Pricing Security

L. Jean Camp
L213 79 JFK St.
Harvard University
Cambridge MA, 02138
Jean_Camp@harvard.edu
617-496-6331

Catherine Wolfram
Littauer 209, Department of Economics.
Harvard University
Cambridge MA, 02138
cwolfram@harvard.edu
617-495-9293

The Internet, and the larger information infrastructure, are not secure (e. g. , National Research Council, 1996). Well known vulnerabilities continue to be exploited long after patches are available. Today too many organizations discover security the day after their Web pages have been rewritten by intruders interesting in attracting attention. Thus the only ubiquitous testing of Internet security is done by egocentric hackers. The information infrastructure is the only infrastructure subject primarily to destructive testing. Those vulnerabilities that are well documented, with free patches, continue to exist on the Internet(Farmer, 1999).

An alternative solution not previously considered is to create a market for the detection of security failures whereby those who have neglect to secure their networks, products, and machines can suffer the consequences according to formal pricing mechanisms rather than destructive incidents. A model for pricing security as an externality can be found in studies of the pricing of pollutants.

The foundation of congestion and environmental economics supports building the pricing of security vulnerabilities as a function of a number of factors. These factors determine the risk, and thus the price, of security vulnerability. We would like to focus on the issues of defining the good and jumpstarting a market in the information survivability workshop.

Security as an Externality

Economists define externalities as instances where an individual or firm's actions have economic consequences for others for which there is no compensation. One important distinction is between positive and negative externalities. Instances of the latter are most commonly discussed, such as the environmental pollution caused by a plant, which may have impacts on the value of neighboring homes. Important examples of positive externalities are so common in communications networks that there is a class of "network externalities. Coordination on a standard is a classic example.

A more useful analogy in the case of computer security is automotive security. When Lojack, the auto theft response system, is introduced in a city, auto theft in general goes down because Lojack is designed so that thieves can't tell whether or not a car has it installed(Ayres, Levitt, & Steven, 1998) . In other words, people who buy Lojack are providing positive externalities to other car owners in the city.

The basic conclusion is that, absent government intervention or other solutions to internalize the externalities, negative externalities are over-provided and positive externalities are under-provided. In our case, to the extent investments in computer security create positive externalities, too little will be provided. There are also several corollaries to the basic conclusion. For one, products that generate security problems will be under-priced. Also, the incentives to invest in learning more about security and taking steps to prevent incidents will be insufficient.

Several attributes of computer security suggest that it is an externality. Most importantly, the lack of security on one machine can cause adverse effects on another. The most obvious example of this is from electronic commerce, where credit card numbers stolen from machines lacking security are used to commit fraud at other sites.

Three common ways in which security from one system harm another are shared trust, increased resources, and the ability for the attacker to confuse the trail. Shared trust is a problem when a system is trusted by another, so the subversion of one machine allows the subversion of another. (Unix machines have lists of trusted machines in . rhosts files). A second less obvious shared trust problem is when a user keeps on one machine his or her password and account information for another. The use of cookies to save passwords as well as state has made this practice extremely common.

The second issue, increased resources, refers to the fact that attackers can increase resources for attacks by subverting multiple machines. This is most obviously useful in brute force attacks, for example in

decryption or in a denial of service attack. Using multiple machines makes a denial of service attack easier to implement, since such attacks may depend on overwhelming the target machine.

Third, subverting multiple machines makes it difficult to trace an attack from its source. When taking a circuitous route an attacker can hide his or her tracks in the adulterated log files of multiple machines. Clearly this allows the attacker to remain hidden from law enforcement and continue to launch attacks.

Because security is an externality the pricing of software and hardware does not reflect the possibility of and the extent of the damages from security failures associated with the item.

Externalities and public goods are often discussed in the same breath (or at least in the same sections of textbooks). They are two similar categories of market failures. A common example of a public good is national security, and it might be tempting to think of the analogies between national security and computer security. National security, and public goods in general, are generally single, indivisible goods. (A pure public good is something which is both non-rival – my use of it doesn't effect yours' – and non-excludable– once the good is produced, it is hard to exclude people from using it.)Computer security, by comparison, is the sum of a number of individual firms' or peoples' decisions. It is important to distinguish computer security from national security (i. e. externalities from public goods) because the solutions to public goods problem and to externalities differ. The government usually handles the production of public goods, whereas there are a number of examples where simple interventions by the government have created a more efficient private market such that trades between private economic parties better reflect the presence of externalities. A better analogy for computer security is pollution, and a number of market-based approaches have recently been implemented to help achieve a more efficient level of pollution abatement.

