HUMANS ARE SMART DEVICES, BUT NOT PROGRAMMABLE: INCENTIVE-CENTERED DESIGN FOR SECURITY

JEFFREY K. MACKIE-MASON

Security problems are incentives problems. Technology comes second.

If Eve and Mallory did not want to increase their wealth at the cost of decreasing mine, I would not be at risk for identity theft.[1] My identity doesn't get stolen and my bank account drained *because* Bob made a mistake in C operator precedence. Bob builds technology to try to keep Eve outside the gates, because she wants to get in. Bob's technology may fail, but it's the *want* that is foremost.

Why does this somewhat obvious point matter? There are miscreants out there, criminals and vandals and political anarchists and the occasional socially-maladjusted teen who is bored. Shouldn't we just take that as given, and focus on building technology to meet this design spec: "keep out bad guys"?

The thing is, the motivations matter. For one thing, it's why the bad guys keep coming, and why they expend (sometimes very considerable) resources to climb over or dig under or unhinge the gates. And there is the first fundamental design design principle that comes from paying attention to incentives: the amount we pay Bob to build the wall higher (the gate thicker, the moat deeper) should depend on *how motivated* Eve is to get inside: how many resources will he throw against us? Is it really *Hannibal ad portas*?

Human motivations matter for design in more subtle ways as well. What does Eve want to do when she gets inside the gates? How, for example, does she want to use my identity if she obtains it? Perhaps we should not work so hard to stop her from getting my identity, but instead design our systems so that she can gain little once it is hers.[2] Or should we make the penalties so high for illicit use that she doesn't want my identity even if she can get it (the "death penalty for parking violations" solution)? Then there are the motivations of the Carol the gate keeper: we may pay Bob and his many relatives to build a very thick gate, but Eve easily gains entry by making Carol a side payment (we can call it a "bribe") to provide a postern key.

Date: Version: May 27, 2009.

My thinking on these topics owes much to my colleagues and students in the Incentive-Centered Design Lab at the School of Information, University of Michigan, especially my co-author on security topics, Rick Wash. See http:/www.si.umich.edu/icd/wiki/.

[1]Yes, there might be other motivations for stealing my identify, but that doesn't change the point: I'm at risk only because other humans have some motivation to do so.

[2]Better to fight King Phyrrhus than Hannibal!

Miscreants are motivated humans; so are gate builders and gate keepers. But so are users: the castle's owner and good-hearted inhabitants, like Alice. When Alice is told the password so that she can be admitted at the gate, will she keep it secure? Will she and her friends form a voluntary community watch to report unfamiliar characters they see inside the gates? And if the screening system at the gate is too onerous for Alice and the other good folks, will they rise up in revolt?[3]

Last but not least introductory point: because they are motivated, humans are adaptive: whatever Bob builds, Eve will adjust her attacks, and with so many motivated (or under-motivated) humans in the system, she has many potential vectors. The war goes on.

Of course, these preliminary observations are not new. Ross Anderson argued that people are the weakest link in security [2]. The human-computer interaction community calls for attention to the usability and cognitive issues; see, e.g., [9] [16] [1] [17], and the ACM CHI 2003 Workshop on Human Computer Interaction and Security systems (`http://www.andrewpatrick.ca/CHI2003/HCISEC/`. There has been less attention to behavioral and motivational issues, although the fundamental nature of the problem was well presented by Anderson in [3]. I offer the use of incentive-centered design (ICD) as a focusing lens, and as a principled source of constructive advice. This focus on concrete design principles distinguishes ICD from much of the recent work in the economics of security.

1. What is ICD?

Incentive-centered design (ICD) is an emerging, *multidisciplinary* research methodology for designing systems whose performance depends unavoidably on the behavior of humans. The starting point is to recognize that human components in the system are smart, distributed — and crucially — *autonomous* components, with their own beliefs and *motivations*. Humans-in-the-loop as a (regrettable?) performance factor have been widely recognized since the early days of aerospace engineering. Only in recent years has a systematic effort to incorporate humans in the *design* loop as responsive components made headway, though bringing the human sciences of motivated behavior to bear on problems previously perceived primarily as non-organic engineering challenges.

There is precedent: the human cognitive sciences have participated in software (and artifact, or ergonomic) design since at least the early 1980s, resulting in the large and successful human-computer interaction (HCI) field. HCI has grown beyond its cognitive science roots, but bringing cognitive science to design was the key to its success. The ICD approach brings the sciences of autonomous motivated behavior to bear on design, with similar transformative possibilities.

