



ADVISE Meta – Alpha Tool Workshop

August 16, 2016



Andy Crapo, Brett Feddersen, Alfredo Galbadon, Ken Keefe, Carol Muehrcke, Michael Rausch, Bill Sanders, and Ron Wright

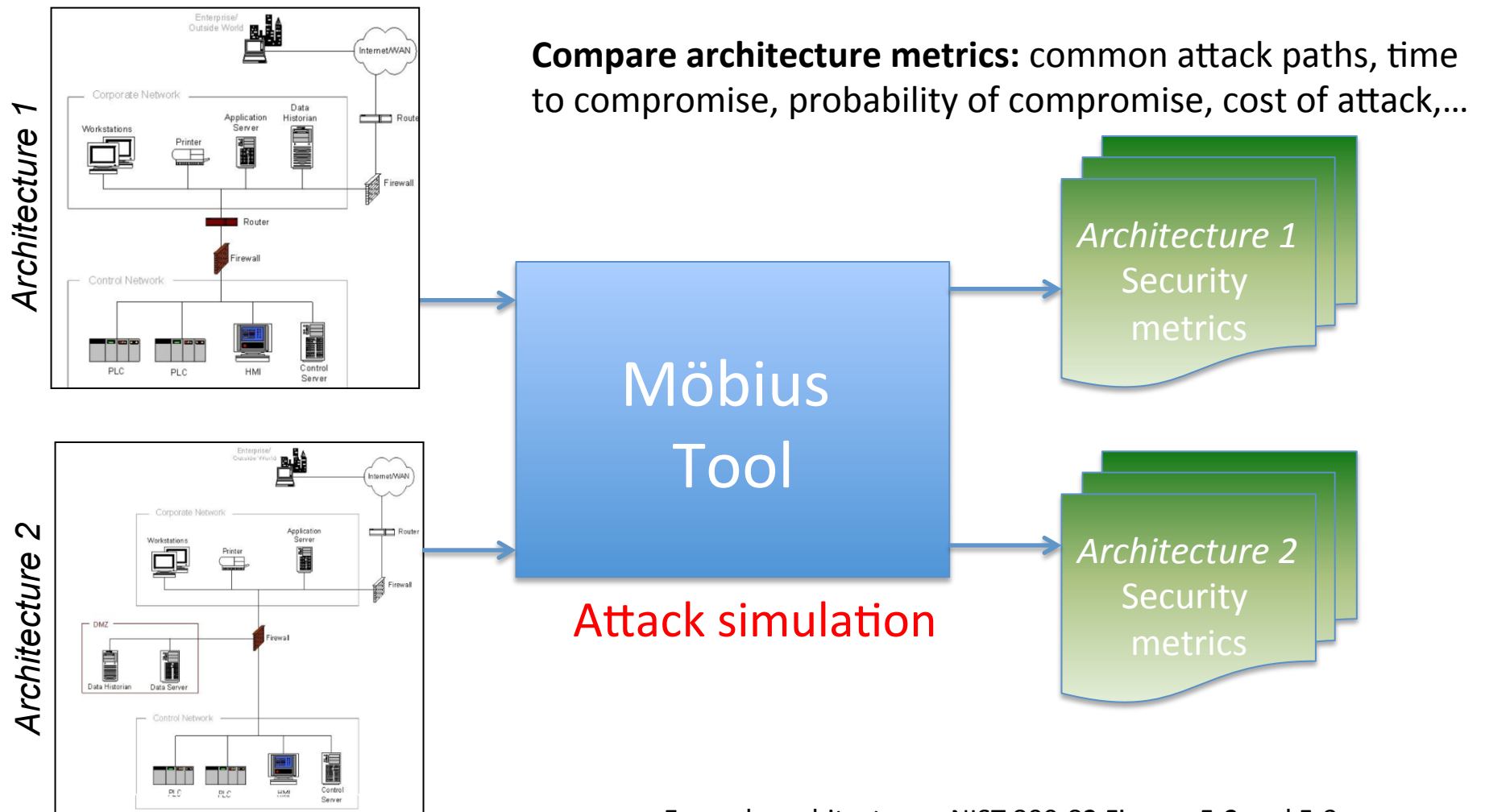
Agenda

- Registration and Continental Breakfast
- Welcome
- Goals
 - Tool
 - Workshop
- Steps to Use ADVISE Meta
- Hands on Sessions
- Advanced Ontology: Selected Details and Customization
- Case Studies
- Wrap Up

ADVISE Meta Introduction

- Today: no scientific basis for designing security architectures
 - Follows from: no scientific basis for estimating effectiveness of security measures before deployment
- Today: security metrics
 - Before deployment, count countermeasures
 - Judge effectiveness based on experience, intuition
 - After deployment, count intrusions
- Purpose of ADVISE Meta
 - Provide scientific basis for design decisions by calculating security metrics at design time
 - Auditable results
 - No requirement for deep modeling or cybersecurity expertise

ADVISE Meta Tool on Möbius



Example architectures: NIST 800-82 Figures 5-2 and 5-3

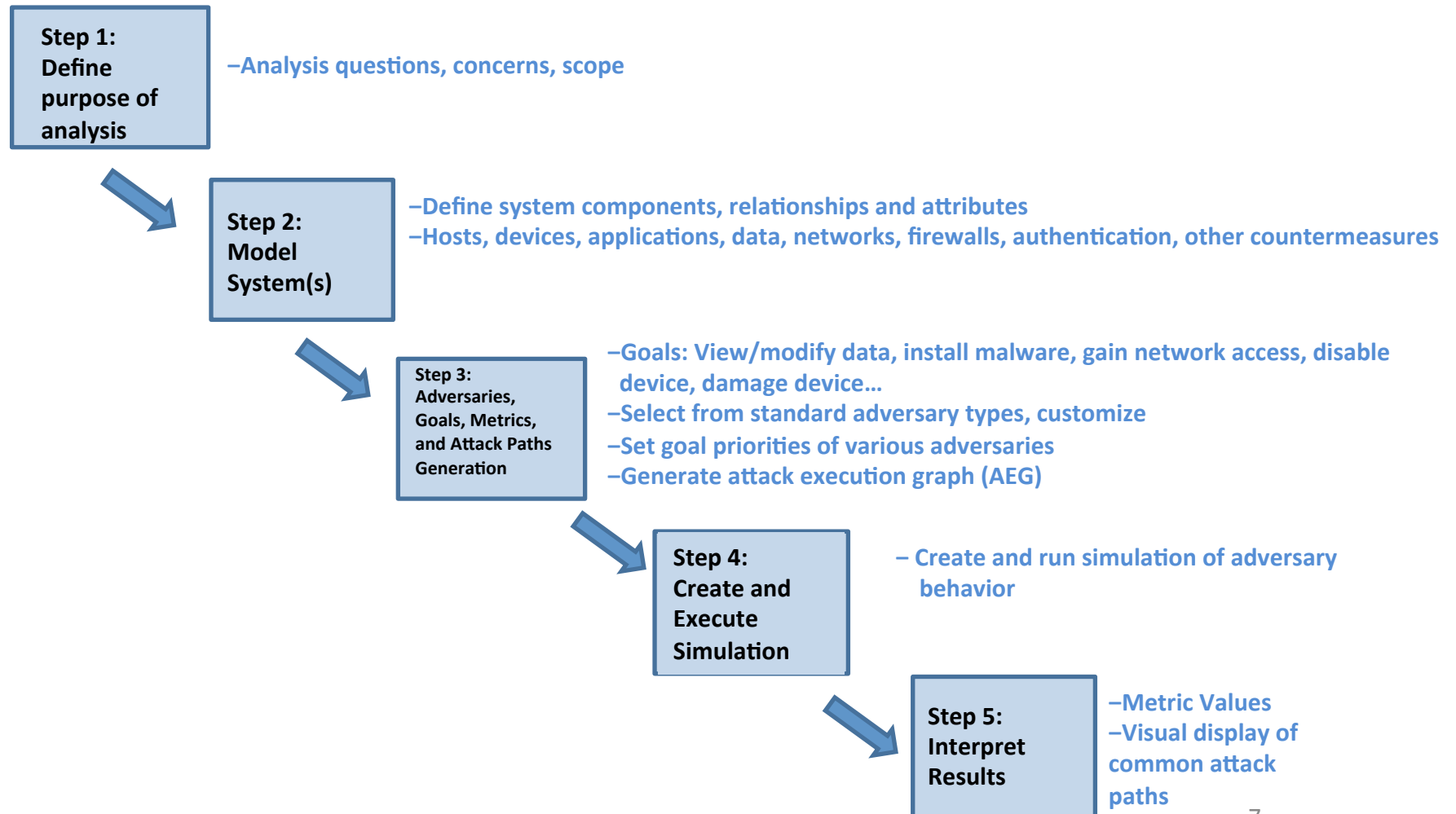
Workshop Goals

- Introduce the tool to the community
- Gather feedback about all aspects of the tool
 - High level concepts
 - Workflow
 - User Interface
 - Usability
- Feedback discussion at the end of each hands-on session

Agenda

- Registration and Continental Breakfast
- Welcome
- Goals
 - Tool
 - Workshop
- Steps to Use ADVISE Meta
- Hands on Sessions
- Case Studies and Custom Ontologies
- Wrap Up

Using ADVISE Meta

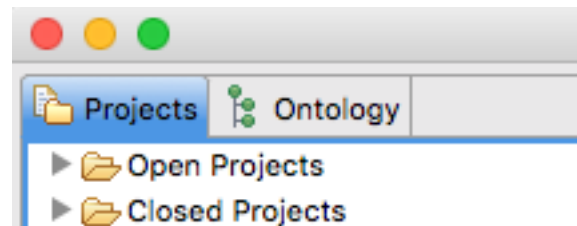


Agenda

- Registration and Continental Breakfast
- Welcome
- Goals
 - Tool
 - Workshop
- Steps to Use ADVISE Meta
- Hands on Sessions
- Case Studies and Custom Ontologies
- Wrap Up

Step 0 – Install and Ontology Overview

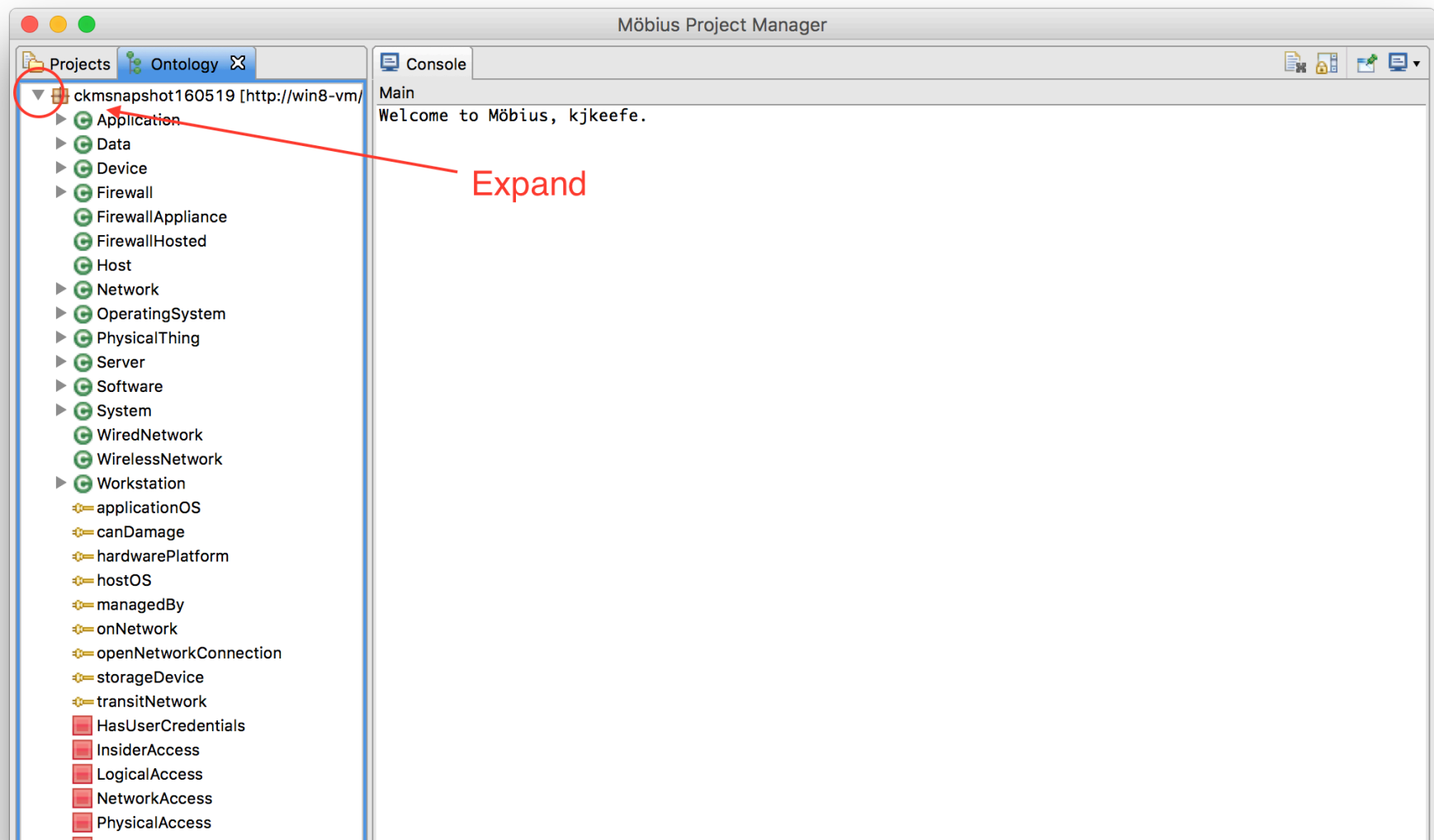
When you start Mobius, you will see two tabs in the upper left hand corner












- The Project tab contains projects with models of systems, attackers, and metrics.
- The Ontology tab is where individual system components, relationships, attributes, etc. are defined.

Step 0 – Install and Ontology Overview

1. On the ontology pane, click on the arrow to left of the ontology name, to expand the ontology



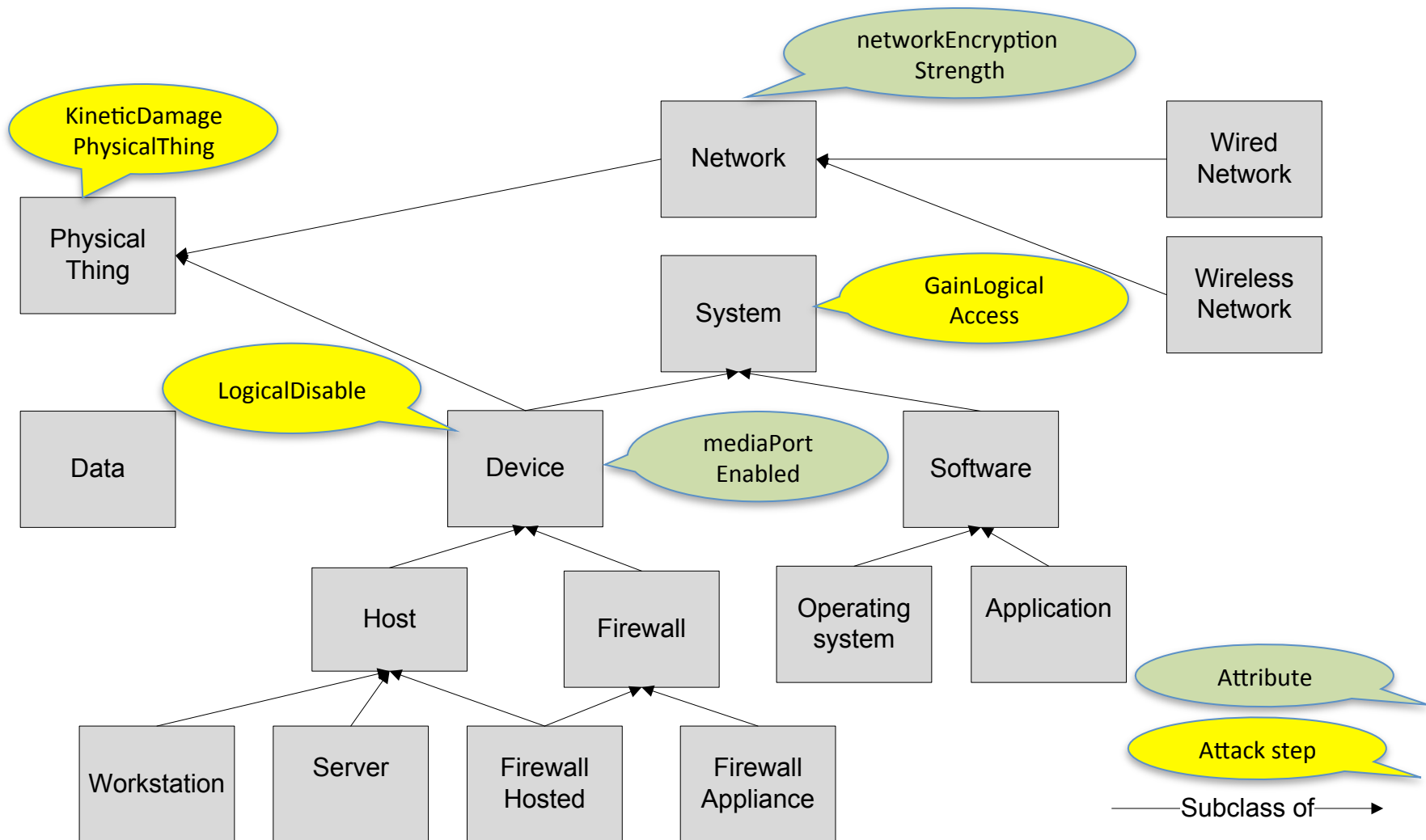
ADVISE Meta - Ontology

- Ontology (general definition): a particular theory about the nature of being or the kinds of things that have existence
- ADVISE Meta Ontology: types of things available to build a system model and simulate cyber attacks against it
- Concepts in tool ontology fall in these categories
 - Types of components 
 - Attributes of components by type 
 - Relationships between components 
 - Types of access,  skills,  and knowledge  an adversary may have
 - Types and characteristics of adversaries 
 - Attack steps 
 - State variables: other system, component, or adversary properties 

