Ensuring System Resilience at Design Time: A User- and Attacker-Oriented Approach

Bill Sanders
University of Illinois at Urbana-Champaign

HCSS 2012 – Annapolis, MD – May 11, 2012

INFORMATIONTRUST INSTITUTE

Coordinated Science Laboratory

Building Interdisciplinary Excellence with Societal Impact

- **Excellence in:**
  - Computing and Networks
  - Circuits, Electronics & Surface Science
  - Communications & Signal Processing
  - Decision & Control
  - Remote Sensing

- **Initiatives:**
  - Computer Vision
  - SRC Focus Center Research Program
  - Neuroengineering IGERT
  - Human-Machine Adversarial Network MURI

- **Statistics:**
  - 60 years as a premier national interdisciplinary research facility
  - 550 Researchers: 110 professors, 330 graduate students, 60 undergraduate students, & 50 professionals
  - Over $300M in active research projects

- **Affiliated Institutes:**
  - ITI: Information Trust Institute
  - ADSC: Advanced Digital Sciences Center (Singapore)
  - PCI: Parallel Computing Institute

- **Major Centers:**
  - Illinois Center for Wireless Systems
  - NSF National Center for Professional and Research Ethics
  - NSF Science of Information Science and Technology Center
  - DOE/DHS Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) Center
  - Boeing Trusted Software Center
  - HHS SHARPS Health Care IT Security Center
  - NSA Science of Security Center
  - Illinois Center for a Smarter Electric Grid

Designed-in Security

- From Federal Cybersecurity Research and Development Program Strategic Plan:
  - Designing and developing SW systems that are resistant to attacks
  - Generating assurance artifacts to attest to the system's capabilities to withstand attacks
- Cites progress in:
  - Dynamic Analysis
  - Model Checking
  - Theorem Proving

More Broadly, one must consider

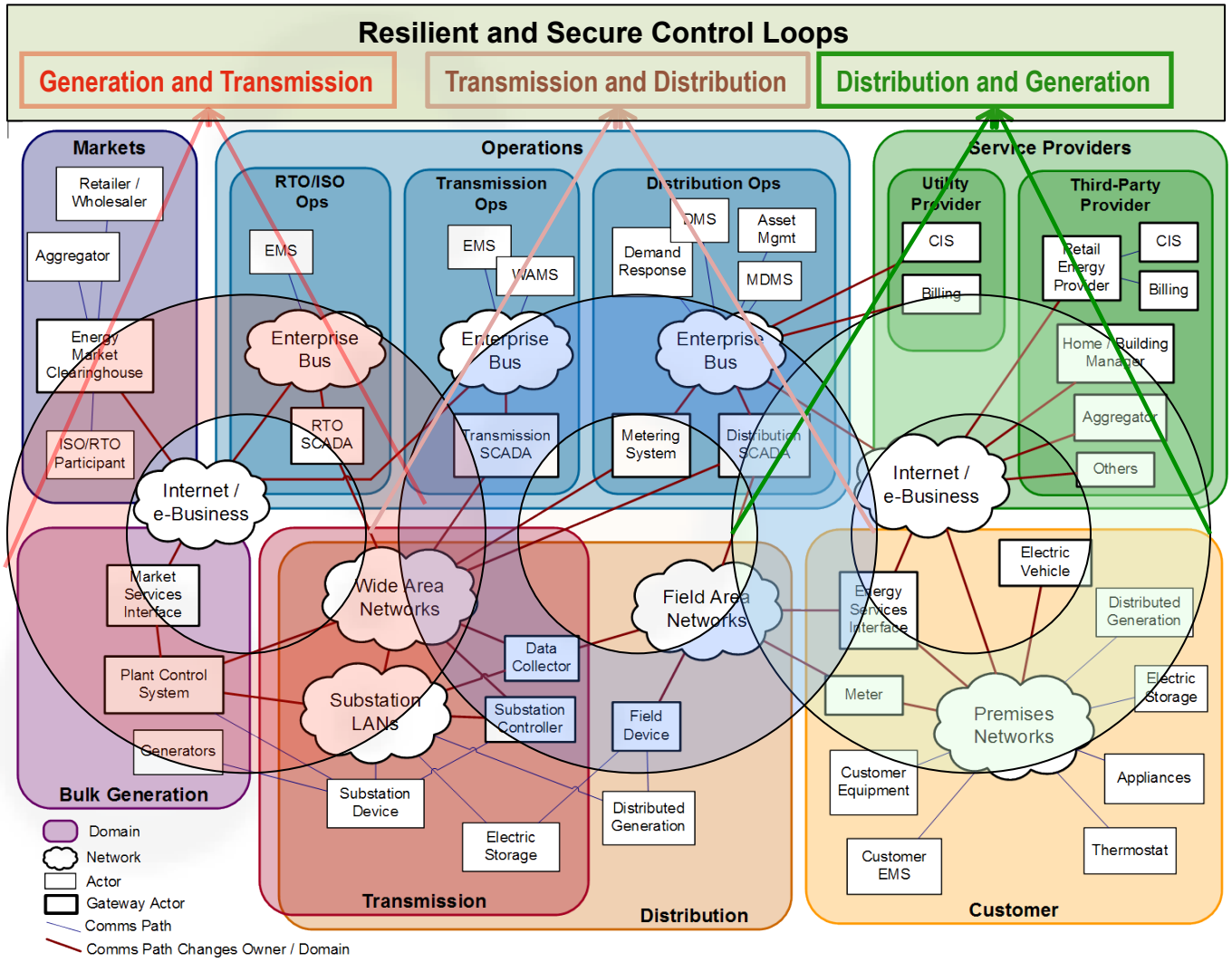
- Systems which may consist of:
  - Software
  - Hardware
  - Physical components (i.e., cyber-physical systems)
  - Humans (both good and bad; i.e., cyber-human systems)
- Systems which may be impossible to make perfectly secure, but can made *resilient*

Resiliency

- Aims to protect when possible (successful attack avoidance), but understands that some attacks will be successful, and that proper operation must be preserved at all times.
- Applies to systems that are intrusion-tolerant, so that there is some capacity left to perform event if some attacks are successful
- Most often applicable at the system-architecture level
- Requires *probabilistic analysis* of security properties, since attacks may be successful

Power Grid Infrastructure: An Example Cyber-Physical-Human System

- **Multi-layer Control Loops**
- *Multi-domain Control Loops*
  - Demand Response
  - Wide-area Real-time control
  - Distributed Electric Storage
  - Distributed Generation
- *Intra-domain Control Loops*
  - Home controls for smart heating, cooling, appliances
  - Home controls for distributed generation
  - Utility distribution Automation
- **Resilient and Secure Control**
  - *Secure and real-time communication substrate*
  - Integrity, authentication, confidentiality
  - Trust and key management
  - End-to-end Quality of Service
  - *Automated attack response systems*
  - *Risk and security assessment*
  - Model-based, quantitative validation tools



Note: the underlying Smart Grid Architecture has been developed by EPRI/NIST.

What is needed to bring these advances to bear on system security?

Tools that

- Generate assurance evidence as a system is built
- Can be easily understood and used by real programmers (and yield benefits they can see)
- Can support integration of evidence about various components
- Can be re-applied easily as systems evolve and adapt

From NITRD Presentation at IEEE SSP 2011

What is needed to bring these advances to bear on system security through resiliency?

Tools that

- Generate assurance evidence as a system is built
- Can be easily understood and used by real system architects (and yield benefits they can see)
- Can support integration of evidence about various components
- Can be re-applied easily as systems evolve and adapt
- Can account for cyber, physical, and human behaviors, and system response to adversarial events

Möbius-SE Approach

- Build on Long-term established formal-basis for probabilistic evaluation (Möbius)
- Add support for attacker, system, and user modeling formalisms that are natural to security analysts
  - Attack Execution Graphs
  - ADVISE Adversary Formalism
  - HITOP User Modeling Formalism
  - System response modeling formalism
- Leverage Möbius model composition approach to build overall (multi-formalism) system models
- Use Möbius capabilities to broadly and efficiently explore system design space to find best approach to security through resiliency

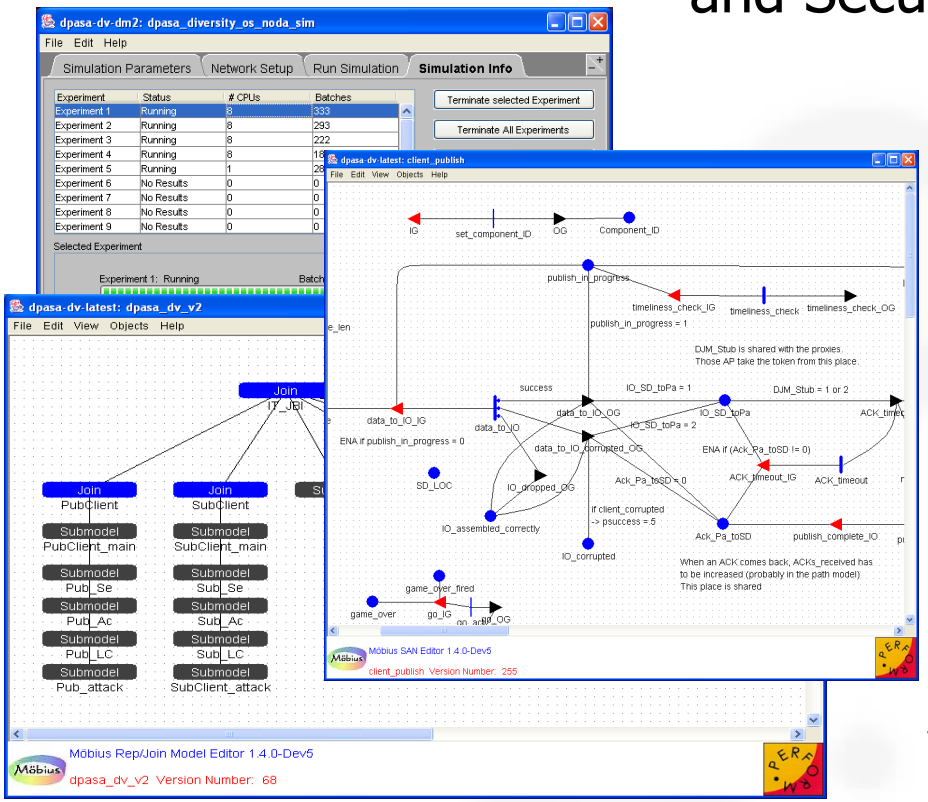
Outline

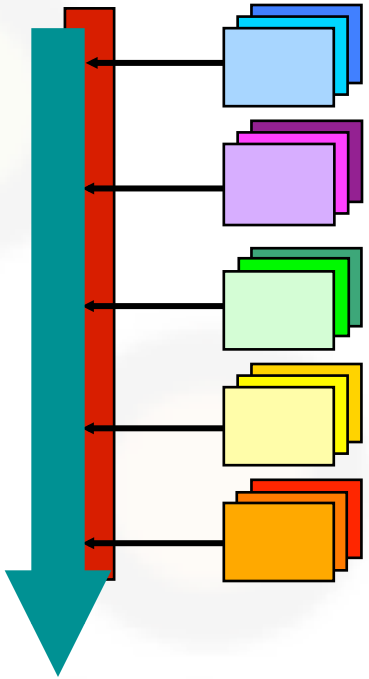
- Möbius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together

