Is there a science of security (and, if so, what might it look like)?

Anupam Datta
CMU

NSF/IARPA/NSA Workshop on the
Science of Security
Nov 17, 2008

What is a Science?

- *Physics:*
  - *Abstraction:*
    - *Mathematical models of physical universe*
    - *Principled analysis of physical phenomena using models*
  - *Validation:*
    - *Soundness: Does model correctly predict physical phenomena?*
    - *Generality: Does model encompass a broad class of physical phenomena?*

Special Relativity

$$\beta = \frac{v}{c} \qquad \gamma = \frac{1}{\sqrt{1-\beta^2}}$$

At v = 0.42c, γ = 1.10, which means the effects of Relativity become noticeable.

Length contraction

$$L' = \frac{L_0}{\gamma}$$

Time dilation

$$t' = \gamma t$$

Science of Security (by analogy)

- *Computer Security:*
  - *Abstraction:*
    - *Mathematical models of security universe (security mechanisms, adversaries, properties)*
    - *Principled analysis of security universe using models*
  - *Validation:*
    - *Soundness: Does model correctly reflect how secure a system is?*
    - *Generality: Does model capture a broad class of security phenomena?*

Science of Security (also…)

- *Computer Security:*
  - *Design:*
    - *Principles for design of secure systems*

"We speak of engineering as concerned with "synthesis", while science is concerned with "analysis"….discover and teach a science of design, a body of intellectually tough, analytic, partly formalizable, partly empirical, teachable doctrine about the design process."

- H. Simon, *The Sciences of the Artificial*

Questions for this Panel

1. Is there a science of security?
   - Yes, we are getting there (in some areas), although many challenges
2. If so, what might it look like?
   - Next

The Security Universe

Replay, mitm, inject & modify code and data, timing, power, statistical analysis, PPT computation, low-level exploits …

Encryption, signature, hash functions, SSL, IPSec, 802.11*, VMMs, security kernels, hypervisors, web browsers & servers, trusted computing, intrusion detection, …

$\phi\Phi$

Confidentiality, integrity, availability, privacy, …

Security mechanisms, adversaries, and properties

Challenge: Abstractions of Secure Systems

- Identify common denominators of classes of secure systems
  - Define (language or machine-based) model
- One area of success:
  - Analysis of cryptographic protocols
    - Generality
    - Soundness
    - Principled analysis
    - Design principles

Scientific basis for security of SSL/TLS, IKE/JFK/IKEv2, IEEE 802.11i, Kerberos, …

- Can we develop scientific bases for other classes of secure systems?
  - VMMs, security hypervisors & kernels, web browsers & servers (“protection”)

Challenge: Adversary Model

- How do we define the capabilities of the adversary?
  - Resource bound (e.g. time), constrained by system interface, economic models, …?
  - Does adversary know the security mechanism?
- How do we arrive at/validate an adversary model?
  - Generality and Soundness
    - Subsumes broad class of known attacks, forward security, experiments, user studies, …?

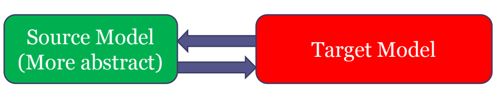
Challenge: Security Properties

- How do we define the universe of security properties?
  - Confidentiality, integrity, availability, non-interference, …
  - Control flow integrity, memory safety, …
  - Properties of single traces, sets of traces, (bi)simulations
- How do we classify and relate security properties?
  - Property A + Property B ⇒ Property C
  - Some results for variants of non-interference [FG01]
- What is a general notion of security for secure systems?
  - Non-interference is too strong in many cases

Challenge: Security Analysis

- Security analysis draws on methods from many fields
  - Logic, programming languages, statistics, complexity theory, machine learning, …
- How is security analysis in the face of an adversary different from other analysis?
  - Traditional program analysis, verification, machine learning
    - Example: PCL [DMP03, DDMP05], learning-based signature generation [VBS08]
- Can we develop principled analysis methods?
  - Secure composition (positive and negative results)
    - Protocols: PCL, Strand Spaces, UC (with case studies)
    - Information-flow: McCullough, McLean, Mantel, …
  - Security-preserving translations (next slide)
- Do we have to give up on soundness?
  - In order to scale (e.g. bug finding efforts)
  - Because of the inherent nature of the problem (e.g. [VBS08])

Security-preserving translations



- From source to target
  - Cryptographic soundness of symbolic (Dolev-Yao) model, type systems(∗) (TAL), run-time enforcement (CFI, ASLR)
- From target to source
  - Model extraction from C source code via software model-checking (∗) techniques (CEGAR), binary analysis
- Research problems
  - Translate models (mechanisms, adversaries) & security properties
  - Soundness theorems: security in source model + conditions ⇒ security in target model

(∗)Methods better developed for software correctness

Challenge: Design principles

- What are the principles for design of secure systems?
  - Saltzer-Schroeder (e.g. economy of mechanism), … what else?
- How do we make these principles precise?
  - System A satisfies Principle P "better" than System B
    - Smaller TCB: one coarse measure of economy of mechanism
    - TCB + property expected of it: complexity of checking property as a measure of economy
- Is there a place for economic and social models and mechanism design in security (processes include humans)?
  - Security risk management in organizations (e.g. WEIS)
  - Theories of privacy (e.g. contextual integrity [Nis04] and its formalization [BDMN06])