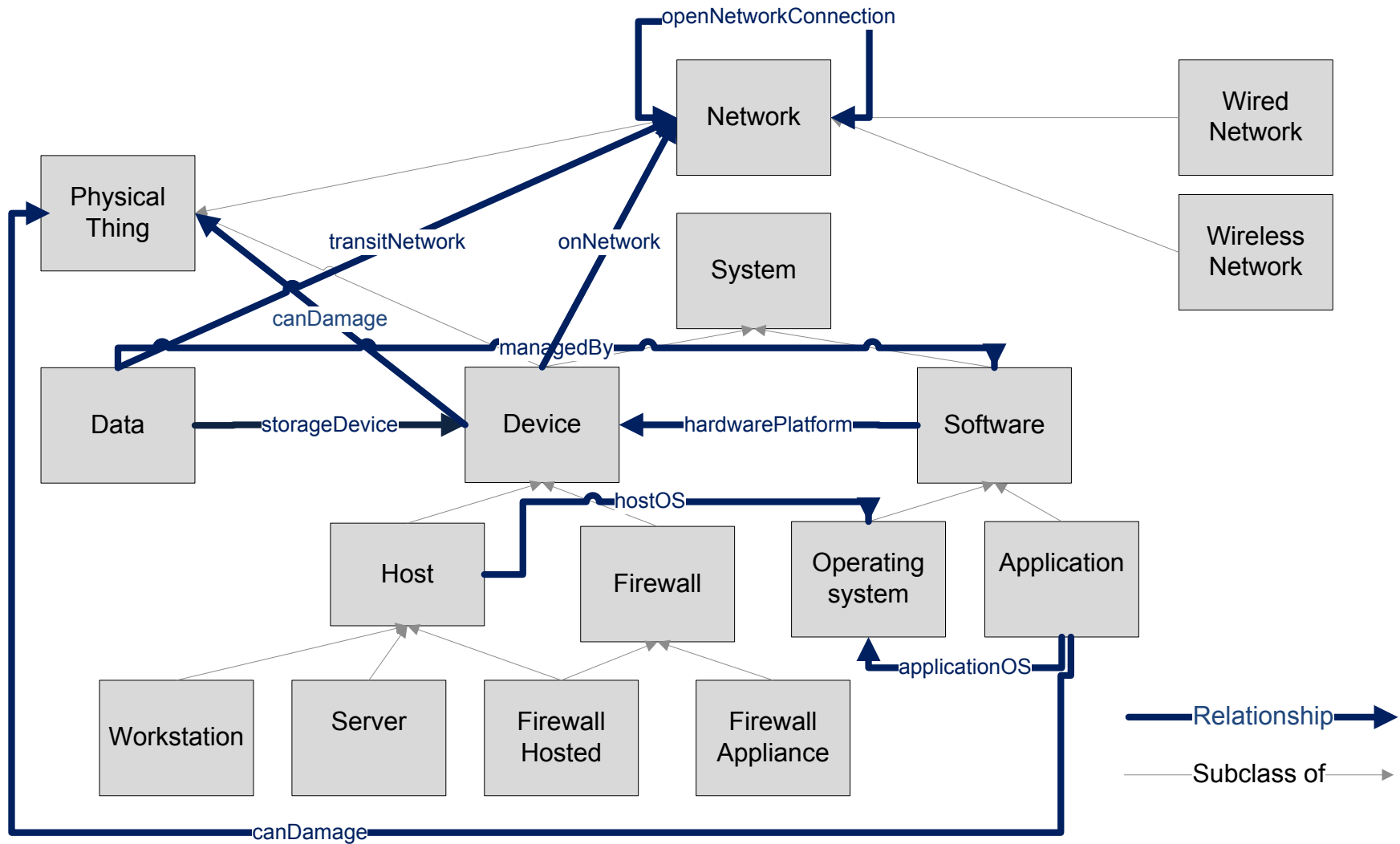
Base Ontology

- Base ontology used for hands-on exercise
- Representative of tool capabilities
- Grounded by:
 - Research on attack methods
 - Study of example analysis previously done with hand-created AEG
- Not yet a “complete” or vetted dataset
- Modifiable, replaceable (more on this later)

Base Component Ontology and Inheritance




Base Ontology Relationships



Base Ontology Types of Access

- PhysicalAccess(X), where X is a PhysicalThing
 - Not achievable via any attack step, must be given as initial condition
- NetworkAccess(X), where X is a Network
 - Able to read and write bits on the network
- UIAccess(X), where X is a Device or Software
 - Able to touch the login function (if any)
- HasUserCredentials(X), where X is a Device or Software
 - Has the password, token, key, or other credential required to access user functions provided by X
- LogicalAccess(X), where X is a Device or Software
 - Able to access user functions provided by X

Base Ontology Skills

- See list of skills on ontology tab with symbol 
- Adversary templates define default skill proficiencies
- Most skills are generic
- Reason for adding system specific skills (“Specialized”)
 - Model the tremendous advantage they provide to adversary
- Reason for using broad skill categories
 - Represents how real adversaries accumulate skills
 - Fine grained skill proficiencies (e.g. at stealing passwords or breaking VPNs) unlikely to be known or even guessable in an actual case
 - Haven’t seen reason yet for increasing data input requirements and complexity in attack step models

Base Ontology Attack Steps

Damage or disable

KineticDamagePhysicalThing
 LogicalDamagePhysicalThing
 PhysicalDisable
 LogicalDisable

Malware

CreateTrustedSiteCauseMalwareInstall
 CreateUnTrustedSiteCauseMalwareInstall
 CreateRemovableMediaCauseMalwareInstall*
 StagePackageCauseMalwareInstall*
 InstallMalwareFromFixedMedia*
 InstallMalwareFromRemovableMedia*

*Not in alpha

Gain access

GainLogicalAccess
 GainUserCredentials
 GainLocalUIAccessDevice
 GainLocalUIAccessOS
 GainRemoteUIAccessDev
 GainRemoteUIAccessOS
 GainNetworkAccessViaNetworkNode
 GainNetworkAccessViaNodeOnConnectedNetwork
 GainNetworkAccessViaConnectedNetwork
 AdminModifyFWOpen
 CircumventFWRules
 PlaceRogueHostOnNetwork

Compromise data integrity

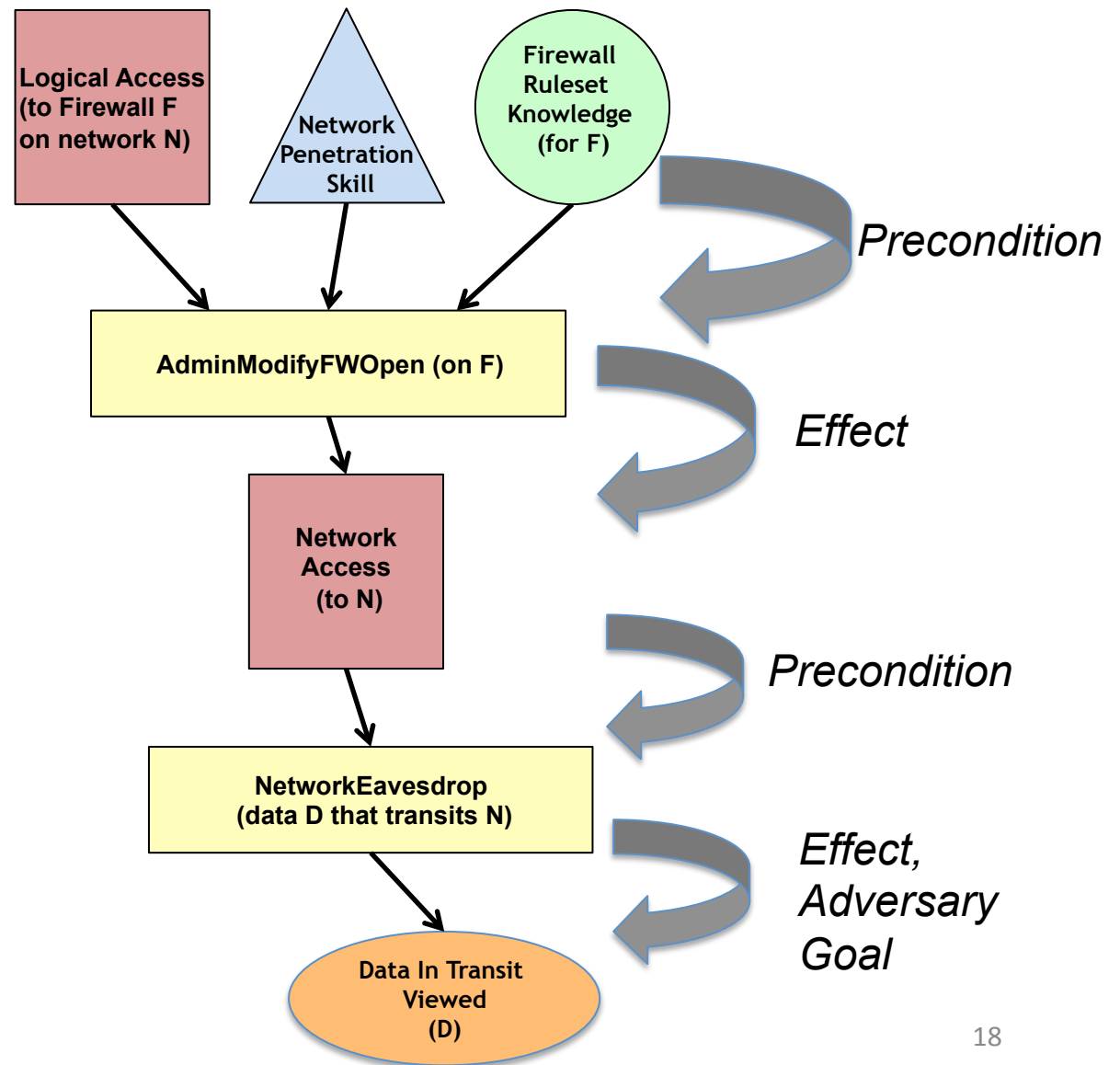
ModifyDataLocally

Compromise data confidentiality

ReadManagedDataLocally
 NetworkEavesdrop

Fragment of Attack Execution Graph (AEG)

- AEG describes potential attack paths from attacker viewpoint
- Attack step attempt requires certain *access, skills, knowledge* and/or *system state conditions*
- Attack outcome can affect any of these elements
- Any of these elements may be selected as an adversary goal



Hands-on Example Step 1 – Define Purpose of Analysis

What kinds of systems can be analyzed?

- With the base ontology: Enterprise system architectures that may have:
 - Networks hosting cyber and cyber-physical devices
 - Applications, data
 - Internet connections
 - Boundary protections and other common countermeasures
 - Design phase or existing
- With arbitrary ontology:
 - In theory, any system built of components, where attacks are constructed by linking attack steps against components

Example Enterprise Systems

- Architecture to support stock trading
- SCADA architecture supporting an electric utility
- Control systems in a water treatment plant
- Operations and administration systems for a telecommunications provider
- 911 computer systems architecture
- Reactor safety architecture for a nuclear power plant
- Systems in a hospital that process patient information
- Air traffic or train control systems
- Computer infrastructure for a research and development facility
- Computer infrastructure for an ISP

Step 1 – Define Purpose of Analysis (cont.)

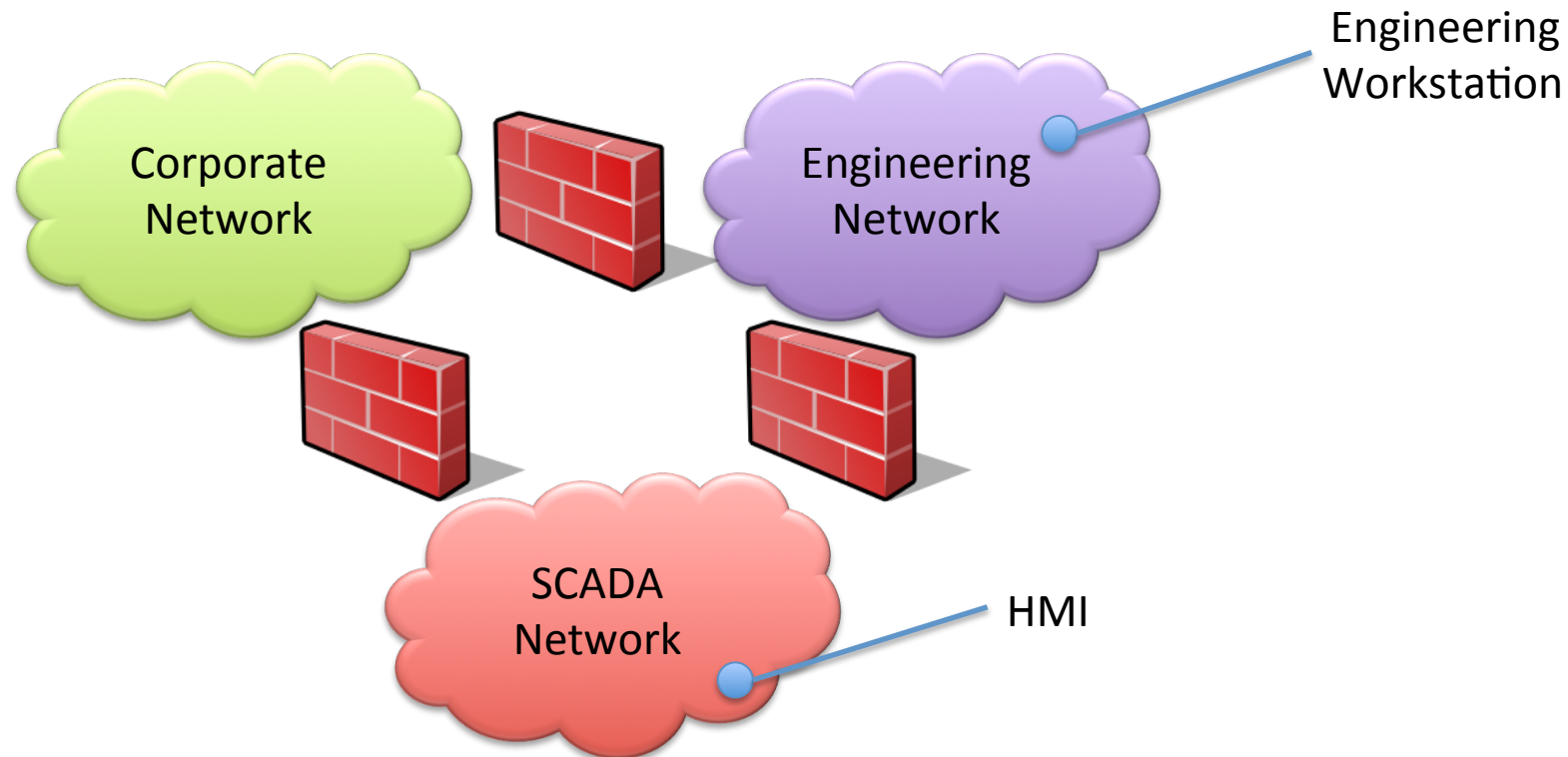
What kinds of questions can be answered?

- How susceptible is a system to cyber attacks?
- What can be done to decrease susceptibility?

Typical examples of analyses

- Which among alternative architectures should be recommended?
- What are security weak points of an architecture?
- Is a proposed countermeasure worthwhile?
- How will proposed functional architecture changes impact security?
 - If security decreases, how can decrease be minimized?

Step 1 – Define Purpose of Analysis – Small SCADA Networks



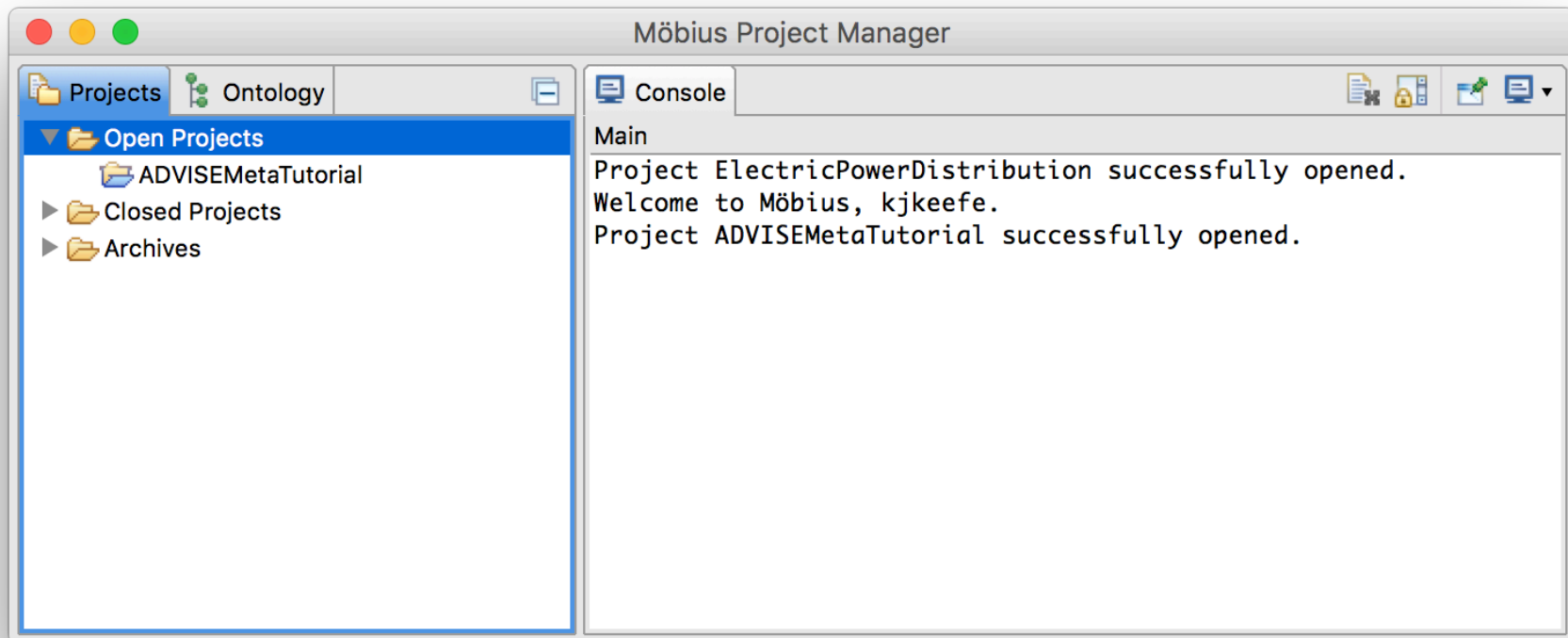
- Predicted electricity demand data stored on the SCADA LAN is found on the desk of an engineer not authorized to view this data. This data can be sold to competing electricity vendors to aid in their pricing.
- How could the engineer even gain access to the SCADA LAN?
- The engineer has physical access to all networks shown and to the HMI, and is an authorized user of the engineering workstation.
- What changes to the architecture would make this less likely to happen again?

