Why Computer Security Fails – An Economic View*

Fu'ad W. F. Al Tabba'
The University of Auckland
Computer Science Department
ftabba_at_gmail_dot_com

Supervisor: Dr. Clark Thomborson

Abstract

Computer security is addressed from the economic point of view rather than the more traditional technical one. The reasons why security fails, such as the cost of security, incentive failures, Tragedy of the Commons and the lack of knowledge are investigated. Finally, some of the proposed (economics-based) solutions to security problems such as enforcing liabilities and government regulation are discussed.

Keywords

security economics, software liability, software regulation

1 Introduction

"Security Flaw Found in Windows", "Worm Blasts Across The Web", "Company's Servers Hacked and Sensitive Customer Data Stolen", such news headlines are becoming more and more common – computer security breaches are more prevalent. Yet with each and every passing day, we are increasingly relying on computers for important services, from banking and shopping to stock trading. In spite of that, it doesn't seem that the security of the systems we are using is getting any better – on the contrary, the number of threats, vulnerabilities and breaches is increasing every year [1].

The question that comes to mind is why. Why does not security seem to be improving? The technology is supposedly advancing at a tremendous speed, system enhancements and updates are being released all the time. This might indicate that the problem is not with the technology that we posses, but with how people are using this technology. Schneier has been arguing for quiet a long time that security essentially boils down to one key element – the human being [2]. His argument goes along the lines that the technology to have systems that for all

* October, 2005

practical purposes are secure enough exist. However, for some reason it seems that users of these systems do not use or abuse whatever security measures that can be applied.

In other words, it appears that people are not very motivated to do anything about security. So what is the greatest motivator when it comes to people and businesses? It is money. Judging from the state of computer systems' security, it would be safe to agree with Lampson that bad security is not costing businesses and individuals that much [3].

The economic side of computer security is being noticed though. Prominent security experts like Anderson have a big section of his webpage devoted to it [4], Schneier talks about it extensively in his books [2] and website [5]. Notwithstanding the work that has been done, this area of computer security still needs more research and hard work for it to mature.

In this paper, I try to look at the economics of computer security in more detail in an attempt to prove that security is primarily an economics problem. I will also examine the issues in the economics of security and see how relevant they actually are to the problem. Finally, I am going to go over the proposed solutions, see how effective these might be and where they could work or fail.

2 What is a Secure System?

Before I go into the discussion of the economics of security, what is meant by a having a secure system ought to be explained. The question of what has to be achieved in order to be able to designate a system as secure should be answered.

Both Lampson and Schneier have come up with different, yet overlapping, requirements for a secure system. These requirements vary depending on the system, who is using it and what it is being used for.

In Lampson's point of view, security is achieved by secrecy (confidentiality), integrity, availability and having accountability [3].

Secrecy means controlling who has access to whatever resources we are trying to protect, whether these resources are information or physical items. ***Integrity*** means preventing these

resources from being changed either accidentally or by parties who should not be allowed to. ***Availability*** is that only those who are intended to have access to the system should have prompt and uninterrupted access to it. Finally, ***accountability*** is keeping track of what is going on in the system; this is done to enable auditing, and to make the detection and reaction to the problems easier.

For Schneier, the pillars of security are authentication, privacy, anonymity, integrity and audit [2].

Authentication for Schneier is essentially what Lampson refers to as availability (actually coupled with both confidentiality and integrity) – only the party that should access the system should be able to do that and no one else. ***Privacy*** and ***anonymity*** would have their corollary in Lampson's secrecy. ***Integrity*** is obviously the same for both Schneier and Lampson, and ***audit*** is the method Schneier uses to achieve accountability.

The boundaries for each of these requirements are not clear, and the definitions are not set in stone. Moreover, having all of these is not necessary for a system to be secure; a subset could suffice depending on the system[1].

3 Economics of Computer Security

Security is a game that involves two sides – the *attack* and the *defence*. Each of these sides obviously has different objectives, but they do have one thing in common – the effect of economics on their respective strategies. I will start with discussing the economic factors that affect the defence then move on to the factors affecting the offence.

3.1 Economics of Defence

3.1.1 Cost of Security

Good security is expensive – it costs a lot of money to implement as well as a lot of inconvenience and frustration to the end user not to mention reduced productivity, all of which eventually translate to even more money [2]. While on the other hand, the cost of

[1] For example, a news article published on the web for the public to see has to maintain its integrity as well as its availability – no need for confidentiality. On the other hand, the newspaper's sources have to have all three requirements.

actually detecting and responding[2] to problems and security breaches (i.e. accountability) is not as high [3].

