Vulnerability Markets

What is the economic value of a zero-day exploit?

Rainer Böhme
Technische Universität Dresden · Institute for System Architecture
rainer.boehme@inf.tu-dresden.de

Vulnerabilities are errors in computer systems which can be exploited to breach security mechanisms. Such information can be very valuable as it decides about the success of attack or defense in computer networks. This essay introduces into the economic perspective on computer security and discusses the advantages and drawbacks of different concepts for vulnerability markets, where security-related information can be traded.

1 Economics and security

What's wrong with today's computer and network security? If you were asked by a journalist to answer this question in just one concise sentence, you'd probably talk tech gibberish. But there is a very elegant answer, which is compelling as well: it's all about people and their motivations—in brief, economics.

Researchers in both fields, computer security and economics, recently found that economic theory can well explain why computer security is so difficult despite the presence of sophisticated security technologies. For a good introduction read Ross Anderson's seminal article [2].

Before discussing the effects of vulnerability markets, let me sketch two examples to illustrate how the market fails in providing computer (or network) security. The first example refers to the supply-side for security technology. Its theoretical background is George Akerlof's famous lemon market problem [1]. Akerlof, meanwhile nobel laureate, studied the rules of a market with asymmetrical information between buyer and seller. For instance, the typical buyer of a second hand car cannot distinguish between good offers and bad ones (so-called "lemons"), because—unlike the seller—he does not know the true story of the car. So he is not willing to pay more than the price of a lemon. As a result, used cars in good condition will be undersupplied on the market. The same applies to computer security: security is not visible, it's a trust good. Since the buyer is unable to differentiate secure from insecure products apart, the market price drops to the level for insecure products. Hence, vendors have little incentive to develop sound security technology and rather prefer to invest in more visible gimmicks.

The second example targets to the demand-side of security. Its theoretical roots lie in the popular "tragedy of the commons", another economic theory published by Garrett Hardin [6]. Consider a computer network and the danger of worms and viruses. If the weakest node gets corrupted then the other nodes face a high risk of contagion and consequently face higher expected loss. Therefore the cost of security incidents is distributed among all nodes. On the other hand, if one node decides to invest in security, then all computers in the network benefit, because the now secure node is less likely to cause harm to others from forwarded malicious traffic. In brief, since both risk and benefits are socialized between all nodes, individuals lack the incentive to unilaterally invest in security. They prefer to remain "free riders" waiting for others to pay in their place (who'll never do so, because of the same rationale; see [14] for a rigorous analysis).

To sum it all up, the lemon market suggests that vendors under-supply security to the market, whereas the tragedy of the commons tells us that users demand less security than appropriate. That's what we call a *market failure.*[1]

[1] There are many other approaches trying to tackle computer security problems with economic theory rather than with technology. Among them is the software liability discussion [13], which I omit for the sake of brevity.

Proc. of 22C3, Berlin, Germany, 27–30 December 2005

2 A short typology of vulnerability markets

There are two ways to fix a market failure. At first, regulation—which is least desirable as there are numerous examples where regulation makes things worse. Indeed, good regulation is really difficult since it often implies a trusted third party (TTP) as "social planner", whom to make incorruptible is costly, if not impossible. Second, one can respond to a market failure by establishing new markets with mechanisms that eventually feedback and thus mitigate the problems at their source. In this context, the following overview on vulnerability markets particularly addresses how well the different concepts are suited to solve the security market failure.

Bug challenges

Bug challenges are the simplest and oldest form of vulnerability markets, where the producer offers a monetary reward for reported bugs. There are some real-world examples for bug challenges. Most widely known is Donald E. Knuth's reward of initially 1.28 USD for each bug in his TEX typesetting system, which grows exponentially with the number of years the program is in use. Other examples include the RSA factoring challenge, or the shady SDMI challenge on digital audio watermarking [4].

Depending on the value of the reward, an adversary would have an incentive to report the bug instead of exploiting it or selling it on the black market. The vendor, in turn, can claim that the product is as secure as the amount alloted. This serves not only as a measurable quantity but also as a means to differentiate from competitors—as security becomes measurable, the lemon problem vanishes. Stuart Schechter's thesis [11] on vulnerability markets actually discusses bug challenges in great detail and he coined the term *market price of vulnerability* (MPV) as a metric for security strength.

