A PRELIMINARY MODEL OF THE VULNERABILITY BLACK MARKET

Jaziar Radianti (jaziar.radianti@hia.no)
Jose. J. Gonzalez (jose.j.gonzalez@hia.no)

Research Cell "Security and Quality in Organizations",
Faculty of Engineering and Science, Agder University College, Serviceboks 509
NO-4898 Grimstad, Norway

Submitted to the 25th International System Dynamics Conference
Boston, USA
29 July-2 August 2007

Abstract

The emergence of vulnerability black markets enhances the opportunities for malicious actors to launch exploits toward computer networks, to commit cyber-crime and to perform other unlawful activities. Asymmetric information, inadequate software testing, lack of incentive to improve quality of the software are presumed to be the most important grounds of the software vulnerability problems. This work is a preliminary model to build a structure that may explain the factors influencing the emergence of vulnerability black market and to simulate undesired consequences from desired effect to eliminate software vulnerabilities. The purpose of the research is to provide better understanding for information security community about the dynamic features of the software vulnerability and the vulnerability black market problems.

Key words: *Economics of Information Security, Software Vulnerability, Vulnerability Black Market, System Dynamics*

Glossary

Bugtraq	:	An electronic mailing list dedicated to issues about computer security. On-topic discussions are new discussions about vulnerabilities, methods of exploitation, and how to fix them. It is a high-volume mailing list, and almost all new vulnerabilities are discussed there.
Black hat hacker	:	"Hacker" in the sense of a "black-hat hacker" – a person who is able to exploit a system or gain unauthorized access through skill and tactics, and not to refer white-hat hacker.
CERT	:	The CERT® Program (Computer Emergency Response Team) is part of the Software Engineering Institute (SEI), a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. The center is established to coordinate communication among experts during security emergencies and to help prevent future incidents. This center was named the CERT Coordination Center (CERT/CC).
Cyber crime	:	Crime encompasses any criminal act dealing with computer and networks (called hacking). Additionally, cyber crime also includes traditional crimes conducted through the Internet. For example; hate crimes, telemarketing and Internet fraud, identity theft, and credit card account thefts are considered to be cyber crimes when the illegal activities are committed through the use of a computer and the Internet.
Net crime	:	Criminals or other malicious activity utilising or directed toward the internet and/ or information technology application
Exploit	:	A piece of software that take advantage of a bug or vulnerability, leading to privilege escalation or denial of service on a computer system.
Vulnerability	:	Weaknesses in information technology (IT) products as supplied by the vendor(s); weaknesses in the ways organizations manage and use the technology
Unknown Vulnerability	:	Vulnerability that nobody has heard of
Script Kiddies	:	An inexperienced malicious cracker who uses programs developed by others to attack computer systems, and websites. It is generally assumed that script kiddies are kids who lack the ability to write sophisticated hacking programs on their own
White hat hacker	:	The term *white hat hacker* is also often used to describe those who attempt to break into systems or networks in order to help the owners of the system by making them aware of security flaws, or to perform some other altruistic activity
Vulnerability Researcher	:	There are several terms in the literature to identify the people searching for the vulnerabilities, such as: "bug bounty hunters" (Greenemeier 2006), "flaw researchers", "vulnerability researchers" (Sutton and Nagle 2006), "testers" (Schechter 2002; Ozment 2004).
Malicious agents	:	Other actors having malicious motives to compromise security, but without skill for doing so.
Black market	:	An illicit trade in goods or commodities in violation of official regulations
Zero day Exploit	:	A zero day exploit is a computer vulnerability that is being actively practiced before knowledge of the exploit becomes public information
Patch	:	A piece of code that is added to fix vulnerability
Virus	:	Viruses are programs that require some action on the part of the user, such as opening an email attachment, before they spread. Users are often enticed to open email attachments, sometimes because of an intriguing or legitimate-sounding subject line and sometimes, when address books have been compromised, because the email appears to be from someone the user knows
Malware	:	Is any computer program that harms the computer running it. Typically, malware is installed without the user's knowledge or consent. Different type of malware include spyware, trojan horses, rootkits, keyloggers, viruses and worms
Spyware	:	Is any computer program that reports usage patterns or specifics of the computer which is installed to a third party. While the software itself doesn't directly harm the computer or the user, the information gained from such attacks is often used in spam advertising, credit card, fraud and identity theft.
Worm	:	Worms are programs that spread with no human intervention after they are started
Spam	:	Process of sending unsolicited emails to large numbers of email addresses belonging to individuals with whom they have no preexisting relationships
Malicious Code	:	Worms and viruses are in a more general category of programs called "malicious code." Both exploit weaknesses in computer software, replicating themselves and/or attaching themselves to other programs

1. Introduction

That the short-lived software vulnerability information may be traded in the black market, particularly if the vulnerability is discovered by black hat hackers, has increasingly been discussed lately. The issue of unknown attacks and the circulation of vulnerabilities in a 'black market' have been getting increasing attention. A worrying aspect of black market presence is that it may involve organized crime. As stated by Schneier[1], organized crime is a significant problem since it likely to be better funded, better skilled and better organized than lone criminals and hackers are. Miller (2007) also indicates that the illegal market for the 0-day exploits has begun to be more economically based, as spammers and criminals are increasingly interested in the use of 0-day exploits for use in illegal activities.

Greater attention should be directed at the vulnerability black market issue because it may create more severe security problems, such as: 1) Lengthening the period of the circulation of the secret vulnerability information; 2) Extending the possibility of malicious actors to launch attacks on unprotected computer networks; 3) Discovering new means to commit more cyber-crime and other illicit activities; 4) Giving incentives and provide alternatives for the hackers to sell the vulnerability findings; 5) Enabling the faster growth of the criminal activities because people can easily find 'tools' offered by 'black-hat hackers' in the black market; 6) Can be an 'opportunity' for hackers to resale the information about an exploitable vulnerability to the black market, after having sold the same information to the legal market.

This paper attempts to answer a question of what factors do affect the emergence and growth of the black market for vulnerabilities. A preliminary system dynamics model is developed to examine the dynamic nature of the problem. From a system dynamics viewpoint, it is pertinent to seek systemic explanations, the nature of the information security/ software vulnerability problems and look for the accumulations, time lag, feedback and non-linearities. System dynamic modeling can help the authors to build structural representations of parts of software vulnerability and software security problem, especially to explain the cause of the problem and to simulate of the model at the root of the system. This paper is organized into five sections: Introduction, An Overview of the Software Vulnerability Issue, Problem Identification, Model Description, Model Behavior, and the last part is Conclusion and Future Research.

2. An Overview of the Vulnerability Black Market and Software Vulnerability Issue

A software vulnerability is a bug, that is a *flaw, weakness or defect* in the code of any program (e.g. application, operating system) that can be exploited by malicious agents (Arbaugh W.A. 2000, p. 54; Wahlström 2005; Seacord and Householder 2005), and makes it susceptible to attack. These vulnerabilities may result from mistakes when writing software, whether a math error, incomplete logic or incorrect use of a function or command (Martin 2001). For further discussion on software errors classification, see Landwehr et al. (1994) and Du and Mathur (1998).

In this section we briefly illustrate the vulnerability black market terms and issues, and how it relates to the general software vulnerability problems, vulnerability discovery and the on-going debates on policy to improve software security.

[1] See interview with Bruce Schneier, http://www.pbs.org/wgbh/pages/frontline/shows/hackers/interviews/schneier.html, quoted September 20, 2006.

2.1. Vulnerability Black Market

The vulnerability black market discussions surface almost as the same time as the increasing public debates on the emergence of legitimate markets where vulnerability researchers can sell vulnerability information. However, the existence of black hat hackers that may be working underground has long been known. The emergence of the vulnerability black market and its existence is considered as a new challenge for IT security, may generate a new dimension of information security threat and may result in more zero-day attack (Kaplan 2006). A thriving black market is appearing because there are small communities of researchers discovering and managing to sell vulnerability information to crime syndicates (Higgins 2006) or like a collaboration between hackers and criminals (Whipp 2006). This market might be a lucrative (Naraine 2006), where hackers and malicious actors can secretly buy, sell and trade these vulnerabilities online (Stone 2007). The black market might be attractive because the prices in the evolving black market may be higher than what legitimate companies would pay (Stone 2007). Exploit code for an unknown flaw is considerably more valuable (Landesman 2007).

The term "black market" usually refers to those transactions which take place illegally at prices higher than a legal maximum. Essentially the same phenomenon is observed when the illegal transactions take place at price below a legal minimum. The Merriam Webster Dictionary defines a black market as "an illicit trade in goods or commodities in violation of official regulations". Clinnard states: "literally, "black market" means illegal business conducted in the dark…safe from the light of day or, more loosely, the light of public gaze. This would include the out-and-out thieves…the professional burglar…organized gangs…prohibition racketeers and hijackers…and unscrupulous businessmen who will do anything for their own personal profit (p.14)".

The term of "black market" originally appears in the Second World War especially in United States, when drastic regulations were issued making it against the law to charge more than a certain ceiling price for nearly all commodities (Clinard 1969). The actual origin of the Black Market term is not quite clear, although it appears to have been identified with "black" to indicate illegal activities occurring under condition of great secrecy (Clinard 1969, p.2). Most of black market behaviour involved violations that were complex, evasive, and wilful in nature, and should be considered "crimes". Perhaps there are similarities between some of wartime disregard for law and vulnerability black market, insofar as both constitute illegal activities under condition of great secrecy.

In every market, there are sellers and buyers. They are the fundament of trade, along with the actual exchange of goods and services and the associated transaction. If the number of buyers increases, the number of sellers tends to increase as well. In particular cases, e.g. when there are incentives for criminal activities, a black or underground market tends to appear. We believe that the vulnerability black market, too, is governed by the laws of supply and demand. As long as there is someone willing to pay, there will be someone willing to sell.

It has been mentioned that vulnerabilities might be discovered by people with an interest in exploiting it. Instead of notifying the vendor, they take themselves advantage of the vulnerability and potentially inform some of their associates. Information about the vulnerability circulates in the hacker community and the vulnerability may be exploited (by a limited numbers of hackers) before it becomes known publicly. Neither vendors, nor users are conscious of this threat.

It has been suggested that there is a massive underground trade in software vulnerabilities, particularly during the period of private disclosure. Organized crime pays high prices for information that helps to break into corporate databases for identity theft and other lucrative criminal activities.

The main reason for hackers to search for vulnerabilities is to obtain higher opportunity for financial achievement through successful exploitation. But recently this phenomenon has been shifting character. Hackers find unknown vulnerabilities and sell them to the highest offer. An example of how hackers advertise their findings can be found in the Finjan's report (2006). Hackers start considering the private disclosure of a vulnerability as a business opportunity. Itzhak (2006) entitles it the "malicious code food-chain". Hackers use the Internet as the main channel for sharing information about exploiting software vulnerabilities and exposures. Since the number of Internet users is growing and intruder tools are becoming more sophisticated and easier to use, more people can become "successful intruders". In addition, malicious code developed lately is easy to launch remotely. It might be a source of attractiveness to buy "secret vulnerabilities".

To understand how actors in the underground vulnerability market develop their revenue stream, Sutton and Nagle (2006) introduce the *contracted model* and the *purchase model*. In the former model the malicious actor hires a hacker to find a vulnerability in a specific target. However, they underline that there is little public information on the contracted model. The purchase model is done in reverse from the contracted model. In that model the hacker finds a vulnerability, creates an exploit and sells it to the malicious actors. Sutton and Nagel (2006) emphasize that all parties have to broker the deal, involving some potentially risky contracts, while making sure that they are not caught by law enforcement.

As indicated in the introduction section, this paper seeks the underlying factors influencing the emergence of the black market for vulnerabilities. To accomplish this goal, we cannot neglect the history and problems surrounding vulnerability discovery and disclosure policy, as well as the current discussion and development of legitimate market. Has the black market for vulnerabilities existed before the vulnerability disclosure and the emergence of the legitimate market, or did the black market surface because of the legitimate market? Is the legitimate market formed to attract hackers and security researchers or to contain the black market? The discussion in next sub-section will briefly investigate this problem, and assess how the vulnerability black market issue is connected to this development.

2.2. Software vulnerability problems

Vulnerabilities in the software attract at least four different parties with various interests (Figure 1). The relationships among them are always a dynamic between the malicious attackers who are searching for methods to exploit the weaknesses in the software, and parties who want to defend the systems from any attacks and computer failures, to eliminate the flaws and to mitigate the security risk in their computer network.