For example, implementing a good password security policy on a corporation's network will reduce the chances of being attacked through a cracked password [6]. Yet the more complicated the password policy, the greater the chance that users will forget their passwords. According to one study, each forgotten password costs a business around £15[3] [7]. Therefore, some businesses could decide that the increased security achieved by having stronger password policies is not worth the cost.

I think Lampson says it best, "When the risk is less than the cost of recovering, it's better to accept it as a cost of doing business, or a cost of daily living, than to pay for better security." [3]

I believe that this is about to change – as a matter of fact I think that it is already changing. The number of people using the internet for sensitive applications like managing their finances is steadily increasing [8]. Companies are relying more and more on the internet for important services like telephony (Voice Over IP) [9] – which has traditionally been a separate entity. This reliance on computers and the internet coupled with the fact that security vulnerabilities are increasing [1] will make it more economically feasible to invest in good security. Thus, businesses will be forced to have a more holistic approach to security rather than focus on just one aspect (e.g. accountability).

3.1.2 Incentive Failures

On one hand, end-users want bug-free and secure software; while on the other hand, commercial software developers like Microsoft have one main goal in mind – it would be great if that goal were to develop bug-free and secure software as well, but unfortunately their main target is moneymaking. The faster they ship their product, the earlier they will start earning money, while the more testing they might spend on their products the later their

[2] To use Schneier's "Defense in Depth" taxonomy [2]. There he argues that *detection* is much more important than *prevention* – which could also explain why businesses prefer *detection* as a security measure. Detection obviously will not work without proper audit and accountability.
[3] Roughly NZ $45 at the time of the publication of the referenced article. Exchange rate history from http://www.oanda.com/convert/fxhistory .

product will ship and less money will be made. This is an example of what Anderson refers to as an incentive failure [10].

The problem is that often the incentives of the people who are responsible for keeping our systems secure are in direct conflict with security itself. A CIO[4] might believe that a certain operating system (such as Linux) might be more secure than some commercial operating systems (such as Microsoft Windows). If the company has been using Windows all along, and if all the other companies in the region are using Windows as well, it is not in the CIO's best interest to switch. For if the company switches and they do undergo an attack, it will be the CIO who will suffer the consequences. While if the company sticks to the status quo and does not switch, even if they do suffer from a security failure, the CIO will not be blamed since "everyone else is doing it".

Let us consider the popularity of firewalls as another example. Even though firewalls are arguably ineffective [11], their popularity could be explained as an incentive failure. Corporate auditors nowadays require firewalls [2], possibly because they are rather simple products that are relatively easy to install and setup and do not interfere much with the usability of the system they're installed on [3]. Therefore, the CIO's incentive of having a firewall installed on the company's system is actually not related to the safety of the network, but is to please the auditors and the board of directors.

Thus, an incentive failure could be considered a failure of accountability, since what it does is fail to hold the right person accountable for the right thing. The saying goes, "If there is a will, there is a way." Therefore, to solve the problem of incentive failures, we have to somehow get the party responsible for the security of the system to actually be *willing* to do something about it.

3.1.3 Tragedy of the Commons

The Tragedy of the Commons is a metaphor the ecologist Garrett Hardin came up with in order to explain how at times the interest of the individual might conflict with the common good [12]. Hardin based his metaphor on the English Commons, a shared plot of grassland where the livestock graze. The commons is used by all the farmers in a village, whenever a

[4] Chief Information Officer: The manager normally responsible for information technology within a company, which normally includes information security as well.

farmer adds to the livestock grazing there he[5] will gain almost full benefit, while the rest of the farmers will only suffer marginally from each added sheep. Eventually, the soil of the commons will be depleted, it will not be usable anymore and all will suffer[6].

What this serves to illustrate is that when a public good is shared by a group of people, and the cost of the use of that public good is not entirely borne by the person using it, people will tend to abuse it until it is of no value to anyone.

How does this relate to computer security? Internet security is a common; the Internet is obviously used by everyone and keeping it working properly and securely benefits all [2]. The price we pay to actually connect to the internet is quite low[7] in comparison to the value we gain by using the internet, ranging from the amount of information accessible there to the services that are provided through the internet such as Voice Over IP. Thus, it is clear that the cost of using this public good (the internet) and keeping it safe is not entirely borne by the party using it, but by the community as a whole.

