

Making Sound Cyber Security Decisions Through a Quantitative Metrics Approach

May 4, 2011

Bill Sanders

Joint work with Elizabeth LeMay, Ken Keefe,
and Carol Muehrcke

University of Illinois at Urbana-Champaign



The Problem

- Systems operate in adversarial environments
 - Adversaries seek to degrade system operation by affecting the confidentiality, integrity, and/or availability of the system information and services
 - “Secure” systems must be able to meet their operational objectives despite attack attempts by adversaries
- System security is not absolute
 - No real system is perfectly secure
 - Some systems are more secure than others
 - *But how much more secure are they?*



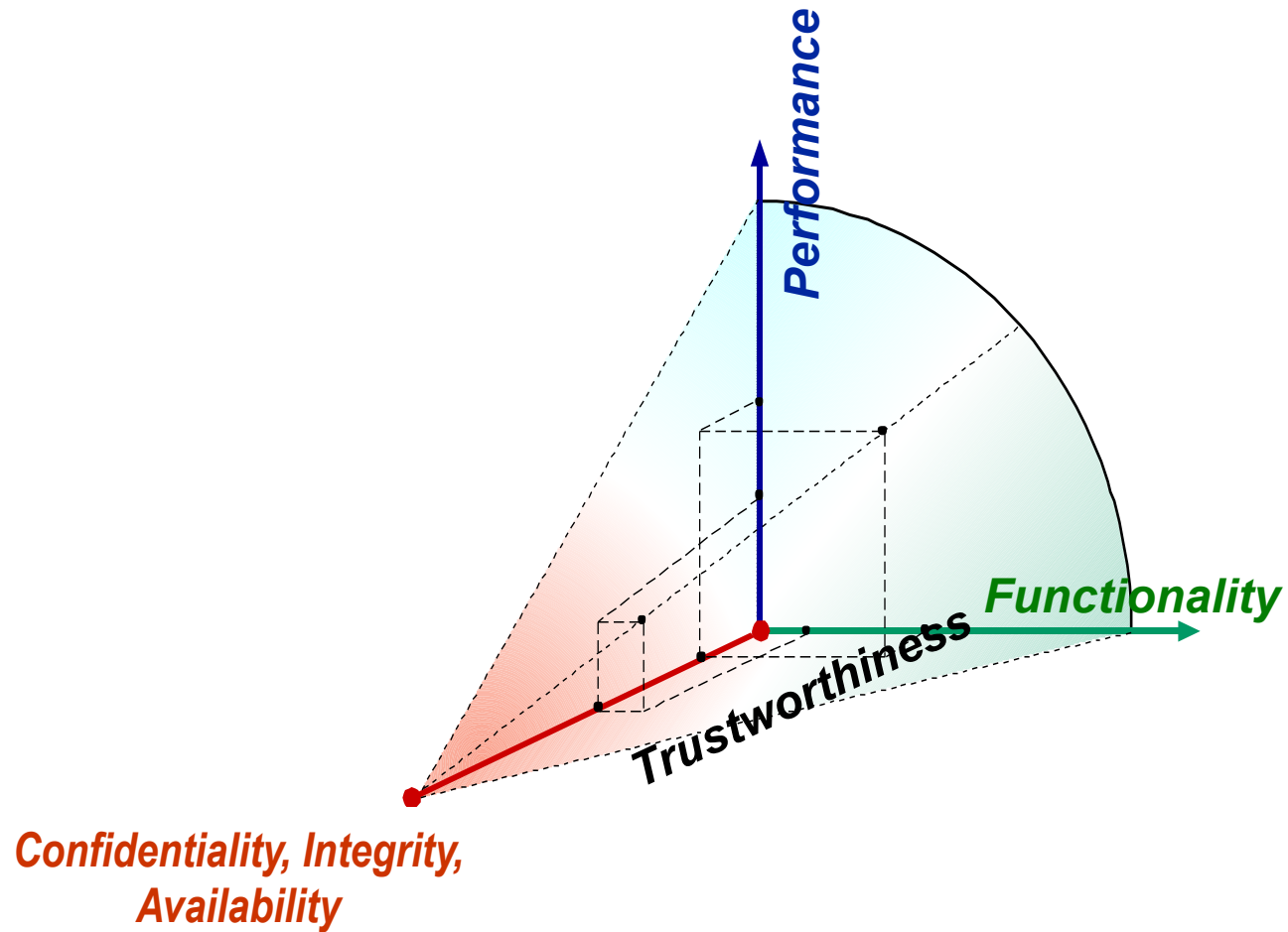
Quantifying Security

- At design time
 - System architects make trade-off decisions to best meet all design criteria
 - Other design criteria can be quantified: performance, reliability, operating and maintenance costs, etc.
 - *How can we quantify the security of different system designs?*
- During system operation and maintenance
 - Modifying the system architecture can improve or worsen system security
 - *How can we compare the security of different possible system configurations?*



Model-based system-level security evaluation

Ultimate Goal: Placing a System in the Cone



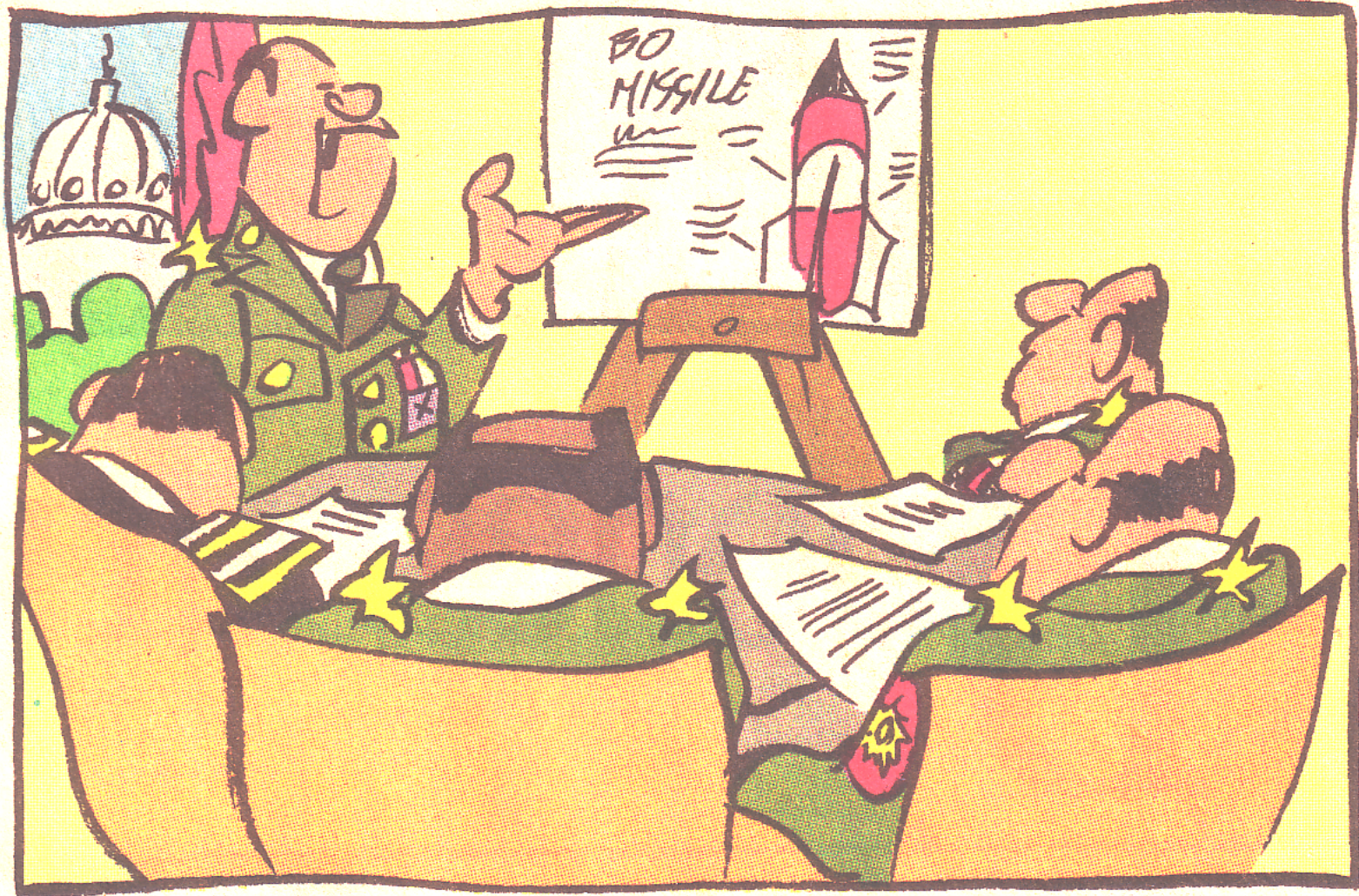
Contrasting Approaches

Typical Situation Today:

- Process:
 - Rely on a trusted analyst that examines situation, and gives advice based on experience, or
 - Form decision in a collective manner based on informal discussions among stakeholder experts
- *Limitations:*
 - No way to audit decision process
 - No quantifiable ranking of alternative options

Goal For Tomorrow:

- Usable tool set that enables diverse stakeholders to express
 - Multi-faceted aspects of model
 - Multiple objectives
- Way for diverse stake holders to express concerns and objectives in common terminology
- Quantifiable ranking of alternate security policies and architectures
- Auditable decision process



***“It has the ultimate fail-safe feature.
It doesn’t work.”***



THE MYTHICAL FIVE NINES. 99.999%. AS CLOSE TO PERFECT AS YOU CAN GET WITHOUT BREAKING SOME LAW OF NATURE.

For a server operating system, the five nines are a measure of reliability that translates into just over five minutes of server downtime per year.* For your business, that means servers are up and running when people need them. Of course, rumors of this 99.999% uptime usually start under ideal lab conditions. But where are these five nines when your business needs them? If you're using Microsoft® Windows® 2000 Server-based solutions, they may be closer than you think. Today Starbucks, FreeMarkets and MortgageRamp, an affiliate of GMAC Commercial Mortgage, are using Windows 2000 Server-based systems designed to deliver 99.999% server uptime. Of course, not all installations require this level of reliability, but one thing is for sure: The Windows 2000 Server family can help you get to the level of reliability you need. In fact, industry leaders such as Compaq, Hewlett-Packard, Unisys, Stratus and Motorola Computer Group can work with you to deliver solutions with up to five nines uptime. To learn more about server solutions you can count on, visit microsoft.com/windows2000/servers Software for the Agile Business.



Microsoft

*The level of reliability is dependent on many factors outside of the operating system, including other hardware and software technologies, physical critical operational processes and geographical services. © 2000 Microsoft Corporation. All rights reserved. Microsoft, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

SigDeB Exchange ...

From: XXX
Sent: 24 January 2001 16:56
To: WG 10.4 SIG INT
Distribution
Subject: Microsoft OS
Availability Claims

Have you all seen the recent very large ads by Microsoft claiming .99999 "Reliability" for their Server OSs?

There's some very small print at the bottom of the ads that points out that many factors other than the OS affect overall delivered service availability.

Does anyone have access to the data and logic that Microsoft used as the basis for their claim?

A review of their approach could provide useful inputs to our SIGINT discussions.

Response:

We applied the same rigorous scientific analysis techniques as other companies who provide availability guarantees.

Followup Question:

Could you give us a Microsoft point of contact who one of us could contact to ask for an explanation of the data and rationale that they're willing to disclose to back up their claim?

Response:

Sadly as with other companies the full data and rationale are company confidential, although we do provide a synopsis of the logic of the argument, see

<http://www.microsoft.com/MSCorp/presspass/Features/2001/Jan01/01-11w2kse>

rver.asp which also provides some additional links. Another useful link in this area is

<http://www.microsoft.com/technet/reliable/default.asp>.

Everyone says it is important, few approaches exist ...

