Science of Security: Historical Perspective

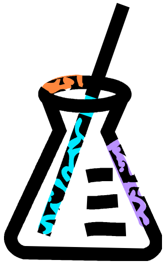
Fred B. Schneider
Samuel B Eckert Professor of Computer Science

Department of Computer Science
Cornell University
Ithaca, New York 14853
U.S.A.

"Science" is a moving target

Science:

- An organized body of knowledge gained through research **-versus-**
- System of acquiring knowledge based on the scientific method **-versus-**
- Laws or theories that are predictive.



A Science Of Security?

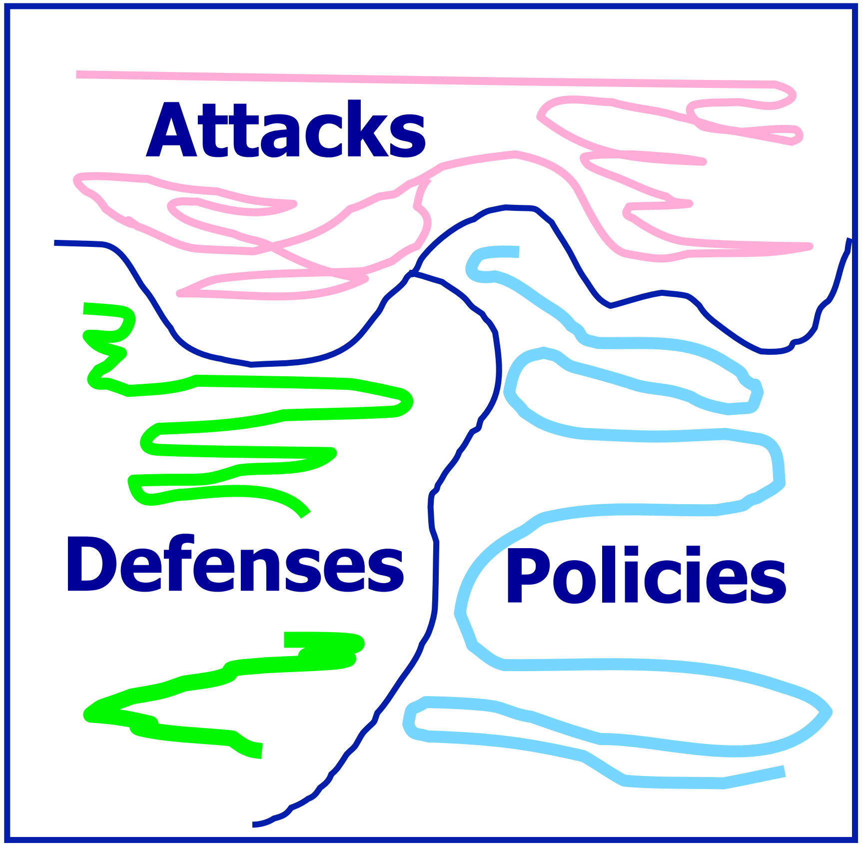
A **body of laws** that are predictive…

- Transcend specific systems, attacks, and defenses.
- Applicable in real settings.
- Provide explanatory value.
  - Abstractions and models
  - Connections and relationships. E.g.,
    - Cannot enforce policy P with mechanism M
    - Interface can leak b bits/sec

Kinds of Laws

- **Analysis**: Given an artifact, predict its properties…
  - Qualitative properties: What it does.
  - Quantitative properties: How well it works.
- **Synthesis**: Compose artifacts with given properties to obtain a new one with predictable properties.

Laws About What?



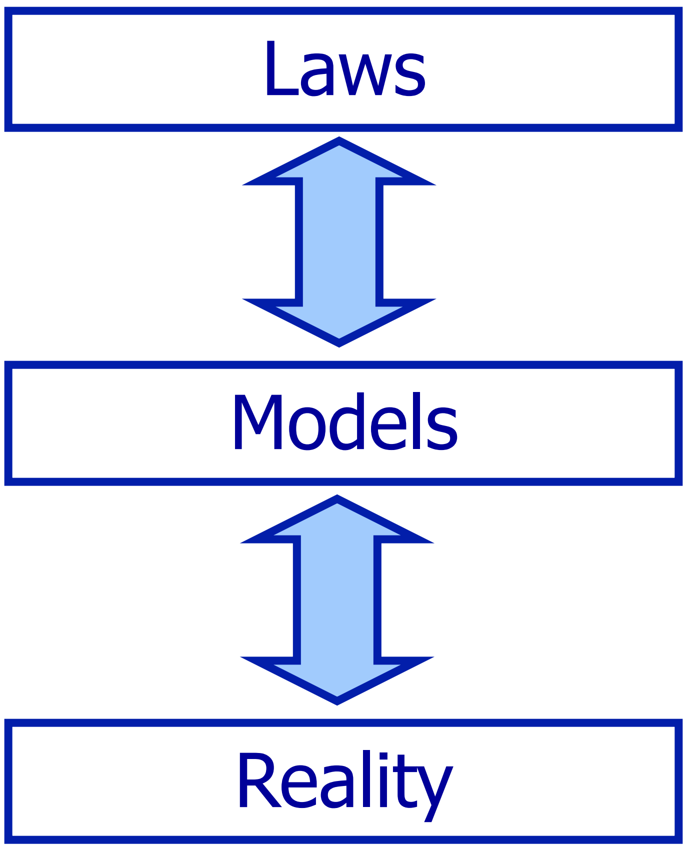
- Classes of policies
- Classes of attacks
- Classes of defenses

Relationships:

"Defense class D enforces policy class P despite attacks from class A."

