Why Information Security is Hard – An Economic Perspective

Ross Anderson

University of Cambridge Computer Laboratory
Ross.Anderson@cl.cam.ac.uk
30th January 2001

1 Executive Summary

According to one common view, information security comes down to technical measures. Given better access control policy models, formal proofs of cryptographic protocols, approved firewalls, better ways of detecting intrusions and malicious code, and better tools for system evaluation and assurance, the problems can be solved.

In this note, I put forward a contrary view: information insecurity is at least as much due to perverse incentives. Many, if not most, of the problems can be explained more clearly and convincingly using the language of microeconomics: network externalities, asymmetric information, moral hazard, adverse selection, liability dumping and the tragedy of the commons.

2 Introduction

In a 1993 survey of fraud against automatic teller machines (ATMs) [2], it was found that patterns of fraud depended on who was liable for them. In the USA, if a customer disputed an ATM transaction, the onus was on the bank to prove that the customer was mistaken or lying; this gave US banks a motive to protect their systems properly. But in Britain, Norway and the Netherlands, the burden of proof lay on the customer: the bank was right unless the customer could prove it wrong. Since this was almost impossible, the banks in these countries became careless. Eventually, an epidemic of ATM fraud demolished their complacency. US banks, meanwhile, suffered much less fraud; although they actually spent less money on security than their European counterparts, they spent it more effectively.

There are many other examples. Medical payment systems, that are paid for by insurers rather then by healthcare providers, fail to protect patient privacy whenever this conflicts with the insurer's wish to collect information about its clients. Digital signature laws transfer the risk of forged signatures from the bank that relies on the signature (and that built the system) to the person alleged to have made the signature. Common Criteria evaluations are not made by the relying party, as Orange Book evaluations were, but by a commercial facility paid by the vendor. In general, where the party who is in a position to protect a system is not the party who would suffer the results of security failure, then problems may be expected.

A different kind of incentive failure surfaced in 1999, with distributed denial of service attacks against a number of high-profile web sites. These exploit a number of subverted machines to launch a large coordinated packet flood at a target. Since many of them flood the victim at the same time, the traffic is more than the target can cope with, and because it comes from many different sources, it can be very difficult to stop [5]. Hal Varian pointed out that this was also a case of incentive failure [13]. While individual computer users might be happy to spend $100 on anti-virus software to protect themselves against attack, they are unlikely to spend even $1 on software to prevent their machines being used to attack a third party such as Amazon or Microsoft.

This is an example of what economists refer to as the 'Tragedy of the Commons' [9]. If a hundred peasants graze their sheep on the village common, then whenever another sheep is added its owner gets almost the full benefit while the other ninety-nine suffer only a very small decline in the quality of the grazing. So they aren't motivated to object, but rather to add another sheep of their own and get as much of the grazing as they can. The result is a dustbowl; and the solution is regulatory rather than technical. A typical tenth-century Saxon village had community mechanisms to deal with this problem; the world of computer security still doesn't. Varian's proposal is that the costs of distributed denial-of-service attacks should fall on the operators of the networks from which the flooding traffic originates; they can then exert pressure on their users to install suitable defensive software, or risk having their service terminated if their machine hosts an attack.

These observations prompted us to look for other ways in which economics and computer security interact.

3 Network Externalities

Economists have devoted much effort to the study of networks such as those operated by phone companies, airlines and credit card companies.

The more people use a typical network, the more valuable it becomes. The more people use the phone system – or the Internet – more people there are to talk to and so the more useful it is to each user. This is sometimes referred to as *Metcalfe's law.* This isn't limited to communication systems. The more merchants take credit cards, the more useful they are to customers, and so the more customers will buy them; and the more customers have them, the more merchants will want to accept them. The effect is that networks can grow very slowly at first – credit cards took almost two decades to take off – but then, once positive feedback gets established, they can grow very rapidly. The telegraph, the telephone, the fax machine and most recently the Internet have all followed this model.

As well as these physical networks, the same principles apply to virtual networks such as the community of users of a particular software architecture. When software developers started to believe that the PC would outsell the Mac, they started developing their products for the PC first, and for the Mac only later (if at all). This made customers more likely to buy a PC than a Mac, and the resulting positive feedback squeezed the Mac out of most markets. A similar effect made Microsoft Word the dominant word processor.

A good introduction to network economics is by Shapiro and Varian [11]. For our present purposes, there are three particularly important features of information technology markets.

