Pricing Security

L. Jean Camp
L213 79 JFK St.
Harvard University
Cambridge MA

Catherine Wolfram
Department of Economics
University of Berkeley
Berkeley, CA

Abstract

We argue that provision of computer security in a networked environment is an externality and subject to market failures. However, regulatory regimes or a pricing schemes can causes parties to internalize the externalities and provide more security. The current mechanisms for dealing with security are security analysis firms; publications of vulnerabilities; the provision of emergency assistance through incident response teams; and the option of seeking civil redress through the courts. The overall effectiveness of these mechanisms is questionable. The foundation of environmental economics supports building a market as a solution to the problem of widespread vulnerabilities. In this work we propose a market for vulnerability credits.

This paper is a first step to developing a pricing scheme for vulnerabilities to increase infrastructure security. We begin by arguing that security is an externality and one which could be priced. We examine security taxonomies in terms of their usefulness for pricing security vulnerabilities. We discuss the parallel with pricing pollution. We address the issue of jump-starting the market. Regulatory mechanisms for collection are not extensively addressed, although pricing without payment is meaningless, the problem must be parsed to be solvable.

Introduction

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 interested 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.

Certainly the controls on the export of cryptography has played a significant role. Other fundamentally flawed policies, such as threats to prohibit basic research in the name of intellectual property, are contributors as well. Yet while these policies do play a part, they are not responsible for the entire situation. 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.

We briefly discuss the intellectual foundations on which this concept rests: pricing pollution. Pollution is similar in that there is no inherently obvious price. There is a value for pollution but it is difficult to get the parties together to transact and set a market price for pollution. For there to be production there must be some pollution; for there to be connectivity and interaction there must be some vulnerabilities. Thus in both cases the there are issues of definition: Is it a feature or a bug? Is it a toxic pollutant or a necessary part of the product? Congestion research is informative because it requires coordinated technical cooperation across jurisdictional boundaries.

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 a security vulnerability. The factors may include expected severity of damage, delay in response time, and costs of correction. We discuss how applicable each factor is or is not in the case of vulnerabilities.

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. For instance, the simple act of installing telephone service to one additional customer creates positive externalities on everyone on the telephone network because they can now each reach one additional person. The literature on network externalities goes on to describe a number of the consequences that network externalities have on firms competing to provide products which have network externalities. Coordination on a standard is a classic example.

Pollution is an example of a negative externality. For example, because they do not internalize the costs they are inflicting on homeowners, polluters will go on producing pollution until the costs *to the polluter* outweigh the benefits. If homeowners could pay the firms not to pollute, however, or if they could extract payment from the firms for every ounce of pollution, the firms' costs of polluting would go up (in the former case, their benefits from *not* polluting would go up) so there would be less pollution.

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. However, this problem preceded electronic commerce, although with the growth of electronic commerce the stakes may be greater. There could also be indirect costs associated with this form of security breach if credit card theft at one site reduces consumers' willingness to engage in electronic commerce 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 authentication information as well as states 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. The last two points suggest that costs to hackers fall with the number of machines (and so the difference between the benefits of hacking and the costs increases), similar to the way in which benefits to phone users increase with the number of other phones on the network.

A fourth point is the indirect effect security breaches have on users' willingness to transact over the network. For instance, consumers may be less willing to use the Internet for e-commerce if they hear of incidents of credit card theft. This is a rational response if there is no way for consumers to distinguish security levels of different sites.

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 affect 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.

In the following section we will briefly discuss common solutions for externalities. Each of the common solutions is currently being tried and none has been found adequate.

Past Solutions

There are several ways in which a government body can address externalities: command and control regulation, the provision of information, support for the market and governmental provision of the good. All of these have been attempted and in fact are continuing. In this section we discuss various attempts to address the issue of network security. Although none of these have explicit in their motivation that security is an externality all of these have the concern that computer security is not adequately provided by the market.

Information Provision

The Federal Government encourages information provision through subsidy of incident response teams, computer security research, and the direct provision of information through the creation of standards. All of these are discussed in the section on subsidies below.

Coordinating Information

The President's Commission on Critical Infrastructure Protection (Critical Foundations, 1997) has focused on information sharing. The proposals to share information include a suggested exemption from the Freedom of Information Act. Thus, the few selected players would have greater information but the majority of computer users would not

only have no additional information but would also be barred from seeking Federal information.

The set of proposals for best practices is reasonable for a corporate intranet but ill-suited to small businesses, home users, or electronic commerce sites. For example, authenticating every user is not appropriate for browsing customers. Small businesses may be unable to conduct security training for every employee, and certainly cannot establish in-house incident response teams. The PCCIP views the critical elements of the infrastructure as being large intranets, and does not address the many home users, small businesses, academics, and hobbyists.

