CSE 127: Computer Security

Security Concepts

Deian Stefan

Slides adopted from Kirill Levchenko and Stefan Savage

Computer Security

Analysis and protection of computer systems in an adversarial setting

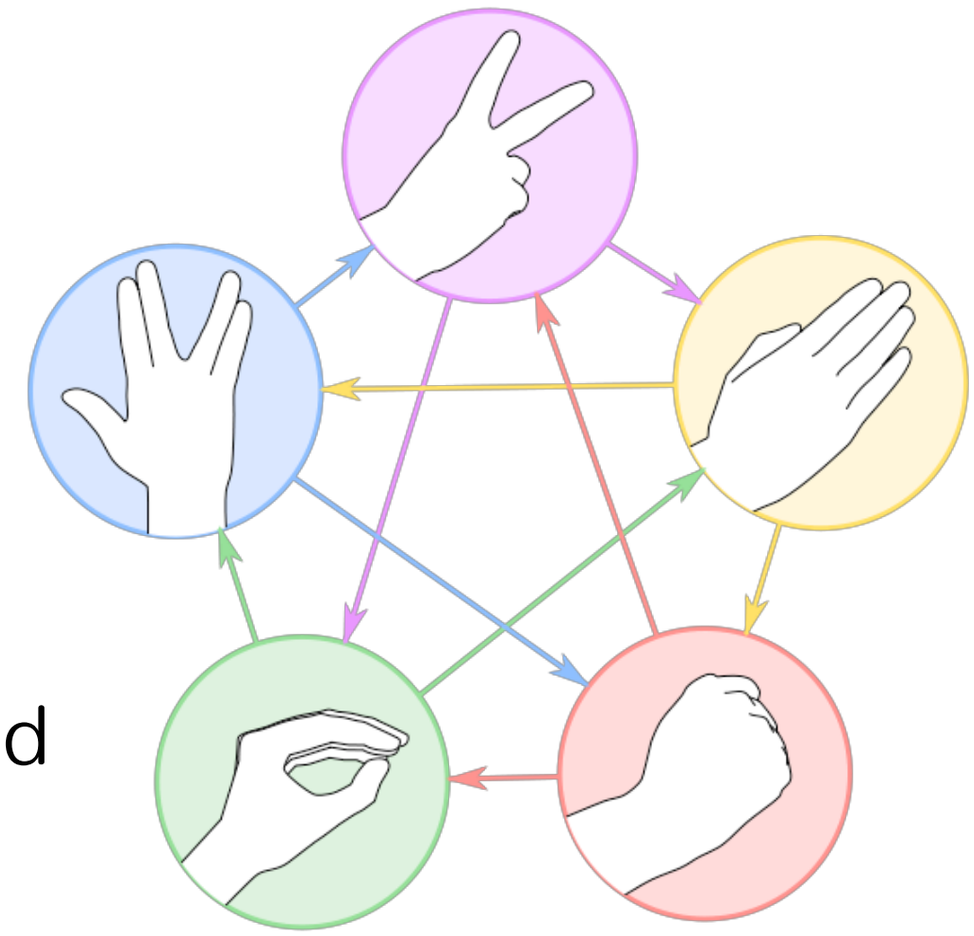
What is an adversary?

- An **adversary** is someone who seeks an outcome detrimental to your interests
- We assume rational adversaries
  - I.e., they act to maximize their payoff