"Defense D + Defense D' = …"

Laws versus reality?



Model → Law

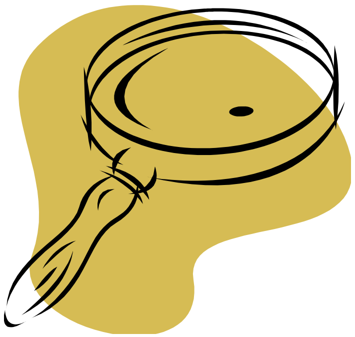
- Logic
- Mathematics
- Game theory

Reality → Model

- Measure and observe
- Hypothesize and experiment

Selections from history …
… through a Science of Security lens

- Authorization
  - Access control mechanisms
  - Information flow policies
- Integrity of mechanism
  - Reference monitors
  - Moving target defense (code obfuscation)



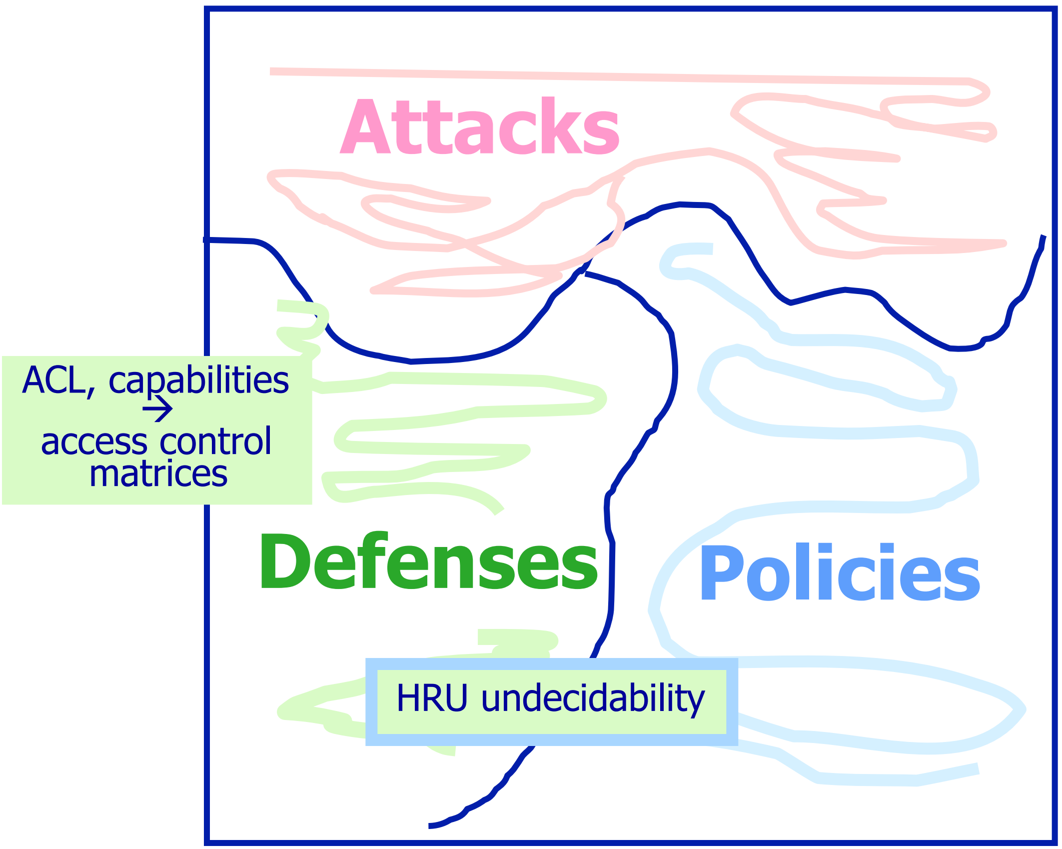
Reality → Model → Laws:

Access control mechanisms

- Reality:
  - Access control lists [CTSS, Mulitics 1965]
  - Capabilities [MIT PDP-1, 1967]
- Model
  - Access control “matrix” (=relation) [Lampson 1971]

