DISCUSSION PAPER SERIES

No. 5269

INTERNET SECURITY, VULNERABILITY DISCLOSURE AND SOFTWARE PROVISION

Jay Pil Choi, Chaim Fershtman
and Neil Gandal

INDUSTRIAL ORGANIZATION

Centre for Economic Policy Research

www.cepr.org

Available online at: **www.cepr.org/pubs/dps/DP5269.asp**

ISSN 0265-8003

INTERNET SECURITY, VULNERABILITY DISCLOSURE AND SOFTWARE PROVISION

Jay Pil Choi, Michigan State University
Chaim Fershtman, Tel Aviv University and CEPR
Neil Gandal, Tel Aviv University and CEPR

Discussion Paper No. 5269
October 2005

Centre for Economic Policy Research
90–98 Goswell Rd, London EC1V 7RR, UK
Tel: (44 20) 7878 2900, Fax: (44 20) 7878 2999
Email: cepr@cepr.org, Website: www.cepr.org

CEPR Discussion Paper No. 5269

October 2005

ABSTRACT

Internet Security, Vulnerability Disclosure and Software Provision*

In this paper, we examine how software vulnerabilities affect firms that license 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; and (iii) 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.

JEL Classification: L86 and O3
Keywords: internet security, network effects, software and vulnerabilities

Jay Pil Choi
Professor of Economics
Department of Economics
Michigan State University
101 Marchall Hall
East Lansing, MI 48824-1038
USA
Tel: (1 517) 353 7281
Fax: (1 517) 432 1068
Email: choijay@msu.edu

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=154834

Chaim Fershtman
The Eitan Berglas School of Economics
Tel-Aviv University
Tel Aviv 69978
ISRAEL
Tel: (972 3) 640 7167
Fax: (972 3) 640 9908
Email: fersht@post.tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=104930

Neil Gandal
Department of Public Policy
Tel Aviv University
Tel Aviv 69978
ISRAEL
Tel: (972 3) 640 6742
Fax: (972 3) 640 7382
Email: gandal@post.tau.ac.il

For further Discussion Papers by this author see:
www.cepr.org/pubs/new-dps/dplist.asp?authorid=144974

*We thank Jacques Lawarree, Shlomit Wagman, several anonymous reviewers from WEIS 2005, and seminar participants from WEIS 2005 for their helpful comments. We are grateful to Ity Shurtz for research assistance. A research grant from Microsoft is gratefully acknowledged. Any opinions expressed are those of the authors.

Submitted 13 September 2005

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 vulnerabilities even though security patches had been released by Microsoft. That is, although the patches were widely available, relatively few users applied them.[4]

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 license[5] 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.
[4] In practice, patches are typically 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.
[5] Like other products based on intellectual property, the intellectual property in software is typically "licensed" for use, not sold outright. Someone who purchases a music CD buys the physical CD and the right to play the music under specific circumstances (which do not include the right to play it on the radio, etc). Software is similarly "licensed for use."

We find that firms announce vulnerabilities and invest a relatively large amount in security when it is very easy for hackers to attack in the absence of an announcement; this policy coincides with the socially optimal policy. Similarly we find that when it is very difficult for hackers to attack in the absence of an announcement, firms do not announce vulnerabilities and invest relatively little in security; the policy again coincides with the socially optimal policy. In this case, a policy of "no announcement" and relatively little investment leads to higher total surplus than a policy of "announcement" and relatively high investment in security. In an intermediate region, the firm will not announce vulnerabilities even when it is socially optimal to announce them. Numerical simulations suggest that this inefficiency occurs for a relatively small portion of the parameter space.

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 an introduction to the topic.[6] 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.[7]

Varian (2000) argues that assigning liability to network operators would likely lead to a market for insurance.[8] Since insurance firms typically will insure only those who engage in preventive measures,[9] he argues that the incentive for providing security would be increased, that is, the security externality would be internalized.

---

[6] Another helpful source is Anderson's "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.

[7] 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.

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

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

Computer Emergency Response Team/Coordination Center (CERT/CC) is a public agency that acts as an intermediary between users who reported vulnerabilities to CERT/CC and vendors who produced the software and the patches. When informed by a user about a vulnerability, CERT/CC conducts research into the matter. If the user has indeed uncovered a security vulnerability, CERT/CC then informs the software vendor and gives it a 45 day "vulnerability window." This allows the firm time to develop a patch. After the 45 day period, CERT/CC will disclose the vulnerability even if a patch has not been made 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.[10] Camp and Wolfram (2004) heuristically discuss this issue. Kannan and Telang (2004) employ a formal model to examine whether a market based mechanism is better than the setting in which a public agency (CERT/CC) 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. August and Tunca (2005) have a strategic software vendor as well, but the vendor strategy is limited to pricing the software. In our paper, we examine incentives for vendors to invest in quality in order to reduce vulnerabilities, how vendors will price the software, and whether vendors will announce vulnerabilities and release a patch.