Outline

- Möbius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together

Mobius: Model-Based Evaluation of System Dependability and Security





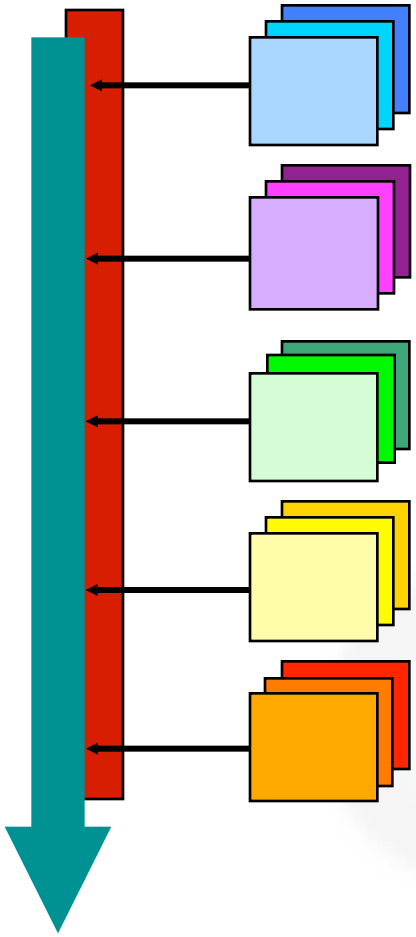
Framework Component

- Atomic Model
- Composed Model
- Solvable Model
- Connected Model
- Study Specifier (generates multiple models)

Use:

- Site licenses at hundreds of academic sites for teaching and research.
- Corporate licenses to a range of industries: Defense/Military, satellites, telecommunications, biology/genetics
- Development of new plugins for Möbius: Univ. of Dortmund, Univ. of Edinburgh, Univ. of Twente, Carleton University, and many others

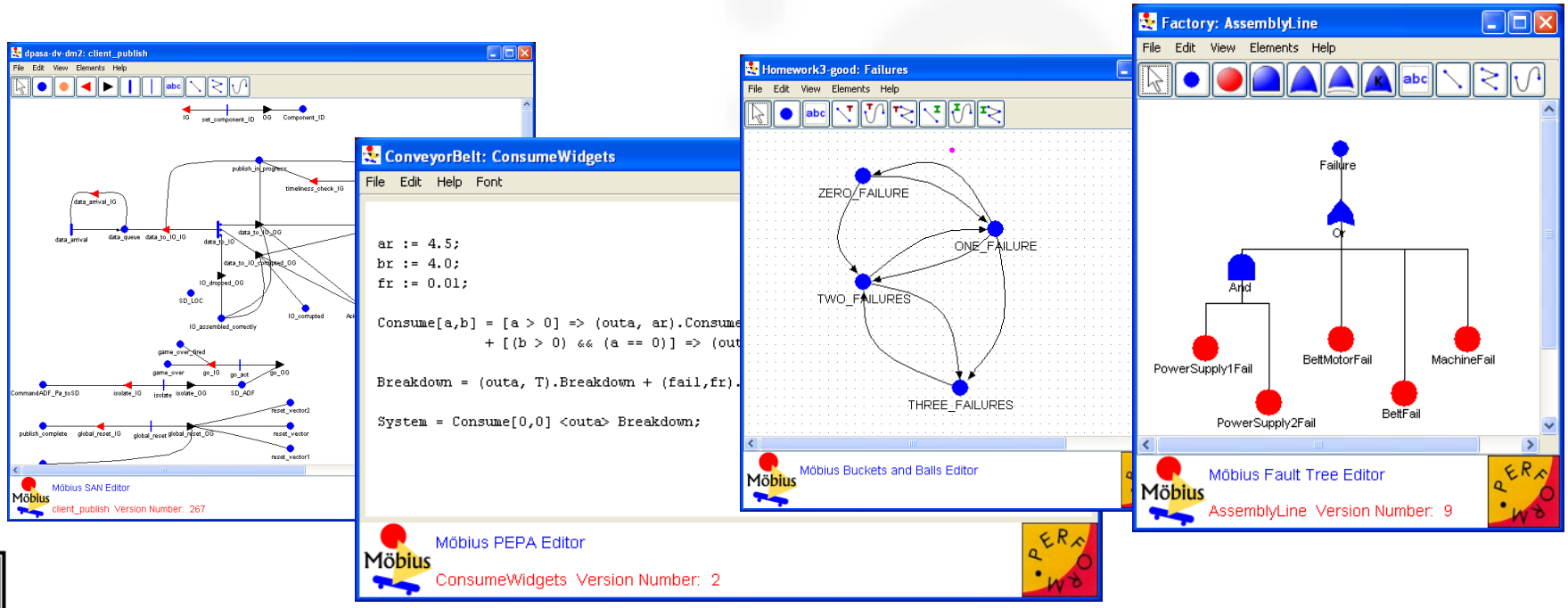
Model Specification in the Möbius Framework



Framework Component	Implemented Formalisms
Atomic Model	PEPA Process Algebra, Stochastic Activity Networks, Buckets and Balls, Fault/Attack Trees, External Atomic
Composed Model	Graph interconnection Replicate/Join Action Synchronization
Reward Model	Rate/Impulse reward variables Path-based reward variables Domain-specific formalisms
Study Methods	Range and Set Variation Design of Experiments
Solution Methods	Simulation Methods: Terminating and Steady State Simultaneous Simulation Numerical Methods: Transient, Iterative Steady State, Direct Steady State, Accumulated Reward, Adaptive Transient, Deterministic Iterative Steady State

Model Representation

- Multiple modeling formalisms available:
  - Stochastic Activity Networks ('SANs', advanced stochastic Petri nets), PEPA (textual-based process algebra), Buckets and Balls (inc. Markov chains), Fault/Attack trees
  - Parameters of the model can be specified variables and set at analysis time.

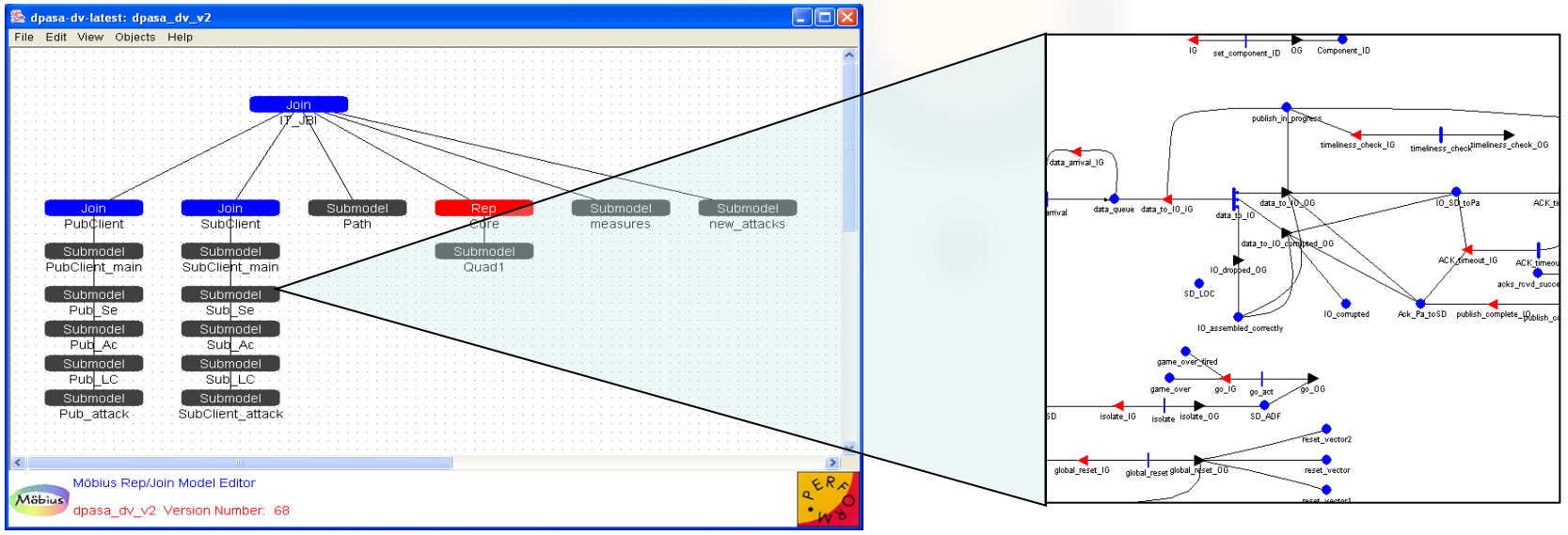


Model Support of the Abstract Functional Interface: State Variables, Actions, and Properties

- Formally, a model in the Möbius framework is a set of "state variables," a set of "actions," and set of "properties"
- State variables "contain" information about the state of the system being modeled
  - They have a type, which defines their "structure"
  - They have a value, which defines the "state" of the variable
- Actions prescribe how the value of state variables may change as a function of time
- Properties specify characteristics that may effect the solution of a model

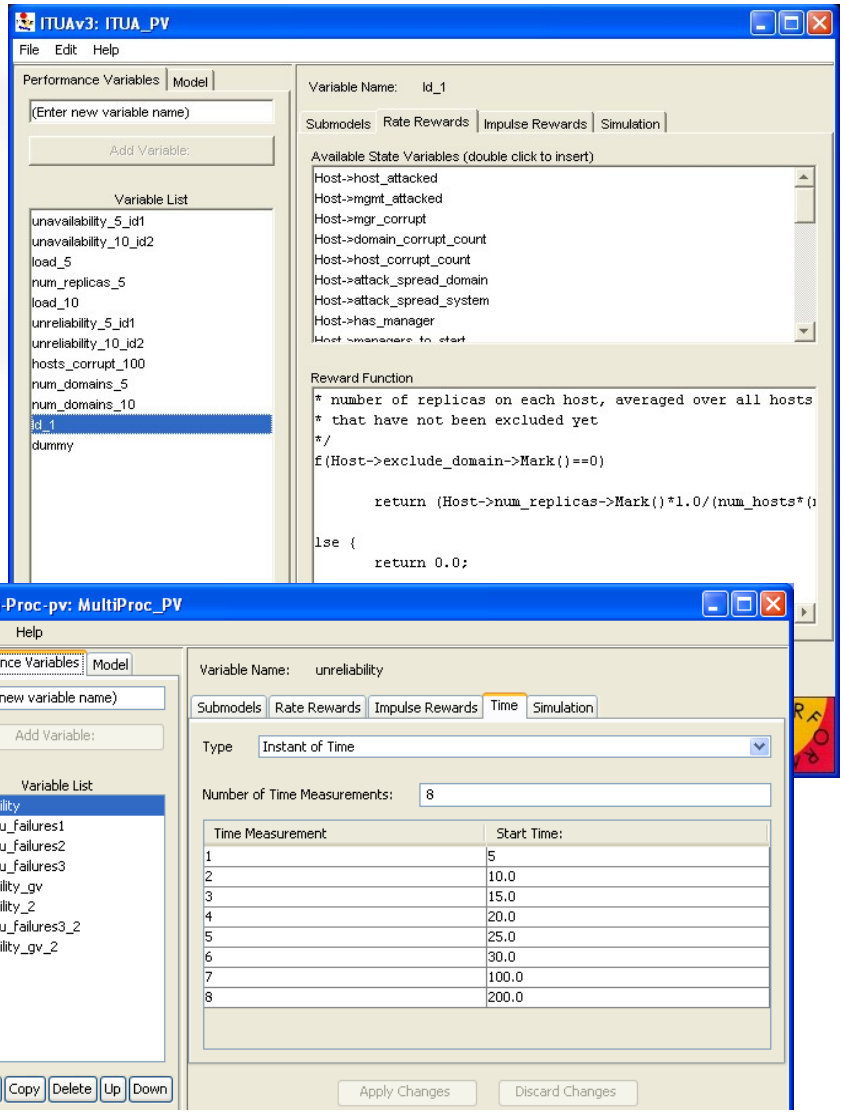
Model Composition

- Hierarchical model construction
  - System model constructed from multiple component models.
  - Can combine models built with different formalisms
- Rapid model development
- Multiple composition techniques provide flexibility in model construction
  - Replicate/Join, Graph, Action Synchronization