- First, the value of a product to a user depends on how many other users adopt it.
- Second, technology often has high fixed costs and low marginal costs. The first copy of a chip or a software package may cost millions, but subsequent copies may cost very little to manufacture. This isn't unique to information markets; it's also seen in business sectors such as airlines and hotels. In all such sectors, pricing at marginal cost will tend to drive revenues steadily down towards the cost of production (which in the case of information is zero).
- Third, there are often large costs to users from switching technologies, which leads to lock-in. Such markets may remain very profitable, even where (incompatible) competitors are very cheap to produce. In fact, one of the main results of network economics is that the net present value of the customer base should equal the total switching costs.

All three of these effects tend to lead to "winner take all" market structures with dominant firms. So it is extremely important to get into markets quickly. Once in, a vendor will try to appeal to complementary suppliers, as with the software vendors whose bandwagon effect carried Microsoft to victory over Apple. In fact, successful networks tend to appeal to complementary suppliers even more than to users: the potential creators of "killer apps" need to be courted. Once the customers have a substantial investment in complementary assets, they will be locked in. Odlyzko observes that much of the lack of user-friendliness of both Microsoft software and the Internet is due to the fact that both Microsoft and the Internet achieved success by appealing to developers. The support costs that Microsoft dumps on users – and in fact even the cost of the time wasted waiting for PCs to boot up and shut down – greatly exceed its turnover [10].

Consultants often explain that the reason a design broke for which they were responsible was that 'the client didn't want a secure system, but just the most security I could fit on his product in one week on a budget of $10,000'. It's important to realise that this isn't just management stupidity. The huge first-mover advantages that can arise in economic systems with strong positive feedback are the origin of the philosophy of ‘we’ll ship it on Tuesday and get it right by version 3’. Although often attributed by cynics to a moral failing on the part of Bill Gates, this is perfectly rational behaviour in many markets where network economics apply.

Another common complaint is that software platforms are shipped with little or no security support, as with Windows 95/98; and even where access control mechanisms are supplied, as with Windows NT, they are easy for application developers to bypass. In fact, the access controls in Windows NT are largely irrelevant, as most applications are written to run with administrator privilege. This is also explained simply from the viewpoint of network economics: mandatory security would subtract value, as it would make life more difficult for the application developers.

Network owners and builders will also appeal to the developers of the next generation of applications by arranging for the bulk of the support costs to fall on users rather than developers – even if this makes effective security administration impractical. The current craze for public key cryptography may simplify some designs, but it has been criticised for placing an unreasonable administrative burden on users who are neither prepared nor willing to undertake it [7].

4 Competitive applications and corporate warfare

Network economics has many other effects on security engineering. Rather than using a standard, well analyzed and tested solution, companies often prefer a proprietary obscure one in order to increase customer lock-in and increase the investment that competitors have to make to create compatible products. Where possible, they will use patented algorithms (even if these are not much good) as a means of imposing licensing conditions on manufacturers. For example, the DVD Content Scrambling System was used as a means of insisting that manufacturers of compatible equipment signed up to a whole list of copyright protection measures [3]. This may have failed because it would have prevented the Linux operating system from running on next generation PCs; but efforts to foist non-open standards continue in many applications from SDMI and CPRM to completely proprietary systems such as games consoles.

A very common objective is differentiated pricing. This means pricing the product or service not to its cost but to its value to the customer. This is familiar from the world of air travel: you can spend $200 to fly the Atlantic in coach class, $2000 in business class or $5000 in first. This business model is spreading widely in the software and online services sectors. A basic program or service may be available free; a much better one for a subscription; and a ‘Gold’ service at a ridiculous price. In many cases, the program is the same except that some features are disabled for the budget user. Many protection mechanisms have as their real function the maintenance of this differential.

Another strategy is to manipulate switching costs. Incumbents try to increase the cost of switching, whether by indirect methods such as controlling marketing channels and building industries of complementary suppliers, or, increasingly, by direct methods such as making systems incompatible and hard to reverse engineer. Meanwhile competitors try to do the reverse: they look for ways to reuse the base of complementary products and services, and to reverse engineer whatever protection the incumbent builds in. This extends to the control of complementary vendors, sometimes using technical mechanisms.