Adversarial Setting

Example: Games

- Structured adversarial setting
- Opposing objectives well-defined



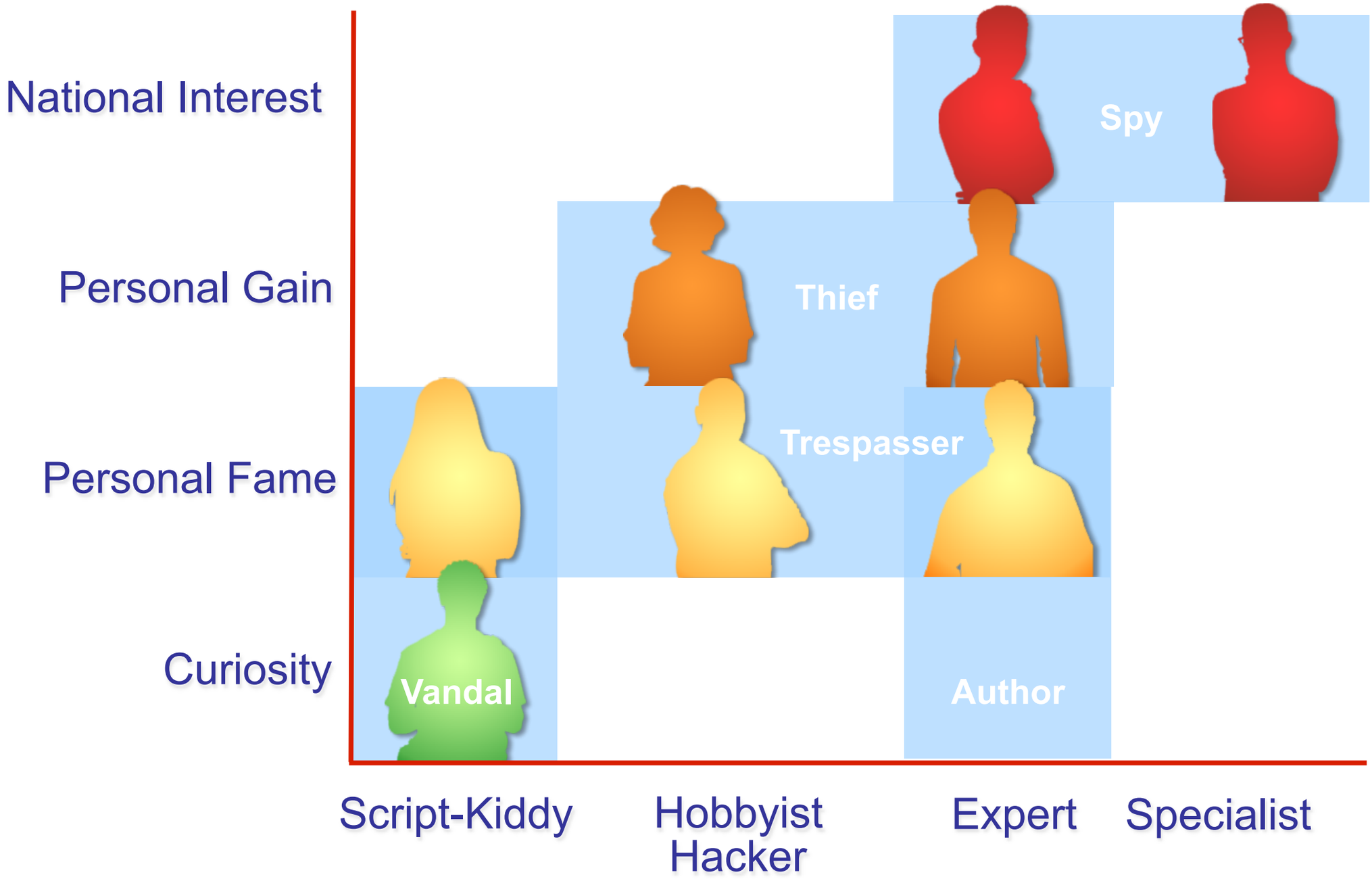
Adversary or Attacker?

- An adversary becomes an attacker when they act in a way that is detrimental to your interests
- Distinction is not hugely important
  - Adversary often used in cryptography
  - Attacker often used in computer security
  - Both ultimately mean “bad person”

How do we define attackers?

- Motives:
  - Curiosity
  - Fame
  - Money
  - National interest
- Resources:
  - Time, money, and training

Classes of Attackers



From David Aucsmith, Microsoft.

Computer Security

Analysis and protection of computer systems in an adversarial setting

What do we mean by protection?

- Protection of systems against an adversary
  - **Secrecy:** Can't view protected information
  - **Integrity:** Can't modify protected info or process
  - **Availability:** Can't deny access to system for others

Computer Security

Analysis and protection of computer systems in an adversarial setting

Traditional definition

Security specialists (e.g., Anderson [6]) have found it useful to place potential security violations in three categories.