Although including a monetary measure, I would not call this concept a genuine vulnerability market, because the underlying market mechanism suffers from a number of imperfections. Rather than resulting from a multi-lateral negotiation process, the market price is set by the demand side (i.e., the vendor, who demands vulnerability reports). Even if the vendor decides to increase the reward over time (and reset it after each report), the price quote is not a timely and reliable indicator for the true security of a product. Consider the case where two vulnerabilities are discovered at the same time. A rational agent would "sell" the first one and then wait with the second release until the reward has slowly climbed back to a worthwhile amount. In the meantime, the mechanism fails in aggregating the information about the security. Hence, prudent users would have to stop using the product in critical environments until the reward signals again the desired level of security. Obviously, this is not very realistic.

Bug auctions

Bug auctions offer a different theoretical framework for essentially the same concept as bug challenges. Andy Ozment [9] first formulated bug challenges in the terms of auction theory, in particular as a reverse Dutch auction, or an open first-price ascending auction. This allowed him to draw on a huge body of literature and thus add a number of efficiency enhancements to the original concept. However, the existence of this market type still depends on the initiative of the vendor.[2] On the one hand, this is an advantage because this market type can easily bootstrap—apart from the need of a trusted third party. On the other hand, the cooperation and financial commitment of the vendor makes it very difficult to use this kind of market mechanisms for small vendors and for open source software in general.[3] Moreover, it is questionable whether the rewards offered will ever be high enough to provide an appropriate counterbalance to the assets at risk for software with large installation bases in critical environments, such as finance, healthcare, or government agencies. After all, even Professor Knuth opted for an upper limit to his exponential payoff function for bug hunters in TEX ...

[2] Though not considered in Ozment's work, one can certainly conceive other types of bug auctions independent of the vendor: for example, offering new exploits on E-Bay (see `http://archives.neohapsis.com/archives/dailydave/2005-q2/0308.html`). This sort of blackmailing the vendor and all honest users is definitively not a welfare maximizing strategy. And it does not provide any useful information on security strength when there is no vulnerability for sale.

[3] Mozilla Foundation's "Security Bug Bounty" program is a commendable exception: it rewards each remote vulnerability report with 500 USD.

Vulnerability brokers

Vulnerability brokers are often referred to as "vulnerability sharing circles". These clubs are built around independent organizations (mostly private companies) who offer money for new vulnerability reports, which they circulate within a closed group of subscribers to their security alert service. In the standard model, only good guys are allowed to join the club. The customer bases are said to consist of both vendors, who thus learn about bugs to fix, and corporate users, who want to protect their systems even before a patch becomes available. With annual subscription fees of more than ten times the reward for a vulnerability report, the business model seems so profitable that there are multiple players in the market: iDefense, TippingPoint, Digital Armaments, just to name a few.

If I were to classify CERT (Computer Emergency Response Team) in this typology, I would probably subsume it here. It also acts as a vulnerability broker, albeit on a non-profit basis. It does not pay any reward for reporting vulnerability information and disseminates that information for free. In a recent paper, Karthik Kannan and Rahul Telang [7] compare the social welfare of vulnerability markets (more precisely: vulnerability brokers) and the CERT approach. They conclude that CERT acting as a social planner always performs better than commercial brokers.[4]

Exploit derivatives

Exploit derivatives transfer the mechanism of binary options from finance to computer security. Instead of trading sensitive vulnerability information directly, the market mechanism is build around contracts that pay out a defined sum in case of security events. For instance, consider a contract that pays its owner the sum of 100 EUR on, say, 30 June 2006 if there exists a remote root exploit against a precisely specified version of `ssh` on a defined platform. It is easy to issue this kind of contacts, since you would sell it as a bundle with the inverse contract that pays 100 EUR if the `ssh` program is *not* broken within the maturity. Then, different parties can trade the contracts on a electronic trading platform that matches bid and ask prices, settles the deals, and publishes the price quotes from the order book. The price is freely negotiable and reflects the probability of occurrence of the underlying event at any time.[5] If `ssh` is considered as very secure by the market participants, then the first contract would be traded for, say, 1 EUR, whereas the second for 99 EUR.

The accuracy of the price information depends on the liquidity of the market, hence the number of participants. This market, however, attracts far more groups of participants than the previous market types: software users would demand contracts paying on breaches in order to hedge the risks they are exposed to due to their computer network. The same applies for insurance companies underwriting their customers' cyber-risks. Investors would buy the inverse contract to diversify their portfolios. Software vendors could demand contracts that pay if their software remains secure as a means to signal to their customers that they trust their own system; or contracts that pay if their *competitors'* software breaks. One could even conceive that software vendors use exploit derivatives as part of their compensation schemes to give developers an incentive to secure programming.