To date, the primary contributions of motivation science to information system design have come from economics, social psychology and game theory, through theories, statistical and experimental results on individual responses to personal and social motivations, and on

[3]Just how much screening should TSA subject all passengers to at the airport?

inter-individual strategic awareness and behavior. Of course, though humans are smart devices, they are not programmable. What we learn from motivation science are systematic, predictable — though often context-dependent and always stochastic — behaviors in response to various incentives. For example, from managerial and labor economics we know about systematic differences in the way that people respond to different types of effort compensation (e.g., piece rates, fixed salary, winner-take-all tournaments) when outcomes are uncertain. From asymmetric information economics (and evolutionary biology and psychology) we know something about how high-quality individuals can signal their desirability. From social psychology we know predictable ways in which people respond to social comparisons and in-group identification. Below I illustrate how systematic motivated behaviors can be employed in security design.

Many incentive problems have structure that we can exploit for design. I will describe a few common incentive problems and illustrate their application to canonical security concerns. I refer to these as *keeping scoundrels out*, *getting good guys to help*, and *discouraging delinquency*.[4]

2. Keeping scoundrels out

This is probably the most familiar type of security problem: how do we prevent a bad actor from getting access to a resource, whether it be an information (a secret), monetary, or physical (e.g., use of my CPU) resource?

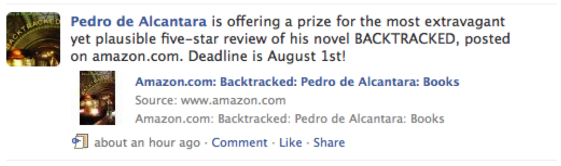
A key feature of the scoundrel problem is that the scoundrel knows who she is before acting, but the system manager does not. This is known in economics as "hidden information" (from its origins in studies of insurance, also called "adverse selection"). The incentives problem: how to motivate the scoundrel to reveal herself?[5]

Keeping scoundrels out is a common problem in social computing settings because contribution platforms tend to be open. Email, for example, is a rather open system and self-aware spammers regularly use it to distribute unsolicited bulk advertisements. Another scoundrel problem is manipulation: authors who pay for favorable Amazon reviews (see figure 1 or hackers who rig the Time Magazine Top 100 Poll (see `http://musicmachinery.com/2009/04/27/moot-wins-time-inc-loses/`).

Familiar security technologies for keeping scoundrels out are in fact *incentive-centered design* solutions. Consider the use of password-controlled authentication [14]. The gate keeper challenges the potential entrant for a valid password. The goal is to separate good guys from scoundrels. One necessary condition for success is that obtaining a valid password is too costly to be worthwhile for a scoundrel. Notice that this is an incentives

[4]I use "guys" in its gender-neutral connotation. "Gals" just doesn't work.

[5]The scoundrel need not admit scoundrelhood in a public confession: "revelation" often occurs by virtue of *not* acting: for instance, by not trying to go through the gate. This reveals, or *separates*, because the good guys do enter.


Pedro de Alcantara is offering a prize for the most extravagant yet plausible five-star review of his novel BACKTRACKED, posted on amazon.com. Deadline is August 1st!

Amazon.com: Backtracked: Pedro de Alcantara: Books
Source: www.amazon.com
Amazon.com: Backtracked: Pedro de Alcantara: Books

about an hour ago · Comment · Like · Share

FIGURE 1. Manipulating Amazon book reviews

proposition: it is not *impossible* to obtain a password, just too costly for the scoundrel to want to incur the cost. [6] There is at least one other necessary incentive condition for a good password system: *it must not be too costly* for the good guys to obtain, maintain and employ valid keys; otherwise, they will not bother entering the gates either! ICD solutions of this type are known as *screening mechanisms*; additional necessary conditions for success, and their properties, are well-developed in the economics literature. Proof-of-work security technologies, such as those proposed and studied to keep out spammers [5], are also incentives-based screens [15].

The ICD approach is illuminating when applied to one family of currently popular challenge-response system: the CAPTCHA, intended to prevent automated agents from hijacking various online resources [12]. The second part of the spelled-out acronym ("to tell computers and humans apart") reveals that its goal is solving a hidden information problem: the bots know they are scoundrels, but hope to pass themselves off as humans. As required by screening theory, the task (recognizing the graphically distorted text, for example) must be harder for the bots than for the humans. CAPTCHAs also display another important feature of effective screens: the cost incurred *in fact* is higher for the humans, because the humans *solve* the puzzle, whereas the bots "reveal" themselves by not bothering to try.

