Invited talk: Symposium on Requirements Engineering for Information Security

Indianapolis 13:45 5th March 2001

Security Requirements Specifications: How and What?

John Rushby

Computer Science Laboratory
SRI International
Menlo Park, California, USA

Requirements for Security
. . . You Mean There's More Than One?

". . . development of security requirements and security specifications for information technologies needed for the protection of government, business and personal information/control systems"

- This is progress
- Not so long ago, there was a strong "suggestion" that military multilevel security was *the* requirement
  - So what are some requirements beyond this?
- There was also (initially) a strong feeling that the Bell and La Padula model adequately captured this requirement, but. . .
  - So how do we specify security requirements?

How Do We Specify Anything (Formally)?

Two basic methods:

Model-based: Describe an ideal system (a "model") that has the characteristics required and stipulate that any implementation must be a refinement, in some suitable sense, or that ideal

- For example: The Bell and La Padula Model

Property-Based: Specify constraints that any implementation must satisfy

- For example: Noninterference

Why Be Formal?

- Formal methods allow us to calculate with requirements (and other discrete aspects of computational problems)
- Just like calculus in traditional engineering disciplines
- Using formal calculations, some activities that are traditionally performed by reviews
  - Processes that depend on human judgment and consensus

  can be replaced or supplemented by analyses
  - Processes that can be repeated and checked by others, and potentially so by machine

  Language from DO-178B/ED-12B
- Save human thought for where it's really needed

Difficulties With Model-Based Security Requirements Specifications

- Example, Bell and La Padula
  - State machine
  - Subjects and objects, with classifications
  - No "read-up" and no "write-down"
- Highly prescriptive: no help when the prescribed mechanisms are unsuitable (hence "trusted process")
- Tricky to interpret: state "objects" of internal mechanisms must not be overlooked (as some were in the "Multics Interpretation")
- Less to it than meets the eye: "Basic Security Theorem" is a property of the state machine, not the security constraints
- But has intuitive appeal, where it's appropriate

Difficulties With Property-Based Security Requirements Specifications

- Example: Noninterference
  - State machine
  - Inputs and outputs, with classifications
  - Input/Output behavior seen by "low" unaffected by "high" inputs
- Bell and La Padula can be seen as (one) model of this
  - I.e., Property-based specification can justify model-based
- Fine for sequential systems, but doesn't compose
  - Compositional variants lose the intuitive appeal
- Tricky to extend to nonhierarchical ("intransitive") policies

There's A Reason For These Difficulties

- Security (information flow) is not a "property"
- Computer systems can be characterized by their "traces" (or "behaviors")
  - All sequences of their observable input/output activity
  - These sequences may be finite or infinite, and there may be infinitely many of them
- A property is a set of traces (e.g., all traces in which a request is eventually followed by an acknowledgment)
  - All properties are the intersection of a "safety" and a "liveness" property
- Most of software engineering is about designing systems to satisfy properties

Security Is Not A Property

- Noninterference (for example) is not a set of traces
  - You cannot tell whether one trace is secure without knowing what other traces there are

  It's a predicate on (i.e., a set of) sets of traces
- So many of the specification and software engineering methods developed for properties are not directly applicable to security

And There's Another Reason

- Many security concerns are expressed as "counterfactuals"
- Hypothetical statements about a different state of affairs
  - Information flows from A to B if different actions by A cause B to see different behavior
  - This certificate authorizes me to do this action because X would not have signed it otherwise
- Counterfactuals are also one basis for assessing causality and blame
  - If you hadn't been driving so fast, you wouldn't have crashed
- A long-standing problem for linguists, philosophers, logicians, and AI: Not solved yet

What To Do?

- The only restrictions we can actually enforce in a computer system are properties
  - Only see one trace at a time
- In fact, they are safety properties
  - Only recourse is to shut things off when they go bad
- Subtle nonproperties are enforced by stronger properties
- And well-behaved properties are compositional
- So forget about high-falutin counterfactual nonproperties?
  - Nonproperties are often closer to the intuitive requirement
  - Security is not preserved under standard refinement

  May need to relate implementation directly to requirement

What To Do!

- Need a metatheory of security requirements
- What *is* a "security requirement"?
- How are these related to nonproperties, and to counterfactuals?
- Is there is systematic way to capture those that are so related?
  - In a "property-oriented" way
  - And to enforce them by properties?
  - And to verify their refinement ("faithful" interpretations?)
- And what characterizes those security requirements that are properties?

From How to What

What Are Current Security Requirements?

- Multilevel security and information flow may be considered passé today
  - But still important to those who manage classified information
  - And also relevant (under the name "partitioning") to fault containment in safety-critical systems
- And the collapse of many dot.coms may suggest that requirements for E-commerce may not be so urgent either
  - Although the importance of the Internet has been seriously underhyped
  - So issues like contract-signing remain interesting
- But one sure source of new requirements is imminent arrival of ubiquitous/pervasive computing and communications

Pervasive Computing

- Backbone communications are unlimited and essentially free
- Everything that costs more than $5 has an IP address
  - And is connected by radio
    (Bluetooth, WiFi—802.11b, Ricochet etc.)
- Products become services
- Everyday interactions become mediated by computers
- Whose records can be mined for information
  - (Cf. supermarket loyalty cards)

A Challenging Requirement: Privacy

- Others have no right to know what you do (within limits)
- Or, rather, no right to know what you do:
  This seems related to "persona"
  - Keep different facets of life separate
    (cf. "handles" on bboards)
- And anonymity
- But must be combined with accountability
- And maybe auditability
- Have mechanisms for some of these (cf. e-cash), but what are the requirements?

Another Challenging Requirement: Accuracy

- Information mined from our data trails without our cooperation has few checks on accuracy or integrity
- False pictures are created by sloppy, inaccurate, incomplete recording
  - Cf. your credit report

  Which can lead to denial of service, incorrect service
- Devastating consequences if these records are not securely attached to you (dual of privacy!)
  - Cf. identity theft
- Do you think they treat medical records better?
- Can you even tell, without articulated requirements?

Other Challenging Requirements

- Voting (cf. Rebecca Mercuri's thesis and web page)
- Visas (cf. Australia)

  Instance of the general problem of showing that you are authorized to do something, in a decentralized way
  - Buy liquor, or a gun, see a movie, drive a car
- Not hard to devise mechanisms, challenge is make them acceptable
  - Cf. E-stamps

  Can only do this if you know the requirements, independently of mechanisms to achieve them

Summary

- Security requirements seem uniquely(?) elusive and difficult to formalize
  - There may be some good reasons for this
- Requirements for privacy, accuracy, authorization, and accountability need to be articulated clearly to cope with the combined forces of decentralized provision of services and pervasive computing
  - Individuals cannot compensate on their own: unlike (some) loss of reliability
- More questions than answers
- But interesting research opportunities for many years to come