Reality → Model → Laws:
Access control mechanisms

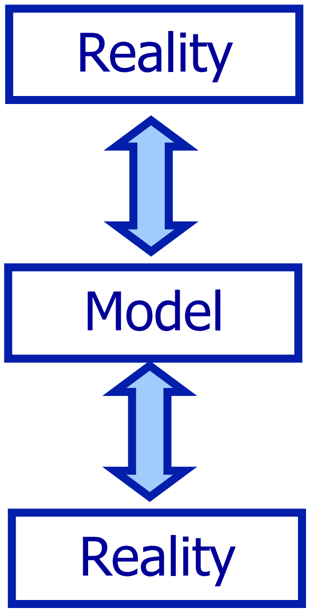
- Reality:
  - Access control lists [CTSS, Mulitics 1965]
  - Capabilities [MIT PDP-1, 1967]
- Model
  - Access control "matrix" (=relation) [Lampson 1971]
- Laws: Can A perform op on Obj?
  - Mono-operational is decidable
  - General case: Reduces to Halting Problem


Attacks
ACL, capabilities
→
access control
matrices
Defenses
Policies
HRU undecidability

Reality → Model → Laws:

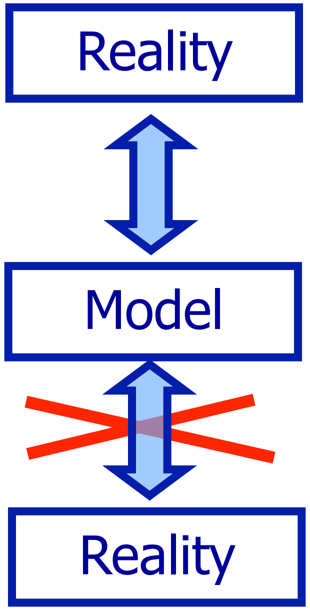
Models of kernel-enforced policies

- Reality: DoD “Need to know”
- Model: [Walter et al, Bell-LaPadula ‘73]
  – Objects have labels (U < C < S < TS)
  – Principals have clearances (U < C < S < TS)
  – Read-down and write-up authorized.
- Laws
  – … xxx is a **secure system** if and only if …



Reality → Model → Laws:

Models of kernel-enforced policies

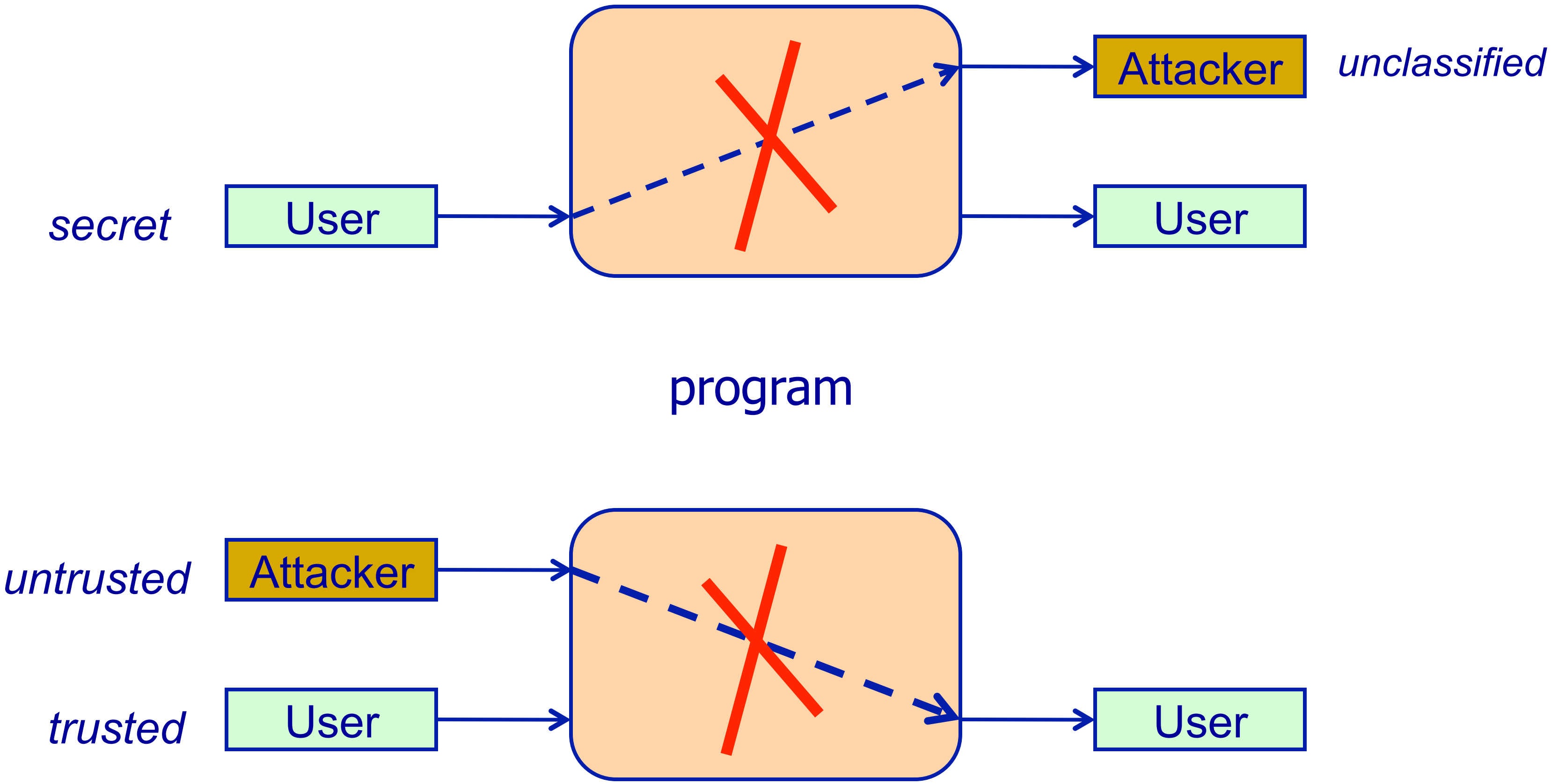


- Reality: DoD "Need to know"
- Model: [Walter et al, Bell-LaPadula '73]
  - Objects have labels (U < C < S < TS)
  - Principals have clearances (U < C < S < TS)
  - Read-down and write-up authorized.
- Laws
  - … xxx is a **secure system** if and only if …
  - No it isn't:
    - Not all transitions specified [McLean's system Z, 1985]
    - **Lab**(F(x,y)) ≤ **Lab**(x) /\ **Lab**(F(x,y)) ≤ **Lab**(y)
      - E.g., **From:** P, P → Q **Infer:** Q

Model → Laws:
Onward to integrity …

- Model: [Biba 77]
  - Objects have labels ($T < U$)
  - Principals have clearances ($T < U$)
  - Read-down and write-up authorized.
- Laws:
  - Confidentiality and integrity are duals.

Confidentiality and Integrity



The Duality!

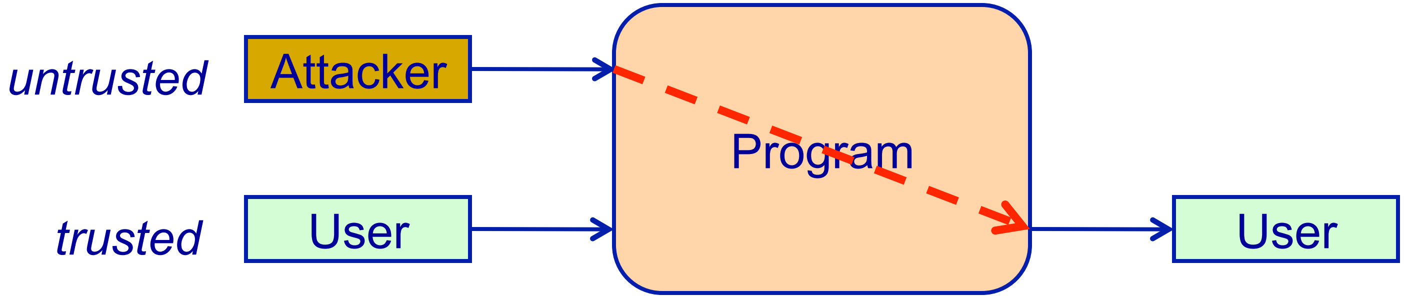
*Joint work with Michael Clarkson. [Computer Security Foundations, 2010]

Attacker consequences:

– Contamination (dual of leakage)

- Output := (t, u)

… Predict untrusted input **from** *trusted input and trusted output*



The Duality is incomplete!

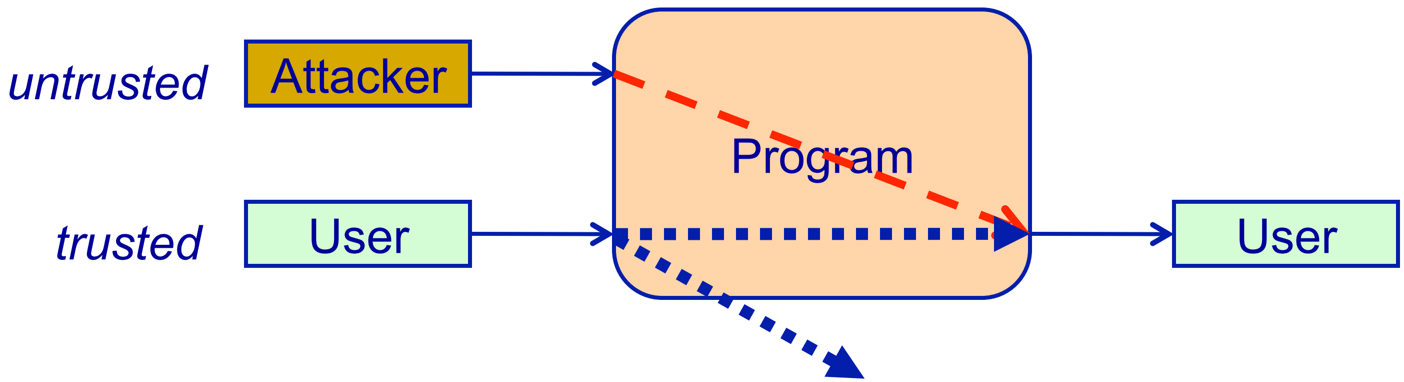
*Joint work with Michael Clarkson. [Computer Security Foundations, 2010]

