Avoiding Liability:
An Alternative Route to More Secure Products
(Draft)

Adam Shostack

March 5, 2005

1 Introduction

A healthy debate is raging over extending liability rules to software companies. Respected security experts and economists argue that it is an effective way to force companies to internalize externalities. After all, if a company can spend nothing on security, and produce a product that customers will buy, why should they spend on security? If customers can't distinguish between a secure and an insecure product, the company that produces an insecure product will get to market first, and have an advantage. This shifts the high cost of dealing with insecurities to customers, who are in a poor position to fix the products they have purchased. Thus, imposing liabilities on software producers will induce them all to take care in the creation of their software.

Responses to this argument include that it will dampen entrepreneurship, because a large companies will find it easier to influence, and then comply with the "industry standard" practices that limit their liability. At the same time, corporate executives are focused on trying to limit their liabilities, rather than shift them around (WSJ). This executive opposition, coupled with contract provisions now being imposed by large buyers may be enough to prevent general software liability over the next several years.

What is a customer who wants better software to do? Twenty years ago, there was no good way for a customer to judge the quality of a used car. The dealer knows more about it than the customer reasonably can. It's expensive to bring twenty used cars to your mechanic to get them checked out, and besides, he may see your lemon as his paycheque.

Studying this market earned Akerlof and Spence a Nobel prize: They talk about assymetric information, lemons markets, and signaling, which is a message that's cheap to send if you are a high quality provider, but expensive if you're not.

Today, we have a number of ways of signaling the quality of a used car, including dealer-backed warranties, certified-pre-owned programs, and Carfax, which is a background checking system for cars. Can we take useful lessons from this for security? This essay will first look at some objections to the idea of signaling for security, examine some possible signals, critique those signals, and then compare signals to liability as a means to achieve appropriate security for an organization.

2 Issues with Signals

This section will focus on signals from producers of a broad class of IT productivity software. There are some nuances regarding the production and purchase of security software, discussed at the end of the section.

The first question to ask is, if we had a great signal, would anyone listen? Some software is purchased because it's a 'killer application' — it has some unique feature that nothing else provides. The software may deliver that value in isolation, or it could exist at the heart of an 'ecology,' a set of interacting programs, formats, or protocols that create that value together. Software vendors can charge a premium for their software when such features exist. Killer value propositions make it very hard to oppose a product over security concerns. So in many markets, signals will not matter much. However, as markets mature, and products become more commodity-like, vendors will try to distinguish themselves from their competitors. One way to do so might be security. Examples of this may be Linux vendors competing with Microsoft on security, or database vendors such as Oracle calling their products 'unbreakable'. [1] It seems likely that as competition on features becomes more difficult, properties of the product, such as security or reliability, will become more important. If it is possible to earn a premium by differentiating an otherwise commodity product, we should expect vendors to do so.

So if signaling could have value, an important objection that may be raised is the difficulty of assessing ones own product. ("'What is security', asked Pilate, and washed his hands.") But signaling is not useful because one side has perfect

[1] This also might come from vendors listening to their engineers, rather than their customers, about what makes their product special.

knowledge, it is useful because it helps to address asymmetries. That assessing security is difficult, and that we don't know how to quantify it doesn't matter.[2] The difficulty in performing this assessment does not mean that no asymmetry exists. The producers of a product know how much energy was put into security in requirements evaluation, design, construction, and testing. They know if their engineers are complaining about security. They know if they are receiving security bug reports. They may have run analysis tools of various sorts on their product. And so, while the producer may not have actuarial data that allows the prediction of future quality costs, they certainly have more information than the prospective customer who has just picked up a data sheet at a trade show. An asymmetry, if you will. In the market for education, there are perhaps similar difficulties. A student getting ready to chose a college may not have a fair assessment of how they'll fare in the wider world. The valedictorian of a small town high school may have troubles boyond merely adjusting to a top-notch school. If they are choosing a college solely as a signal they could choose either a top flight school, where they may be average or below average, or a second or third tier, where they could excel. (Spence) There are many signals available. Some of them, such as a warranty on a used car, can be given with varying levels of investment. Mechanics could remove components and test them for wear and tear and plug those numbers into a statistical analysis program, or they could take a quick look at the engine, and say "no warranty on this one." Thus, the assessment does not need to be numerical in order to have value.

