Making Sound Design Decisions Using Quantitative Security Metrics

Bill Sanders

University of Illinois at Urbana-Champaign

January 6, 2012

ADVISE Team

University of Illinois Urbana-Champaign

Mike Ford

Ken Keefe

Elizabeth LeMay

Bill Sanders

Cyber Defense Agency, Inc.

Carol Muehrcke

Research sponsored by Doug Maughan at Science and Technology Directorate, Department of Homeland Security

The Problem: Assessing Security and Resilience

- 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
  - "Resilient" systems aim to meet their ongoing 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*?

Why use model-based system-level security and resiliency evaluation?

- Gain a big-picture system security perspective
  - How component-level insecurities impact overall system security
  - How individual attack actions threaten overall system security
- Improve security design and investment decisions
  - Compare system configuration alternatives before implementing them
  - Estimate how well the system will function (withstand attacks and accomplish its mission) in a particular threat environment

Contrasting Approaches

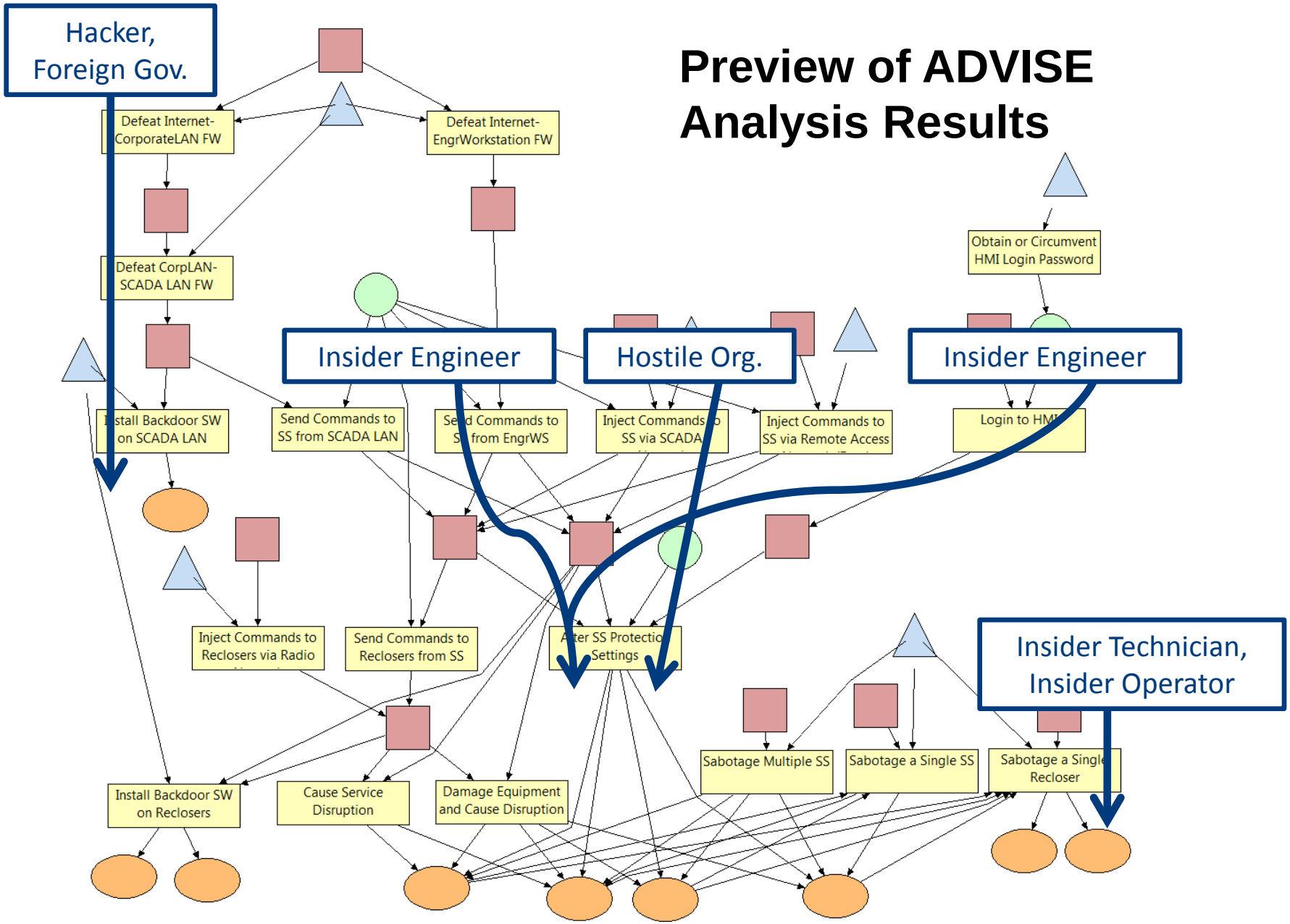
Typical Situation Today:

- Process:
  - Rely on a trusted analyst (wizard?) 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

Preview of ADVISE Analysis Results



Related Work Motivating ADVISE

- Model-based security analysis
  - Attack Trees
  - Attack Graphs and Privilege Graphs
- Adversary-based security analysis
  - MORDA (Mission-Oriented Risk and Design Analysis)
  - NRAT (Network Risk Assessment Tool)

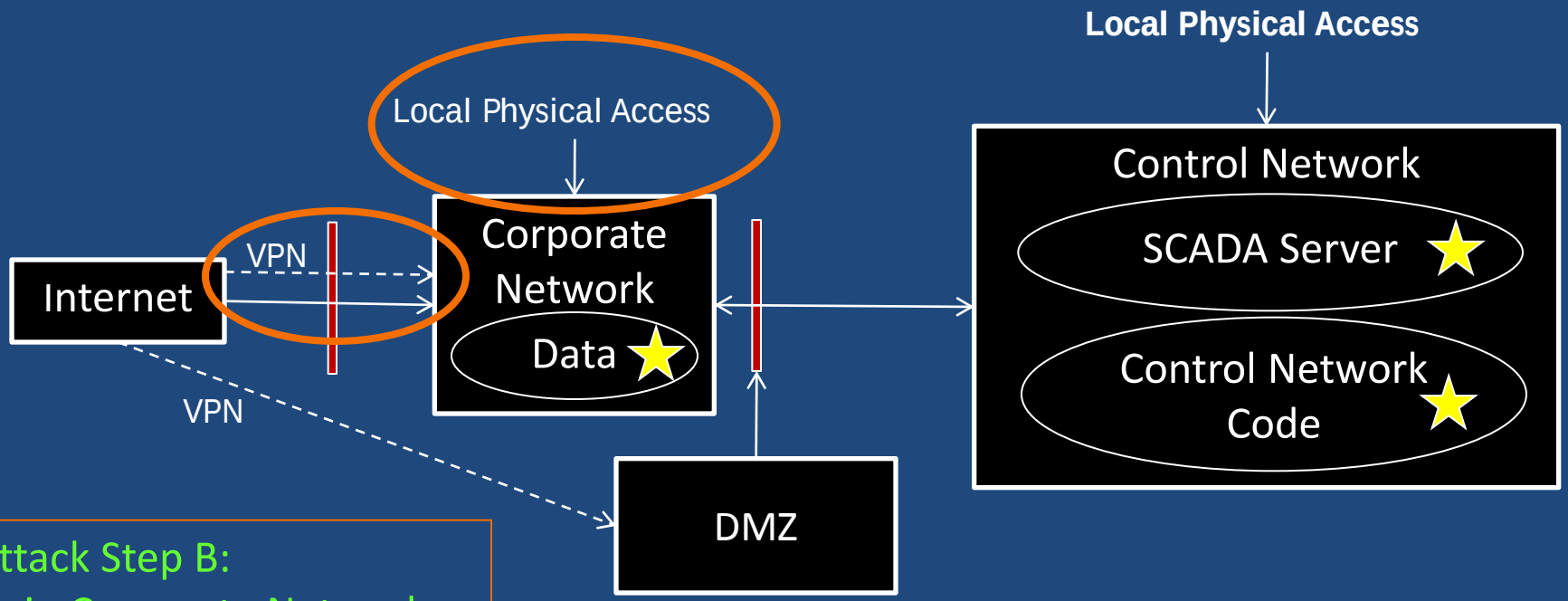
ADVISE integrates the benefits of both model-based and adversary-based security analysis