Network effects are prevalent in the computer software industry. Network effects are typically thought to benefit consumers and firms that have coalesced around a standard. However, large

[10] 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.

networks are also more vulnerable to security breaches, because the success of the network provides hackers with a greater incentive to exploit potential vulnerabilities. Our model incorporates this "negative network effect" feature.

We consider a profit maximizing software vendor in a three-stage game. In the first stage, the firm chooses the level of investment that determines the quality of software, i.e., the number of vulnerabilities. In the second stage, the firm sets the software price and announcement policy. In the third stage, consumers make purchasing decisions. As usual, we apply backward induction to solve the subgame perfect equilibrium of this dynamic game. More specifically, 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 investment, price, 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.

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. This assumption reflects the fact that high valuation consumers tend to suffer more damage from an attack. 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 $c<D$. If $c>D$, no one will ever install a patch and the analysis is completely uninteresting.

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.[11] 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 and simultaneously releasing a patch means that $A=1$,[12] while not announcing means that $A=0$.

- p = price of the software.

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

[11] The model assumes a fixed number of vulnerabilities at release time. That is not always the case; for example new technologies can create new vulnerabilities. The results are robust to the number of vulnerabilities at release time being a multiple of n(I).

[12] In our setting, an announcement is only made when the patch is available; this is the typical practice in the industry.

a larger unprotected network.[13] The following parameters describe technology and hacker behavior.

- 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.

- γ -- The probability of attack if the problem is not announced. If the firm announces the vulnerability and releases a patch, the probability of attack equals one.[14]

- γN^e -- The expected damage from an attack, where N^e is the expected number of consumers who purchase the software but do not install a patch. The expected damage is a function of the number of consumers who do not patch, because this provides the incentive for hackers to exert more effort and make a more sophisticated attack.[15]

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 that 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 who have a patch.

---

[13] Of course, hacker incentives are more complicated. Some hackers, for example, are likely motivated by the possibility of recognition within the hacker community that comes from finding and exploiting vulnerabilities. See Nissenbaum (2004) and the references cited within.

[14] 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.

[15] The formal model is identical under the following interpretation: γN^e = the probability of attack if the problem is not announced. Hence, a larger network (of consumers without patches) increases the probability of attack by hackers. In this interpretation, $\gamma<1$ represents the difficulty of exploiting a vulnerability when reverse engineering (via a patch) is not possible, and N^e is the probability of attack if the problem is announced ($\gamma=1$ when 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. Arora, Krishnan, Nandkumar, Telang, and Yang (2004) find empirical evidence that not disclosing vulnerabilities may result in fewer attacks.

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 and patching 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. Then (A^*,p^*) and $\Psi(\theta|\ldots)$ is an equilibrium of the subgame given n(I) if (1) $\Psi(\theta|\ldots)$ is the optimal consumer strategy given (A^*,p^*) and n(I) and (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 to adopting a patch or not and determine the demand for software, given that consumers will react optimally with respect to whether to apply patches. We assume that $v_2 > n(I)D$. This insures that $\theta[v_2 - n(I)D]$ is positive and growing with θ, i.e., that is, the value less the damage is increasing in consumer type.

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 there is a higher probability of attack when vulnerabilities are not announced. 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 who do not employ a patch because the release of the patch increases the probability of attack to one.

When the firm sets its price, it will either license to two sets of consumers (those who patch and those who don't) or it will just license 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.

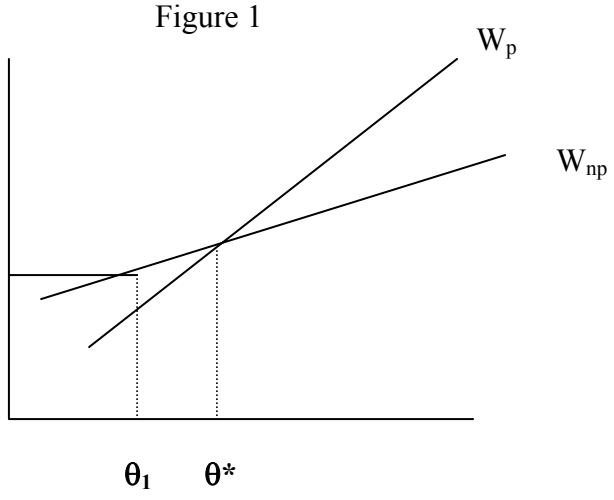
Thus the firm has two options: (I) announce vulnerabilities and license the software both to consumers who will apply patches and to consumers who will not apply patches and (II) not announce vulnerabilities. Of course, in the latter case no one patches.

