Rationally Opting for the Insecure Alternative: Negative Externalities and the Selection of Security Controls

Craig S. Wright and Tanveer A. Zia

School of Computing and Mathematics
Charles Sturt University, NSW 2678
cwrigh20@postoffice.csu.edu.au, tzia@csu.edu.au

Abstract. As with all aspects of business and the economy, information security is an economic function. Security can be modeled as a maintenance or insurance cost as a relative function but never in absolute terms. As such, security can be seen as a cost function that leads to the prevention of loss, but not one that can create gains (or profit). With the role of a capital investment to provide a return on investment, security is a defense against unforeseen losses that cost capital and reduce profitability. In this paper we assess the individual security cost and model our assessment in economic terms. This assessment is vital in determining the cost benefit in applying costly security controls in our systems in general and software in particular.

Keywords: Software Development Life Cycle, Model Checking, Software Verification, Empirical studies.

1 Introduction

Absolute security does not exist and nor can it be achieved. The statement that a computer is either secure or not is logically falsifiable [6], all systems exhibit a level of insecurity. An attacker with sufficient resources can always bypass controls. The goal is to ensure that the economic constraints placed upon the attacker exceed the perceived benefits to the attacker [15]. This generates a measure of relative system security in place of the unachievable absolute security paradigm that necessarily results in a misallocation of resources.

The result is that security is a relative risk measure that is related to organisational economics at the micro level and the economics of national security toward the macro level. This consequentially leads to a measure of security in terms of one's neighbour. The question is not, "*am I secure*", but rather, "am *I more secure than my neighbour*?"

This can be assessed in many ways as any other system is your neighbour on the Internet when viewed from the perspective of a Worm. Conversely, targeted attacks have a purpose. Neighbours may be other government systems, critical infrastructure, and a class of companies or an industry sector. In each instance, security is achieved in relative terms.

The rest of the paper is organised as follows: In section 2 we assess individual security costs. We then provide analysis and discussion on security assessment in terms of its economic value in Section 3. Finally, the paper is concluded in Section 4.

Á. Herrero and E. Corchado (Eds.): CISIS 2011, LNCS 6694, pp. 207–214, 2011.

2 Assessing Individual Security Costs

The most effective security solution is that which provides the best level (that which is optimised) for "the least cost". Costs to the consumer are minimised at the point where security costs exactly equal the expected loss that is associated with the risk function.

> *More security costs = higher costs to the consumer.*
> *Higher expected loss from risk = higher costs to the consumer.*

As expenditure on security is expected to decrease the expected loss, the costs to the consumer are minimised were the additional expenditure of $1 on security reduces the expected risk based loss by exactly $1.

Security is a cost function that is passed to the consumer if profitability is to be retained or which reduces profit directly where alternatives exist (this is the product is elastic or consumers are willing to reduce their use if costs increase). The expected cost formula for the supply of these types of services against a loss function can be expressed by:

$$C_s = D\left(x, y\right) + x + y \tag{1}$$

Where the loss function *D(x,y)* and the damage to *x* (the producer) and *y* (the consumer) are modelled arithmetically. As in all areas of economics, the marginal gains in D_x offset those of D_y.

In these calculations, calculations, D_{xy} D_{xy} > D_{xx} D_{xy} which creates the inference that the inputs are substitutes. As the producer spends more on security, the consumer spends less and vice versa. The exact composition of these values varies based on the nature of the product with elastic supply being affected more than an inelastic supply.

The real issue and goal in security becomes the creation of a Cournot-Nash equilibria [11]. This is an outcome where X_e and Y_e are together form a Cournot-Nash equilibria for a given value of Y_e the *x* which maximises *X*'s utility is X_e and given X_e that *y* which maximises *Y*'s utility is Y_e. This does not require that the equilibria be Pareto optimal [12].

