Information Security and Externalities

By Bruce Schneier

January, 2007

http://www.schneier.com/essay-150.html

ENISA (European Network and Information Security Agency) Quarterly [PDF]

This essay is an update of Information security: How liable should vendors be?, Computerworld, October 28, 2004.

Information insecurity is costing us billions. There are many different ways in which we pay for information insecurity. We pay for it in theft, such as information theft, financial theft and theft of service. We pay for it in productivity loss, both when networks stop functioning and in the dozens of minor security inconveniences we all have to endure on a daily basis. We pay for it when we have to buy security products and services to reduce those other two losses. We pay for the lack of security, year after year.

Fundamentally, the issue is insecure software. It is a result of bad design, poorly implemented features, inadequate testing and security vulnerabilities from software bugs. The money we spend on security is to deal with the myriad effects of insecure software. Unfortunately, the money spent does not improve the security of that software. We are paying to mitigate the risk rather than fix the problem.

The only way to fix the problem is for vendors to improve their software. They need to design security in their products from the start and not as an add-on feature. Software vendors need also to institute good security practices and improve the overall quality of their products. But they will not do this until it is in their financial best interests to do so. And so far, it is not.

The reason is easy to explain. In a capitalist society, businesses are profit-making ventures, so they make decisions based on both short- and long-term profitability. This holds true for decisions about product features and sale prices, but it also holds true for software. Vendors try to balance the costs of more secure software – extra developers, fewer features, longer time to market – against the costs of insecure software: expense to patch, occasional bad press, potential loss of sales.

So far, so good. But what the vendors do not look at is the total costs of insecure software; they only look at what insecure software costs them. And because of that, they miss a lot of the costs: all the money we, the software product buyers, are spending on security. In economics, this is known as an externality: the cost of a decision that is borne by people other than those taking the decision.

Normally, you would expect users to respond by favouring secure products over insecure products – after all, users are also making their buying decisions based on the same capitalist model. Unfortunately, that is not generally possible. In some cases software monopolies limit the available product choice; in other cases, the 'lock-in effect' created by proprietary file formats or existing infrastructure or compatibility requirements makes it harder to switch; and in still other cases, none of the competing companies have made security a differentiating characteristic. In all cases, it is hard for an average buyer to distinguish a truly secure product from an insecure product with a 'trust us' marketing campaign.

Because of all these factors, there are no real consequences to the vendors for having insecure or low-quality software. Even worse, the marketplace often rewards low quality. More precisely, it rewards additional features and timely release dates, even if they come at the expense of quality. The result is what we have all witnessed: insecure software. Companies find that it is cheaper to weather the occasional press storm, spend money on PR campaigns touting good security and fix public problems after the fact, than to design security in from the beginning.

And so the externality remains…

If we expect software vendors to reduce features, lengthen development cycles and invest in secure software development processes, it needs to be in their financial best interests to do so. If we expect corporations to spend significant resources on their own network security – especially the security of their customers – it also needs to be in their financial best interests.

Liability law is one way to make it in those organisations' best interests. If end users could sue software manufacturers for product defects, then the cost of those defects to the software manufacturers would rise. Manufacturers would then pay the true economic cost for poor software, and not just a piece of it. So when they balance the cost of making their software secure versus the cost of leaving their software insecure, there would be more costs on the latter side. This would provide an incentive for them to make their software more secure.

Basically, we have to tweak the risk equation in such a way that the Chief Executive Officer (CEO) of a company cares about actually fixing the problem – and putting pressure on the balance sheet is the best way to do that. Security is risk management; liability fiddles with the risk equation.

Clearly, liability is not all or nothing. There are many parties involved in a typical software attack. The list includes:

- the company that sold the software with the vulnerability in the first place
- the person who wrote the attack tool
- the attacker himself, who used the tool to break into a network
- and finally, the owner of the network, who was entrusted with defending that network.

100% of the liability should not fall on the shoulders of the software vendor, just as 100% should not fall on the attacker or the network owner. But today, 100% of the cost falls directly on the network owner, and that just has to stop.

Certainly, making software more secure will cost money, and manufacturers will have to pass those costs on to users in the form of higher prices. But users are already paying extra costs for insecure software: costs of third-party security products, costs of consultants and security services companies, direct and indirect costs of losses. But as long as one is going to pay anyway, it would be better to pay to fix the problem. Forcing the software vendor to pay to fix the problem and then passing those costs on to users means that the actual problem might get fixed.

Liability changes everything. Currently, there is no reason for a software company not to offer feature after feature after feature, without any regard to security. Liability forces software companies to think twice before changing something. Liability forces companies to protect the data they are entrusted with. Liability means that those in the best position to fix the problem are actually responsible for the problem.

Information security is not a technological problem. It is an economics problem. And the way to improve information security is to fix the economics problem. If this is done, companies will come up with the right technological solutions that vendors will happily implement. Fail to solve the economics problem, and vendors will not bother implementing or researching any security technologies, regardless of how effective they are.