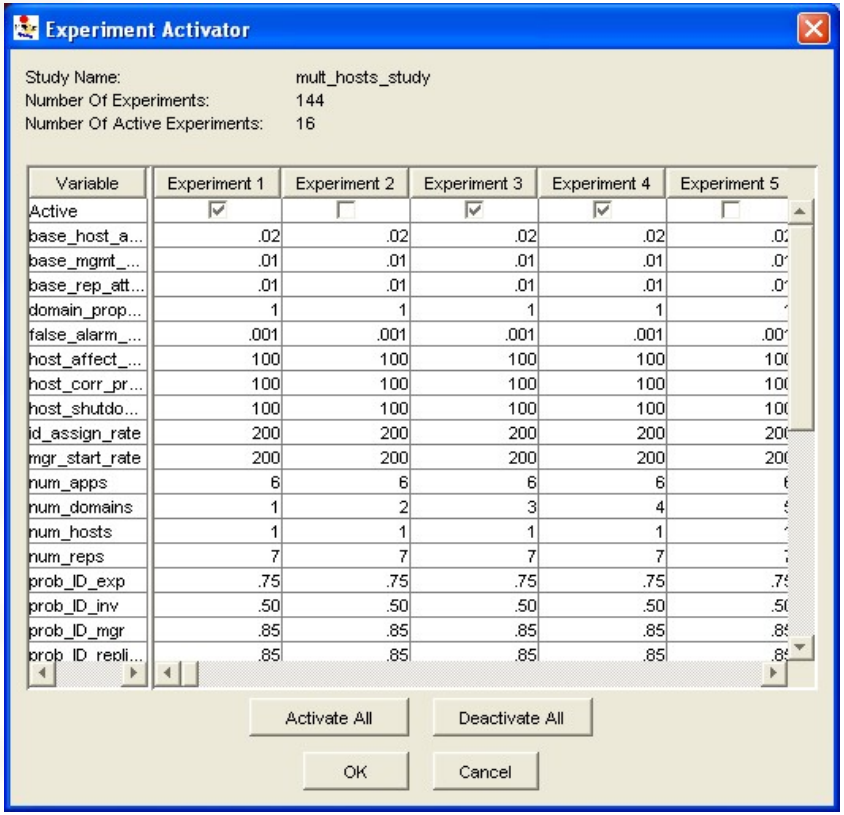
Custom System Measures

- Write customized functions to measure system properties of interest.
- Functions map system state to real number
  - Can be simple or complex expressions
- Functions evaluated:
  - At specific time points
  - Over an interval of time
  - In steady-state



Parameter Space Exploration

- Model parameters are specified as constants or variables
- Variables values can be varied to study how the system behaves under wide ranges of conditions
- Design of Experiments module that guides parameter value selection to efficiently explore parameter space

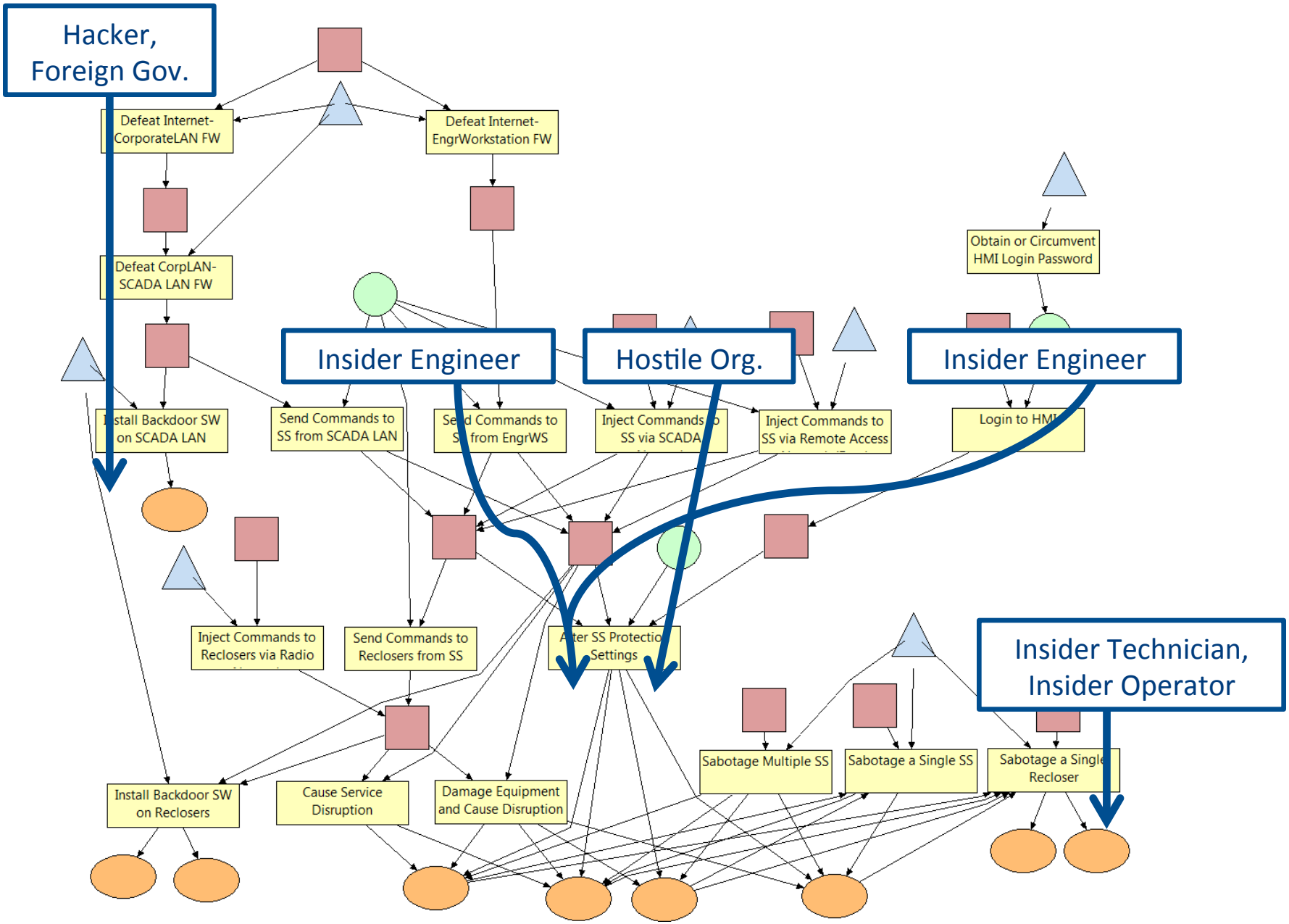


Model Solution Techniques/Algorithms

- Analytic/Numerical
  - Stochastic Process Generation
    - Implicit Matrix Representation (largeness tolerance)
    - Lumping (Largeness avoidance)
  - Transient Numerical Solution
    - Point Uniformization (trs, atrs)
    - Interval Uniformization (ars)
  - Steady State Numerical Solution
    - LU Decomposition (dss)
    - Gauss-Seidel Iteration (iss)
    - Deterministic/Exponential Solution (diss, adiss)
- Simulation (Steady State & Terminating)
  - Parallel Simulation
  - Simultaneous simulation

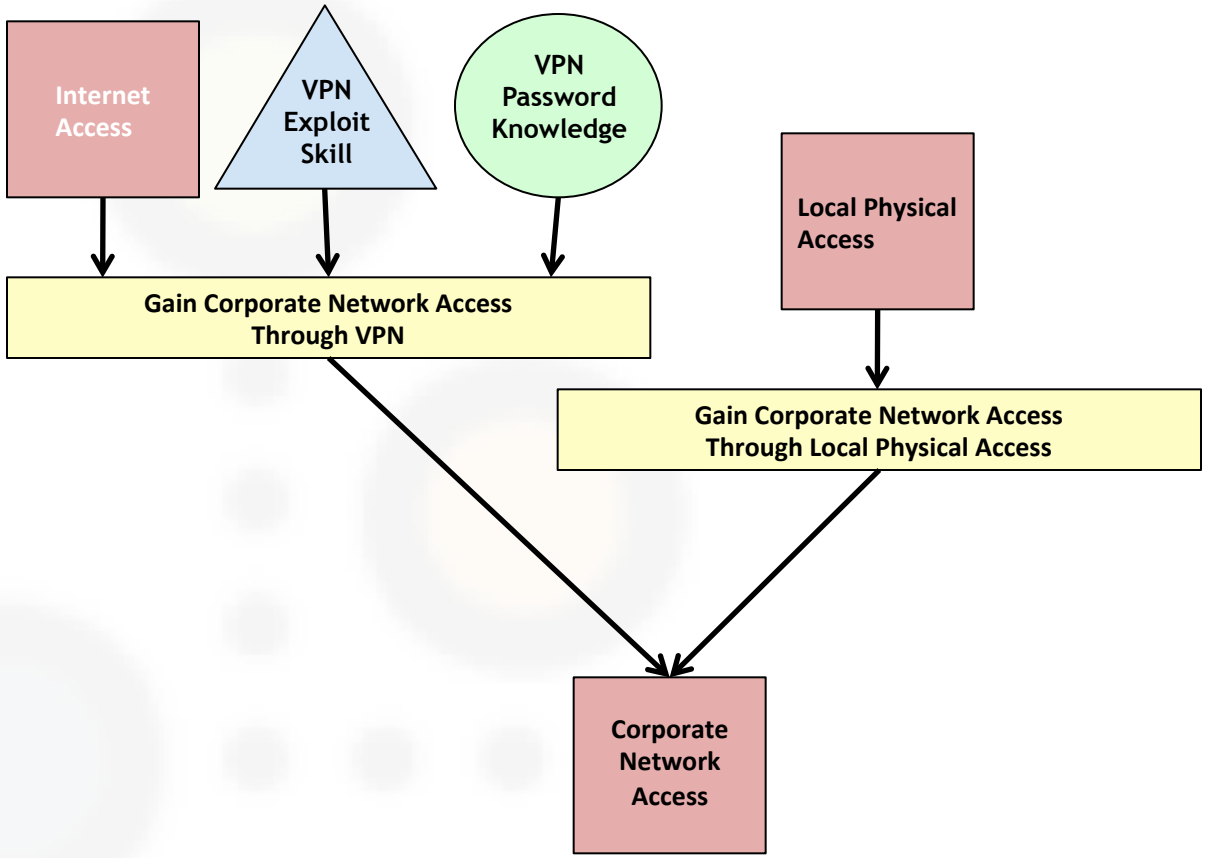
Outline

- Mobius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together


Hacker, Foreign Gov.
Defeat Internet-CorporateLAN FW
Defeat Internet-EngrWorkstation FW
Defeat CorpLAN-SCADA LAN FW
Obtain or Circumvent HMI Login Password
Insider Engineer
Hostile Org.
Insider Engineer
Install Backdoor SW on SCADA LAN
Send Commands to SS from SCADA LAN
Inject Commands to SS via Remote Access
Inject Commands to Reclosers via Radio
Send Commands to Reclosers from SS
Insider Technician, Insider Operator
Sabotage Multiple SS
Sabotage a Single SS
Install Backdoor SW on Reclosers
Cause Service Disruption
Damage Equipment and Cause Disruption

Representing Attacks Against the System

An "attack execution graph" describes potential attack vectors against the system from an attacker point of view. Attempting an attack step requires certain skills, access, and knowledge about the system. The outcome of an attack can affect the adversary's access and knowledge about the system.



