imagination at work

MOBIUS ADVISE META MODELING EXPERIENCE REPORT

Prepared for Mobius Production Team

Smith, Sean (GE Global Research, US)
sean.m.smith@ge.com

Contents

1 Abstract

An evaluation of the Mobius Tool (Model-Based Environment for Validation of System Reliability, Availability, Security, and Performance) was conducted at the General Electric Global Research Center in Niskayuna, NY in July and August, 2016. The evaluation was undertaken to examine program relevance and performance using test scenarios of two existing GE systems. The purpose and application of the tool, development and execution of a model, and usefulness of tool outputs were assessed. The goal of this document is to record the initial challenges, observations, and experience in use of the tool.

Both the ADversary VIew Security Evaluation (*ADVISE*) and ADVISE meta-model formalisms were used to develop two scenarios based on existing GE systems for the model development exercise. The models used in the scenarios were based on a previously created and populated meta-model ontology. System-level security metrics were created for the evaluation.

Once the general architecture of the tool and representation of an attack scenario were understood, development of the attack model in the tool GUI canvas was found to be straightforward. For the evaluation scenarios, system attack graphs were built. Nodes to model the advisory skills, knowledge, and access were easily generated and modified to suit the test scenario. The system model access points, intrusion protections and relationships were created in a similar fashion.

The meta-model concept was found to reduce development time and error, and increase code reuse of the model for alternate scenarios. Several ideas for tool extensions were recorded to further aid model development. However, execution of the simulation and display of simulation progress and results were found to be difficult to understand. Several tool display modifications and extensions to improve illustration and progress of the simulated attack were documented.

2 Introduction

Security risks are high in automation and control systems due to increased architectural openness and interoperability. There is significant need for a tool that can accurately analyze a specified system and provide useful information on particular vulnerabilities – aspects of a system's architecture, exposed surfaces, or communication interfaces – that are open to attack. Today, with the proliferation of network connected control systems, much can be done to assess the security of these systems of systems.

Developed in cooperation with the Department of Homeland Security, Möbius is a software tool that utilizes several formalisms to model systems and components. One of these introduced in Möbius is the ADVISE modeling and meta modeling formalism which allows creation of systems models for security risk analysis. A possible application of such a tool would be trade-off analysis of security options, for example the selection of a secure protocol. Mobius simulation could indicate probabilities of access to a given system through a specified exploit.

This document examines the use of the Mobius tool provided by the (Mobius production) team at the University of Illinois by evaluating the overall process of constructing models, use of the ADVISE meta model, assessment of system security risks, and review and analysis of simulation output. Recommendations and feedback are provided for the Möbius production team and questions are posed about the development of models and test scenarios and the usability of the tool.

The report is organized into five main sections. Section 1, (Introduction), presents a profile of the tool created by the Möbius team and the as viewed potential purposes and applications for the tool. Section 2 presents the methodology for the evaluation and discusses steps used for tool familiarization and additional information required. Section 3 presents the process of the evaluation and walks through steps taken to evaluate the Möbius tool. Section 4 presents the findings of the evaluation. Finally, Section 5 presents the conclusion of the evaluation and any additional comments.

Note:

Throughout this report, "tool" refers to the Mobius software tool. "system" refers to the system that is modeled and subject to analysis

3 Evaluation Plan

During the process of developing GE test scenarios, the Möbius tool was examined with respect to usability -- the ease with which inputs are entered and modified, simulations are configured and executed, and simulation outputs are accessed and organization. Performance and of the resultant model simulations were inspected. As well, the usefulness of the tool was examined – does the software produce meaningful and correct outputs? Tool usability and usefulness were measured against common experience and practice as opposed to direct comparison with alternate tools.

Scope was restricted to individual models within the ADVISE formalism using the meta-model ontology. The evaluation included model development and sample execution but was limited in attempts to validate the models against real case studies or historical attack data.

3.1 Evaluation Criteria

Table 1 details the criteria for evaluation and the questions addressed in each category.

Table 1 - Summary of Evaluation Criteria and Questions

Evaluation Criteria	Evaluation Question
Relevance	• What is the purpose of the tool? • What are the applications for the tool? • Is there a need for the tool?
Performance (Effectiveness)	• Was the output from the tool useful? • How accurate is the data? • Are there any gaps in the data? • What metrics are output?
Performance (Efficiency)	• How efficient is the tool at modeling systems? • Is there an alternate way to get the same information more efficiently than the tool? • How long does it take to run an analysis?
Performance (Usability)	• Is the tool intuitive and easy to use? • How big is the learning curve? • Is there sufficient documentation for the tool? • How is the process flow when analyzing a system?

