\ B^*=0.775,\ \pi=17.88,\ \text{and}\ TS=27.14$$[18]

If the firm chooses not to announce software vulnerabilities, it maximizes profits by choosing I=1.95. The equilibrium is such that

$$I=1.95,\ p=19.13,\ \theta_2=1.054,\ N^*=0.946,\ \pi=16.15,\ \text{and}\ TS=24.07$$[19]

The equilibrium investment in this case is quite high because the probability that hackers will be able to exploit vulnerabilities in the absence of an announcement (and a patch) is relatively high.

Comparing the two choices, the firm would announce vulnerabilities in this case. In this case, social surplus is maximized when the firm announces vulnerabilities. Hence the firm's announcement policy corresponds with the social optimal policy.

[16] As noted in section 3.5, an increase in the cost of the patch (c) and/or a decrease in the damage (D) reduce the probability that the software vendor will announce vulnerabilities.
[17] In equilibrium, $v_2>n(I)D$ in all of these examples. Hence, we are indeed in case 2.
[18] TS in this case is maximized at TS=27.33 when I=1.65.
[19] TS in this case is maximized at TS=24.48 when I=2.85.

Example 2: Same as example 1, except that γ=0.2.

Compared to example 1, there is lower probability that hackers will be able to exploit the vulnerabilities in the absence of an announcement.

First suppose that the firm chooses to announce the vulnerabilities and provide patches. It can be shown that the firm maximizes profits by choosing I=1. The equilibrium is such that

$$I=1.0,\ p=19.39,\ \theta_1=1.015,\ N^*=0.205,\ B^*=0.78,\ \pi=18.11,\ \text{and}\ TS=26.16.^{20}$$

Now suppose that the firm chooses not to announce software vulnerabilities. It can be shown that the firm maximizes profits by choosing I=1.25. The equilibrium is such that

$$I=1.25,\ p=19.45,\ \theta_2=1.03,\ N^*=0.97,\ \pi=17.59,\ \text{and}\ TS=26.33.^{21}$$

Note that there is a reduction in the equilibrium investment under both cases ("announce" and "no announce") relative to example 1. This is because it is more difficult for hackers to exploit vulnerabilities in the absence of an announcement.

Comparing the two choices, the firm would announce vulnerabilities, although social surplus is higher in the case in which vulnerabilities are not announced.

Example 3: Same as example 1, except that γ=.05.

In this example it very unlikely that hackers will be able to exploit the vulnerabilities in the absence of an announcement.

If the firm chooses to announce the vulnerabilities and provide patches, it maximizes profits by choosing I=0.95. The equilibrium is such that

[20] TS in this case is maximized at TS=26.19 when I=1.5.
[21] TS in this case is maximized at TS=26.57 when I=1.8.

$$I=0.95,\ p=19.47,\ \theta_1=1.015,\ N^*=0.205,\ B^*=0.78,\ \pi=18.24 \text{ and } TS=27.77.^{22}$$

If the firm chooses not to announce software vulnerabilities, it maximizes profits by choosing I=0.65. The equilibrium is such that

$$I=0.65,\ p=19.75,\ \theta_2=1.013,\ N^*=0.987,\ \pi=18.84, \text{ and } TS=28.27^{23}$$

Comparing the two choices, the firm would not announce vulnerabilities in this case. In this case, social surplus is indeed maximized when the firm does not announce vulnerabilities.[24] Again the equilibrium investment in security is lower since in the case of no announcement, the probability that hackers will be able to exploit the vulnerability is much lower. This case confirms the intuition that (I) it is not always optimal for the firm to announce vulnerabilities and (II) higher investment in security may not necessarily raise total surplus.