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

Example: SCADA System Attack

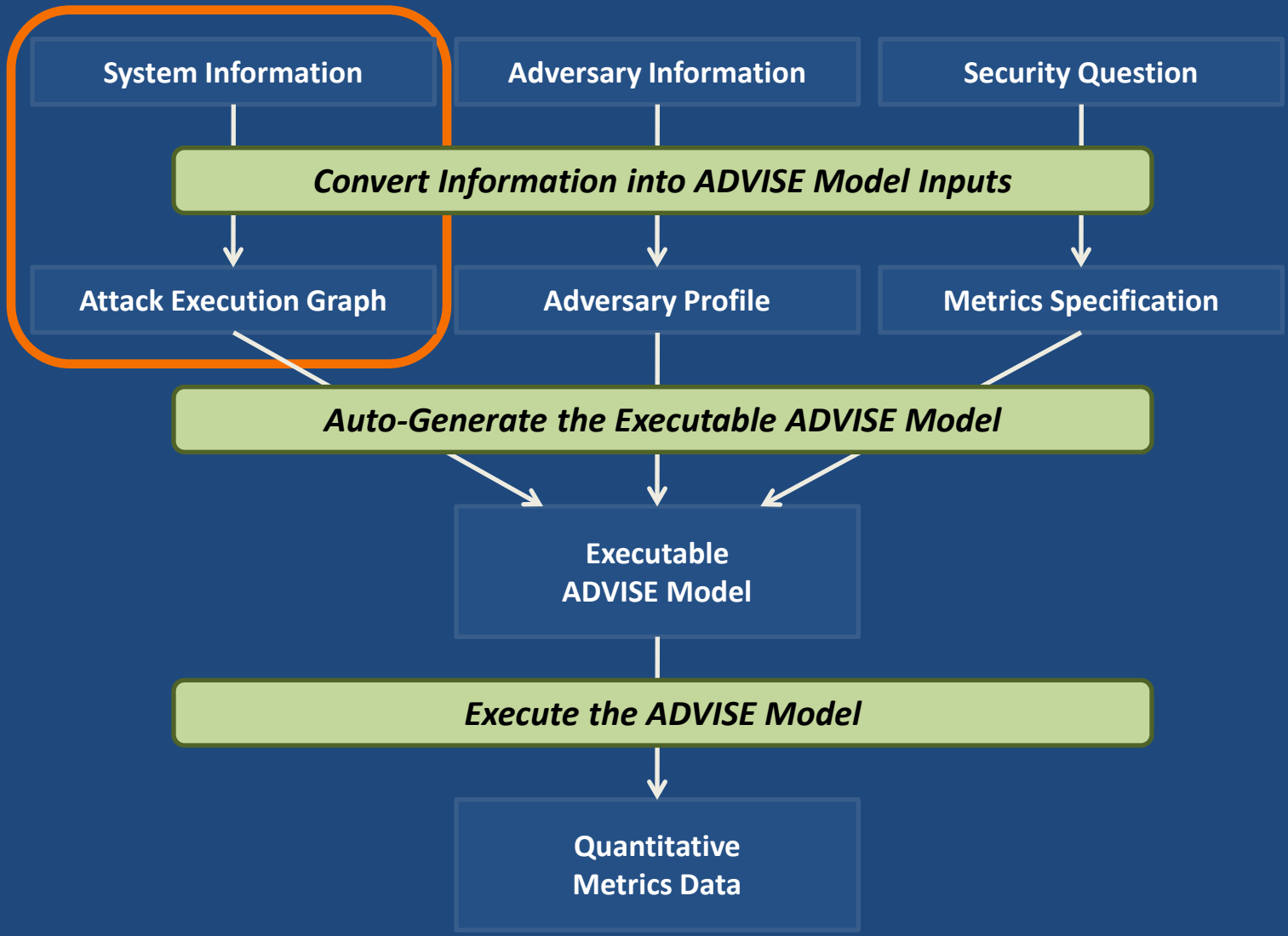
Attack Step A:
Gain Corporate Network Access
Through Local Physical Access



Attack Step B:
Gain Corporate Network
Access Through VPN

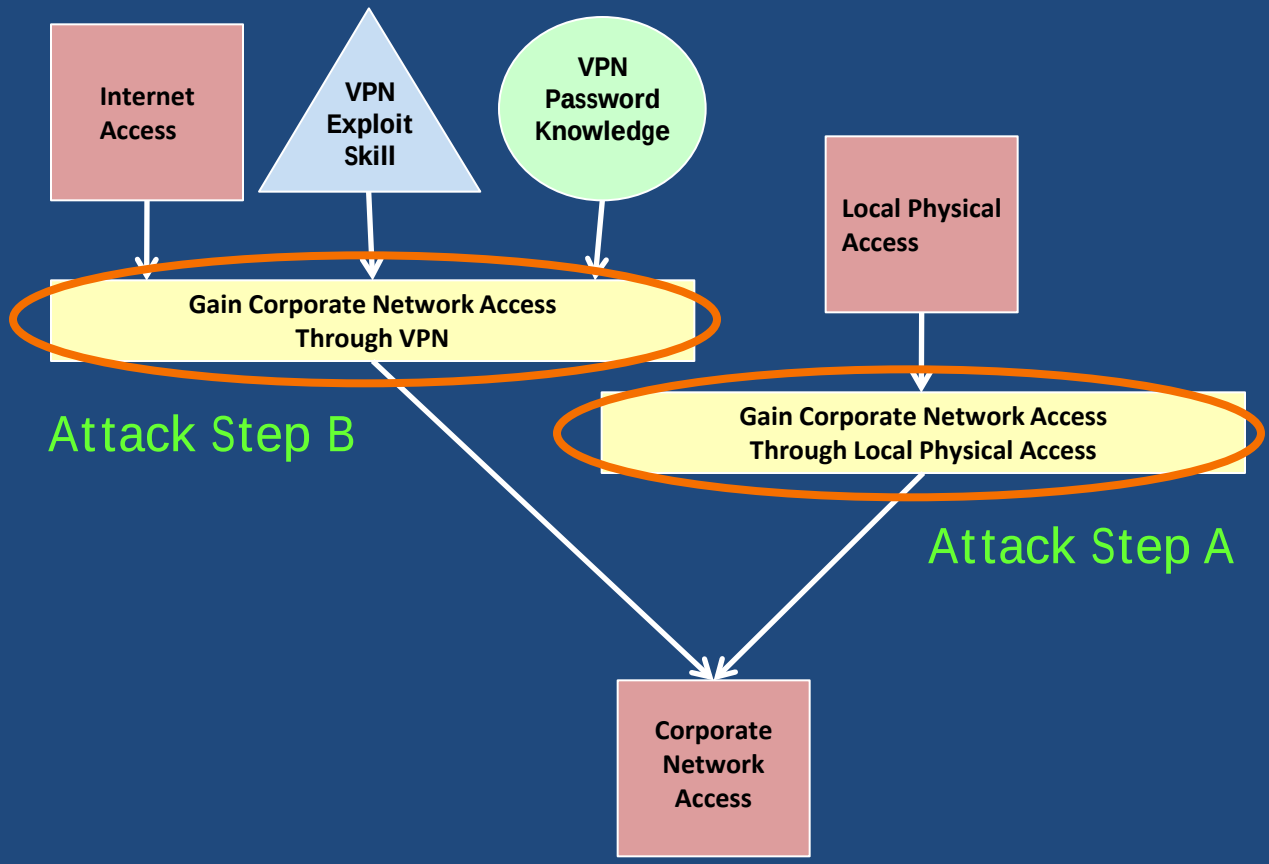
★ = Attack Target

ADVISE Method Overview



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

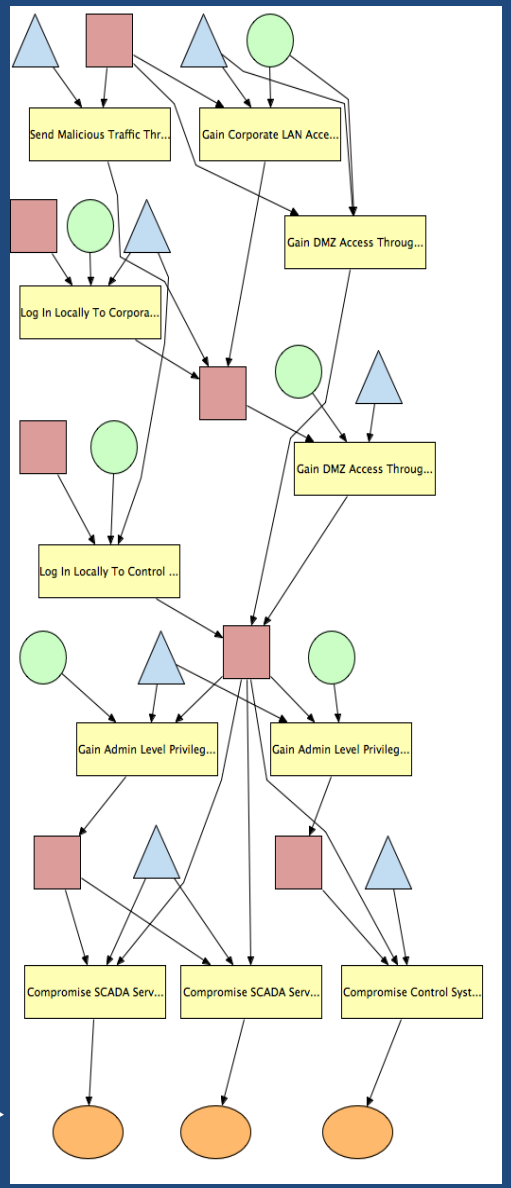
A is the set of attack steps,
e.g., “Access the network using the VPN,”

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.”



Attack Step Definition

An attack step a_i is a tuple:

$a_i = \langle B_i, T_i, C_i, O_i, Pr_i, D_i, E_i \rangle$

Note: X is the set of all states in the model.

B_i: $X \rightarrow \{True, False\}$ is a Boolean precondition,
e.g., (Internet Access) AND ((VPN account info) OR (VPN exploit skill)).

T_i: $X \times R^+ \rightarrow [0, 1]$ is the time to attempt the attack step,
e.g., 5 hours.

C_i: $X \rightarrow R^{\geq 0}$ is the cost of attempting the attack step, e.g., $1000.

O_i is a finite set of outcomes, e.g., {Success, Failure}.

Pr_i: $X \times Oi \rightarrow [0, 1]$ is the probability of outcome $o \in O_i$ occurring,
e.g., if (VPN exploit skill > 0.8) {0.9, 0.1} else {0.5, 0.5}.

D_i: $X \times Oi \rightarrow [0, 1]$ is the probability of the attack being detected when outcome $o \in O_i$ occurs, e.g., {0.01, 0.2}.