- Security metrics were an **important problem in the 2005 INFOSEC Research Council Hard Problems List**
- **New security metrics that are linked to the business were ranked first** among six key security imperatives developed by over twenty Fortune 500 firms
- **New regulatory requirements of Sarbanes-Oxley and the Basel II Accord have created more urgency for metrics that integrate security risk with overall business risk**
- **Almost every critical infrastructure roadmap lists security metrics as a critical challenge**
- **The list goes on ...**

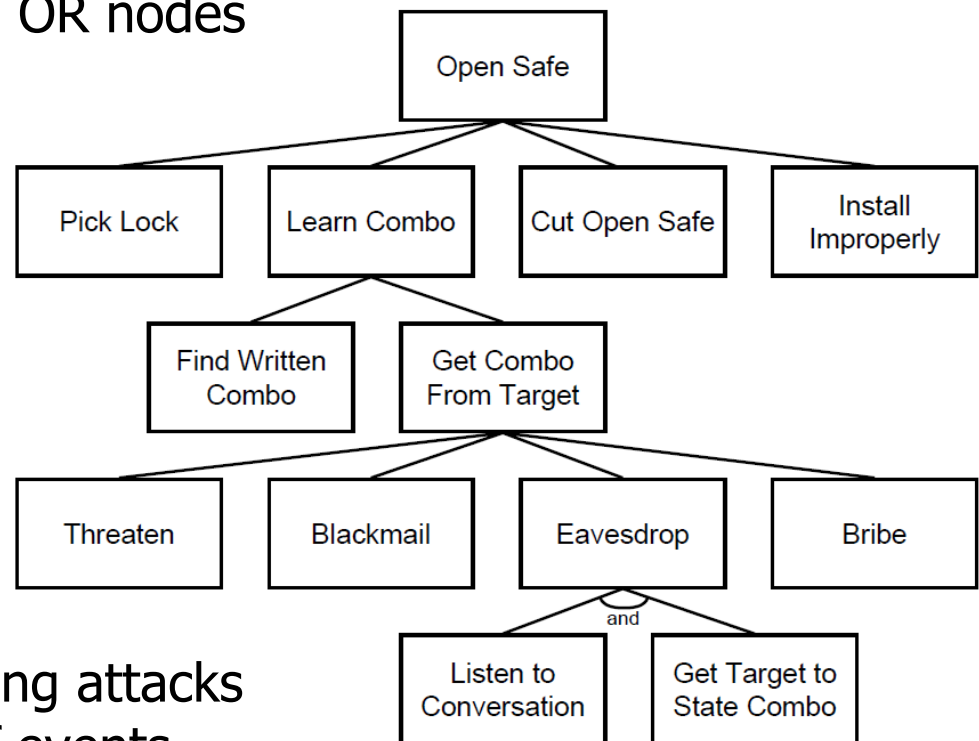


Related Work

- System Model-based security analysis
 - Attack Trees
 - Attack Graphs and Privilege Graphs
- Adversary-based security analysis
 - MORDA and MIRROR
 - NRAT

Attack Trees

- Attack trees describe how sets of events can constitute a security compromise
 - Events are represented as leaf nodes and are joined together with AND and OR nodes
 - A security compromise is represented as the root node
- Contribution
 - Useful for thinking about multiple ways an attacker can reach an attack goal
- Limitation
 - Do not contain a notion of time, which prohibits expressing attacks as time-ordered sequences of events

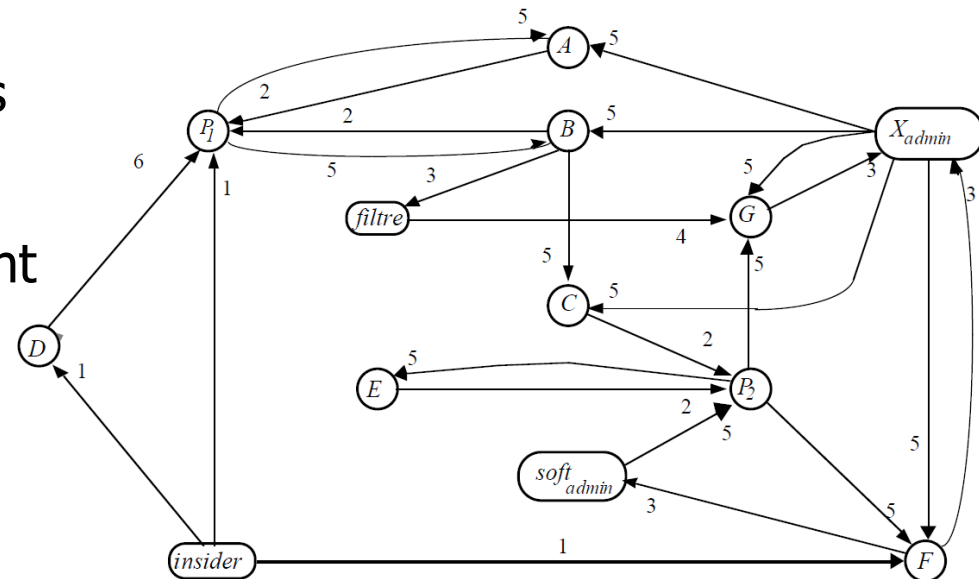


From:

B. Schneier, "Attack trees: Modeling security threats," in Dr. Dobb's Journal: CMP Media, Dec. 1999.

Attack Graphs and Privilege Graphs

- Attack graphs and privilege graphs are both state-based attack analysis methods.
 - The nodes in a privilege graph represent privilege states
 - An attacker starts at one node and works toward an attack goal by transitioning to new privilege states
- Contribution
 - Enable state-based analysis
- Limitation
 - Do not consider the different attack goals and attack preferences of individual adversaries



1) X can guess Y's password; 2) X is in the Y's ".rhost"; 3) Y is a subset of X; 4) X can attack Y via Email; 5) Y uses a program owned by X; 6) X can modify a "setuid" program owned by Y.

From:

M. Dacier, Y. Deswartes, and M. Kaaniche. Quantitative assessment of operational security models and tools. Technical Report Research Report 96493, LAAS, May 1996.



Adversary-based security analysis

- MORDA
 - MORDA assesses system risk by calculating attack scores for a set of system attacks. The scores are based on adversary attack preferences and the impact of the attack on the system mission.
 - A version of MORDA is commercially available as MIRROR
- NRAT
 - NRAT assesses mission risk by computing the attack competency of potential attackers and the system vulnerability.
 - These computations are performed by examining attributes of the threat actors (adversaries), the attacks, and the information system protection (defense).
- Contribution
 - Provide a security analysis informed by adversaries' attributes
- Limitation
 - Not designed for state-based analysis. The adversarial decision is represented as a one-time selection of a full attack vector.



ADversary VIEW Security Evaluation (ADVISE) approach

- Adversary-driven analysis
 - Considers characteristics and capabilities of adversaries
- State-based analysis
 - Considers multi-step attacks
- Quantitative metrics
 - Enables trade-off comparisons among alternatives
- Mission-relevant metrics
 - Measures the aspects of security important to owners/operators of the system
- Auditable and repeatable analysis method

ADVISE Model-Based Metrics Vision

Ask Security Decision Question

Specify Metrics

Describe Adversaries

Describe System

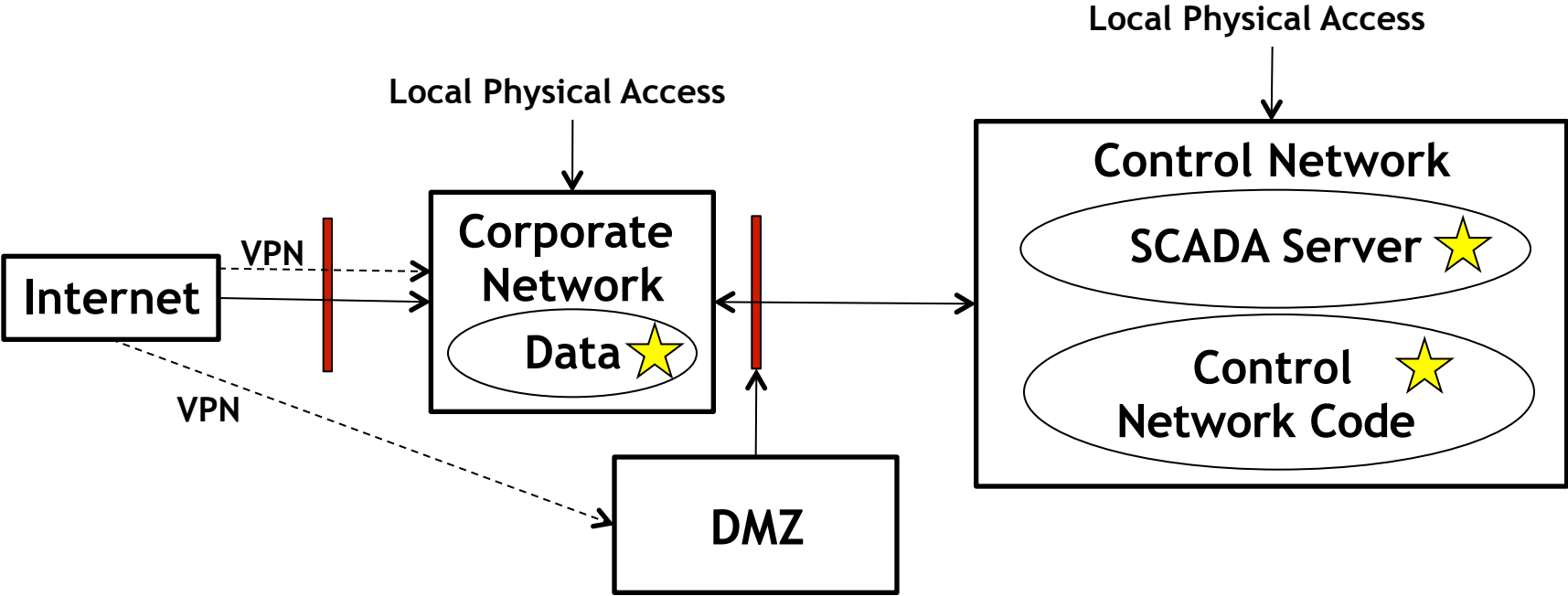
Build an executable stochastic model describing how the adversaries are likely to attack the system