Step 1 – Define Purpose of Analysis - Feedback

- Open questions/discussion on Step 1
- Questions for group:
 - What are some typical security architecture decisions for which rationale is hard to come by?
 - Are there specific systems that don't fall under enterprise systems, but that might benefit from this type of analysis?

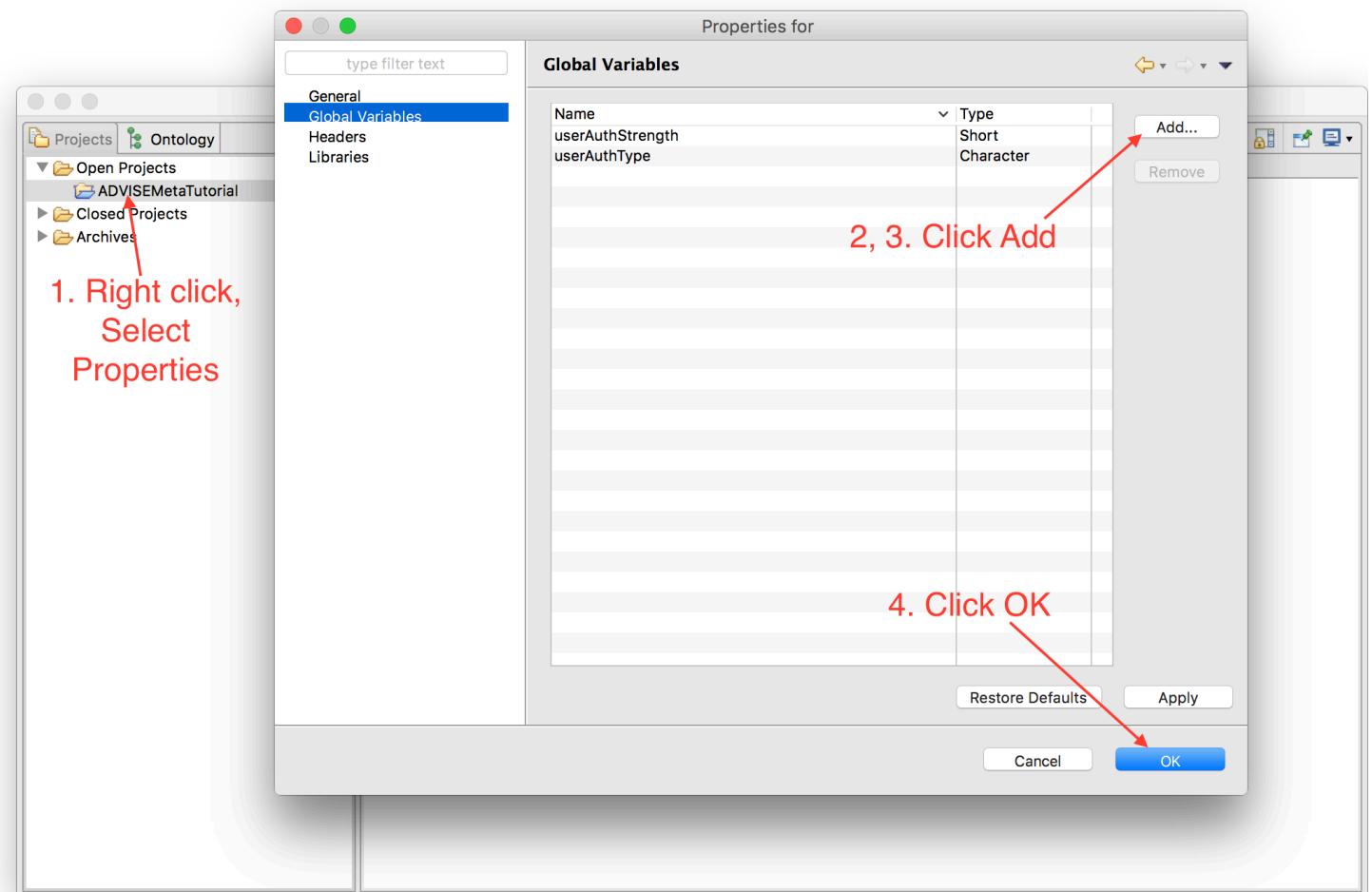
Step 2 – Model Systems

1. Select the Projects tab.
2. Right click Open Projects and select New Project...
3. Name the project **ADVISEMetaTutorial** and click Finish



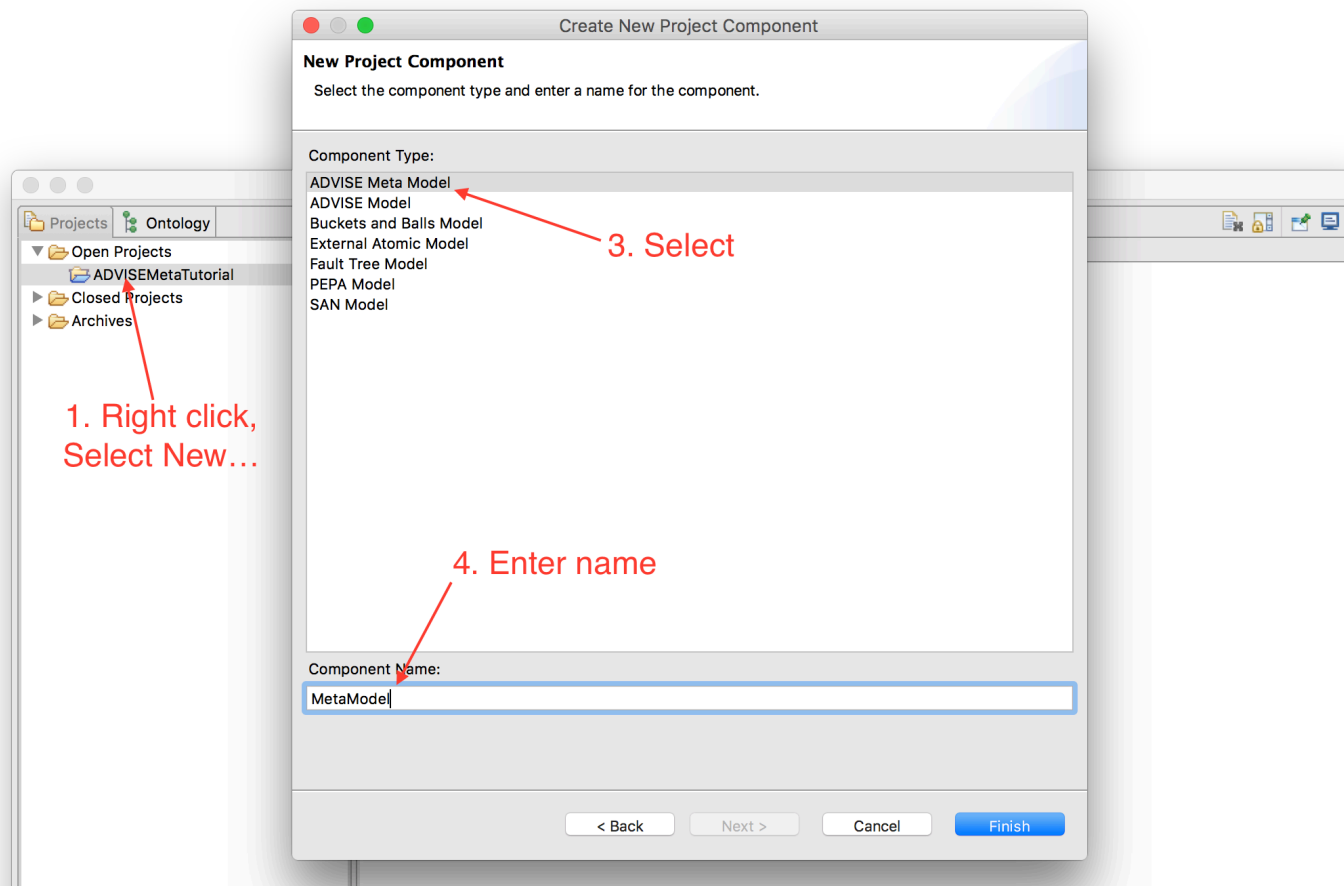
Step 2 – Model Systems

1. Right click the ADVISEMetaTutorial project, select Properties...
2. Add a global variable called **userAuthStrength** with type **Short**.
3. Add a global variable called **userAuthType** with type **Character**.
4. Click OK



Step 2 – Model Systems

1. Right click the ADVISEMetaTutorial project, select New...
2. Select **Atomic** in the bottom pane and click Next.
3. Select **ADVISE Meta Model** in the list.
4. Enter the name **MetaModel** and click Finish.



Step 2 – Model Systems

The screenshot displays the MetaModel application window. On the left, there is a sidebar with two main sections: 'Components' and 'Ontology'. The 'Components' section lists 'Data', 'PhysicalThing', and 'System'. The 'Ontology' section has an 'Edit' button and a table with columns 'Name' and 'Version'. The table contains one entry: 'ckmsnapshot16...' with version '2016.07.06.0'. Below the sidebar is a navigation bar with tabs for 'System', 'Goals', 'Adversaries', 'Metrics', 'Configurations', and 'Generator'. The main area of the window is a large white space labeled 'System Model Canvas' in red text. Two red arrows point from the text '1. Available Components' and '2. Loaded Ontologies' to the 'Components' and 'Ontology' sections respectively.

MetaModel

Components

- Data
- PhysicalThing
- System

1. Available Components

System Model Canvas

Ontology

Name	Version
ckmsnapshot16...	2016.07.06.0

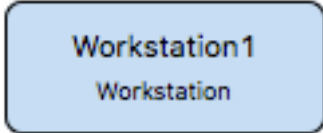
2. Loaded Ontologies

System Goals Adversaries Metrics Configurations Generator

Step 2 – Model Systems

- Component

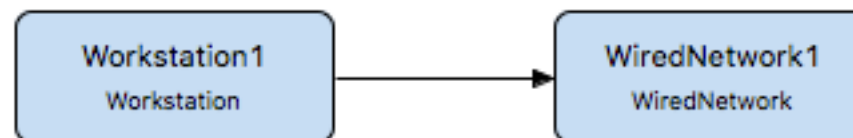
- Part or element of the system
- Physical objects, e.g., computer, firewall, building, etc.
- Logical objects, e.g., data, software, etc.
- Components are represented as blue rectangles on the system diagram



Workstation1
Workstation

- Relationship

- A semantic connection between two components.
- For example, a computer is connected to a network through a **onNetwork** relationship, or data is managed by a software application through a **managedBy** relationship.
- Relationships are represented as arcs on the system diagram.



Step 2 – Model Systems

- Attributes
 - Security relevant properties associated with a component
 - For example, a component might use a specific type of authentication mechanism
 - Listed in the “Details” of the component

Workstation1 Details

Component Details
Specify the details for the component.

Name:

Attributes

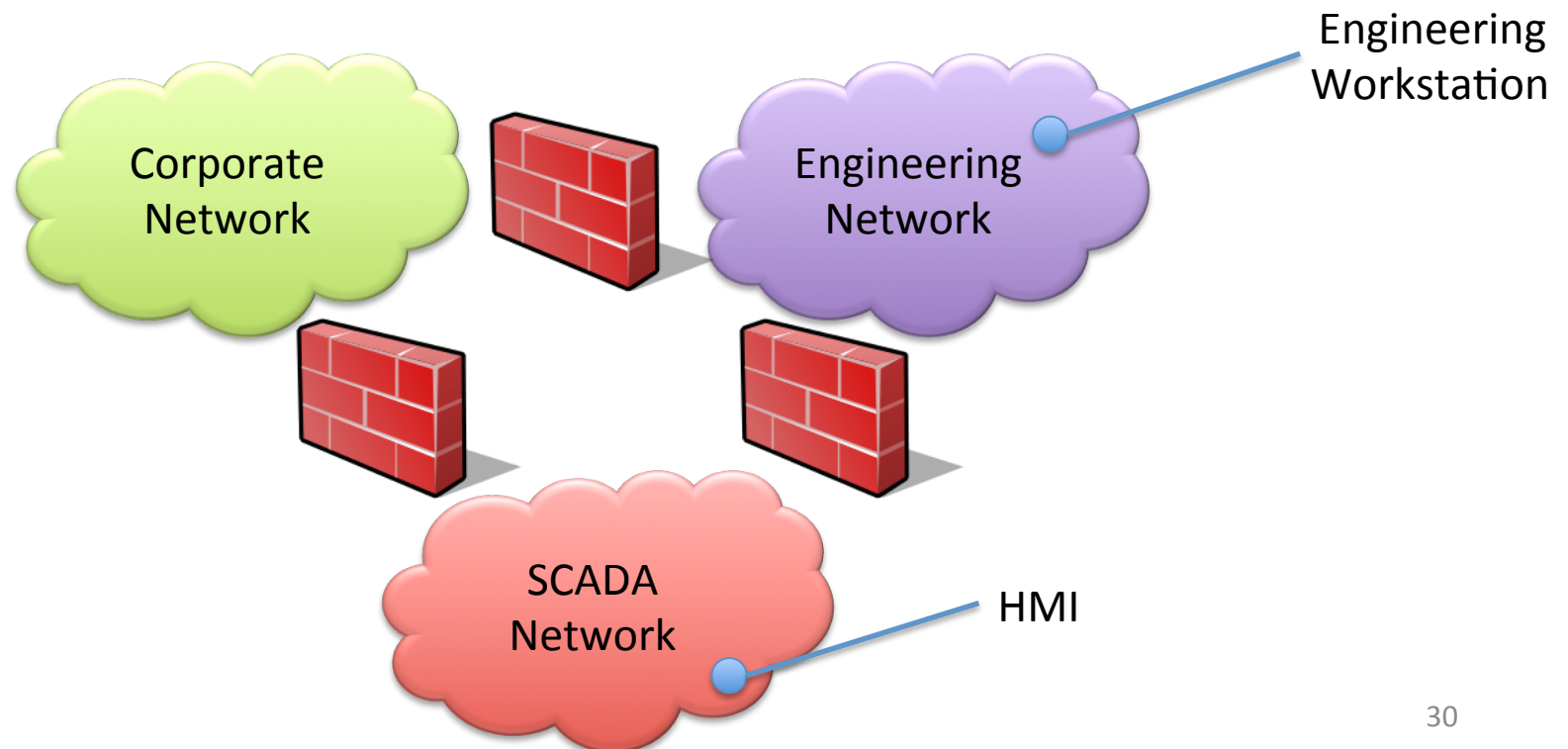
componentAnomalyDetectionStrength	0
credentialMonitoring	0
deviceStatusControl	1
deviceStatusDetection	2
mediaPortEnabled	1
physicalAttackAttribution	2
resistanceToKineticDamage	0
resistanceToLogicalDisable	0
resistanceToPhysicalDisable	0
softwareTrustedSourceSecurity	5
softwareTrustedSourceSecurity	5
strengthOfUserAuthentication	0
userAuthenticationType	N
userCyberSecurityAwareness	3
userCyberSecurityAwareness	3

Cancel Finish

Step 2 – Model Systems

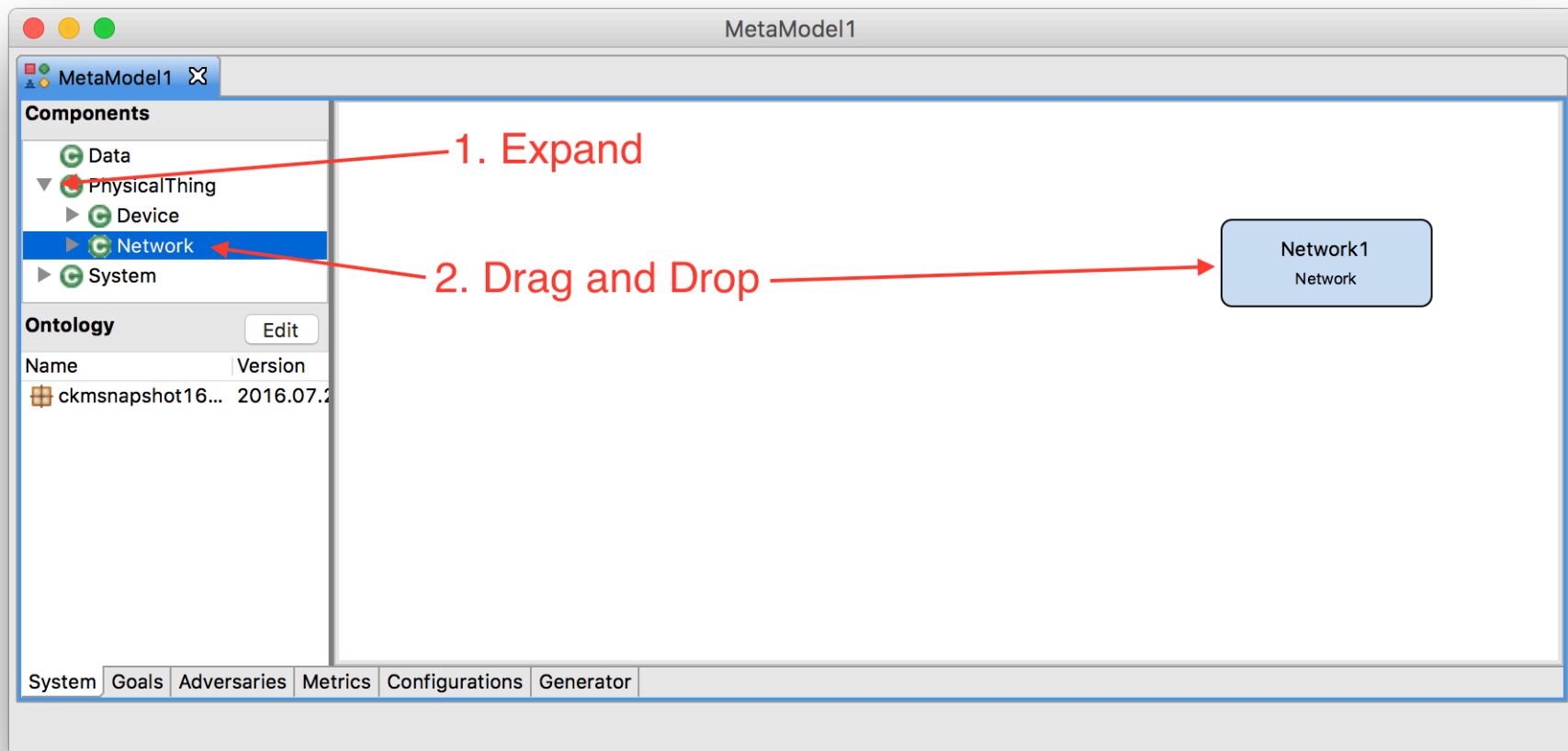
Our simple example consists of:

- SCADA Network with a local terminal (HMI)
- Engineering Network with a local Linux workstation running an SSH server
- Corporate LAN
- All networks are interconnected through firewalls



Step 2 – Model Systems

1. Expand the **PhysicalThing** node in the available components tree.
2. Drag and drop an instance of a **Network** onto the canvas.