Sometime, security mechanisms have both product differentiation and higher switching costs as goals. An example which may become politicised is 'accessory control'. According to one company that sells authentication chips into the automative market, some printer companies have begun to embed cryptographic authentication protocols in laser printers to ensure that genuine toner cartridges are used. If a competitor's cartridge is loaded instead, the printer will quietly downgrade from 1200 dpi to 300 dpi. In mobile phones, much of the profit is made on batteries, and authentication can be used to spot competitors' products so they can be drained more quickly. (I wonder how long it will be before the research which toner cartridge and battery manufactures will do to defeat these systems will hit the street in the form of better car theft tools?)

Another example comes from Microsoft Passport. This is a system whose ostensible purpose is single signon: a Passport user doesn't have to think up separate passwords for each participating web site, with all the attendant hassle and risk. Instead, sites that use Passport share a central authentication server run by Microsoft to which users log on. Servers use web redirection to connect their Passport-carrying visitors to this server; authentication requests and responses are passed between them by the user's browser in encrypted cookies. So far, so good.

But the real functions of Passport are somewhat more subtle [12]. First, by patching itself into all the web transactions of participating sites, Microsoft can collect a huge amount of data about online shopping habits and enable participants to swap it. If every site can exchange data with every other site, then the value of a network of web sites is the square of the number of sites and there is a strong network externality. So one such network may come to dominate, and Microsoft hopes to own it. Second, the authentication protocols used between the merchant servers and the Passport server are proprietary variants of Kerberos, so the web server must use Microsoft software rather than Apache or Netscape. So Passport isn't so much a security product, as a play for control of both the web server and purchasing information markets. It comes bundled with services such as Hotmail, is already used by 40 million people, and does 400 authentications per second on average. Its known flaws include that Microsoft keeps all the users' credit card details, creating a huge target; various possible middleperson attacks; and that you can be impersonated by someone who steals your cookie file. Passport has a 'logout' facility that's supposed to delete the cookies for a particular merchant, so you can use a shared PC with less risk, but this feature doesn't work properly for Netscape users [8].

The constant struggles to entrench or undermine monopolies and to segment and control markets determine many of the environmental conditions that make the security engineer's work harder.

So much for commercial information security. But what about the government sector? As information attack and defence become ever more important tools of national policy, what broader effects might they have?

5 Information Warfare – Offence and Defence

One of the most important aspects of a new technology package is whether it favours offence or defence in warfare. The balance has repeatedly swung back and forth, with the machine gun giving an advantage to the defence in World War 1, and the tank handing it back to the offence by World War 2.

Let's take a slightly simplified example. Suppose a large, complex product such as Windows 2000 has 1,000,000 bugs, each with an MTBF of 1,000,000,000 hours. Suppose that Paddy works for the Irish Republican Army and his job is to break into the British Army's computer to get the list of informers in Belfast, while Brian is the army assurance guy whose job is to stop Paddy. So he must learn of the bugs before Paddy does.

Paddy has a day job so he can only do 1000 hours of testing a year. Brian has full Windows source code, dozens of PhDs, control of the commercial evaluation labs, an inside track on CERT, an information sharing deal with other UKUSA member states – and he also runs the government's scheme to send round consultants to critical industries such as power and telecomms to advise them how to protect their systems. Suppose that Brian benefits from 10,000,000 hours a year worth of testing.

After a year, Paddy finds a bug, while Brian has found 100,000. But the probability that Brian has found Paddy's bug is only 10%. After ten years he will find it – but by then Paddy will have found nine more, and it's unlikely that Brian will know of all of them. Worse, Brian's bug reports will have become such a firehose that Microsoft will have killfiled him.

In other words, Paddy has thermodynamics on his side. Even a very moderately resourced attacker can break anything that's at all large and complex. There is nothing that can be done to stop this, so long as there are enough different security vulnerabilities to do statistics: different testers find different bugs. (The actual statistics are somewhat more complicated, involving lots of exponential sums; keen readers can find the details at [4].)

There are various ways in which one might hope to escape this statistical trap.

– First, although it's reasonable to expect a 35,000,000 line program like Windows 2000 to have 1,000,000 bugs, perhaps only 1% of them are security-critical. This changes the game slightly, but not much; Paddy now needs to

recruit 100 volunteers to help him. Still, the effort required of the attacker is still much less than that needed for effective defence.
- Second, there may be a single fix for a large number of the security critical bugs. For example, if half of them are stack overflows, then perhaps these can all be removed by a new compiler that traps them somehow.
- Third, you can make the security critical part of the system small enough that the bugs can be found. This was understood, in an empirical way, by the early 1970s. However, the discussion in the above section should have made clear that a minimal TCB is unlikely to be available anytime soon, as it would make applications harder to develop and thus impair the platform vendors' appeal to developers.