Solve the executable model

Calculate metrics from model results

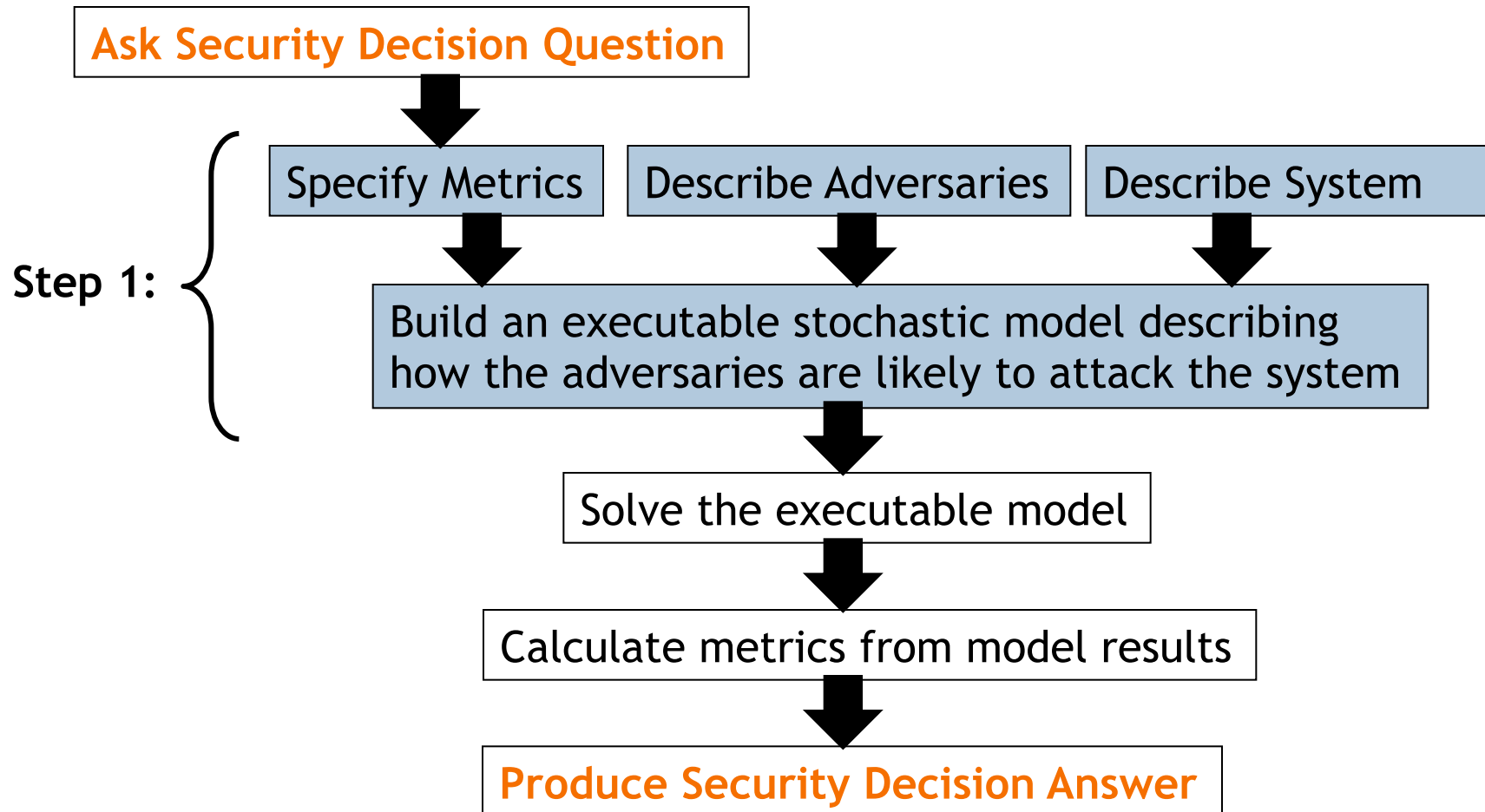
Produce Security Decision Answer

Example: SCADA System

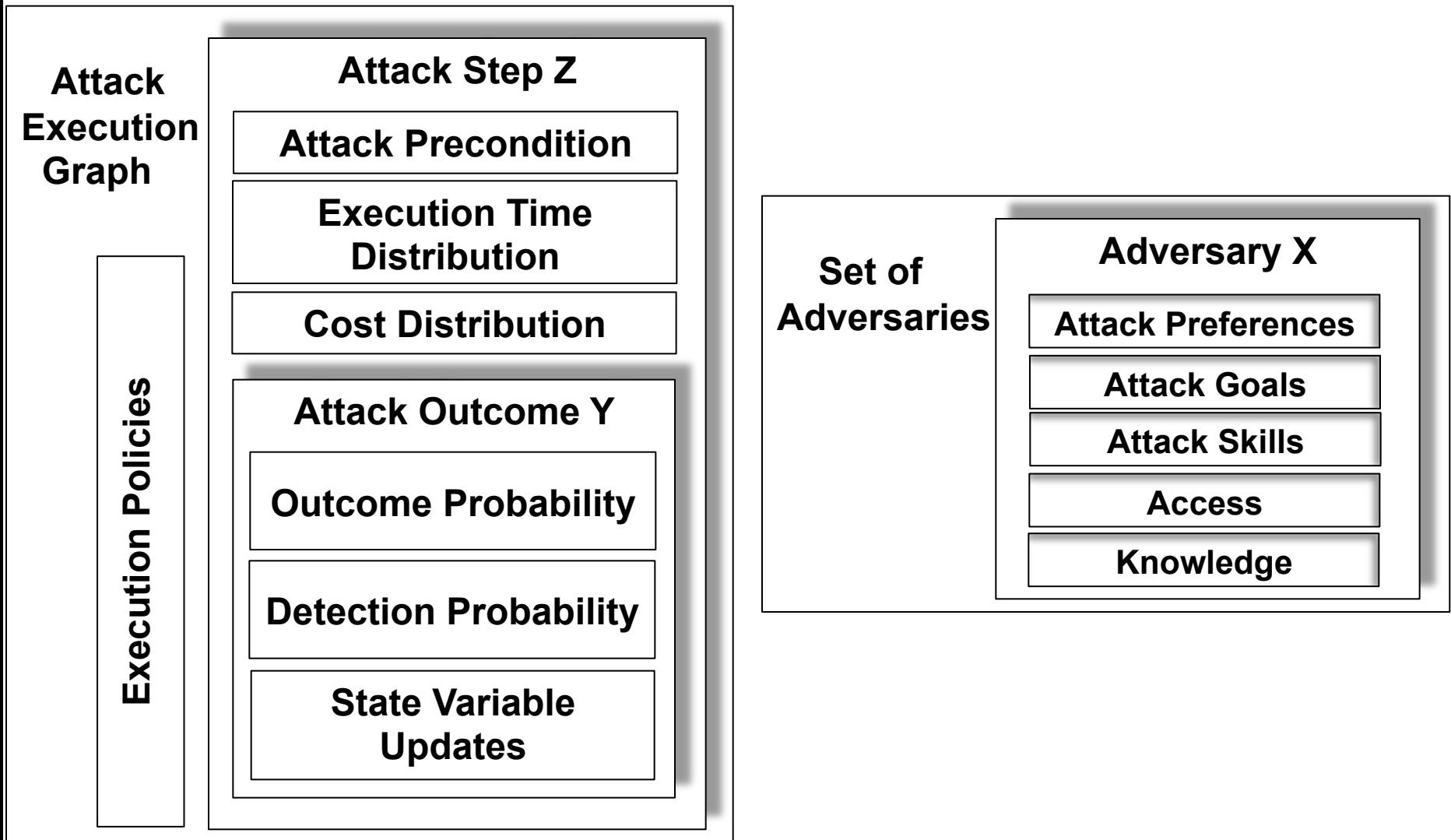


★ = Attack Target

Process for Producing Metrics



ADversary View Security Evaluation (ADVISE)



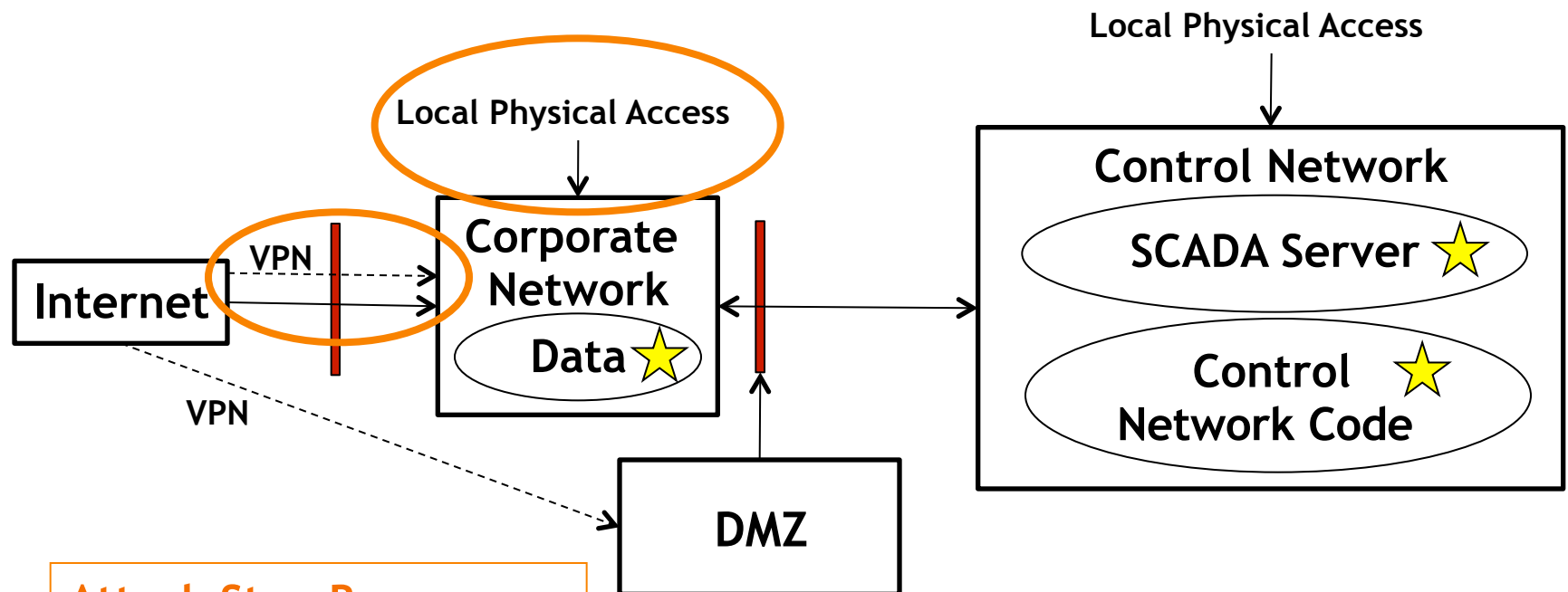
Example: Security Decision

- Question
 - Is the corporate network security budget better spent on upgrading the local physical security or the remote access network security?
- Adversary
 - Economic competitor who wants confidential corporate data
- Metric
 - Probability of confidential corporate data compromise in one year
- System configurations to compare
 - Baseline: current system configuration
 - Physical security upgrade
 - Network security upgrade



Example: SCADA System Attack

**Attack Step A:
Gain Corporate Network Access
Through Local Physical Access**

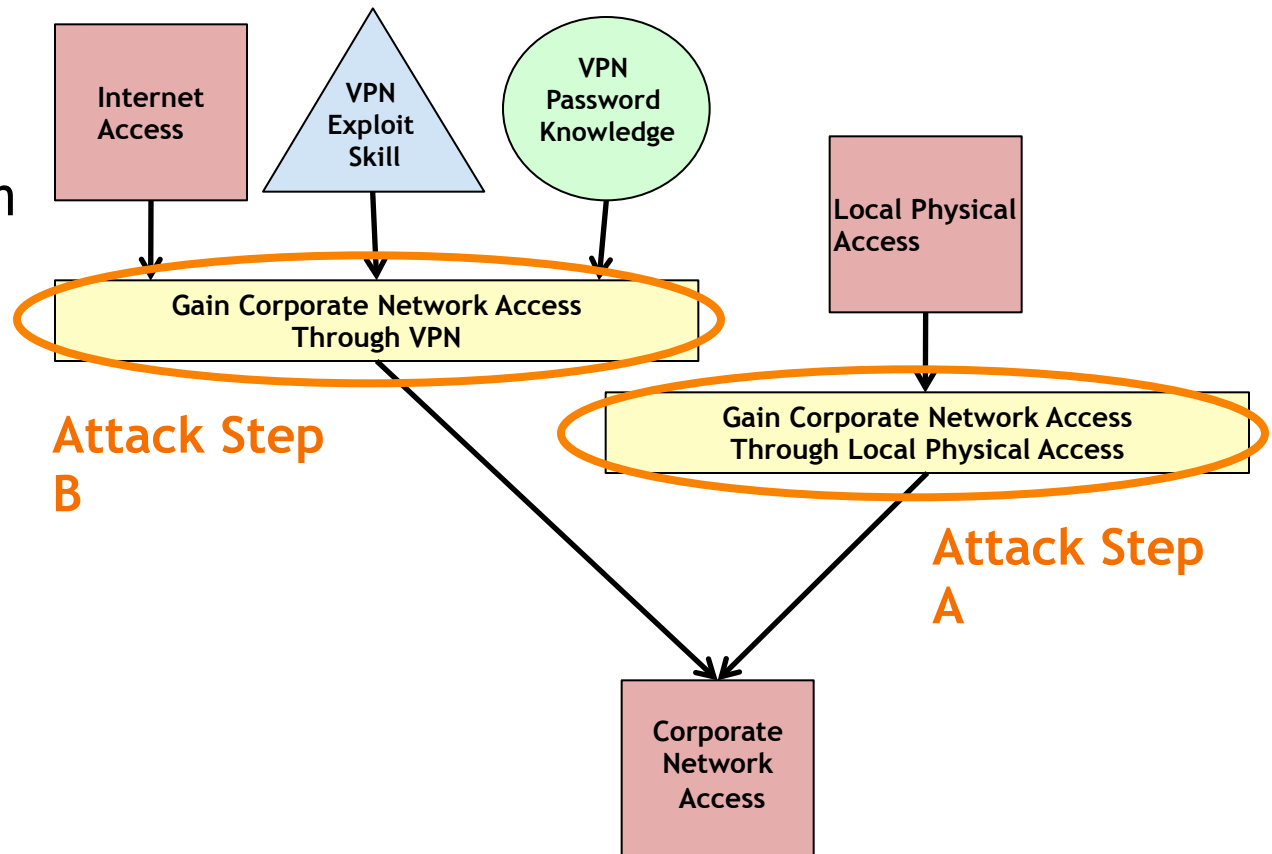


**Attack Step B:
Gain Corporate
Network Access
Through VPN**

★ = Attack Target

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.