3.2 Scenario Development

Test Scenarios using existing GE systems were developed in the tool and used to evaluate tool usability, providing unique feedback in areas not covered by tutorials and examples.

The first scenario implements a model of a power distribution substation. This example uses a system ontology (formal representation of a set of domain concepts and the relationships between those

concepts), providing a view of tool performance using pre-defined system definition. This ontology, provided by the Möbius team, covered networked computer system components.

The second scenario implemented a reverse osmosis water treatment system. No pre-defined ontology was available for this system type and one was created using the networked computer system ontology as an example, allowing evaluation of Mobius ontology creation functions.

3.3 Observations

The data collection consisted of notes, observations, recommendations, impressions, and diagrams.

However, the evaluation did not include, for example, examination of model creation time, simulation execution time, or simulation compute resource utilization. No numerical measurements or comparisons are presented.

4 Evaluation Process

The evaluation process was a straightforward set of steps which made significant use of provided examples, tutorials, and documentation. The main steps were as follows:

1. Investigation of Möbius formalities and their use; Industrial Control Systems (ICS) vulnerabilities and components.

   Become familiar with the tool through the provided tutorials.
   In particular, import example bank robbery attack scenario.
   Execute scenario, examine simulation output, experiment through modification of the model, parameters, and code.

2. Model Development through use of Möbius ADVISE meta-model ontology, rewards model, and simulation.

   Develop test scenario modeled from bank robbery and network intrusion tutorials.
   Program control system vulnerabilities from reference papers.

3. Simulation and Analysis
   Run the models created through the Möbius simulation.
   Review the results provided and make a conclusion.
   Apply refinements to test scenario.
   Execute simulation, examine output, examine sensitivity of selected parameters. (repeat)
   Modify model to include alternate attacker, hardened exploit/vector, alternate exposure.

4.1 Investigation

The initial phase of the evaluation consisted of tool exploration – user interface, functions, model creation and execution. Several documents were helpful in understanding modeling methods, representation of vulnerabilities in an Industrial Control System (ICS), and the project execution within the Möbius tool.

The Möbius wiki provided an example called Bank Robbery implementation. This example allowed examination of manual creation of an ADVISE model and additional model components required to create a simulation. Several models were integrated in the simulation.

The ADVISE Model was created first to represent the attack paths that could be used against a system. Avenues of intrusion are modeled using the paths; probabilities of success toward a goal using a given path were entered.

A Rewards Model was added to define variables of interest to be displayed during simulation and their output interval.

The model was then incorporated in a Study containing the set of experiments or simulation runs and their input parameters. The Study specified initial values of variables within the ADVISE Model, such as skill levels, global variables, access values.

Lastly, a simulation configuration incorporating the three models was specified including output to be displayed and maximum iterations.

The networked systems ontology was used to create components that were then incorporated in the (attack) model. The ability to specify unique or particular components specific to the modeled system was found to be very flexible. A capability to model a multitude of different systems, while maintaining the internal analysis appeared to be powerful.

> ***Recommendation #1:*** *Add the ability for the user to specify additional ontology components supporting new or additional specific domain knowledge.*

> ***Recommendation #2:*** *Additional documentation of the ontology and its components would be helpful. More complex and complete ontologies could be used for more accurate system models.*

Following integration of a meta-model ontology, vulnerabilities of Industrial Control Systems (ICS) were developed. The model was extended to include different types of attackers and attack groups. Types included nation states, terrorists, lone attackers, and insiders, each with a profile characterized by a distinct set of skills, knowledge, and access. The ability of the Möbius tool to link components of a system into an ADVISE model providing a simulation of different avenues of attack by different attacker types was found to be a powerful feature of the tool.

4.2 Model Development

In the model development stage, the substation test scenario was programmed using the bank robbery example as a foundation. Model refinement included additions to make the bank robbery example look and behave more like a network hacker attack on a networked control system. The attack model was extended with additional access points – wireless communications to electrical grid control, wired connection to the internet, and connection between controllers and monitoring systems.]