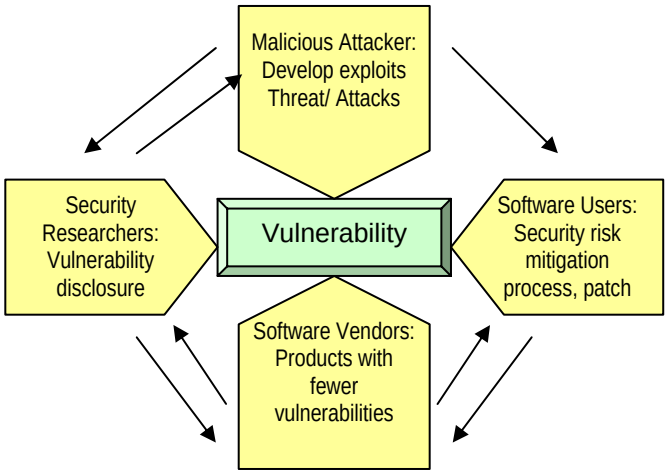

Figure 1
Vulnerability and Different Interests of the Involved Parties

The ideal situation when encountering the software vulnerabilities is that the non-malicious parties play they role as they should be; *vendors* develop more secure software and patch the vulnerabilities as well as offer a good protection to the clients, *security researchers* implement "responsible" vulnerability disclosure, and *software users* keep their computer with new updates to mitigate security risks. Imbalance of this system happens because each actor plays inappropriately and leads to more complex relationships and further software vulnerability problems: users do not patch, vendors produce buggy software, or software vendors don't reward security researchers, while security researchers (can be black hat and white hat hackers) eventually trade their findings (they may sell to the security companies or to malicious criminals). In addition, it is unclear how to channel the vulnerability discoveries and there are some disagreements among non end-user actors regarding how to disclose vulnerabilities. This problem will be elaborated in the next sub-section.

2.3. The Policy to Improve Software Quality

Given the enormity of damage and the fact that vulnerabilities cannot be completely eliminated in the software, vulnerability disclosure has become a critical area of concern for policy makers. However, there are ongoing and vivid debates over time in the information security community against different vulnerability disclosure models: *vulnerability secrecy/non-disclosure/ security by obscurity* (to suppress publication entirely until patches or updates are available), *vulnerability disclosure* (to publish full details) and *responsible disclosure* (to conceal some details). This sub-section also briefly reviews the emerging vulnerability discovery through the market.

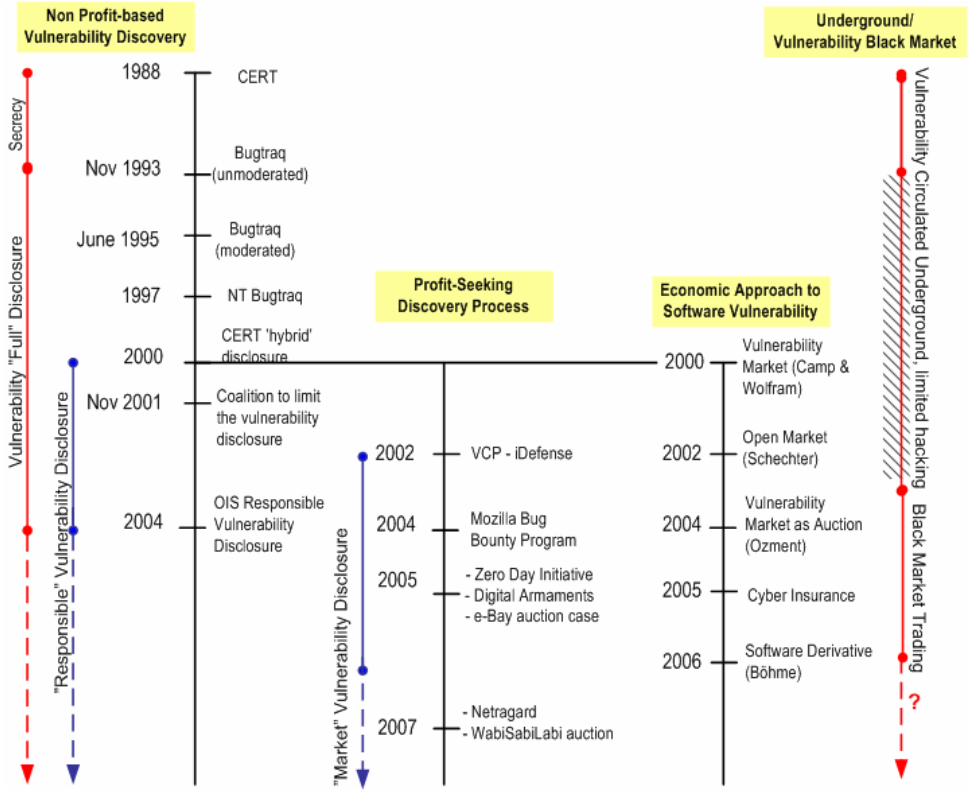

Figure 2
Summary Of Vulnerability Discovery and Vulnerability Market History

VULNERABILITY SECRECY:

For years, full-disclosure was not practiced by vulnerability researchers or vendors. Security by obscurity attempts to use secrecy to provide security, because if flaws are not known and attackers are unlikely to find them. Small groups of interested parties exchanged information amongst themselves, unwilling to disclose it to the masses. As these bugs were slowly found by others or passed on to vendors, they eventually got fixed. This "security through obscurity approach" didn't lead to secure software. Before the disclosure policy was introduced, the software companies were inclined to take no notice of the vulnerability reported by security researchers and trusted the vulnerability secrecy. Furthermore, it was considered to be 'illegal action' if security researchers disclosed vulnerabilities (Schneier 2007).

CERT (*Computer Emergency Response Team*) established by DARPA (*The Defense Advance Research Projects Agency*) in 1988, to coordinate and response internet attacks, including the vulnerability reports (Schneier 2000c). Over the years, CERT has acted as a central agency for reporting vulnerability. Researchers are supposed to alert discovered vulnerabilities to CERT. This agent will verify the vulnerability and silently inform the vendor and make public the details (and the fix) once patches are available. In sum, people were keeping software vulnerabilities secret.

Vulnerability concealment has been criticized for making delay between vulnerability finding and patch development. Secrecy prevents people from accurately assessing their own risk. Secrecy precludes public debate about security and inhibits security education that leads to improvement.

VULNERABILITY DISCLOSURE:

The full-disclosure movement started because of the dissatisfaction with above "slow" process. CERT obtained a great deal of vulnerability reports, but was very slow in verifying them; the vendors were slow to fixing the vulnerabilities after the notification, and CERT was slow to publish reports even after the patches were released (Schneier 2000c). Internet mailing lists such as Bugtraq (begun in 1993) and NT Bugtraq (begun in 1997) become open forum for people believing that the only way to improve security is to understand and publicize the problems (Schneier 2000c; Rauch 1999; Schneier 2007). Full-disclosure mailing lists and newsgroups fill this role. Vulnerabilities and solutions are disclosed and discussed openly. Many researchers publish vulnerabilities they discover on these mailing lists, sometimes accompanied by press release. In essence, full-disclosure is the practice of making the details of security vulnerabilities public. Since 1995, the growth of people participating in "Full Disclosure" has increased significantly (Rauch 1999).

The proponents of this idea believe that the policy will force the vendors to be more responsive in fixing vulnerabilities, and security will be better (Rauch 1999). Full disclosure proponent argues that public scrutiny is the only reliable way to improve security (Schneier 2000b, 2000c, 2001, 2007; Levy 2001). Keeping software vulnerabilities secret intended to keep the information out of hands of the hackers. But hackers have proven to be skilful at discovering secret vulnerabilities, and full disclosure is the only reason vendors regularly patch their systems. Critics to the full-disclosure movement especially points out that hackers at the same time can use these mailing lists to learn about vulnerabilities and write attack programs (called "exploits"). Previous to public vulnerability disclosure, the actors exploiting the vulnerability may only be the ones who discovered it, and they can compromise only a limited number of machines. If they do automated exploits or use a worm, the chances of being discovered are high and their zero-day backdoor becomes publicly known and subsequently patched. However, after the vulnerability is publicly disclosed, the world learns about the flaw, and the number of computer victims will increase significantly (Grimes 2005).

The debates between proponents and opponents of full disclosure (Farrow 2000; Jericho 2001; Schneier 2001; Grimes 2005; Mimoso 2001; Ranum 2007) can be summarized as follows:

Table 1.
Summary of the reasons of proponents and opponents of the Full Disclosure (FD)

Disagree	Agree
• Nobody except researchers need to know the details of flaws	• FD helps the good guys more than the bad guys
• FD results in information anarchy	• Effective security cannot be based on obscurity
• Good guys who publish virus code may also have malicious intention	• Making vulnerabilities public is an important tool in forcing vendors to improve their products
• Safer if researchers keep details about vulnerabilities and stop arming hackers with offensive tools	• If an exploit is known and not shared, the vendor might be slower to fix the hole
• The risk associated with the publishing information outstrip its benefit	• Sharing information security with other professionals is an absolute necessity
• It serves to arm hackers with tools to break systems	

RESPONSIBLE DISCLOSURE: Accordingly, software companies and some security researchers proposed "responsible disclosure". The basic idea was that the threat of publishing the vulnerability is almost as good as actually publishing it. A responsible researcher would quietly give a software vendor a start on patching its software. CERT/CC (2000) introduced a new vulnerability disclosure policy, although the information security community still have doubts about this proposal (CyberEye 2001). All vulnerabilities reported will be disclosed to the public 45 days after the initial report, regardless of the existence or availability of patches or workarounds from affected vendors. In addition, pressure also came from a coalition of well-known software developers and some security companies established to push a standard policy of limiting public disclosure of security vulnerability (Middleton 2001), and a number of guidelines are currently available to govern the relationship between the vendors and the identifiers. Software vendors and security research firms have begun to jointly develop a unified framework for vulnerability disclosure under OIS (*Organization for Internet Safety*) Guidelines (2004). Basically they set a certain grace period for not disclosing any information to third parties until the manufacturer releases a patch. Some issues that may appear from responsible disclosure has been also discussed by Cavusoglu et.al (2005).

"LEGITIMATE" MARKET: THEORIES AND PRACTICES

In line with the vulnerability disclosure debates and the emergence of the economic of information security in early 2000s, a new "stream" of the "Market" Vulnerability Disclosure surfaces, both in the theoretical and practical level. Anderson (2001) argues that most security problems cannot be solved by technical means only. Instead, some microeconomics terms are more able to explain some of security problems (Anderson and Moore 2006). Schneier advocates that economics has appropriate theories to deal with computer security issues (Schneier 2006).

On the theoretical level, the initial thoughts on the economic of vulnerabilities concerns measuring software security through market mechanisms. Camp and Wolfram (2004), proposed a market through which vulnerability credits could be traded; such markets have worked previously to create incentives for the reduction of negative externalities like environment pollutants. Schechter (2002) proposed creating markets for reports of previously undiscovered vulnerabilities, while Ozment (2004) proposed a vulnerability market as an auction. Böhme adds (2006) vulnerability brokers, exploit derivatives and cyber-insurance in his discussion to compare the best vulnerability market type where security-related information can be traded and to find which type serves best to counter security market failures. However, Kannan and Telang (2005)

criticize that the business models of these organizations are not socially optimal and Rescola (2005) finds no support for the usefulness of vulnerability finding and disclosure.

On a practical level, the 'legitimate' market for vulnerabilities is developing as well. Apparently, this is also a period of "commercialization" of vulnerability research. Sutton and Nagle (2006) wrote a paper based on the model that already exists in various markets rather than a theoretical model and classify the current vulnerability market discussion as government market, open market, underground market, auction market and vendor market. iDefense announce the VCP (*Vulnerability Contributor Program*) in one security mailing list in year 2002, [2] offering reward for verified vulnerability information. In 2004, Mozilla Foundation offers payment to those who find critical security flaws in its product, including the Firefox Web browser (Lemos 2004), and follow by TippingPoint that announce the ZDI (*Zero Day Initiative*) in 2005 (Evers 2007) and DACP (*Digital Armaments Contribution Program*) in the end of 2005. iDefense gains revenue by directly reselling the information, while TippingPoint profits by offering exclusive protection against the vulnerability they purchase via Intrusion Detection System product. In 2007, two new marketplaces emerge: Netragard, and Wabisabi Labi as an auction site. However, Ozment and Schechter (2006) criticize the obscurity of the price of vulnerabilities that hinders toward open market. The worrying aspect of all market-based approaches is that they may increase the number of identified vulnerabilities by compensating people who would otherwise not search for flaws.

2.4. Review on The Vulnerability Discovery Discussion and the Black Market

The previous historical depiction is to trace the birth of the underground discussion and development of the black market trading. The vulnerability black market may already have existed before the vulnerability disclosure policy. Some full disclosure proponents has already used the argument of avoidance of vulnerabilities being known by attackers and passed about quietly in the hacker underground (Schneier 2000c), to support Full disclosure.

If we observe carefully, as full disclosure is implemented, there is an indication that the information from full disclosure discussions in the mailing lists may also be traded among the underground actors looking to break into machines (Rauch 1999). Rauch also concludes that full disclosure results in a grey market economy in exploits, where independent "vulnerability researchers" attempt to sell their findings to security companies or spyware manufacturers, whichever bid higher. Some opinions consider the full disclosure may trigger further the underground vulnerability circulation.