Step 2 – Model Systems

1. Select the **Network1** component and click the Edit Details button.
2. Change the name to **EngrLAN** and click Finish.

MetaModel1

Components

- Data
- PhysicalThing
 - Device
 - Network
 - System

Ontology

Name

ckmsnapshot16...

System Goals Adv

EngrLAN Details

Component Details

Specify the details for the component.

Name:

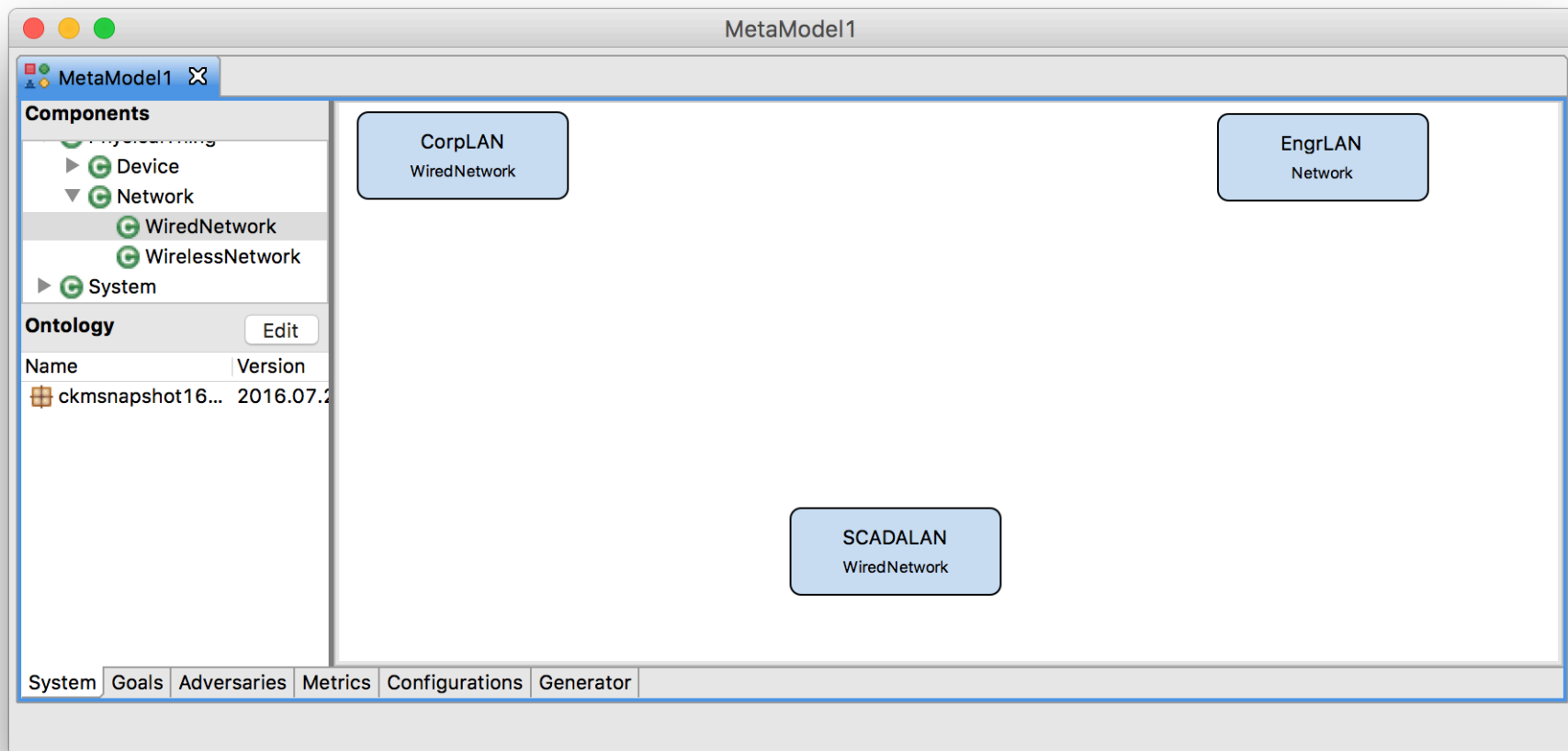
Attributes

eavesdropResponse	<input type="text" value="0"/>
limitedIncomingProtocols	<input type="text" value="4"/>
networkAnomalyDetectionStrength	<input type="text" value="2"/>
networkAnomalyResponseStrength	<input type="text" value="2"/>
networkEncryptionStrength	<input type="text" value="0"/>
networkWhiteList	<input type="text" value="0"/>
physicalAttackAttribution	<input type="text" value="2"/>
physicalNetworkProtection	<input type="text" value="4"/>
resistanceToKineticDamage	<input type="text" value="0"/>
rogueDeviceControl	<input type="text" value="2"/>
rogueDeviceDetection	<input type="text" value="3"/>
strengthOfUserAuthentication	<input type="text" value="4"/>
userAuthenticationType	<input type="text" value="S"/>

EngrLAN
Network
Add Relationship
Edit Details

Step 2 – Model Systems

1. Create a **WiredNetwork** called **CorpLAN**
2. Create a **WiredNetwork** called **SCADALAN**



Step 2 – Model Systems

1. Create a **FirewallHosted** and define its attributes like so:

Attribute	Value
Name	CorpLanScadaLanFW
strengthOfUserAuthentication	userAuthStrength
userAuthenticationType	userAuthType

The screenshot shows a dialog box titled "CorpLanScadaLanFW Details" with a "Component Details" section. The "Name" field is set to "CorpLanScadaLanFW". Below, a list of attributes is shown with corresponding values. The "strengthOfUserAuthentication" and "userAuthenticationType" attributes are highlighted with a red circle.

Attribute	Value
componentAnomalyDetectionStrength	0
credentialMonitoring	0
deviceStatusControl	1
deviceStatusDetection	2
firewallConfigControl	2
firewallConfigDetection	5
mediaPortEnabled	1
physicalAttackAttribution	2
resistanceToKineticDamage	0
resistanceToLogicalDisable	0
resistanceToPhysicalDisable	0
softwareTrustedSourceSecurity	5
softwareTrustedSourceSecurity	5
strengthOfUserAuthentication	userAuthStrength
userAuthenticationType	userAuthType
userCyberSecurityAwareness	3

Step 2 – Model Systems

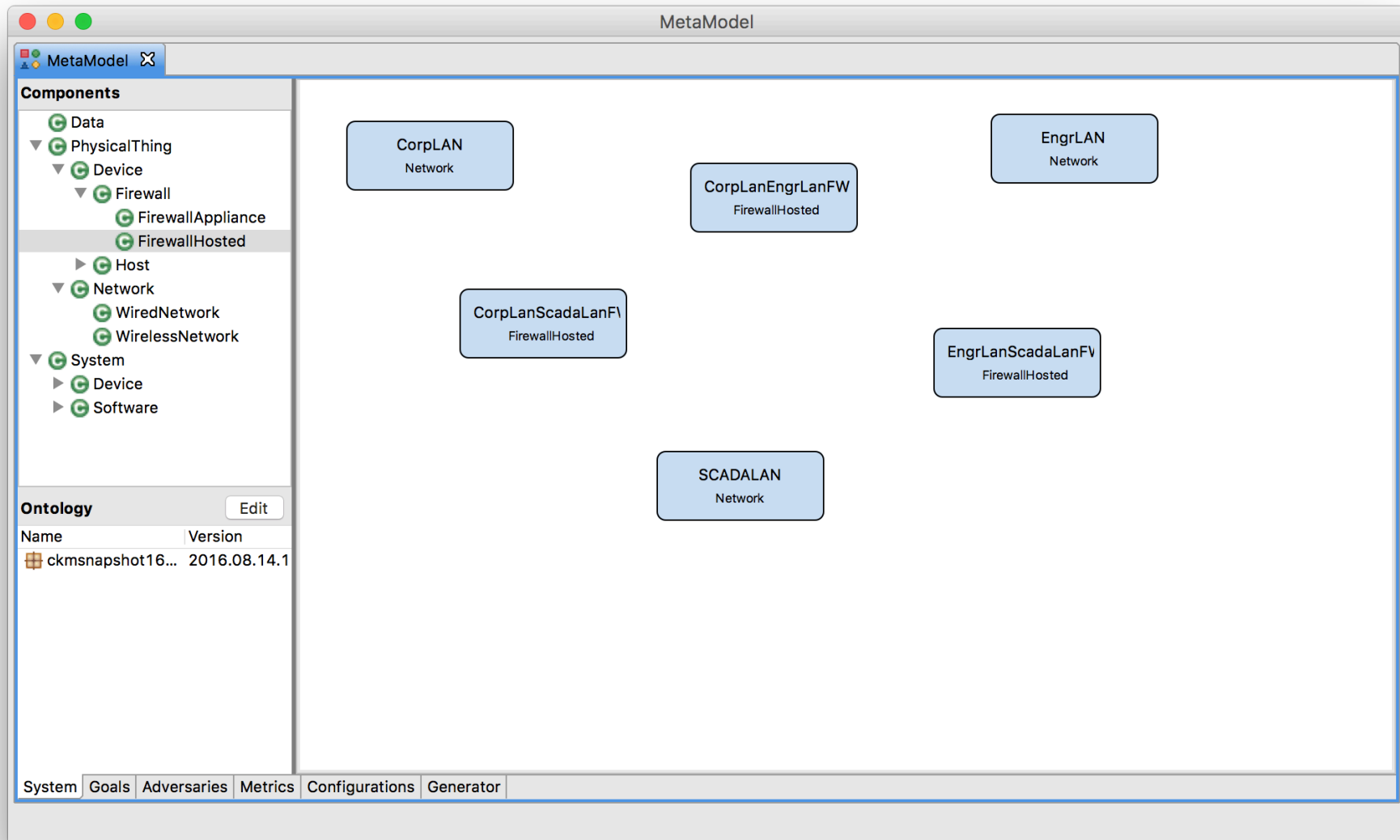
1. Create another **FirewallHosted**

Attribute	Value
Name	CorpLanEngrLanFW
strengthOfUserAuthentication	userAuthStrength
userAuthenticationType	userAuthType

2. Create another **FirewallHosted**

Attribute	Value
Name	EngrLanScadaLanFW
strengthOfUserAuthentication	userAuthStrength
userAuthenticationType	userAuthType

Step 2 – Model Systems



Step 2 – Model Systems

1. Select the **CorpLanEngrLanFW** component and click the Add Relationship button.
2. Select the **CorpLan** component.
3. Select the **onNetwork** relationship and click Finish.

The screenshot displays a software interface for modeling systems. The main window, titled "MetaModel1", shows a diagram with several components: "CorpLAN" (WiredNetwork), "CorpLanEngrLanFW" (FirewallHosted), "CorpLanScadaLanF" (FirewallHosted), and "SCADALAN" (WiredNetwork). A context menu is open over the "CorpLanEngrLanFW" component, with "Add Relationship" selected. An "Add Relationship" dialog box is open in the foreground, showing the source component as "CorpLanEngrLanFW", the target component as "CorpLAN", and the relationship type as "onNetwork". The dialog box has "Cancel" and "Finish" buttons at the bottom.

MetaModel1

Components

- PhysicalThing
 - Device
 - Firewall
 - FirewallAppliance
 - FirewallHosted

Ontology

Name	Version
ckmsnapshot16...	2016.07.2

System Goals Adversaries Metrics Configurations Generator

Add Relationship

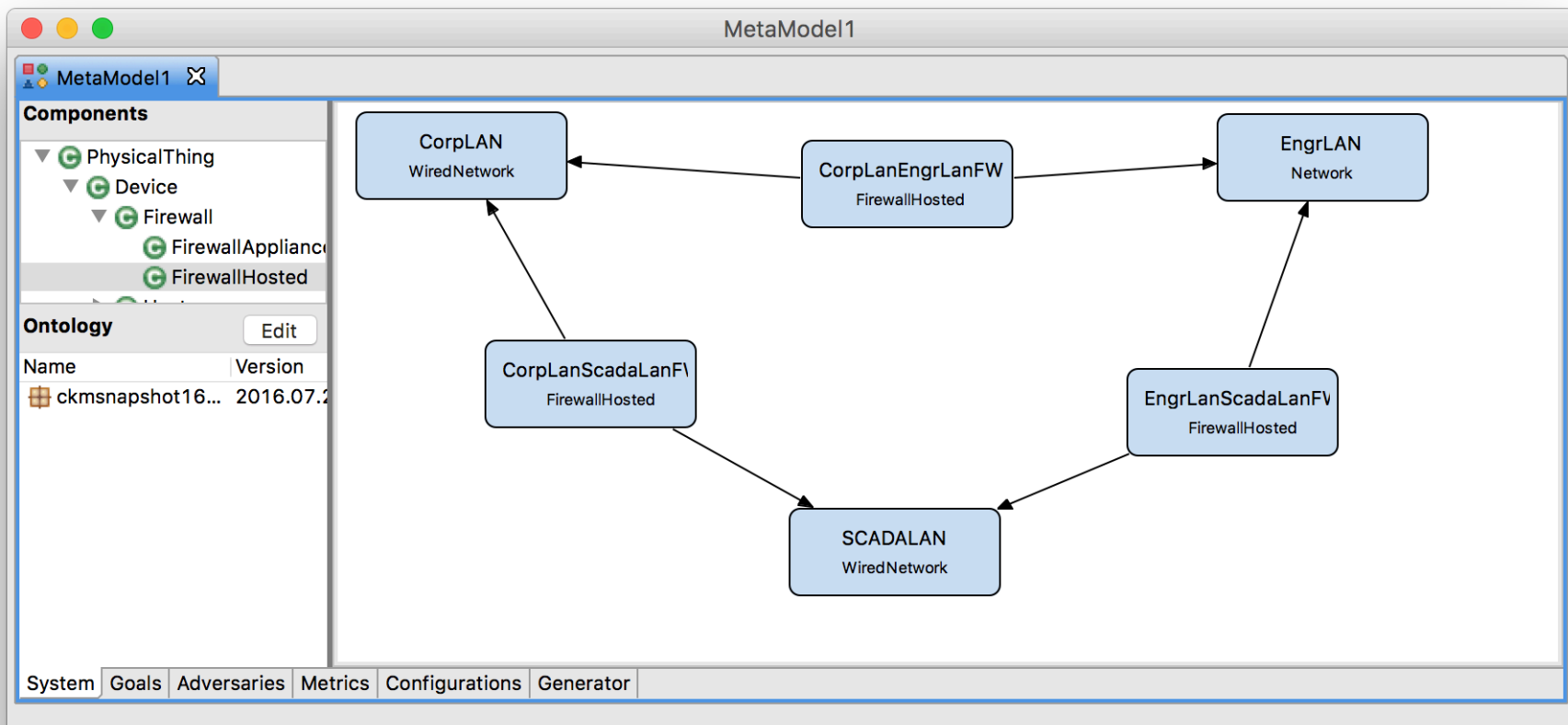
Select the relationship you wish to add between the source and target components.