> ***Recommendation #3:*** *Provide a clearer representation of the results. Currently, results are listed in the order defined in the Rewards Model. A clearer way might be, columns from left to right, Time, Reward Variable 1, Reward Variable 2, etc. In addition, list mean values and confidence intervals in the appropriate Reward Variable column.*

> ***Recommendation #4:*** *In ontology creation and entry of parameters such as attack cost or outcome probability, display upper/lower bounds accepted by the program.*

> ***Recommendation #5:*** *Improve documentation or provide aids to entry of ontology object coding. The addition of code to object input boxes was confusing; function of resultant executable code was uncertain. Give a brief description on what impact the value has on the system.*
>
> ***Question #1:*** *Is there sufficient documentation for the tool? Documentation on the methods that are implemented and the modeling strategies was adequate. More documentation on entry fields would be helpful -- description of how used in calculations.*

The procedural coding required for an attack model was found to be reduced using the meta-model. Use of the meta-model probably also reduced the chance for entry and transcription errors. With the ADVISE meta model created and extended, a rewards model was added to the attack, providing a study definition. The study was configured for Performance Variables. Options were examined for the use of the study model in a range study, in particular, creation of an ADVISE model experiment for execution with specification of global variables. The global variables were used to set different skills, knowledge, or access points. Simulation output display variables were selected; time interval of display and resolution were specified.

> ***Recommendation #6:*** *Currently, rewards model variables can be added to a watch list and monitored at a specified interval. A custom range, or viewing for an entire simulation would be useful.*
>
> ***Recommendation #7:*** *Currently, rewards variables are added to the watch list at creation. It would convenient to also specify ADVISE model elements for display.*

The range study generated combinations of selected variables and defined a separate experiment for each combination. And so, was used for the creation of multiple experiments. Alternatively, it appeared the study model creation function performs only an open/close operation. In this mode is was difficult to interact with the study and incorporate it with multiple simulations.

> ***Recommendation #8:*** *Similar to rewards model auto generation (refer to recommendation #4), generate a range study that corresponds to the global variables defined in the ontology. This would include a range specified by the tool encompassing possible global variables values.*

After the range study was constructed, the appropriate solver was integrated. Using the solver, the range study was incorporated to the simulation execution configuration. Within the ADVISE model study variables were specified and initial values given.

> ***Recommendation #9:*** *Improve model creation efficiency through increase use of the meta-model ontology. Much of the work for modeling creation was eased through the use of the ontology -- ADVISE model elements are defined in the ontology, all of the connections are defined, all of the coding is done within the ontology.*

Recommendation #10: *Provide a less flexible but easier method for model creation. Currently the user is required to place all of the logic code, initialization code and parameter code within the ontology. Provide more auto generation or base class definition or parameter defaults to enhance rapid model development.*

4.3 Simulation and Analysis

Once the system was represented using the ADVISE meta model and associated models -- the reward model, the study model -- the solver model was created to complete the project.

The next steps were to run the model through the simulation and investigate the results. Thoughts and additional questions regarding simulation output were generated. The output from the simulation listed rewards variables set by the user in the reward model. Columns displaying mean with confidence interval were displayed. These were assumed to indicate mean probabilities for a specific instance in time, however it was not clear.

Recommendation #11: *Provide a default/standard simulation output so that models and studies can more easily be compared.*

Recommendation #12: *Provide headings for simulation outputs. Provide descriptions of simulation outputs meaning and how calculated.*

Recommendation #13: *Improve display of attack (simulation goal achieved) success details -- what path was taken, what accesses was gained, and what skills and knowledge were used. Output the path the attacker took and possibly the other paths that were attempted and how far the attacker proceeded on alternate paths.*

Illustration of the successful path of the attacker was generated using a Mobius system function. Recommendation #12 documents additional thoughts for the path highlights; output of the path taken by the attacker.

Question #2: *How is the simulation output best tested? Given simulation output from the tool as probabilities, a real world example that validates the model would be useful but difficult to model. A representation of an existing system could be constructed; however, it would be difficult to estimate attacker skill and knowledge profiles. construct a system and then get an adversary matching the profile modeled to attack the system.*

Question #3: *Was the output from the tool useful? It appears the output from the tool is best used for model comparisons. For example, system with a certain level of firewall protection could be compared to another system with an estimated higher degree of firewall protection.*

Question #4: *Would it be useful to provide output to allow the user to track the progress of the attacker.*

Next, an analysis of the simulation results display was performed to measure tool usefulness. A comparison of simulation to a recorded break-in was considered. A model of a documented (historical) break-in would be required, with a timeline and detailed list of steps taken in the actual break-in.

Comparison of two similar simulated attacks was considered to test the sensitivity of individual component parameters, for example, reduction of the level of difficultly of an attack step to increase the probability that a break-in success as might be expected. This latter use of simulation results appeared the most promising -- comparison of system configurations. The tool might be best used to answer questions concerning the relative impact of strengthening vulnerable components.