ADVISE System Information: Attack Execution Graph

An attack execution graph is defined by

<A, R, K, S, G>,

where

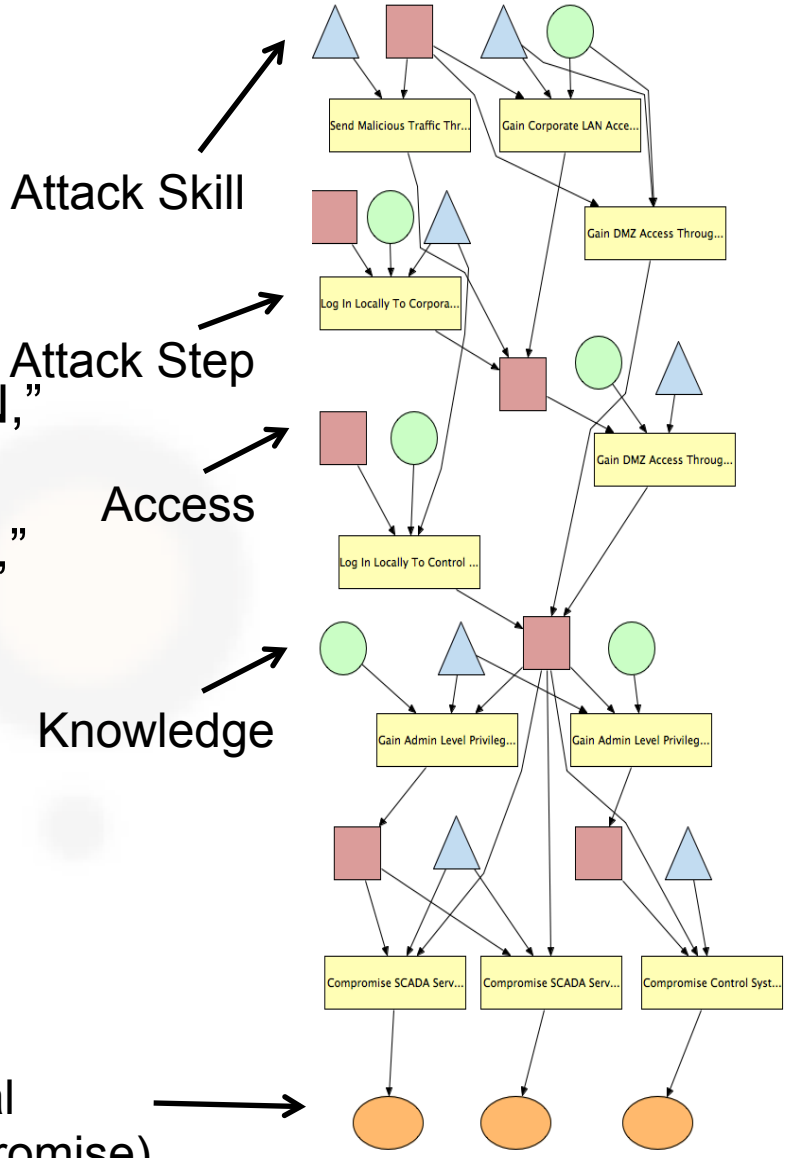
A is the set of attack steps,
e.g., "Access the network using the VPN,"

R is the set of access domains,
e.g., "Internet access," "Network access,"

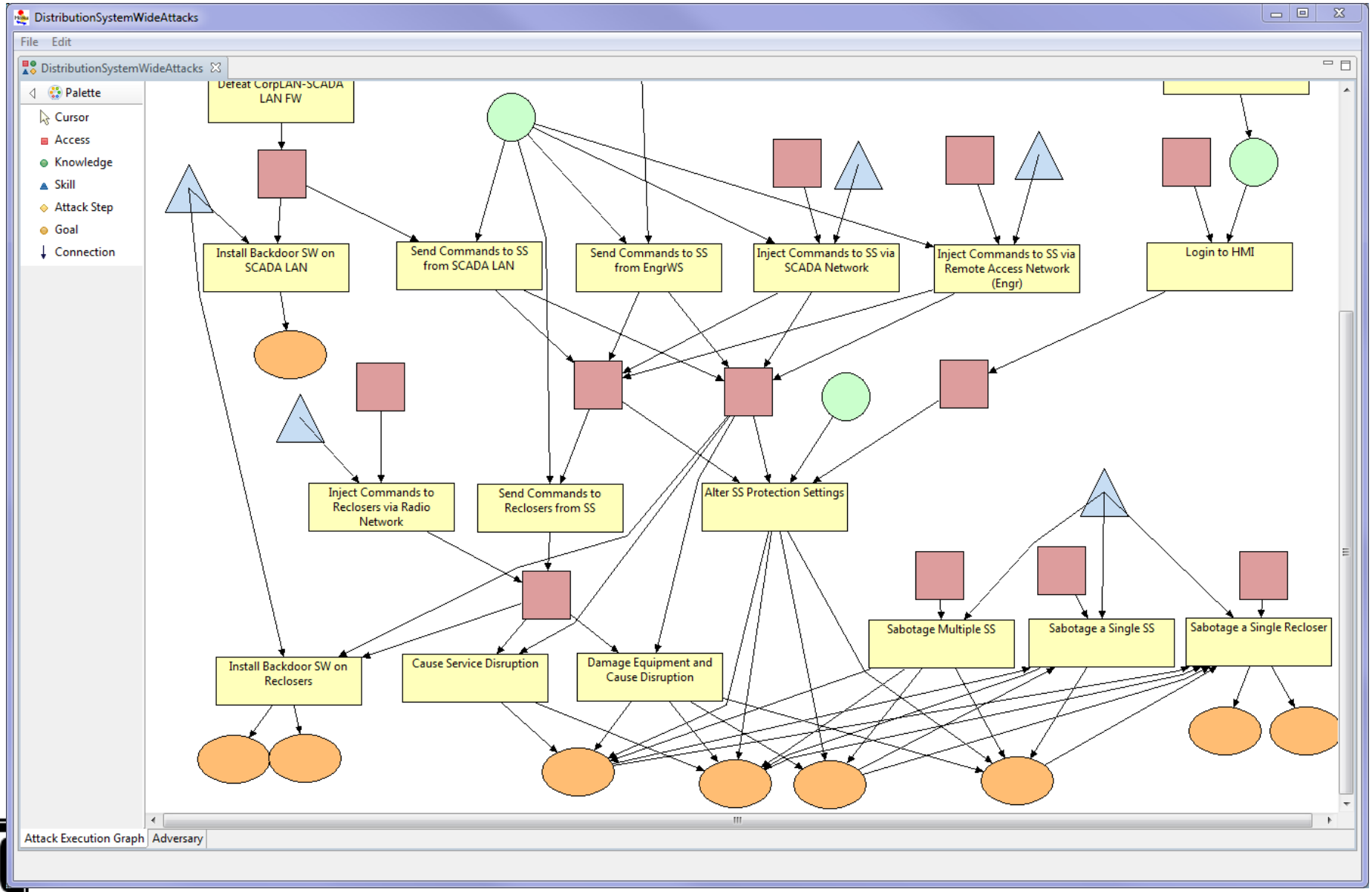
K is the set of knowledge items,
e.g., "VPN username and password"

S is the set of adversary attack skills,
e.g., "VPN exploit skill," and

G is the set of adversary attack goals,
e.g., "View contents of network."



Mobius Attack Execution Graph Editor



Outline

- Mobius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together

ADVISE Adversary Information: Adversary Profile

The adversary profile is defined by the tuple

$<s_0, L, V, w_C, w_P, w_D, U_C, U_P, U_D, N>$,

where

$s_0 \in X$ is the initial model state, e.g., has Internet Access & VPN password,

L is the attack skill level function, e.g. has VPN exploit skill level = 0.3,

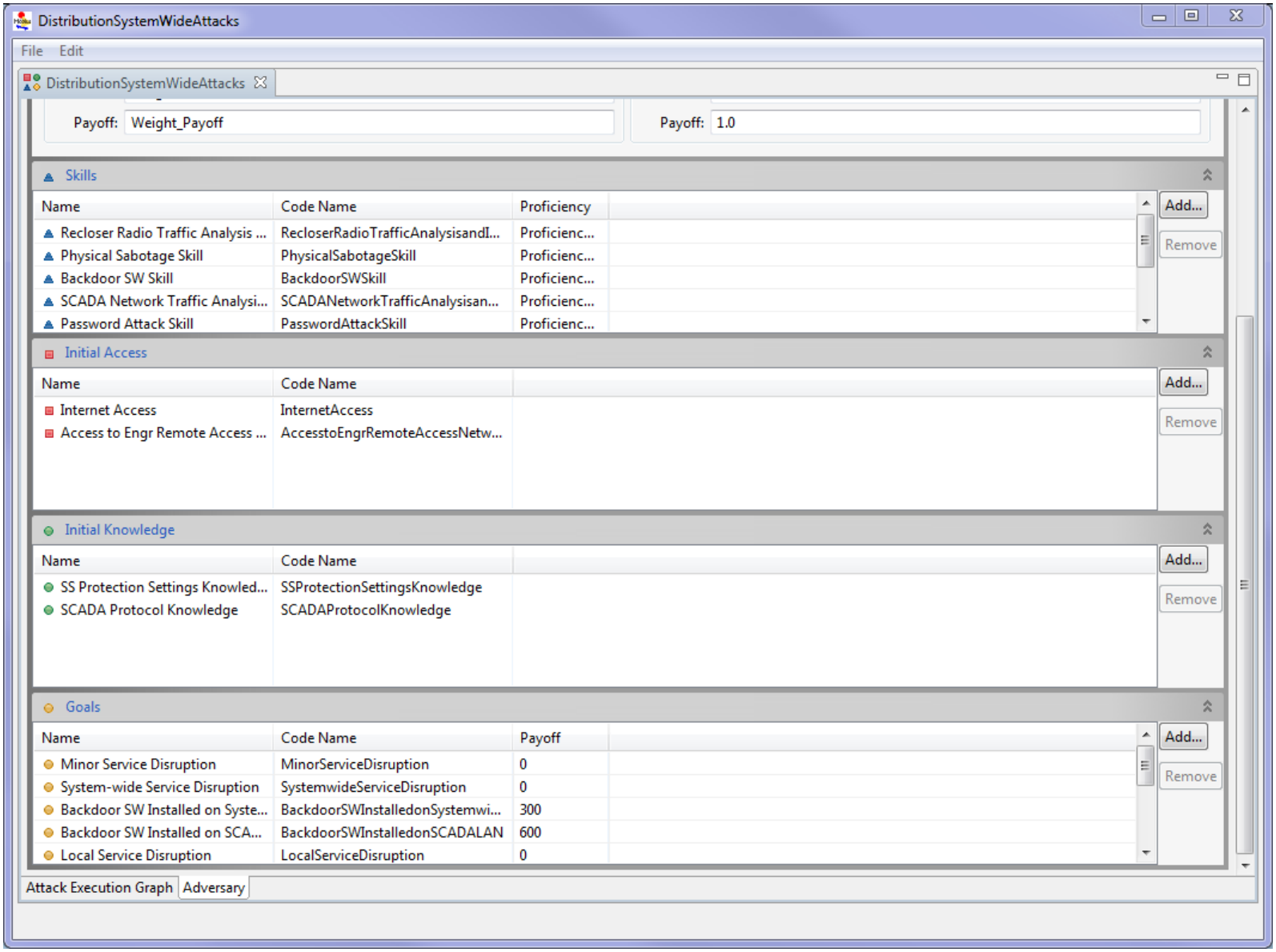
V is the attack goal value function, e.g., values "View contents of network" at $5000,

w_C, w_P, and w_D are the attack preference weights for cost, payoff, and detection probability, e.g., $w_C = 0.7$, $w_P = 0.2$, and $w_D = 0.1$,

U_C, U_P, and U_D are the utility functions for cost, payoff, and detection probability, e.g., $U_C(c)=1 - c/10000$, $U_P(p)=p/10000$, $U_D(d)=1 - d$, and

N is the planning horizon, e.g., N = 4.