3.3 License 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 $c=\theta^*DN^*$.[16] 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.

[16] To insure that the highest type ($\theta=2$) has an incentive to patch even in the worst case scenario (i.e., $N^*=1$), we require $c<2D$. If we had assumed that $\theta\in[0,1]$, this condition would be $c<D$. All of the qualitative results are the same under $\theta\in[0,1]$. We chose $\theta\in[1,2]$ so that we can allow for the possibility that $v_1=0$, that is, there is no standalone utility.

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 find the optimal 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 expected 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 substituted $N^*=2-\theta_2$. 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

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

where $a[n(I)]=3\gamma Dn(I)$, $b[n(I)]=2v_2- 8\gamma Dn(I)$, and $c[n(I)]=4\gamma Dn(I)+ v_1-2v_2$.[17]

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

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 turn to the firm's choice of investment.

4. Firm Choice of Investment

Here we 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 (regardless of 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 profits, which can be expressed as 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.

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 $\pi = p_1(2-\theta_1) - I$. Hence total surplus (TS) = CS (patch) + CS (don't patch) + π =

$\{v_1 - \eta n(I)c\}[2-\theta^*] + \{v_2 - \gamma(1-\eta)N^*Dn(I)\}(4-\theta^{*2})/2 + v_1[\theta^*-\theta_1] + \{v_2 - \gamma(1-\eta)N^*Dn(I) - \eta N^*Dn(I)\}(\theta^{*2}-\theta_1^2)/2 - I.$

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

6. Results

In order to illustrate the key results, we first consider several examples.[18] Since γ is the critical parameter in the model, these simulations focus on changes in this parameter. In section 6.2, we provide intuition for our results and for how the other parameters affect the results.

6.1 Examples

Example 1: The exogenous parameters are: $v_1=0.1$, $v_2=20$, $D=4$, $\gamma=0.5$, $\eta=0.5$, $c=2$

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=1.05. The equilibrium is such that

$I=1.05$, $p=19.31$, $\theta_1=1.013$, $N^*=0.3632$, $B^*=0.6237$, $\pi=18.01$, and $TS=27.38$

If the firm chooses not to announce software vulnerabilities, it maximizes profits by choosing I=1.40. The equilibrium is such that

$I=1.40$, $p=19.39$, $\theta_2=1.0356$, $N^*=0.9644$, $\pi=17.30$, and $TS=25.88$

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

[18] In equilibrium, $v_2 > n(I)D$ in all of these examples. Hence, we are indeed in case 1.

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.

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=0.95. The equilibrium is such that

$$I=0.95,\ p=19.39,\ \theta_1=1.011,\ N^*=0.3637,\ B^*=0.6252,\ \pi=18.23,\ \text{and}\ TS=27.71.$$

Now suppose that the firm chooses not to announce software vulnerabilities. It can be shown that the firm maximizes profits by choosing I=0.90. The equilibrium is such that

$$I=0.90,\ p=19.62,\ \theta_2=1.021,\ N^*=0.979,\ \pi=18.32,\ \text{and}\ TS=27.47.$$

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 not announce vulnerabilities, although social surplus is higher in the case in which vulnerabilities are announced.

Example 3: Same as example 1, except that γ=0.05.

Here, there is a very low 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=0.85. The equilibrium is such that

$$I=0.85,\ p=19.41,\ \theta_1=1.011,\ N^*=0.3637,\ B^*=0.6252,\ \pi=18.35,\ \text{and}\ TS=27.86.$$

Now suppose that the firm chooses not to announce software vulnerabilities. It can be shown that the firm maximizes profits by choosing I=0.45. The equilibrium is such that

$$I=0.45,\ p=19.83,\ \theta_2=1.009,\ N^*=0.991,\ \pi=19.21,\ \text{and}\ TS=28.81.$$

Note that there is a reduction in the equilibrium investment under both cases ("announce" and "no announce") relative to example 2. This is because it is very difficult for hackers to exploit vulnerabilities in the absence of an announcement. In the "no announcement" case, the firm indeed invests very little in finding vulnerabilities. As a comparison of the two options show, a policy of "no announcement" and little investment is both socially optimal and profit maximizing. Example 3 confirms the intuition that (i) it is not always socially optimal for the firm to announce vulnerabilities and (ii) higher investment in security may not necessarily raise total surplus.

