CYBER SECURITY DIVISION

2013 PRINCIPAL INVESTIGATORS'

Methodology for Assessment of Security Properties

Naval Postgraduate School

Cynthia Irvine

16-18 September 2013

Homeland Security

Science and Technology

Team Profile

- PI: Cynthia E. Irvine
- Mark Gondree, Research Assistant Professor
- Timothy Levin, Security Engineer (retired recently)
- Thuy Nguyen, Security Engineer
- Team Project History Samples
  - CyberCIEGE: US Navy, NSF, OSD
  - Trusted Computing Exemplar (TCX): ONR, NRO, OSD
  - Separation Kernel Protection Profile (SKPP): NSA
  - 3Dsec and RCsec: NSF
  - Secure Core: NSF
  - Monterey Security Architecture (MYSEA): NRO
- Decades of collective experience
  - Analysis, design and development of secure systems
  - Systems requirement analysis

Customer Need

- *Large* safety-critical systems are *composed* systems
  - Ex: SCADA, Automotive, Aerospace, Railway
  - Challenges: components evaluated in a specific context
    - Composition changes context, undermines eval. results
    - Emergent behavior (interference, cascading faults)
- Common methods for Security Engineering w/ composed systems:
  1. Prove composition has no effect
     - Show isolation of components (time-space division)
     - Eval in isolation ➔ Eval unchanged by composition
  2. Prove composition has desired effect
     - TCB subset methodology [Shockley, 1987]

"*We do not understand how to combine systems in ways that ensure that the combination is more, rather than less, secure and resilient than its weakest components*" - *A Roadmap for Cybersecurity Research*, DHS, 2009.

Approach

- Formalize TCB (Trusted Computing Base) subset model
  - First attempt to prove the TCB-Subset methodology has its claimed properties
  - Practice: ubiquitous ("[S87] *is* how we build stuff")
  - Theory: unclear ("[S87] worked out the details, *right*?")
  - Desire: formalize theory & connect it to practice better
- Formal Model in Event-B / Rodin
  - Popular w/ formal analysis of safety-critical systems
  - Leverage existing tools (ProB, B4Free, UML-B)
  - Model can be "refined" into real system specification