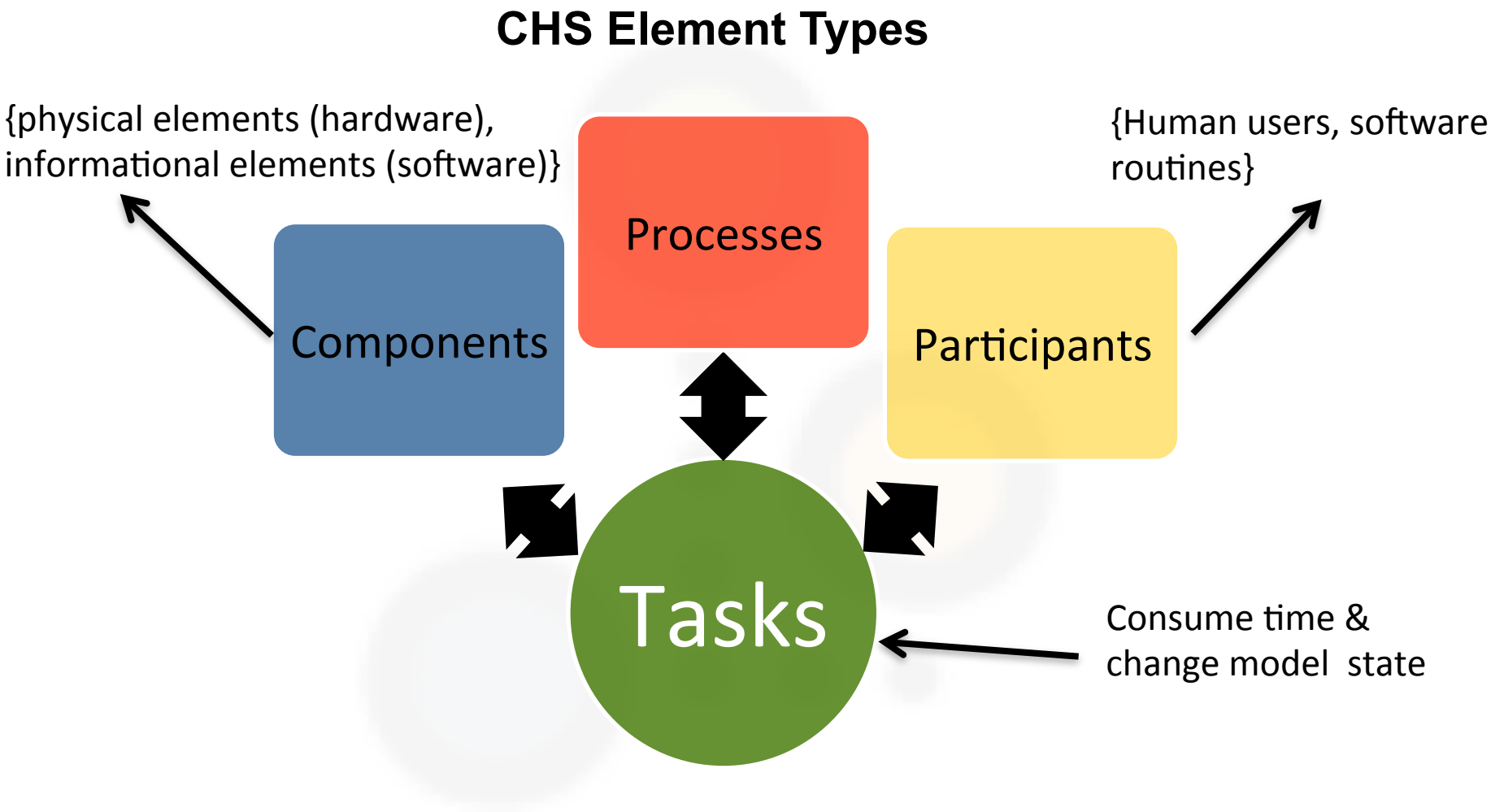
ADVISE Adversary Editor



Outline

- Mobius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together

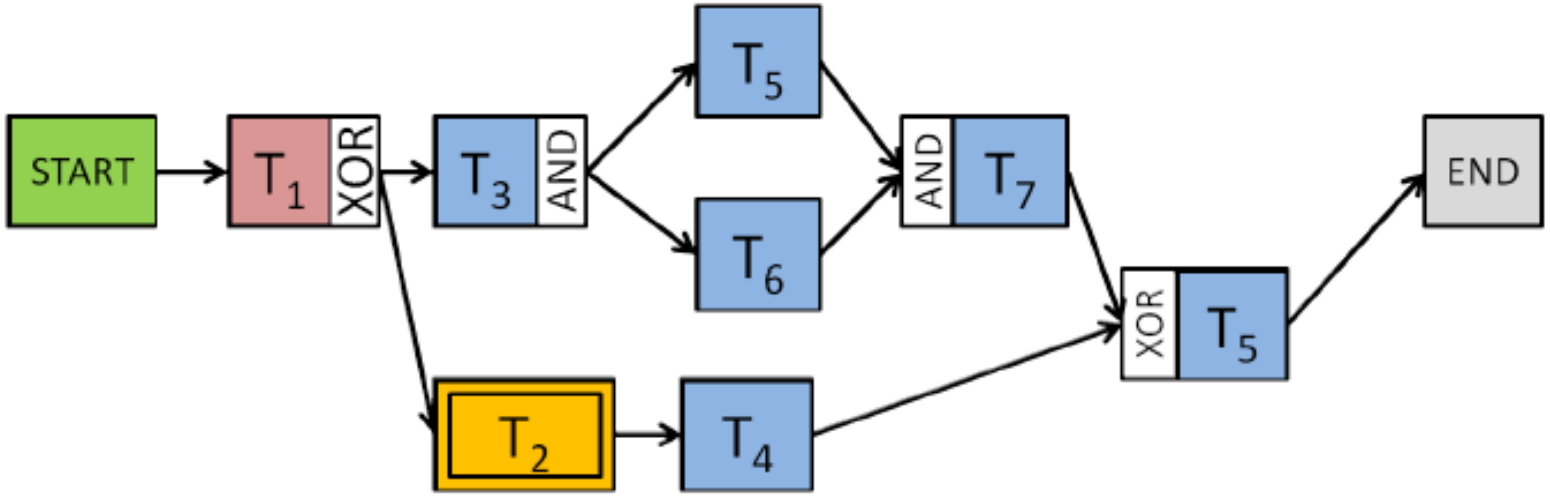
Cyber-Human Systems



a process is a structured flow of tasks performed by one or more participants using system components.

HITOP

- **Process**: a structured flow of tasks performed by one or more participants using system components.
- **Process Instance (PI):** One instance of execution of the process. The state history of a PI describes one path through a process. Also called a *PI token or just token*.



Opportunity-Willingness-Capability

To properly perform a task, a human user must exhibit (as evidenced by the value his/her state variables):

- **Opportunity** – prerequisites for performance like *context*, *participants*, and *tools (Example: possess USB stick)*
- **Willingness** – decision to attempt task (Example: decide to encrypt USB data)
- **Capability** – things which affect task performance like *knowledge*, *skills*, and/or *abilities (Example: training to use encryption software)*

Define *human decision points* for all tasks that involve humans, allowing them to make decisions that effect system security

HITOP Elements and State Variables

Components

C_1

Properties
- Shared: 0
- Avail Resources: 3

Variables
- Speed: 6
- Has Charger: 1
- Charged: 0

C_2

Properties
- Shared: 1
- Avail Resources: 5

Variables
- Speed: 12
- Auto Backup: 1
- Location: Office

C_3

Properties
- Shared: 0
- Avail Resources: 1

Variables
- Speed: 1
- Needs Power: 0
- Location: Home

Participants

H_1

Properties
- MultiTask: 1
- Avail Resources: 10

Variables
- Training Type: B
- Experience Level: 5

H_2

Properties
- MultiTask: 1
- Avail Resources: 5

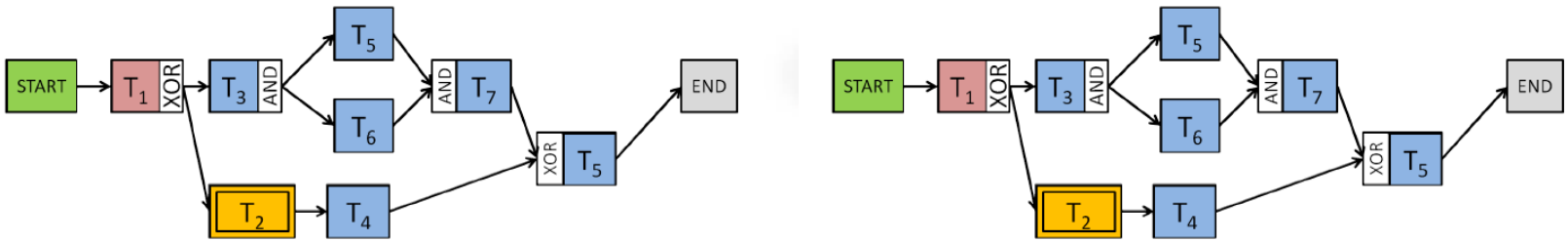
Variables
- Training Type: C,B
- Experience Level: 2
- Role: Trainee

H_3

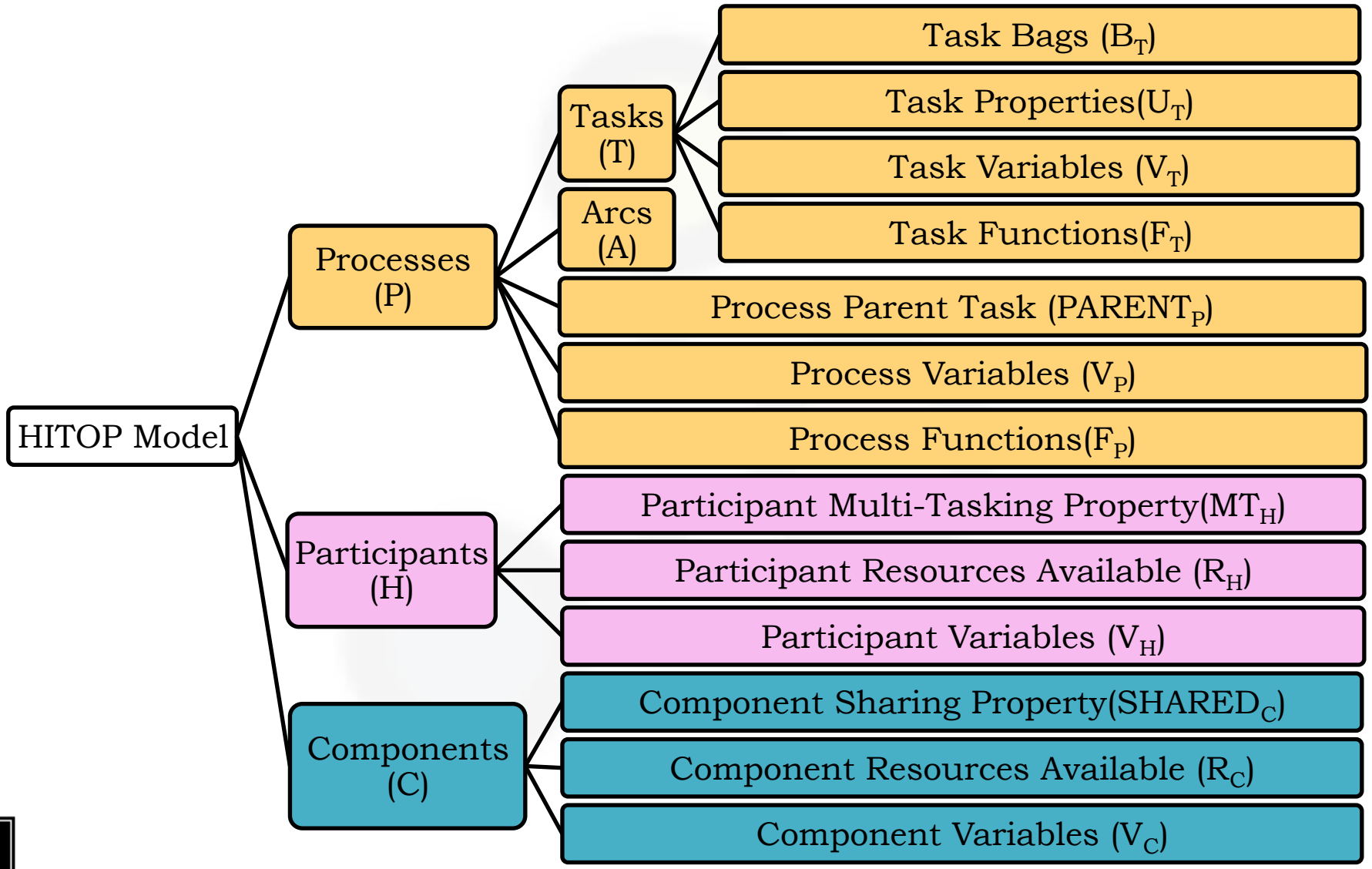
Properties
- MultiTask: 0
- Avail Resources: 8