6.2 General Results

By using numerical methods, we are able to span the parameter space for γ and η. Figure 2 shows the case when $v_1=0.1$, $v_2=20$, D=4, and c=2, while Figure 3 shows the case when $v_1=0.1$, $v_2=20$, D=8, and c=2.[19]

Figures 2 and 3 show the iso-profit curve, where the firm is indifferent between announcing (A=1) and not announcing vulnerabilities (A=0). These figures also show the iso-suplus curve, where social surplus is equal when vulnerabilities are announced (A*=1) and when they are not announced (A*=0). The figures show that there is a relatively small region in which the firm

[19] We confirmed that we are in a "case 1" equilibrium for all parameter values in Figure 2 and Figure 3.

will not announce vulnerabilities (A=0) even when it is socially optimal to announce them (A*=1).[20]

The figures show that for small γ the firm will not announce vulnerabilities. This makes sense since when γ is very small, it is very difficult for the hacker to attack without an announcement. This makes consumers willing to pay a relatively high price when the firm doesn't announce vulnerabilities. Indeed, in such cases, the equilibrium price will be higher when the firm does not announce vulnerabilities, despite the larger unprotected network. (See example 3.) Announcing vulnerabilities in this case significantly increases the expected hacker damage. Hence, when γ is small the firm will not announce vulnerabilities. Otherwise the firm will announce vulnerabilities.

A decrease in the cost of the patch (c) and/or an increase in the damage (D) shift both the iso-profit line and the iso-surplus lines up and to the left. That is, the firm announces vulnerabilities for a larger region of the parameter space and it is socially optimal to do so. The "inefficient" region is also reduced. See figure 3.

If the vulnerability is announced, a decrease in c and/or an increase in D mean that more consumers will apply patches. For example, when D=8 (D=4), and the other parameters are as in example 1, the unprotected network size is N*=0.204 (N*=0.3632) when firm chooses to announce the vulnerabilities and provide patches. Hence the equilibrium unprotected network is much larger when the potential damage is smaller.

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

[20] The discrepancy arises in this region because the firm's announcement policy depends on its impact on the *marginal* consumer whereas the social planner's incentive depends on its effect on the *average* consumer. Since the marginal consumer is the lowest type who purchases the software and has the least incentive to apply patch, the firm has less than the socially optimal incentives to announce vulnerabilities and supply a patch.

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 showed that:

(i) firms announce vulnerabilities and invest a relatively large amount in security when it is very easy for hackers to attack in the absence of an announcement; this policy coincides with the socially optimal policy.
(ii) when it is very difficult for hackers to attack in the absence of an announcement, firms do not announce vulnerabilities and invest relatively little in security; the policy again coincides with the socially optimal policy. In this case, a policy of "no announcement" and relatively little investment leads to higher total surplus than a policy of "announcement" and relatively high investment in security.
(iii) in intermediate cases, the firm will not announce vulnerabilities even when it is socially optimal to announce them.

Our paper, of course, leaves some research questions unanswered. In this paper, we did not allow for intermediaries, like CERT/CC, who obtain vulnerability information from end users and encourage firms to develop patches and eventually disclose this information. Such an intermediary could improve welfare in the intermediate cases discussed above. Our simulations suggest, however, that this inefficiency occurs for a relatively small portion of the parameter space. Hence, this suggests that one should be cautious before regulating this market.

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.

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

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.

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

August, T., and T. Tunca, "Network Software Security and User Incentives," Stanford University mimeo, 2005.

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.

Helen Nissenbaum, "Hackers and the Contested Ontology of Cyberspace, New Media and Society, 6:195-217, 2004, available at www.nyu.edu/projects/nissenbaum/papers/hackers.pdf

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.

WEIS 2005: Held at Harvard University. Papers are available at http://infosecon.net/workshop/schedule.php.