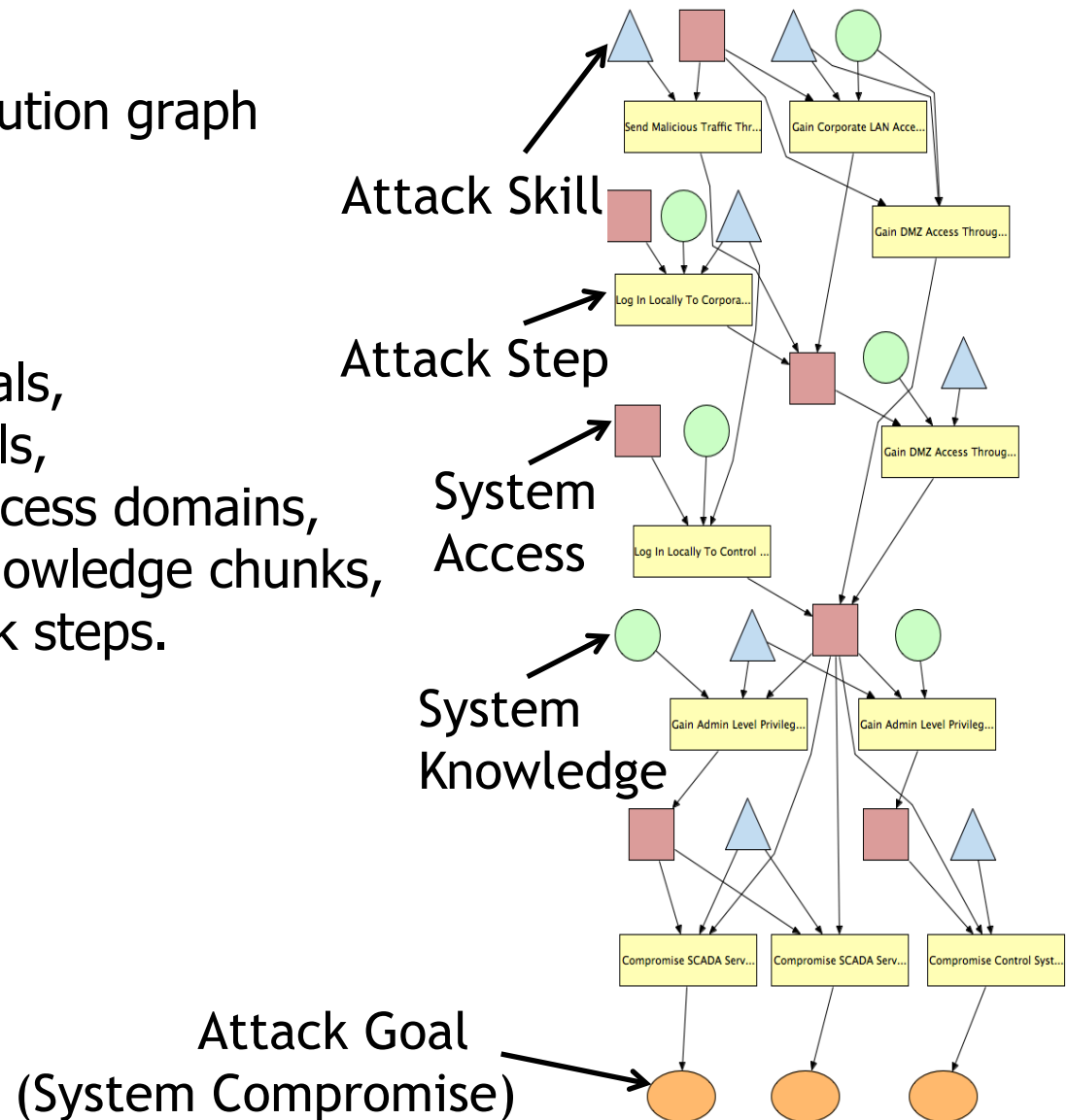
Attack Execution Graph

Formally, an attack execution graph is defined by the tuple

$$\langle G, S, A, K, Q \rangle$$

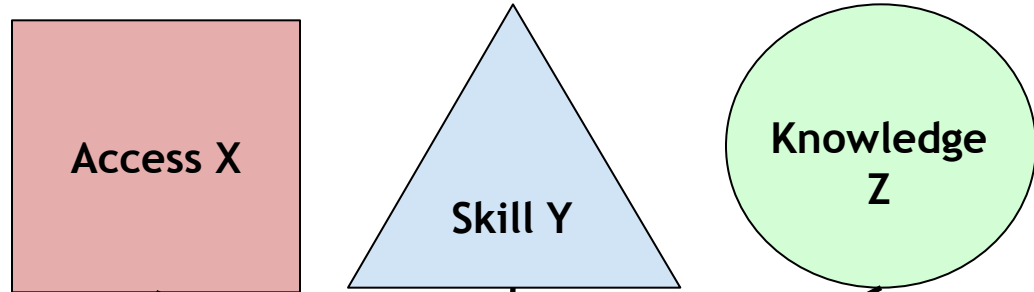
where

G is the set of attack goals,
 S is the set of attack skills,
 A is the set of system access domains,
 K is the set of system knowledge chunks,
and Q is the set of attack steps.

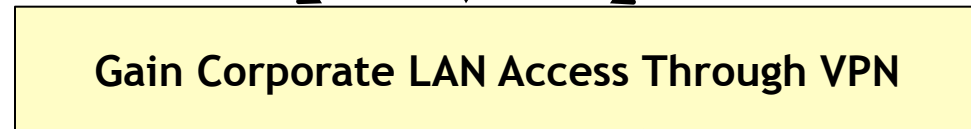


Describing an Attack Step

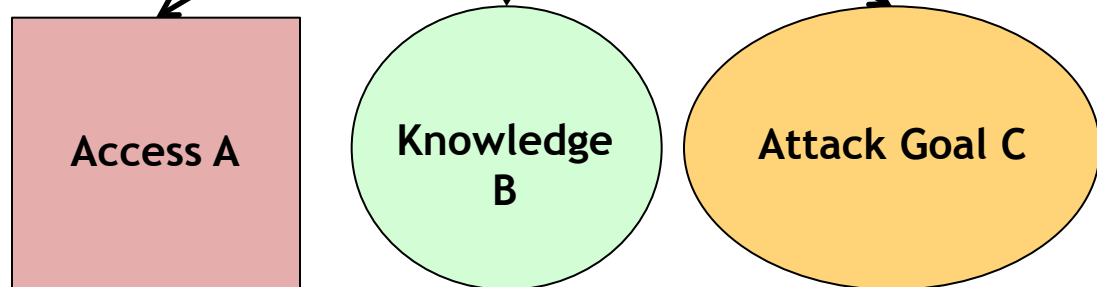
Inputs to
Attack
Precondition



Attack Step



Affected by
Attack Outcomes



Specifying an Attack Step

Category	Definition	Examples
Attack Precondition	Skills, knowledge, and access the adversary must have to attempt this attack step	(firewall traffic exploit skill > 0.6) && (Internet access)
Execution Time Distribution	Time required for the adversary to attempt this attack step	Normal with mean 300 minutes and variance 50 minutes
Cost Distribution	Resource cost to the adversary to attempt this attack step	Deterministic \$6000
Outcome Distribution	If the attack step is attempted, the likelihood of each outcome	P[success] = 0.4 P[partial failure] = 0.2 P[complete failure] = 0.4
Set of Outcomes	The set of all possible outcomes if the attack step is attempted	Success, partial failure, complete failure

Specifying an Attack Step Outcome

Category	Definition	Examples
Detection Probability Distribution	Likelihood of the attack step attempt being detected by the system defense	For success outcome, $P[\text{detection}] = 0.1$ For failure outcome, $P[\text{detection}] = 0.4$
Attack Payoff Distribution	Value to adversary of achieving this attack step outcome	For success outcome, payoff = \$1000 For failure outcome, payoff = -\$600
State Variable Updates	How the state of the model changes due to this attack set outcome	For success outcome, Corporate_LAN_Access = true

Representing an Adversary

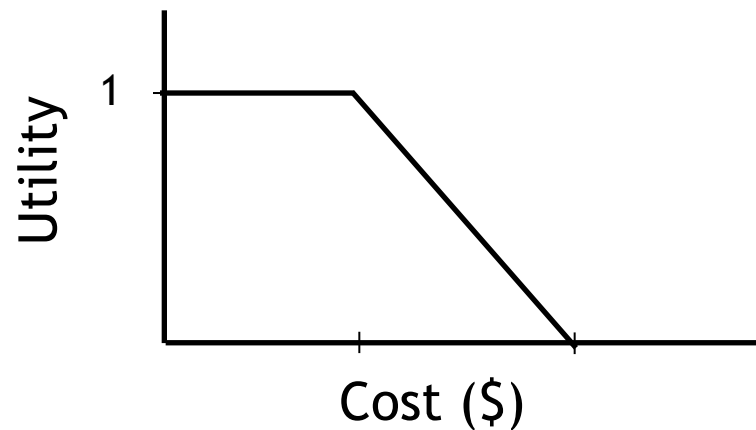
- The security-relevant resources of adversaries are described by the sets of attack skills, system access, and system knowledge that they possess
 - **Attack skills** are exploit skills such as using attack tools, writing malware, finding vulnerabilities, etc.
 - **System access** domains describe the system as a set of domains where getting from one access domain to another requires an attack step, such as logging in to a file system
 - **System knowledge** is the key security information about a system such as user account names and passwords, configuration settings, etc.
- **Attack preferences** describe how heavily adversaries factor risk (detection probability and success probability), payoff, and cost into their attack decisions
- **Attack goals** describe what system compromise(s) the adversaries

Adversary Parameter Values

Category	Definition	Examples	Value Ranges
Attack Preferences	The relative importance of each decision criterion when evaluating the options for the next attack step	preference weights for ... low cost, low probability of detection, high probability of success, high payoff	Between 1 (important) and 0 (not)
Attack Goals	System-specific definitions of system security compromise	compromise control system database integrity, compromise server availability	1 (a goal) or 0 (not a goal)
Attack Skills	General skills or abilities that increase the adversary's probability of success when attempting some attack step	firewall traffic exploit skill, local network log-in exploit skill	Between 1 (proficient) and 0 (not)
Access (dynamic)	Access to network domains within a system; physical access to system components	Internet access, Corporate Network access, Control System Network access	1 (access) or 0 (no access)
Knowledge (dynamic)	Possession of information useful for attacks: user account and passwords, system architecture details	Corporate Firewall VPN user account and password, Control System architecture	1 (known) or 0 (not)

Utility Functions

- Uc_x , Up_x , Ud_x , and Us_x are the utility functions for cost, payoff, detection probability, and success probability, respectively
- Utility functions convert from conventional units to the value to an adversary on a $[0,1]$ scale
- For example,



Process for Producing Metrics

Ask Security Decision Question

Specify Metrics

Describe Adversaries

Describe System

Build an executable stochastic model describing how the adversaries are likely to attack the system

Solve the executable model

Calculate metrics from model results

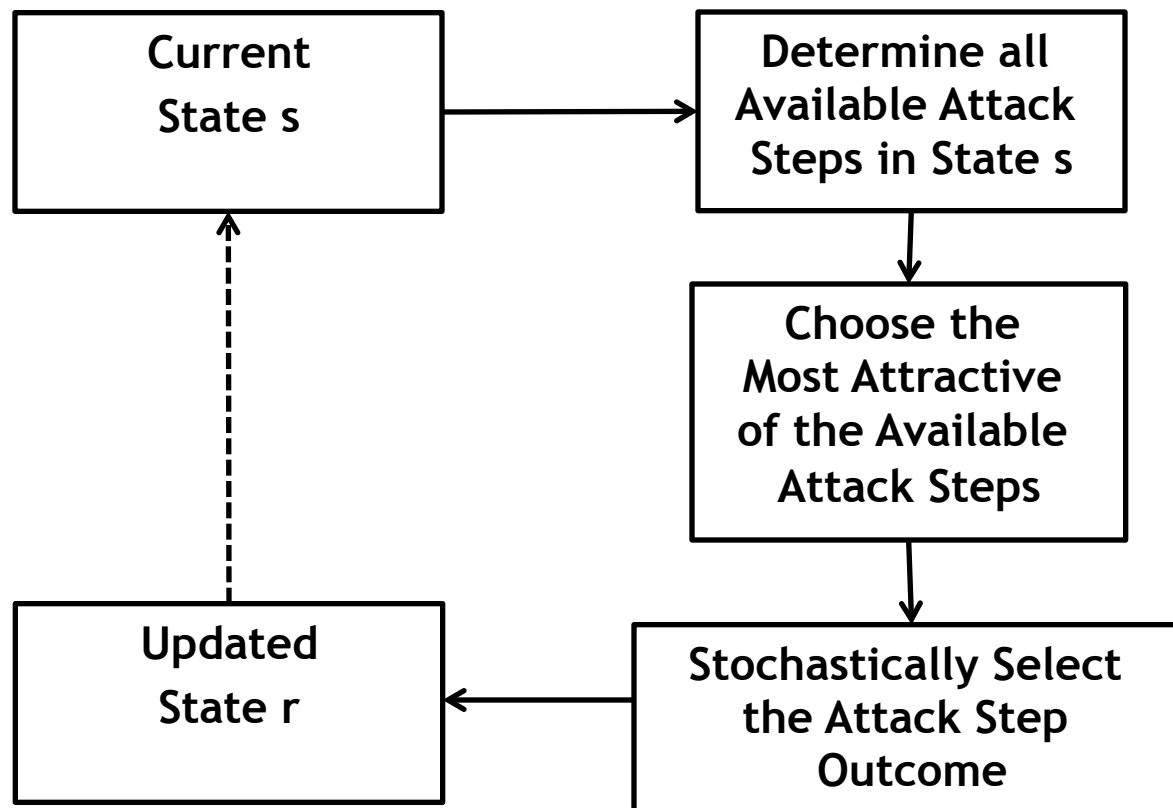
Produce Security Decision Answer

Step 2:

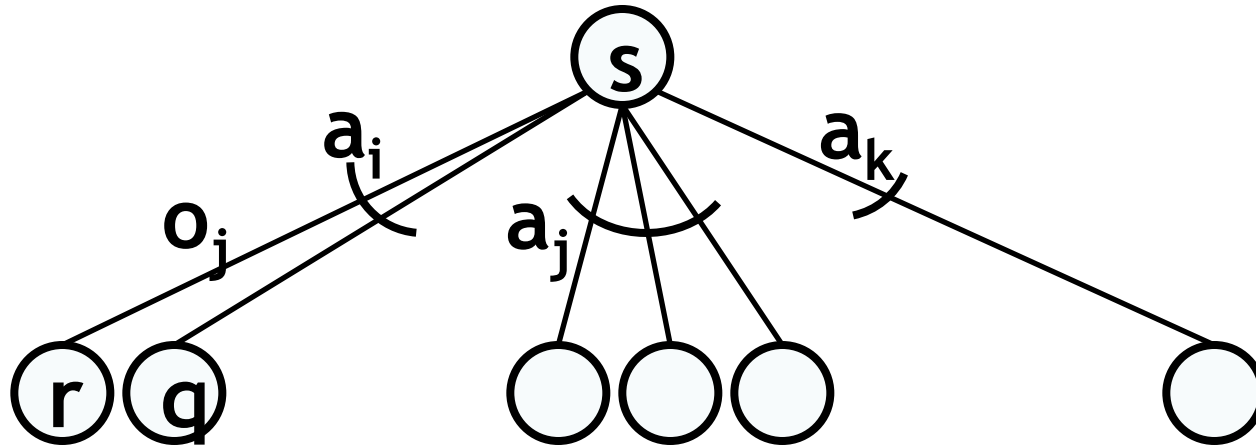


Model Execution: The Attack Decision Cycle

- The adversary selects the most attractive available attack step based on his attack preferences.
- State transitions are determined by the outcome of the attack step chosen by the adversary.



Determine all Available Attack Steps in State s



Here, the three Available Attack Steps in State s are a_i , a_j , and a_k .

Choose the **Most Attractive Available Attack Step** (Planning Horizon = 1)

To rate the attractiveness of each available attack step, the adversary considers

- Properties of the attack step
 - Cost
 - Expected probability of detection
 - Expected payoff
- Attack preference weights of this adversary
(W_{cost} , W_{payoff} , and $W_{detection}$)

Attractiveness =

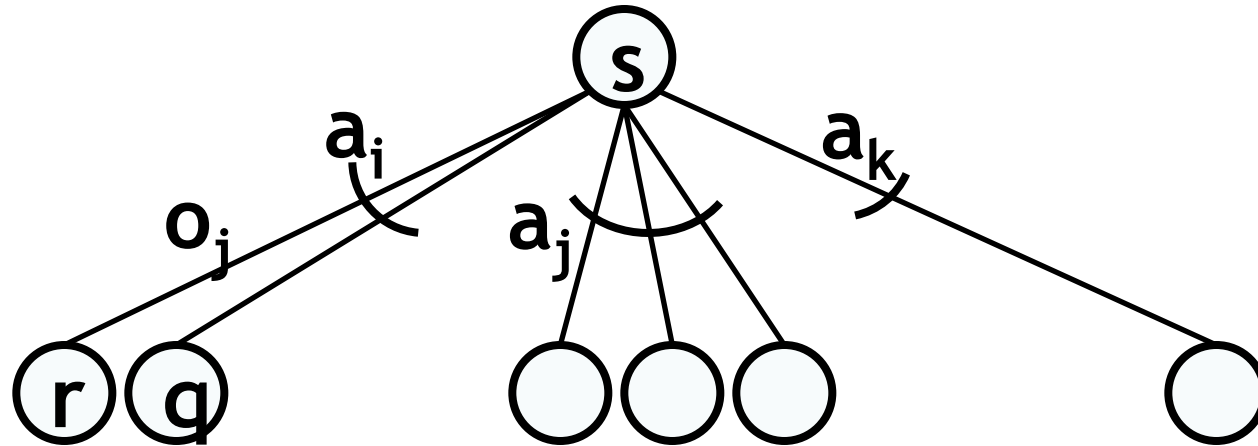
$$\text{Cost} * W_{cost} + E[\text{Payoff}] * W_{payoff} + E[\text{Detection}] * W_{detection}$$

Note that this attractiveness calculation only considers the immediate next attack step and the immediate next states.

An adversary with a planning horizon greater than one can consider future attack steps and future next states.

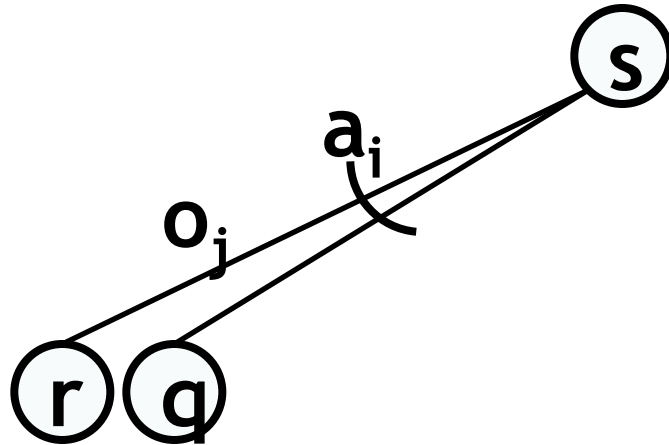


Choose the Most Attractive Available Attack Step (Planning Horizon = 1)



Here, the most attractive Available Attack Step is a_i .

Stochastically Select the Attack Step Outcome

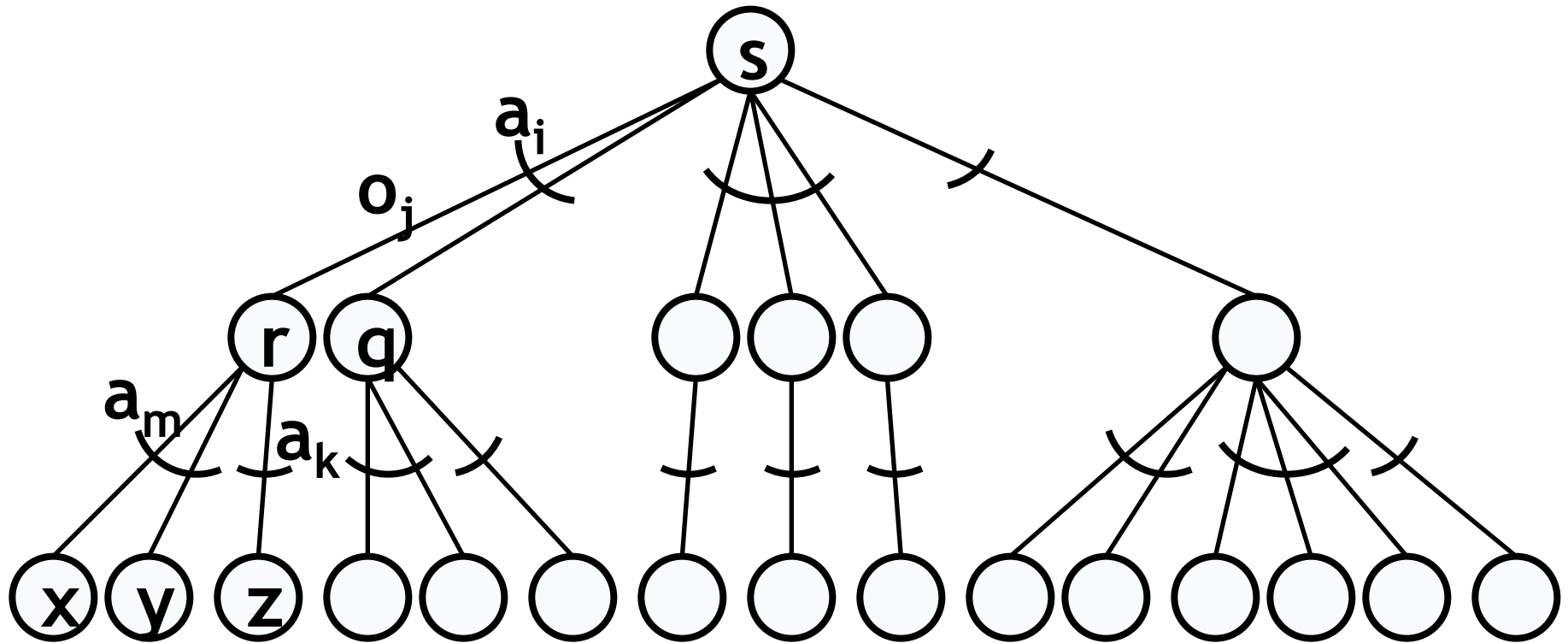


If outcome o_j is stochastically selected, then the next state is state r .

The adversary now repeats the attack decision cycle in state r .

Choose the **Most Attractive Available Attack Step** with a Planning Horizon > 1 using a State Look-Ahead Tree (SLAT)

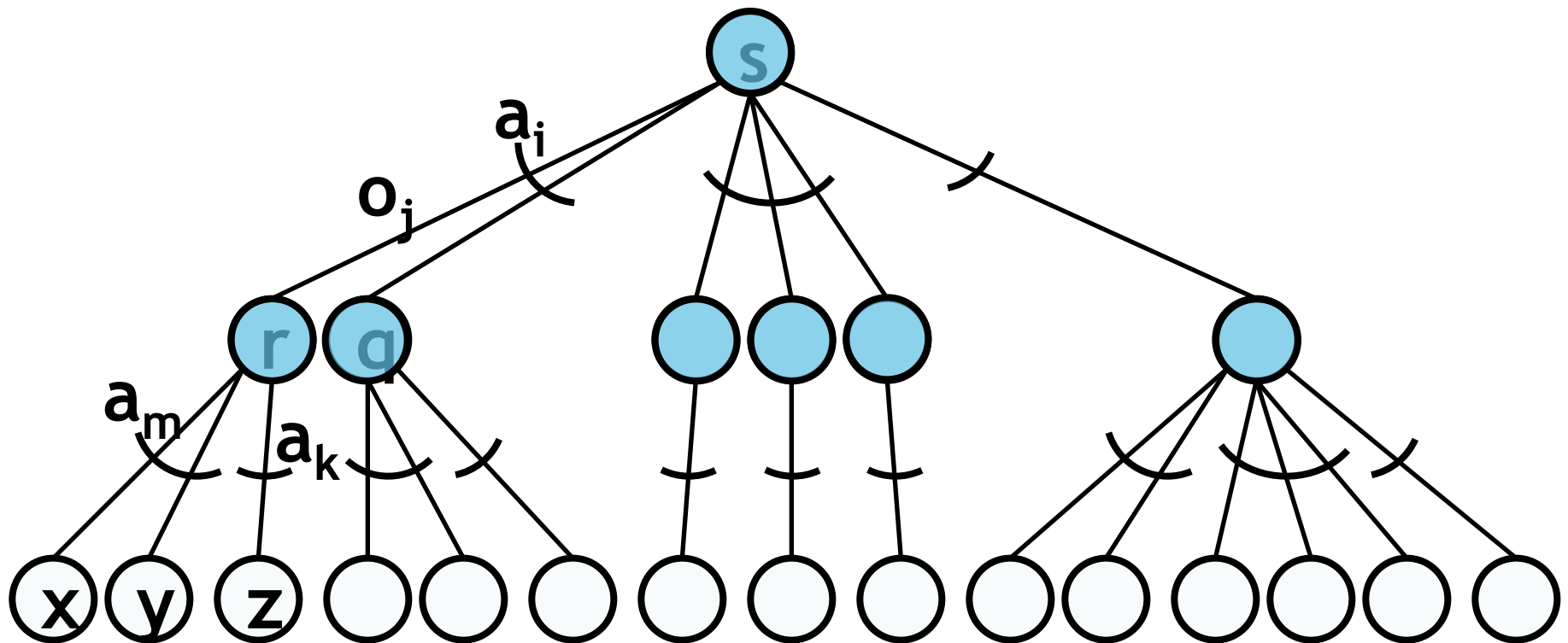
Step 1: Build the SLAT by exploring the available next steps and possible next-states to the depth of the planning horizon.