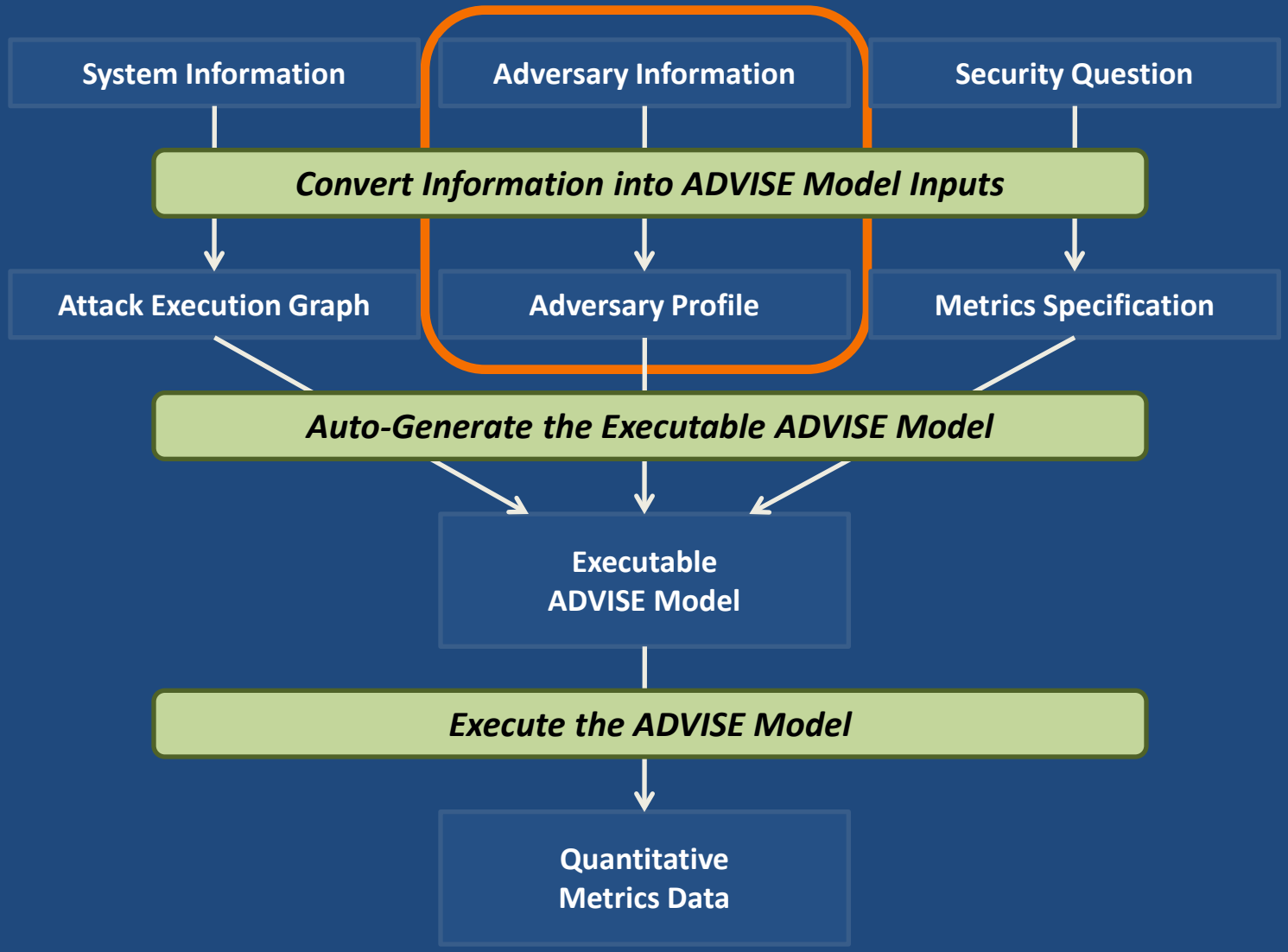
E_i: $X \times Oi \rightarrow X$ is the next-state that results when outcome $o \in O_i$ occurs,
e.g., {gain Network Access, no effect}.

The “Do-Nothing” Attack Step

- Contained in every attack execution graph
- Represents the option of an adversary to refrain from attempting any active attack
  - The precondition $B_{DoNothing}$ is always true.
- For most attack execution graphs,
  - the cost $C_{DoNothing}$ is zero,
  - the detection probability $D_{DoNothing}$ is zero, and
  - the next-state is the same as the current state.

- The existence of the “do-nothing” attack step means that, regardless of the model state, there is always at least one attack step in the attack execution graph whose precondition is satisfied

ADVISE Method Overview



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.

Model State

The model state, $s \in X$, reflects the progress of the adversary in attacking the system and is defined by the tuple

$$s = \langle R_s, K_s, G_s \rangle$$

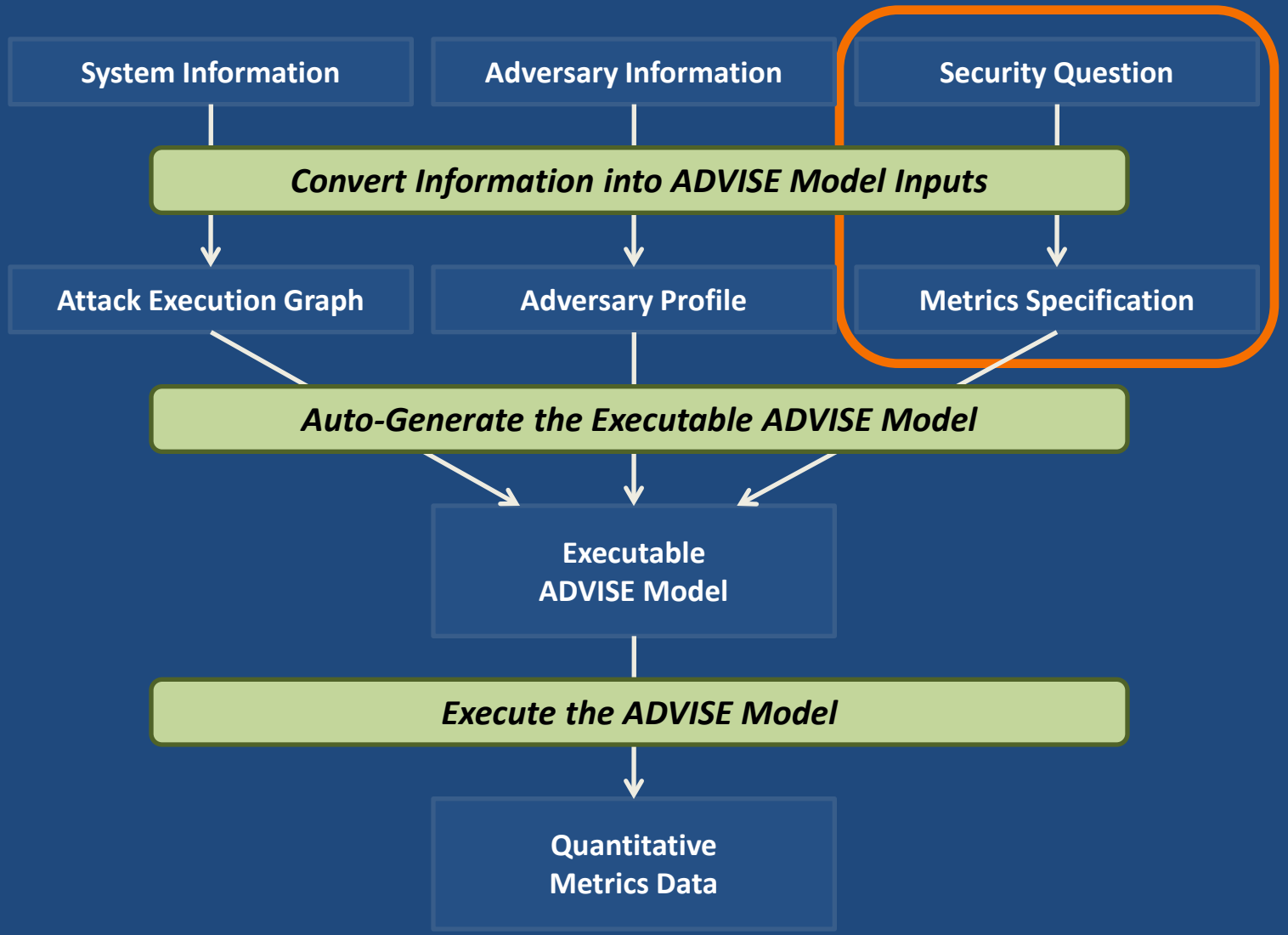
where

$R_s \in R$ is the set of access domains that the adversary can access,

$K_s \in K$ is the set of knowledge items that the adversary possesses, and

$G_s \in G$ is the set of attack goals the adversary has achieved.