In addition, with the awareness about the vulnerability black market presence, even after the legitimate market became available, some critics claim that actually not too much critical vulnerabilities being sold in the legal market. Hackers want to keep those to themselves and use them to exploit systems in the wild, and it is doubtful that the hackers underground are motivated to sell vulnerabilities to security company if they earn more by holding the private vulnerability information (Evers 2007).

3. Problem Identification

The literature on misperceptions motivates the authors to view this problem using a system dynamics approach. Sterman (2000, p. 25) writes that people perform quite poorly in systems

[2] See: http://lists.canonical.org/pipermail/kragen-tol/2002-August/000729.html, and the vulnerability contributor program, see: http://labs.idefense.com/vcp/

with even modest level of dynamic complexity. Observed dysfunction in dynamically complex settings arises from misperceptions of feedback.

In our case, we assume that the root of the vulnerability black market has to do with inability to understand closed-loop causality and factors diverting the policy from its original goals. Insecure software permeates the market because most users cannot distinguish it from secure software, due to asymmetric information: Vendors are not compensated for costly efforts to strengthen the software's code. There has been a continuous increase of exploitable vulnerabilities with a parallel increase in the computer security breaches. A vulnerability disclosure policy as previously described is carried out to press the vendors to patch faster.

Incomplete information about the software quality means that software users can't frame a good decision to "judge" the software security. Asymmetric information between buyers and sellers makes the software product market a "*market for lemons*" (Akerlof 1970). As it is now, the software users are unwilling to pay higher prices for uncertain security features. Accordingly, the vendors have little incentive to improve the security of their software products. The evidence for this claim is strengthened by a few studies that companies are not conducting adequate software testing process to assure the quality of the software in advance of the product marketing (Minasi 2000; Tassey 2002). A simplified, idealized problem of asymmetric information, software quality, computer security breaches and the vulnerability black market can be shown in Figure 3:

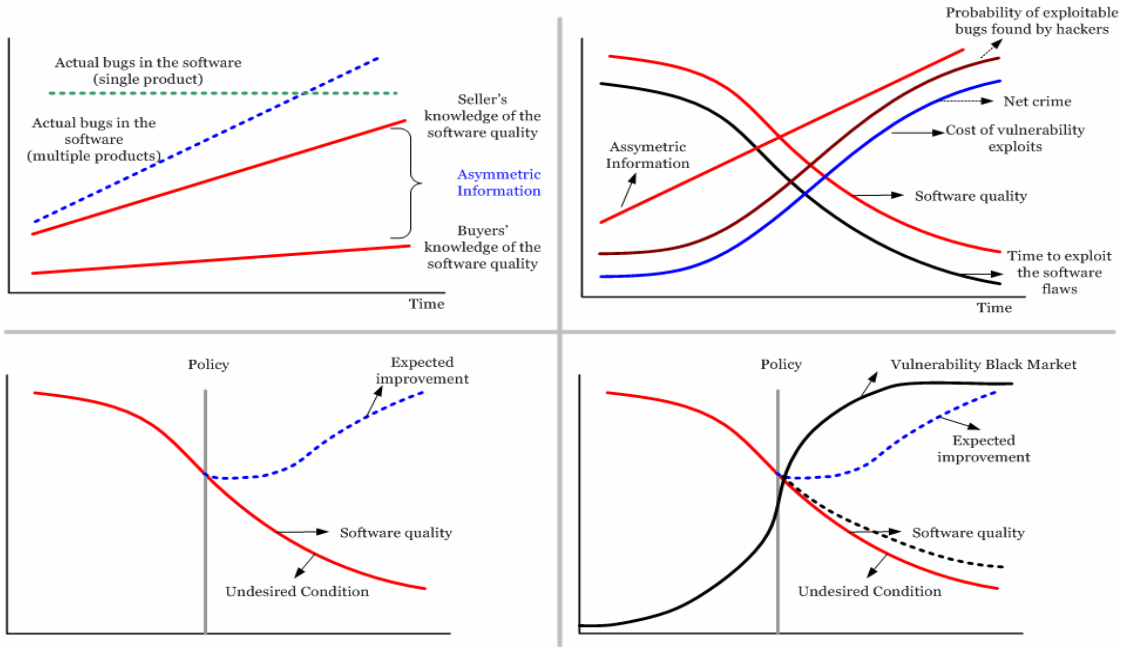

Figure 3
Figure 3.1 (top left) and 3.2 (top right)
Consequences of asymmetric information in the software market
Figure 3.3 (bottom left) and Figure 3.4 (bottom right)
Software quality and undesired conditions

Software producers produce continuously new releases with richer features and updated software but, as experience shows, with increasingly more bugs. As software is becoming more and more complex and larger, bugs in the software tend to increase too. The gap between sellers' knowledge and buyers' knowledge of the quality of the software is growing. The enlarged asymmetric information creates further consequences. Insecure software keeps circulating in the

market, the software quality declines, and the probability of exploitable bugs found by hackers might get higher over time. The presence of parties willing to purchase secret information of the vulnerability could make the situation even worse. The vulnerability black market will have a greater opportunity to develop. What kind of activities might be performed in the black market? The possibilities include: a malicious actor/ a group of criminal hires a hacker to find vulnerabilities and create exploits; or the hackers find the vulnerability, create exploits and advertise/offer them to anyone who is interested in buying and using the exploits.

On the other hand, as the time to exploit the software flaws is getting shorter, the possibilities of malicious actors to take an advantage of this weakness and commit cyber crime are increasing. This circumstance will result in more computer-crimes. Attacks exploiting software vulnerabilities cause more economic damage. [3]

Time Horizon
We shall consider the time horizon for the model to be the period of 260 weeks.

Model Boundaries
The model boundaries will be drawn around the asymmetric information, software quality, policy to improve software quality, unintended impacts of the policy, including the emergence of the vulnerability black market. We assume that the vulnerability black market has already existed and we analyze which factors force the black market development and which factors hinder the software quality improvement.

4. Model Description

Although there is not yet enough empirical data to develop a detailed model representing the environment of the case study, a basic model representing a simple structure of the problem can be built with the current available information and literature.

Even such a basic model provides some advantages. First, issues and problems concerning the technical difficulties of this modelling approach in this research will be revealed earlier. Second, as the empirical data of the research accumulates the basic model will grow to a more comprehensive model.

The model intends to explain how the vulnerability black market may emerge and develop. The model consists of four main sub models (Figure 4), namely: *vulnerabilities trading* sub-model, *software security quality* sub-model, *computer incidents* sub-model and *risky environment* sub-models. The vulnerabilities trading sub-model is sub-divided further into *unknown vulnerabilities*, *vulnerability traded in black market* and *vulnerability traded in legal market.*

The model is based on the following assumptions:

- There are a fixed number of unknown vulnerabilities.
- There are a fixed number of potential buyers and sellers in the vulnerability black market.

[3] The 2005 *FBI Computer Crime Survey* found out that nearly 9 out of 10 organizations experienced computer security incidents in a year; 20% of them showed they had experienced 20 or more attacks. The types of attacks particularly are viruses (83.7%) and spyware (79.5%). Over 64% of the respondents incurred a loss. Viruses and worms cost the most, accounting for $12 million of the $32 million in total losses. See, http://www.fbi.gov/page2/jan06/computer_crime_survey011806.htm

- The unknown vulnerabilities in the model are equally exploitable and hazardous (it is possible that a vulnerability is found but theoretically is not exploitable, but we don't intend to differentiate between these two types of vulnerabilities).
- Vendors' knowledge on the software bugs is equal to the discovered bugs in the software (because actual number of bugs may be higher than vendors' knowledge on the bugs).
- The model assumes that the vulnerability trading in the black market in small scale starts at the beginning of the simulation time.
- We are talking about vulnerabilities that is discovered after the software is released.
- There is a risk that people will exploit it after the discovery.

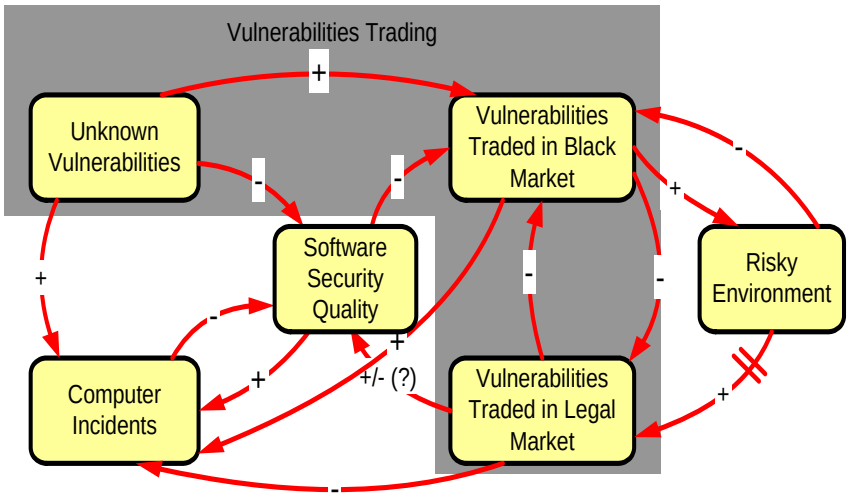

Figure 4
Sector Diagram

4.1. Software Security Quality Sub Model

We begin the model to explain how the vulnerability black market appears, by describing the current software security quality problems. The Software Security sub-model is shown in Figure 4, and covers three loops: software security, market rewards and asymmetric information.

LOOP B1 THE SOFTWARE SECURITY

This is a balancing loop. We assume that there is a software quality gap and the situation is formulated as the discrepancy between actual *Software Security Quality* and *Desired Software Security Level*. Ideally, responsible vendors will invest more resources to improve the software quality to close the gap, and to accomplish the highest security level possible. We name *Security Improvement* as an action to close the gap, so that the software security will get better and move toward the desired software security level.

LOOP R1 MARKET REWARDS

On the other hand, the loop *Market Rewards* works as a reinforcing loop that may strengthen or weaken the effort to improve security. Software users don't immediately acknowledge the quality of the software. They react to the quality of software through a perception delay after experiencing how good the quality of the software is. We capture this perception through a variable *Security Observed by Software Buyers*. The positive relationship exists between

perception about software quality and the market rewards. Dissatisfaction of software users about security quality of the software affects them not to provide higher appreciation on the software and thus, lower the market rewards. This situation affects positively the *Incentive to Improve Security* and *Desired Software Security Level*, respectively. This reinforcing loop declines the effort to improve security. In addition market rewards also affects the "goal" of the vendors to increase the quality. Low appreciation not only prevents the action to improve the software quality, but also impede the efforts to have a higher goal to improve security.

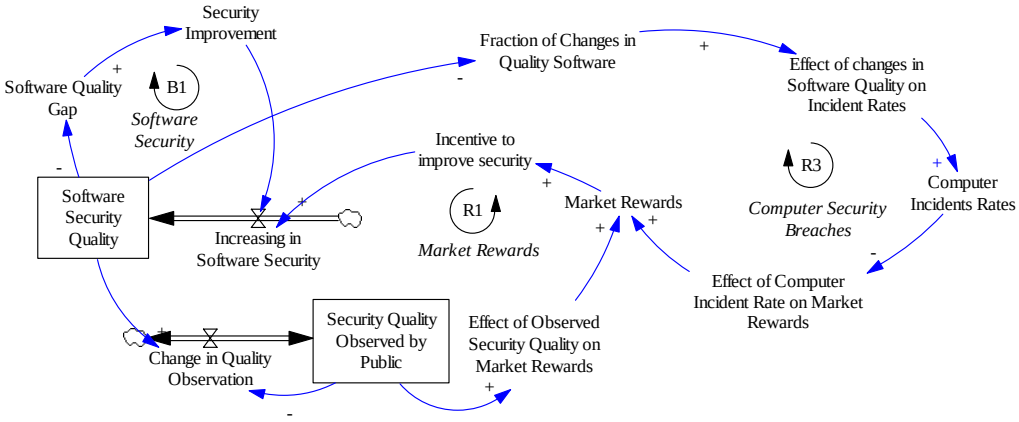

Figure 5
Software Security Quality and Computer Security Breaches

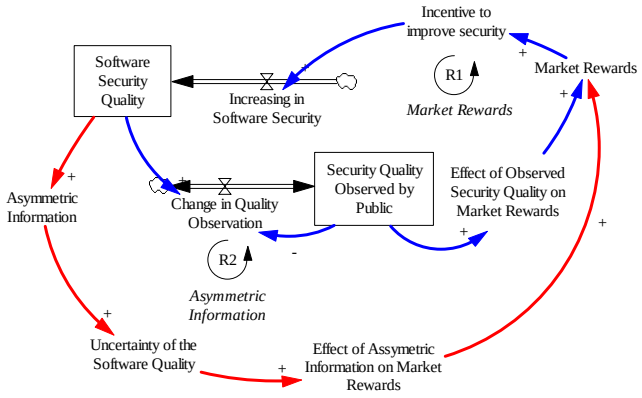

Figure 6
Asymmetric Information

LOOP R2 ASYMMETRIC INFORMATION

The asymmetric information as a reinforcing loop also influences the state of the software security quality. Loop R2 captures the asymmetric information problem that has been mentioned in the problem definition section (*see* section 3). Basically the asymmetric information in this model emerges as a discrepancy between buyers' and vendors' knowledge on software quality (not shown in Figure 4). The assumed relationship between *Asymmetric Information* and *Uncertainty of the Software Quality* is positive. The inability of the software users to differentiate good and bad software generate uncertainty and influences them to disreward software security, that is, the market is unwilling to pay more for uncertain software quality. In this model, buyers' knowledge on software quality is assumed to be smaller than the vendors' knowledge. It means that the uncertainty about software quality is high. The uncertainty in the model is defined as a value between 0 and 1. If the value is equal

to one, buyers have perfect knowledge about the software quality. If the value is <1 or near to zero, it means that buyers lack knowledge about the software flaws, and uncertainty becomes high. Consequently, the situation further decreases the market reward and also suppresses the vendors' willingness to refine the security of the software. This condition results in more products of low security quality circulating in the market.