Defining the Good: A Vulnerability

One critical point to decide in developing a market for security is, what is the good in question? Are we discussing the provision of more security of the provision of fewer vulnerabilities? Consider that an increase in security can include changes in institutional practices, upgrading platforms, increasing training, removing or adding services, or the removal of vulnerabilities. In order for the market to function it must be targeted on a definable discrete good. We propose that this good, or item that can have a deterministic value, is the vulnerability.

What is a vulnerability? What is a feature? In order to price vulnerabilities one must classify them. Before classification must come definition. A formal definition from computer security is that a vulnerability is an error that enables unauthorized access. This definition does not clarify the issue of feature versus vulnerability. An error may be an error in judgement and this definition would still hold. Thus we offer the following.

A vulnerability can be defined as follows:

- A technical flaw allowing unauthorized access or use,
- Where the relationship between the flaw and access allowed is clear,
- Which has been documented to have been used to subvert a machine,

For example, the ability to send and receive email can be used for social engineering to obtain passwords. Using email to obtain passwords has been documented to be a useful attack. There is no correcting code or technical procedure available to end social engineering. The sending and receiving of email may be an error in judgement -- one can forbid email from passing through firewalls -- but it not a technical error.

Given the definitions of vulnerabilities classify security vulnerabilities, consider the available taxonomies to determine the best fit.

Classifying Computer Security Failures

Any taxonomy that is used to price security failures should be deterministic and complete. No security failure should be left unclassified and no security failure should fall into more than one classification. Given this fundamental limitation now review security taxonomies developed by experts in the field.

The most basic classification scheme for pricing is the original security classification scheme of top secret, secret, and sensitive. This security classification applies to the files that are the subjects of

computer security. That is, this classification is based on the material to be protected rather than the mechanisms used for protection. Our entire focus is on the mechanisms for protection so this classification method, and others based upon classification of documents according to content ,are not useful.

Consider three attempts to classify security failures, (Aslam, Krsul, & Spafford, 1996), (Landwhere, et al. , 1993), (Howard, 1997). How applicable these attempts are to pricing?

In his analysis of security incidents on the Internet, Howard focuses exclusively on incidents. An incident is an attack or series of attacks using the same set of tools by a single set of attackers. An attack may begin with a single subverted account and subvert multiple sites over time. Howard focuses upon the exploitation of vulnerabilities rather than the existence of vulnerabilities.

A result of our work being on those extant but not necessarily exploited vulnerabilities is that any work which focuses on motivation is inappropriate. Clearly the attack is exactly what this work on pricing vulnerabilities would prevent. Thus while complete and unambiguous the taxonomy addresses variables that are not available for this work. Motivation is also the reason that the work by Landwhere et. al. does not apply.

The work of Aslam, Krsul, & Spafford was an effort to classify security weaknesses and thus is the closet in spirit to this effort. There are four basic types of faults in this classification.

Synchronization faults and condition validation errors are classified as coding faults. Coding faults are faults that are included in the code. These result from errors in software construction.

Configuration errors and environmental faults subcategories of emergent faults. Emergent faults can occur when the software performs to specification but the result, when installed in specific environment, is still a security vulnerability.

Allocating Property Rights

For the purpose of pricing vulnerabilities to increase security rights could be assigned two ways. First, computer owners and operators could be charged for having vulnerabilities and coders could be charged for creating them. In the case of shrink-wrapped software charging coders would be effective. However, in the critical arena of free software identifying contributions and charging effectively would require very high transaction costs in terms of overhead and organization.

The examples of freeware, shareware, free software and other downloaded software of potentially amorphous ownership illustrates that there would in some cases be high transactions cos. It follows that in assigning the property right there is a legitimate concern about the equilibrium. Requiring payment to upgrade would create an incentive for initial low levels of security. Requiring payments to maintain vulnerabilities would have an opposite incentive: to avoid vulnerabilities.

We present here an alternative. We argue that this is effective in many ways but not that it is the only possible configuration. We suggest that every machine, (client, server regardless) should be allocated certain initial properties, and a set of vulnerability credits. In pollution the issues of jump starting trading were resolved by providing to each utility a certain number of pollution credits based upon the total output of the utility (with explicit bias against nuclear generation).