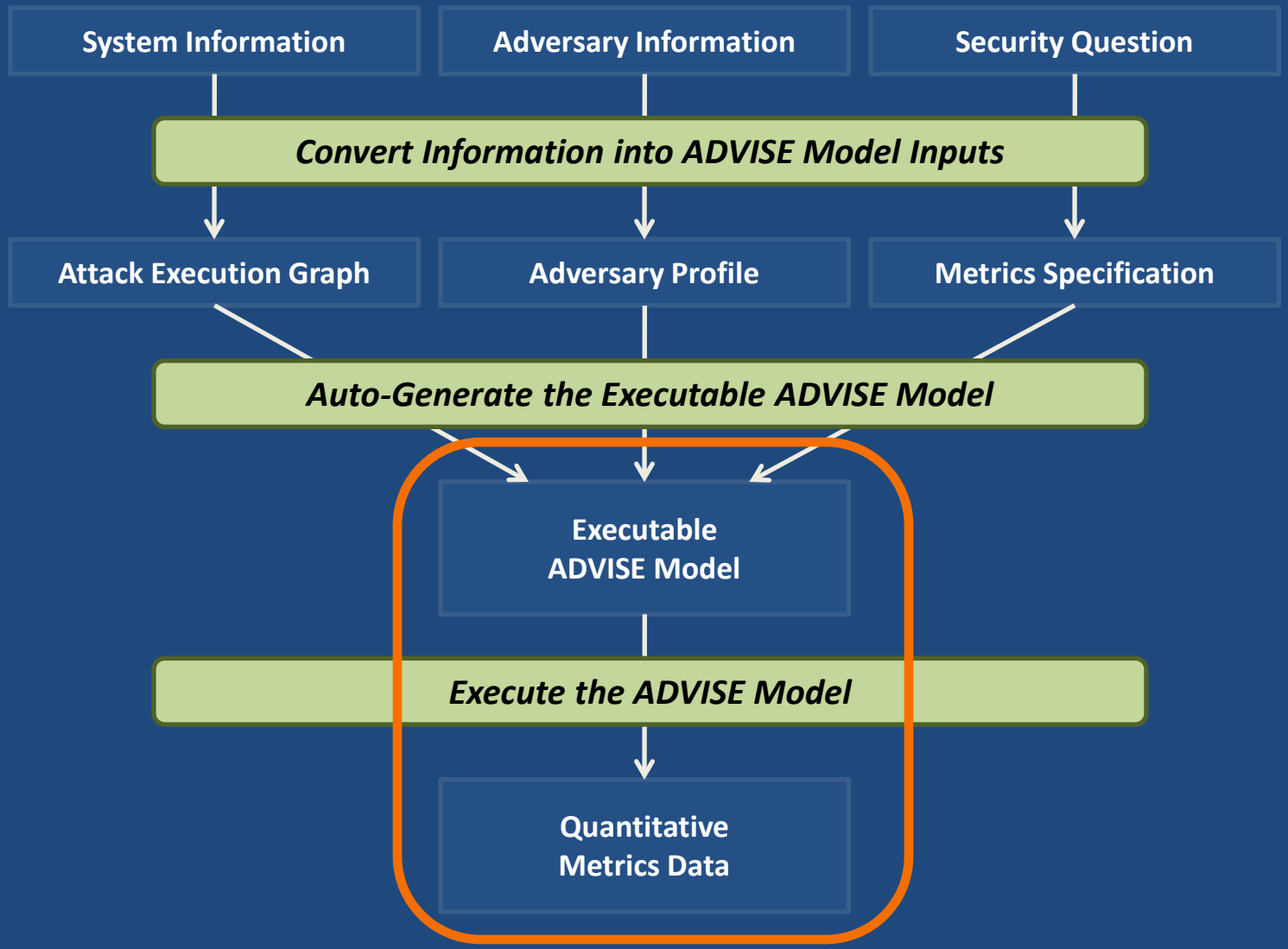
ADVISE Method Overview



ADVISE Security Question: Metrics Specification

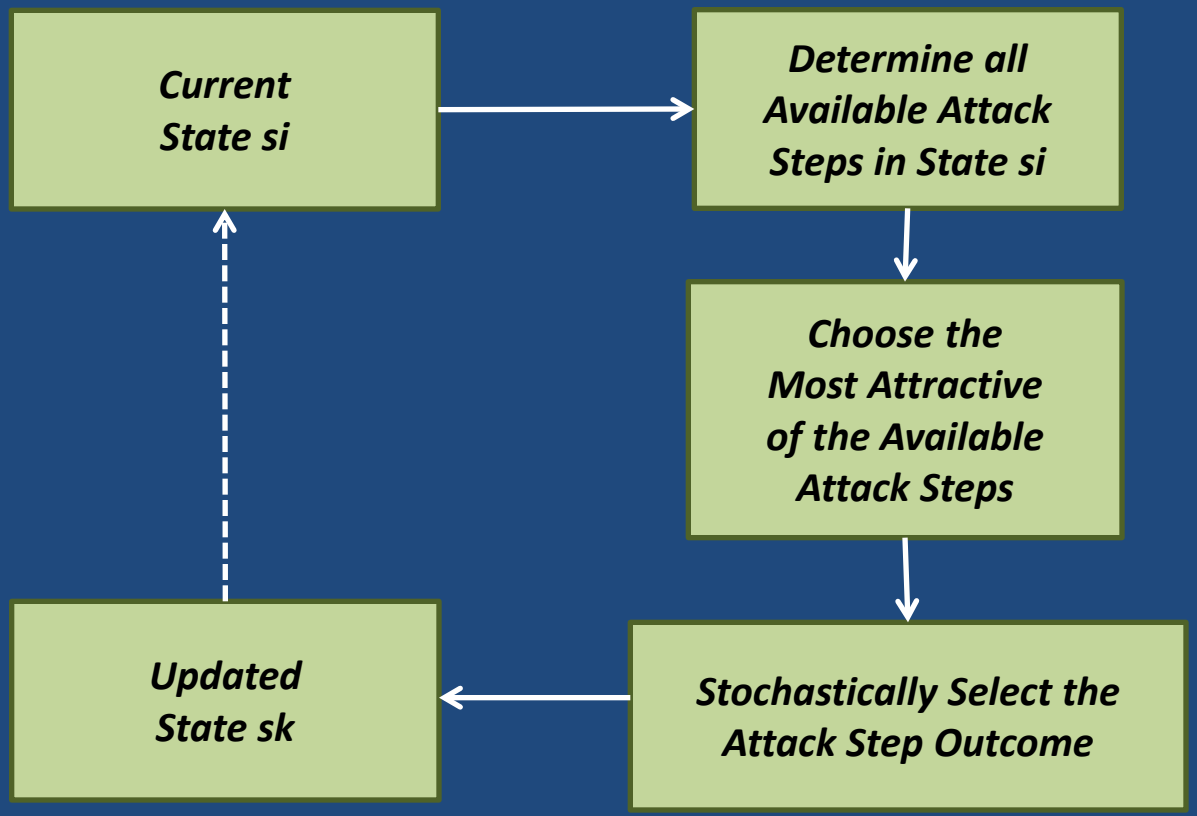
- State metrics analyze the model state
  - State occupancy probability metric (probability that the model is in a certain state at a certain time)
  - Average time metric (average amount of time during the time interval spent in a certain model state)
- Event metrics analyze events (state changes, attack step attempts, and attack step outcomes)
  - Frequency metric (average number of occurrences of an event during the time interval)
  - Probability of occurrence metric (probability that the event occurs at least once during the time interval)

ADVISE Method Overview



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.



ADVISE Model Execution Algorithm

1: Time ← 0

2: State ← s_0

3: **while** Time < EndTime **do**

4: Attack$_i$ ← β^N(State)

5: Outcome ← o, where o ~ Prob$_i$(State)

6: Time ← Time + t, where t ~ T_i(State)

7: State ← E_i(State, Outcome)

8: **end while**

Simulation time and model state initialization

Adversary attack decision

Stochastic outcome

Time update

State update

$\beta^N(s)$ selects the most attractive available attack step in model state *s* using a planning horizon of *N*

Goal-driven Adversary Decision Function

When the planning horizon N is greater than 1, the
attractiveness of an available next step
is a function of
the payoff in the expected states
N attack steps from the current state
(the **expected horizon payoff**)
and
the expected cost and detection
of those N attack steps
(the **expected path cost** and **expected path detection**).

Goal-driven Adversary Decision Function

Attractiveness of an attack step a_i
to an adversary with planning horizon N =
$U_C(E[C]) * w_c + U_P(E[P]) * w_p + U_D(E[D]) * w_d$

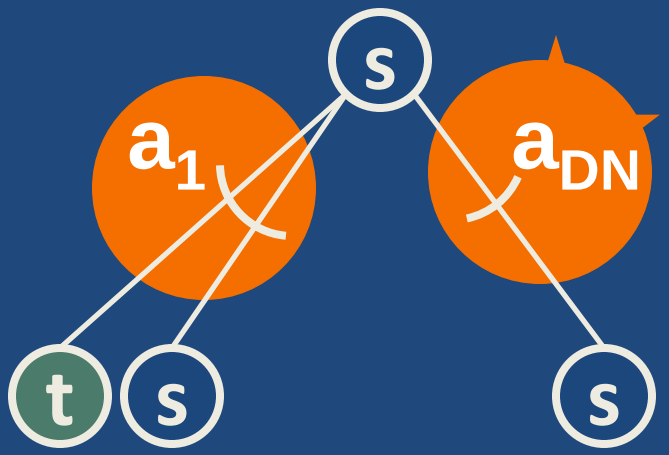
E[C] = Expected Path Cost to get to a state N attack steps away via attack step a_i.

E[P] = Expected Horizon Payoff in a state N attack steps away via attack step a_i.

E[D] = Expected Path Detection to get to a state N attack steps away via attack step a_i.

E[C], E[P], and E[D] are computed using a State Look-Ahead Tree.