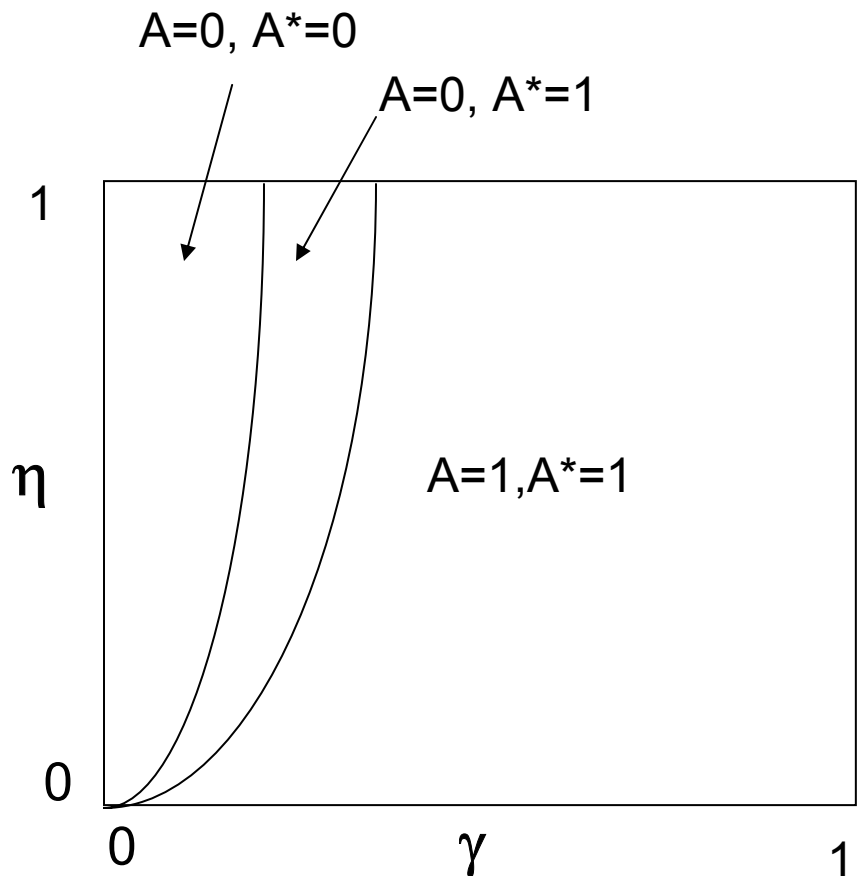


Figure 2: v_1=0.1, v_2=20, D=4, c=2

A= Announcement Policy of Firm
A*= Socially Optimal Policy

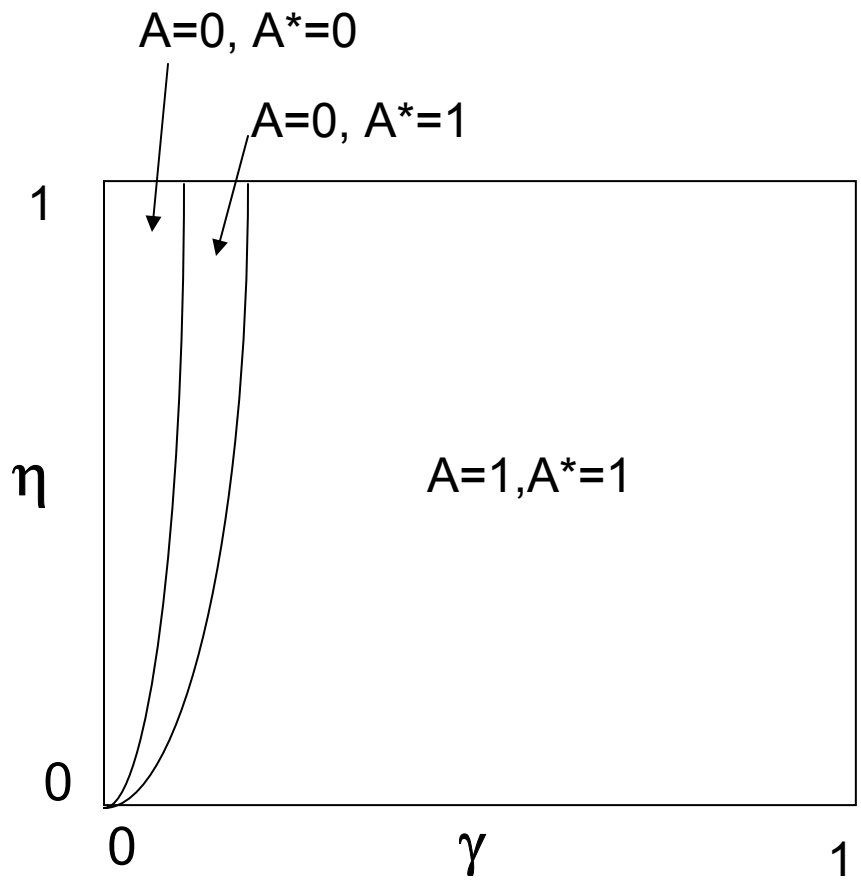


Figure 3: v_1=0.1, v_2=20, D=8, c=2

A= Announcement Policy of Firm
A*= Socially Optimal Policy