Variables
- Training Type: A
- Experience Level: 9
- Location: Home

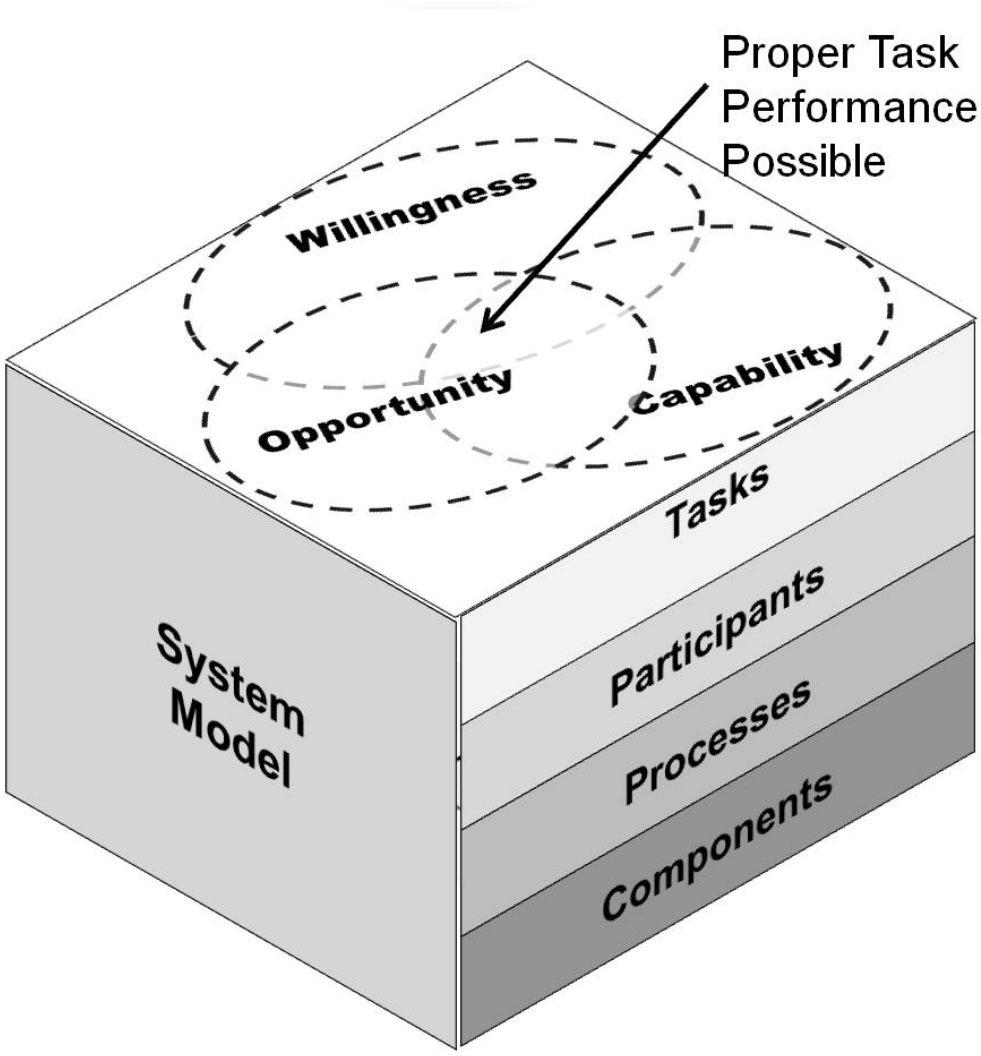
Processes



HITOP System State



HITOP Cyber-Human Modeling Approach

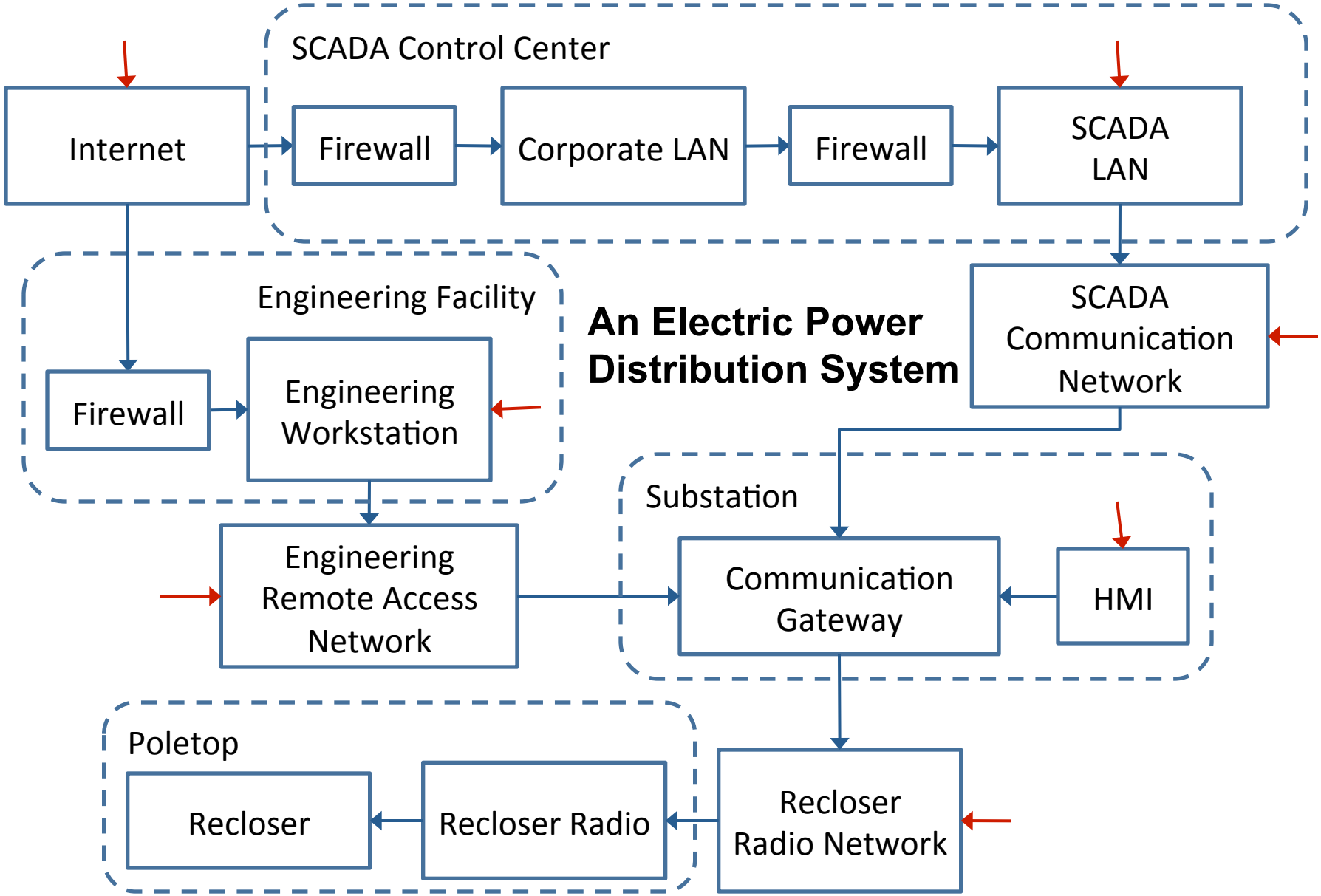


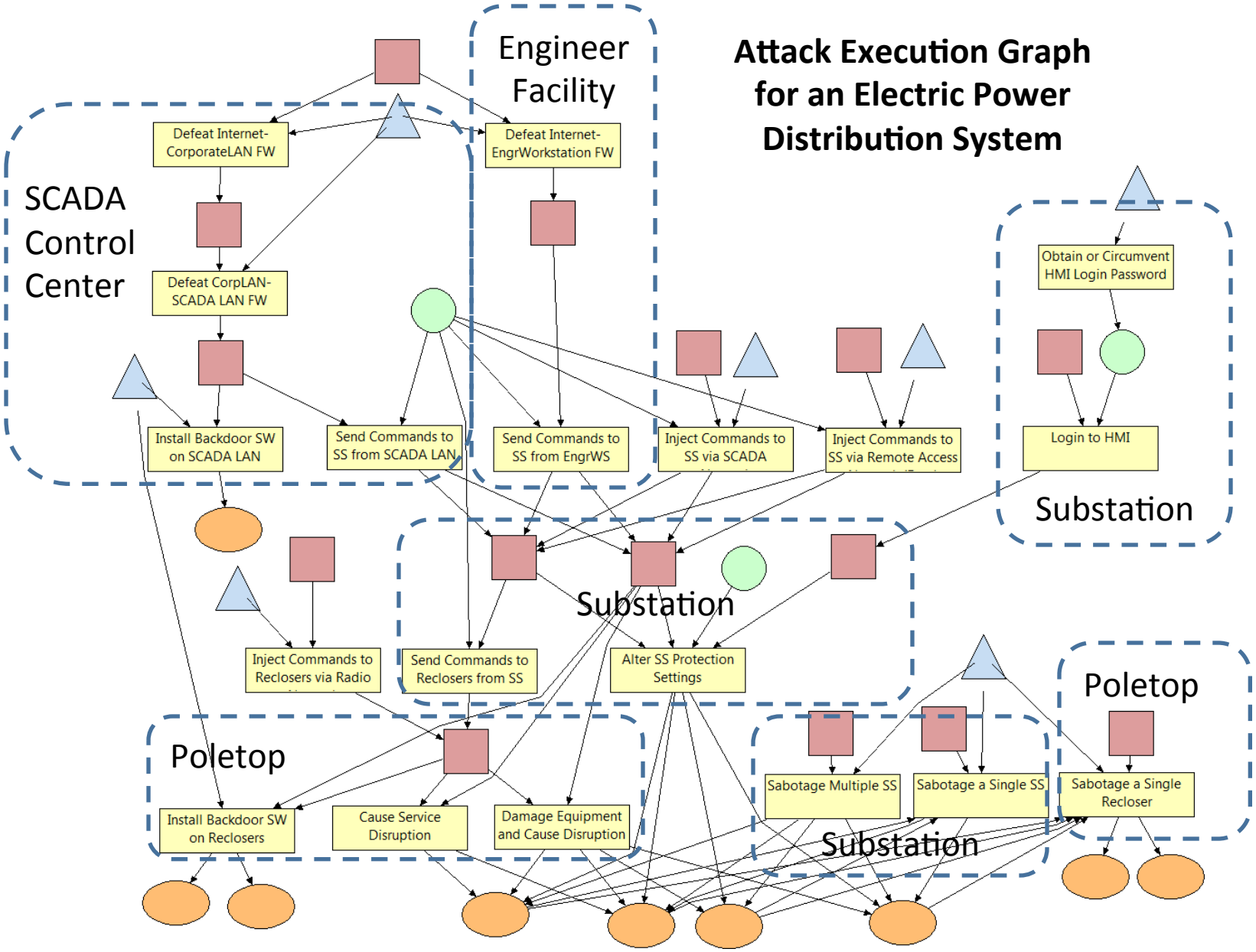
Outline

- Mobius Review
- Attack Execution Graphs
- ADVISE Adversary Modeling Formalism
- HITOP (Human) User Formalism
- Putting it all Together