So information warfare looks rather like air warfare looked in the 1920s and 1930s. Attack is simply easier than defence. Defending a modern information system could also be likened to defending a large, thinly-populated territory like the nineteenth century Wild West: the men in black hats can strike anywhere, while the men in white hats have to defend everywhere. Another possible relevant analogy is the use of piracy on the high seas as an instrument of state policy by many European powers in the sixteenth and seveteenth centuries.

The technical bias in favour of attack is made even worse by asymmetric information. Suppose that you are the head of an agency with economic intelligence tasks, and a computer scientist working for you has just discovered a beautiful new exploit on Windows 2000. If you report this to Microsoft, you will protect 250 million Americans; if you keep quiet, you will be able to conduct operations against 400 million Europeans and 100 million Japanese. What's more, you will get credit for operations you conduct successfully against foreigners, while the odds are that any operations that they conduct successfully against U.S. targets will remain unknown to your superiors. This further emphasises the motive for attack rather than defence. Finally – and this appears to be less widely realised – the balance in favour of attack rather than defence is still more pronounced in smaller countries. They have proportionally fewer citizens to defend, and more foreigners to attack.

In other words, the increasing politicization of information attack and defence is likely to be a destabilising factor in international affairs.

6 Distinguishing Good from Bad

Another insight from the theory of asymmetric information is the 'market for lemons', an explanation of why bad products drive out good products from many markets.

Consider a used car market, on which there are 100 good cars (the 'plums'), worth $3000 each, and 100 rather troublesome ones (the 'lemons'), each of which is worth only $1000. The vendors, of course, know which is which, but the buyers don't. So what will be the equilibrium price of used cars?

If customers start off believing that the probability they will get a plum is equal to the probability they will get a lemon, then the market price will start off at $2000. However, at that price only lemons will be offered for sale, and once the buyers observe this, the price will drop rapidly to $1000 with no plums being sold at all. In other words, when customers don't have as much information about the quality of the products as the vendors do, then there will be severe downward pressure on both price and quality. This clearly happens in the markets for information security products and services. It can be made even worse when the people doing the evaluation of these products or services aren't the people who will suffer when they fail.

Much has been written on the ways in which corporate performance can be adversely affected when executives have incentives at odds with the welfare of their employer. For example, managers often buy products and services which they know to be suboptimal or even defective, but which are from big name suppliers. This is known to minimize the likelihood of getting fired when things go wrong. Corporate lawyers don't condemn this as fraud, but praise it as due diligence. Over the last decade of the twentieth century, many businesses have sought to fix this problem by extending stock options to ever more employees. However, these incentives don't appear to be enough to ensure prudent practice by security managers. (This might be an interesting topic for a PhD; can it be explained by the mathematics of insurance markets, or is it simply down to adverse selection among security managers?)

This problem has long been perceived, even if not in precisely these terms, and the usual solution to be proposed is an evaluation system. This can be a private arrangement, such as the equipment tests carried out by insurance industry laboratories for their member companies, or it can be public sector, as with the Orange Book and the Common Criteria.

For all its faults, the Orange Book had the virtue that evaluations were carried out by the party who relied on them – the government. The European equivalent, ITSEC, introduced a pernicious innovation – that the evaluation was not paid for by the government but by the vendor seeking an evaluation on its product. This got carried over into the Common Criteria.

This change in the rules provided the critical perverse incentive. It motivated the vendor to shop around for the evaluation contractor who would give his product the easiest ride, whether by asking fewer questions, charging less money, taking the least time, or all of the above. To be fair, the potential for this was realized, and schemes were set up whereby contractors could obtain approval as a *commercial licensed evaluation facility* (CLEF). The threat that a CLEF might have its license withdrawn was supposed to offset the commercial pressures to cut corners.

In none of the half-dozen or so affected cases I've been involved in has the Common Criteria approach proved satisfactory. Some examples are documented in my book on Security Engineering [1]. The failure modes appear to involve fairly straightforward pandering to customers' wishes, even (indeed especially) where these were in conflict with the interests of the users for whom the evaluation was supposedly being prepared. The lack of sanctions for misbehaviour – such as a process whereby evaluation teams can lose their accreditation when they lose their sparkle, or get caught in gross incompetence or dishonesty, is probably a contributory factor.