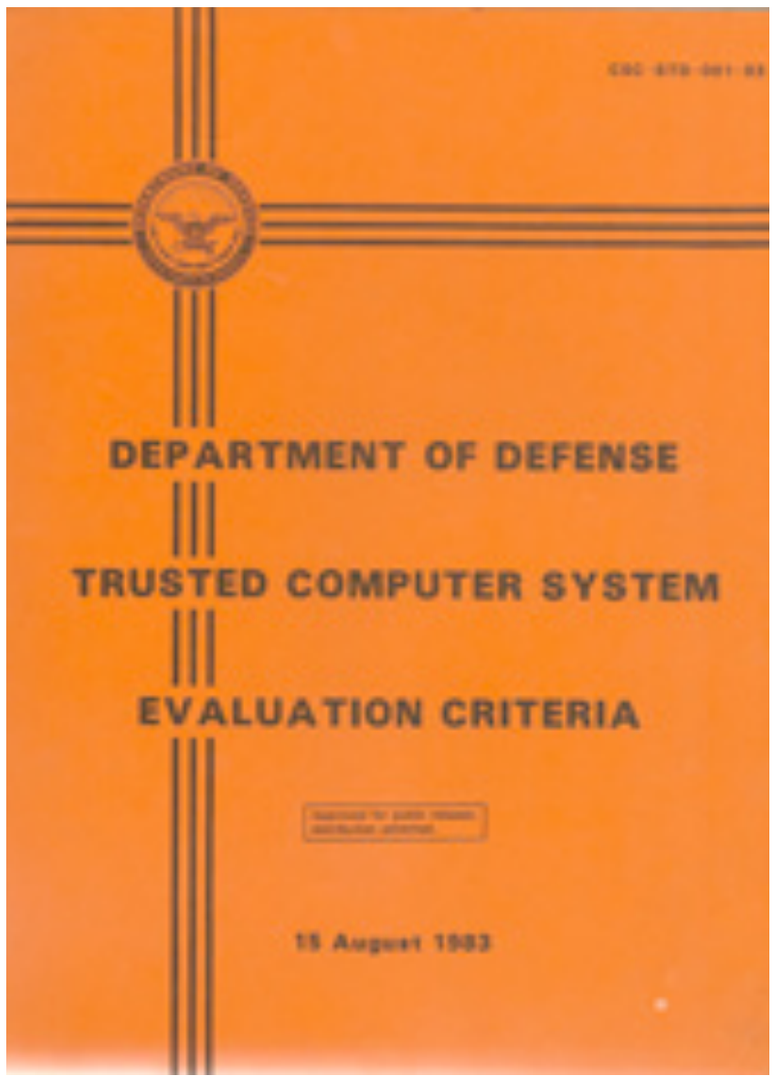
1) Unauthorized information release: an unauthorized person is able to read and take advantage of information stored in the computer. This category of concern sometimes extends to "traffic analysis," in which the intruder observes only the patterns of information use and from those patterns can infer some information content. It also includes unauthorized use of a proprietary program.

2) Unauthorized information modification: an unauthorized person is able to make changes in stored information—a form of sabotage. Note that this kind of violation does not require that the intruder see the information he has changed.

3) Unauthorized denial of use: an intruder can prevent an authorized user from referring to or modifying information, even though the intruder may not be able to refer to or modify the information. Causing a system "crash," disrupting a scheduling algorithm, or firing a bullet into a computer are examples of denial of use. This is another form of sabotage.

Traditional definition

"Orange Book"1983



Authorization

- What is authorized?
  - Allowed by the operator of the system
- Clear when there is a central authority and explicit policy
  - E.g., DoD time-sharing systems
- Can be awkward to apply in some settings
  - E.g., Click fraud malware on your smart phone

Secrecy (or confidentiality)

- Prevent unauthorized access to information
- Real world scenarios where you want secrecy?
  - E.g., SSNs, power plant designs, nuclear codes
- Scenarios involving computers?
  - E.g., password managers and email clients

Integrity

- Prevent unauthorized modification of information, process, or function
- Real world scenarios where you want integrity?
  - E.g., increasing bank account balance without deposit
- Scenarios involving computers?
  - E.g., downloading files from the net

Information Integrity

- The focus of traditional computer security has been protection of information
- Why not just say integrity is “protection of information?”
  - What about control or function of system?
  - Everything is information!

Authenticity

- Prevent impersonation of another principal
  - Some authors call this origin integrity
- Real world scenarios where you want authenticity?
  - E.g., depositing checks
- Scenarios involving computers?
  - E.g., login

Does integrity include authenticity?