Classification

The Department of Defense began a decade-long experiment in classifying trustworthy components in 1985. The networks are to be classified by existence of features (e.g. use of passwords), design, and implementation methodology. Together these factors are assumed to illustrate the overall level of security (Department of Defense, 1985). Although this taxonomy is widely taught in introductory computer security classes for the concepts which it embodies, this effort has failed. There are no major computer systems marketed with a Department of Defense rating.

The basic concepts embodied in the Department of Defense rating continues to be popular, with systems built logically from a trusted computing base. However the ratings themselves and the mechanisms are widely ignored by the market.

Setting Standards

The National Institute of Standards sets cryptographic standards. The adoption rate of particular Federal Information Processing Standards (FIPS) has varied dramatically. The Data Encryption Standard (DES)as described in FIPS 46 (National Bureau of Standards, 1977) has been widely implemented. DES is the most widely used encryption algorithm in the world. Alternatively the "Clipper" standard, (National Institute of Standards and Technology, 1994) has been subject to wide objections.

Standards setting is a manner of providing information. Selected standards are examined by the Federal Government and pronounced trustworthy. The original Clipper FIPS was the first information processing standard based on a classified algorithm . Thus it provided limited information. In contrast DES was developed with IBM with the result being an open standard. Information provision in terms of standards-setting has improved network security, but has not proven adequate to address all security vulnerabilities.

Subsidies

The provision of information security is subsidized by the government in three ways: support for incident response teams (e.g. provision of the good), purchase of secure technologies, and support for research in computer security.

A clear subsidy of computer security is the provision of incident response teams. Incident response teams assist in detecting, preventing, defeating, and recovering from attacks on

computer systems. Incident response teams provide service free or at subsidized rates. The Federal Government completely funds the Computer Incident Advisory Capability or CIAC (http://ciac.llnl.gov/).

The Computer Emergency Response Team (CERT) was initially a fully federally funded operation. CERT competes for federal research funds, and the organization's long term goal is to be self-supporting. Despite the high quality of services and strong confidentiality. CERT has not yet met this goal.

The government also provides a market for computer security technologies. In particular, the Department of Defense and the Department of Energy both provide a certain market for computer security technology. In addition federally funded R&D centers, e.g. MITRE and RAND, and DoD contractors and suppliers also add to the market for cutting-edge security technologies.

Arguably the support for research in computer security reflects the fact that research is an externality. Computer security can also be seen as a subcategory of national defense, which is a classic public good. Regardless, research support for computer security has proven more effective in finding weaknesses and resulting responses, and less successful in disseminating the results in terms of widespread adoption of optimal security practices.

Defining the Good: A Vulnerability

Thus we have argued that security is an externality. Federal efforts other than creating an explicit market have proven inadequate. Now we move forward to the first step of creating a market, defining the good.

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 which can have a deterministic value, is the vulnerability.

Consider which vulnerabilities are subject to pricing. Those vulnerabilities which have been exploited have been priced in that the destructive use of the vulnerability has placed a cost on the institution subject to the loss. However, the externalities discussed above (shared trust, additional resources and preventing detection) have not been included in this price.

Another issues is determining -- 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 which enables unauthorized access. This definition does not clarify the issue of feature versus vulnerability. An error may be an error in judgment 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. Social engineering is not inherently a technical problem. The sending and receiving of email may be an error in judgment -- one can forbid email from passing through firewalls. Yet the relationship between sending email and obtaining unauthorized access is not clear. Is it allowing passwords to be transmitted? Is it allowing bad judgment? How is this a technical flaw? The option of allowing email to be sent and received in an organization is too broad to fit under our more constrained definition.

As we have now defined vulnerabilities we now consider the available security taxonomies and how we might classify them. In the next section we evaluate a few security taxonomies to determine if there is a need for a new taxonomy when many useful ones are extant. While reviewing this keep in mind that a vulnerability is a flaw which could allow unauthorized access or use. Almost by definition, vulnerabilities are not known until they are exploited . A feature may be considered a vulnerability as soon as its misuse is illustrated. If an organization wants to keep a feature active despite potential for misuse without following good security practice, we propose that this organization face the social cost to the system that such a desire imposes. Simply requiring "no vulnerabilities" is a command and control regulatory intensive solution.

Classifying Computer Security Failures

Any taxonomy 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 we now review security taxonomies developed by experts in the field.

An early work on systems (Amoroso, 1994) argued that in addition to being complete and exclusive taxonomies should also be unambiguous, repeatable, acceptable, and useful. Consider how this applies to classifying only vulnerabilities for the purpose of pricing.