Loop R3 Computer Security Breaches

This loop describes another effect of the low security quality software problem. This weakness allows hackers to seek the flaws in the software and for malicious agents to launch attacks by misuse of the software flaws. Decreases or increases in the software security are represented by the *Fraction of Changes in Software Quality*. The more exploitable bugs are in the software, the higher is the likelihood of the computer incidents rates. If the computer incidents rates increase then more people are unwilling to give higher appreciation on computer security.

Our description so far relate to feedback that influence the dynamics of the software vulnerability problem toward low software security. The situation is captured in the model by three loops (*market rewards, asymmetric information and computer security breaches*) that influence the action to close the software quality gap. These three effects counteract the improvement effort. The next feedback described below captures people's reaction ("policy") to mitigate the vulnerability problem in the software, by making the discovered vulnerability announced publicly.

Loop B2 Pressure to Vendors: Further Action to Close the Gap

As indicated in the Introduction section, Full Disclosure policy is the practice of making the details of security vulnerabilities public. By announcing discovered vulnerabilities in software products, it is expected that vendors turn out to be more responsive in fixing them. Consecutively, the software security will improve faster. The model captures the situation through a feedback from *Security Quality Observed by Public* that leads to the idea of implementing the *Full Disclosure Pressure*. This causal relationship is to "correct" the vendors' goal in producing the software products, such as to give more attention to conducting adequate testing in advance of software marketing and to fix vulnerabilities faster. Higher stress from *Full Disclosure* will enhance the desired software security level, and drive further effort to close the quality gap. Eventually, more computer incidents can be avoided or prevented as software quality increase (less exploitable bugs). However, the potential that the gap could be higher than originally exists too. It happens when the *Desired Software Security Level* increases and no alterations in the software users' perception to appreciate the software security quality.

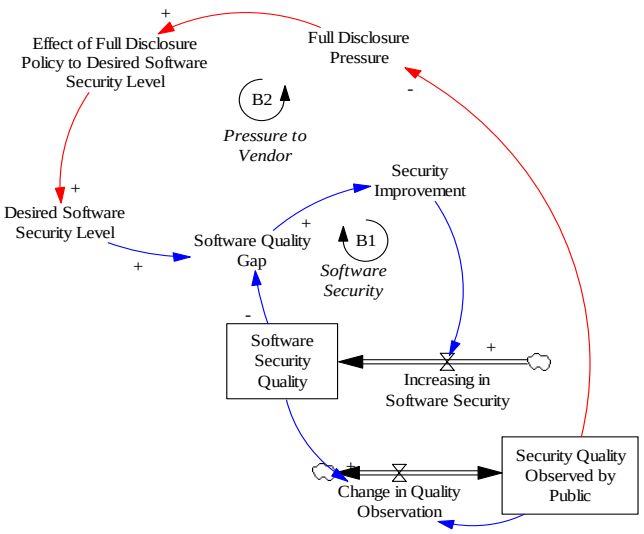

Figure 7
Full Disclosure and Software Security Quality

4.2. Computer Security Incidents Sub Model

We may call this sub model a *policy resistance*. As we mention in section 2, the vulnerability disclosure policy was introduced in 1993 to mitigate the impacts of the vulnerabilities and to pressure vendors to provide patches faster. Using mailing lists, such as *Bugtraq* and *Full-Disclosure*, vulnerability hunters can publish vulnerability report details about new vulnerabilities. However, undesired consequences arise, such as *rush patch cycle*, which may lead into a greater number of application failures since vendors might not have enough time to test new patches, and *zero day exploits*, which may lead to a greater number of security breaches. Since hackers learn about a new vulnerability at the same time as the vendors and public, hackers may develop exploits faster than the vendors can develop patches and the public can install those patches.

Zero Day Exploit and *Rush Patch Cycle* loops as illustrated in Figure 8 exemplify unintentional effects of the full disclosure policy. These two loops may hinder further endeavors to improve security.

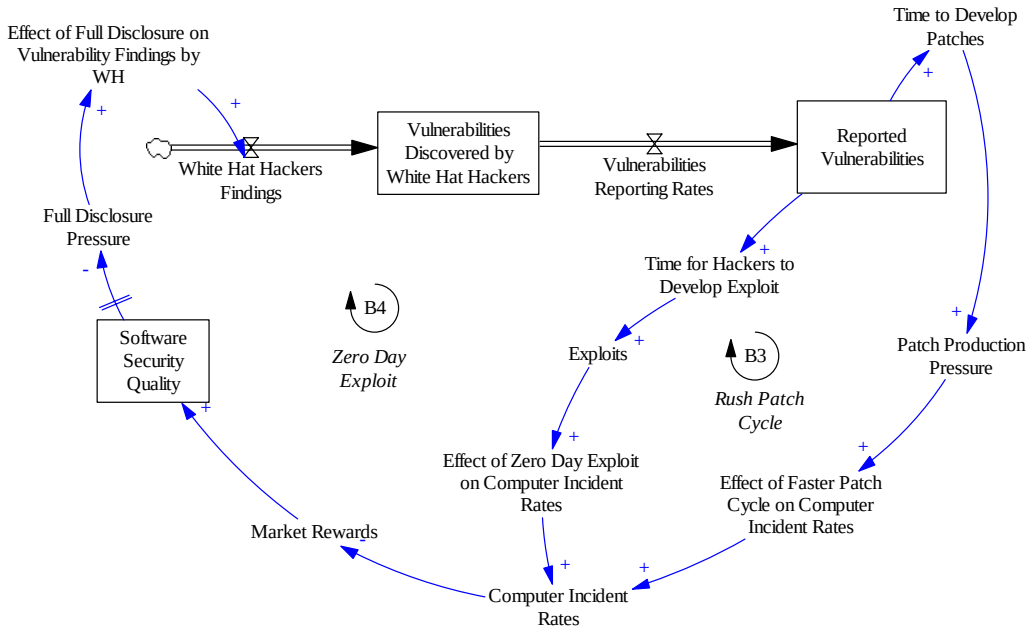

Figure 8
Unintended Impacts of the Full Disclosure Policy

LOOP B5 ZERO DAY EXPLOIT

The Full disclosure movement allows the hackers to learn about a new vulnerability at the same time as it is announced. This situation provides an opportunity for hackers to develop exploits, before the vendors develop patches. It has been known that the time needed by the hackers to develop an exploit after discovery of the vulnerability, is also getting shorter. It is estimated that the average time needed by hackers to develop exploits is around six days. In some cases, they even only need one day. On the other hand, vendors need longer time to develop patches; on average 54 days. In this model, it is shown that "full disclosure" attracts security/ vulnerability researchers to actively seeking vulnerabilities, which is represented by *Effect of Full Disclosure on Vulnerability Findings by WH*. It will affect stocks of *Vulnerabilities Discovered by White Hat Hackers* and *Reported Vulnerabilities*. The more reported vulnerabilities, the faster hackers can develop exploit. And it leads to more exploits, and affects positively the *Computer Incident Rates*. Once more this unintentional impact

makes the security goals become further from the initial intention. Therefore, it strengthens the computer security breaches loop which lowers market rewards, if the frequency of incidents increases.

LOOP B5 RUSH PATCH CYCLE

On the other hand, vulnerability reports force vendors to rush in providing workaround solutions and security updates that customers can use to mitigate exploitation of the reported vulnerabilities. To release updates on a compressed schedule, shortcuts must be made in the development process. These shortcuts can increase the risk that a fix won't resolve similar vulnerabilities in surrounding code or that a fix could have quality issues due to a shortened testing cycle. Rush patch cycle is demonstrated by the faster time to develop patch. And it has the potential that the patch developed for fixing the vulnerability is also not be tested properly. As a result, the potential of the computer failure because of untested patches is becoming higher, and again affects further the computer security breaches as well as the market rewards.

4.3. Vulnerability Finding, Trading and Vulnerability Buyers Sub Model

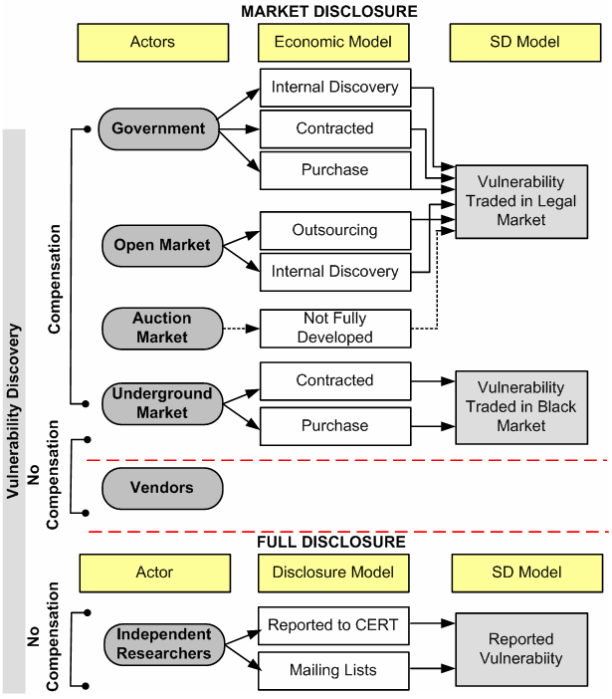

Figure 9
Vulnerability Discovery Flows and Simplification in SD Model

This sub model portrays the further impacts of the vulnerability discovery and the demand for vulnerabilities. The sub models will be presented and explained in Figure 10 and Figure 11, is an extension of the initial thoughts and model that have been developed in two papers (Radianti and Gonzalez 2006, 2007) . The openness about vulnerability disclosure drives black hat hackers and white hat hackers to actively seek the software flaws. The more incidents, the more software users turn into security companies. And the nature of vulnerabilities seeking activity has shifted: there is a supply and demand for vulnerabilities. Now, independent vulnerability researchers can decide to sell 0-day exploit to security companies or spyware manufacturers that may pay higher. How the vulnerability flows from market discovery process (*profit-seeking motive*) as well as from the full-disclosure process or is reported quietly (*non-profit motive*), and how the model captures this flow of vulnerability discovery can be seen in Figure 9. Basically, all vulnerability flows are simplified in the model as *Vulnerability Traded in Black Market*, *Vulnerability Traded in Legal Market* and *Reported Vulnerability*.

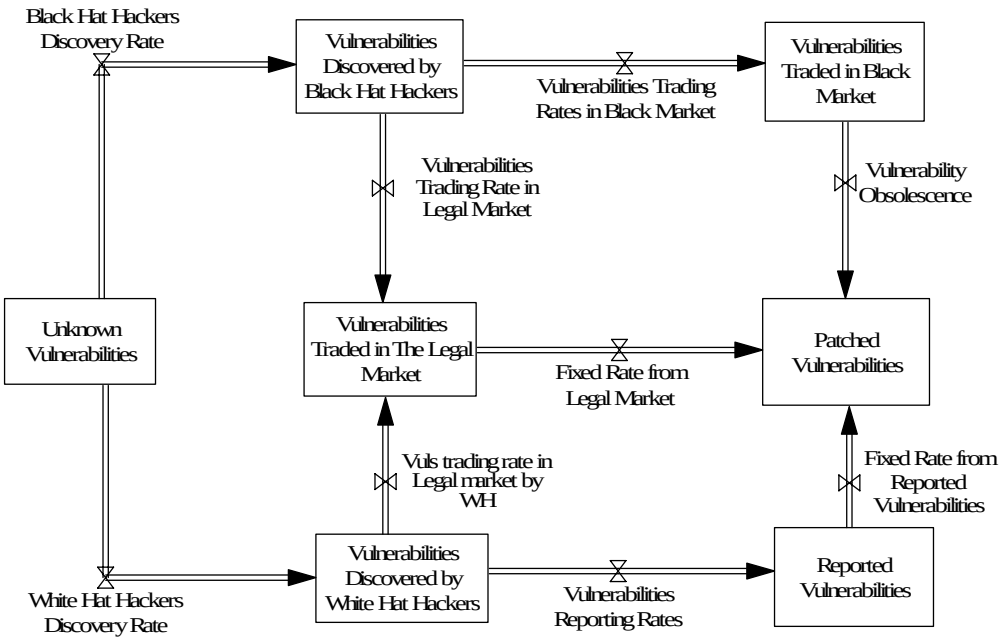

Figure 10
The Structure of Vulnerabilities Finding and Trading

Our simple model explains the growth of the vulnerability black market as follows: On the demand side (see Figure 13), there is a group of people who are willing to pay for secret knowledge about new vulnerabilities; on the supply side, hackers develop malicious code after they discover vulnerabilities. Hackers are finding new vulnerabilities and developing exploits, and then selling the whole package based on contract, or to a criminal group. In other words, hackers may trade the findings on the black market. It is assumed that hackers do not want to execute the exploits themselves out of concern for their own safety.