As an example, let us consider distributed denial of service attacks[8]. Such attacks succeed because the people whose unprotected computers are used as launching pads for these attacks are not doing their part in keeping the Commons safe, like installing an anti-virus program for example. They have gained all the benefits they desire from using the Commons and do not have much to gain by securing their part, since they are not liable for what is done [14].

As more people connect to the internet, a way to motivate the users of this Commons to do their part has to be applied in order to keep the internet useful for everyone. In other words, even though attacks that are a result of the Tragedy of the Commons are mainly attacks on availability (in the case of DDOS), integrity (if the compromised computers start spreading viruses), or even confidentiality (if the computer sends out personal information like credit card numbers); it is clearly a problem of accountability – for each person should be held responsible for what goes on in their systems.

---

[5] Most, if not all farmers of that era were men.

[6] Other modern problems related to the Tragedy of the Commons are pollution, spam and over-fishing. Wikipedia has a good introductory article http://en.wikipedia.org/wiki/Tragedy_of_the_Commons .

[7] The cost of getting a broadband connection in New Zealand from a provider such as Xtra (http://jetstream.xtra.co.nz/) is about NZ $40 at the time this paper was written.

[8] "*A distributed denial-of-service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.*" Quoted from searchSecurity.com [13]

3.1.4 Knowledge is Power

The truth of the matter is that end-users do not know much about the products they are using. The average person is not qualified to judge the quality and security of a system[9]. What is even worse, those qualified enough to evaluate often have a conflict of interest, since they either work for the company making these products or simply are not affected by their breakdown, thus suffer from the problem of incentive failure.

Gresham's law states, "Bad money drives good money out of circulation" [15]. Anderson opines that the same applies to the security product market if there is not sufficient knowledge on the quality of these products. In other words, bad security tends to drive out good security [10].

Anderson reaches this conclusion by assuming that the product vendors actually know which products are secure enough and which are not, and that good security costs more than bad security[10]. The buyer, not knowing which product is which will assume that the probability of getting a good security product is the same as getting a bad one, and the market value for both products will end up being somewhere in between the actual value of the two.

Since the market value of the products is now less than the actual value of the good security products, the vendors have no incentive to sell the good ones at a discount so only the bad ones will be offered for sale. Eventually, the buyers will notice that the security products they ended up buying are not as good as they ought to be, so the price will drop to that of the bad security products rather quickly. Thus, we will end up with a market that offers only the bad products where all the good ones will go into hiding.

This issue is one that involves ethics as well. Here we have an interplay of the customers' right to know about the product they are purchasing, and the vendors' right to privacy in terms of protecting their intellectual property as well as the right to fair compensation for the work they are providing [16].

[9] Even experts have a hard time doing that!
[10] Even though these assumptions might seem reasonable enough, I think that they might not be valid all the time, but still I do not believe that their validity will detract from the essence of Anderson's argument.

Customer awareness and a proper mechanism for reviewing products are the obvious solutions to this problem. This of course has to be done by a party that has the right motivation in order to avoid any incentive failures.

3.2 Economics of Offence

3.2.1 The Usual Suspects

Whenever a security system is designed, one of the most important questions that should be asked is what are the risks to the system [17]. In particular we are concerned with the attackers [18] and the economic forces that drive them.

Schneier categorizes the adversaries according to their objectives, access, resources, expertise, and risk [2]. If the objective is financial gain then economics clearly is an important factor. It is true that not all attackers are motivated by pure financial gain (like terrorists[11] for example), but money could facilitate or limit their attacks. Thus, economics will still be a significant consideration.

Even if financial gain is not the goal; access, resources, expertise and risk are all affected by finance. For if the attackers do not have the required access, they could always bribe their way in. Resources could be bought for the right price and proper training could be obtained as well. Finding people willing to risk almost anything is also achievable with the right amount of money.

Schneier has a comprehensive list of potential adversaries [2]. Of the ones he mentions, it is the lone criminals, malicious insiders, industrial espionage and organized crime that have money as the primary motive. While hackers, the press, terrorists, the police and national intelligence organizations are usually not motivated by financial gain[12].