At present, the cost functions directed towards many industries (such as banks in regulated countries including Australia) are sufficient in that there is but a trivial increase in marginal demand for the consumer for an incremental increase in security expenditure. The producing company is likely to do little and that which they do conduct has a minimal effect. For instance, Microsoft is unlikely to greatly improve the security of its operating system through minimising patches due to the increasing cost of finding additional bugs in its software. If it did so, the cost point is such that Microsoft's profit would be diminished as consumers are generally unwilling to bear the cost increment that this would entail. The incremental cost of finding additional bugs exceeds the total cost to all consumers of taking an alternative course of action such as installing HIDS (Host Intrusion Detection Software) and Host firewalls.

The loss for the consumer is lessened to a lower extent than the loss of the producer. With fraud loss limits of $50 in countries such as Australia for online transactions, banks in these locations have an incentive to minimise the loss to the consumer. Perversely, this can incentivise the consumer against adequately securing their system. If the consumer expects to lose a maximum of L_{iy} (which is set at $50 for credit card transaction fraud in Australia) for any given incident i where the total expected damage is defined as:

$$D_y = \sum_{i=1}^{n} L_{iy} \qquad D_x = \sum_{i=1}^{n} L_{ix} \tag{2}$$

The expected annual number of incidents per consumer n can be calculated as the total number of incidents that have occurred divided by the total number of consumers of a class (i.e. the total pool of credit card users).

$$E(n) = \frac{\#incidents}{\#consumers} \tag{3}$$

Setting C_{Ty} as the total cost to the consumer of implementing controls, if the expected total loss to the consumer $D_y < C_{Ty}$, it is doubtful that the consumer will pay for additional protection. For instance, if a high-end HIDS and anti-malware product costs $C_{Ty}= \$225$, and the consumer experiences $n=4$ incidents in a usual year, the expected damage $D_y = \sum_{i=1}^{n} L_{iy} = \$200.$ As $D_y < C_{Ty}$, it is not in the interest of the consumer to adequately protect their system. The user of a system that requires more security then the mean level of control provided by a vendor can implement increased security controls on their system, but this would either require that the consumer experience other measurable losses or that $D_y > C_{Ty}$ for this consumer.

Here we see that the anti-fraud efforts by banks and credit card companies create a negative incentive to consumers. The loss to the vendor L_{ix} currently averages $237 [1] for each lost set of credentials. The result is that it is in the interest of the financial company to provide the consumer with a compensating control. Holding the consumer liable if they had failed to use the enhanced controls over security would result in $D_y > C_{Ty}$ and hence an incentive for the consumer to protect their system.

Capital invested by the consumer in securing their system has a greater marginal effect than that of the producer in the case of an organisation such as Microsoft. A consumer can purchase HIDS and host firewall software for less than the cost that it would cost Microsoft to perfect their software through formal verification and hence remove more bugs.

The expected damage, $E(Damage)_i = P(x_{ai}).D_{Tot}$ or the expected damage is equal to the probability of a breach times the amount of damage suffered in a breach. This can be expressed as a function for each user or as a total cost function for all users, $E(Damage) = \sum_{i} \left(P(x_{ai}).D_{Tot} \right)$. Here we can clearly see that the total amount of damage is a function of not only the producer, but also the consumer. The optimal solution is to find a point that minimises the total costs. This is the expected damage as a loss function plus the costs of damage prevention of a compromise of other loss. The damage can also be expressed as a function of both the producer and consumer (user) costs,

$$C_T = Cost_{Tot} = \sum_i \left[P(x_{ai}) D(x_{ai}) \right] + C_v + \sum_i \left[C_u(i) \right] \tag{4}$$

The first order conditions are:

$$P'(x_{ai}) D(x_{ai}) + 1 = 0 \tag{5}$$

$$D'(x_{ai}) P(x_{ai}) + 1 = 0 \tag{6}$$

That is, the user should increase the expenditure on precaution (preventing a breach) until the last dollar spent on precaution by the user reduces the expected damage by $1. And the producer should increase the expenditure on reducing the possible damage in case of a breach until the last dollar spent on precaution by the producer reduces the expected damages by $1.