Source: CorpLanEngrLanFW
Target: CorpLAN
Relationship: onNetwork

Cancel Finish

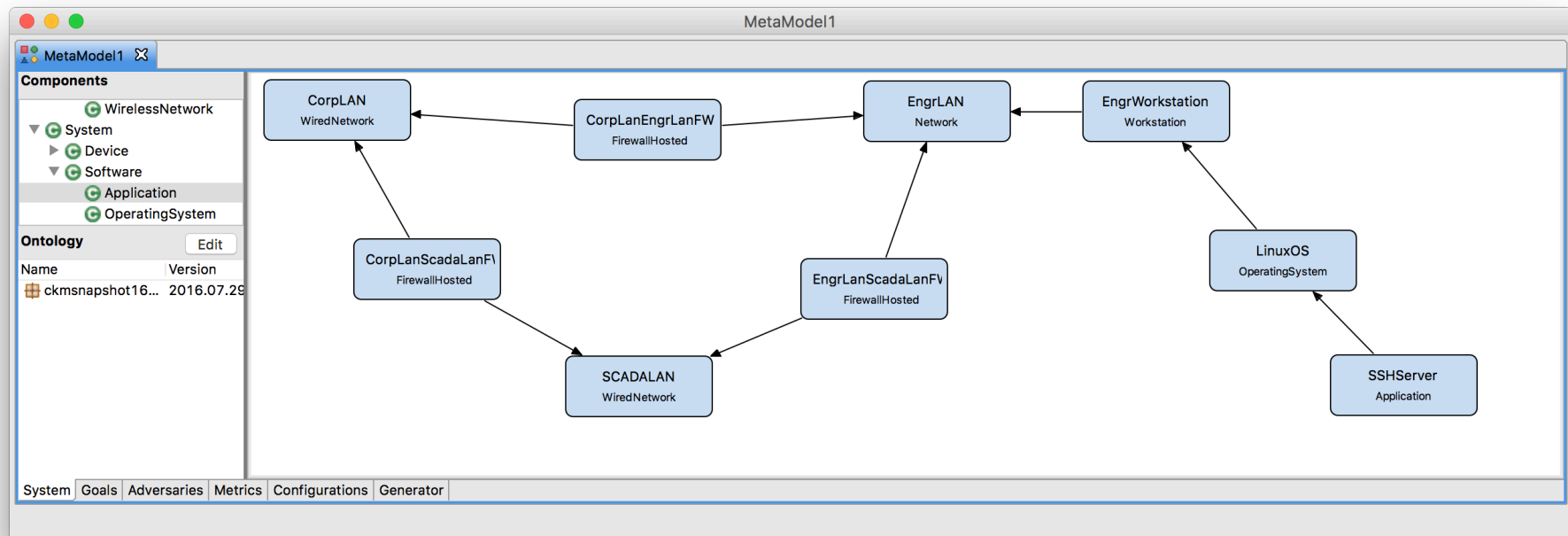
Step 2 – Model Systems

1. Create additional **onNetwork** relationships between:
 - **CorpLanEngrLanFW** to **EngrLAN**
 - **CorpLanScadaLanFW** to **CorpLAN**
 - **CorpLanScadaLanFW** to **SCADALAN**
 - **EngrLanScadaLanFW** to **EngrLAN**
 - **EngrLanScadaLanFW** to **SCADALAN**



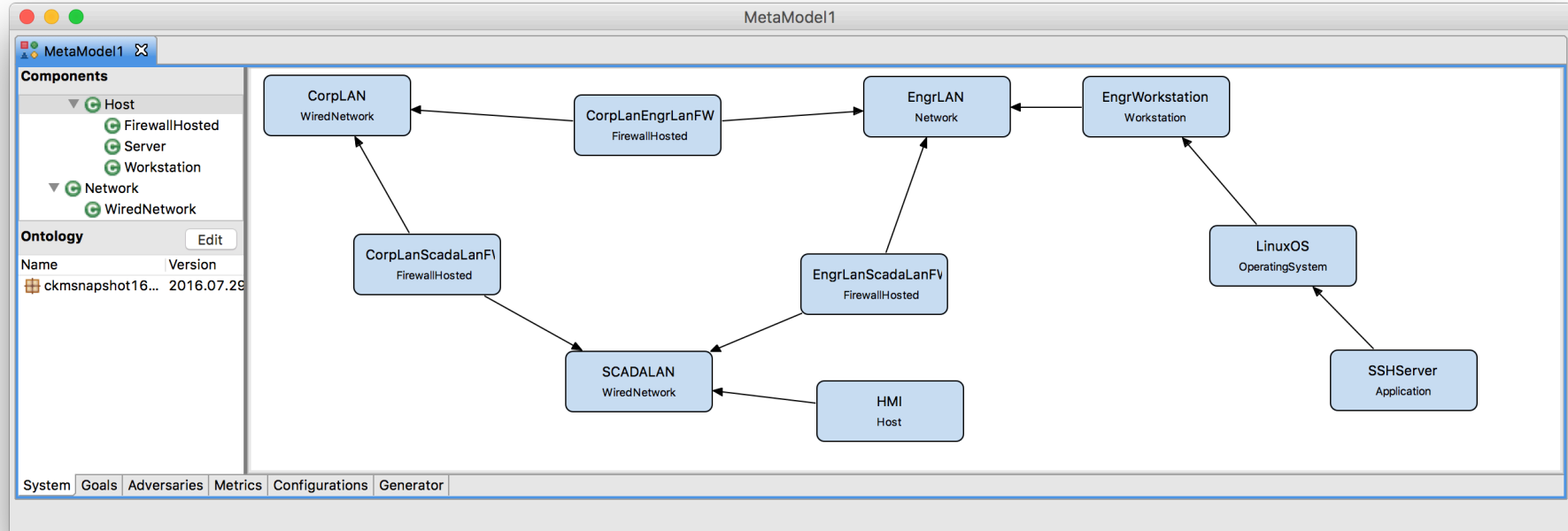
Step 2 – Model Systems

1. Create a **Workstation** called **EngrWorkstation**
2. Define an **onNetwork** relationship from **EngrWorkstation** to **EngrLAN**.
3. Create an **OperatingSystem** called **LinuxOS**.
4. Define a **hardwarePlatform** relationship from **LinuxOS** to **EngrWorkstation**.
5. Create an **Application** called **SSHServer**.
6. Define an **applicationOS** relationship from the **SSHServer** to **LinuxOS**.



Step 2 – Model Systems

1. Create a **Host** called **HMI**.
2. Define an **onNetwork** relationship from **HMI** to **SCADALAN**



Step 2 – Model Systems – Feedback

- How challenging was this step?
- Was adding components, creating relationships, and defining attributes easy?
- Could this part of the tool be useful for designing system diagrams for uses outside of the tool?
- How would you handle larger, more complex models? How would you expect the tool to help you with those models?
- Was the available components tree intuitive?

Step 3 – Attack Goals, Adversaries, Metrics, and Generation

- Possible attack goals are dependent on the system diagram
 - Choose a set of state variables (Access, Skill, Knowledge, SSV) the goal state is a function of.
 - Define the functional expression that indicates whether the goal has been achieved.

Small SCADA Networks – Step 3

```
return SCADALAN_NetworkAccess->Mark();
```

The screenshot shows the MetaModel software interface. On the left, the 'Available State Variables' tree is expanded to 'SCADALAN', where 'HasUserCredentials' is selected. Below it, the 'Ontology' table shows 'ckmsnapshot16...' with version '2016.08'. The main 'Goals' panel shows a goal named 'Goal_GainScadaNetworkAccess' with the expression 'return SCADALAN_NetworkAccess->Mark();'. Red arrows and text annotations indicate the steps: '1. Select' points to the goal name, '2. Enter name' points to the 'Name' field, '3. Drag and Drop' points to the 'HasUserCredentials' variable in the 'Dependent State Variables' list, and '4. Enter expression' points to the goal expression text area.

MetaModel

*MetaModel

Available State Variables

- Global State Variables
- CorpLAN
- CorpLanEngrLanFW
- CorpLanScadaLanFW
- EngrLAN
- EngrLanScadaLanFW
- EngrWorkstation
- HMI
- LinuxOS
- SCADALAN
 - Damaged
 - HasUserCredentials

Ontology

Name	Version
ckmsnapshot16...	2016.08

Goals

Name: Goal_GainScadaNetworkAccess

Dependent State Variables:

Name
SCADALAN_NetworkAccess

Goal Expression:

```
return SCADALAN_NetworkAccess->Mark();
```

System Goals Adversaries Metrics Configurations Generator

1. Select

2. Enter name

3. Drag and Drop

4. Enter expression

Step 3 – Attack Goals, Adversaries, Metrics, and Generation

- Adversaries are created from Adversary Templates (defined in the ontology)
 - Attributes are customizable
 - Possible initial state depends on system diagram

Small SCADA Networks – Step 3

1. Drag and Drop

2. Specify Name and Decision Parameters

3. Specify Initial Access

4. Specify Skills and Goals

MetaModel

Adversary Templates

- Customer
- EconomicCompetitorLimited
- EconomicCompetitorWellRes
- ForeignGovernmentLimitedI
- ForeignGovernmentWellRes
- HackerGroup
- IndependentInsider**
- OrganizedCrime
- TerroristOrganization

Adversaries Delete

Name:

Decision Parameters

Planning Horizon:

Cost of Detection:

Access

Name	Initial Value
SSHServer_HasUserCredentials	1
LinuxOS_HasUserCredentials	1
EngWorkstation_HasUserCredentials	1

Knowledge

Name	Initial Value
------	---------------

Skills

Name	Initial Value
BasicCyberOffense	1000
Cryptanalysis	200

Ontology Edit

Name	Version
ckmsnapshot...	2016.07.28.0

System Goals Adversaries Metrics Configurations Generator

Small SCADA Networks – Step 3

1. Independent Insider
2. Name: EngineerInsider
 Cost of Detection: 100,000
3. Access: CorpLAN_PhysicalAccess
 CorpLanEngrLanFW_PhysicalAccess
 CorpLanScadaLanFW_PhysicalAccess
 EngrLAN_NetworkAccess
 EngrLAN_PhysicalAccess
 EngrLanScadaLanFW_PhysicalAccess
 EngrWorkstation_HasUserCredentials
 EngrWorkstation_LogicalAccess
 EngrWorkstation_PhysicalAccess
 EngrWorkstation_UIAccess
 HMI_PhysicalAccess
 LinuxOS_HasUserCredentials
 LinuxOS_LogicalAccess
 LinuxOS_UIAccess
 SCADALAN_PhysicalAccess
 SSHServer_HasUserCredentials
 SSHServer_LogicalAccess
 SSHServer_UIAccess
4. Goal: Goal_GainNetworkAccessOnScadaNetwork 50,000

Step 3 – Attack Goals, Adversaries, Metrics, and Generation

- Metrics are created from Metric Templates (defined in the ontology)
 - Input parameters are defined in the ontology
 - Input values specified by the user



Small SCADA Networks – Step 3

The screenshot shows the MetaModel application interface. On the left, the 'Metric Templates' pane contains a 'Goal Achieved' template. A red arrow points from this template to the 'Metrics' pane, labeled '1. Click and drag'. In the 'Metrics' pane, the 'SCADANetworkCompromised' metric is selected, and a red arrow points to its name field, labeled '2. Rename'. Below the name field, the 'Goal' dropdown menu is set to 'Goal_GainScadaNetworkAccess', with a red arrow pointing to it labeled '3. Select'. Further down, the 'Upper Bound' field is set to '24', with a red arrow pointing to it labeled '4. Enter timing'. The 'Step Size' field is set to '2'. At the bottom, a navigation bar shows 'System', 'Goals', 'Adversaries', 'Metrics', 'Configurations', and 'Generator' tabs.

MetaModel

MetaModel

Metric Templates

- Goal Achieved

Metrics Delete

Name: SCADANetworkCompromised

Enter the goal you wish to observe.

Goal: Goal_GainScadaNetworkAccess

Enter the first observation time point.

First Observation: 0

Enter the upper bound on observation times.

Upper Bound: 24

Enter the length of time between observations.

Step Size: 2

Ontology Edit

Name	Version
ckmsnapshot16...	2016.08.16.0

System Goals Adversaries Metrics Configurations Generator

Step 3 – Attack Goals, Adversaries, Metrics, and Generation

- Configurations bring together a goal set, an adversary, and a metric set.
 - Multiple configurations will generate as one model
 - Each configuration will be an experiment
 - Simulator will run each experiment separately and report results

Small SCADA Networks – Step 3

MetaModel

MetaModel

Configur: Add Delete

Name

- StrongFirewalls
- WeakFirewalls

Name: StrongFirewalls **1. Enter name**

Description

Goals

Name

- Goal_GainScadaNetworkAccess **2. Select goals**

Adversary: EngineerInsider **3. Select adversary**

Metrics

Name

- SCADANetworkCompromised **4. Select metrics**

Global Variables

Name	Type	Value
userAuthStrength	Short	8
userAuthType	Character	'T'

5. Define GV values

System Goals Adversaries Metrics Configurations Generator



Small SCADA Networks – Step 3

The screenshot shows the MetaModel application window. On the left, a sidebar lists configurations: 'StrongFirewalls' and 'WeakFirewalls' (selected). The main area displays the configuration for 'WeakFirewalls' with the following details:

- Name:** WeakFirewalls
- Description:** (empty)
- Goals:** Includes 'Goal_GainScadaNetworkAccess'.
- Adversary:** EngineerInsider
- Metrics:** Includes 'SCADANetworkCompromised'.
- Global Variables:** A table with the following data:

Name	Type	Value
userAuthStrength	Short	1
userAuthType	Character	'W'

At the bottom of the window, a navigation bar contains the following tabs: System, Goals, Adversaries, Metrics, Configurations, and Generator.

Step 3 – Attack Goals, Adversaries, Metrics, and Generation

- The generator creates multiple components in the Mobius Modeling Tool and begins simulation.
 - The components are an ADVISE model, a Performance Variables model, a Set study, and a Discrete Event Simulator.
 - Simulation should take about a minute.
 - Numerical results are show once completed.



Small SCADA Networks – Step 3

The screenshot shows the MetaModel application window. The 'Destination Project' is set to 'ADVISEMetaTutorial'. Under 'Configurations', 'StrongFirewalls' and 'WeakFirewalls' are selected. The 'New Component Name' is 'newADVISEModel'. Under 'Generate Components', all options are checked: ADVISE Atomic Model, Performance Variables Model, Set Study, Simulator, and Start Simulation. The 'Generate' button is highlighted with a red arrow. The 'Status' pane on the right shows a list of completed tasks with their respective durations.

1. Select both

2. Check all

3. Click generate

Destination Project: ADVISEMetaTutorial Select...

Configurations:

Name

- StrongFirewalls
- WeakFirewalls

New Component Name: newADVISEModel

Generate Components

- ADVISE Atomic Model
- Performance Variables Model
- Set Study
- Simulator
- Start Simulation

Generate Stop

Status

Constructing complete attack execution graph... Complete! (0.009sec)
Trimming AEG for configurations...Complete! (6.318sec)
New ADVISE model: newADVISEModel
Creating Reward Model...Complete! (2.731sec)
Creating Study...Complete! (1.391sec)
Creating Simulator...Complete! (0.113sec)
Starting Simulator...Complete! (1.907sec)

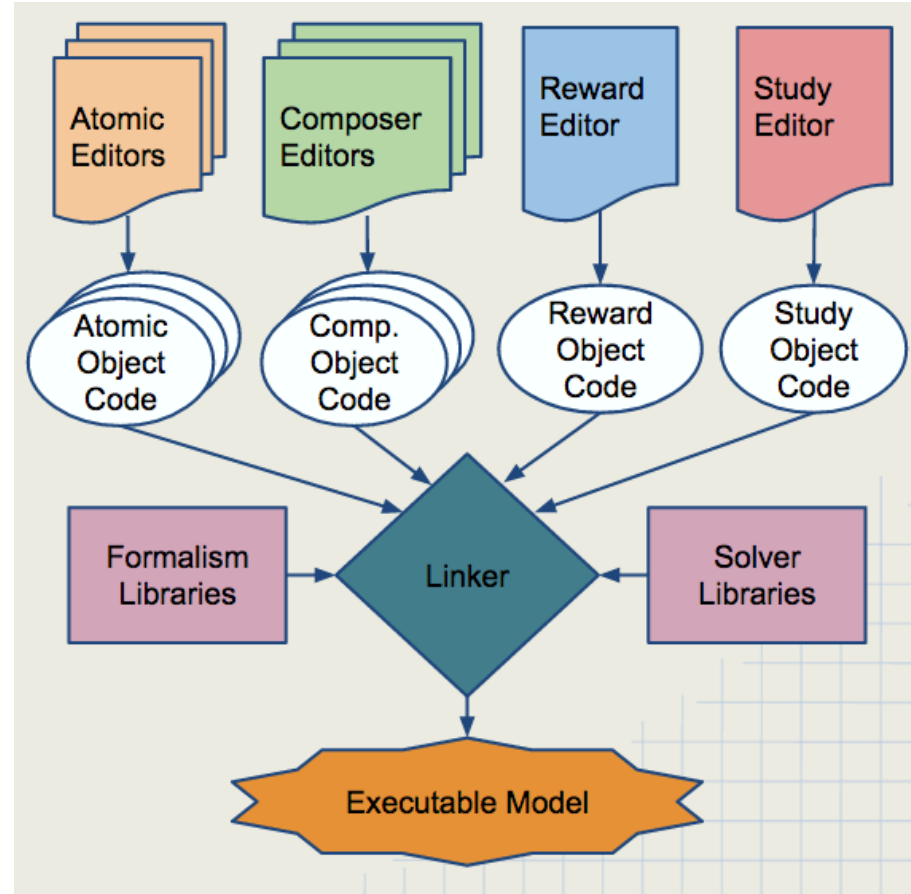
System Goals Adversaries Metrics Configurations Generator

Step 3 – Attack Goals, Adversaries, and Generation – Feedback

- How challenging was this step?
- Did the goal definition seem intuitive?
- What other adversary templates would you look for?
- Does the configuration of an adversary's attributes make sense?

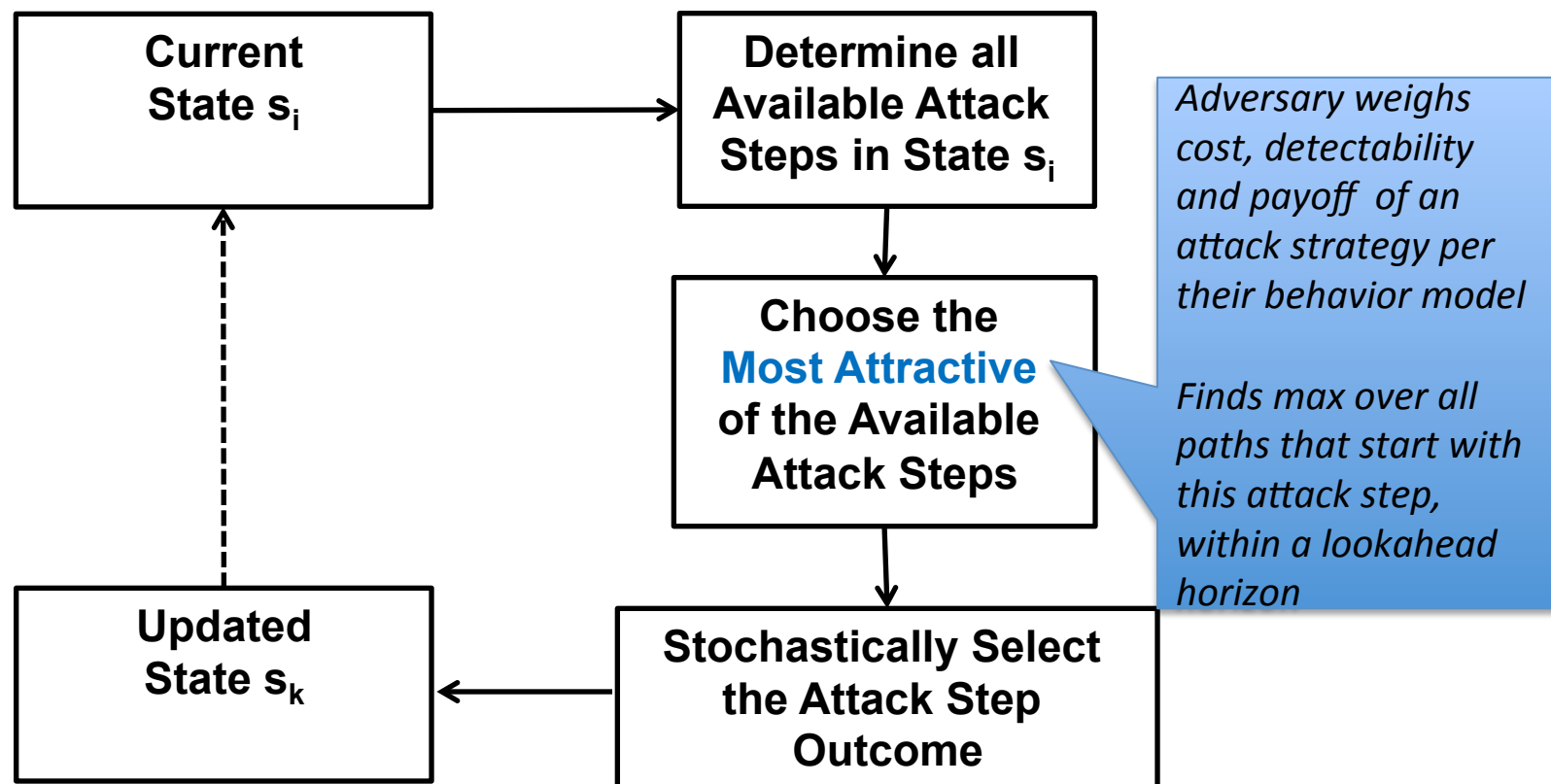
Step 4 – Execute Models

- Möbius creates an executable model by:
 - Generating C++ code representations of project models
 - Compiling the code and linking formalism and solver libraries
 - Executing the binary to gather observations and calculate statistics