Of these potential attackers, the ones where the economic factor is the weakest would be the government-related organizations such as the police and the intelligence. When national

[11] I use Schneier's definition of terrorist here without an attempt to make any moral judgements, which is basically a catchall phrase for any ideologically motivated group or person.
[12] This is not always true though. For example, a malicious insider might be motivated by revenge while a terrorist organization might be looking for ways to finance their next operation. Moreover, sometimes the distinction between these different attackers could get blurry; e.g. a hacker working for a national intelligence agency.

security and safety of the people is concerned, money should not be an issue at all. In general, attacks by government organizations that exploit computer security flaws are not very common, or at least are not as common as the other kinds of attacks, thus will be of no concern to us as far as the scope of this paper goes[13].

3.2.2 Economics of the Offence/Defence Arms Race

Most battles between sides are essentially an arms race, and the fight on the digital frontier is no different. Whenever there is a technological advancement, one side tends to benefit from it more than the other. For example, the invention of the machine gun gave the advantage to the side of the defence in World War I, while the advent of the tank in World War II gave the edge back to the offence [10].

Anderson has observed that this is also true on the digital frontier. The current state of the digital world tends to favour the attackers. The reason is that from the defence's point of view, there is a huge area to cover, from securing the operating systems used, to securing the applications as well as the network. While all the attacker needs to do is find one weak point and exploit it [2].

What makes this worse is that software is complex, and it is getting even more and more complex. Windows 3.1 is estimated to have 3 million lines of code, for Windows 98 the figure is around 18 million, while Windows 2000 is estimated to have between 35-60 million lines of code. If we go by the assumption that there are 5-15 bugs in every 1000 lines of code [2], this would result in a modest estimate of 150,000 bugs in Windows 2000.

To use the same line of reasoning as Anderson [10], we will assume that the number of security-critical bugs that are unique is only one percent of all bugs. This leaves us with 1,500 critical bugs in Windows 2000. Therefore, if all the offence needs to do is just exploit one bug, then the defence would need to work orders of magnitude as hard as the offence to stay on the same level.

In a study that proves even more that the offence has the edge, Rescorla did an in-depth investigation on the economics of finding security holes and fixing them. In his paper he

[13] With apologies to George Orwell.

reached the conclusion that there is no empirical data to support that finding security flaws and patching them offers any quality improvement or even makes economic sense[14] [19].

What does this mean? I think that this proves even more that just working on the technological side of the equation will not be of much help – but that a more economic approach might be the key.

3.3 Opposing Views

The idea that security is mainly an economics problem, like any other idea, has its strengths and weaknesses.

One argument against it would be a recent study which shows that flaw disclosure hurts software makers' stock price [20]. The release of 146 vulnerabilities was analyzed and it was shown that the stock price of a company drops on average by 0.63% compared to the NASDAQ[15] on the day the flaw is announced. This might prove that the market already factors in the security problems and thus there is no need to address the issue of economics any further.

It could be said that all this shows is that it is the disclosure of the flaws and not the actual flaws themselves that hurt the stock price. Moreover, as Schneier noted, it could just be that it is the bad news about the company that is affecting the stock price, not the actual vulnerability – thus the effect of the price drop could be short-term. The question whether there are any long-term effects has not been answered [21].

Another argument against the economics of security would be the recent controversy surrounding the critical security flaws found in Mozilla Firefox. Symantec reported that 25 vulnerabilities were disclosed for Mozilla browsers while only 13 were disclosed for Microsoft Internet Explorer this year so far [22]. Since Firefox is open-source and is distributed for free, it could be argued that economic factors should not come into play which should make it a more secure platform than Microsoft Internet Explorer.

[14] I have to admit that when I first read the abstract of this paper I thought that the whole concept was counter-intuitive. However, having read the paper, I feel inclined to agree with Rescorla's conclusions.
[15] The NASDAQ is mainly composed of technology-related stocks, that is why it was used as a reference.

In my opinion, the flaw in the above reasoning lies in the assumption that Firefox, being open-source, is not affected by economic factors[16]. It is true that open-source software is not sold for profit, but as can be seen from the business models adopted by companies such as Red Hat, profit can be made by means other than actual sale of software. Moreover, the foundation of the Mozilla Corporation in August 2005 shows that Mozilla is a company that is financially aware even if its prime goal is not to make money [23].

4 Solutions

4.1 Enforcing and Transferring Liabilities

If I buy a car, and it turns out that the tires are faulty and may cause accidents, then the manufacturer is obliged by law to recall them. This is guaranteed since the manufacturer is liable for any problems that may arise from using faulty tires [24].