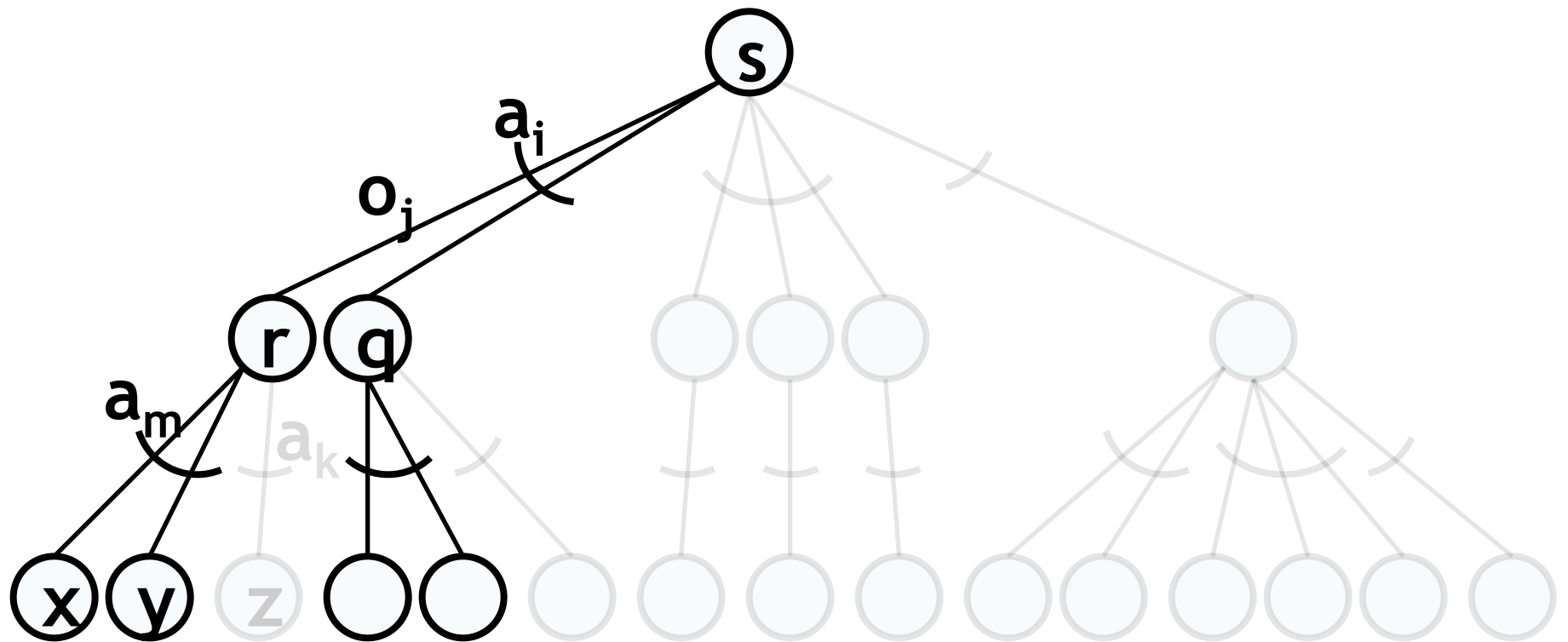
Here, the Planning Horizon is 2.

Choose the **Most Attractive Available Attack Step** with a Planning Horizon > 1 using a **State Look-Ahead Tree (SLAT)**

Step 2: Prune the SLAT by working from the bottom to the top, leaving only the most attractive attack step from each state.



Pruned SLAT Shows the Most Attractive Available Attack Step with a Planning Horizon of 2



Recursive Attractiveness Calculation Algorithm

$$\beta^N(s) \in \{a^* \in A_s | attr^N(a^*, s) = \max_{a_i \in A_s} attr^N(a_i, s)\}.$$

$$attr^N(a_i, s) = w_C \cdot C_i^N(s) + w_P \cdot P_i^N(s) + w_D \cdot D_i^N(s)$$

$$C_i^N(s) = \begin{cases} C_i(s), & \text{when } N = 1 \\ C_i(s) + \sum_{o \in O_i} (C_*^{N-1}(r) \cdot Pr_i(s, o)), & \text{when } N > 1, \end{cases}$$

$$P_i^N(s) = \begin{cases} \sum_{o \in O_i} (P(E_i(s, o)) \cdot Pr_i(s, o)), & \text{when } N = 1 \\ \sum_{o \in O_i} (P_*^{N-1}(r) \cdot Pr_i(s, o)), & \text{when } N > 1. \end{cases}$$

$$D_i^N(s) = \begin{cases} \sum_{o \in O_i} (D_i(s, o) \cdot Pr_i(s, o)), & \text{when } N = 1 \\ \sum_{o \in O_i} ((1 - (1 - D_i(s, o)) \cdot (1 - D_*^{N-1}(r))) \cdot Pr_i(s, o)), & \text{when } N > 1. \end{cases}$$

ADVISE Simulation Algorithm

Algorithm 1 ADVISE Model Simulation

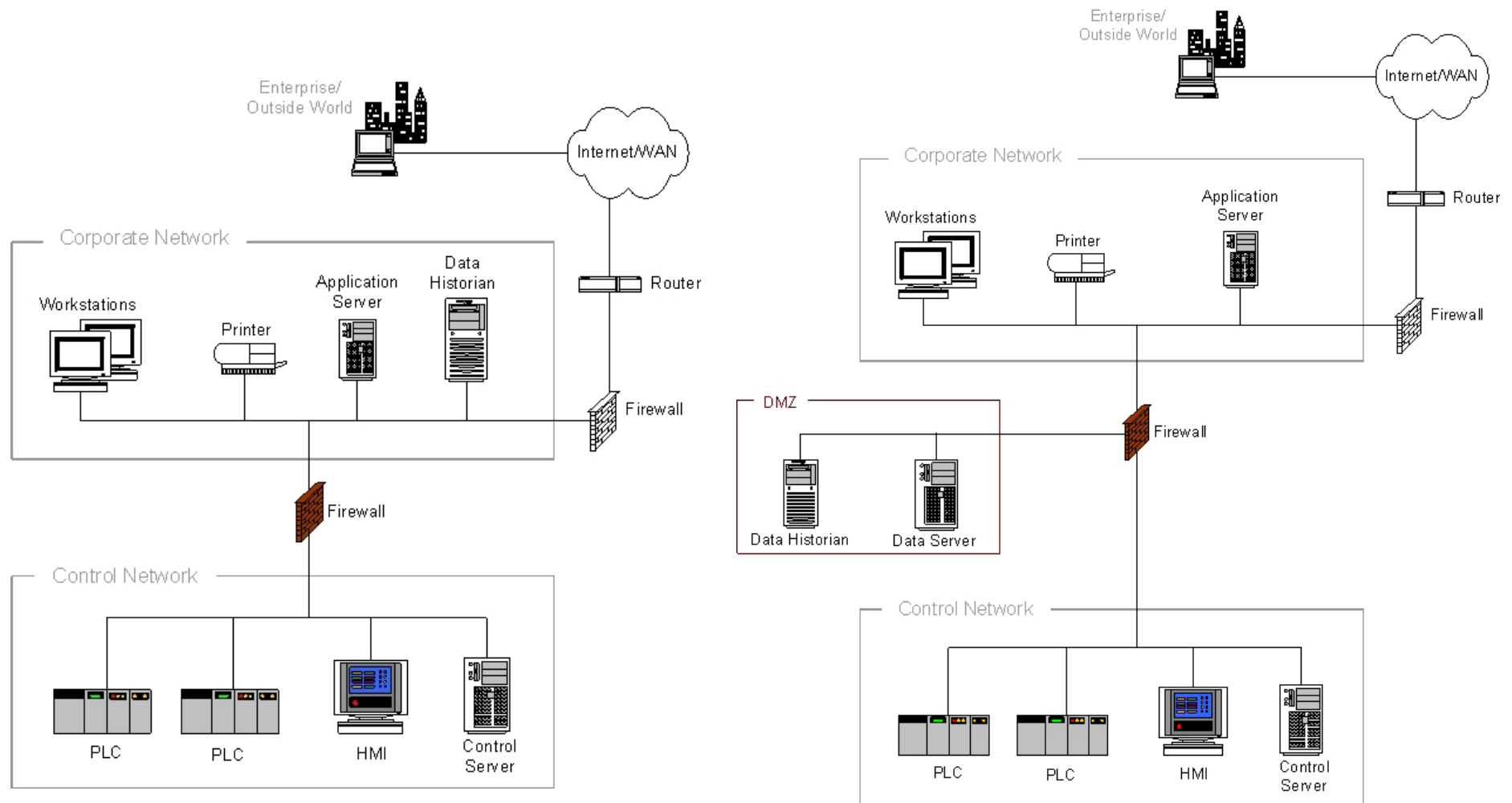
- 1: Time $\leftarrow 0$
 - 2: State $\leftarrow s_0$
 - 3: **while** Time $< \tau$ **do**
 - 4: Attack $_i \leftarrow \beta^N(\text{State})$
 - 5: Outcome $\leftarrow o$, where $o \sim \text{Prob}_i(\text{State})$
 - 6: Time $\leftarrow \text{Time} + t$, where $t \sim T_i(\text{State})$
 - 7: State $\leftarrow E_i(\text{State}, \text{Outcome})$
 - 8: **end while**
-

Calculating Metrics from Model Results

- Question
 - Is the corporate network security budget better spent on upgrading the local physical security or the remote access network security?
- Metric
 - Probability of confidential corporate data compromise in one year
- System configurations to compare
 - Baseline: current system configuration
 - Physical security upgrade
 - Network security upgrade
- Solution by simulation using execution algorithm described earlier.