A: Yes, B: no

Availability

- Prevent unauthorized denial of service to others
- Real world scenarios where you want availability?
  - E.g., ATMs, bathrooms
- Scenarios involving computers?
  - E.g., network denial of service, IoT heaters


Insert Here
VISA
VISA
CANCEL
CLEAR
ENTER

Examples

- Which security property is violated if someone ...
  - ... unplugs your alarm clock while you're sleeping?
    - A: secrecy, B: integrity, C: availability, D: none

Examples

- Which security property is violated if someone ...
  - ... change the time on your alarm clock?
    - A: secrecy, B: integrity, C: availability, D: none

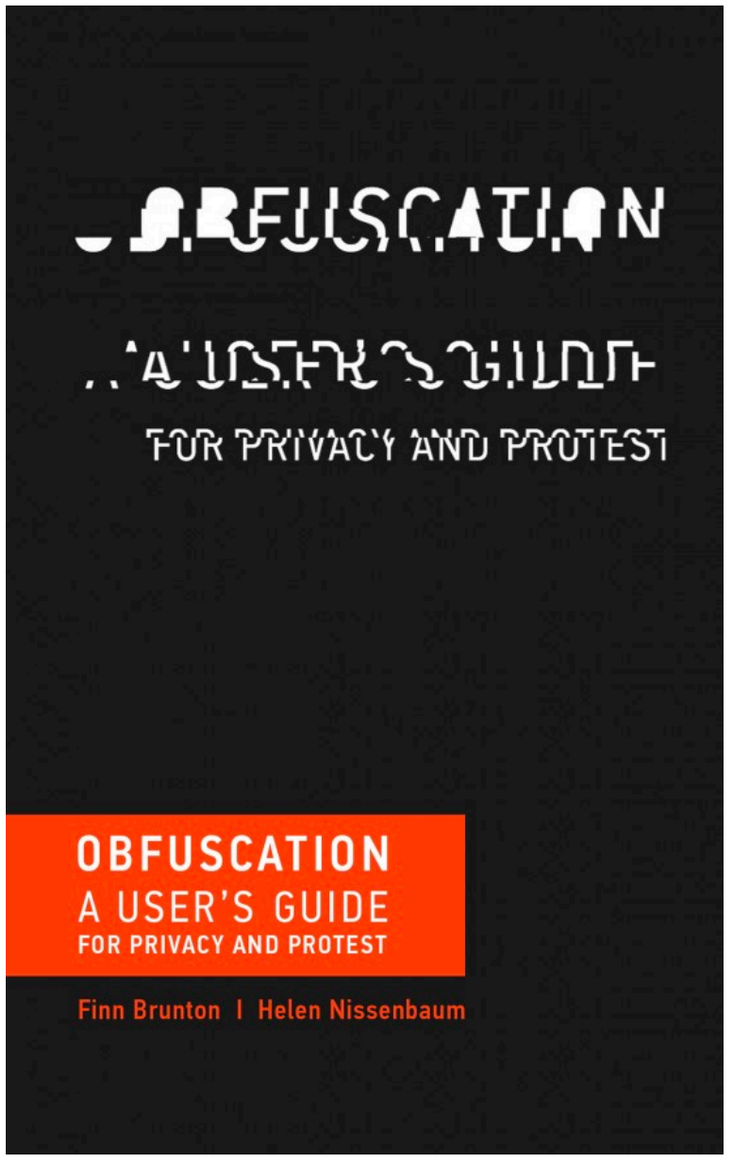
Examples

- Which security property is violated if someone ...
  - ... installs a camera in your room?
    - A: secrecy, B: integrity, C: availability, D: none