Another standard feature for good screens is that they satisfy the Spence-Mirrlees single-crossing condition: essentially, not only must the response cost be higher for scoundrels, but the incremental cost of responding as the challenge is increased in difficulty must also be higher for scoundrels. It is this latter condition that leads to CAPTCHA technology arms races: bot creators look for ways to make the incremental solving cost for bots close to the same as for good guys. If they succeed, then the only way to keep the bots out is to make the challenge so tough that the good guys will no longer be sufficiently motivated to solve the puzzle, generating a success disaster. Much of the effort to come up with better CAPTCHA tests is directed to increasing the incremental cost differential between bots and humans, though the technology designers do not always realize that this is the condition they need to satisfy. One of the most striking attack vectors on CAPTCHAs is a direct assault on this condition: rather than improve text-recognition technology enough that bots can solve puzzles almost as easily as humans, some bots simply hire real humans to do the work for

[6]Remember, obtaining a teraflop computer to find a 128-bit key is only one costly approach: another might be to kidnap and torture, or merely blackmail, a legitimate key holder, the cost of which might include expected jail time.

them, notably by offering "porn bribes" to humans that solve a CAPTCHA the bot then uses to get access to a target system[4]. See [15] for a formal analysis of CAPTCHAs as an ICD problem.

3. Getting good guys to help

Botnets, or networks of hacked machines under the control of a single attacker, have been used for a number of malicious purposes [10], including distributed denial-of-service attacks, and for sending spam and phishing emails. This is serious; for example, nine of out ten email messages are spam, and 80% of those messages are being sent through botnets [8]. Zhuge et al. [18] report that more than half of recent botnet exploits are executed through three well-known flaws for which patches were published and available for months. This is a painfully familiar problem: why don't users patch their systems?

To understand the central incentives problem, let's look a bit deeper. Botnets frequently hide by only using a system when it is idle. Thus, the machine's user bears little of the cost of having a zombie. He may not even notice his machine has been infected, and if he does, he may not have much incentive to clean it. Thus, the botnet is a classic economic externality problem: the costs are borne by others (who suffer from spam, denial of service attacks, etc.), not by the person who is "causing" the problem. That is, the machine owner has little incentive to make the effort to patch.

There are many incentive mechanisms to induce people to take socially desirable action when their private incentives to do so are insufficient. One, of course, is to fine or punish them if they do not. Another is to subsidize or pay them to do so. These schemes are often worth designing into a security system, however, they have shortcomings in distributed network settings, for which there is no central authority that can mete out punishments, or that wishes to underwrite subsidies. For such cases, another category of incentives to induce non-scoundrels to help out is getting increasing attention: those that address the *private provision of public goods* problem. A public good in general is one that is *nonrivalrous*: many can benefit from it without exhausting its value. Patched computers in a network are an example: every one else in the benefit benefits from the reduced number of zombie machines controlled by bots. Incentive designers are proposing and testing various *contribution* mechanisms to induce users to contribute their effort to the common good. My student Rick Wash and I, for example, have been designing non-monetary exclusion mechanisms to induce users to share their security knowledge, creating a "social firewall" [13].

4. Discouraging delinquents

It is sometimes desirable to keep bad actors outside of the system in the first place, but the cost of doing so is not always worth it. Then, we want to discourage them, once inside, from acting (too) badly. One example is the legitimate user tempted to go bad: for example, the financial manager who considers embezzling, or gate keeper Carol who was proffered a bribe to open a back door. Or, when the system is the Internet, it is unreasonable to keep bad actors out, yet we want to discourage them from launching, say, denial of service attacks.

These are known as *hidden action* (or moral hazard) problems. It is not possible to sufficiently monitor the behavior of some actors to prevent them from delinquency. In one well-studied class of problems, we can observe an outcome that is statistically correlated with the unobservable behavior: for example, the sales of a firm are correlated with the imperfectly observable effort by salespeople.

Then there are many design strategies that involve assigning some share of the risky outcome to actors whose behavior is unobservable. If enough of the cost of a bad act can be shifted to the actor, then temptations with expected payoffs smaller than the cost will be foregone. Thus we see those in the best position to harm an organization through harmful (or even simply insufficient) effort most dependent on *ex post* bonuses and options, so that they lose the most from malfeasance. Some responsible managers are fired automatically after a big enough failure on their watch, without trying to prove responsibility.

In other situations, the delinquency might be observable, but only after the action. This is largely the same as the hidden information problem: these bad actors know that they are bad but the system manager does not. However, in addition to thinking of *ex ante* (signaling and screening) mechanisms for preventing the malfeasance before it occurs, we might also consider *ex post* mechanisms that punish the malfeasance sufficiently after it is discovered that the potential bad actor simply chooses not to act in the first place. One such security design is criminal law: observe, convict and punish. Another common form of this incentive structure is to require the actor to post a substantial bond that is forfeit upon bad performance. Such bonds need not be monetary: for example, Friedman and Resnick showed that persistent pseudonyms in a reputation system such as eBay uses function as bonding against fraud or below-standard service [6]. In another familiar security setting, Loder, van Alstyne and Wash proposed a repudiable bond mechanism to discourage spammers [7].