The main structure of the *Vulnerability Finding and Trading* sub-model (Figure 10) consists of two aging chains ("co-flows") vulnerability findings by black hat hackers and white hat hackers. The aging chain in the upper part depicts three stages of unknown vulnerabilities: before they are discovered until they are traded in the black market or reported. We split the vulnerability findings by black hat hackers and white hat hackers. The vulnerability findings by black hat hackers won't be announced and disclosed publicly.

In this model, the discovery rates from *Unknown Vulnerabilities* by black hat hackers and white hat hackers depend on *BH Productivity* and the number of *Black Hat Hackers*. *BH Productivity* representing the capability of hackers to find vulnerability/person/week is influenced by *Effect of Remaining Vulnerabilities on Discovery*. Once the vulnerabilities are discovered, they will be traded in black market. The circulation of such vulnerabilities will shrink after the vulnerabilities become publicly known, and patched, as the vendors learn about a new attack on computer systems.

Criminals or terrorist organizations are not the only potential buyers of yet unreported vulnerabilities. More actors now purchase unknown vulnerabilities, such as governments and security companies. We capture this by the rate *Vulnerability Traded to Legal Market* flowing

from the stock *Vulnerability Discovered by Black Hat Hackers* to the stock *Vulnerabilities Traded in Legal Market*.

The bottom part is the discovery by White Hat Hackers. They will report the vulnerabilities after they find the flaws publicly or quietly to CERT. There are also another option, that they may keep the vulnerability secret and don't report to any vendors and mailing lists. The grounds why white hat hackers don't want to report it, could be security reasons considering that reporting vulnerabilities is an unsafe action. But we haven't yet considered this possibility in the model. Similar to black hat discovery structures, the inflow to *Vulnerabilities Trading Rate in Legal Market* also comes from *Vulnerability Discovered by White Hat Hackers* as a possibility to channel their findings.

Vulnerabilities cease to be traded in the black market mainly because of two factors: 1) vendors create patches to fix the vulnerabilities or 2) post-depreciation phase, when the producer is no longer interested in actively improving the product or its security, usually because a successor product has become available. In this model this is captured by the outflow *Vulnerability Obsolescence*. In addition there is another feedback affecting this outflow, namely *Effect of Successful Attacks on Vulnerability Obsolescence*. This is to capture that the more people succeed in using unknown exploits (Figure 13), the faster the vendor develops a patch.

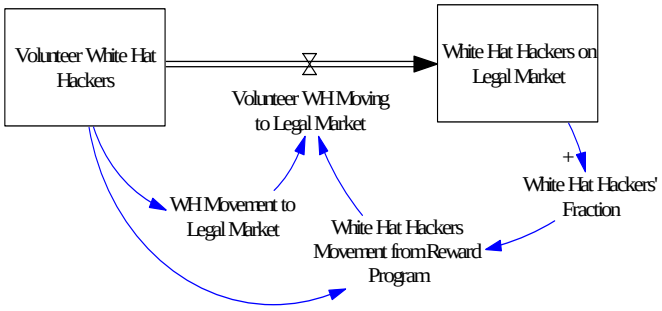

Figure 11
White Hat Hackers Movement

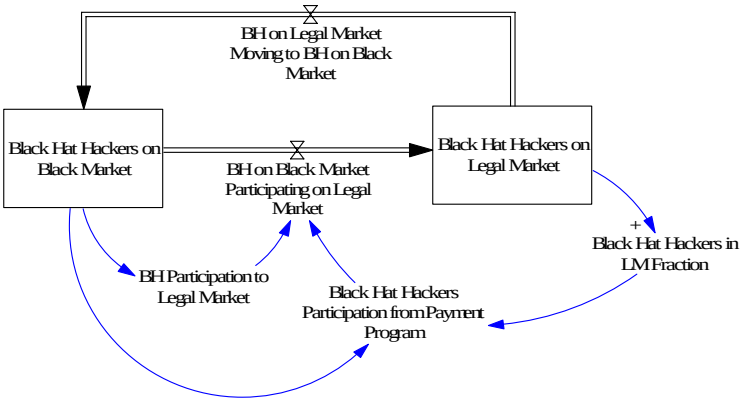

Figure 12
Black Hat Hackers Movement

In the vulnerability discovery sub model, the vulnerabilities flows from the unknown vulnerability are influenced as well by the number of hackers. The model initially assumes that hackers sell all their findings to the black market since the reward is bigger than for the legal market. All white hat hackers will notify their flaw findings. That is, at the outset the parameters *BH Trading Fraction and WH Reporting Fraction* have the value of one, implying that 100 percent of hacker's findings are traded to the black market, or are reported. By changing the values of these parameters, one obtains different simulation scenarios with varying degrees of trading to the black market. The legal market attractiveness has not yet been incorporated, so that there is no addition in the number of the *Black Hat Hackers on Legal Market* (Figure 12) and *White Hat Hackers on Legal Market* (Figure 11). Black-hat hackers may prefer selling their findings to black market because of higher rewards. In our model, hackers are assumed to be rational actors and they will sell their finding to the parties offering the highest price.

Once the black hat hackers are attracted by the legitimate market program, they will move to the legal market. The black hat hackers may shift to the legal market also because of safety reasons. Similar structures are also applied for the white hat hackers. Some researchers who always voluntarily report the vulnerabilities may also be attracted to the rewards from the legal market. The black hat and white hat hackers' participation on the legal market will decrease the number of of vulnerabilities traded in the black market or reported voluntarily.

VULNERABILITY BLACK MARKET BUYERS LOOP

Figure 13 shows two stages of black market buyers: from potential vulnerability black market buyers until they become successful intruders. It borrows the diffusion model structures to describe the flow of buyers among these two stages. The intention of showing this sub model is to describe how the vulnerability black market buyers may expand and grows.

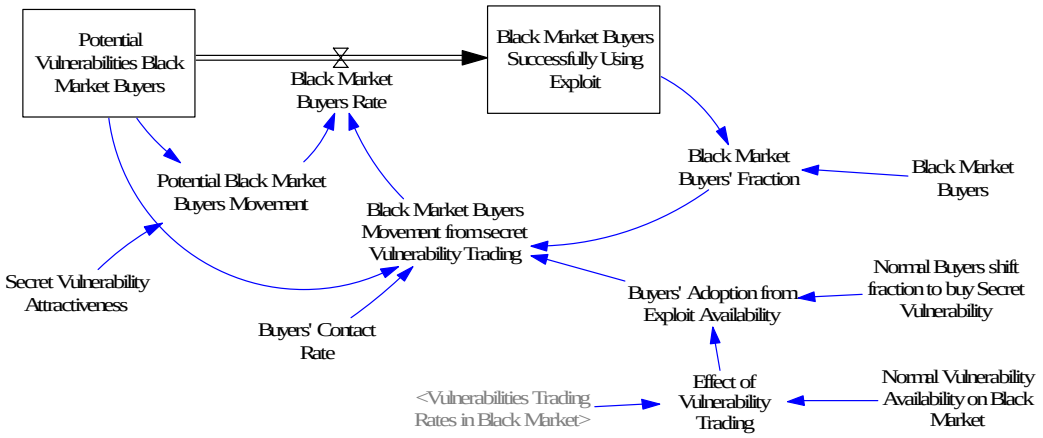

Figure 13
The Vulnerability Black Market Buyers

The buyers with malicious motive in the model are represented by the stock *Potential Vulnerabilities Black Market Buyers*. Hackers may advertise their findings online and we assume it is intended for a group of potential buyers. In order to become a real buyer, a potential buyer needs to be convinced that he will get a financial gain by investing his money on an unknown exploit. *Secret Vulnerability Attractiveness* will shift the *Potential Vulnerabilities Black Market Buyers* to decide to buy it. Once purchased, the vulnerabilities lead to exploits and increase the stock *Black Market Buyers Successfully Using Exploit*.

4.4. Risky Environment: Will Risk Hamper the Vulnerability Black Market Growth?

There is a further question about the likely growth of the vulnerability black market. Will the black market grow unlimited? What factors may hinder the vulnerability black market growth? Now we add a risk into the model. A hacker that identifies vulnerabilities in the software applications, can sell them on the black market or on the legal market. Hackers are assumed to be rational actors. Therefore we regard as two considerations for hackers to sell vulnerabilities on the black market: the financial gain and the *risk* of being caught by the law enforcement.

Loop Risk

Suppose a hacker consider selling his findings on the black market. For simplicity we concentrate on situations in which there are only two possible outcomes. If a hacker finds a buyer(s) and sells the secret vulnerability, he will earn higher income but he also has a risk of being caught by law enforcement. We assume the cost of going into jail is the same as his earning if he sells to the black market and the cost of finding the vulnerabilities is assumed to be zero. If he sells to the legal market, he earns less income. A hacker also has a possibility of failure, say if other people also find a similar bug. Since we assume the cost of finding is zero, he won't earn anything, but he has no risk being caught by legal officers. The advantage between selling vulnerabilities in black or legal market incorporates risk, will be stated in Expected Value (EV):

EV_{bm} is: $[Pr_{bm}$ *(succeed) x Value (succeed)*$]+[Pr_{bm}$ *(caught) x Value (caught)*$]$
EV_{lm} is: $[Pr_{lm}$ *(succeed) x Value (succeed)*$]+[Pr_{lm}$ *(competitor) x Value (competitor)*$]$

One such measure of risk is the variance and standard deviation (SD,σ). The smaller the standard deviation or variance, the smaller the risk is. If the EV is constant, and $\sigma_{bm} > \sigma_{lm,}$ then to sell vulnerabilities to the black market is riskier than selling to the legal market.

Will a hacker sell his findings on the legal market or the black market? To answer that question, we need to know a hacker's attitude toward bearing risks. In a theory of decision making under uncertainty, whether people choose a risky option over a non-risky one depends on their attitudes toward risk and on the expected payoffs of each option. The hacker will earn more if he can avoid law enforcement or lose more if he is being caught. He'll sell to the black market if he is a "risk lover". If people made choices to maximize expected value, they would always choose the option with the highest expected value regardless of the risks involved. However, most people care about the risk as well as expected value. Indeed, most people are "risk averse" and will choose a bundle with higher risk only if its expected value is substantially higher than that of a less risky bundle.

We capture the situation into our system dynamic model. We also incorporate the perceived risk concept so that the model is able to show whether hacker's attitude will change over time depend on the perceived risk.

