Secure Access Control Problem: Secure Command and Control for Dynamic Air Tasking

Shiu-Kai Chin
Department of Electrical Engineering & Computer Science
Syracuse University
http://www.ecs.syr.edu/faculty/chin
Objective

• Develop a secure system to support dynamic air tasking
  – Army units to specify targets in real time; AF controllers to issue/change air tasking orders to airborne assets in flight; Navy units to provide guidance

• You will develop
  – Security requirements & abstract model
  – Functional specification
  – Implementation
  – A means to assure accountability

• “System” includes people, organizational structure and roles, operations, networks, cryptographic keys, protocols, and computers
System Development Process*

- **Requirement**: establishing generic needs
- **Specification**: defining precisely what the system is supposed to do including *specification verification*, which involves demonstrating the specification meets the requirements
- **Implementation**: designing and building the system, including *implementation verification*, which involves demonstrating that the implementation meets the specification
- **Accountability**: designing and implementing auditing mechanisms so all actions are accounted for

Secure System Development Process*

Informal Development Path
- Security Requirements
  - (demonstration)
  - Functional Specification
    - (testing)
    - Implementation

Formal Development Path
- Abstract Model
  - (proof)
  - Formal Specification
    - (proof)
    - Implementation

Security Architecture*

• Security architecture describes how the system is put together to satisfy the security requirements

• Informal path is conventional: functional specifications and implementations shown to meet security requirements by demonstration and testing

• Formal path, using mathematical techniques, is employed for systems where an extremely high level of assurance regarding the security controls is required

How to Think About this Problem

- **Requirements**
  - What is the system supposed to accomplish?
  - What are the threats to security?
- **Specifications**
  - Who are the *subjects* (principals, actors, agents, roles)?
  - What are the *objects* (orders, services)?
  - What *operations* are performed on objects and by whom?
- **Implementation**
  - Organizational structure: roles and associated rights
  - Cryptographic protocols for privacy and integrity
  - Trust networks for cryptographic binding of keys, roles, and privileges to principals
- **Accountability**
  - What authority is associated with each role? How is assignment to roles tracked? How are orders issued and signed?
Requirements

• What is the system supposed to do?
  
  
  
• What are security threats?
  
  
  

Specifications

• Who are the subjects (actors)?
  
• What are the objects (orders, services)?
  
• What are the operations on objects? Who has permission to act on objects?
Implementation

• How are principals organized or grouped into roles?
• How are principals and their roles identified and authenticated?
• How is information determined to have integrity?
• How are cryptographic keys distributed?
• How is trust in cryptographic keys established and maintained?
Accountability

• How will authority be defined, assigned, verified, and accounted for?
• How will requests for air strikes be verified and accounted for?
• How will authority to order air strikes be verified and accounted for?
• How will guidance to targets be verified and accounted for?
Note on Implementation

• Implementation does not mean writing code
• Identify precisely, accurately, and in detail:
  – What components are used
  – The functionality of the components
  – How the components are organized, connected, and operate together to form the system
• Justification that the implementation meets the specification.
Your Task

• Answer the questions – when you’ve answered them all you’ll likely have a fairly complete system design
• Make the case that your specifications satisfy the requirements
• Make the case that your implementation satisfies your specifications
• Show how accountability is maintained throughout the system
Useful Tools

• Useful abstractions
  – Reference monitors
  – Access control matrix
  – Role-based access control
  – Cryptographically signed statements (certificates)
  – Integrity-checked channels

• Analytical tools
  – Access control logic for reasoning about principals, credentials, delegation, roles, trust, and access control