More Security or Less Insecurity (Transcript of Discussion)

Bruce Christianson

University of Hertfordshire

This is actually work done by Partha, it's his talk, but the UKBA[1] decided we could do without him, which is why it's me talking rather than him. The purpose of this talk is to explore the possibility of an exploitable analogy between approaches to secure system design and theories of jurisprudence. The prevailing theory of jurisprudence in the West at the moment goes back to Hobbes. It was developed by Immanuel Kant and later by Rousseau, and is sometimes called the contractarian model after Rousseau's idea of the social contract. It's not the sort of contract that you look at and think, oh gosh, that might be nice, I might think about opting in to that, it's more like a pop up licence agreement that says, do you want to comply with this contract, or would you rather be an outlaw. So you don't get a lot of choice about it. Sometimes the same theory, flying the flag of Immanuel Kant, is called transcendental institutionalism, because the basic approach says, you identify the legal institutions that in a perfect world would govern society, and then you look at the processes and procedures, the protocols that everyone should follow in order to enable those institutions to work, and then you say, right, that can't be transcended, so therefore there's a moral imperative for everyone to do it. So this model doesn't pay any attention to the actual society that emerges, or to the incentives that these processes actually place on various people to act in a particular way. It doesn't look at any interaction effects, it simply says, well you have to behave in this particular way because that's what the law says you have to do, and the law is the law, and anybody who doesn't behave in that way is a criminal, or (in our terms) is an attacker.

In the vanilla model of secure systems design that we find in books for children, we advocate the same kind of approach; you have a security policy, you identify the institutions and protocols that are going to implement this policy, and you say, well people have to behave in accordance with the security policy because otherwise they are policy breakers, and that's the definition of an attacker. And the problem with these assumptions is not only do they make systems hard to use, and unpleasant, but they also have the effect of building in the weak links to the system.

Paulo Verissimo: Would you say that those weak links are proportional to what we might call the realism of the assumptions, meaning you want it to be, that's why it's not possible?

Reply: Precisely so, that is where the tension comes from. We put the assumption in because we need to do a proof, but that assumption means that we

[1] United Kingdom Border Authority

are not considering that attack, because we have defined it away. So precisely your point.

We don't look at the incentives that our implementation places on people. Most security breaches happen due to insiders being dishonest or incompetent, and because we made assumptions about ideal behaviour these weaknesses spread. For example, we assume that trust is transitive, and the consequence of that is that users now are compelled to enter into a non-negotiable trust relationship with large parts of the system infrastructure, so the individual client has no control over the risks that they are exposed to. Public key infrastructure is a classic example of that, where you have these innocuous looking assumptions, that even look as if they ought to be true for an ideal world, and yet they expose you to all these risks. Ross Anderson's "Why cryptosystems fail" comes back to exactly this point: designers know a lot about how their products might fail in theory, but they don't have the experience to know how they would fail in practice, and they don't really have a way of getting feedback into their model of policy. So they've got a particular threat model, and instead of saying when the system fails, oh the problem is with the threat model, they either say, oh the problem is with the protocol, we need more countermeasures, or they say, oh the problem is people didn't use the system properly, people didn't understand the licence before they clicked agree.

Now this brings me back nicely to a point that Sandy raised yesterday[2], which is, to what extent there's an analogy between bugs in ordinary systems where users just want performance and usability and things like that, and security violations in systems where we have some sort of anti-requirement; the things that we want the system not to do are just as important as what we do want it to do. It's almost a definition of maturity; a system is mature when most of your bugs come from fixes that you have put in to correct bugs that were there before. But there are two important differences. The first is, bugs don't get fixed unless they actually caused a problem, right, there's no notion of prophylactic fixing, if it doesn't cause a problem, it's not a bug yet. And the second thing is that after any piece of code has been in the field for a certain length of time, the fix is to document the bug carefully and propose a work-round. In other words, you change the behaviour of the client so as not to get the system to mis-behave, because that's got a good chance of working, whereas any bug that's been out there for that long that hasn't been fixed yet, either it's not really a problem, or it's not easy to fix.