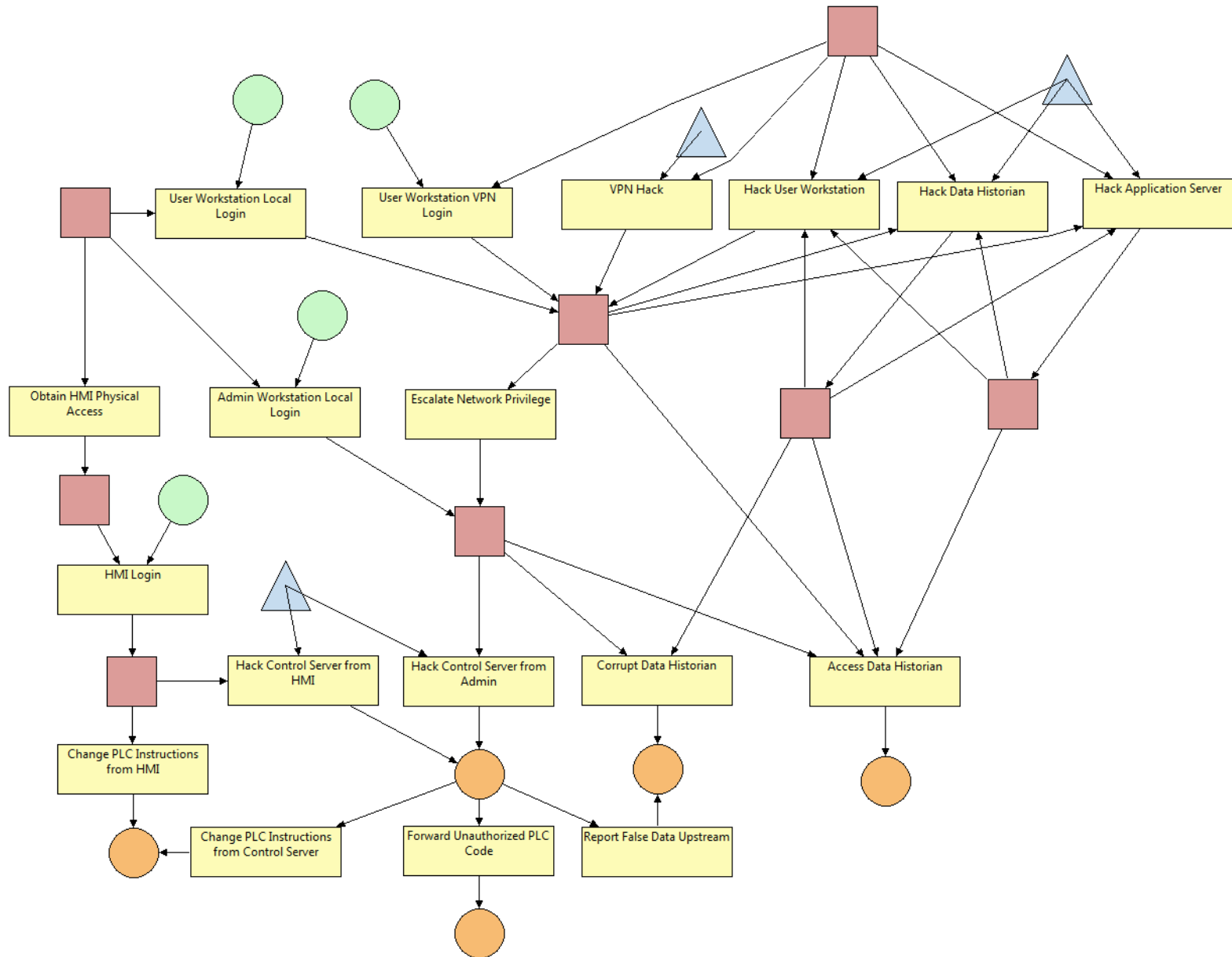


Example: Generic SCADA System Architectures



Figures 5-1 and 5-3 of the National Institute of Standards and Technology Guide to Industrial Control Systems Security (NIST SP 800-82)

Attack Execution Graph for Non-DMZ SCADA System



Adversary Models

Code Name: NationState

Decision Parameters

Planning Horizon: 4

Attack Preference Weights

Cost: cw
 Detection: dw
 Payoff: pw

Future Discount Factors

Cost: 1.0
 Detection: 1.0
 Payoff: 1.0

Skills

Name	Code Name	Proficiency
▲ Hack Skill	HackSkill	hs
▲ VPN Hack Skill	VPNHackSkill	vhs

Initial Access

Name	Code Name
■ Internet Access	InternetAccess

Initial Knowledge

Name	Code Name
------	-----------

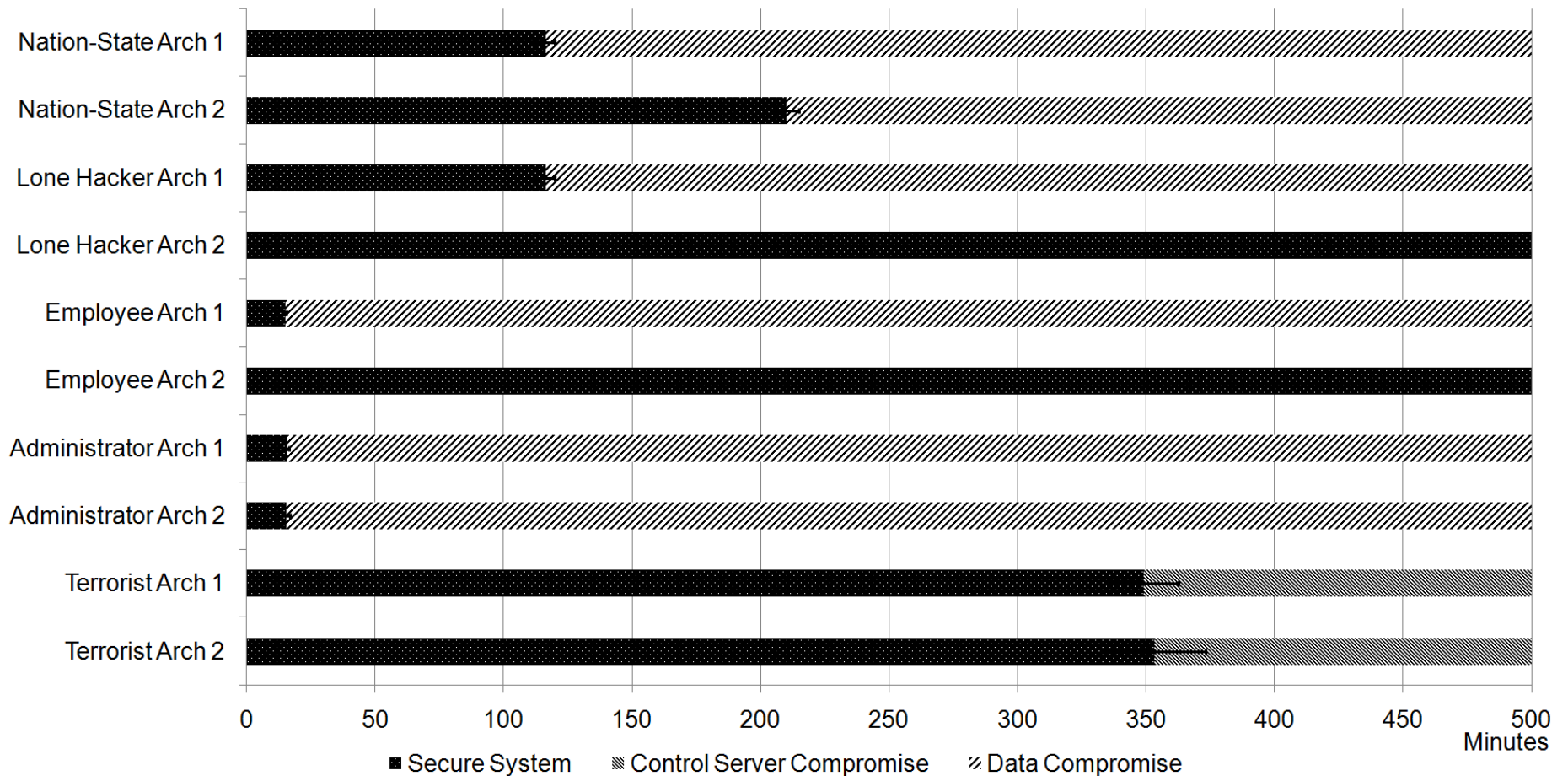
Goals

Name	Code Name	Payoff
● Access Data	AccessData	400
● Run Unauthorized PLC Code	RunUnauthorizedPLCCode	100
● Corrupt Data	CorruptData	200
● Run Authorized PLC Code	RunAuthorizedPLCCode	300
● Run Unauthorized Control Server	RunUnauthorizedControlServerCode	500

ATTACK PREFERENCE WEIGHTS FOR FOUR ADVERSARIES

<i>Adversary</i>	<i>Cost</i>	<i>Payoff</i>	<i>Detection</i>
Nation-State	0.01	0.40	0.59
Lone Hacker	0.20	0.40	0.40
Terrorist Organization	0.05	0.80	0.15
Disgruntled Employee	0.40	0.50	0.10

Model Results



For each adversary and architecture combination (Arch 1 = non-DMZ; Arch 2 = DMZ), this graph shows the average time during [0,500 min] that the system is in a secure state.



Insights from the Model Results

- The DMZ SCADA architecture offers better protection than the non-DMZ architecture against data compromise by a nation-state, a lone hacker, or a standard employee.
- However, an administrator employee remains undaunted by the addition of a DMZ.
- Also, the DMZ does not impact the ability of the terrorist to compromise the control server in the control network.
- In summary, adding one security defense mechanism does not protect against all types of adversaries and all types of compromise.

Summary of Metrics Process

Ask Security Decision Question

Specify Metrics

Describe Adversaries

Describe System

Build an executable stochastic model describing how the adversaries are likely to attack the system

Solve the executable model

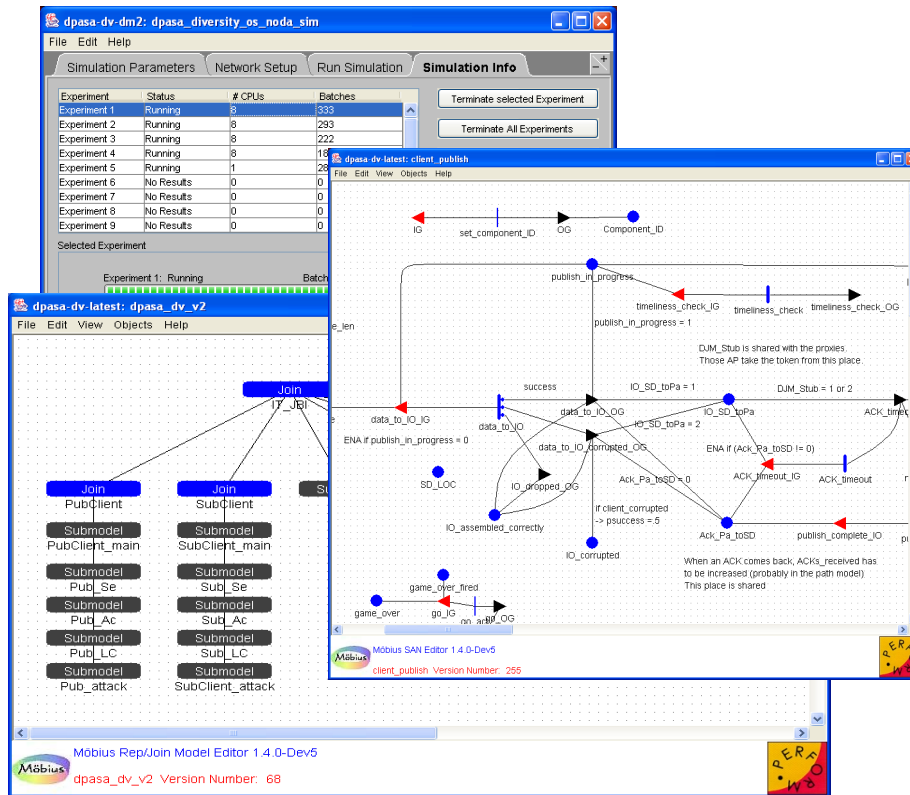
Calculate metrics from model results

Produce Security Decision Answer

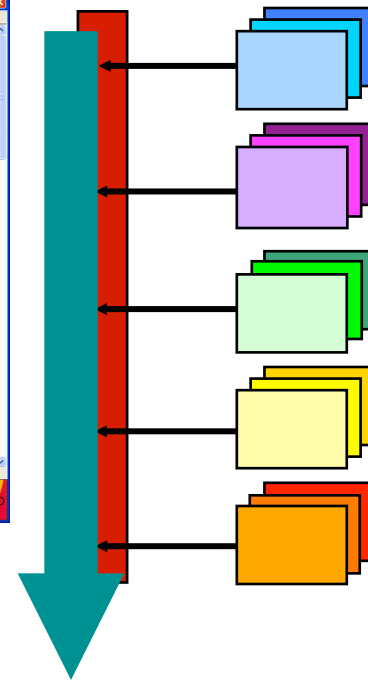
Tool Development

- Build on existing Mobius modeling toolset:
 - Enhance Mobius modeling tool to support Adversary and Attach Execution Graph Models as Atomic Models
 - Metrics results generated by discrete event simulator and analytic solvers in Mobius
 - Link to Mobius analytic solvers to provide proofs of certain (non-stochastic) properties

Möbius: Model-Based Evaluation of System Dependability, Security, and Performance



Framework Component



Atomic Model

Composed Model

Solvable Model

Connected Model

Study Specifier
(generates multiple models)