Clearly, the greater the likelihood of the user experiencing a breach, or the larger $P(x_{ai})$ is for the user, the greater the precaution that they should undertake. In the case of a producer who is a software vendor, they will (generally) sell their products to a wide range of users with varying levels of likelihood that each will experience a breach. That is, the software vendor is acting with imperfect information.

The optimal amount of precaution is the solutions to Equations (2) and (3) and is denoted by the expressions C_v^{Ω}, $C_u^{\Omega}(i)$ and where the total costs for all users is optimised at $\sum_i \left[C_u^{\Omega}(i) \right]$.

The marginal utility expenditure of security means that the value of security decreases the more we add. There is reason for this. If we spend more than the value of the organisations capital, it is simple to see that the producer will not survive long. It is more than this, we only need to reduce profitability for a producer to fail, not the capital.

The level of damages suffered by a user depends on both the pre-breach behaviour of the user and the vendor. The vendor is in a position where reputation impacts sales (demand) and hence the willingness to add layers of testing and additional controls (all of which increase the cost of the software). As the market for software varies in its elasticity [9] from the highly inelastic in small markets with few competitors (e.g. Electricity markets) to highly elastic (e.g. Operating Systems), the user has the ability to best determine their needs. The user may select customised software with warranties designed to reduce the levels of breach that can occur. This comes with an increased cost.

Software vendors normally do not face strict liability for the damage associated with a breach due to a software vulnerability [4, 7]. Although negligence rules for software vendors have been called for [7], this creates a sub-optimal outcome. The user can: (1) select different products with an expectation of increased security [2], (2) add external controls (through the introduction of external devices, create additional controls or use other software that enhances the ability of the primary product), and (3) increase monitoring for attacks that may be associated with the potentially vulnerable services such as by the use of IDS (Intrusion Detection System).

By limiting the scope of the user's responsibility, the user's incentive to protect their systems is also limited [4]. That is the user does not have the requisite incentive to take the optimal level of precautions. Most breaches are not related to zero day attacks [3]. Where patches have been created for known vulnerabilities that could lead to a breach, users will act in a manner (rational behaviour) that they expect to minimise their costs [10]. Whether risk seeking or risk adverse, the user aims to minimise the costs that they will experience. This leads to a wide range of behaviour with risk adverse users taking additional precautions and risk neutral users can accept their risk by minimising their upfront costs, which may lead to an increase in loss later.

In any event, the software vendor as the cause of a breach is not liable for any consequential damages. This places the appropriate incentives on the user to mitigate the risk. At the same time, the vendor has a reputational incentive to minimise the risk to their reputation. This was seen a number of years ago where the costs of bugs to the consumer from Microsoft was deemed as being exceedingly high. The vendor response was to change their coding practices and to significantly reduce the number of vulnerabilities in their released code.

A better game model for the software industry is the “Stag Hunt”. This was based on Jean Jacques Rousseau’s postulations of a co-operation strategy between two hunters [8]. These individuals can either jointly hunt a stag or individually hunt a rabbit. The largest payoff is assigned against the capture of a stag which provides a larger return than the hare. The hunting of a stag is more demanding and requires mutual cooperation. If either player hunts a stag alone, the chance of success is negligible and sub-optimal. Hunting stags is most beneficial for society in that this activity creates the optimal returns. The problem with this game is that it requires a lot of trust among the players.

		Software User	
		Create Secure Software	Add Features
Software Vendor	Create Secure Software	10, 10 A, W	1, 7 B, X
	Add Features	7, 1 C, Y	5, 5 D, Z

Fig. 1. Software Markets as a “Stag Hunt”

This game has two pure strategy equilibria in which both of the players prefer the lower risk equilibrium to the higher payoff equilibrium. The game is both Pareto optimal and Hicks optimal, but the sub-optimal and hence inefficient equilibrium poses a lower risk to either player. As the payoff variance over the other player's strategies is less than that of the optimal solution, it is more likely that this option will be selected. Another way of stating this is that the equilibrium is payoff-dominant while the other strategy is risk-dominant.