Case Study

- Investigates the effects of architectural changes on the security of an electric power distribution system
- In particular, analyze the security impact of adding radio communication between substations and poletop reclosers


SCADA Control Center
Internet
Firewall
Corporate LAN
Firewall
SCADA LAN
Engineering Facility
An Electric Power Distribution System
SCADA Communication Network
Firewall
Engineering Workstation
Substation
Engineering Remote Access Network
Communication Gateway
HMI
Poletop
Recloser
Recloser Radio
Recloser Radio Network

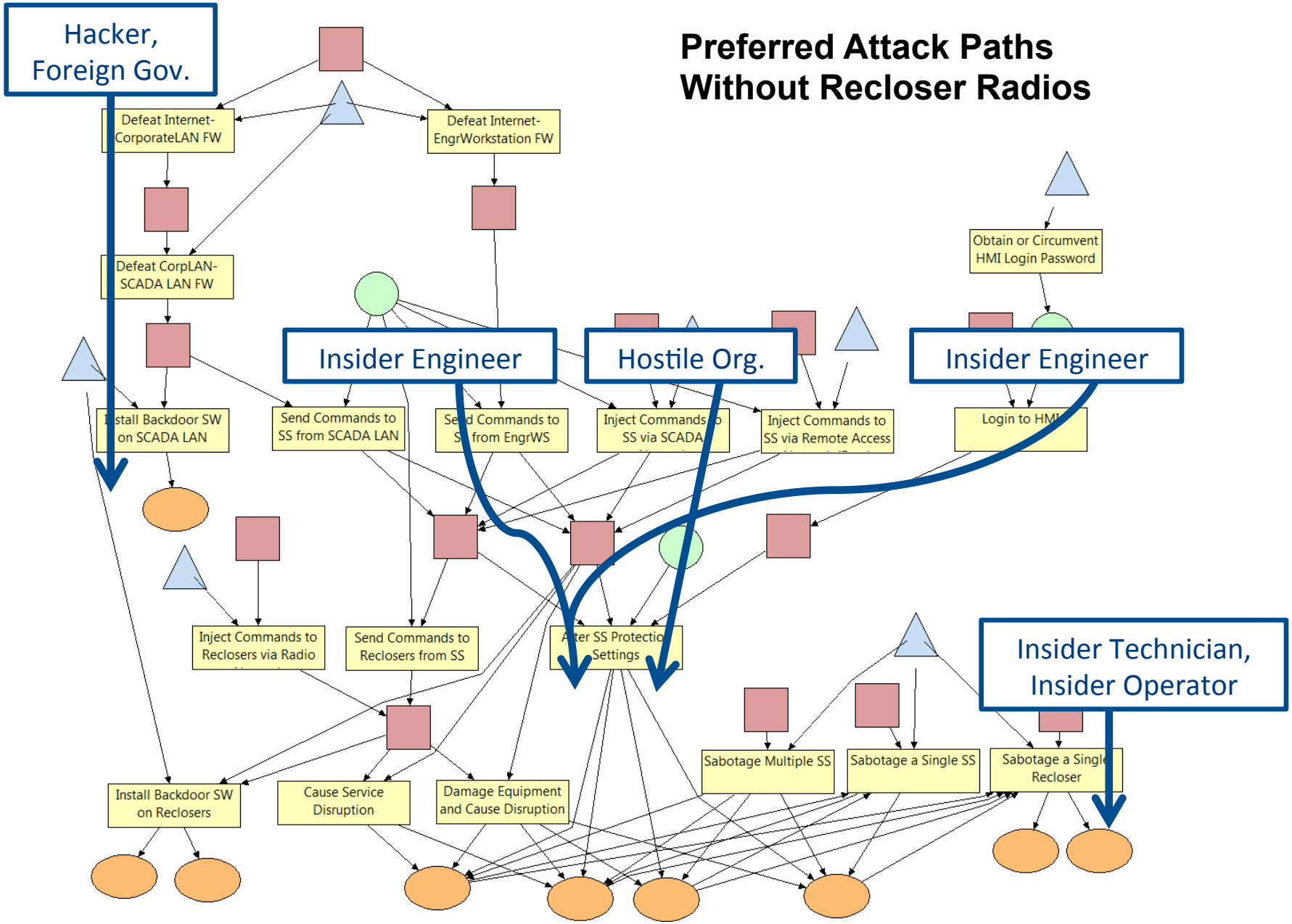

Attack Execution Graph for an Electric Power Distribution System
Engineer Facility
SCADA Control Center
Substation
Substation
Substation
Poletop
Poletop
Defeat Internet-CorporateLAN FW
Defeat Internet-EngrWorkstation FW
Defeat CorpLAN-SCADA LAN FW
Install Backdoor SW on SCADA LAN
Send Commands to SS from SCADA LAN
Send Commands to SS from EngrWS
Inject Commands to SS via SCADA
Inject Commands to SS via Remote Access
Obtain or Circumvent HMI Login Password
Login to HMI
Inject Commands to Reclosers via Radio
Send Commands to Reclosers from SS
Alter SS Protection Settings
Install Backdoor SW on Reclosers
Cause Service Disruption
Damage Equipment and Cause Disruption
Sabotage Multiple SS
Sabotage a Single SS
Sabotage a Single Recloser

Adversary Profiles: Decision Parameters

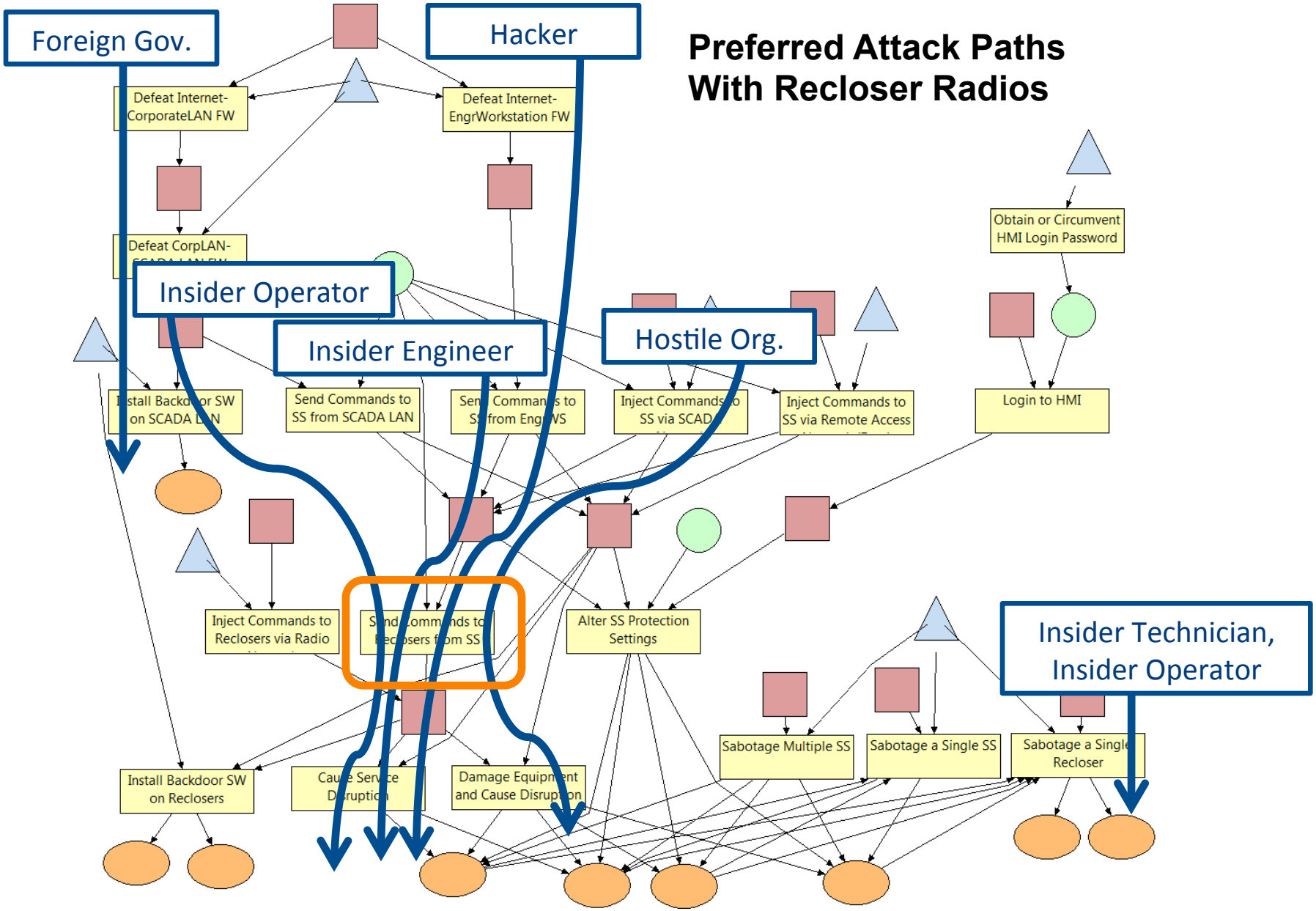
	Foreign Government	Hacker	Hostile Organization	Insider Engineer	Insider SCADA Operator	Insider Remote Technician
Cost Preference Weight	0	0.2	0.05	0.2	0.2	0.2
Detection Preference Weight	0.5	0.4	0.2	0.1	0.1	0.1
Payoff Preference Weight	0.5	0.4	0.75	0.7	0.7	0.7

- The Foreign Government adversary is very well-funded but risk-averse.
- The Hacker is resourced-constrained.
- The Hostile Organization is moderately well-funded and more driven by payoff than the others.
- The Insider Engineer, Insider Technician, and Insider Operator are resource-constrained but willing to take risks.