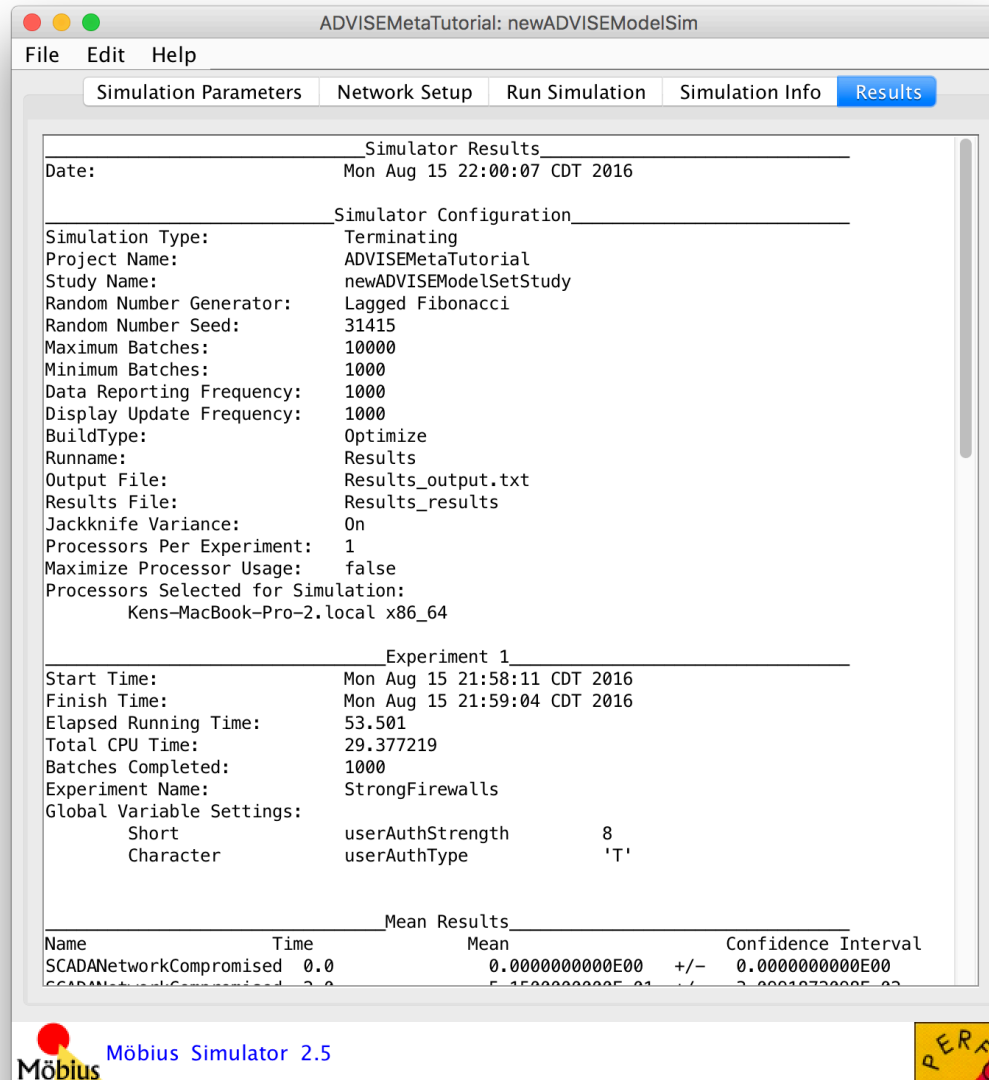
Step 4 Execute Models - Adversary Decision Cycle

- Adversary selects most attractive available attack step in AEG, repeats
- State transitions determined by outcome of selected attack step



Small SCADA Networks – Step 4

Simulation Is Complete



ADVISEMetaTutorial: newADVISEModelSim

File Edit Help

Simulation Parameters Network Setup Run Simulation Simulation Info **Results**

Simulator Results

Date: Mon Aug 15 22:00:07 CDT 2016

Simulator Configuration

Simulation Type: Terminating
 Project Name: ADVISEMetaTutorial
 Study Name: newADVISEModelSetStudy
 Random Number Generator: Lagged Fibonacci
 Random Number Seed: 31415
 Maximum Batches: 10000
 Minimum Batches: 1000
 Data Reporting Frequency: 1000
 Display Update Frequency: 1000
 BuildType: Optimize
 Runname: Results
 Output File: Results_output.txt
 Results File: Results_results
 Jackknife Variance: 0n
 Processors Per Experiment: 1
 Maximize Processor Usage: false
 Processors Selected for Simulation:
 Kens-MacBook-Pro-2.local x86_64

Experiment 1

Start Time: Mon Aug 15 21:58:11 CDT 2016
 Finish Time: Mon Aug 15 21:59:04 CDT 2016
 Elapsed Running Time: 53.501
 Total CPU Time: 29.377219
 Batches Completed: 1000
 Experiment Name: StrongFirewalls
 Global Variable Settings:
 Short userAuthStrength 8
 Character userAuthType 'T'

Mean Results

Name	Time	Mean	Confidence Interval
SCADANetworkCompromised	0.0	0.0000000000E00	+/- 0.0000000000E00
SCADANetworkCompromised	2.0	5.1500000000E-01	+/- 2.0000000000E-02

Möbius Simulator 2.5

PERT

Step 5 – Interpret Results

- We will examine...
 - Numerical results from the simulation
 - A visual presentation of the model's behavior

Small SCADA Networks – Step 5

Numerical Results

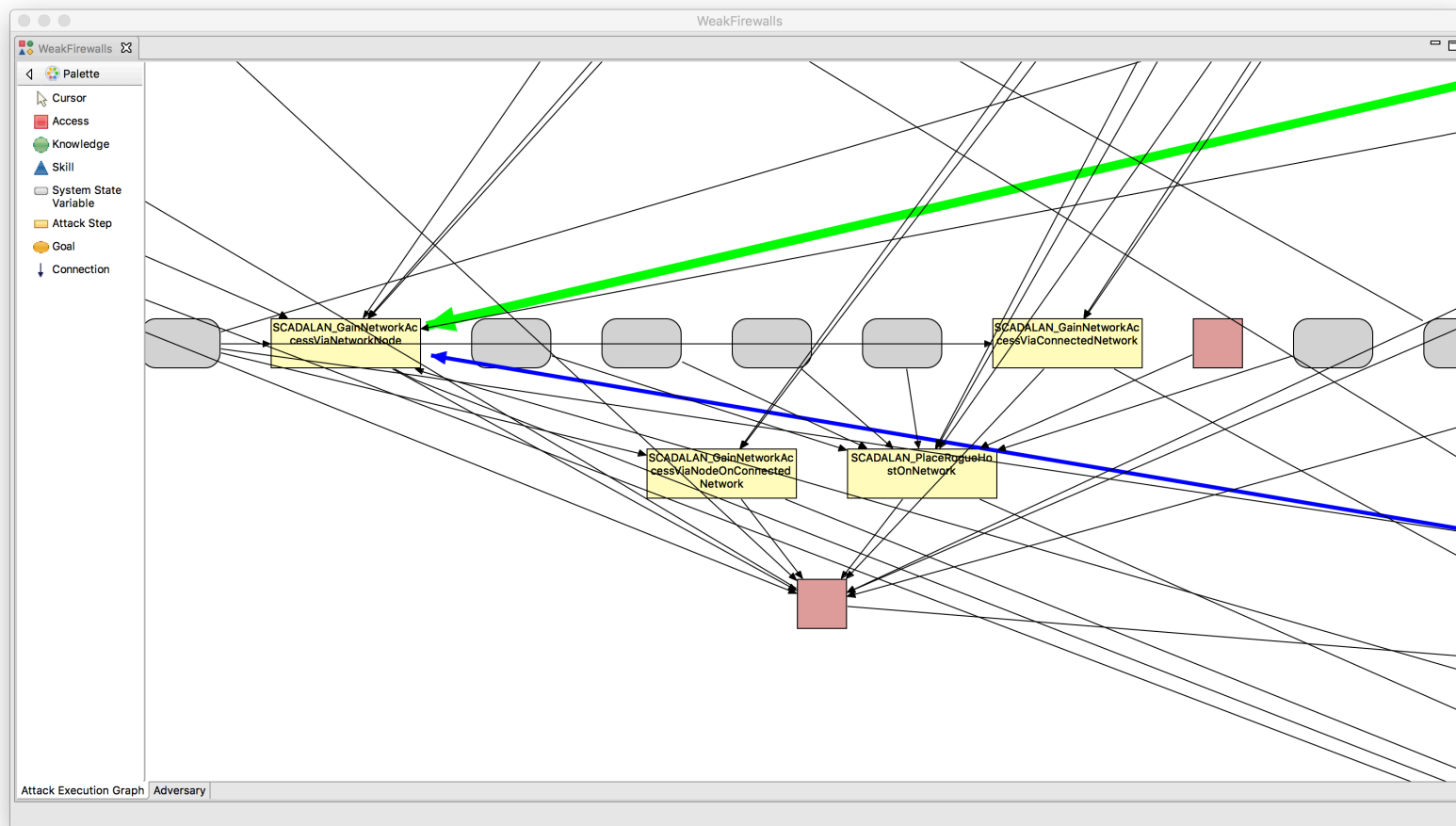
- Adversary was more successful, more quickly when firewalls were hardened.

Experiment Name:		WeakFirewalls		
Global Variable Settings:				
Short	Character	userAuthStrength	1	
		userAuthType	'W'	
Mean Results				
Name	Time	Mean		Confidence Interval
SCADANetworkCompromised	0.0	0.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	2.0	0.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	4.0	3.000000000E-03	+/-	1.0719782991E-03 (*)
SCADANetworkCompromised	6.0	3.630000000E-02	+/-	3.6660807717E-03 (*)
SCADANetworkCompromised	8.0	1.629000000E-01	+/-	7.2381403552E-03
SCADANetworkCompromised	10.0	3.483000000E-01	+/-	9.3385271315E-03
SCADANetworkCompromised	12.0	4.920000000E-01	+/-	9.7992354937E-03
SCADANetworkCompromised	14.0	5.829000000E-01	+/-	9.6648453634E-03
SCADANetworkCompromised	16.0	6.623000000E-01	+/-	9.2698070620E-03
SCADANetworkCompromised	18.0	7.331000000E-01	+/-	8.6702911980E-03
SCADANetworkCompromised	20.0	7.863000000E-01	+/-	8.0347836302E-03
SCADANetworkCompromised	22.0	8.266000000E-01	+/-	7.4207881630E-03
Experiment Name:		StrongFirewalls		
Global Variable Settings:				
Short	Character	userAuthStrength	8	
		userAuthType	'T'	
Mean Results				
Name	Time	Mean		Confidence Interval
SCADANetworkCompromised	0.0	0.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	2.0	5.150000000E-01	+/-	3.0991872098E-02
SCADANetworkCompromised	4.0	9.700000000E-01	+/-	1.0578396025E-02
SCADANetworkCompromised	6.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	8.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	10.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	12.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	14.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	16.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	18.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	20.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	22.0	1.000000000E00	+/-	0.000000000E00
SCADANetworkCompromised	24.0	1.000000000E00	+/-	0.000000000E00

Small SCADA Networks – Step 5

Visual Results

- Adversary chose to directly compromise the HMI, rather than go through the firewalls when the firewalls were hardened.



Step 5 – Interpret Results – Feedback

- Were you surprised by the results?
- Do you believe unexpected results could be useful?
- What more would you like to know about the model to make design decisions based on what you've learned?
- How could the results presentation be improved?

Agenda

- Registration and Continental Breakfast
- Welcome
- Goals
 - Tool
 - Workshop
- Steps to Use ADVISE Meta
- Hands on Sessions
- Case Studies and Custom Ontologies
- Wrap Up

Attack Step Details: Example Probability Calculation

NetworkEavesdrop
 (data D that transits network N)

Probability of adversary failure

Equals

Coded in the Ontology tab
 ->NetworkEavesdrop ->Failure

Probability can't break crypto
 (if any)

OR

Probability do break crypto but
 are kicked off network before D
 is captured

Increases
 with

Decreases
 with

Increases
 with

Application layer
 encryption
 strength
 (attribute of
 data)

Network layer
 encryption
 strength
 (attribute of
 network)

Cryptanalysis
 skill
 proficiency of
 adversary
 (adversary
 parameter)

Strength of network
 countermeasures to
 detect and respond
 to eavesdropping
 (attribute of network,
 defaulted to zero,
 since this is very
 difficult)










Skill Details: What does skill proficiency mean?

Basic cyber offense	This is a set of skills not further distinguished, some level of which are available via relatively inexpensive tools to any adversary. These include the following elements from the Ethical Hacking Certification Syllabus at https://www.eccouncil.org/Certification/professional-series/ceh-course-outline : scanning, enumeration, phishing attack, password cracking based on external information (guessing, replay), privilege escalation, hijacking web servers, hacking web applications, SQL injection, buffer overflow, straightforward DoS attacks, network sniffing, social engineering without human contact (for which phishing is an example).	1000	Lead individual employed by a nation state to conduct cyber attacks
		800	Individual with broad skills including stealth, which would earn admirers in the hacking community
		600	Individual with solid skills that could be employed to perform ethical hacking engagements
		400	Individual with solid skills in many areas listed, but weak in a few
		50	Individual can perform simple script-based attacks

In the base ontology, probability of success/failure of an attack step often has a linear relationship with one or two skills

- e.g. a generic skill OR a specialized skill impacts outcome

Custom Ontologies

- The base ontology is data, it is not baked into the tool
- “Library designer” (tool distributor or user) may on the ontology tab:
 - Add to or modify base ontology
 - Define a new custom ontology
- This includes all categories of ontology elements:
 - Types of components 
 - Attributes of components by type 
 - Relationships between components 
 - Types of access,  skills,  and knowledge  an adversary may have
 - Types and characteristics of adversaries 
 - Attack steps 
 - State variables: other system, component, or adversary properties 

Features for Future Base Ontology

- Component ontology
 - Types of networks (LAN, WAN, VLAN)
 - VPN connections
 - Routers
 - Gateways
 - Device authentication
 - User roles

Features for Future Base Ontology (cont.)

- Attack steps
 - Disable or Damage
 - PhysicalDisconnect
 - NetworkFlood
 - Botnet
 - Malware
 - CreateRemovableMediaCauseMalwareInstall
 - Stage PackageCauseMalwareInstall
 - InstallMalwareFromRemovableMedia
 - InstallMalwareFromFixedMedia
 - Impact of installed malware
 - Data Confidentiality
 - Exfiltrate data
 - Network Infrastructure
 - Router and switch attacks
 - 0 - days

Examples for Custom Ontologies

- Add component types with unique defaults, attributes, and/or attack steps
 - Virtual OS – build model of data center
 - ATM machine – build model of banking organization
 - Smartmeter – build model of planned smart grid architecture
- Add customized adversary(e.g. contractor with specific types of access)
- Modify formulas used to calculate attack step characteristics
 - Probability of success/failure of attack step
 - Detectability of an attack step outcome
 - Cost of attack step
 - Time to execute attack step
- Build ontology to model internal architecture of a modern electric vehicle together with associated charging stations
- See tutorial to try out creating an ontology
https://www.mobius.illinois.edu/wiki/index.php/ADVISE_Meta_Two_Nets_Tutorial

River Zonal Dispatcher Case Study

Purpose of analysis

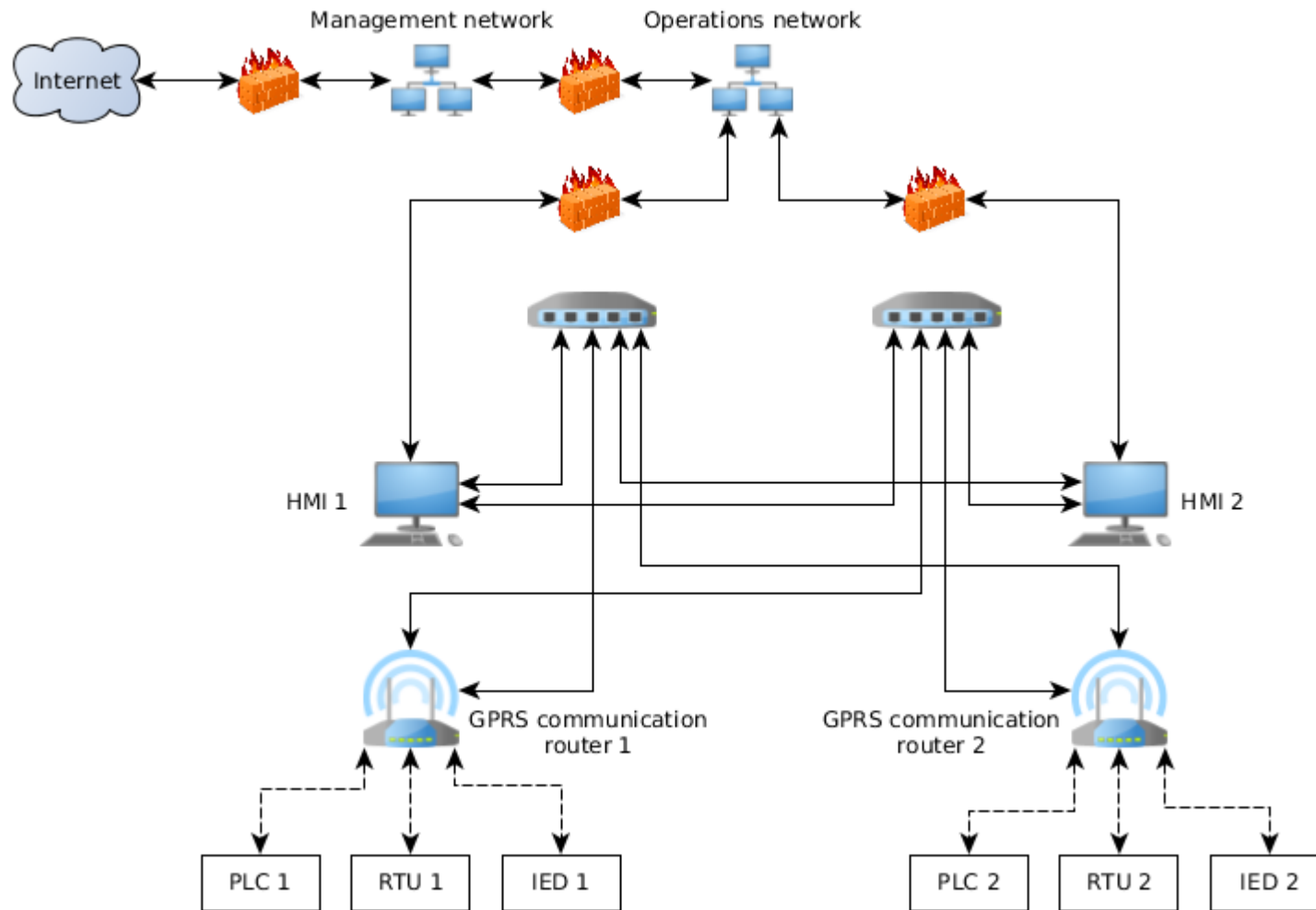
- Investigate the effects of architectural changes on system security
- Analyze the security impact of intrusion detection systems (IDSes) and isolation
- Analyze the security impact of multiple subsystems