Another clever approach to such problems, that illustrates the richness of the incentive-centered design paradigm, is to shift the cost of the malfeasance to a third party who is in a better position to block or discourage the bad action. For example, individual computer users, in either an enterprise or at home, may have insufficient personal incentive to bear the cost of learning and implementing a security strategy to prevent their machine from becoming a botnet zombie. If it is possible for a system manager (say, the Federal Communications Commission) to shift the burden to the enterprise or ISP, say by imposing a heavy fine if a denial of service attack is launched from machines in their domain, these organizations may be able to take security management control of the machines or provide the users with the needed motivations (like the threat of lost wages) to resolve the problem. This example (due to [11] is an instance of the well-known incentive strategy of assigning property rights (or liability) to obtain an efficient (or at least better outcome).

All of these mechanisms face limits, of course. Whether it be correlated outcomes or behaviors that we observe *ex post* but imperfectly, we are faced with the trade-off between Type I and Type II errors. For example, we can raise the punishment enough to discourage bad acts even with some Type I probability of not catching the criminal or proving the crime, but only at the unavoidable cost of an increase in the cost of Type II errors (wrongful convictions), which is one explanation for why we do not see the death penalty for parking violations. Nonetheless, assigning *ex ante* uncertain burdens (punishments, bond revocations, etc.) to *ex post* observables is one important design tool in the security kit.

5. Open the gates

Incentive solutions are often overlooked by technology designers, who tend to focus on making bad acts impossible (thicker gates) rather than on making them undesirable (cauldrons of boiling oil maintained above thinner gates). Even when using technologies that fundamentally rely on human responses to incentives — such as passwords and CAPTCHAs — by not understanding the underlying incentive structure of the problem, effort may be directed to the wrong feature, or otherwise applied inefficiently.

Given the title of this department, I have focused on design advice we can harvest from incentive economics. As rich and well-studied a toolkit as economics provides for incentives problems, as I mentioned on we can learn much from other branches of the sciences of motivated behavior as well. Social psychology is particularly rich in experimental results on the responses of individuals to a variety of social signals, comparisons, identifications, and so forth.

Security is fundamentally an incentives problem: the payoffs from applying the sciences of motivated problems to security engineering will be enormous.

References

[1] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, December 1999.

[2] R. Anderson. Why cryptosystems fail. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, pages 215–227. ACM Press, 1993.

[3] R. Anderson. Why information security is hard – an economics perspective. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)*, December 2001.

[4] C. Doctorow. Solving and creating captchas with free porn, 27 January 2004.

[5] C. Dwork and M. Naor. Pricing via processing for combatting junk mail. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, Lecture Notes in Computer Science, pages 139–147, London, UK, 1993. Springer-Verlag.

[6] E. J. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, June 2001.

[7] T. Loder, M. V. Alstyne, and R. Wash. An economic solution to unsolicited communication. *Advances in Economic Analysis and Policy*, 6(1), 2006. Available from `http://www.bepress.com/bejeap/advances/vol6/iss1/art2`.

[8] D. Sieberg. Experts: Botnets no. 1 emerging internet threat. *CNN.com, Technology Section*, January 2006.

[9] S. W. Smith. Humans in the loop: Human-computer interaction and security. *IEEE Security & Privacy*, 1(3):75–79, 2003.

[10] The Honeynet Project. Know your enemy: Tracking botnets. Published on the Web.

[11] H. R. Varian. Managing online security risks. *The New York Times*, 1 June 2001.

[12] L. von Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard AI problems for security. In *Proceedings of EUROCRYPT 03*, Lecture Notes in Computer Science, 2003.

[13] R. Wash and J. MacKie-Mason. A social mechanism for anti-botnet security. In *Workshop on Information System Economics (WISE)*. WISE, 2008.

[14] R. Wash and J. K. MacKie-Mason. Incentive-centered design for information security. In *USENIX Hot Topics in Security (HotSec 06)*, Vancouver, BC, 2006. USENIX. Available from `http://www.usenix.org/events/hotsec06/tech/wash.html`.

[15] R. Wash and J. K. MacKie-Mason. Security when people matter: Structuring incentives for user behavior. In *Ninth International Conference on Electronic Commerce (ICEC-07)*, pages 7–14, New York, NY, USA, 2007. ACM.

[16] A. Whitten and J. D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Proceedings of the 8th USENIX Security Symposium*, August 1999.

[17] K.-P. Yee. User interaction design for secure systems. In *Proceedings of the International Conference on Information and Communication Security (ICICS)*. Institute for Infocomm Research / Chinese Academy of Sciences, 2002.

[18] J. Zhuge, T. Holz, X. Han, J. Guo, and W. Zou. Characterizing the IRC-based botnet phenomenon. Technical Report TR-2007-010, The Honeynet Project, 2007.