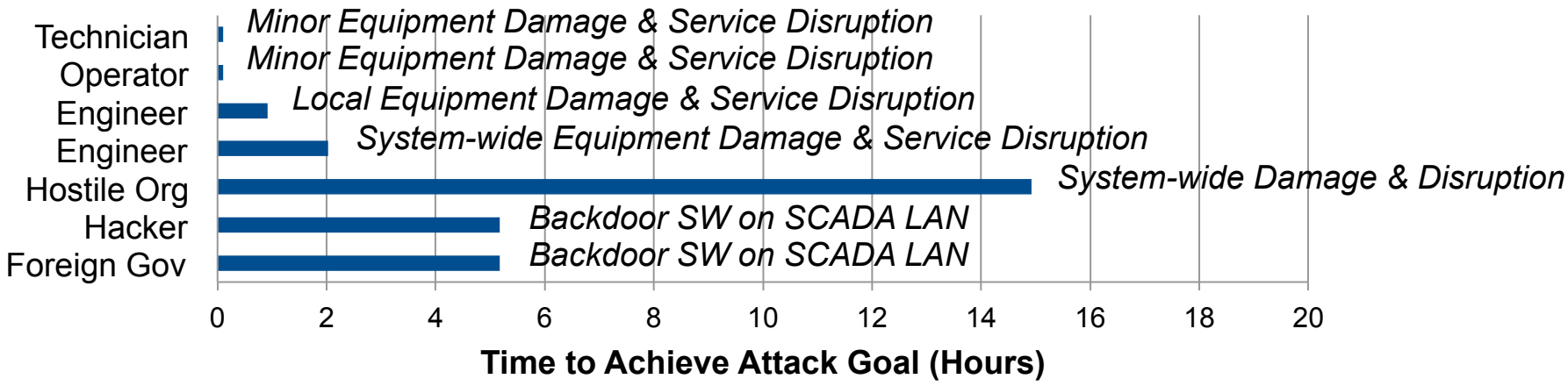
Preferred Attack Paths Without Recloser Radios



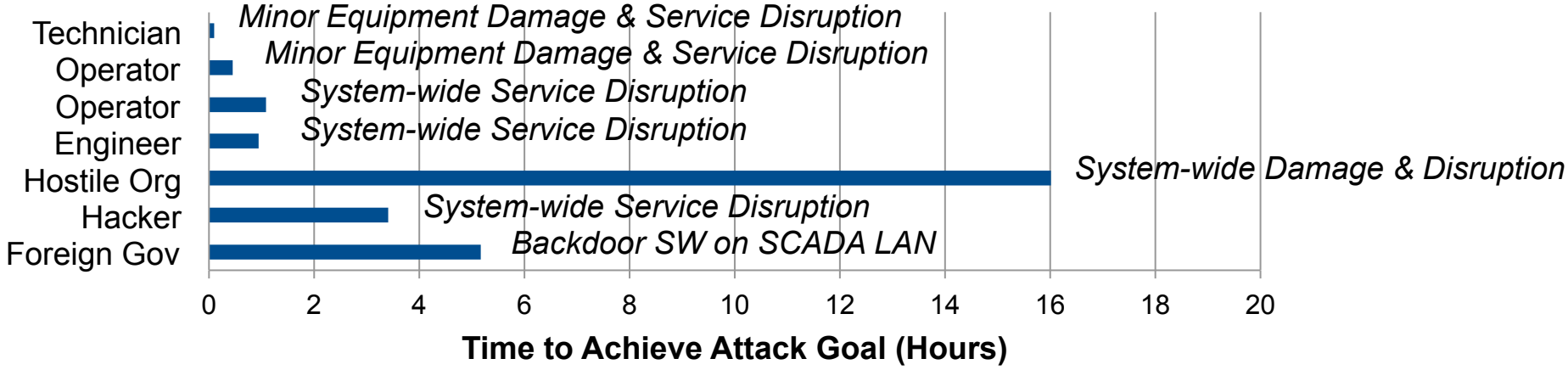
Preferred Attack Paths With Recloser Radios



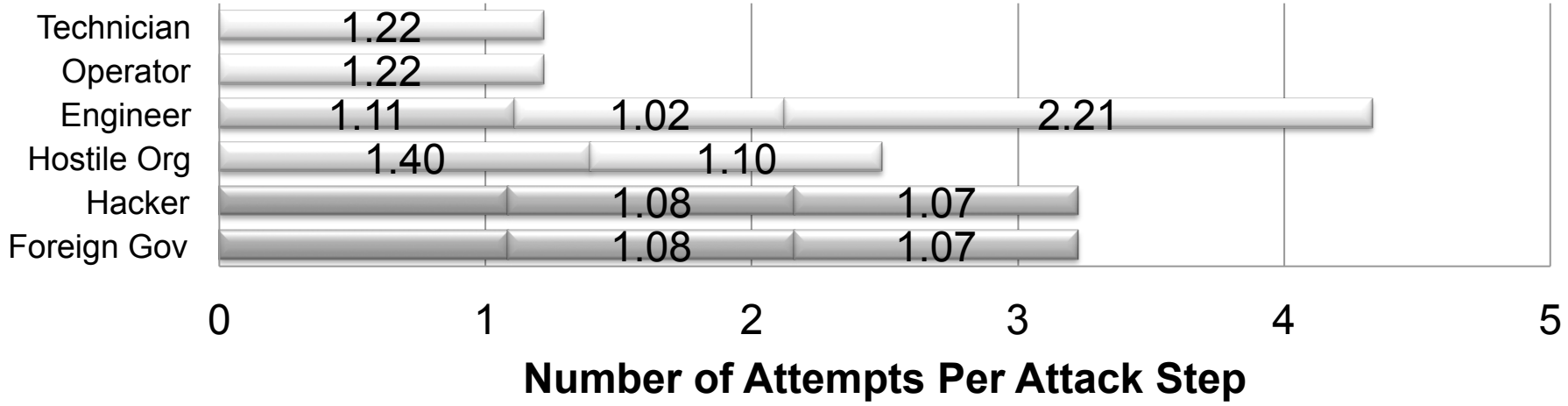
Attack Speed Without Recloser Radios



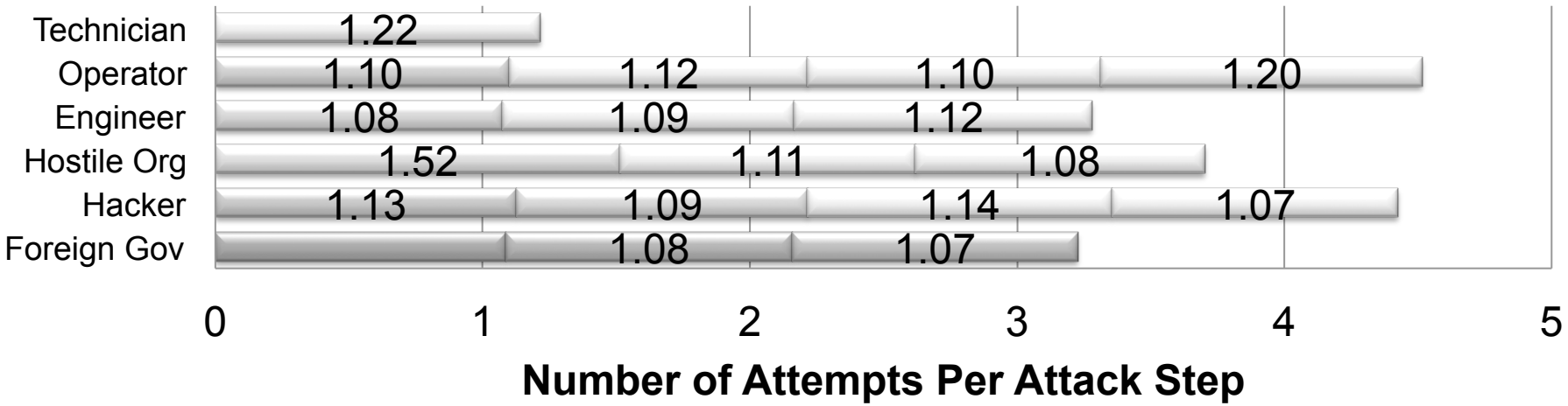
Attack Speed With Recloser Radios



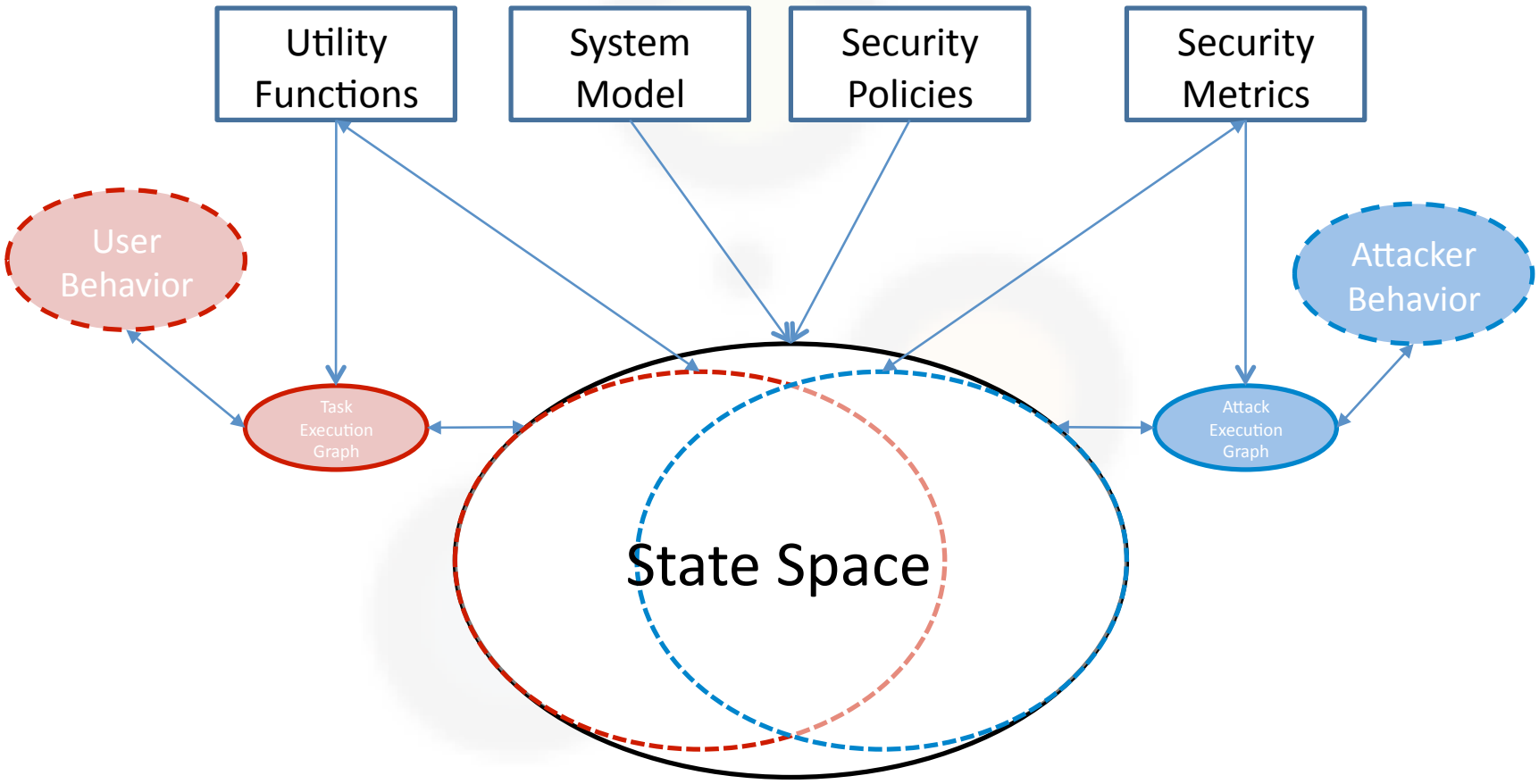
Number of Attack Attempts Without Recloser Radios



Number of Attack Attempts With Recloser Radios



Ultimate Goal: Simultaneous Modeling of Cyber-System (Cyber and Physical Elements), Users, and Attackers



"Built-In" Resilience of Societal-Scale Systems

- Must:
  - Consider Software, Hardware, Physical, and Human Components in a Holistic Manner
  - Understand that many systems cannot be perfectly secure, and must instead be resilient
  - Be designed using tools that allow one to make architectural design choices based on quantitative resiliency assessments
  - Be built from components that have known security and resiliency properties