It could be said that that is an extreme example, since there is a potential of loss in human life. To use a less severe example; when Apple launched its latest MP3 player, the iPod Nano, it was discovered that there was a flaw in the screen of some of the iPods shipped that results in the cracking of the screen. Apple acknowledged the flaw and said it would replace the screen of all affected units for free [25].

On the other hand, if a company relies on a certain database management system for all its finances, and the system crashes causing all the data to be gone forever with financial losses to the company estimated to be in the range of millions of dollars, there is not much that can be done about it. The software vendor responsible for the system is just not liable; they do not follow the same customer protection laws as other product manufacturers.

The way software companies have avoided being under the same laws as other industries is that it is not an actual product that is sold, but a license to use the product (i.e. software). This license comes with a long list of various disclaimers that covers almost everything that can go wrong, practically relieving the vendor of all responsibility [26].

[16] Whether the figures Symantec reported were accurate, and whether the flaws in Firefox are actually severe is irrelevant to this point. Just for the sake of argument, I am assuming that Symantec's report proves that Firefox is less secure than Microsoft Internet Explorer – even though I personally believe the opposite to be true.

Software companies should be expected to deliver good software and security personnel expected to make the right decisions, or be held liable for any damages incurred. This way they would have an actual incentive to deliver their best. The cost of the potential lawsuits against them would be a big factor to consider, thus the problem of incentive failure would be solved.

According to Schneier, having companies liable for the security of their products would automatically lead to the emergence of some sort of mechanism for the transfer of these liabilities. The transfer of liabilities is what is also known as insurance. Companies like insurance because it turns variable-cost risks into fixed-cost expenses which can be more easily controlled and fitted into their budgets [2].

When insurance companies move into the digital security arena, they would move the security field in new directions. It is reasonable to assume that insurance premiums would cost a company with good security practices less than it would cost a company with bad ones. Distinguishing between good security and bad security is difficult. In this case, since the insurance companies have the proper financial incentive to get it right, it will be possible [2].

Insurance companies, I believe, would end up covering all three aspects of a Lampson secure system. They will make sure that the system conforms to the proper specifications and will set up the proper security policies for it. They will also oversee the implementation of these policies to verify that it is done correctly. Finally, even though there will be no assurance that the system will not fail, there will be an assurance that if it does there will be compensation.

For instance, most companies are fitted with fire detection and extinguisher systems. This is not because these companies are safety-conscious when it comes to fire hazards, but because insurance companies demand them. Either the insurance premiums will be high or there would be no insurance without such systems [27]. Insurance firms set the fire-security policies, inspect the premises to make sure these policies have been properly implemented. In case of failure, they compensate the company for their losses.

Even Anderson acknowledges the importance of insurance for the security field. From his point of view, the ability to insure a system guarantees that the system is going to be trustworthy. This is not because there will be no security bugs or flaws within the system; but the fact that the system is insured guarantees that if it breaks, you will not loose an

unpredictable amount of money. In other words, "A trusted component or system is one which you can insure." [28]

Imposing liabilities and transferring them via insurance will solve the problem of the Tragedy of the Commons since all users of an insured Commons will pay their part through the insurance premium. Those who abuse the Commons will end up having to pay higher premiums. Moreover, the lack of knowledge on the user's side will not be an issue anymore since the pricing of the different insurance plans will be all the knowledge the user needs to secure their systems.

In practice, this will not be that easy. The problem in imposing liability on software vendors is that software is much more complex than other kinds of products. Even the general attitude in the American[17] law arena is that software is too complex and it should be expected that there would be bugs and defects in it [26].

However, I do believe that this would not deter insurance companies from taking on the risk of software security. This might though make it a bit more complicated to estimate the risk that might be incurred and thus come up with the proper premiums. The premiums might start off being too high or too low, but I think that they would finally settle into a reasonable equilibrium.

4.2 Government Regulation

The FDA[18] in the United States as well as the Ministry of Health in New Zealand have done an arguably good job of making sure that the food, medicine and other things that are important to consumers are safe. The Ministry of Transport maintains high standards on the quality of the vehicles allowed on the roads for (also arguably) the safety of the drivers, passengers and pedestrians. Why not have the same kind of regulation for software, especially now since software could control potentially critical systems?