Attacker consequences:

- Contamination (dual of leakage)
  - Output := (t, u)

  ... Predict untrusted input **<u>from</u>** *trusted input and trusted output*
- Suppression (trusted input suppressed from trusted output):
  - n := rand(); Output := t XOR n

  ... Predict trusted input **<u>from</u>** *trusted output.*
- Both contamination and suppression
  - Output := t XOR u



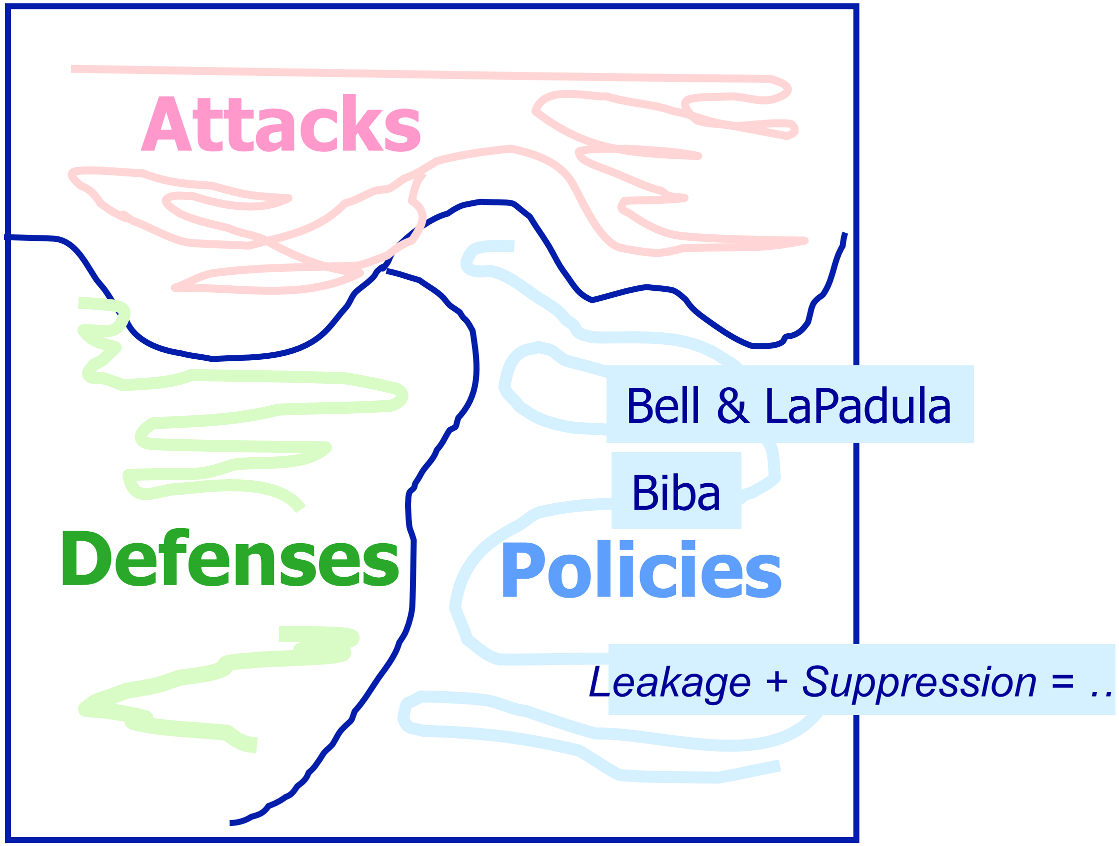
Law: Leakage vs Suppression

*Joint work with Michael Clarkson. [Computer Security Foundations, 2010]

Declassifier: program that reveals some information but suppresses the rest.

What isn't leaked is suppressed…

LS Thm: Leakage + Suppression = Constant


Attacks
Bell & LaPadula
Biba
Defenses
Policies
Leakage + Suppression = …

Reality → Model → Laws:
Execution Monitoring (EM)

Reference monitor [Anderson 1972]

- Gets control on every policy-relevant event
- Blocks execution if allowing event would violate policy
- Integrity of EM protected from subversion.