Consider an adversary attack decision in state s with N = 1



Attractiveness of attack step a_i =
U_C(cost of a_i) * w_c +
U_P(E[payoff of a_i]) * w_p +
U_D(E[detection of a_i]) * w_d

C_1 = \$1000
$Pr_1(s,1)$ = 0.9
$Pr_1(s,2)$ = 0.1
$D_1(s,1)$ = 0.01
$D_1(s,2)$ = 0.1
Payoff(t) = \$0
Payoff(s) = \$0

$Attr(a_1)$ = 0.28

C_{DN} = \$0
$Pr_{DN}(s,1)$ = 1
$D_{DN}(s,1)$ = 0
Payoff(s) = \$0

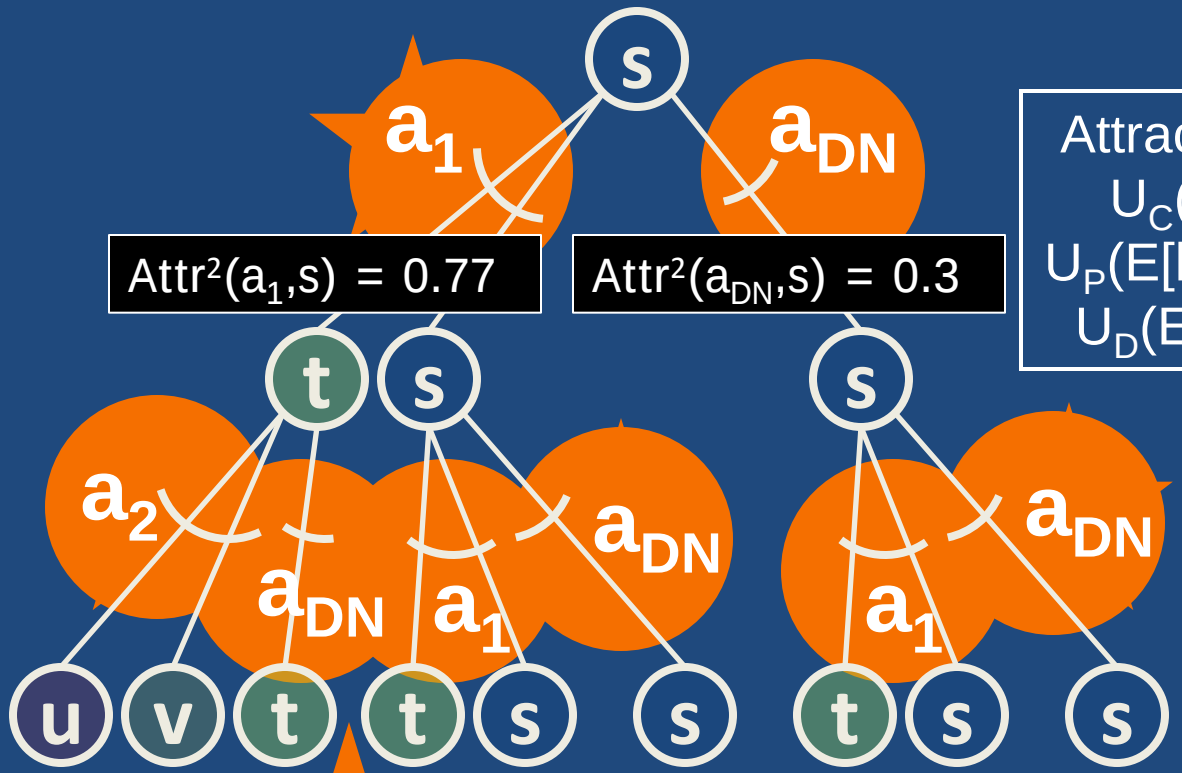
$Attr(a_{DN})$ = 0.3

$Attr(a_{DN})$ =
$U_C(\$0) * w_c$ +
$U_P(0*1) * w_p$ +
$U_D(0*1) * w_d$
= 0.3

$Attr(a_1)$ =
$U_C(\$1000) * w_c$ +
$U_P(\$0*0.9 + \$0*0.1) * w_p$ +
$U_D(0.01*0.9 + 0.1*0.1) * w_d$
= 0.28

$\beta^1(s) = a_{DN}$

Consider an adversary attack decision in state s with N = 2



$\beta^2(s) = a_1$

Attractiveness of attack step a_i = U_C(E[**path cost** of a_i]) * w_c + U_P(E[**horizon payoff** of a_i]) * w_p + U_D(E[**path detection** of a_i]) * w_d

$Attr^2(a_{DN},s)$ = $U_C(\$0) * w_c$ + $U_P(\$0) * w_p$ + $U_D(0) * w_d$ = 0.3

$Attr^2(a_1,s)$ = $U_C(\$500*0.9+\$0*0.1+\$1000) * w_c$ + $U_P(\$8000*0.9 + \$0*0.1) * w_p$ + $U_D(0.038*0.9 +0.1*0.1) * w_d$ = 0.77

$Attr^2(a_2,t)$ = $U_C(\$500*0.9+\$0*0.1) * w_c$ + $U_P(\$10000*0.8 + \$0*0.2)*w_p$ + $U_D(0.01*0.8 + 0.1*0.2) * w_d$ = 0.85

Recursive Attractiveness Calculation Algorithm

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

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

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

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

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

Optimality of the Original ADVISE Decision Rule

- ***Bellman's Principle of Optimality***

  "an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision"

- The original ADVISE decision rule implements a **provably optimal policy** when the attractiveness function is
  - wholly linear (cost and payoff only) **OR**
  - wholly multiplicative (detection only).
- The original ADVISE decision rule does ***not*** always produce an optimal decision when the decision rule combines
  - additive rewards (cost and/or payoff) **AND**
  - multiplicative rewards (detection).

Optimality of the Alternative ADVISE Decision Rule

- Alternative ADVISE Decision Rule:

$$\beta^N(s) = \arg\max_{a_i \in A_s} \{attr^N(a_i, s)\},$$

$$attr^N(a_i, s) = w_C \cdot U_C(C_i^N(s)) + w_P \cdot U_P(P_i^N(s)) + w_F \cdot U_F(\log(F_i^N(s)))$$

log nondetection

(additive)

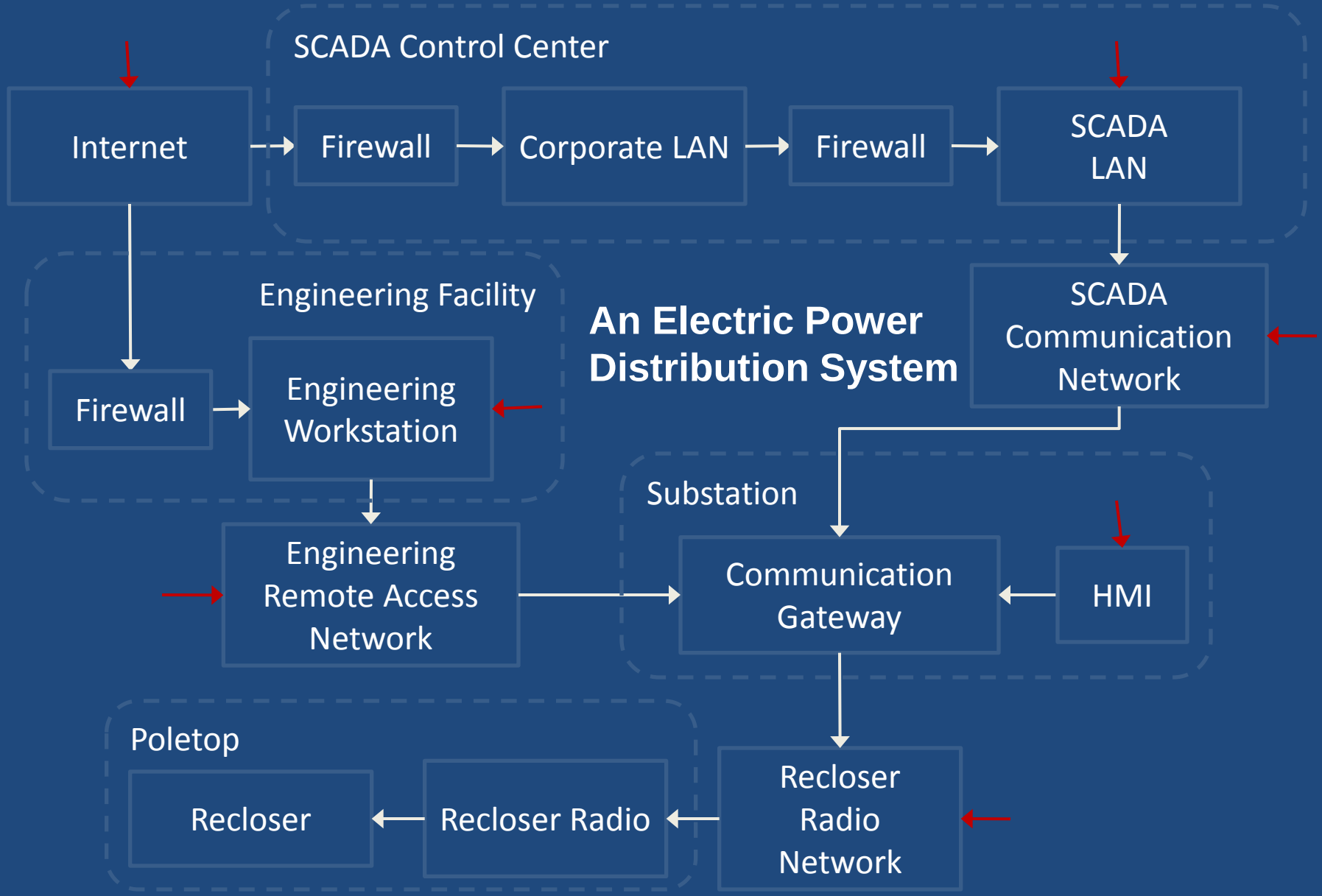
- The multiplicative *detection* term is replaced by an additive *log nondetection* term to create an alternative ADVISE decision rule that is wholly additive and, therefore, always optimal.