First it is most important that the mechanism be mutually exclusive. Any vulnerability must fit into only class in order to be defined. The price must in part be determined by the classification; therefore the classification must also be unambiguous.

A taxonomy of computer security need not be exhaustive for our interests. In particular viruses and worms are not of interest in terms of classification. Malicious actions are not the point of interest here. Rather the effort to price vulnerabilities would therefore remove

vulnerabilities from the network, thereby curbing widespread diffusion of viruses and worms.

Clearly the classification system must be repeatable to be unambiguous. However, once a vulnerability is classified there is no need to do so twice. Therefore this condition is less strenuous in this case than in the case of analysis of incidents.

All classifications would meet the last criteria: acceptability and usefulness. Amoroso (defines acceptable as being logical and intuitive so that the taxonomy might be widely adopted.

A taxonomy is also defined as "useful" by Amoroso if it provides insight into computer security. However, insight into computer security for the purposes of computer security research per se is not our point of interest here. Thus we will discard that requirement as inappropriate.

Now consider various security taxonomies.

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 which 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. This analysis includes issues of results of attacks and motivations of attackers. 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 which are not useful for this work.

Motivation is also the reason that the work by Landwhere et. al. does not apply. He focuses on genesis, time of introduction, and location. Time of introduction and location are of interest. Lanwhere's work is not applicable because of its inclusion of malicious code. His work was reproducible, but not generalizable. In this work we are not interested in the actively malicious attacks, which are the proper realm of law or national security, but of all extant vulnerabilities which we argue in the previous section is reasonably within the realm of economics.

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 Spafford's classification.

Synchronization faults and condition validation errors are classified as coding faults. Coding faults are faults which 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.

Pollution: The Pricing Analogy

The total amount of pollution generated by industrial processes is a function both of how dirty given plants are and how much output each plant produces. Pollution levels can be lowered both by giving consumers incentives to purchase products from clean plants and by encouraging plant owners to clean up their plants. Note that one policy, such as a tax on pollution in a competitive industry, can have both effects. Similarly, with coding, any pricing mechanism must create incentives for two parties: those installing the software and those creating it. Furthermore no perverse incentives, such as incentives to delay releasing patches, should be created.

Consider this classification of security failures and how this matches to conceptual pricing of pollution. Pollution can be priced based on total output, location of output and toxicity. Are there comparisons to security vulnerabilities?

Pollution is generated at businesses, at home and during the commute. The taxonomy of Aslam, Krsul, & Spafford illustrates that vulnerabilities are generated during code production and during use. The code production can be compared to the industrial creation of pollution; and the code in use compared to the consumer.

Compare the factors which might be used in pricing vulnerabilities. The factors include expected severity of damage, delay in response time, and costs of correction. Severity of damage is a function of connectivity. Delay in availability or access is a function of the service rendered by the machine under attack as well as network conditions. A besieged router would have far more effect than a home owner's machine, as least from the network perspective.

Measuring severity of damage on the network would require measuring the chance that a vulnerability would be exploited, the damage likely given that the vulnerability was exploited, and the increased risk of other machines given that the particular machine was subverted.

To determine risk of exploitation would require data which are not now available and likely never to be available. Not only are specific risks to specific machines unknown,

there are not public data on the overall pattern of use of vulnerabilities. The validity of extant proprietary data is unknown. Not only can the risk not be known in the specific it cannot be known in the aggregate. One cannot measure ambient crackers in the way one might measure ambient air quality and then extrapolate to cancer risk.

The losses on the exploited machine ideally reflect the investment of the owner of the machine in security. These losses are suffered by the same party which failed to secure the machine, thus are not at issue.

The increased risk to other machines is a function of the connectivity and the processing power of the machine. The connectivity is a function of the topology of the Internet. Unfortunately, (for our purposes) the topology of the Internet is not mapped. Thus this element of price would be highly uncertain and establishing a 'fair' price would be problematic.

An alternative approach is to treat vulnerabilities as commodities, and allocate an initial level of vulnerabilities, and then allow trading to set a market value. Thus the subtleties and high transactions costs of discrete pricing are avoided. Note that to make a computer perfectly secure it may be theoretically necessary to disconnect it from the network. Thus, just as continued production requires continued it may be necessary to tolerate security vulnerabilities to continue connectivity.

An assumption about payment for vulnerabilities can be made from the observation of geographic indeterminacy of the Internet. We assume that any entity connected to the Internet can demand some form of payment or validation of credit ownership upon the discovery and documentation of a vulnerability.

In summary, we ignore the foundation of risk assessment and instead begin with a cost model based on the concept of vulnerabilities as flat-priced commodities.