Finally, security experts could use the market to capitalize effort in security analyses. If, after a code review, they consider a software as secure, they could buy contracts on the secure state at a higher rate than the market price. Otherwise they buy contracts that pay off on vulnerabilities and afterwards follow their preferred vulnerability disclosure strategy. As any interaction with the market influences the price, the quotes can be used as reliable indicators for security strength. Note that this concept does not require the cooperations of the vendor, and the number of different contracts referring to different pieces of software, versions, localizations, etc., is solely limited by demand.

The concept requires a trusted third party as well to test candidate exploits at the end of each contract and announce the result. However, if the TTP is required to publish the exploit together with the announcement, it becomes verifyable and cannot cheat. The job can also be distributed to a number of TTPs. Hence, the assumptions about the TTP are much gentler in this scenario than in other market types.

[4] Note that the authors come from Carnegie Mellon University, which hosts the headquarters of CERT.

[5] For simplicity, I refrain from discussing interest rates, which can easily be handled by selling the bundles with a deduction equivalent to the return from a reference interest rate over the remaining maturity.

Cyber-insurance

Cyber-insurance is among the oldest proposals for market mechanisms to overcome the security market failure (see [5, 8, 12, 13, 15]). The logic that cures the market failure goes as follows: end users demand insurance against financial losses from information security breaches and insurance companies sell this kind of coverage after a security audit. The premium is assumed to be adjusted by the individual risk, which depends on the IT systems in use and the security mechanisms in place. Therefore it would be costly to buy insurance coverage for less secure software. This gives users an incentive to invest in security technology. One would even raise the willingness to pay more for secure products if—in the long run—the total cost of ownership including insurance premiums is below the expenses for a less secure product.

However, despite the presence of potent insurance companies, there is surprisingly little supply for cyber-insurance contracts. One of the reasons for this under-supply is the fact that insurance companies hesitate in underwriting cyber-risks, because the losses from virus outbreaks and worms are highly correlated globally. This concentration of risk is contrary to the insurance principle of portfolio balancing (see [3]). Apart from the fear of "cyber-hurricanes", there are other operational obstacles, such as the difficulty to substantiate claims, the intangible nature of cyber-assets, and unclear legal grounds.

3 Boon or bane?

We have seen that quite a number of possible instances for vulnerability markets is conceivable. So the question to answer is whether we are better off with or without them—frankly, shall we hype or fight them? It is obvious that any claim of a universal answer to this question cannot be serious, so the remainder of this essay tries to collect arguments for and against the markets, and in particular the pros and cons between different market types.

To judge the markets it is useful to define a set of criteria. An ideal vulnerability market fulfills three functions: first, the **information function** refers to the ability to use market prices as forward-looking indicators of security properties. This is important to counter the lemon effect. Second, the **incentive function** allows monetary compensation for security research and development. It motivates firms and individuals to give security a higher priority. Third, the **risk balancing function** means, that the market provides instruments to hedge against large information security risks. This is important to mitigate the financial impact of (occasional) security breaches, which may help firms to survive a virus attack rather than filing for bankruptcy with all its adverse social and economic consequences. Orthogonal to these functions, market **efficiency** is a criterion under which I subsume other desirable properties, such as low transaction costs, liquidity, transparency (public quotes, fair rules), and accountability (low counterparty risk). Table 1 compares the different types of vulnerability markets along the defined criteria.

Table 1: Comparison of Vulnerability Markets

	Information	Incentives	Risk balancing	Efficiency
Bug challenges ...	−	+	−−	−
Bug auctions	−	+	−−	−
Vuln. brokers	−−	±	−−	−−
Exploit derivatives	++	+	+	+
Cyber-insurance ..	+	++	++	−

Both bug challenges and bug auctions provide vulnerability hunters with an incentive to report and also give developers a motivation to take security more seriously. However, there is no possibility for risk balancing at all, and the information obtained from the market price is only a lower bound, which fails to be accurate when vulnerabilities are reported frequently. As to efficiency, the vendor has to bear most of the burden and the existence of a market depends on his cooperation. Vulnerability brokers (excluding CERT) do worse because they create questionable incentives (i.e., for bad boys to join the circle) and deliver no information to the public at all. It appears that exploit derivatives and cyber-insurance are both acceptable, with exploit derivatives having an advantage as timely indicator whereas cyber-insurance gets a

deduction in efficiency due to the presumably high transaction costs. What's more, both concepts complement one another. Please note the limitations of this qualitative assessment, which should be regarded as a starting point for discussion and exchange of views.