With vulnerabilities a comparative approach can be used, by providing vulnerability credits appropriately to each entity using machines. However, how to distinguish the entities and even the machines and define appropriate is the essence of jump starting trade. Here we offer only an alternative. Note that the division of pollution allowances under the Clean Air Amendments (Schmalensee, Joskow, Ellerman, Montero, & Bailey,1998,) was at best highly political yet the resulting market still functions.

There are many variables that can be used to determine how many 'machines' are run by a company. Counting boxes is not a particularly clever approach since boxes have different numbers of processors and different processing power. One web site may have a small fraction of a server, or tens of servers accessing heavy backend hardware.

Counting processing power may then appear reasonable; however, clearly a video processor inserted into a PC does not make the machine the equivalent of two Pentium III class machines. There is at least a common and recognizable metric in processing power that would recognize that supercomputers are not equivalent to aging dedicated printer servers. Thus we would advocate considering processing power.

Without having home users as part of the market the ability of users to respond to security failures in the computer market as a whole will suffer. By including home users, a successful market for effectively

blackmailing users who do not know how to alter their machines will be created. However, we believe that an equivalent market for upgrading home machines would then arise.

Jump Starting Trading

For pricing to be valid there must beheld a liquid market for the goods over which you have defined property rights. In the case of pollution building such a market has proven possible but not trivial (Schmalensee, Joskow, Ellerman , Montero, & Bailey, 1998).

We recognize that in terms of politics this is the most problematic set of questions: who decides? However given the role of computer security is to define questions of how to organize decision-making power over electronic resources we go so far as to offer a set of alternatives. Here are the decision-making roles that must be fulfilled:

- creation/validation of vulnerability credits
- price of a vulnerability credit
- organizational compliance , i. e. the vulnerability/credit balance
- payment after an imbalance has been identified

For the last three there is no readily apparent reason for any but the market itself to decide. After initial allocation of vulnerabilities the market can determine the price, given that the discoverer of a vulnerability can demand remediation or payment. Any entity that is discovered to have a vulnerability and no credit has a finite window in which to either correct its system or purchase a vulnerability. In either case, an initial payment will be required to the entity discovering the vulnerability that creates the imbalance.

However, the creation of vulnerability credits is effectively the creation of money. One alternative is to have the Federal Government validate and create vulnerability credits. A second is to create a corporation for the process. The Domain Name System is now being developed under these auspices, with the Internet Corporation for the Assignment of Names and Numbers assigning IP addresses and coordinating assignment of domain names. A third is to license existing companies, perhaps those in the business of creating processing power, to create credits and distribute credits.

Conclusions

In this paper we have introduced a mechanism for creating a market for security vulnerabilities based on vulnerability credits which can be exploited. We have discussed a first cut at a market for vulnerability credits. We note that there exist many mechanisms for implementing such a scheme in the literature of mechanism for Internet commerce.

Companies that do nothing but charge others for security violations could exist if this approach were adopted as economic policy. We would argue that this would be a positive outcome. The volunteer hackers, who destroy nothing but fleeting web sites could be replaced by entrepreneurs with activities which are legal, sustainable, and for the common good.

References

Aslam, Krsul, and Spafford. (1996) “A Taxonomy of Security Vulnerabilities”, *Proceedings of the 19th National Information Systems Security Conference*, pages 551-560, Baltimore, Maryland, October.

Ayres, Levitt, & Steven D. 1998, "Measuring Positive Externalities from Unobservable Victim Precaution: An Empirical Analysis of Lojack", *The Quarterly Journal of Economics*,V113, 43-77.

Farmer, 1999, Security Survey of Key Internet Hosts & Various Semi-Relevant Reflection, http://www.fish.com/survey/

Howard, J., 1997, *An Analysis Of Security Incidents On The Internet 1989 - 199*5,Ph. D. dissertation, Carnegie Mellon University. Available at http://www.cert.org/research/JHThesis/Start. html.

Landwhere, Bull, McDermott & Choi, 1994, "A Taxonomy of Computer Program Security Flaws, with Examples, *ACM Computing Surveys*, Vol. 26, Sept. pp. 3. -39.

National Research Council, 1996, *Cryptography’s Role in Securing the Information Society*, National Academy Press, Washington, DC.

Schmalensee, R., Joskow, L., Ellerman, A. D., Montero,J. P. , & Bailey, E. M. , 1998,"An Interim Evaluation of Sulfur Dioxide Emissions Trading", Journal of Economic Perspectives.V12(3)53-68.