River Zonal Dispatcher Case Study



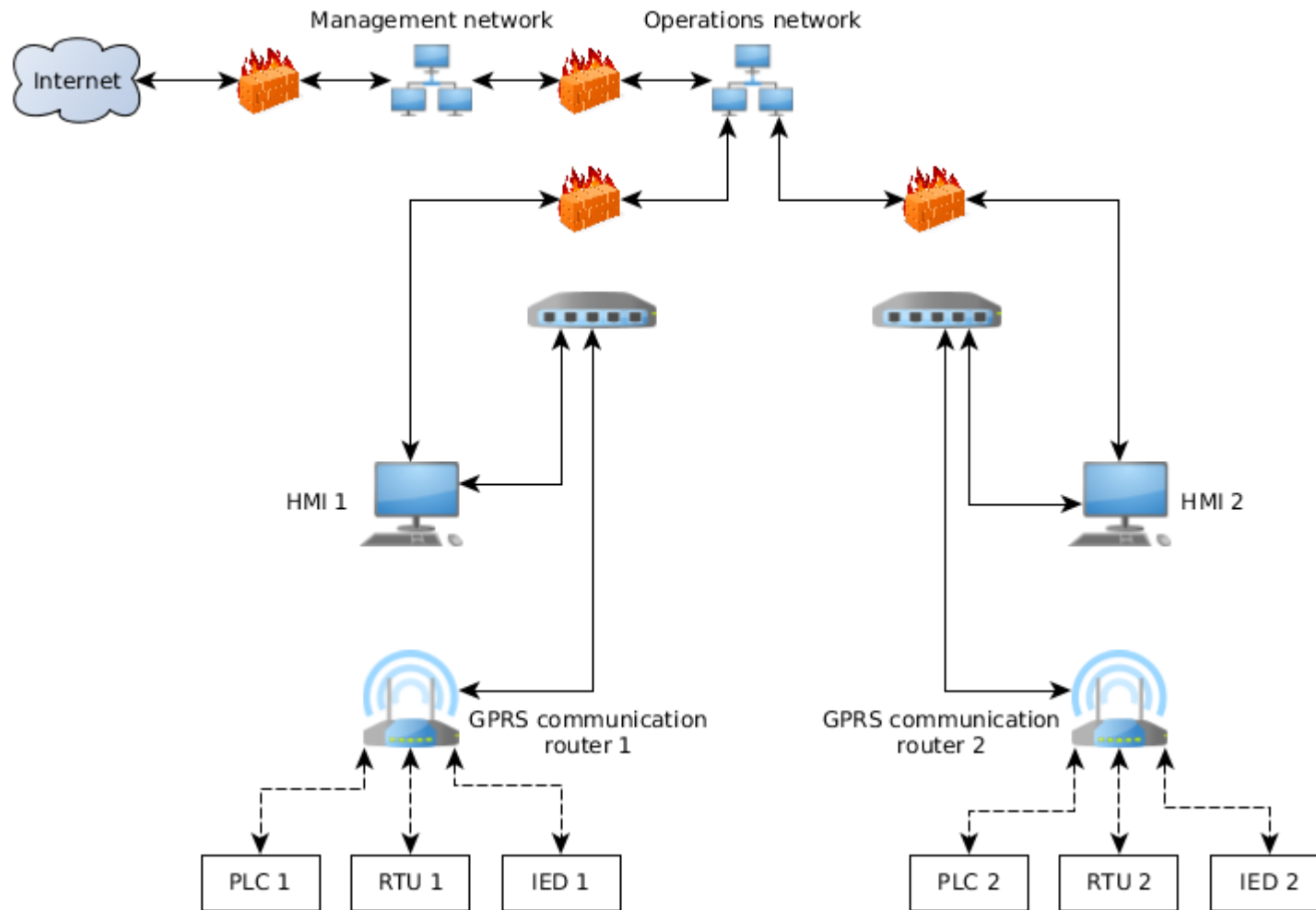
River Zonal Dispatcher Case Study

Without isolation

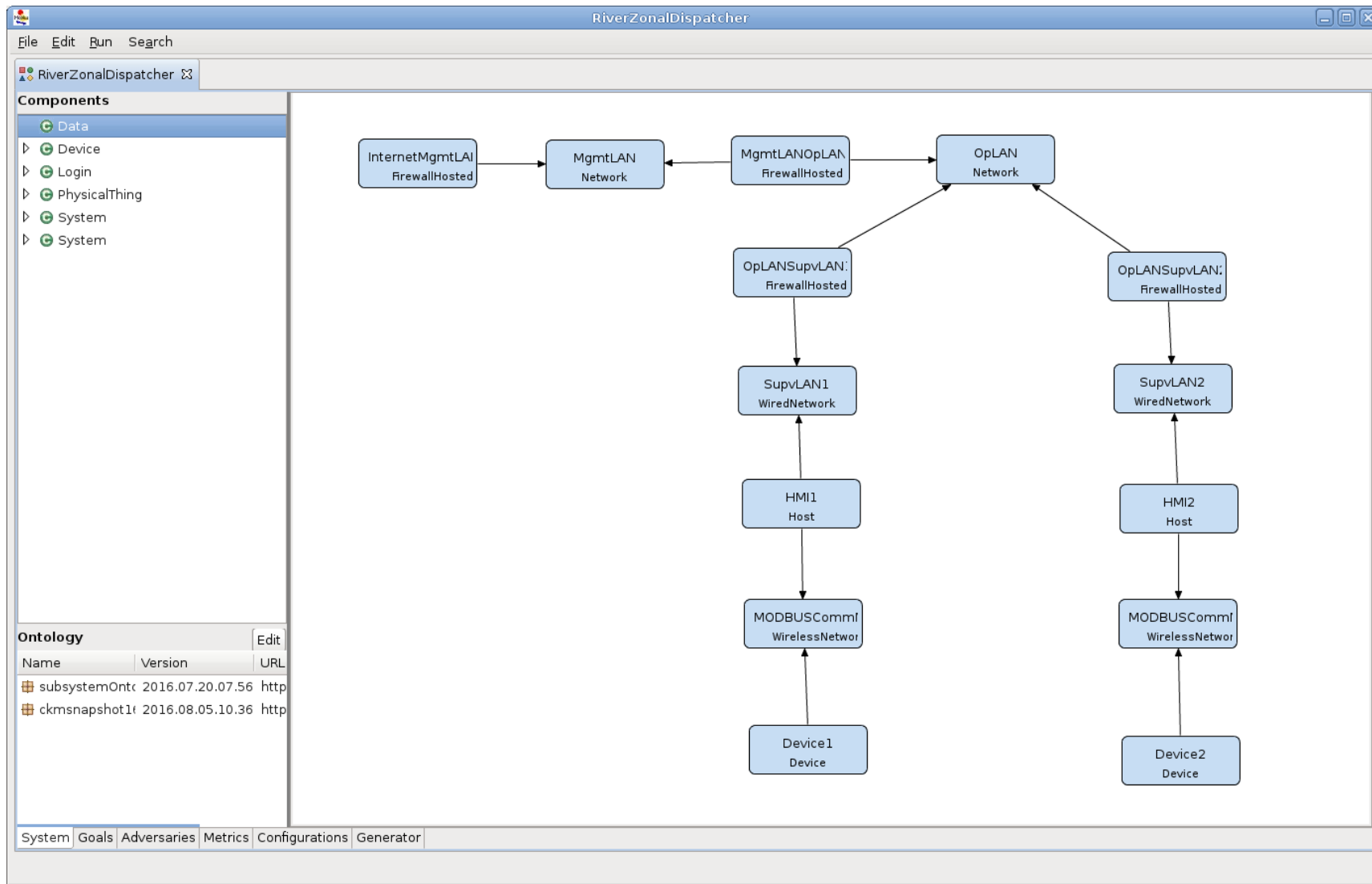


River Zonal Dispatcher Case Study

With isolation



River Zonal Dispatcher Case Study



River Zonal Dispatcher Case Study

Model attack goals

- Install malware on HMI
- Compromise system via router
- Compromise system via devices

Five adversaries are

- Foreign government
- Hacker
- Hostile Organization
- Insider Engineer
- Insider Operator

River Zonal Dispatcher Case Study

Select metrics

- Average Number of Successful Attacks
- Probability of Attack Goals Achieved at End Time
- Average Time-To-Achieve-Goal

River Zonal Dispatcher Case Study

Generate and execute models

- Set up 20 configurations for execution
 - Each of 5 adversaries X 4 system models
 - Calculate all metrics
- Simulation models adversary behavior over 1 year
- Ran 1,000 to 10,000 iterations

River Zonal Dispatcher Case Study

Results from hand-built model

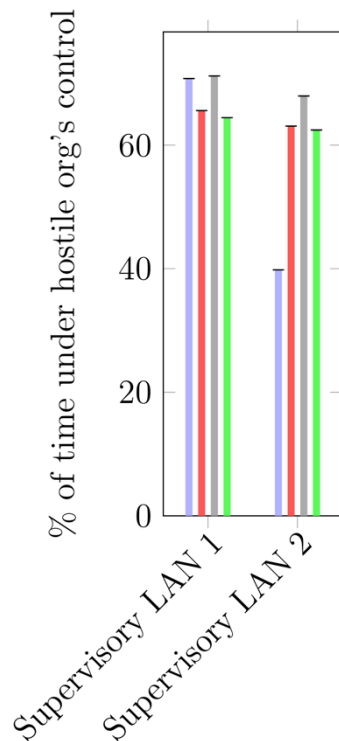


Figure 1: Average percentages of time in which the hostile organization has control of an on-site device on a particular supervisory network.

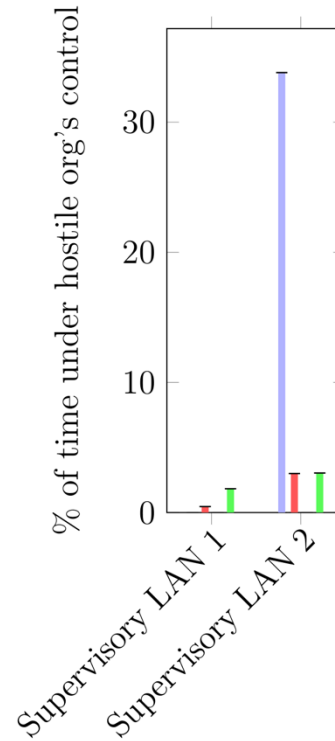


Figure 2: Average percentages of time in which the hostile organization has control of a GPRS communication router on a particular supervisory network.

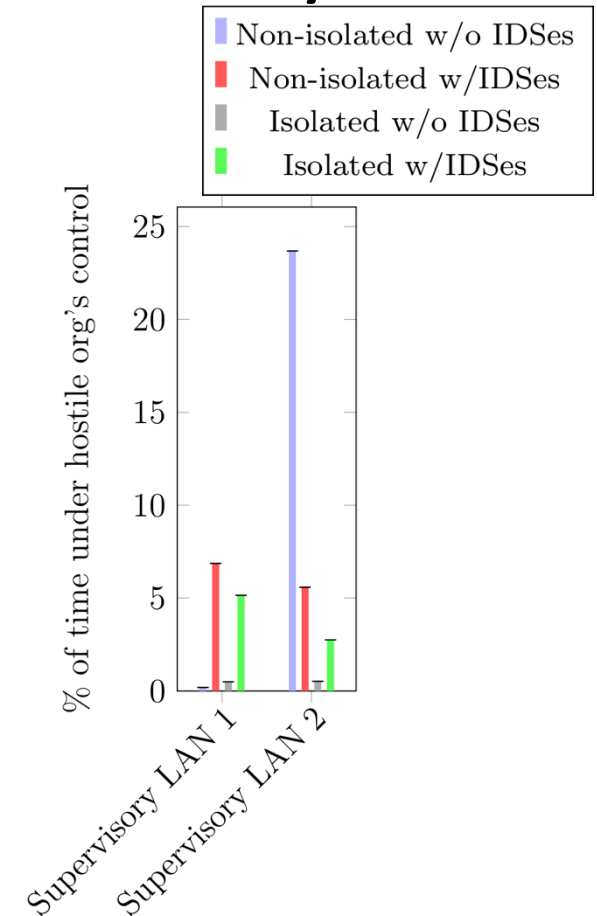


Figure 3: Average percentages of time that an HMI on a particular network has backdoor software installed by a hostile organization.

River Zonal Dispatcher Case Study

Results from hand-built model

- Greatest value seen in device attacks
- Neither IDSes nor isolation effective in minimizing device attacks
- IDSes very effective in minimizing router attacks and backdoor software installation on LAN 2 with non-isolated HMIs
- Isolation effective in minimizing router attacks and backdoor software installation overall
- LAN 2 preferred for router attacks and backdoor software installation, despite higher payoffs for attacks on LAN 1

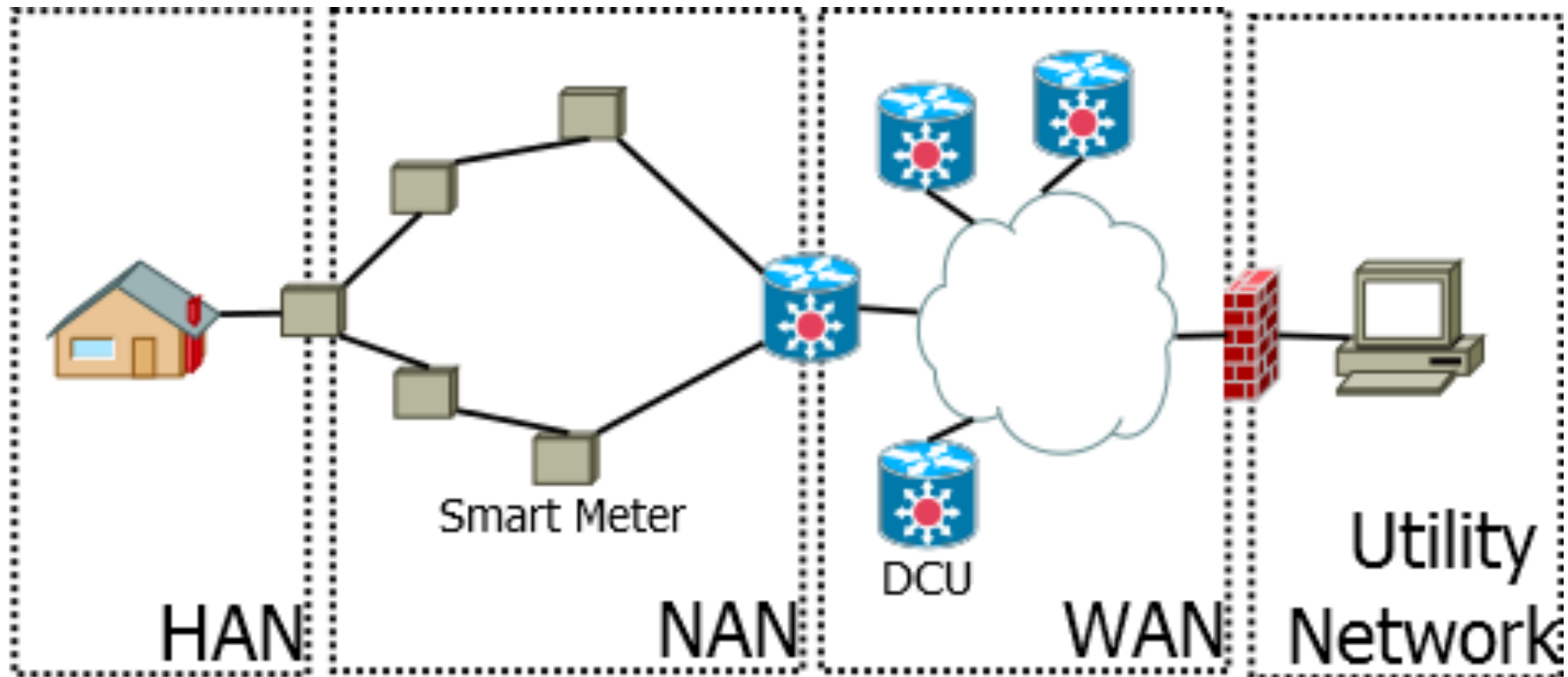
Advanced Metering Infrastructure Case Study

Define purpose of analysis

- Determine the cost-effectiveness of different intrusion detection systems (IDSes) in an Advanced Metering Infrastructure (AMI) network.
- In particular, compare
 - Centralized IDS,
 - Distributed IDS, and
 - Embedded IDS.