Now in the security world there's an argument that says, in a mature system most security failures are enabled by countermeasures that we put in to counteract other threats. And the problem is that we have real genuine security threats coming from countermeasures that are there because of pretend threats. OK, we have a threat, and I don't think it's real, I'm not bothered about it, the world I live in is horribly insecure, I'm quite willing to take risks that are much bigger than this everyday, but the designer of the system thinks that it is a threat, and so they put in this countermeasure, and the countermeasure is what's giving me

[2] Clark, these proceedings

the grief, because the counter-threat is real. In order to prove perfect security we have to make behavioural assumptions, and the assumptions have the effect of masking the real threats. For example, I don't have a choice but to trust my bank, it's that or nothing. I don't have a choice but to trust the doctor or the boss. What's their incentive to be honest and competent? Well their incentive is that the policy says they have to be, yes, the law is the law, and it applies to everyone, so that's your guarantee. So what will be our Enlightenment? We're actually working with a medieval model of secure systems design, and so the argument is that this Holy Roman Emperor approach to system security[3] is not just a bad deal for users, it's actually a bad deal for system security as well.

The second part of the argument is to say that there are alternative approaches to jurisprudence, there are actually a couple of alternative theories about how judicial systems should be built. In the West, the opponents of Hobbes never really got their act together, so you've got all these guys like Woolston, Croft, and Bentham, and so forth, they never quite managed to agree on an alternative. But in Indian jurisprudence (my Sanskrit is even worse than my Latin, I'm sure that some of these should have a vowel on the end) there are essentially two Sanskrit terms for justice in Indian jurisprudence, one of which corresponds pretty much to the Western mainstream institutional transcendentalism, which says, well you have correct organisations, and people do what the protocols say they should do. And then you have this idea of Nyay, which is looking at what actually occurs, and coming to a view about whether or not justice has been done, and there's plenty of examples of what's called Fish Justice. The justice of Fish is that big fish eat little fish, and the little fish have no redress, right, and that's generally regarded as being a bad outcome. And there's all sorts of examples that you can think of where the key point is that according to the transcendental institutionalist view of justice, no injustice has been done. Institutions that have been constituted for that purpose have done things the protocol says they ought to do, and what has happened is exactly and perfectly correct. And yet it's very clear that the outcome isn't right, and something else ought to happen.

We have a very narrow view of what the system is there to do, but in fact people build systems because they believe that that system will enable them to interact socially in some way that they couldn't without it. So the idea is to look at one of these other systems of jurisprudence, obviously Partha's used in Nyay because it's a concept that he's familiar with, I might have wanted to run with Bentham or Rawls and Croft, but the focus shifts from saying, well we must have absolute justice, or absolute injustice, you know, absolute security, or no security at all, to saying, well there's a continual retreat and advance, and the trick isn't whether a system is secure or not, it's whether the insecurity manifests itself,

[3] *Fiat justitia, et pereat mundus*: "follow the security policy though the world be destroyed" was the motto of Ferdinand I, Holy Roman Emporor, 1558-1564. This phrase was also used by Kant in his "Perpetual Peace" (1795) to emphasise the counter-utilitarian aspect of his moral philosophy.

and if it does, is it redressable? What do you pay attention to, and what do you do about it? So it's actually much more like the bug fixing function.

The point is that we want to conform the system to what users want it to do, rather than saying, well the reason you have that login and password is so that you can comply with the security policy, that's what you were given those credentials for. So the consequence of that is, we have to build in some way of allowing manifest behaviour to feed back into the security policy. This is I guess the slightly heretical part that I'm putting forward, that we shouldn't take security policy as an input and read-only parameter for the design process. The design of the system shouldn't simply model incompetence and dishonesty as an attack, but should model it as motivated behaviour, and look to see what gives rise to it. So if you've got a problem with giving people the wrong incentives, then build the system in a way that gives different incentives, or move the weak links to somewhere where you believe you can redress them. The second slightly heretical suggestion is that, contra Mansfield[4], it is a very good idea if you're trying to take somebody's liberty away, or to restrict the freedom with which they can exercise the system, to explain to them as part of the design process exactly why you want to impose that restriction. Because quite often the provision of the reasons allows the right debate to take place, instead of falling back on the approach that says, well you've got to have an identity card because otherwise the security policy won't work.