7. Further Discussion and Preliminary Conclusions

In this paper we developed a model that endogenizes three decisions of the firm: (I) An upfront investment in the quality of the software to reduce potential vulnerabilities, (II) a policy decision whether to announce vulnerabilities, (III) and a price for the software. We also modeled two decisions of the consumer: (I) whether to purchase the software and (II) whether to apply a patch. We examined some examples and showed that:

[22] TS in this case is maximized at TS=27.88 when I=1.2.
[23] TS in this case is maximized at TS=28.36 when I=0.9.
[24] Note that the market equilibrium of the "no announcement" case leads to higher total surplus than the socially optimal total surplus obtained in the case of announcing vulnerabilities.

(I) firms are likely to announce vulnerabilities when there is a relatively high probability that hackers will be able to exploit the vulnerabilities in the absence of an announcement. This policy coincides with the socially optimal announcement policy.

(II) when the when there is a relatively low probability that hackers will be able to exploit the vulnerabilities in the absence of an announcement, firms do not announce vulnerabilities and it is socially optimal not to announce them.

(III) it is possible that firms will announce vulnerabilities even when it is socially optimal not to announce them. This result obtains for intermediate values of the probability that hackers will be able to exploit the vulnerabilities in the absence of an announcement.

Our paper, of course, leaves many research questions unanswered. In this paper, we did not allow for intermediaries, like CERT, who obtains vulnerability information from end users and eventually publishes this information. We showed that in our setting, announcing vulnerabilities may lead to lower social surplus. If there is an intermediary, it may not always be possible for a firm to adopt a "do not announce" policy.

Additionally, we assumed a single software vendor and did not examine the time at which software is released. With competition in software provision and a dynamic setting with new consumers over time, there would potentially be two additional effects: (I) there would likely be increased investment in reducing software vulnerabilities due to competition and (II) If consumer valuations depended on network size, software firms might have an incentive to release products earlier to build up an installed base.

References

American Online and the National Cyber Security Alliance, *AOL/NCSA Online Safety Study*, October 2004.

Arora, A., Krishnan, R., Nandkumar, A., Telang, R., and Y. Yang, "Impact of Vulnerability Discolsure and Patch Availabilitry – An Empirical Analysis, mimeo 2004, available at http://www.dtc.umn.edu/weis2004/telang.pdf.

Aora, A., Telang, R., and X., Hao, "Optimal Policy for Software Vulnerability Disclosure," Carnegie Mellon Working Paper, 2004

Anderson, R., (2001), "Why Information Security is Hard," available at http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/econ.pdf.

Camp, L.J., and C. Wolfram, "Pricing Security," in L.J. Camp and S. Lewis, eds., Economics of Information Security, vol. 12, Advances in Information Security. Springer-Kluwer, 2004.

Kannan, K., and R. Telang, "Market for Software Vulnerabilities? Think Again," Carnegie Mellon Working Paper, 2004.

Kawamoto, D., *Study: Few Corporations Use Anti-Spyware Tools*, CNET News, October 27, 2004.

Ozment, A., "Bug Auctions: Vulnerability Markets Reconsidered," mimeo, available at http://www.dtc.umn.edu/weis2004/ozment.pdf

Schechter, S., "Computer Security, Strength and Risk: A Quantitative Approach," 2004, available at http://www.eecs.harvard.edu/~stuart/papers/thesis.pdf

Varian, H., "Managing Online Security Risks," New York Times; New York, N.Y.; Jun 1, 2000, available at http://www.sims.berkeley.edu/~hal/people/hal/NYTimes/2000-06-01.html.

Varian, H., 2002, "System Reliability and Free Riding," available at http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/econws/49.pdf.

Wattal, S., and R. Telang, "Effect of Vulnerability Disclosure on Market Value of Software Vendors -- An Event Study, CMU mimeo, 2004.

WEIS 2002: Held at UC-Berkeley. Papers are available at http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurity/.

WEIS 2003: Held at the University of Maryland. Papers are available at http://www.cpppe.umd.edu/rhsmith3/agenda.htm.

WEIS 2004: Held at the University of Minnesota. Papers are available at http://www.dtc.umn.edu/weis2004/agenda.html.