Privacy

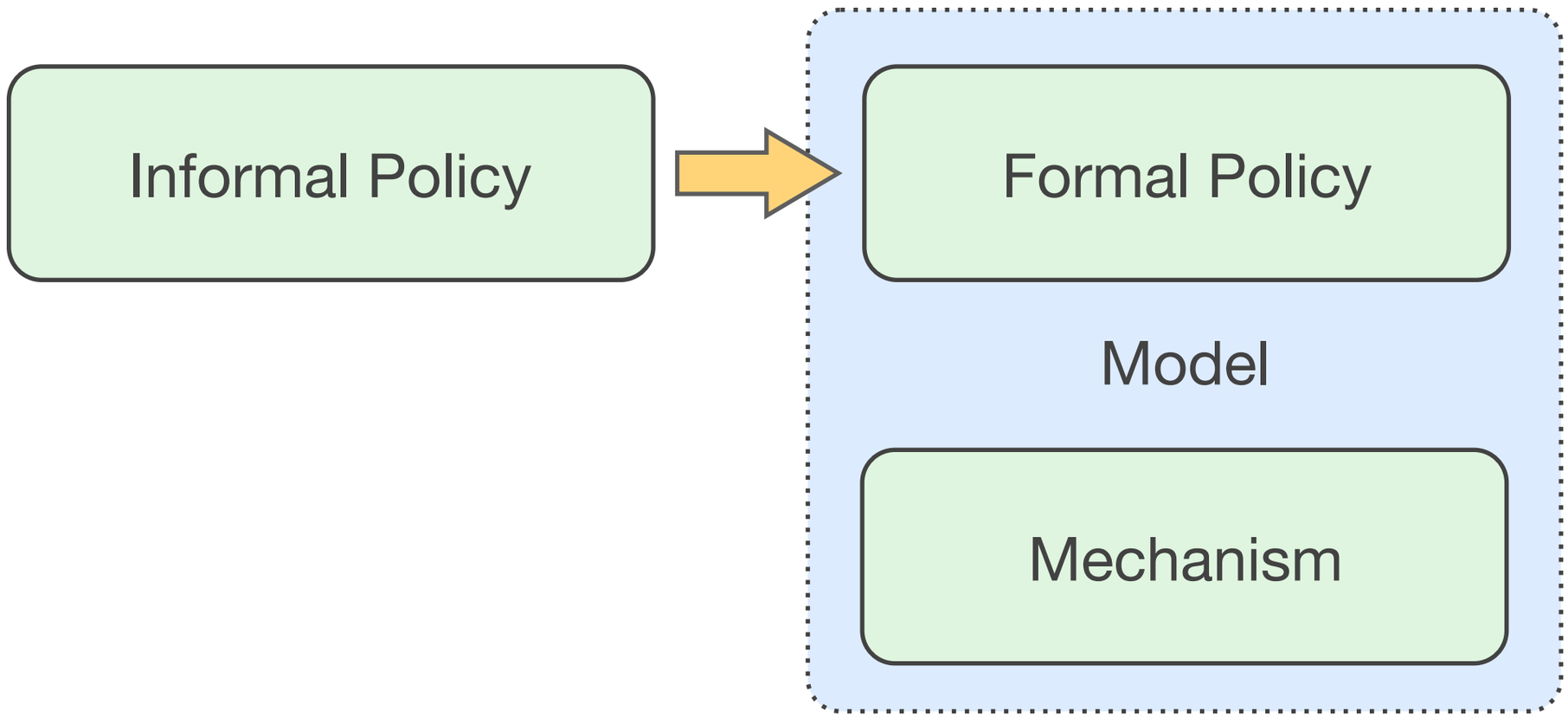
- A person's right or expectation to control the disclosure of their personal info
- What's the diff between privacy and secrecy?

Great book:


OBFUSCATION
A USER'S GUIDE
FOR PRIVACY AND PROTEST
Finn Brunton | Helen Nissenbaum

What can we do we these concepts?

Putting concepts to use



Putting concepts to use

- **Model:** Abstractions for expressing policy/ security mechanism
- **Policy:** Set of allowed actions in a system
- **Mechanism:** Part of system responsible for enforcing the security policy

Assurance

- Procedures ensuring that policy is enforced
- How can we get assurance?
  - E.g., testing, code audits, formal proofs
- Why do we care about assurance?
  - Justifies our trust in a system

Trust

- Belief that system or component will perform as expected or required
  - Trusted: assumed to perform as expected/required
    - If trusted component fails ⇛ no security guarantee
  - Trustworthy: will perform as expected/required
    - Enough evidence (e.g., clean API) that it will perform as expected
  - Untrusted: may not perform as expected/required
    - Don't rely on it to do the right thing

Trusted Computing Base

- Part of the system assumed to function as required
- Malfunction in TCB can lead to loss of protection
- Assurance gives us confidence of our trust of TCB

Trusted Computing Base

- Want TCB to be as small as possible
  - Fewer things you trust ➠ fewer things you assume to do the right thing
  - Ideal scenario: build systems that preserve security (secrecy, integrity, availability) out of untrusted components

Trusted Computing Base

- In Unix?
  - CPU, memory, boot disk, operating system kernel, operating system utilities (e.g. passwd)
- On a Web server?
  - OS, libc, HTTP parser, etc.
- What assurance do we have for above?
  - How do we know TCB will work as required?