The other point is that there's a very consequentialist view of behavioural transgressions that emerges in the current approach, which says well they did something wrong, and that had bad consequences, whereas actually this is an opportunity to allow the implementation to feed back into the policy. So we shouldn't just look at low-level details and say, well that's implementation, we specified what you have to do, and you do it, it doesn't really matter how you do it, just make sure you meet the specification at the higher level and you're fine. It's important to look at the impact of choosing one mechanism rather than another and to be willing not just to change the mechanism, but to go back up a level, reconfigure the user, and change the policy in such a way that when you re-implement it, it will manifest itself differently.

"The best is the enemy of the good", that one's Voltaire. There's still debate about which way round he meant it. But certainly in security there's an argument that pursuit of excellence is pernicious, and rather than pretend to be able to get it right, we need to accept that as soon as we release the system there will be a honeymoon period where people will start finding out the things that are wrong, and the question is, how can we redress an insecurity that emerges once it manifests itself. Part of that involves accepting that there isn't a unique Platonic threat model that we just haven't guessed right yet, instead we do have to cope with multiple intersecting views. And configuring the users is a perfectly

[4] "Tut, man, decide promptly, but never give any reasons for your decisions. Your decisions may be right, but your reasons are sure to be wrong." William Murray, 1st Earl of Mansfield, 1705–1793, quoted by John Cordy Jeaffreson in "A Book about Lawyers", Volume 1, page 85 (1867).

legitimate thing to do, we tend to try and do it by coercion and re-education at the moment, doing it by providing reasons and matching the incentives would probably be a better approach. And we have to allow for the effects that these mechanisms should have on the security policy.

Sandy Clark: The problem with configuring the users and the idea of using different motivations is that users aren't rational, they don't think logically, they don't make decisions logically, they make them emotionally.

Reply: Absolutely.

Sandy Clark: And so trying to reason with them, it just wouldn't work.

Reply: Ah yes, I should distinguish very carefully between reasoning with people and offering them an incentive, OK. So offering an incentive, for example might be, vote for the Healthcare Bill because it will bring the Rapture closer, that's a perfectly reasonable incentive, and it's the deep South mostly that's not got the healthcare anyway.

We've only got to do better than the techniques that systems are using at the moment. But I think you've made a very good point about the fact that people's responses to incentives are not necessarily rational, and thinking about it all in a remorselessly Benthamite way isn't necessarily a good approach. But even when people are behaving completely rationally, they don't make decisions based on what is true and what isn't, they make decisions based on what they believe to be true, which in turn is based on their perception, and even if you're dealing with perfectly rational people, that's a real stumbling block that the current approach can't cope with, or at least doesn't.

Paul Wernick: And not just when dealing with rational people. How many times have you gone onto a website, it demands a password, they're not going to come here again, I don't want it, chuck in something so the wretched thing will allow me access. And you know, the model that the designer of the website had was, this has got to be secure, a secure network will have passwords, if you have a password you will be secure, and people are saying, stuff this, I just want to get this one thing and go away.

Jonathan Anderson: But that is rational behaviour, for me to say, I don't care about your security policy, I just want to read the New York Times.

Reply: The irrational behaviour there is on the part of the designer. What they've effectively done is to pop up a box that says, do you really want to use this website, and you type yes in as your password, and away you go. That isn't what they had in mind, partly because they never explained their reason, so there was never any discussion about whether this was a reasonable thing to do or not, and partly because of the incentive that they've offered you, enter a password and win a chocolate fish right now.

Paul Wernick: And there is also a reluctance to learn from safety engineering; I've seen in the past a paper on security protocols that seemed to say, we have a generally single track railway, but we might slow down for the threat of these two trains coming towards each other on that track. If you take a holistic view of the entire thing, you can't design out of it some particular threats, you just need to accept that the real world is the message.