But from the user's point of view, an evaluation may actually detract from the value of a product. For example, if you use an unevaluated product to generate digital signatures, and a forged signature turns up which someone tries to use against you, you might reasonably expect to challenge the evidence by persuading a court to order the release of full documentation to your expert witnesses. A Common Criteria certificate might make a court very much less ready to order disclosure, and thus could severely prejudice your rights. A cynic might suggest that this is precisely why it's the vendors of products which which are designed to transfer liability (such as smartcards), to satisfy due diligence requirements (such as firewalls) or to impress naive users (such as PC access control products) who are most enthusiastic about the Common Criteria.

So an economist is unlikely to trust a Common Criteria evaluation. Fortunately, the economics discussed above should limit the uptake of the Criteria to sectors where an official certification, however irrelevant, erroneous or mendacious, offers some competitive advantage.

7 Conclusions

Much has been written on the failure of information security mechanisms to protect end users from privacy violations and fraud. This misses the point. The real driving forces behind security system design usually have nothing to do with such altruistic goals. They are much more likely to be the desire to grab a monopoly, to charge different prices to different users for essentially the same service, and to dump risk. Often this is perfectly rational.

In an ideal world, the removal of perverse economic incentives to create insecure systems would de-politicize most issues. Security engineering would then be a matter of rational risk management rather than risk dumping. But don't hold your breath. Information security is about power; it is about raising barriers to trade, segmenting markets and differentiating products. Its tools include asymmetric information and moral hazard. As fast as one perverse incentive can be removed by regulators, businesses (and foreign governments) are likely to create two more. In other words, the management of information security is a very much deeper and more political problem than is usually realised; solutions are likely to be subtle and partial, while many simplistic technical approaches are bound to fail. The time has come for engineers, economists, lawyers and policymakers to try to forge common approaches.

Acknowledgements

Hal Varian provided very useful comments on early drafts of some of this material.

References

1. RJ Anderson, *'Security Engineering – A Guide to Building Dependable Distributed Systems'*, Wiley (2001) ISBN 0-471-38922-6
2. RJ Anderson, "Why Cryptosystems Fail" in *Communications of the ACM* vol 37 no 11 (November 1994) pp 32–40
3. JA Bloom, IJ Cox, T Kalker, JPMG Linnartz, ML Miller, CBS Traw, "Copy Protection for DVD Video", in *Proceedings of the IEEE* v 87 no 7 (July 1999) pp 1267–1276
4. RM Brady, RJ Anderson, RC Ball, *'Murphy's law, the fitness of evolving species, and the limits of software reliability'*, Cambridge University Computer Laboratory Technical Report no. 476 (1999); at `http;//www.cl.cam.ac.uk/~rja14`
5. CERT, Results of the Distributed-Systems Intruder Tools Workshop, Software Engineering Institute, Carnegie Mellon University, `http://www.cert.org/reports/dsit_workshop-final.html`, December 7, 1999
6. W Curtis, H Krasner, N Iscoe, "A Field Study of the Software Design Process for Large Systems", in *Communications of the ACM* v 31 no 11 (Nov 88) pp 1268–1287
7. D Davis, "Compliance Defects in Public-Key Cryptography", in *Sixth Usenix Security Symposium Proceedings* (July 1996) pp 171–178
8. DP Kormann, AD Rubin, "Risks of the Passport Single Signon Protocol", in *Computer Networks* (July 2000); at `http://avirubin.com/vita.html`
9. WF Lloyd, *'Two Lectures on the Checks to Population'*, Oxford University Press (1833)
10. AM Odlyzko, "Smart and stupid networks: Why the Internet is like Microsoft", ACM netWorker, Dec 1998, pp 38–46, at `http://www.acm.org/networker/issue/9805/ssnet.html`
11. C Shapiro, H Varian, *'Information Rules'*, Harvard Business School Press (1998), ISBN 0-87584-863-X
12. J Spolsky, "Does Issuing Passports Make Microsoft a Country?" at `http://joel.editthispage.com/stories/storyReader$139`
13. H Varian, Managing Online Security Risks, Economic Science Column, The New York Times, June 1, 2000, `http://www.nytimes.com/library/financial/columns/060100econ-scene.html`