Allocating Property Rights

In an article for which he later won the Nobel Prize, R.H. Coase proposed that an efficient production of goods usually associated with externalities could be achieved if all parties (e.g. the polluters and the homeowners) could get together to make arrangements to internalize the externalities (Coase, 1960). Coase argued that it did not matter who had the property rights if transactions costs were sufficiently low. Thus one could argue that the allocation of property rights and determination of direction of payment does not matter. The Coase Theorem argues that if transactions costs are high then the allocation of the property right and the law seriously affect the equilibrium.

For the purpose of pricing vulnerabilities to increase security rights could be assigned two ways. First, computers owners and operators could be charged for having vulnerabilities and coders could be charged for creating them. Second, users of the network could pay others not to use software or engage in practices with known vulnerabilities. The second option would obviously give users heavy incentives to employ vulnerabilities in order to be paid not to use them. We focus on the first option,

which allocates the right to a network free of vulnerabilities to all users and requires those that want to use vulnerabilities to buy that right.

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 transactions 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 costs.

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, 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 vulnerabilities a comparable approach can be used, by providing vulnerability credits appropriately to each entity using machines. However, distinguishing the entities and defining "appropriate" are 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 which 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 which would recognize that supercomputers are not equivalent to aging dedicated printer servers. Thus we would advocate considering processing power regardless of platform. Notice that this treats implementation and coding errors as equivalent. The hope is that producers of code with well-documented vulnerabilities would see a correcting market response when their code was identified as having many vulnerabilities.

Now having defined 'machines' we consider 'entities'. Defining the distinction between home and work, production and consumption is not trivial with information networks.

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 be 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 which 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.

Each vulnerability credit must be linked to a machine or device but not with a specific vulnerability. Software can be bundled with the appropriate vulnerability credits when sold, as part of the marketing of the good, just as consumers are not responsible for pollution generated during production.

Emergent faults can be owned by the network access provider (the ISP or organization) to prevent individual users from being harmed by the market. For example, an ISP might run scans and offer credits or computer security support to users. Prohibiting by contract that naïve users behave in a naïve manner is not reasonable.

Constructing A Vulnerabilty

A vulnerability should be associated with a specific machine. Each vulnerability should be a has chain which begin with ccreation and linkage to first

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.

We argue for a market mechanism to address the continued existence of well-documented vulnerabilities. Vulnerabilities can be defined as clear coding errors, such as buffer overrun problems, or implementation errors, e.g. implementing ftp incorrectly or not installing virus protection.

We would involve the government in that there would be an initial creation of vulnerability credits or designation of organizations to create such credits. Credits would be linked to a specific machine and have a finite lifetime. Records of machine use, including current location, would be required documentation.

We would further involve the government in that its subsidized incident response teams would be charged with continuing to post vulnerabilities. A vulnerability would be defined as actionable after it had been posted for some number of days by at least two incident response teams or some days after it has been used to subvert a system. Since some IRTs do not post until a patch is available this would give vendors limited veto power over vulnerabilities. Thus the adoption of the market would require that the existence of the vulnerability be posted immediately, thought certainly not the attack code.

References

Amoroso, E. G., 1994, *Fundamentals of Computer Security Technology*, Prentice-Hall PTR, Upper Saddle River, NJ, 1994.

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. 1998, "Measuring Positive Externalities from Unobservable Victim Precaution: An Empirical Analysis of Lojack", The Quarterly Journal of Economics, Vol. 113, p 43-77. February.

Coase, R.H., 1960, "The problem of social cost"' Journal of Law and Economics, Vol. 3, pp. 1-44.

Critical Foundations: Protecting America's Infrastructure: The Report of the President's Commission on Critical Infrastructure Protection, 1997, the President's Commission on Critical Infrastructure Protection, Washington DC.

Department of Defense, 1985, *Department of Defense Trusted Computer System Evaluation Criteria*, National Computer Security Center, Fort George G. Meade, MD.

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 Bureau of Standards, 1977, *Federal Information Processing Publication 46: Specifications for the Digital Encryption Standard,* United States Government Printing Office; Gaithersburg, MA.

National Institute of Standards and Technology, 1994, *Federal Information Processing Standards Publications 185: Escrowed Encryption Standard*, United States Government Printing Office; Gaithersburg, MA.

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. Vol. 12 (3). p 53-68. Summer.

Tygar and Whitten, 1996, "WWW Electronic Commerce and Java Trojan Horses", *Second USENIX Electronic Commerce Workshop*, Berkeley, CA. Also at http://www.cs.cmu.edu/afs/cs/project/decaf/web/usenix96/main.html