Reply: Well it's our policy that we don't, and from the software point of view, it's a sort of a looking glass game: if somebody says, well in our policy we assume we have tamper-proof hardware, that sounds really secure, you know, tamper-proof hardware, brilliant, but what they're actually saying is, we are not going to consider the threat of somebody getting the secret out of the box, we're defining it away. It's the same again: wouldn't it be better to look at the consequences of what we assume?

Paul Wernick: And perhaps treating people as human beings rather than as rational creatures?

Reply: Well we have this idea, as I said, in the books for children, that says, come up with the perfect threat model, then design the transcendental countermeasures, and then implement. Whereas to do something that's more or less right, and stand by it with a mallet, and play whack the mole when they start appearing, that's a terrible and wicked thing to do.

What I'm trying to say is, actually if you do that properly it's not such a bad approach[5].

Paul Wernick: Doesn't that also suggest the idea of redundancy: the mechanism is connected together by three bolts in case some goon forgets to put one of them in; it will continue to work with only two until somebody comes along and notices the hole.

Reply: That's the other problem with the current approach to formal proof, it just gives you true or false.

Paul Wernick: It's not resilient.

Reply: It doesn't tell you how redundantly true it is. It's not very good, because you end up, even when you do have a system that's workable and secure, it's very fragile, because it relies on some very tendentious assumption that's right out on the edge of what you can comprehend: change slightly, and suddenly you've gone from one to zero (although it takes people a little while to work out how to exploit it).

Saša Radomirović: I think the problem we have here is the problem of trade-offs, we try to design security to be perfect, and then it isn't, and then we have a mess. So now we're saying, well maybe we want to design something that we acknowledge is not perfectly secure, and start squashing bugs. But eventually you will have this one bug that won't squash, and hitting that mole will cause terrible other consequences, and not quite get rid of it. Not knocking it out will have other bad consequences. So it might be that no-one should choose perfect security, but the problem is these trade-offs.

Reply: Yes exactly, but the point is that security is a trade-off, it's not an absolute. Shopkeepers all understand that shoplifting is a cost of doing business, they understand that they could reduce shoplifting to zero by using simple mea-

[5] This talk was given in 2010. The following year saw the publication of David Deutsch's book "The Beginning of Infinity" which makes a very similar point about social institutions as well as science, see especially the chapter "Unsustainable"; designing the perfect system is a non-goal, a good system is one where the problems are easy to identify and the system is easy to change.

sures involving dogs and armed guards, but that will also cost them all their business. So they say, well I would rather have shoplifting so long as it's below three percent of my turnover. Banks do have that approach to security, but not in a good way.

Paul Wernick: Exactly, they pass it onto their customers.

Reply: Yes, they pass it onto their customers, but the point is the customers have no choice. But if I'm a small grocer and I pass it onto my customers, if I'm bad at it they'll go and shop somewhere else. But our clients don't even have a choice.

Joseph Bonneau: Yes, I guess the question can be, can you point to a real world example of something, some system that's been vaguely designed in this way successfully?

Sandy Clark: Joseph, your immune system is a vital defence, it's not perfect, but it moderates itself as needed for attacks the best way it can.

Joseph Bonneau: I'm going to rephrase, something designed by people. Because it seems like some dispensables that banks actually took when designing EMV is that there's a million different options, and they can put leverage theoretically, if certain movements start happening, and they can check for this, and they can stop accepting that, and I think we'd all say that hasn't worked out well.

Reply: No, but people have said about some of the IPSec protocols that it's almost as if they were holding back on a release because they wanted to see the old holes exploited before they deployed the next set of features. So there is an underground view that they are already playing this game, possibly covertly, or possibly even subconsciously.

Jonathan Anderson: This is also an argument for defence-in-depth things, because all of a sudden it doesn't matter too much if there's a buffer overflow in your jpeg library as long as it's being run inside a chromium sandbox, then it gives you a little bit of time to try and whack that particular mole.

Reply: Yes, so the fact that you have to hold off on whacking a particular mole doesn't mean you've got no security at all.