Practical Implications of Algorithm Optimality

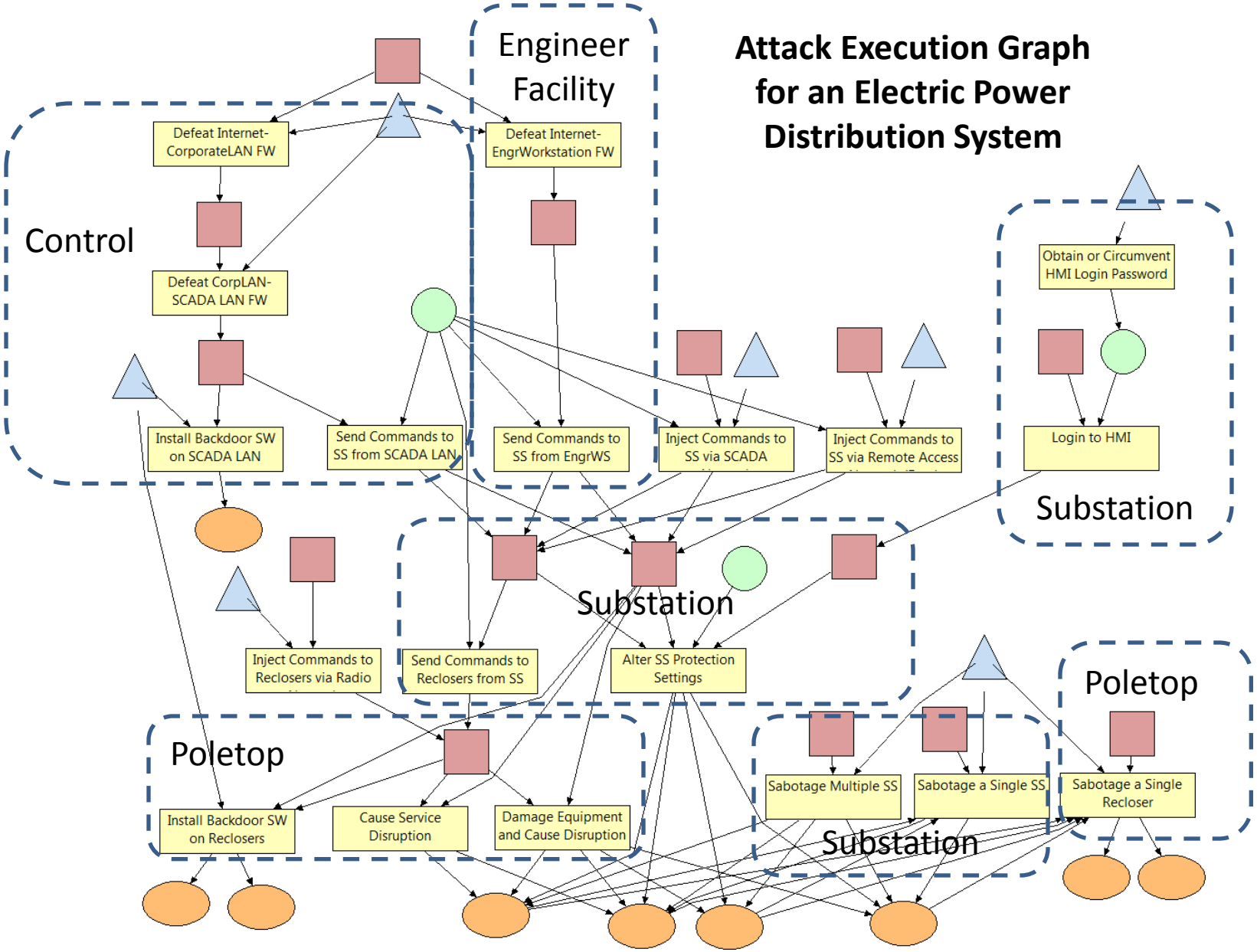
- Adversaries modeled using this algorithm exhibit “worst case” behavior, that is, they always select a next attack step that is best for them considering
  - Adversary attack preferences
  - Adversary planning horizon
  - Available attack steps
  - Attractiveness function definition

Case Study

- We investigated the effects of architectural changes on the security of an electric power distribution system
- In particular, we analyzed 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



Adversary Profiles: Decision Parameters

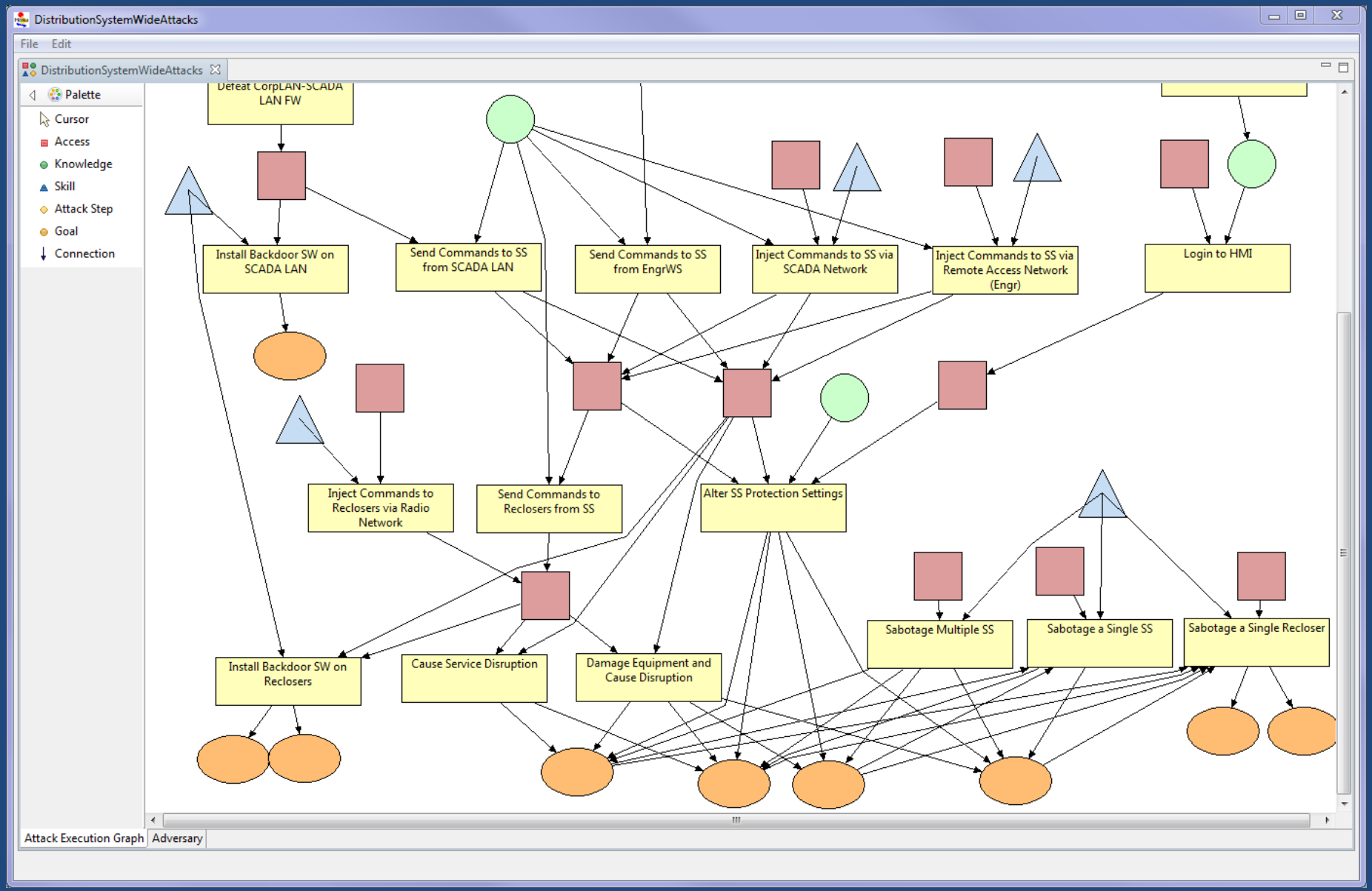
	Foreign Government	Hacker	Hostile Organization	Insider Engineer	Insider SCADA Operator	Insider Remote Technician
Cost Preference Weight	0	0.2	0.05	0.2	0.2	0.2
Detection Preference Weight	0.5	0.4	0.2	0.1	0.1	0.1
Payoff Preference Weight	0.5	0.4	0.75	0.7	0.7	0.7

- 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.