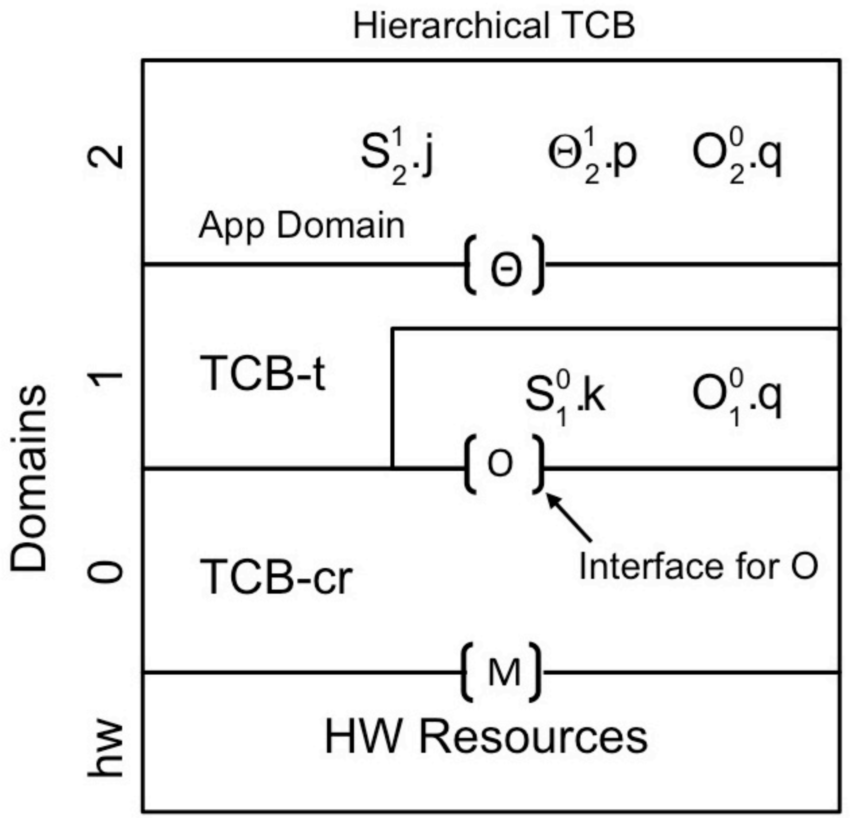
Example of TCB Subset Problem

- Approach to assess multicomponent systems
- Outlined in seminal paper by Shockley & Schell (1987)
  - Widely accepted engineering approach reflected in a number of subsequent documents
- Yet imprecise original: leads to opaqueness & ambiguity

Thus, the assumption that multiple independently evaluatable TCB subsets can be found leads to the conclusion that they are independently evaluatable precisely because the objects under their control can be partitioned into rigorously disjoint protection domains.

… a TCB subset is tamperproof [if] *the representations of the abstract objects it implements for interpretive access must be stored in storage objects which cannot be directly* [accessed] *by external subjects*

Hierarchical TCB with Tampering



Bypass-Tampering Scenario

TCB-t:

creates Type Θ to *encapsulate* type O
exports interface for Θ
instantiates $\Theta_2.p$ with $O_2.q$ as *constituent*
allocates object $O_1.q$ to domain 2 (= $O_2.q$)

$S_2.j$:

invokes TCB-cr's O interface with $O_2.o$
(bypasses the TCB-t policy for $\Theta_2.p$)

Solution: do not allocate $O_1.q$ to domain 2

$A^b_c.d$ =
Entity is type **A**
Created by domain **b**
Allocated to domain **c**
Unique ID = **d**

- TCB-cr – creating TCB subset
- S – subject (active entity)
- HW – hardware

Policy Representation

- Abstraction of policy to simple boolean for each possible access request (ignores IDs and labels)

  access: [s: subject, o: object, m: mode]
  accessSet: access*
  TCBsubset.localPolicy = accessSet

  $\text{TCBpolicy} = \bigcup_{\text{p:accessSet} \mid \exists\, \text{t:TCBsubset} (\text{t} \in \text{M} \wedge \text{p} \in \text{t.localPolicy})}$

- Each *access* entry can indicate an allowed access or a denied access.
  - Paper represents system with *denied* access interpretation, which is <u>not intuitive</u> for engineering
- What are the impacts of the policy representation?
  - Would the opposite access interpretation prove, too?
  - Need to ensure that the abstract choice does not inhibit refinements (interfaces for real systems)

Event-B / Rodin Advantages

- Incorporates
  - Type checking
  - Proof obligations
  - Theorem proving
  - Model checking
- Previous work
  - Suggests this is the “right framework” for formal SCADA component modeling
  - Claims automatic ladder logic generation from the B state machine
- Composition is still manual, and a case study

Benefits

- TCB subsets model formalized
  - Assumptions delineated
  - Terminology clarified / standardized
- Make model “operational”
  - Formalization connected to useful model tool chain
  - Theory connected to Practice via “refinement”
- Describe SCADA system in TCB-Subset terms
  - Define subjects, policy, hosts, channels, etc:
    - Comms Module, Digital I/O Module, Sensors, PLC
  - Possibility to define least-privilege with SCADA

Current Status

- Partial formalization of TCB Subset framework
  - Verbatim Shockley model
  - “Just enough” to validate approach is possible
  - Full assumptions not delineated
  - Not type-checked
- Survey existing tools for SCADA design
- Selection of model language and tool chain

Next Steps

- Express “Verbatim Shockley” model in Event B
  - Type checking
  - Incorporate secondary [S87] features
  - Translate [S87] claims into theorems
  - Prove theorems
- Expand into “Final TCB Subset” model
  - Add assumptions that allow us to prove theorems
- Decompose ex. SCADA system into TCB Subsets
- Demonstrate model refinement into ex. System
- Event-B TCB-Subset models released as FOSS tool

Contact Information

Cynthia Irvine
irvine@nps.edu
(831) 656 2461

Mark Gondree
mgondree@nps.edu
(831) 656 2025

Thuy Nguyen
tdnguyen@nps.edu
(831) 656 3989