Essential attributes:

- Acceptance based **solely** on the current execution
- Rejection based on **solely** prefix of execution

Thm: EM only enforces prefix-closed sets (aka "safety properties"). [Schneider 2000]

Reality → Model → Laws → Reality:
Execution Monitoring (EM)

Examples of EM-enforceable policies:

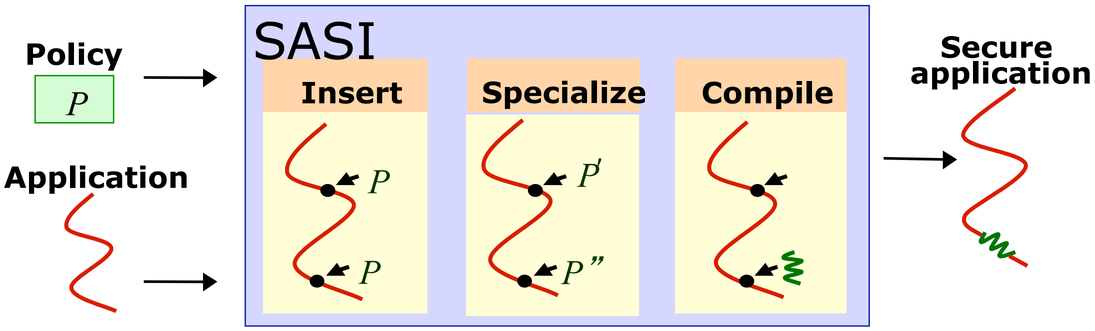
- Only Alice can read file F.
- Don't send msg after reading file F.
- Requests processing is FIFO wrt arrival.

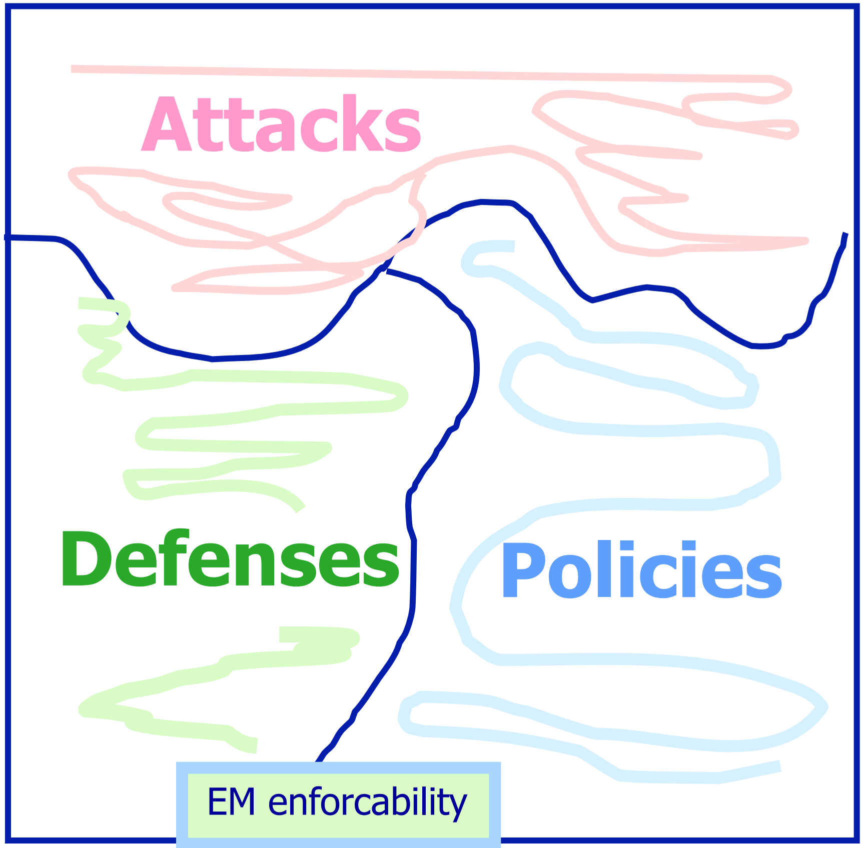
Examples of non EM-enforceable policies:

- Every request is serviced
- Value of x is not correlated with value of y.
- Avg execution time is 3 sec.

In-lined reference monitoring: New approach to enforcement

- Safety property → automaton
- Automaton → rewriter




Attacks
Defenses
Policies
EM enforcability

Reality → Model → Laws:
Independence by Program Obfuscation

Periodic semantics-preserving random program rewriting

Goals: Attacker does not know:

- address of specific instruction subsequences.
- address or representation scheme for variables.
- name or service entry point for any system service.

Options:

- Obfuscate source (arglist, stack layout, …).
- Obfuscate object or binary (syscall meanings, basic block and variable positions, relative offsets, …).
- All of the above.

Reality → Model → Laws:

The Question …

Given program S, obfuscator computes **morphs**:

T(S, K1), T(S, K2), … T(S, Kn)

- Attacker knows:
  - Obfuscator T
  - Input program S
- Attacker does not know:
  - Random keys K1, K2, … Kn

    … Knowledge of the Ki would enable attackers to automate attacks!

Will an attack succeed against a morph?