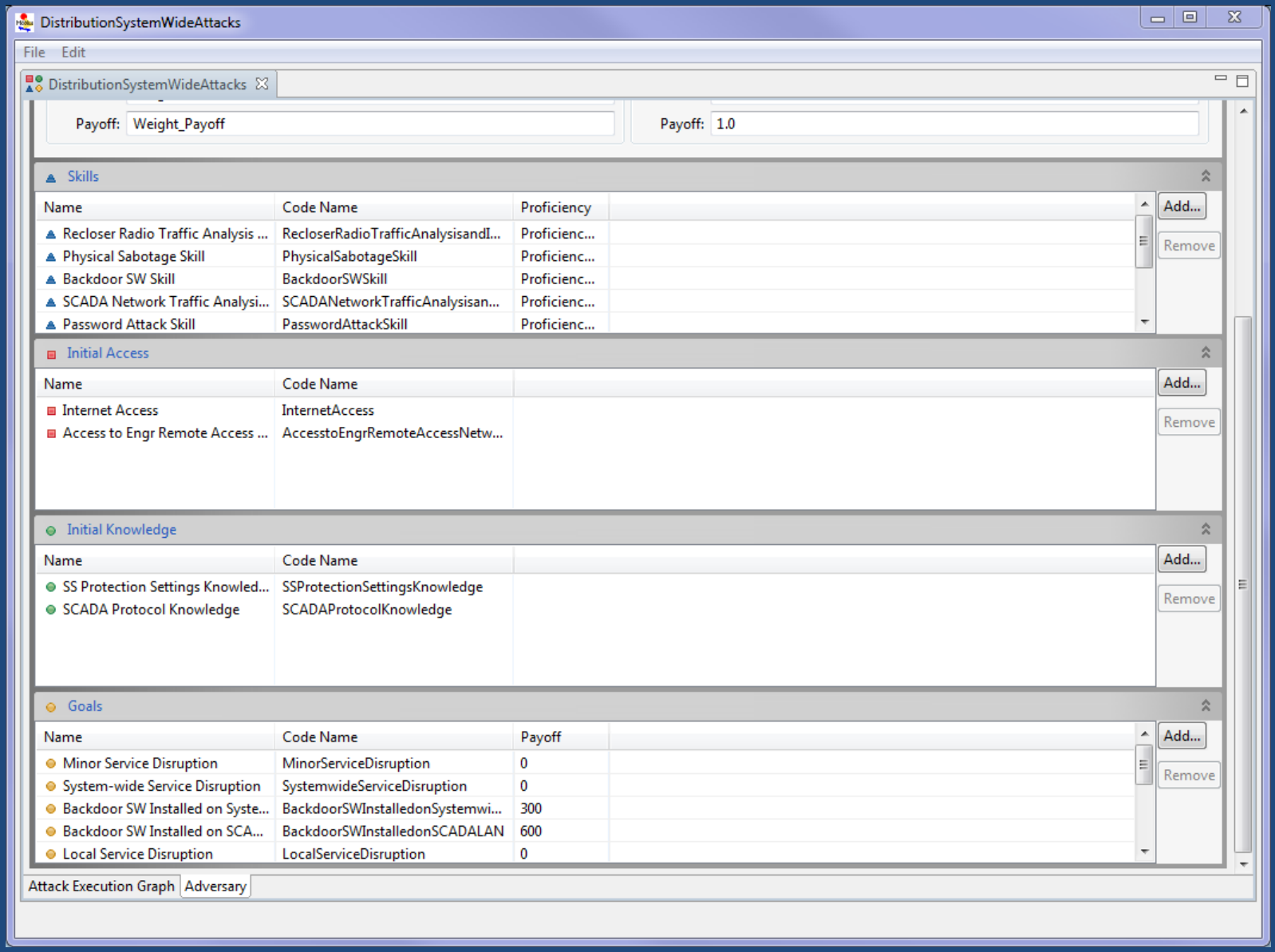
Security Metrics

- Average Number of Attempts
  - Report for each attack step
  - Gives insight on preferred attack path of adversary
- Probability of Attack Goal Achieved at End Time
  - Report for each attack goal
  - Gives insight on what goals the adversary is actively pursuing and reaching
- Average Time-To-Achieve-Goal
  - For attack goals where the above probability metric is 1
    (or close to 1)
  - Gives insight on the speed of the adversary's attack