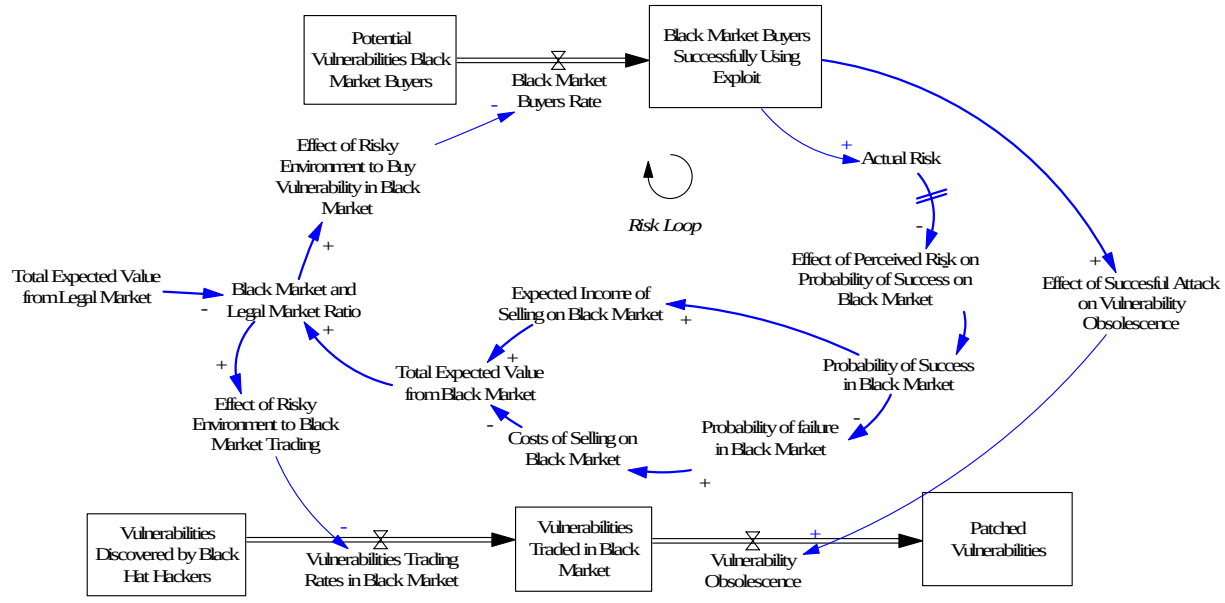

Figure 14
The Risk Sub-Model

Perceived Risk in the model is depicted as a perception delay of hackers toward risk, and the concept shows how a hacker adjusts his perception with actual risk. People need a time to feel that the environment is riskier than he perceived. The model starts with a low perceived risk. The change in actual risk is affected by the feedback from the vulnerability black market buyer growth (See Figure 14). On on hand, the growth of the vulnerabilities black market buyers also increases the likelihood of the net-crime using secret exploits. On the other hand, the more vulnerabilities black market buyers, the higher the risk of hackers for selling the vulnerability on the black market. We capture the above situation through *Effect of Vulnerability Black Market Buyer Growth on Actual Risk*. When the vulnerability black market growth is becoming higher, actual risk is becoming higher as well. And a hacker has to adjust his perceived risk in accordance to the actual risk.

Black Market and Legal Market Ratio is a calculation of the expected value of selling on the black market or the legal market. If EV on the black market and the legal market are equal, the value will be one. In the model, the hacker prefers selling bugs on the black market if the ratio is equal to one. But when EV on the black market is smaller than EV on the legal market, rational hackers will sell it on the legal market. This gradual change in the model occurs as the *Perceived Risk* becomes higher. Higher Perceived risk reduces the probability of success for selling vulnerabilities on the black market. But there is still a 'tolerance' limit from hackers; we called it *Acceptable Risk*. Higher perceived risk is considered safe as long as it is below the acceptable risk. As the value moves higher than acceptable risk, it means that the environment is riskier. Therefore the probability of success to sell vulnerabilities on the black market is becoming smaller. And it means less expected value from the black market. Once a hacker learns that risk increases and probability of success of selling on black market as well as the expected value shrink, he prefers selling a vulnerability on the legal market. This is portrayed by variable *Effect of Risky Environment to Black Market Trading*; this variable will affect the flow of *Vulnerabilities Trading Rates in Black Market.*

5. The Model Behavior

The simulations are conducted to observe the behavior of the system over time under four assumptions: *first*, if there is no intervention for the software security problems (business as usual, or the base run), *second*, when the full disclosure policy is applied, *third*, when the same policy is implemented, but with the legal market implementation, and *fourth*, when the third assumption is carried out with the stronger legal enforcement for committing a cyber crime. The purpose of the preliminary model in this paper is to illustrate the unintended impacts of the intended action that is diverting the policy from accomplishing its goal. Consequently, the aim of this simulation is not yet to test what policies that fit best to meet the software quality and vulnerability black market problem.

The first assumption to be explored is titled "*base-run*". This consists of keeping all parameters fixed throughout the course of the simulation run, i.e., no policy intervention to encounter the software quality and the software vulnerability problems. This scenario is set as a base run for the subsequent policy test.

The second assumption is named "*full-disclosure*". In the base-run, we didn't activate the full disclosure policy structure. We add this structure to represent the policy applied to improve software security.

The third assumption is called "*legal market*". Here we assume that the legal market attracts security researchers to sell vulnerabilities on the legitimate market.

The fourth assumption is called "*risky environment*". Here we consider that legal enforcement is taken into consideration so that there is a possibility for any criminal action related to cyber crime will be punished.

5.1. The Base Run

The base run in the model is intended to show some basic behavior of the main variables in the model. This simulation doesn't incorporate the full disclosure structures and some effects that may affect these initial behaviors.

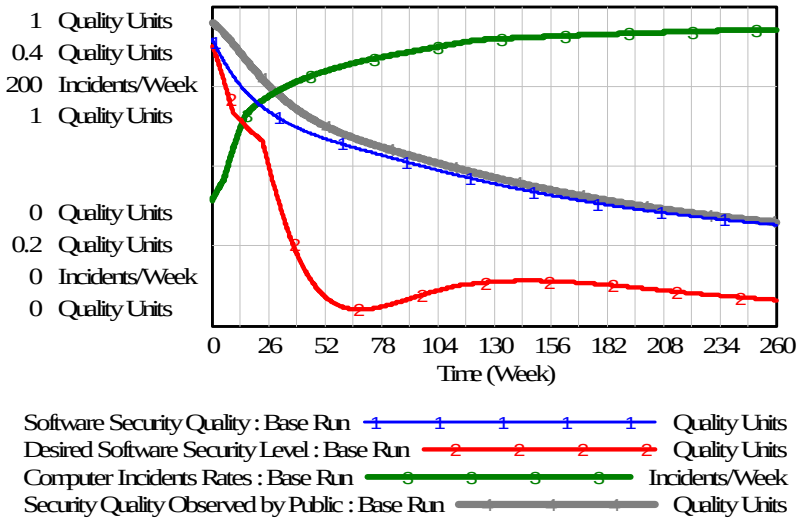

Figure 15
The Base Run of the Desired Software Security Level, Software Quality and Computer Incidents Rates

Figure 15 shows the *Desired Software Security Level*, *Software Security Quality* and *Computer Incident Rates*. The behavior of the base run of those variables has a tendency to decline over time in the variables *Desired Software Security Level* and *Software Security Quality*, while the computer incident rates are likely to rise. As we have already mentioned in the model description, a discrepancy exists between state variable (*Software Security Quality*) and *Desired Software Security Level*, and there is no disturbance in the system, then the system will adjust itself over time to close the gap. However the action to increase the software quality and the "goal" of software vendors (desired security quality) are also affected by other loops such as a *market reward loop* and an *asymmetric information* loop. If these loops are weak, the entire efforts to increase the software security would weaken the system and gradually decay the software security. Figure 15 shows that if no alternative action is taken in response to the software quality problem, then software quality declines. This decline results from an effect of unavailability of incentive to improve security.

On the other hand, the *Computer Incidents Rates* are increasing over time, because of the software weaknesses. The loop generated by computer security breaches is also strengthening the effect to the market rewards that may distract the action to enhance the security.

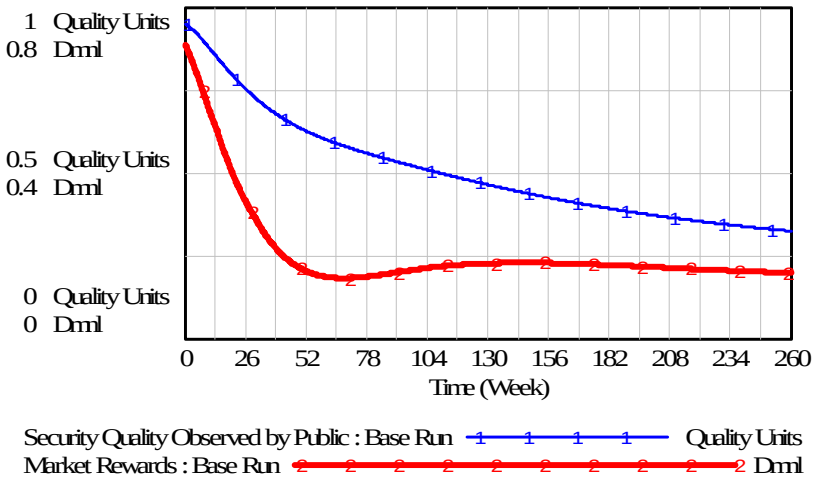

Figure 16
The Base Run of the Software Quality Observed by Public and Market Rewards

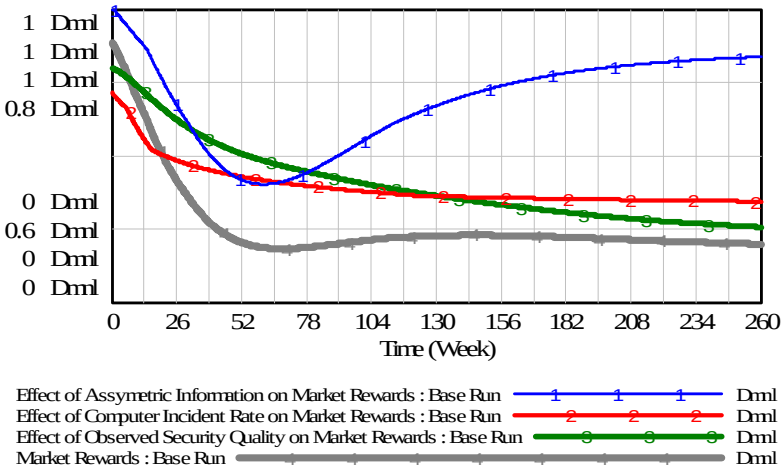

Figure 17
Some Effects Influencing Market Rewards

We can understand the behavior software security over time, by examining the behavior of other variables such as *Security Observed by Public* (Figure 16) that is actually a delay perception on the actual software quality, and *Market Rewards* variables. Three loops that are influencing the market rewards, are simultaneously affecting this graph to have downward trends over time (Figure 17). The characteristic of the reinforcing loops is that they will push upward or weaken the system, depend upon the strength of certain variables. This will explain the behavior in Figure 15.

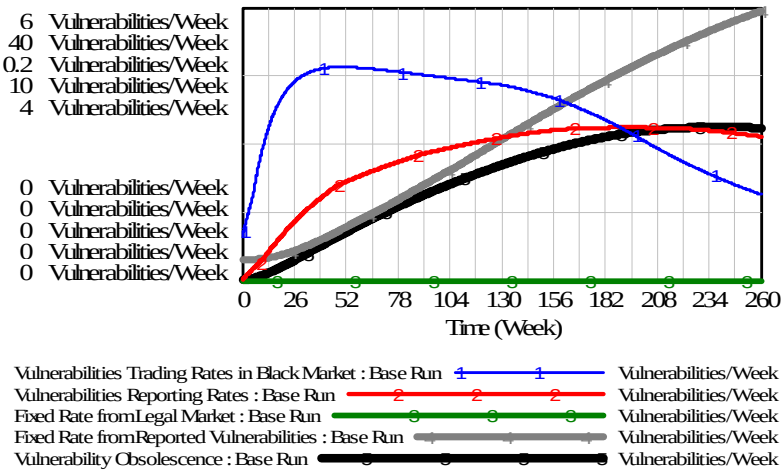

Figure 18
The Flow of Vulnerabilities

In the base run, after vulnerability discovery by black hat and white hat hackers, there are five vulnerability flows to the stock of vulnerability traded in the black market, legal market and reported vulnerabilities, before they are patched. However, we assume that the vulnerability fixed rate from the legal market is zero, because there are no vulnerabilities traded by black hat and white hat hackers on the legal market (see Figure 18).

5.2. The Unintended Consequences of Intended Action

This part will display the behavior of the system, caused by of the intended policy to tackle the software vulnerability problems. This simulation is performed to simulate the action taken by security researchers ("full disclosure") to press the vendors in order to be more cautious when releasing software products and to patch faster when a new vulnerability discovered (See the details in the section 2, *An Overview of the Software Vulnerability Issue*).

The second simulation (*full disclosure, red color*) is conducted to observe the changes of some important variables in the system over time after the policy implementation. It is to observe how severely the unintended effects will influence the overall system.

The third simulation (*legal market, green color*) is to examine the vulnerability discovery distribution between legal market and black market, if the model allows the vulnerability inflow to the legal market, and to see as well the shifts of some variables.

The intention of the fourth simulation (*risky environment*, *grey color*) is to apply a risky environment in the system (for example strong legal enforcement for anyone who commits any type of cyber crime). It is to observe whether the full disclosure policy with risky environment will provide better situation although there are some unintended impacts of the policy.

5.1. Desired Software Quality, Software Security Quality and Computer Incidents

As the disclosure policy (red color) is applied, the "goal" of the software vendors is corrected (Figure 19). The intention to "refine" the software vulnerabilities by releasing patches is bigger because of the feedback from the "full disclosure" loop. This behavior change is happening when the full disclosure and some "unintended consequences" loops are incorporated in the simulation. At the beginning after the pressure, the desired software security level rises, and then gradually it go down. The reason for this, is that there is a positive effect from market rewards that influences as well this variable, beside the effect from the full disclosure pressure. If the market rewards lesser and the effect from full disclosure pressure are bigger, the desired software security level will go down.

