Addressing Trustworthiness in Design

Michael Waidner
IBM Research, Zurich Research Laboratory
http://www.zurich.ibm.com/~wmi
New Paradigms Amplify the “Trust” Problem

Identity / Policy Provider X

Domain B

Domain A

Domain C

out-of-band contract, few protocols
→ dynamically negotiated, Web services

Shared IT Resources / Virtualization, Outsourcing

Shared Business Resources / Business Process Outsourcing

Criminal?
New Paradigms Amplify the “Trust” Problem

- Identity / Policy Provider X
- Shared IT Resources / Virtualization, Outsourcing
- Shared Business Resources / Business Process Outsourcing

Fine-grained componentization → more “independent” domains

Criminal?
Theses

- Trust must be stated in explicit, application-oriented and business-oriented terms.
  - “Does A trust B for x?”
    → “How does A’s overall risk change if it uses service B(x)?”
  - How can A assess B’s trustworthiness?
    How can A calculate the risk implications of interacting with B?
  - Explicit promises and arguments, policies and verification: Service-oriented Assurance
  - Examples
    - Distributing trust through service replication: SINTRA
    - Hardware-based attestation: Trusted Computing
“Trust/Risk” is a Business Property, not an IT Property

Security and privacy must be monitored and managed on the business level!

Exclusive focus of today’s security and privacy technology.

Based on: IBM Research GTO 2004
Service-oriented Architecture as General Principle

- Published and discoverable service interfaces
- Partners agree on service through service/business level agreements
- CORBA, Web Services

Security for SOA still means …
- System and network security
- Web services security stack (e.g., trust and identity)
- Security management
- Intrusion defense
Service-oriented Assurance

SLA / Policy specifies properties and assurances, + responsibilities, procedures, recourse

Service (protocol) conveys evidence

Produces evidence (measurements, logs, signatures)

[Pfitzmann, Waidner 2004]
Example 1: Trusted Computing

Machine A

Does your config satisfy my SLA requirements?

Yes + SLA + Proof

Message $m$, produced/sent by $B$ in context $c$

Non-repudiable proof of origin, $B$, and remote context, $c$.

One specific basis:

Machine B

I exist, hence I’m in a correct state. And only I can access my secrets.

“Trusted booting” from secure (HW) module

Non-repudiable audit records, incl. context, $c$.

© 2004 IBM Corporation
Property-based Attestation for Scalability and Privacy

- Low-level assurances:
  - Statements by trusted processes
  - HW-assured measurements

- Property Attester deployment scenarios
  - On dedicated machine e.g., enterprise firewall, server farm proxy
  - On dedicated OS compartment on verified machine
  - Application on untrusted platform

[Poritz, Schunter, Van Herreweghen, Waidner 2004]
Example 2: Distributing Trust (SINTRA)

- Redundancy and security seen as incompatible
  - one trusted component
  - good security
  - poor availability
  - redundant trusted
  - poor security
  - good availability

- Solution: distribute keys and control
  - threshold cryptography + replication protocols

[Cachin et.al. 2000]
Secure Service Replication

- Replicating a state machine in a group with corrupted replicas; e.g., the authoritative name servers of a secure DNS zone:

- Technical problems:
  - how to sign?
  - how to keep replicas consistent?
    - threshold cryptography
    - secure atomic broadcast
    - agreement using cryptography
Theses

- Trust must be stated in explicit, application-oriented and business-oriented terms.
  - “Does A trust B for x?”
    → “How does A’s overall risk change if it uses service B(x)?”
  - How can A assess B’s trustworthiness?
    How can A calculate the risk implications of interacting with B?
  - Explicit promises and arguments, policies and verification: Service-oriented Assurance
  - Examples
    - Distributing trust through service replication: SINTRA
    - Hardware-based attestation: Trusted Computing
Don’t forget …

- **Assurance** needs a basis!
  - Modeling, verification and evaluation of security properties
  - Model-driven security and privacy