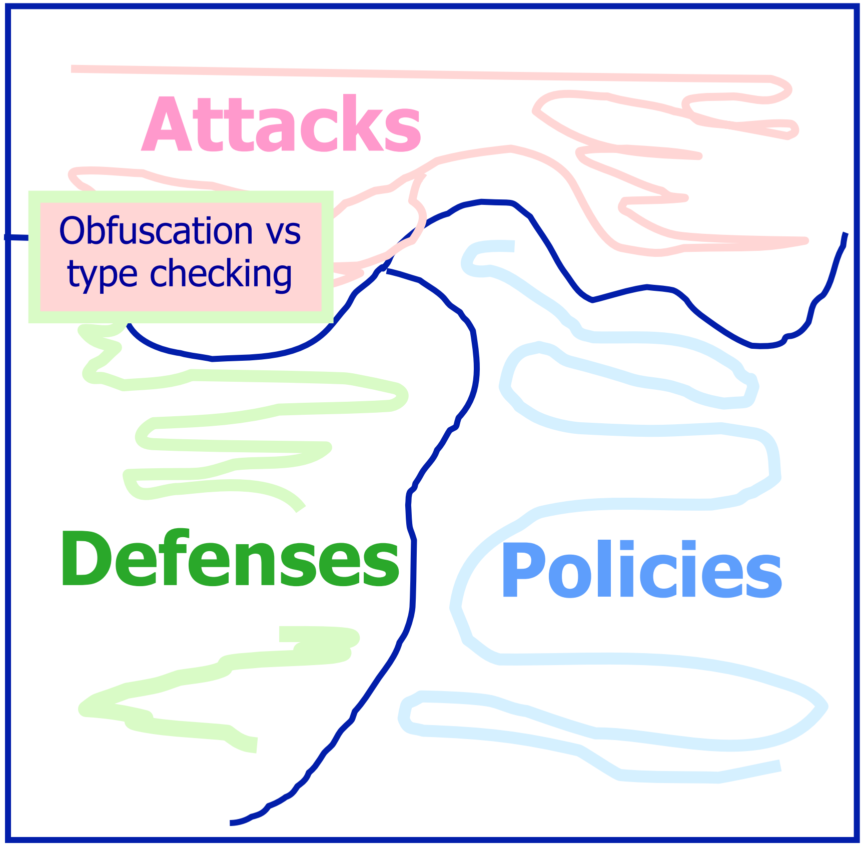
- Seg fault likely if attack doesn't succeed.

  integrity compromise → availability compromise.

Reality → Model → Laws→ Reality:
Obfuscation versus Type Checking

Thesis: Obfuscation and probabilistic dynamic type systems "defend against" the same attacks.

- Type systems:
  - Prevent attacks (always---not just probably)
  - If static, they add no run-time cost
  - Not always part of the language.
- Obfuscation
  - Works on legacy code.
  - Doesn't always defend.



Attacks
Obfuscation vs type checking
Defenses
Policies

But…
isn’t this all “just”
Computer Science?

What about…

Formal Methods and Refinement

If: Pgm **sat** S ***and*** Pgm' $\subseteq$ Pgm

Then: Pgm' **sat** S

… depends on (=implicit assumptions!)

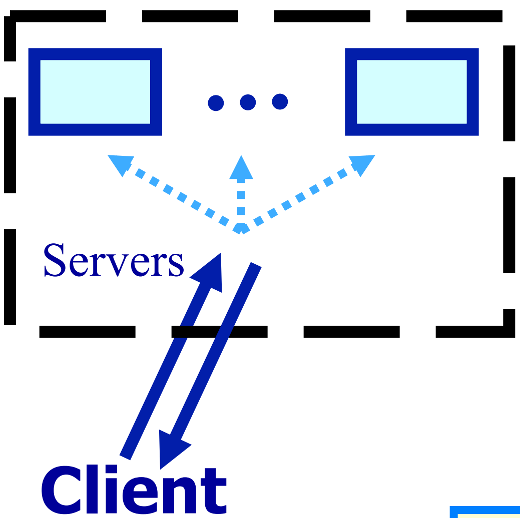
- Modeling execution by sequences (or equiv)
- Equating properties (and pgms) with sets of seqs

- *Useful for integrity (access control).*
- *Useless for confidentiality.*
- *Need richer model than sets of sequences.*

What about…

Replication and Masking

Byzantine failure: Arbitrary and malicious behavior, including collusion.