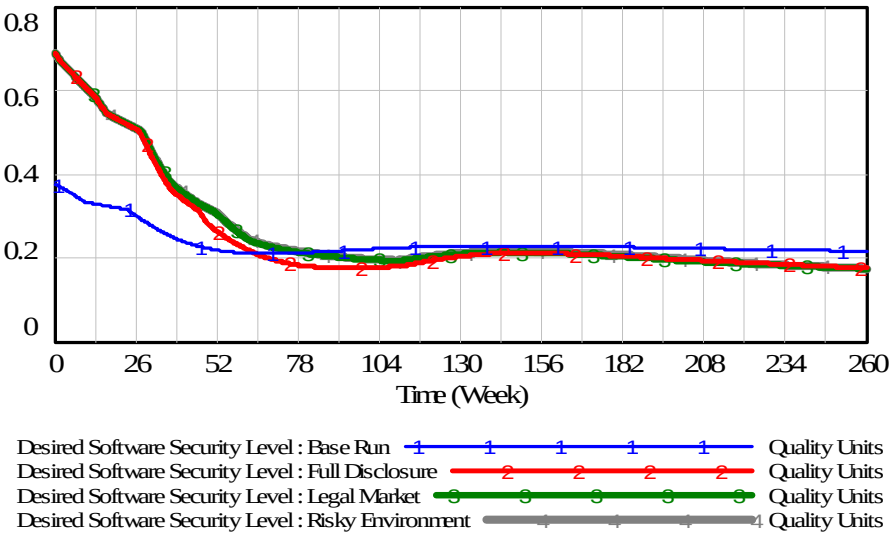

Figure 19
The Simulation Results of the Desired Software Security Level

How does this second simulation affect the software security quality? In the simulation, after the full disclosure, the software quality is rising, but in the long run it is also gradually shrinking, although not as low as the base run simulation. There are some effects influencing the decline in the software quality indirectly, through the incentive to the software quality. Three unintended effects influence the computer incidents rates to become higher. In the simulation, the number of incidents increases around 3-4 times from the base run simulation over time (Figure 21). These unintended effects come from the vulnerability trading loop, rush patch cycle loop and zero day exploit loop. Higher computer incident rates (because of attacks from malicious actors, patching failures or more various type of the net-crime) give a feedback to the market rewards. Finally, these feedbacks lower the incentive (See Figure 20), and in turn weaken the effort to increase the security.

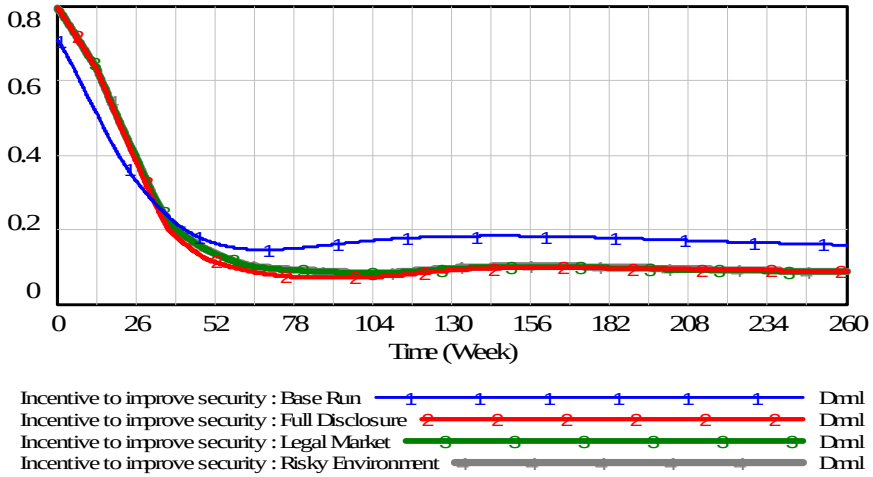

Figure 20
The Simulation Results of the Incentive to Improve Security

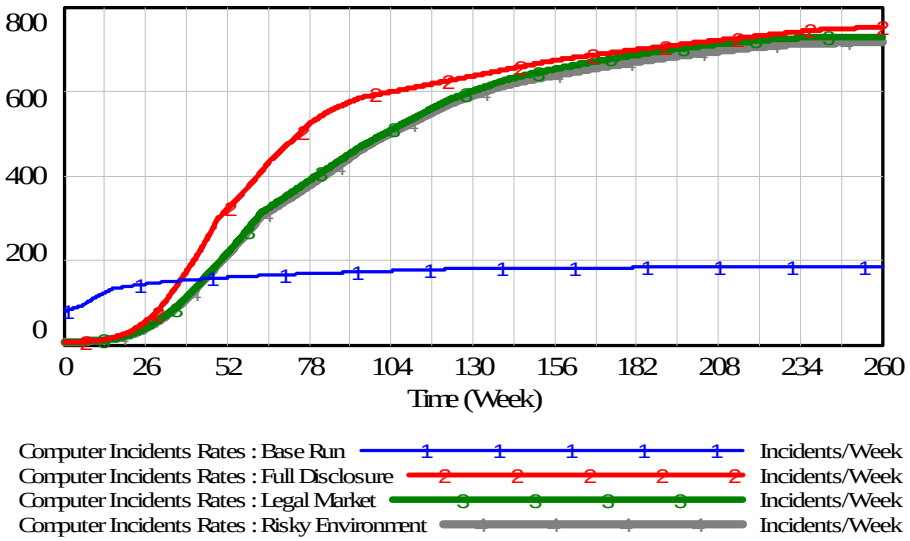

Figure 21
The Simulation Results of the Computer Incidents Rates

5.2. Vulnerabilities Traded in Black Market

We have pointed out that we assume in the model, that vulnerability discovery had already been started in the base run. However the intensity is not too high. There is a feedback loop from the full disclosure pressure to the vulnerability discovery rate, both for white hat hackers and black hat hackers. Basically, these two feedback loops speeds up the discovery rate of the vulnerabilities in the software. Therefore, the vulnerabilities traded in the black market are also rising, after the disclosure policy (red line). As legal markets are opened, the vulnerability black market circulations are decreasing.

In the Vulnerability Trading structure, there are two outflows from the stock of Vulnerabilities Discovered by Black Hat Hackers (See Figure 10) to the stock of Vulnerabilities Traded in Black Market, and Vulnerabilities Traded in Legal Market. Similar

structure is also found in white hat hackers' discoveries. In the base-run, full disclosure and legal market simulations, the risky environment structure is not connected to the main model. In the fourth scenario, risky environment structure is activated by 'switch' parameter. This time, hackers will consider the risk environment.

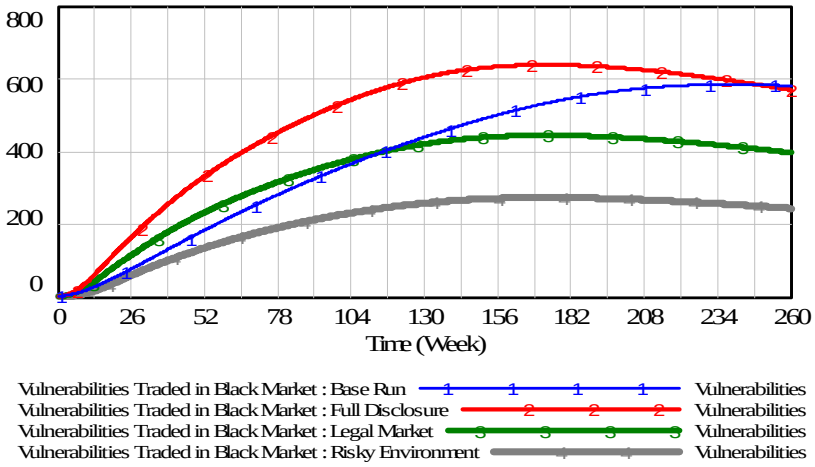

Figure 22
The Simulation Results of the Vulnerabilities Traded in the Black Market

As long as the perceived risk is below the tolerable risk, the behavior might be the same as in the first base run simulation. It implies that hackers still sell vulnerabilities on the black market. But in this simulation, the risky environment has been started since the very beginning. Therefore, as the environment becomes riskier, the vulnerability traded in the black market starts declining faster than the base run. There are two effects responsible for this faster declining behavior: effect of risky environment and effect of higher successful attack on vulnerability obsolescence. However, the former has greater influence than the latter, in this fourth simulation. The depletion in the stock of *Vulnerabilities Traded in Black Market* is caused by lower rates of vulnerability trading rates (**Error! Reference source not found.**). This situation affects the increasing in the stock of *Vulnerabilities Traded in Legal Market* which is caused by the inflow of vulnerabilities traded in legal market (Figure 24).

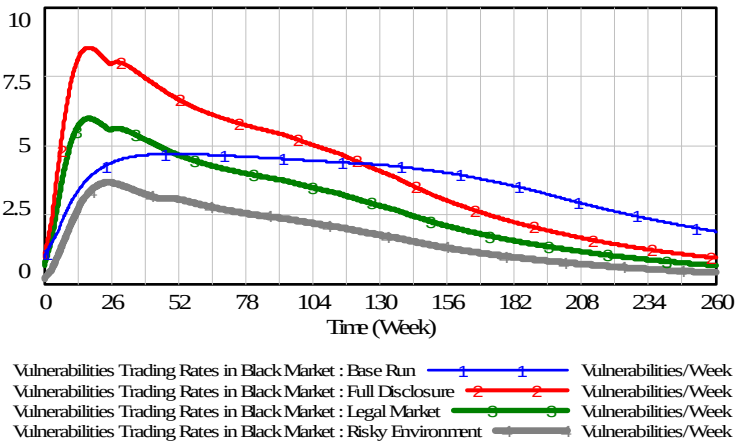

Figure 23
Simulation Results of the Vulnerabilities Trading Rates in Black Market

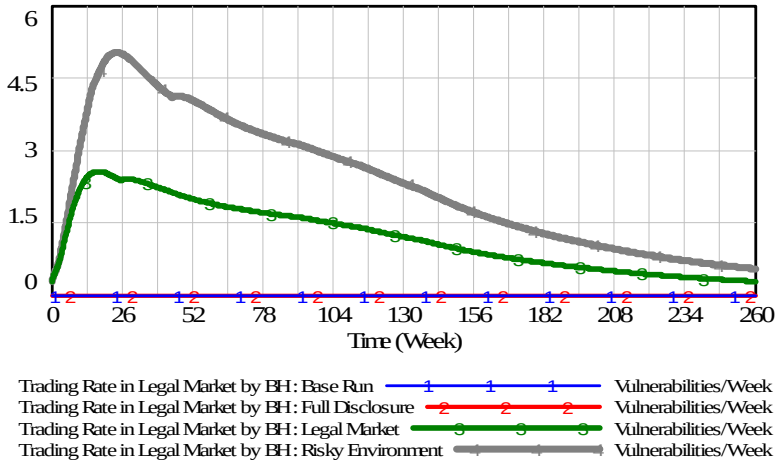

Figure 24
Simulation Results of the Vulnerability Trading Rate in Legal Market

We also can see some changes in the vulnerabilities reporting rates in Figure 26. Full disclosure fastens the white hat discovery rates (red line). The model can also show that the legal market presence decrease the amount of reported vulnerabilities (green line). The risky environment scenario (grey line) doesn't affect the reporting rates, because the risk is only considered by the black hat hackers.

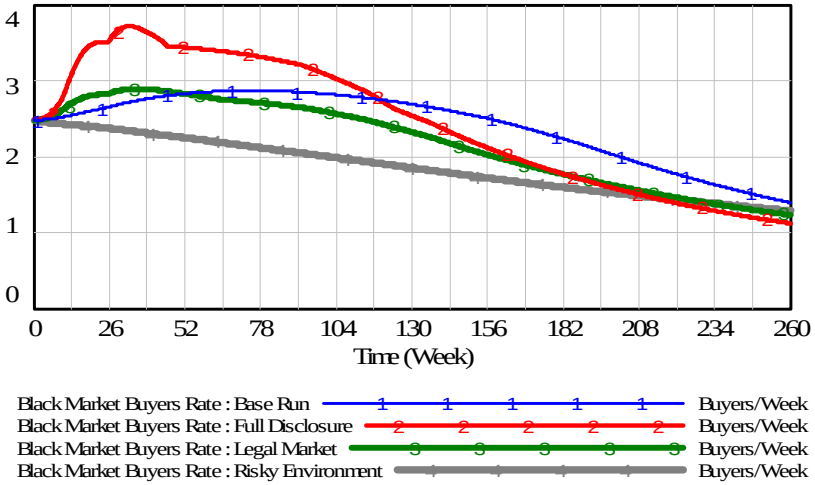

Figure 25
The Simulation Results of the Black Market Buyers

In the model, higher black market trading affects the black market buyers' movement. We also can observe the simulation in Figure 25, as vulnerability black market trading increase in the full disclosure scenario, the number of vulnerability buyers also increase. However, the legal market trading pushes down the black market buyers, and it is even lower in the risky environment scenario. This behavior occurs because of the feedback loop from the declined vulnerability trading in the black market under "legal market" and "risky environment" scenarios.