[17] I use the American law arena as an example since the U.S. is home to the biggest software development houses such as Microsoft, IBM and Oracle.
[18] Food and Drug Administration, it is the governmental agency in the United States responsible for regulating food, dietary supplements and drugs among other things. (http://www.fda.gov)

Leveson has actually argued for just that. She drew a parallel between the introduction of computers and how it revolutionized our lives now and the introduction of steam engines and how it revolutionized the world then.

With the advent of steam engines, more power was needed than available in the traditional ones so high-pressure engines were used. The problem with high-pressure steam engines was the safety risk they posed – they were more prone to explosions. Unfortunately, accidents did happen, and in the period from 1816 to 1848, almost 5,000 people were killed and injured in the U.S. alone, with financial losses of over US $3,000,000[19]. It was not until Congress passed a law in 1852, which was on of the first regulatory laws in the U.S., that the number of explosions fell dramatically. By 1905, there were only[20] 383 deaths resulting from such explosions [29].

The above seems like a good argument for software regulation. However, the problems with regulating the software industry are many. For starters, to be able to regulate something, some system for standardizing and evaluating the security practices of the industry has to be developed.

There has been a lot of work in this area. Firesmith's paper on Specifying Reusable Security Requirements is one example [18]. On a more governmental level, the U.S. Department of Defense published the Trusted Computer System Evaluation Criteria, better known as the Orange Book[21] in 1985. It was meant to be a standard for security requirements to be used by the government and to enable computer manufacturers to measure the security of their systems.

Not to be outdone by the Americans, the European Union developed the Information Technology Security Evaluation Criteria, known as the ITSEC in 1995. Finally, the ISO came up with the Common Criteria (standard 15408), which is an international standard recognized by many countries including New Zealand [2].

[19] About US $70,000,000 or NZ $100,000,000 after accounting for inflation at the time of the writing of this paper. Inflation calculation data from http://oregonstate.edu/dept/pol_sci/fac/sahr/sahr.htm and currency exchange rate from http://www.xe.com/ucc/ .
[20] The loss of human life is always unfortunate; I use the word "only" in comparison with the previous figures, not to trivialize the loss of life in any way.
[21] Because it had an orange cover [2].

In my opinion, there are many problems with such methods if the intended purpose behind them is to guarantee security. First, the fact that a certain system conforms to a Security Template or passes a government standard means only that – that it had all the required documentation and all the mandatory items in a certain checklist. However, it does not mean that the system is secure. What is even worse, such a certification might give a false sense of security thus prove to be detrimental to security rather than enhance it.

This is what happened to the Orange Book and there is no reason to believe that it would not happen to other proposed standards [2]. After all, even Microsoft Windows 2000 is ISO 15408 certified [30], in spite of the fact that it has at least 134 published security vulnerabilities [19].

Other than the difficulties of having a security standard, I think that government regulation will not work because the very nature of software is complex, and technology changes exceedingly fast. Governments are not good at adapting to quick changes. Therefore, by the time a law passes, the landscape that that particular law was supposed to address would have changed so much that it is not applicable anymore. On the contrary, such a law might even prove to be more harmful than useful.

Another issue when it comes to software regulation; it could be argued that software, being ideas, is protected under the freedom of speech principle [31]. Thus, any attempt to regulate software could be likened to censorship.

While I believe that enforcing liabilities will cover all three aspects of securing a system according to Lampson [3] – regulation will only cover one. Regulation will impose policies on the specification of systems. These policies, since they are not based on market value and experience, might not be the best ones to impose. There might be some sort of mechanism of monitoring the implementation of the system, but I really doubt that the government will expend all the required resources to keep on monitoring the compliance with their security standards on a regular basis. Finally, regulation will not give an assurance that the system will not fail, or any kind of compensation if it does.

I do not believe that regulation will properly address any of the economics issues either. The cost of good security will increase, but that is because going through the trouble of dealing with official bureaucracy is expensive and not because the actual security will get any better.

While regulation might give vendors an incentive to produce secure software, their real incentive would be just to get some certification. As for the problem of the Tragedy of the Commons, government regulation was not very successful in dealing with other economic problems related to the Tragedy, such as pollution and over-fishing. Therefore, there is no reason to believe that it will do a better job when it comes to solving the Tragedy for security. Finally, government regulation will not solve the problem of the lack of knowledge. As mentioned earlier, knowing that a product has a certain security certification will not mean that much.