Basic recipe (=implicit assumptions):

- …
- Replicas fail independently
- 2t+1 replicas tolerate t Byzantine

- *Useful for integrity (access control).*
- *Useless for confidentiality.*
- *Need: Calculus for independence.*

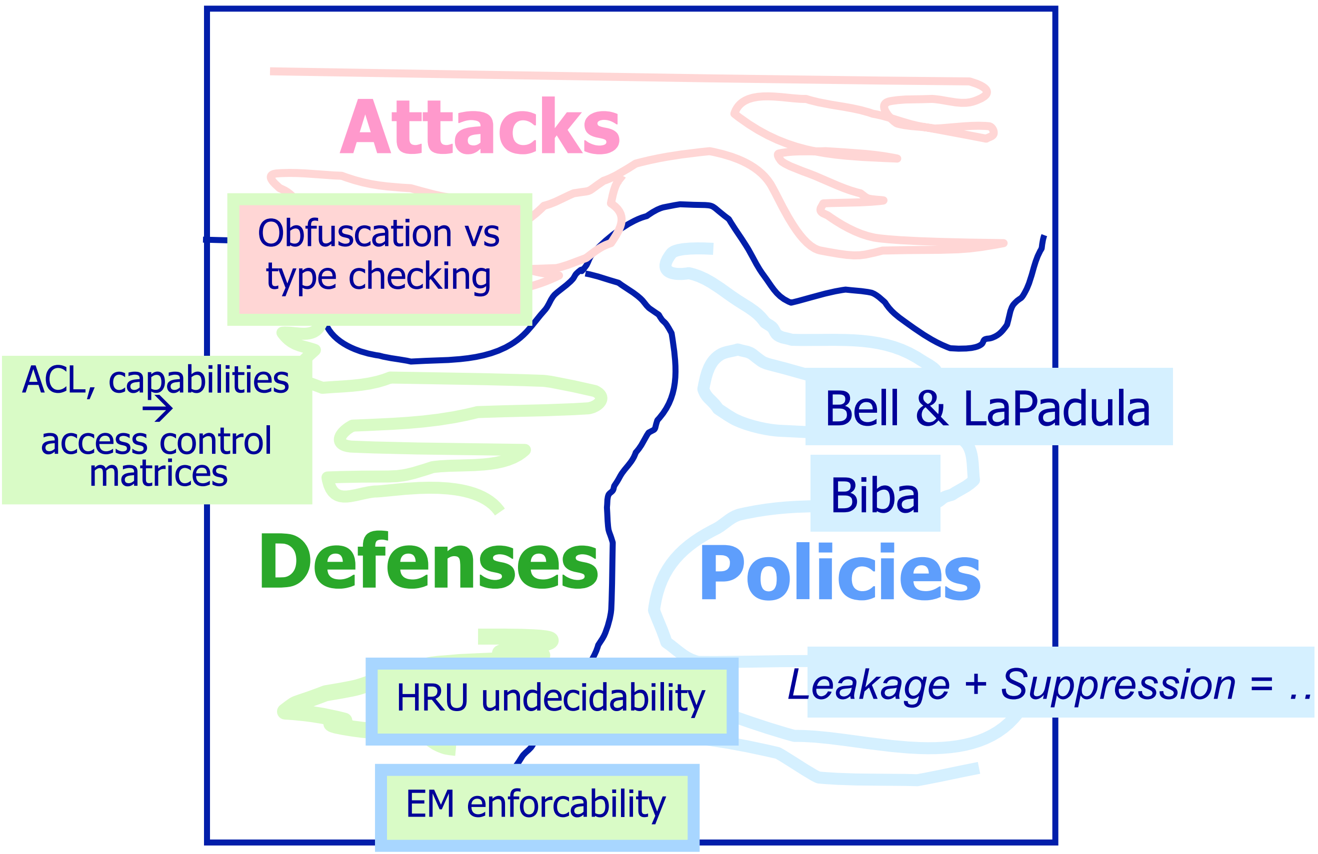
What about…
Cryptography

> If you think cryptography is the answer to your problem, you don't know what your problem is. [P.G. Neumann]

- Science**s** of Cryptography:
  - Information theory [Shannon]
  - Computational complexity
- Handles **limited** kinds properties
  - Confidentiality, integrity, …
  - *Not arbitrary computations*
- Employs **limited** set of mechanisms
  - Secrets, channels, storage, obfuscation
  - *Ignores isolation, reference monitors (access control), re-writing, …*

A Science of Security!

- Concerned with connections between
  - reality,
  - models,
  - laws.
- Reality: Interfaces and actions
- Laws: Ways to predict …
  - qualitative or quantitative
  - analysis or synthesis
  - Classes of defenses, policies, and mechanisms



Attacks
Obfuscation vs type checking
ACL, capabilities → access control matrices
Defenses
Bell & LaPadula
Biba
Policies
HRU undecidability
Leakage + Suppression = …
EM enforcability

Some Open “Science” Problems

- Characterize classes of attacks. Eg, identify attack classes with
  - type-system strength or class of defenses for prevention
  - classes of properties (confidentiality, integrity, …) affected
- **Law**: Trust cannot be created, it can only be relocated.
  - basis for composing defenses and trust relocation.
- **Law**: Trade-off between introspective active defenses and vulnerability to subversion?
  - Consequences for HIV / AIDS / cancer.
- **Law**: Characterize when components are independent.