Application-Centric Security Models

Prof. Ravi Sandhu
Executive Director and Endowed Chair
Institute for Cyber Security
University of Texas at San Antonio
July 2009

ravi.sandhu@utsa.edu
www.profsandhu.com
Institute for Cyber Security (ICS)
Founded 2007

Dr. George Perry
Dean of the College of Science

Dr. Robert W. Gracy
Vice President for Research

Dr. Ravi Sandhu
Executive Director, ICS

Center for Infrastructure Assurance and Security (CIAS)
Dr. Gregory White

Dark Screen Exercises and Training

National Collegiate Cyber Defense Competition

Sponsored Research Projects
Dr. Ravi Sandhu

Numerous projects from NSF, AFOSR, AFRL, ONR, with 10+ UTSA researchers in collaboration with 11 University partners

ICS Research Operations
To be appointed

Innovative research infrastructure including experimental cloud and honeyfarm

World leading research with real world impact
ICS Key Assets

- World leading security modeling and analysis research
  - Role-Based Access Control (RBAC) Model (1996)
    - Catalyzes dominance of RBAC in commercial systems
    - Develops into a NIST/ANSI Standard (2004)
    - Attribute-Based Access Control on Steroids
    - Unifies numerous extensions/enhancements
    - Policy, Enforcement, Implementation Models
    - From what to how
  - Group-Centric Information Sharing (2007)
    - Sharing metaphor of meeting room
    - Equivalently: mission centric
  - Security for Social Networks (2008)
  - Botnet Analysis, Detection and Mitigation (2008)
  - Multilevel Secure Architectures (2009)

- Bring in partners from leading research universities worldwide as appropriate
- Ready to commercialize when appropriate
Application Context

- **Our Basic Premise**
  - There can be no security without application context
  - Courtney’s Law (1970s, 1980s ??):
    - You cannot say anything interesting (i.e. significant) about the security of a system except in the context of a particular application and environment

- **Corollary**
  - There can be no security model without application context

- **Reality**
  - Existing security models are application neutral
    - Assumption is they can be readily “configured” or “policy-ified” to suit application context
Existing Security Models (1)

- **Discretionary Access Control (DAC)**
  - Characteristic: Owner-based discretion
  - Drawbacks:
    - Classic formulation fails to distinguish copy from read
    - Application context drives ownership and its delegation

- **Lattice-Based Access Control (LBAC)**
  - Characteristic: One directional information flow in a lattice of security labels
    - Also known as: Bell-LaPadula, Multi-Level Security, Mandatory Access Control (ignoring subtle differences)
  - Drawbacks: Many applications
    - Many applications violate one directional information flow
    - Many applications do not fit within preexisting security labels
Existing Security Models (2)

- **Role-Based Access Control (RBAC)**
  - Characteristic: Role is central, administration is simple
  - Drawbacks:
    - Need to define the roles for each application/environment
    - Lack of standardized roles results in lack of interoperability
    - Too open: can be configured to do DAC or LBAC

- **Attribute-Based Access Control (ABAC)**
  - Characteristic: subsume security labels, roles and more as attributes and enforce attribute-based policies
  - Drawbacks:
    - All the RBAC drawbacks on steroids
    - Administrative complexity
Usage Control Model (UCON)

- unified model integrating
  - authorization
  - obligation
  - conditions
- and incorporating
  - continuity of decisions
  - mutability of attributes

UCON is Attribute-Based Access Control on Steroids
Usage Control Model (UCON)

- DAC
- LBAC
- RBAC
- ABAC
- … and many, many others
- UCON
  - ABAC on steroids
  - Simple, familiar, usable and effective use cases demonstrate the need for UCON
    - Automatic Teller Machines
    - CAPTCHAs at Public web sites
    - End User License Agreements
    - Terms of Usage for WiFi in Hotels, Airports
    - Rate limits on call center workers
Our Basic Premise
- There can be no security model without application context

So how does one customize an application-centric security model?
- Combine the essential insights of DAC, LBAC, RBAC, ABAC and UCON in a meaningful way
- Directly address the application-specific trade-offs
  - Within the security objectives of confidentiality, integrity and availability
  - Across security, performance, cost and usability objectives
- Separate the real-world concerns of practical distributed systems and ensuing staleness and approximations (enforcement layer) from the policy concerns in an idealized environment (policy layer)
PEI Models: 3 Layers/5 Layers

- Security and system goals (objectives/policy)
  - Necessarily Informal
- Policy models
  - Formal/quasi-formal
- Enforcement models
  - System block diagrams, Protocol flows
- Implementation models
  - Pseudo-code
- Trusted Computing Technology (mechanisms/implementation)
  - Actual Code

- Horizontal view
- Vertical view: Looks across layers
- Looks at individual layer
• Extensive research in the last two decades
  - ORCON, DRM, ERM, XrML, ODRL, etc.
• Copy/usage control has received major attention
• Manageability problem largely unaddressed

Dissemination Chain with Sticky Policies on Objects
Group-Centric Sharing (g-SIS)

- Brings users & objects together in a group
  - Focuses on manageability using groups
  - Co-exists with dissemination-centric
  - Two metaphors
    - Secure Meeting Room (E.g. Program committee)
    - Subscription Model (E.g. Secure multicast)

- Operational aspects
  - Group characteristics
    - E.g. Are there any core properties?
  - Group operation semantics
    - E.g. What is authorized by join, add, etc.?
  - Read-only Vs Read-Write

- Administrative aspects
  - E.g. Who authorizes join, add, etc.?
  - May be application dependent

- Multiple groups
  - Inter-group relationship
g-SIS Operation Semantics

GROUP
Authz (u,o,r)?

- join
- leave
- add
- remove

Users
Objects

© Ravi Sandhu
g-SIS Operation Semantics

GROUP Authz (u,o,r)?

- Users
  - Strict Join
  - Liberal Join
  - Strict Add
  - Liberal Add

- Objects
  - Strict Leave
  - Liberal Leave
  - Strict Remove
  - Liberal Remove
Family of g-SIS Policy Models

Subject Model

- $<\text{SJ, SL}>$
- $<\text{LJ, SL}>$
- $<\text{LJ, LL}>$

Object Model

- $<\text{SA, SR}>$
- $<\text{LA, SR}>$
- $<\text{LA, LR}>$

Part (a)
Part (b)
Part (c)
Part (d)
Part (e):
Part (f):

$\text{g-SIS models: (e) X (f)}$

Traditional Groups: $<\text{LJ, SL, LA, SR}>$
Secure Multicast: $<\text{SJ, LL, LA, *>}$

Most Restrictive g-SIS Specification:

$$\Box (\text{Authz} \leftrightarrow (\neg \text{SR} \land \neg \text{SL}) \ S (\text{SA} \land (\neg \text{SL} \ S \text{SJ}))))$$
g-SIS Enforcement Model

Subject Attributes: {id, Join-TS, Leave-TS, ORL, gKey}
- ORL: Object Revocation List
- gKey: Group Key

Object Attributes: {id, Add-TS}

Refresh Time (RT): TRM contacts CC to update attributes
From Policy to Enforcement

- Additional Trusted/Semi-Trusted Servers
- Approximate Enforcement

- Finally, the Implementation layer models spell out protocol details and details of TRM algorithms
Conclusion

- Application-Centric Security Models require
  - State-of-the-art approaches such as UCON, PEI
  - Mix-and-match DAC, LBAC, RBAC, UCON, g-SIS
  - ..... 
  - ..... 

- The future of cyber security research will revolve around
  - Application-centric models 
  - Technology-centric models 
  - Attack models
  - ..... 
  - .....