The strategy between the Software Vendor and the Software User is displayed in Fig 1. In this, the numerical representations represent the payoff figures for the specific case (the software market) and the generalized relations take the form:

$$\begin{aligned} A > C \geq D > B \\ W > X \geq Z > Y \end{aligned} \tag{7}$$

The outcomes are not definitive statements of what will be produced. In this game, the "Stag" is a desire to "Create Secure Software" and the "Hare" the fallback to adding more features. A desire is not a case of creating fewer bugs by itself, but rather a combination of adding controls and testing to software. Such an example would be the addition of the XP to Windows XP SP2 by Microsoft. Additional testing is effective to a point and more can be done than is occurring at present.

The payoffs for creating more secure software are great for both the vendor and the user, but the risk of a misaligned strategy leads to the sub-optimal equilibria. What is needed is a signaling process. A signal will allow the players to align to the more optimal strategy. It is not only in the user's interest to have more secure software, but also is in the interest of the vendor. Patching is expensive and the vendor can reasonably charge more for secure software.

As the ratio between the payoff for stag hunting and the payoff for hare hunting is reduced, the incentives to move towards stag hunting decreases. As a result, it becomes less likely that software security will be made into a primary goal of either party. As such, where the introduction of features and the "*new killer app*" occur more frequently, software security lags and it becomes more likely that a change from a stag hunting equilibrium to a hare hunting equilibrium will occur. It is hence less probable that an alteration of the players strategy from hare to stag.

Since neither player has an incentive to deviate, this probability distribution over the strategies is known as a correlated equilibrium of the game. Notably, the expected payoff for this equilibrium is 7(1/3) + 2(1/3) + 6(1/3) = 5 which is higher than the expected payoff of the mixed strategy Nash equilibrium.

3 Assessing Economic Value of Security

Being a relative function, not only does the profitability of an individual class (be that organization, group or nation) factor into the calculation of security risk, but the relation to a classes neighbors also needs to be measured.

The cost function is in the criminals favor without additional input from the consumer. There is no impetuous for the bank to move to a more secure (and also more costly) means of protecting consumers when the criminal can still gain to the consumers system. One part of the problem is the regulations that plague banking. The requirement to authenticate customers when calling for their privacy makes it simple for a criminal to pose as the bank and obtain the information. So even if a more secure means is selected, it is trivial to bypass many controls using social engineering and other less technical methods.

Whilst there are greater losses from consumer inaction then supplier inaction, the consumer's failure to secure their system and refrain from the use of systems at insecure locations all compound to make it more likely to have a loss through this means.

At all points of an assessment, we have to also take the time value of money into account. The value of capital is not set and fluctuates with time. To evaluate costs, we need to take both cost and the point at which the cost is expressed into account.

In order to compare any set of two or more alternatives, the financial characteristics of the alternatives must be compared on an equivalent basis. Two options are said to be equivalent when they have the same effect. Monetary values are termed as equivalent when they have the same exchange value. This can be defined as:

1. The comparative amount of each monetary sum,
2. The times of the occurrence of the sums can be aligned.
3. An interest rate can be used to compare differences in the time of payment.

The general equivalence function is defined as:

$$\text{PE, AE or FE} = f(F_i, i, n) \tag{8}$$

This equation holds for values of *t* between *0* and *n*. The equivalence equation uses:

$F_t =$ the rate of monetary flow at the end of time period *t*.
$i =$ the rate of interest for the time period.
$n =$ the number of discrete time periods.

The security and risk product lifecycle defines the function of the acquisition and utilisation phases. A system with a longer MTBF (Mean Time Between Failure) has a greater return on the initial investment. Similarly, larger upfront investments in security reduce the amount of capital available for investment. The financial present equivalent function [PE(i)] is defined as a value calculation that is related to the difference between the present equivalent capital value and the present equivalent costs for a given alternative at a given interest rate.