There is also room for more general critiques on the market approach. One might question if vulnerability hunting actually leads to more secure products (see [10] for a discussion and evidence for vulnerability hunting), so why bother putting market incentives in place for something allegedly useless? Moreover, we all know that markets tend to err in the short term—but it's still very difficult to outpace existing markets in the long run. Therefore we need to assess the harm a "vulnerability market bubble" potentially causes, and weight it against the welfare gains from better information, more secure products, and the possibility to hedge information security risks. Finally, there remains to be written a chapter on conflicts of interest.

To conclude, we have seen that economic models can well explain the computer security dilemma and that vulnerability markets are a way to tackle the problem. However, there is not one "vulnerability market" but rather a family of different concepts. After regarding their individual mechanisms, it becomes evident that the market types close to reality, namely bug challenges and vulnerability brokers, are not the best possible solutions.

*

Back to the journalist's question at the beginning, how would you answer in, say, 20 years when computer security is so important that it has entirely melted into finance? You would probably mention the New York Stock Exchange having closed with a 5.23 % decline in the Standard & Poor's composite kernel hardness index. So it's only a matter of time when the next big kernel exploit hits the cyber-world ...

The author gratefully acknowledges the valuable comments he received from Thorsten Holz and Gaurav Kataria.

[1] George A. Akerlof. The market for 'lemons': Quality, uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84:488–500, 1970.

[2] Ross J. Anderson. Why information security is hard – an economic perspective, 2001. Online `http://www.cl.cam.ac.uk/~rja14/econsec.html`.

[3] Rainer Böhme. Cyber-insurance revisited. In *Workshop on the Economics of Information Security (WEIS)*, Cambridge, MA, 2005. Online `http://infosecon.net/workshop/pdf/15.pdf`.

[4] Scott Craver et al. Reading between the lines: Lessons from the SDMI challenge. In *Proc. of the 10th USENIX Security Symposium*, Washington, DC, 2001. USENIX Association. Online `http://www.usenix.org/events/sec01/craver.pdf`.

[5] Lawrence A. Gordon, Martin P. Loeb, and Tashfeen Sohail. A framework for using insurance for cyber-risk management. *Communications of the ACM*, 46(3):81–85, 2003.

[6] Garrett Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[7] Karthik Kannan and Rahul Telang. An economic analysis of markets for software vulnerabilities. In *Workshop of Economics and Information Security (WEIS)*, Minneapolis, MN, 2004. Online `http://www.dtc.umn.edu/weis2004/kannan-telang.pdf`.

[8] Jay P. Kesan, Ruperto P. Majuca, and William J. Yurcik. The economic case for cyberinsurance. In *Workshop on the Economics of Information Security (WEIS)*, Cambridge, MA, 2005. Online `http://infosecon.net/workshop/pdf/42.pdf`.

[9] Andy Ozment. Bug auctions: Vulnerability markets reconsidered. In *Workshop of Economics and Information Security (WEIS)*, Minneapolis, MN, 2004. Online `http://www.dtc.umn.edu/weis2004/ozment.pdf`.

[10] Andy Ozment. The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In *Workshop on the Economics of Information Security (WEIS)*, Cambridge, MA, 2005. Online `http://infosecon.net/workshop/pdf/10.pdf`.

[11] Stuart E. Schechter. *Computer Security Strength & Risk: A Quantitative Approach*. PhD thesis, Harvard University, Cambridge, MA, 2004.

[12] Bruce Schneier. Hacking the business climate for network security. *IEEE Computer*, pages 87–89, April 2004.

[13] Hal R. Varian. Managing online security risks. *New York Times*, June 1st, 2000. Online `http://www.nytimes.com/library/financial/columns/060100econ-scene.html`.

[14] Hal R. Varian. System reliability and free riding. In *Workshop on Economics and Information Security (WEIS)*, Berkeley, CA, 2002. Online `http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurit%y/`.

[15] William Yurcik and David Doss. Cyberinsurance: A market solution to the internet security market failure. In *Workshop on Economics and Information Security (WEIS)*, Berkeley, CA, 2002. Online `http://www.sims.berkeley.edu/resources/affiliates/workshops/econsecurit%y/`.