Some people have claimed that TCSEC ratings, or their "Common Criteria" successors are signals, and this is somewhat correct. CC evaluations send a message that the producer of the product is willing to jump through hoops to sell to governments. However, no one has ever failed a CC evaluation (cite). Neither the price, or length of time the evaluation took are published. As such, most vendors end up with the same result. So while the message may be expensive to send, it may be worthwhile to fake. Another important property of signals is that they should be easy to read, and no one who has ever read a CC evaluation would claim they meet that criteria. Thus, the general market failure of the CC mechanism should not be held against signals.

Security software vendors, such as firewalls or anti-virus may behave differently for reasons particular to their business: They need to employ lots of security experts, and they may have harsh language in their licenses. These signals should, but probably won't be, read differently for security companies.

[2] For purposes of this discussion, anyways. If signals are useful in the absence of good assesment, they will be more useful and more precise as assessment improves.

3 Some Possible Signals

Having looked at some objections to signals as a class, we can turn our attention to some potential methods of signaling.

Economists studying advertising have claimed that advertising can be understood as a signal that a manufacturer believes their product is of high quality, and will be around for a while. Seals for websites, such as those offered by the BBBOnline or Trusty are pure signals of this form. Seal programs imposed by those with a financial stake in what might happen, like Visa or Mastercard, indicate that some real effort has been put into security. Do customers care about the differences between these types of seals?

Employing experts, either as employees or "advisors," may act as a signal. The buyer may assume that the expert has done deep due diligence on a company before choosing to join, and most experts will.

A much more negative signal can be sent by EULAs: "Even if we know about a vulnerability, and don't fix it, we are not liable." This signal tends to be buried in the noise of legalese, but once noticed, it can be widely discussed.

Some advocates of open source claim that that making source code available acts as a signal, indicating the vendor's confidence that the software contains fewer bugs than commercial equivalents. There is enough public disagreement about the relative security qualities of open and closed source products that this may not be an effective signal. Better agreement about how to measure security, followed by measurements, may change this.

As a thought experiment, I've proposed using the output of RATS as a signal. RATS, the Rough Auditing Tool for Security, analyzes source code of a program for well known types of flaws. It catches simple problems, and rates them as high, medium, or low import. Because of the simplicity of the analysis, it labels many safe bits of code as problems. This last flaw is not fatal, as long as everyone is using the same tool. As long as each organization follows the same process, it should have roughly the same proportion of false positives. Slightly more troublesome is the three levels of output: Should you pick a project with 10 medium issues, or 4 highs? What about 10 highs in 100 lines of code, versus 40 in 1000? The 'rate' of issues is higher for the former, but there are more issues in the later, each of which has some chance of being found to be exploitable. Finally, there is a moral hazard, where organizations will be rewarded for fixing things that don't need fixing, rather than more productive work.

Security analysis reports, written by well-known individuals or firms, can act

as a signal. The willingness of the firm to undergo a security review by someone who is well-motivated to attack it sends a signal. That signal is weakened somewhat by the expert's desire for more review work, which may cause them to be gentle with the product.

4 Signals or Liability?

When there is a new market with a killer app, signals about security will be ignored. What's more, the provider of the killer app is probably collecting monopoly rents. It may make sense to apply liability, either through contract or law. As products tend towards commodity status, the application of liability, with the attendant barriers to entry, may prevent new, low cost competitors from entering the market. If signals can aid a purchaser in evaluating two commodity products, then that purchaser is better off. Thus signals may serve more useful function in a commodities market than in a new one.

I expect that evaluation of possible signals and comparisons between them will be a fruitful area of research in the future.

Acknowledgements

Andrew Stewart, Dave Clauson, and Jon Amsler engaged in many useful wide-randing discussions of security economics. Dan Geer provided encouragement to turn this into something approaching a paper. Eric K. Rescorla pointed out that security usually can't compete with features. Pete Lindstrom, Kevin Soo Hoo, and Gunnar Peterson critiqued the ideas.

References

Barnes *DeWorming The Internet*, Doug Barnes,

Spence As cited in *The 2001 Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel Information for the Public*, at `http://nobelprize.org/economics/laureates/2001/public.html` (undated).

WSJ *Companies Seek to Hold Software Makers Liable for Flaws*, David Bank, The Wall Street Journal, February24,2005,PageB1