6. Conclusion and Future Research

This model only shows unintended consequences of a policy intending to mitigate the software vulnerability problem. Our system dynamic approach shows that there are some loops that counteract the effects of the intended policy. Zero-day exploits, rush patch cycle as well as supply and demand on vulnerabilities are further unintended effects of the disclosure policy. The later problem involves the emergence of the vulnerability black market. This market permits 'sellers' (*hackers*) and buyers (criminals/terrorist groups) to trade the secret vulnerability information.

This model also confirms that the vulnerability black market may not grow so fast or might even be contained if the legal system effectively can create a situation where hackers will have higher risk of conducting cyber crime.

This model needs to be supported by more empirical evidence and data. Further validation is also required until we reach a fully validated system dynamics model of the vulnerability black market problem. For the next step of this research, we intend to implements following steps: We intend to gather extensive data for our case as well as to validate further the structure and the behavior of the model. And we will build further the model (in progress—current model described in this paper).

We also want to explore some policy levers relevant for our case. Given the enormous unintended impacts of the full disclosure policy, the idea of "responsible disclosure" develops lately. We could consider this idea as a part of the policy extension for this research. Responsible disclosure is reporting vulnerability directly to the vendor and allowing sufficient time to produce an update, benefits the users and everyone else in the security system by providing the highest quality security update possible. Vendors are given an appropriate amount of time to investigate a security report, reproduce it against all supported platforms, analyze it for variations and similar vulnerabilities in surrounding code and test the resulting update to ensure an appropriate level of quality for mass distribution. Responsible disclosure is considered doesn't increase risk or introducing additional risk as full disclosure can.

In addition we plan to simulate some policies that are pertinent to the software quality improvement and the vulnerability black market issue. Various authors have mentioned the following policies as solutions to overcome the software vulnerability problems: to raise the users' awareness about the quality of the software products (Minasi 2000); to strengthen the legal measurements for anyone who commits cyber-crime (Grannick 2004); to open the market by creating competition among hackers and by providing monetary rewards to discover vulnerabilities can serve to improve the software quality (Böhme 2005, 2006; Schechter 2002; Ozment 2004; Camp and Wolfram 2004). Schechter (2002) for example, proposes that vendors/security firms create a vulnerability market in order to ascertain the cost to break of their system. Schechter's main proposal is to offer an economic approach where a producer would offer rewards at the market price to the first testers (persons or organizations who identify vulnerabilities in return for payment) who inform the producers of new vulnerability in their product. The market price is governed by the competition among those testers. Andy Ozment (2004) formulated the vulnerability market as a bug auction theory based on the "Dutch auction" template that has a key advantage: a reward is always offered, ensuring what vulnerabilities are reported immediately if they are being traded on the black market.

Further steps are to compare some policy runs and to find the best policy to contain the black market and to increase the software quality issue. And finally, we will perform the policy analysis to reach the final conclusion of this problem. At this point, we feel confident that an effective model will provide valuable insights and lessons to learn and to understand the vulnerability black market problems.

7. Literature

Akerlof, G.A. 1970. The Market for "Lemons": Quality Uncertainty and Market Mechanism. *The Quarterly Journal of Economics* 84 (3):488-500.

Anderson, R. 2001. Why Information Security Is Hard, an Economic Perspective. Paper read at 17th Annual Computer Security Applications Conference.

Anderson, Ross, and Tyler Moore. 2006. The Economics of Information Security. *Science* 314:610-613.

Arbaugh W.A., Fithen, W.L and Hugh., J.M. 2000. Windows of Vulnerability: A case Study Analysis. *Computer* 33 (12):52-59.

Böhme, R. 2005. Vulnerability Markets: What Is The Economic Value of a Zero-Day Exploit? . Paper read at 22 C3, at Berlin, Germany.

———. 2006. A Comparison of Market Approaches to Software Vulnerability Disclosure. Paper read at International Conference, ETRICS 2006, LNCS 3995 June 6-9, 2006, at Freiburg, Germany.

Camp, L Jean, and Catherine Wolfram. 2004. Pricing Security, A Market in Vulnerabilities. In *Economics of Information Security*, edited by L. J. Camp and S. Lewis. Boston: Kluwer Academic Publishers.

Cavusoglu, Hasan, Huseyin Cavusoglu, and Srinivasan Raghunathan. 2005. Emerging Issues in Responsible Vulnerability Disclosure. Paper read at 4th Workshop of Economic and Information Security (WEIS), at Cambridge, MA, USA.

CERT/CC. 2007. *Vulnerability Disclosure Policy* 2000 [cited June 10 2007].

Clinard, Marshall B. 1969. *The Black Market: A Study of White Collar Crime*. Montclair, New Jersey: Patterson Smith.

CyberEye. 2007. *CERT's full-disclosure policy is responsible, but mistrust remains* 2001 [cited April, 15 2007]. Available from http://www.gcn.com/state/vol7_no1/tech-report/946-1.html.

Du, W., and A.P.Mathur. 1998. Categorization of Software Errors that Led to Security Breaches Paper read at 21st National Information Systems Security Conference, at Crystal City, Virginia, VA.

Evers, Joris 2007. *Offering a bounty for security bugs* 2007 [cited June, 15 2007]. Available from http://news.com.com/Offering+a+bounty+for+security+bugs/2100-7350_3-5802411.html?tag=sas.email.

Farrow, Rik. 2007. *Vulnerability Disclosure Debate* 2000 [cited June 10 2007]. Available from http://www.spirit.com/Network/net0800.html.

Finjan. 2006. *Web Security Trends Report*. Finjan Malicious Code Research Center Q2 2006 2006 [cited September 20 2006]. Available from http://www.finjan.com/Content.aspx?id=827.

Grannick, Jennifer. 2004. Faking It: Criminal Sanctions and the Cost of Computer Intrustions. Paper read at The 4th Workshop on Economics and Information Security, at Kennedy School of Government, Harvard University.

Greenemeier, Larry. 2006. *The Fear Industry* 2006 [cited August 20 2006]. Available from http://www.informationweek.com/story/showArticle.jhtml?articleID=185301289.

Grimes, Roger A. 2007. *The Full Disclosure Debate* 2005 [cited June 19 2007]. Available from http://www.infoworld.com/article/05/09/30/40OPsecadvise_1.html.

Higgins, Kelly Jackson 2007. *Bucks for Bugs* 2006 [cited April, 10 2007]. Available from http://www.darkreading.com/document.asp?doc_id=99518.

Itzhak. 2006. *Malicious Code for Sale* 2006 [cited August 3 2006]. Available from http://ipcommunications.tmcnet.com/hot-topics/Security/articles/1942-malicious-code-sale.htm.

Jericho. 2007. *Microsoft's Responsible Vulnerability Disclosure, The New Non-Issue* 2001 [cited June 10 2007]. Available from http://attrition.org/security/rant/z/ms-disclose.html.

Kannan, Karthik, and R Telang. 2005. Market for Software Vulnerabilities? Think Again. *Management Science* 51 (5):726-740.

Kaplan, Dan. 2007. *Threats for $ale* 2006 [cited April 10 2007]. Available from http://www.scmagazine.com/us/news/article/556843/threats+ale.

Landesman, Mary. 2007. *Malware Revolution: A Change in Target* 2007 [cited April 10 2007]. Available from http://www.microsoft.com/technet/community/columns/secmgmt/sm0307.mspx.

Landwehr, C.E, A.R. Bull, J.P. Mc. Dermott, and W.S. Choi. 1994. A Taxonomy of Computer Program Security Flaws, with Examples. *ACM Computing Surveys* 26 (3).

Lemos, Robert. 2007. *Mozilla puts bounty on bugs* 2004 [cited June, 10 2007]. Available from http://news.com.com/Mozilla+puts+bounty+on+bugs/2100-1002_3-5293659.html.
Levy, Elias. 2007. *Full Disclosure is a Necessary Evil* 2001 [cited June 10, 2007]. Available from http://www.securityfocus.com/news/238.
Martin, R.A. 2001. Managing Vulnerabilities in Networked Systems. *Computer* (November 2001):32-38.
Middleton, James. 2007. *Coalition Condemns Full Disclosure* 2001 [cited April 10 2007]. Available from http://www.vnunet.com/vnunet/news/2116546/coalition-condemns-full-disclosure.
Miller, Charles. 2007. The Legitimate Vulnerability Market: the Secretive World of 0-day Exploit Sales. Paper read at Workshop on Economics of Information Security, at Pittsburg, USA.
Mimoso. 2007. *The Disclosure Debate Rages* 2001 [cited June 19 2007]. Available from http://www.searchsecurity.techtarget.com/originalContent/.
Minasi, M. 2000. *The Software Conspiracy*. New York: Mc Graw-Hill.
Naraine, Ryan 2007. *Researcher: WMF Exploit Sold Underground for $4,000* 2006 [cited April 10 2007]. Available from http://www.eweek.com/article2/0,1895,1918198,00.asp.
OIS. 2007. *Guidelines for Security Vulnerability Reporting and Response*. Organization for Internet Safety 2004 [cited June 1 2007]. Available from http://www.oisafety.org/guidelines/.
Ozment, A. 2004. Bugs Auctions: Vulnerability Market Reconsidered. Paper read at Workshop of Economics and Information Security (WEIS), at Mineapolis, MN.
Ozment, A, and S Schechter. 2006. Milk or Wine: Does Software Security Improve with Age?" Paper read at The Fifteenth Usenix Security Symposium. July 31 - August 4 2006, at Vancouver, BC, Canada.
Radianti, J, and J.J. Gonzalez. 2006. Toward a Dynamic Modeling of the Vulnerability Black Market. Paper read at Workshop of Economic of Securing Information Infrastructures, 23-24 October 2006, at Washington, D.C.
———. 2007. Understanding Hidden Information Security Threat: The Vulnerability Black Market Paper read at The Fortieth Annual Hawaii International Conference on System Sciences at The Big Island, Hawaii.
Ranum, Marcus. J. 2007. *The Vulnerability Disclosure Game: Are We More Secure?* 2007 [cited June 10 2007]. Available from http://www2.csoonline.com/exclusives/column.html?CID=28072.
Rauch, Jeremy. 2007. *The Future of Vulnerability Disclosure?* 1999 [cited June 19 2007]. Available from http://www.usenix.org/publications/login/1999-11/features/disclosure.html.
Rescola, E. 2005. Is Finding Security Holes a Good Idea? Paper read at The Third Workshop on the Economics of Information Security, 13-14 May 2004, at Minneapolis.
Schechter, S. 2002. How to Buy Better Testing: Using Competition to Get The Most Security and Robustness for Your Dollar. Paper read at Infrastructures Security Conference, October 2002.
Schneier, Bruce. *Full Disclosure and Window of Exposure* 2000b [cited 10 April, 2007. Available from http://www.schneier.com/crypto-gram-0009.html.
———. 2007. *Publicizing Vulnerabilities* 2000c [cited April 10 2007]. Available from http://www.schneier.com/crypto-gram-0002.html.
———. 2007. *Bug Secrecy vs. Full Disclosure* 2001 [cited April 10 2007]. Available from http://news.zdnet.com/2100-9595_22-531066.html.
———. 2006. *Economics and Information Security* 2006 [cited 12 December 2006]. Available from http://www.schneier.com/blog/archives/2006/06/economics_and_i_1.html.
———. 2007. *Schneier: Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'* 2007 [cited June 19 2007]. Available from http://www.schneier.com/essay-146.html.
Seacord, R.C. , and A.D. Householder. 2005. *A Structured Approach to Classifying Security Vulnerabilities*. Carnegie Mellon Software Engineering Institute 2005 [cited December, 22 2005]. Available from http://www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn003.pdf.
Sterman, John D. 2000. *Business Dynamics : Systems Thinking and Modeling for a Complex World*. Boston: Irwin/McGraw-Hill.
Stone, Brad. 2007. *Moscow Company Scrutinizes Computer Code for Flaws*. International Herad Tribune 2007 [cited April 28 2007]. Available from http://www.iht.com/articles/2007/01/29/business/bugs.php.
Sutton, Michael, and Frank Nagle. 2006. Emerging Economic Models for Vulnerability Research. Paper read at The Fifth Workshop on the Economics of Information Security (WEIS), 26-28 June 2006, at Robinson College, University of Cambridge, England.
Tassey, G. 2002. The Economic Impacts of Inadequate Infrastructure for Software Testing: National Institute of Standards and Technology (NIST).
Wahlström, B. 2005. Risk Assessment and Safety Engineering; Applications for Computer Systems. Paper read at the 24th International Conference on Computer Safety, Reability and Security at Fredrikstad, Norway.
Whipp, Matt. 2007. *Black market thrives on vulnerability trading* 2006 [cited April, 10 2007]. Available from http://www.pcpro.co.uk/news/84523/black-market-thrives-on-vulnerability-trading.html.