However, I do believe that some regulation is necessary. I think that there should be as little regulation as possible – just enough to enforce liabilities. If there were no regulations at all, then the enforcement of liabilities would not be possible. To go back to freedom of speech comparison, people should be able to say whatever they want, but that freedom comes with the responsibility for what is said.

5 Conclusion and Further Work

The other day I went to a restaurant for dinner. When the waiter gave me the check, I just handed him my credit card. The communication between the waiter and me was not encrypted. Other than the waiter's nametag, I did not use any sophisticated methods of authenticating who he was. When I got the receipt, I quickly glanced at it to make sure that it is the right amount then I discarded it. Yet I was confident that the transaction was completed without any problems.

The security in this transaction was not due to any technologies used, but came from the knowledge that the credit card company has taken the liability upon itself by allowing me to repudiate any charge before paying the bill [2].

This anecdote shows what I have been trying to conclude in this paper. Security is mainly driven by economics; technology just facilitates it and makes it easier. I think that the best way to go around solving the economic imbalances is to impose liability, which in itself might require a bit of regulation, and everything else will follow from there.

This paper barely begins to explore the economics of security. Some of the work that could be done in this field is on:

- The laws and regulations surrounding imposing liabilities on software makers
- Calculating the risk of undertaking software liabilities for the estimation of insurance premiums
- The interplay between the security and economics of open source software
- The long term effects, if any, of bug and vulnerability announcements on a company's stock price

Talking about the importance of economics in all aspects of our lives, a popular quote by Ludwig von Mises[22] says it all. "The body of economic knowledge is an essential element in the structure of human civilization; it is the foundation upon which modern industrialism and all the moral, intellectual, technological, and therapeutical achievements of the last centuries have been built." [33]

Acknowledgments

I would like to thank the following (alphabetically) for their help in proofreading this paper: Kilian Foerster, Samar Hindawi, Stefan Johansson, Jordan Pousse and Wael Tabba. Their feedback was really helpful and appreciated.

[22] An Austrian economist who was one of the greatest economists of the 20th century, and the dean of the Austrian School of economics for almost four decades [32].

References

[1] Symantec, "Symantec Internet Security Threat Report Highlights Rise In Threats To Confidential Information," 2005.
[March 21, 2005. Available from http://www.symantec.com/press/2005/n050321.html]

[2] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*. New York; Chichester: Wiley, 2004.
http://www.schneier.com/book-sandl.html

[3] B. W. Lampson, "Computer security in the real world," *Computer*, vol. 37, pp. 37-46, 2004.
http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=1306384

[4] R. J. Anderson, "Economics and Security Resource Page," *University of Cambridge Computer Laboratory*, 2005.
[August 25, 2005. Available from http://www.cl.cam.ac.uk/~rja14/econsec.html]

[5] B. Schneier, "Bruce Schneier's Website - Economics Search," 2005.
[October 1, 2005. Available from http://www.schneier.com/cgi-bin/search/search.pl?Terms=economics&Realm=whole+site]

[6] A. Cliff, "Password Crackers - Ensuring the Security of Your Password," *SecurityFocus*, 2001.
[February 19, 2001. Available from http://www.securityfocus.com/infocus/1192]

[7] G. Hayday, "Counting the cost of forgotten passwords," *ZDNet UK*, 2003.
[January 14, 2003. Available from http://news.zdnet.co.uk/business/employment/0,39020648,2128691,00.htm]

[8] L. Enos, "More Europeans Relying on Internet Banking Sites," *CRM Buyer*, 2001.
[May 7, 2001. Available from http://www.crmbuyer.com/story/11763.html]

[9] ZDNet, "30% of US companies plan to give VOIP a try," *ZDNet Research*, 2005.
[February 27, 2005. Available from http://blogs.zdnet.com/ITFacts/?p=7192]

[10] R. Anderson, "Why Information Security is Hard - An Economic Perspective," in *Proceedings of the 17th Annual Computer Security Applications Conference*: IEEE Computer Society, 2001.
http://www.cl.cam.ac.uk/ftp/users/rja14/econ.pdf

[11] A. Singer, "Security without Firewalls," seminar at the University of Auckland, 2005.
http://www.cs.auckland.ac.nz/compsci725s2c/lectures/AbeSanger.txt

[12] G. Hardin, "The Tragedy of the Commons," *Science*, vol. 162, pp. 1243-1248, 1968.
http://www.sciencemag.org/cgi/content/full/162/3859/1243