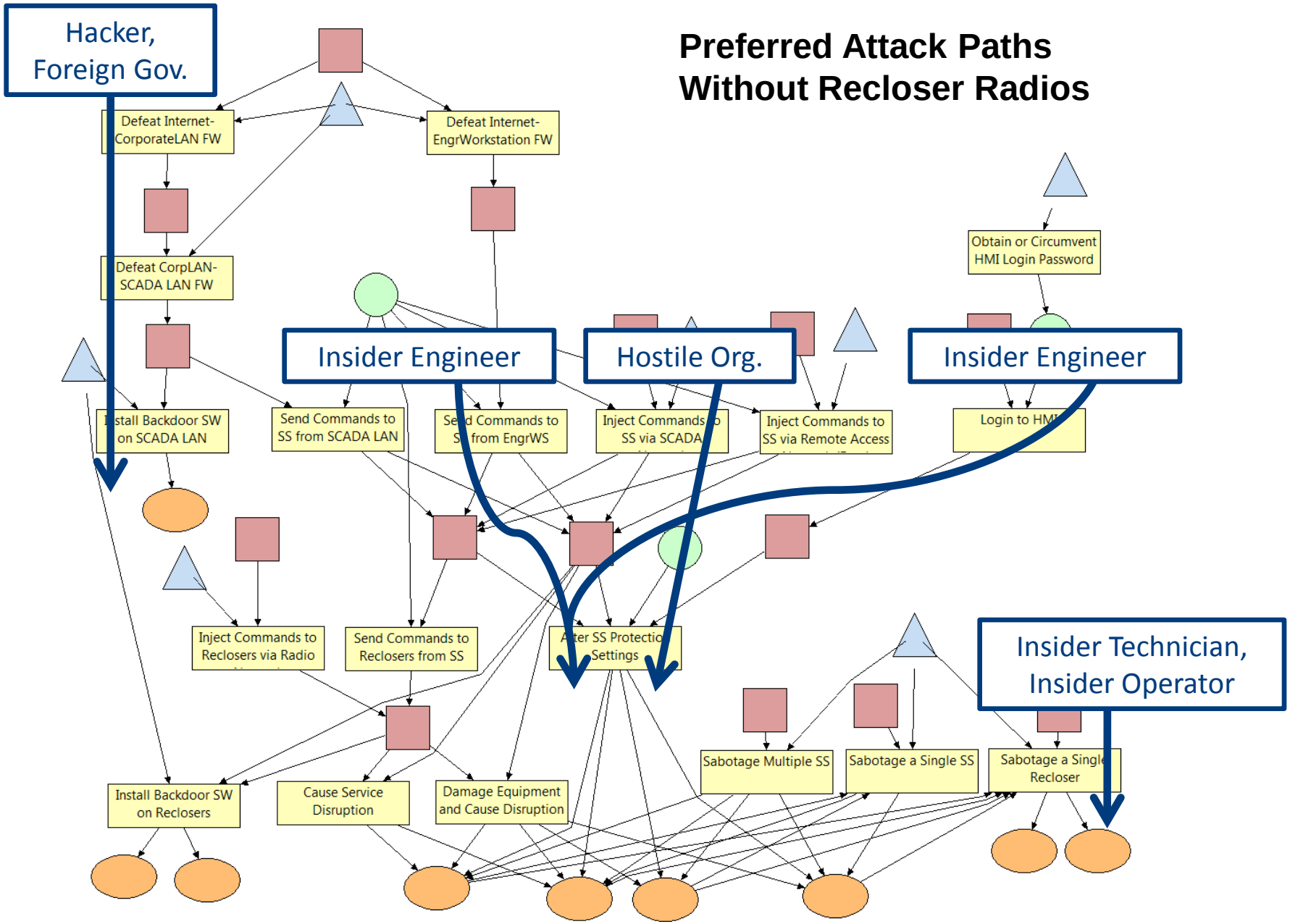
Attack Execution Graph Editor



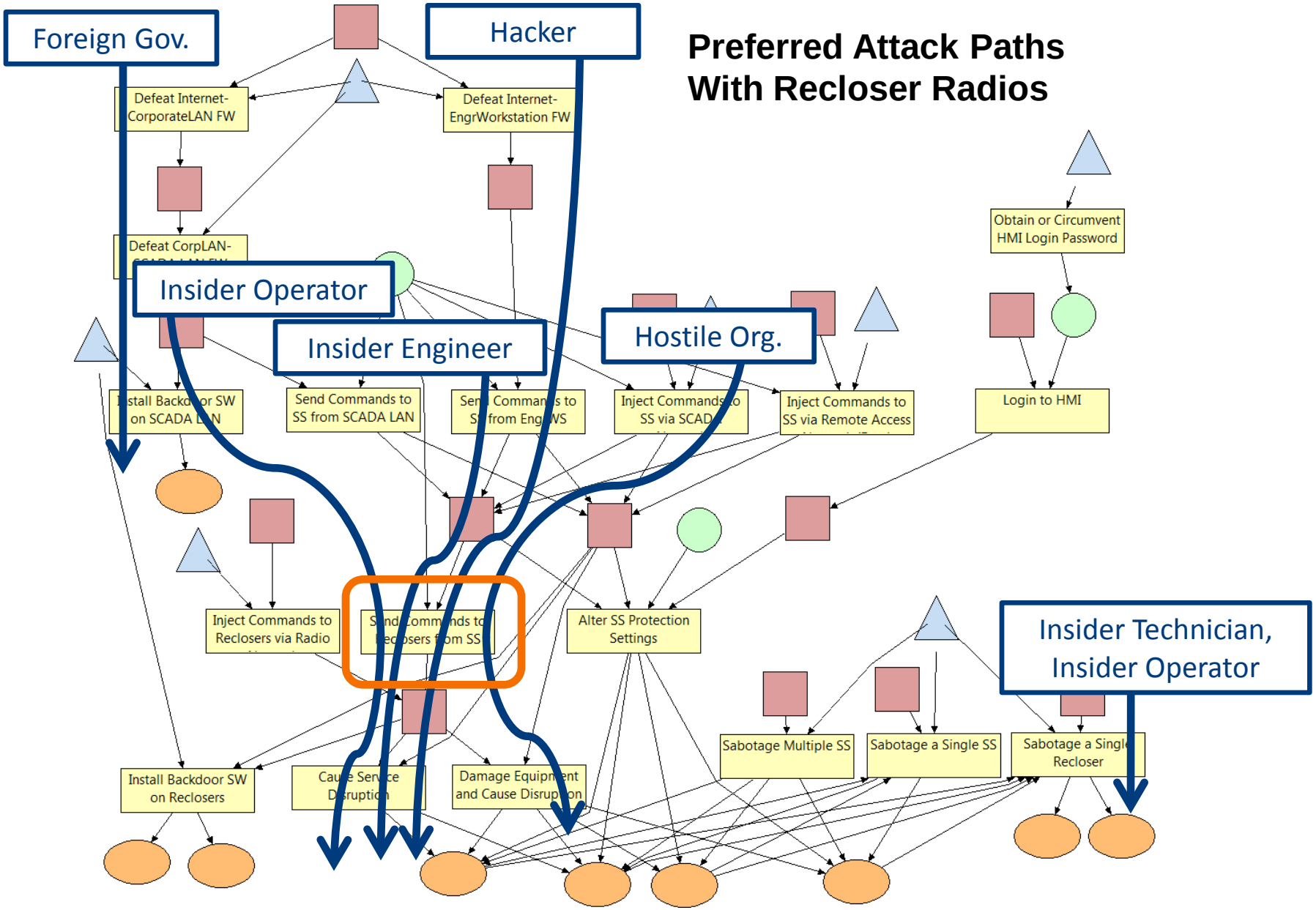
Adversary Editor



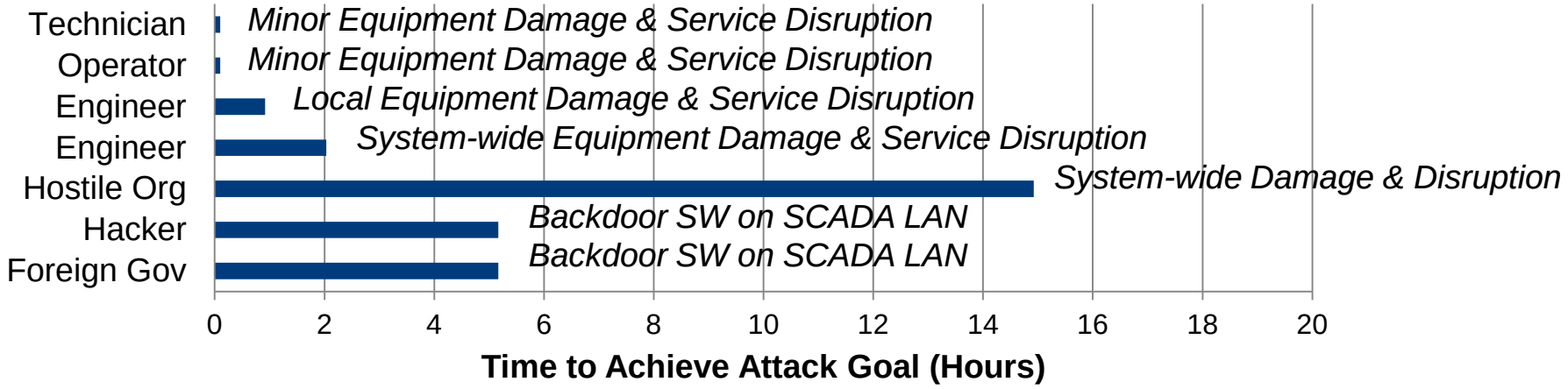
Preferred Attack Paths Without Recloser Radios



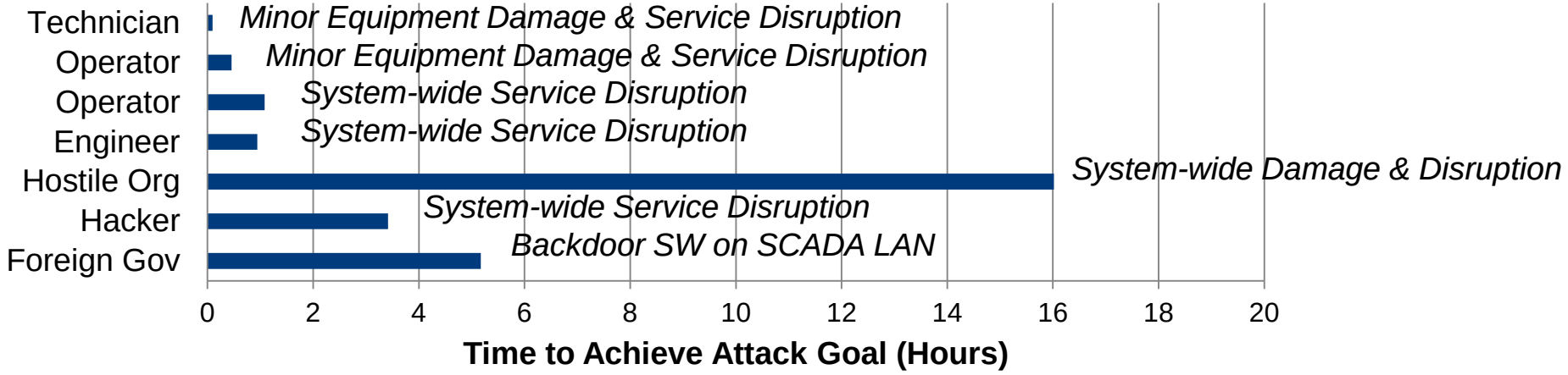
Preferred Attack Paths With Recloser Radios



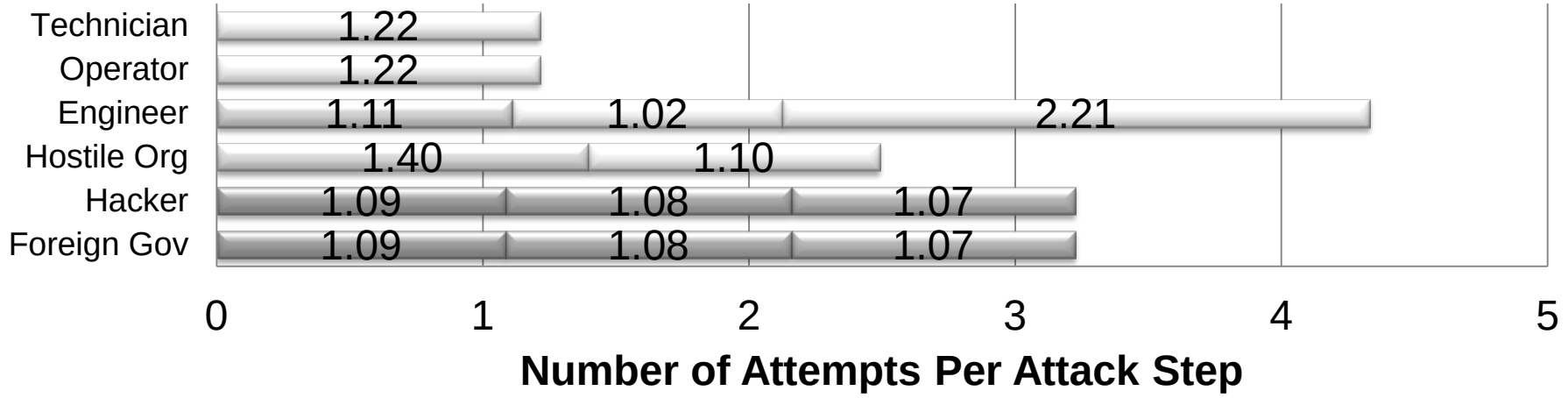
Attack Speed Without Recloser Radios



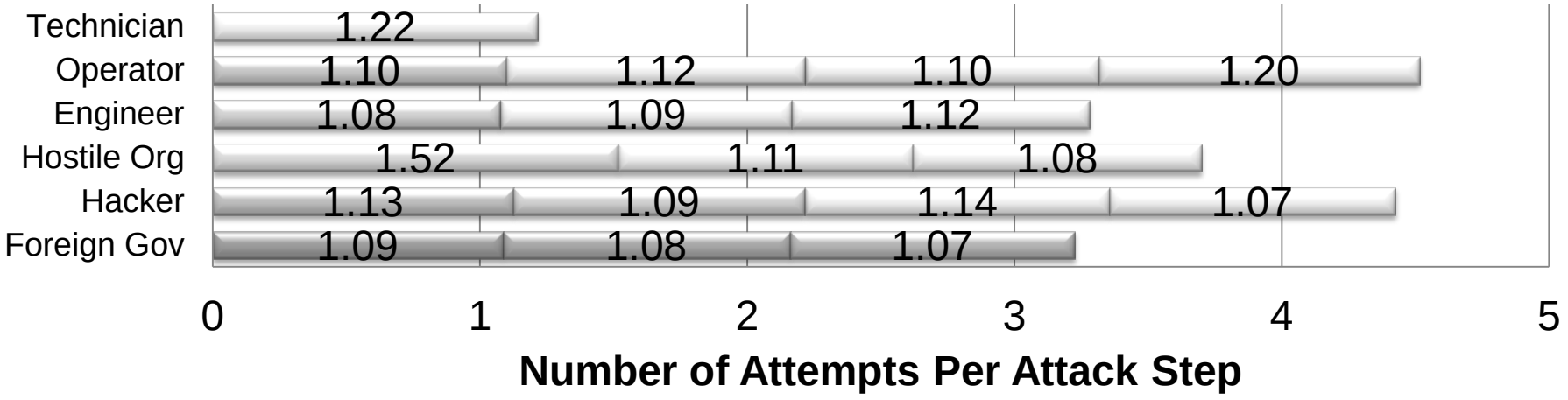
Attack Speed With Recloser Radios



Number of Attack Attempts Without Recloser Radios



Number of Attack Attempts With Recloser Radios



Acknowledgments

- Elizabeth LeMay, PERFORM Group Member
- Michael Ford, PERFORM group Member
- Ken Keefe, Lead Möbius Developer
- Carol Muehrcke, Cyber Defense Agency, LLC
- Willard Unkenholz and Donald Parks, U.S. Department of Defense

Case study collaborators

- Bruce Barnett and Michael Dell'Anno, GE Research

Conclusions

- Since system security cannot be absolute, quantifiable security metrics are needed
- Metrics are useful event if not perfect; e.g., relative metrics can aid in critical design decisions
- The ADVISE formalism, and its implementation in Mobius-SE
  - Is rich enough to adversary, user, and system behavior
  - Natural for security analysts
  - Semantically precise
- Mobius-SE is in alpha-test, and has been distributed to 10 organizations (industry, govt., & academics) who are using it in real case studies
- Work is on going on 1) analytic solution methods and 2) modeling human user behavior

Questions and Discussion

Bill Sanders
whs@illinois.edu