

Incentive Centered Design and Information Security

Rick Wash and Jeff MacKie-Mason

School of Information
University of Michigan

DIMACS Workshop on Information Security Economics
January 18, 2007

Humans in Security

Important in Security

- Choose what technologies to use
- Make security policies
- Evaluate security information
 - And choose what information to provide to others
- Take security actions
- Use system

Typical Approaches

- Outside the design loop
 - Make assumptions about how they will act
- Throw up hands; people are unpredictable
 - Design for the worst case

Autonomous Actors

- Non-programmable
- Predictable

Incentive Centered Design

Basic Idea

Humans respond to incentives in strategic settings

Mathematical Models

- Describe possible choices
- User preferences on those choices
 - Different types of users
- Strategic interactions
- Derive behavior from system design

Body of Knowledge

- 30 years of study
 - Economics
 - Psychology
- Integrate tools into what we are already doing

Other Tools

- User studies
- Empirical results
- Validation methods

Example: Screening

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

Model Setup

- Two types of users, Good and Bad
- Both want access to a system
- System designer prefers Good to Bad
- System can't easily tell users apart

Example: Screening (cont.)

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

Screening Task

- Systems asks them to perform a *screening* task
(e.g. compute hard problem)
- Task can be done with different intensities
(different difficulties of problems)

Example: Screening (cont.)

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

Task Properties

In order to work as a screen:

- ① Cost is increasing in task intensity
- ② Cost is convex (greater than linear)
- ③ Cost is greater for Bad than Good
- ④ Incremental cost of harder tasks is higher for Bad than Good
 - Theory also specifies how much harder it needs to be

Example: Screening (cont.)

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

Proof of Work Properties

Do proposed proof of work systems satisfy these properties?

Prop 1-2: Uses crypto problems that are super-polynomial

Prop 3-4: Legit users have spare CPU cycles that are virtually free

Prop 3-4: Spammers have to upgrade since they are running at capacity

Example: Screening (cont.)

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

Proof of Work Literature

Dwork and Naor propose CPU-bound functions that might satisfy Prop. 1-4

Abadi et. al. introduce memory-bound functions that better satisfy Prop. 1-4, particularly 4

Laurie and Clayton argue that Prop. 3 and 4 don't hold because of botnets

Liu and Camp propose a reputation system that increases difference between Good and Bad (Prop 3-4)

Example: Screening (cont.)

Unsolicited Email and Proof of Work

What constraints must a proof of work system satisfy to be effective at distinguishing between legitimate email and spam?

User Behavior

How will users react to this design?

Result 1: Bad users go away

(No more spam sent)

Result 2: Good users still want to access the system

(Legitimate email is still sent)

Result 3: Good users gain access but perform a harder task

(Normal people's email is delayed while computing)

Incentive Problems

Keeping the Bad Stuff Out

- Log-ins
 - Passwords
 - CAPTCHA's
- Spam
- Spyware

Getting the Good Stuff In

- Labeling vulnerabilities
- Knowledge workers
- Botnets and home users
- Privacy-enhancing technologies