[13] searchSecurity.com, "distributed denial-of-service attack," *TechTarget*, 2004.
[May 21, 2004. Available from http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci557336,00.html]

[14] H. R. Varian, "Managing Online Security Risks," *The New York Times On The Web*, 2000.
[June 1, 2000. Available from *http://www.nytimes.com/library/financial/columns/060100econ-scene.html*]

[15] G. Selgin, "Gresham's Law," *EH.Net Encyclopedia*, 2003.
[June 10, 2003. Available from *http://eh.net/encyclopedia/?article=selgin.gresham.law*]

[16] C. P. Pfleeger, *Security in computing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1997.
http://www.phptr.com/title/0130355488

[17] B. Schneier, *Beyond fear: thinking sensibly about security in an uncertain world*. New York: Copernicus Books, 2003.
http://www.schneier.com/book-beyondfear.html

[18] D. Firesmith, "Specifying Reusable Security Requirements," *Journal of Object Technology*, vol. 3, pp. 61-75, 2004.
http://www.jot.fm/issues/issue_2004_01/column6

[19] E. Rescorla, "Is finding security holes a good idea?" *Security & Privacy Magazine, IEEE*, vol. 3, pp. 14-19, 2005.
http://www.rtfm.com/bugrate.html
http://www.rtfm.com/bugrate.pdf

[20] R. Telang and S. Wattal, "Impact of Software Vulnerability Announcements on the Market Value of Software Vendors – an Empirical Investigation," presented at Fourth Workshop on the Economics of Information Security, Kennedy School of Government, Harvard University, 2005.
http://infosecon.net/workshop/pdf/telang_wattal.pdf

[21] R. Lemos, "Study: Flaw disclosure hurts software maker's stock," *SecurityFocus*, 2005.
[June 6, 2005. Available from *http://www.securityfocus.com/news/11197*]

[22] J. Leyden, "Firefox and Mac security sanctuaries 'under attack'," *The Register*, 2005.
[September 19, 2005. Available from *http://www.theregister.co.uk/2005/09/19/symantec_threat_report/*]

[23] Mozilla, "Mozilla Foundation Forms New Organization to Further the Creation of Free, Open Source Internet Software, Including the Award-Winning Mozilla Firefox Browser," *Mozilla Corporation*, 2005.
[August 3, 2005. Available from *http://www.mozilla.org/press/mozilla-2005-08-03.html*]

[24] CNNfn, "Firestone tires recalled," *CNN Money*, 2000.
[August 9, 2000. Available from *http://money.cnn.com/2000/08/09/news/firestone_recall/*]

[25] Slashdot, "Apple to Replace Faulty Nano Screen," *Slashdot*, 2005.

[September 29, 2005. Available from http://apple.slashdot.org/article.pl?sid=05/09/29/1233254]

[26] M. A. Cusumano, "Who is liable for bugs and security flaws in software?" *Commun. ACM*, vol. 47, pp. 25-27, 2004. http://doi.acm.org/10.1145/971617.971637

[27] B. Allen, "Insurance Savings Tips & Techniques," 2005. [October 6, 2005. Available from http://www.eqgroup.com/afsavings1.htm]

[28] R. J. Anderson, "Liability and Computer Security: Nine Principles," in *Proceedings of the Third European Symposium on Research in Computer Security*: Springer-Verlag, 1994. http://www.cl.cam.ac.uk/ftp/users/rja14/liability.pdf

[29] N. G. Leveson, "High-Pressure Steam Engines and Computer Software," *Computer*, vol. 27, pp. 65-73, 1994. http://www.safeware-eng.com/index.php/publications/HiPreStEn http://sunnyday.mit.edu/steam.pdf

[30] NISCC, "NISCC Technical Note 02/03: Understanding Common Criteria Evaluation," *NISCC*, 2003. [January 21, 2003. Available from http://www.niscc.gov.uk/niscc/docs/re-20030121-00722.pdf?lang=en]

[31] P. Salin, "Freedom of Speech in Software," 1991. [July 15, 1991. Available from http://www.philsalin.com/patents.html]

[32] J. G. Hülsmann, "Ludwig von Mises," *American National Biography Online*, 2003. [August 2003. Available from http://www.anb.org/articles/14/14-01132.html]

[33] L. Von Mises, *Human action: a treatise on economics*, 4th rev. ed. Irvington-on-Hudson, N.Y.: Foundation for Economic Education, 1996. http://www.mises.org/humanaction.asp