The present equivalent value at interest rate *i* over a total of *n* years is stated as:

$$\begin{aligned} PE(i) &= F_0(^{P/F,i,0}) + F_1(^{P/F,i,1}) + \ldots + F_n(^{P/F,i,n}) \\ &= \sum_{t=0}^{n} F_t(^{P/F,i,t}) \end{aligned} \tag{9}$$

The addition on measures that take externalities into account act as a signaling instrument that reduce information asymmetry and improve the overall risk position of both the consumer and the vendor. The development of a software risk derivative mechanism would be beneficial to security [5] through the provision of a signaling process to security and risk.

4 Conclusion

As we move security expenditure from a lower to higher value, the returns on that expenditure increases to a maxima and then decreases. The optimal point is where security expenditure and expected returns result in positive growth. In this paper we have rigorously assessed the security expenditure and their expected returns and conclude that the rational choice for selection of security controls is important. Before we invest our valuable resources into protecting the information assets it is vital to address concerns such as the importance of information or the resource being protected, the potential impact if the security is breached, the skills and resources of the attacker and the controls available to implement the security. The value on stack is not the capital, but rather expected return on capital. In any event, security expenditure fails where it costs more than it is expected to save [14]. This paper validates reasons why the cost of vendors in share price [13] and reputational losses exceed the perceived gains from technical reasons where the fix might break existing applications.

References

[1] Ben-Itzhak, Y.: Organised cybercrime and payment cards. Card Technology Today 21(2), 10–11 (2009)

[2] Devanbu, P.T., Stubblebine, S.: Software engineering for security: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering. ACM, Limerick (2002)

[3] DShield (2006-2010), http://www.dshield.org

[4] Hahn, R.W., Layne-Farrar, A.: The Law and Economics of Software Security, p. 283. Harv. J.L. & Pub., Pol'y (2007)

[5] Jaziar, R.: Understanding Hidden Information Security Threats: The Vulnerability Black Market. Paper presented at the 40th Annual Hawaii International Conference on System Sciences HICSS (2007)

[6] Peisert, S., Bishop, M.: How to Design Computer Security Experiments. In: WG 11.8 International Federation of Information Processing. Springer, Boston (2007)

[7] Scott, M.D.: Tort Liability for Vendors of Insecure Software: Has the Time Finally Come. Md. L. Rev. 67(425) (2007-2008)

[8] Skyrms, B.: The Stag Hunt and the Evolution of Social Structure. Cambridge University Press, Cambridge (2004)

[9] Stolpe, M.: Protection Against Software Piracy: A Study Of Technology Adoption For The Enforcement Of Intellectual Property Rights. Economics of Innovation and New Technology 9(1), 25–52 (2000)

[10] White, D.S.D.: Limiting Vulnerability Exposure through effective Patch Management: threat mitigation through vulnerability remediation. Master of Science Thesis, Department of Computer Science, Rhodes University (2006)

[11] Kolstad, C.D., Mathiesen, L.: Computing Cournot-Nash Equilibria. Operations Research 39, 739–748 (1991)

[12] Kurz, M., Hart, S.: Pareto-Optimal Nash Equilibria Are Competitive in a Repeated Economy. Journal of Economic Theory 28, 320–346 (1982)

[13] Arora, A., Telang, R.: Economics of Software Vulnerability Disclosure. IEEE Security and Privacy 3(1), 20–22 (2005)

[14] Bacon, D.F., Chen, Y., Parkes, D., Rao, M.: A market-based approach to software evolution. Paper presented at the Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications (2009)

[15] Cavusoglu, H., Cavusoglu, H., Zhang, J.: Economics of Security Patch Management. In: The Fifth Workshop on the Economics of Information Security, WEIS 2006 (2006)