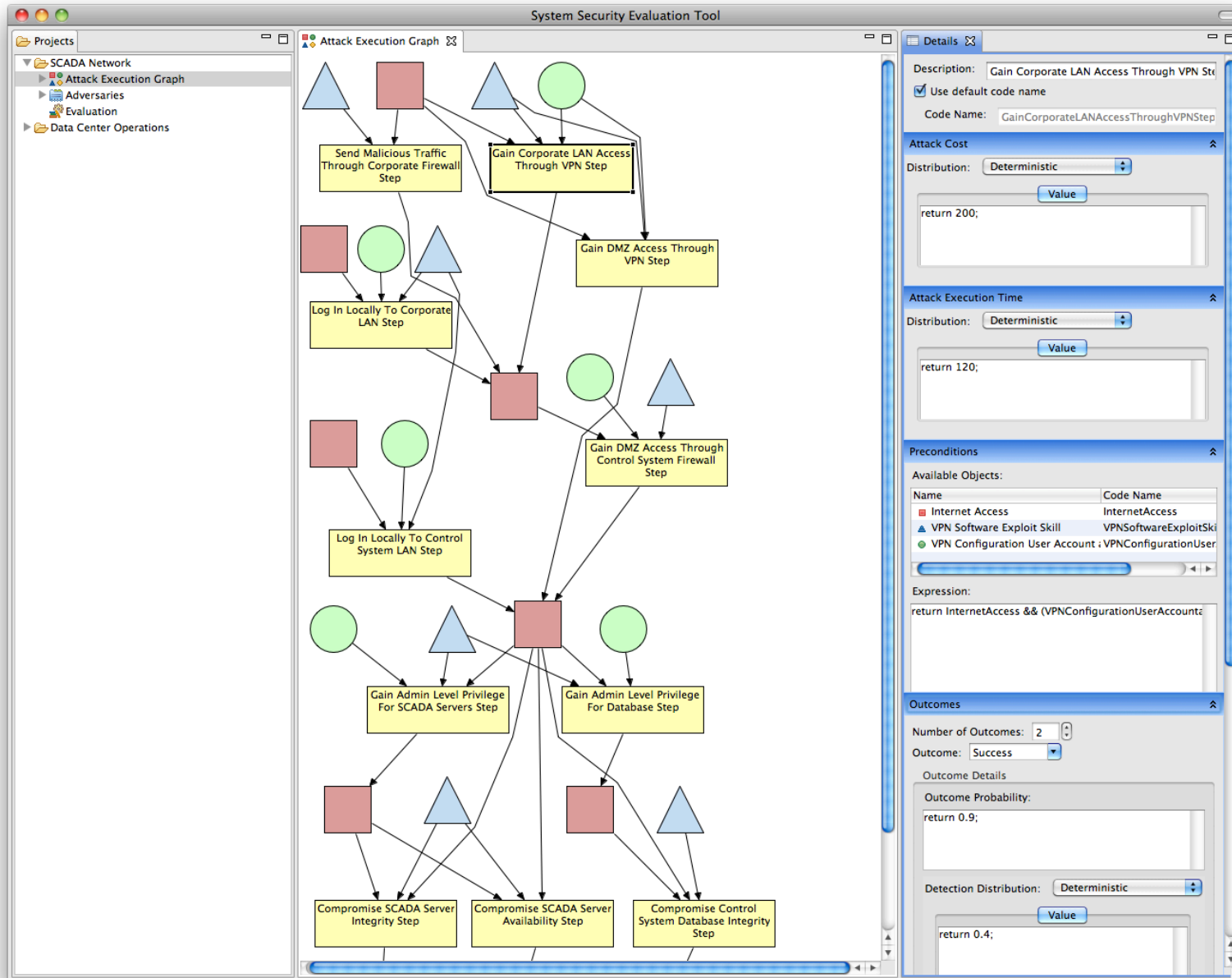
Use:

- 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

Attack Execution Graph Atomic Model Editor



Adversary Atomic Model Editor

The screenshot displays the 'System Security Evaluation Tool' interface. The main window is titled 'Lone Hacker' and contains the following sections:

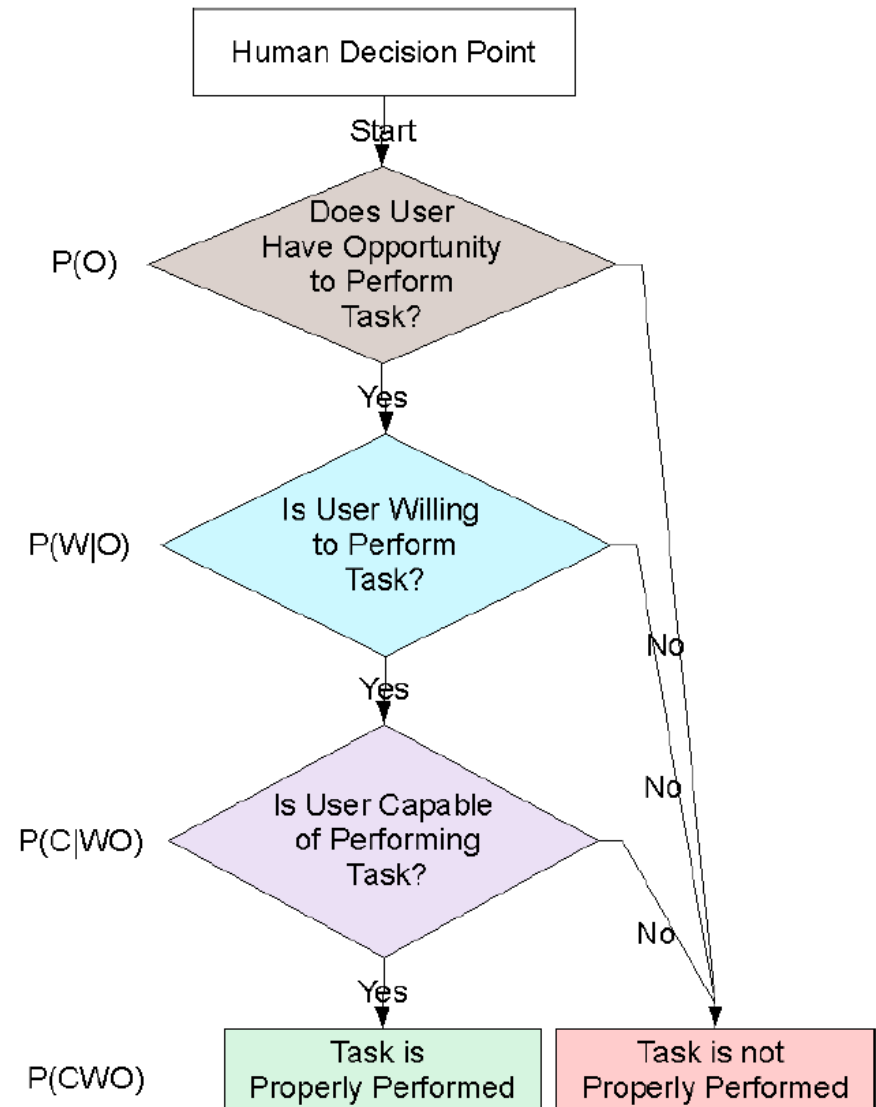
- Name:** Lone Hacker
- Use default code name:
- Code Name:** LoneHacker
- Attack Preference Weights:**
 - Detection: 0
 - Success: 0
 - Cost: 0
 - Payoff: 0
- Skills:**

Name	Code Name	Proficiency
Database Exploit Skill	DatabaseExploitSkill	0.700
SCADA Server Exploit Skill	SCADAServerExploitSkill	0.200
- Initial Access:**
 - Insider Physical Access to Control System LAN
 - SCADA Server Access (with Write Access)
 - Insider Physical Access to Corporate LAN
 - Internet Access
 - Control System LAN Access (User Level)
 - Control System Database Access (with Write Access)
- Initial Knowledge:**
 - Control System Authentication Credentials Knowledge
 - Admin Authentication Details for Database Knowledge
 - VPN Configuration User Account and Password Knowledge
- Goals:**
 - Control System Database Integrity Compromise Goal

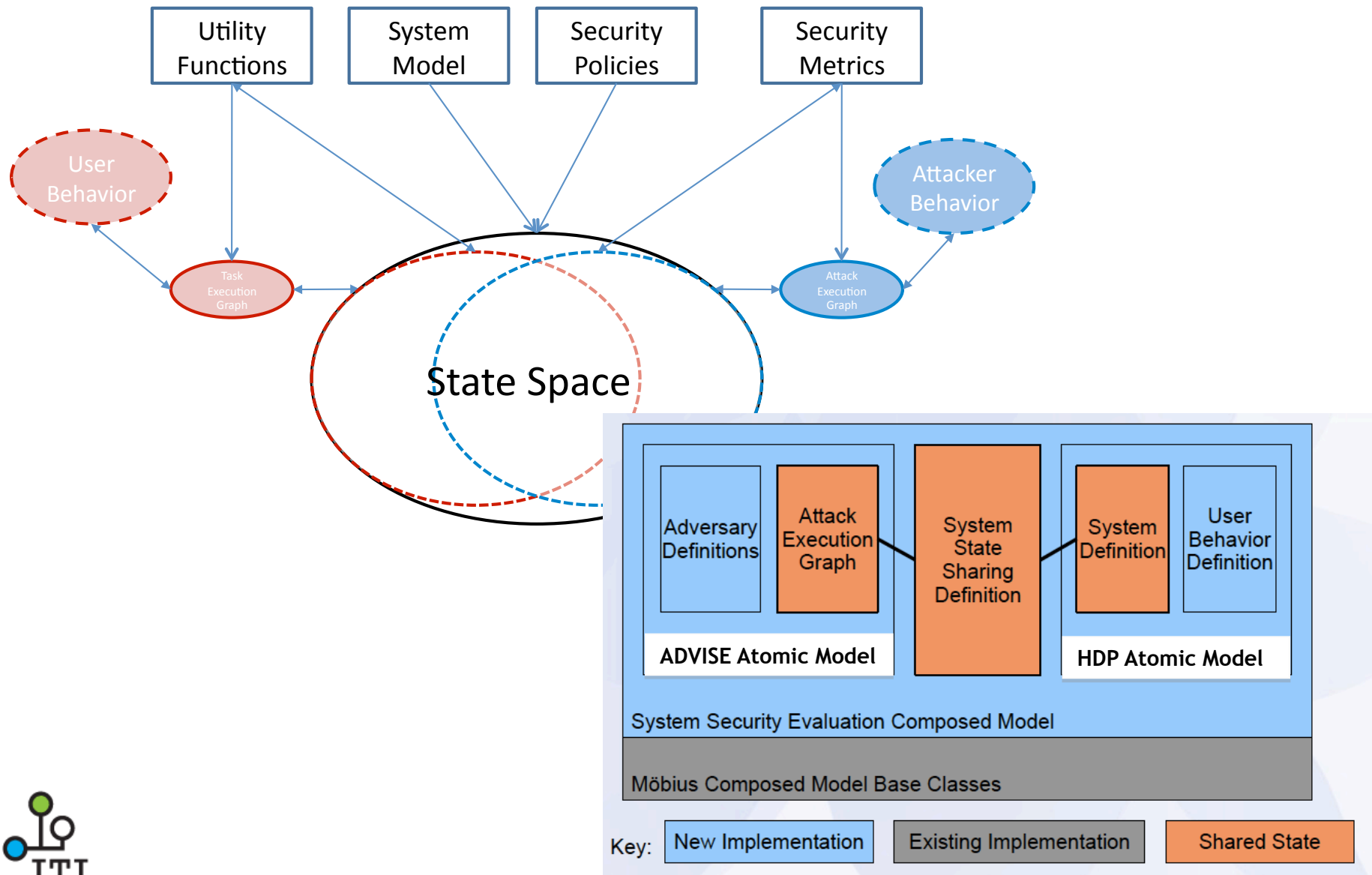
New Work: Model Impact of HDPs on System Security

Model “Good” humans, as well as attackers in the system

- *Users can significantly effect system security*
- *Example: More stringent security measures can lead to behaviors that compromise security*
- *Solution: Model user behavior using Human Decision Points*
- Human Decision Point (HDP) - Task in which Human decision is “deemed” significant
- Understand: Opportunity, Willingness, and Capability



Putting it all together: Understanding User-Attacker Behavior Tradeoffs



Conclusions

- Since system security cannot be absolute, quantifiable security metrics are needed
- Metrics are useful even if not perfect; e.g., relative metrics can aid in critical design decisions
- **New formalisms/modeling approaches are needed**
 - Rich enough to model adversary, user, and system behavior
 - Natural for security analysts
 - Semantically precise
- Work in these directions is ongoing (see other work in RACI and MetriSec workshops), but much more needs to be done