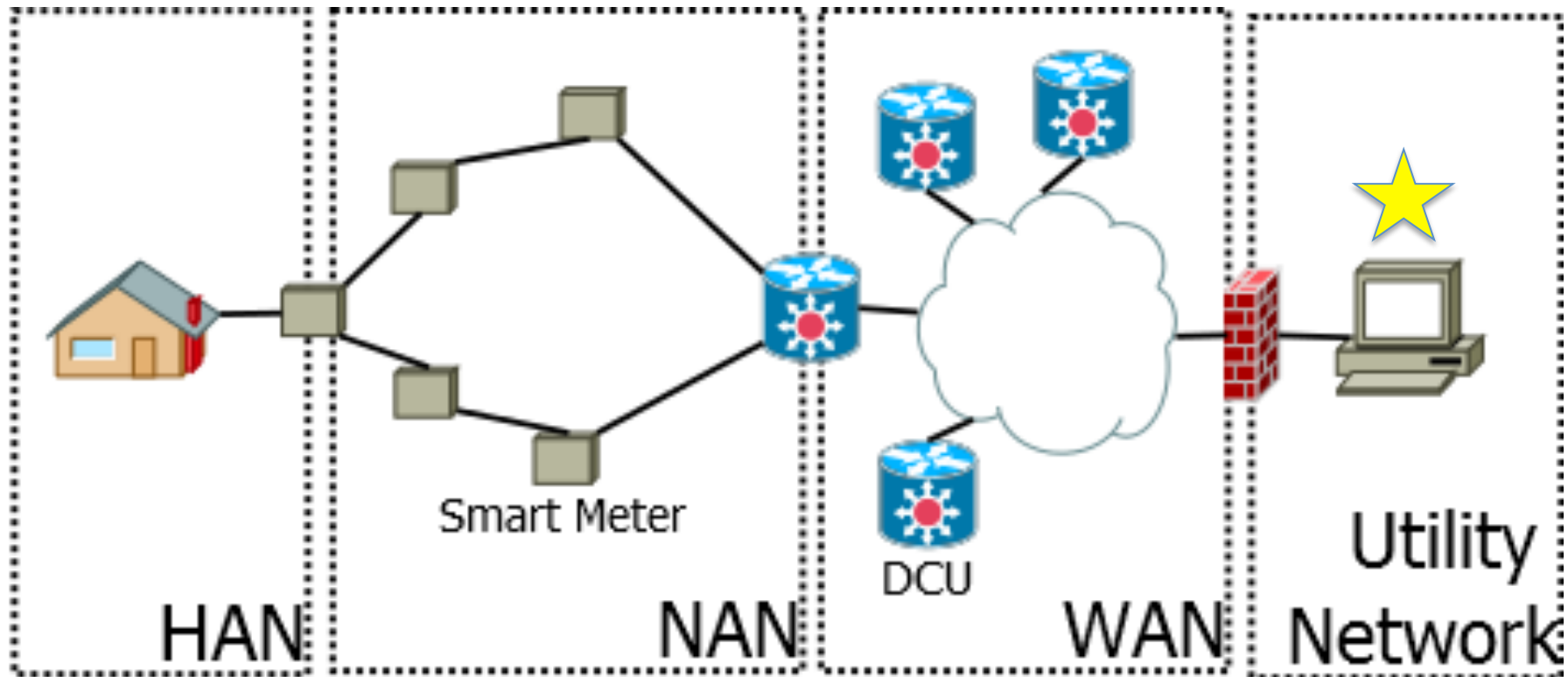
Advanced Metering Infrastructure Case Study

Define system components, relationship and attributes



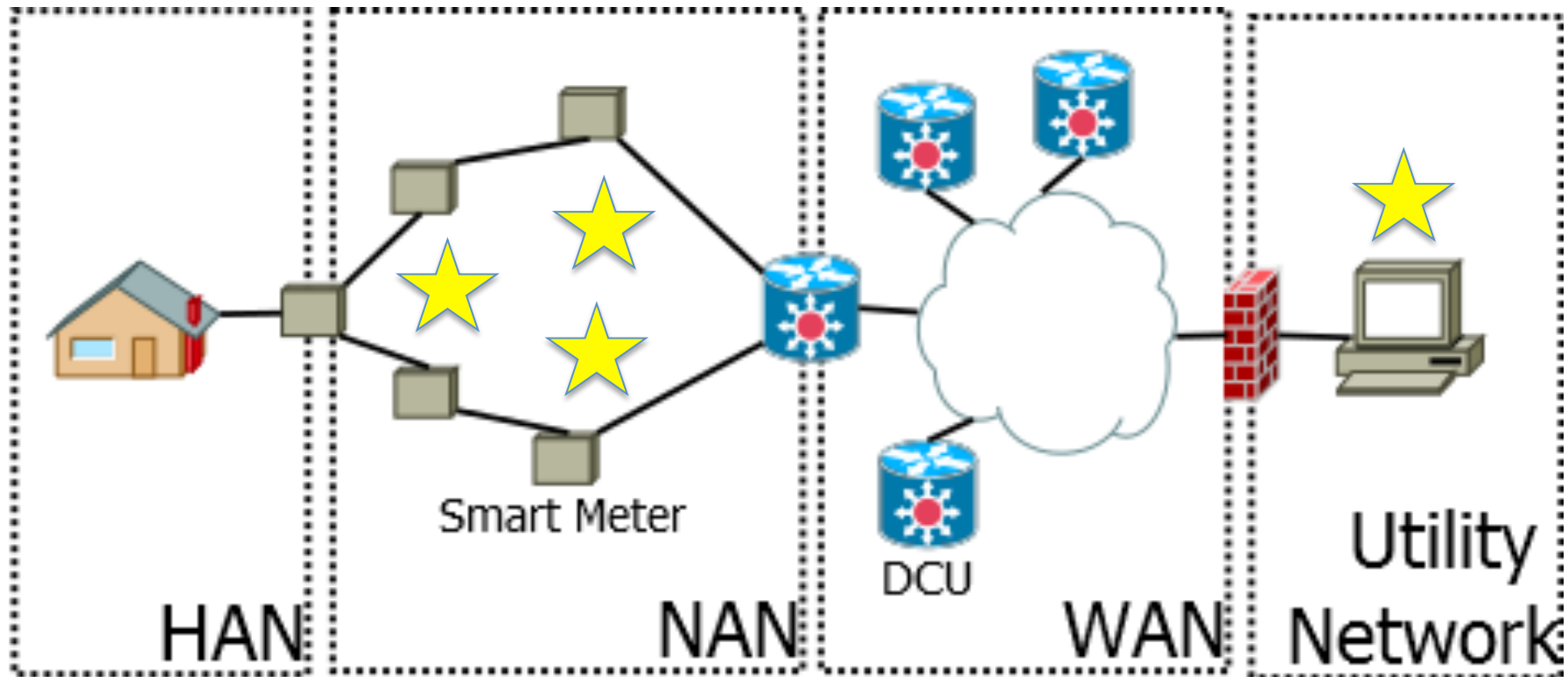
Advanced Metering Infrastructure Case Study

Define system components, relationship and attributes



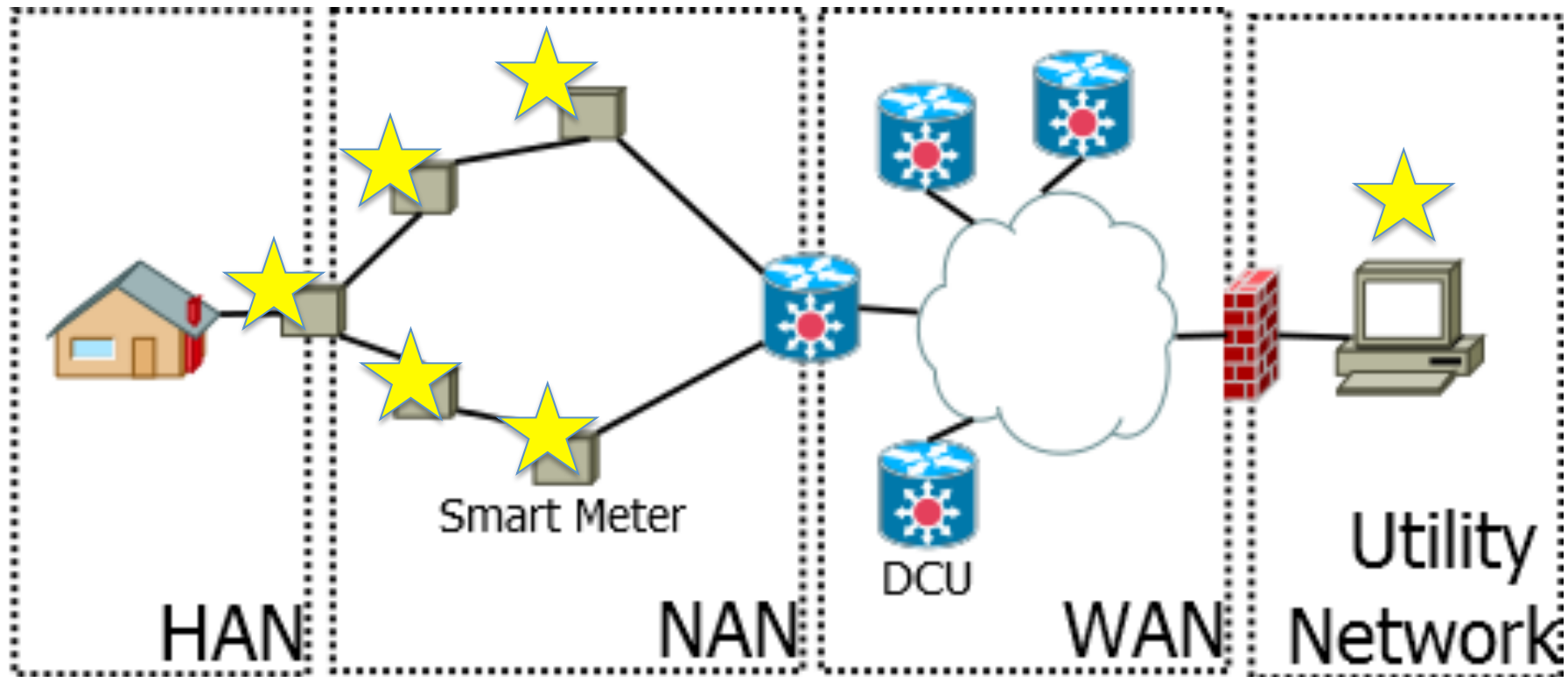
Advanced Metering Infrastructure Case Study

Define system components, relationship and attributes

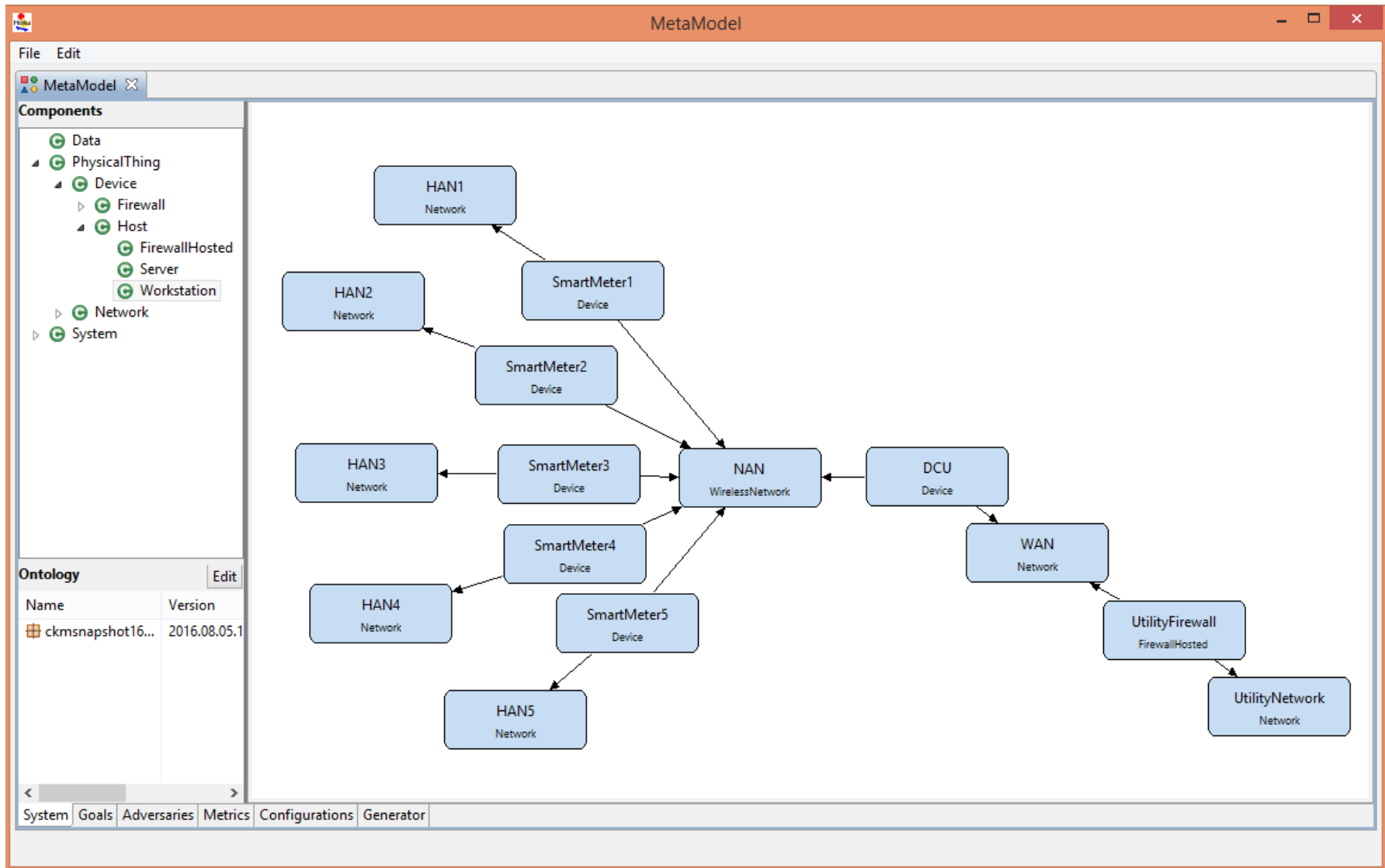


Advanced Metering Infrastructure Case Study

Define system components, relationship and attributes



Advanced Metering Infrastructure Case Study



Advanced Metering Infrastructure Case Study

Model Attacker Goals

The screenshot displays the MetaModel application window. The interface is divided into several sections:

- Available State Variables:** A tree view showing a hierarchy of variables: Global State Variables, DCU, HAN1, HAN2, HAN3, HAN4, HAN5, and NAN.
- Ontology:** A table with columns for Name and Version. One entry is visible: ckmsnapshot16... with version 2016.08.05.10.36.
- Goals:** A list of goals with 'Add' and 'Delete' buttons. Two goals are listed: Goal_DamageNAN and Goal_DamageWAN. The 'Goal_DamageNAN' goal is selected, and its configuration is shown in the right pane.

The configuration for the selected goal 'Goal_DamageNAN' includes:

- Name:** Goal_DamageNAN
- Dependent State Variables:** A list containing 'NAN_Damaged'.
- Goal Expression:** A text area containing the code: `return NAN_Damaged->Mark();`

At the bottom of the window, there is a navigation bar with tabs for System, Goals, Adversaries, Metrics, Configurations, and Generator. The 'Goals' tab is currently active.

Advanced Metering Infrastructure Case Study

Model Adversaries

The screenshot displays the MetaModel application window, which is used for configuring model adversaries. The interface is divided into several sections:

- Adversary Templates:** A list of templates including Customer, EconomicCompetitorLimitedResource, EconomicCompetitorWellResourced, ForeignGovernmentLimitedResource, ForeignGovernmentWellResourced, HackerGroup, IndependentInsider, OrganizedCrime, and TerroristOrganization.
- Adversaries:** A list of configured adversaries: InsiderAdversary, NationStateAdversary, and TerroristAdversary. The "InsiderAdversary" is currently selected.
- Configuration Panel:** A detailed view of the selected "InsiderAdversary" configuration, including:
 - Name:** InsiderAdversary
 - Decision Parameters:** Planning Horizon: 20, Cost of Detection: 5000.
 - Access:** A table with columns for Name, Initial Value, and actions (Add..., Remove). One entry is "InsiderAccess" with an initial value of 1.
 - Knowledge:** A table with columns for Name, Initial Value, and actions (Add..., Remove).
 - Skills:** A table with columns for Name, Initial Value, and actions (Add..., Remove). One entry is "BasicCyberOffense" with an initial value of 500.
 - Goals:** A table with columns for Name, Initial Value, Payoff, and actions (Add..., Remove).
- Ontology:** A table with columns for Name, Version, and URL. One entry is "ckmsnapshot16..." with version "2016.08.05.10.36.52" and URL "http://win8".
- Navigation:** A bottom bar with tabs for System, Goals, Adversaries, Metrics, Configurations, and Generator.

Advanced Metering Infrastructure Case Study

Select Metrics

- Damage to System
- Probability of Detecting Adversary
- Attack Path

Advanced Metering Infrastructure Case Study

Create Configurations

- Create one configuration for every pair of adversary/IDS, for a total of 3 adversaries X 4 IDS options = 12 configurations.

Advanced Metering Infrastructure Case Study

Results

IDS	Adversary	Attack	Monetary Damage
None	Insider	Routing	\$1.07M
	Terrorist	Physical	\$4.98M
	Nation-State	Routing	\$876K
Centralized	Insider	Routing	\$435K
	Terrorist	Physical	\$4.98M
	Nation-State	Routing	\$357K
Dedicated	Insider	None	\$0
	Terrorist	Physical	\$4.98M
	Nation-State	None	\$0
Embedded	Insider	None	\$0
	Terrorist	Physical	\$5.02M
	Nation-State	None	\$0

Agenda

- Registration and Continental Breakfast
- Welcome
- Goals
 - Tool
 - Workshop
- Steps to Use ADVISE Meta
- Hands on Sessions
- Case Studies and Custom Ontologies
- Wrap Up

Wrap Up

- General feedback
 - Was this tool useful?
 - How could you and your organization use it?
 - What areas need work?
- The Near Future
 - Improvements to ADVISE/Actor Model
 - Expanded Ontology
 - System to Easily Share Ontology Packages
 - Ontology Editor Improvements / Validation

Thank You!

- Contact Info
 - staff@mobius.illinois.edu
 - <http://www.mobius.illinois.edu>