> ***Recommendation #14:*** *Better conclusions about the output data if the tool had prebuilt (and somehow validated) system ontology components and adversary models.*
>
> ***Question #5:*** *What metrics are output? Probabilities of the current state of an element within the ADVISE model are output. These can be set to goal elements, representing the probability that a specific adversary achieves a particular goal, or to access points or knowledge elements. So the tool necessarily assesses the vulnerabilities of the system. With the vulnerabilities for the system defined within the ontology and the Möbius ADVISE meta model linked to these ontology components, the vulnerabilities of the system are already known to the user and the Möbius tool. The Möbius tool does not report the cyber-security attack vulnerabilities a system, it only reports the probability of a particular model element's state at an instance of time.*
>
> ***Recommendation #15:*** *Improve the format of the simulation output. This suggestion is illustrated below in figures 2 and 3. The alternate format displays variables of interest at each time step.*

Mean Results

Name	Time	Mean		Confidence Interval	
SCADANetworkCompromised	0.0	0.0000000000E00	+/-	0.0000000000E00	
SCADANetworkCompromised	2.0	0.0000000000E00	+/-	0.0000000000E00	
SCADANetworkCompromised	4.0	3.0000000000E-03	+/-	1.0719782991E-03	(*)
SCADANetworkCompromised	6.0	3.6300000000E-02	+/-	3.6660807717E-03	(*)
SCADANetworkCompromised	8.0	1.6290000000E-01	+/-	7.2381403552E-03	
SCADANetworkCompromised	10.0	3.4830000000E-01	+/-	9.3385271315E-03	
SCADANetworkCompromised	12.0	4.9200000000E-01	+/-	9.7992354937E-03	
SCADANetworkCompromised	14.0	5.8290000000E-01	+/-	9.6648453634E-03	
SCADANetworkCompromised	16.0	6.6230000000E-01	+/-	9.2698070620E-03	
SCADANetworkCompromised	18.0	7.3310000000E-01	+/-	8.6702911980E-03	
SCADANetworkCompromised	20.0	7.8630000000E-01	+/-	8.0347836302E-03	
SCADANetworkCompromised	22.0	8.2660000000E-01	+/-	7.4207881630E-03	
SCADANetworkCompromised	24.0	8.6220000000E-01	+/-	6.7562607951E-03	
EngrLanCompromised	0.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	2.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	4.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	6.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	8.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	10.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	12.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	14.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	16.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	18.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	20.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	22.0	0.0000000000E00	+/-	0.0000000000E00	
EngrLanCompromised	24.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	0.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	2.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	4.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	6.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	8.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	10.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	12.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	14.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	16.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	18.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	20.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	22.0	0.0000000000E00	+/-	0.0000000000E00	
CorpLanCompromised	24.0	0.0000000000E00	+/-	0.0000000000E00	

Figure 1 Simulation results (as displayed)

	SCADANetworkCompromised		EngrLanCompromised		CorpLanCompromised	
Time [h]	Mean	CI [+/-]	Mean	CI [+/-]	Mean	CI [+/-]
0.0	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
2.0	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
4.0	3.00000000E-03	1.07197830E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
6.0	3.63000000E-02	3.66608077E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
8.0	1.62900000E-01	7.23814036E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
10.0	3.48300000E-01	9.33852713E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
12.0	4.92000000E-01	9.79923549E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
14.0	5.82900000E-01	9.66484536E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
16.0	6.62300000E-01	9.26980706E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
18.0	7.33100000E-01	8.67029120E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
20.0	7.86300000E-01	8.03478363E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
22.0	8.26600000E-01	7.42078816E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00
24.0	8.62200000E-01	6.75626080E-03	0.00000000E+00	0.00000000E+00	0.00000000E+00	0.00000000E+00

Figure 2 Simulation results (proposed)

5 Test Scenarios

Two example test scenarios were programmed using existing GE systems. These scenarios were included in the evaluation to allow direct comparison of Mobius models to actual GE protection system design estimates and data from recorded actual attacks.

The first example was a power distribution substation which included a network of controllers, wireless and wired communication interfaces, and physical and computer system entry points. The model for this system was adapted from the Mobius bank robbery example.

The second example was a reverse osmosis water treatment system which contained several physical security risks in addition to the network communication and controller intrusion vulnerabilities.

5.1 Test Scenario 1 -- Power Distribution Substation

The modeled medium power distribution substation consists of …

1. (4) networked breaker protection controllers and associated sensors,
2. An engineering workstation for instrumentation, configuration, and diagnostics,
3. Wireless communications link to a power grid operations center,
4. Optional wired internet connection to the engineering workstation,
5. IRIG-B time signal receiver,
6. Connection to high voltage input and medium voltage output transmission lines.

The system electrical schematic is shown in figure 5.1 and system block diagram is shown in figure 5.2.

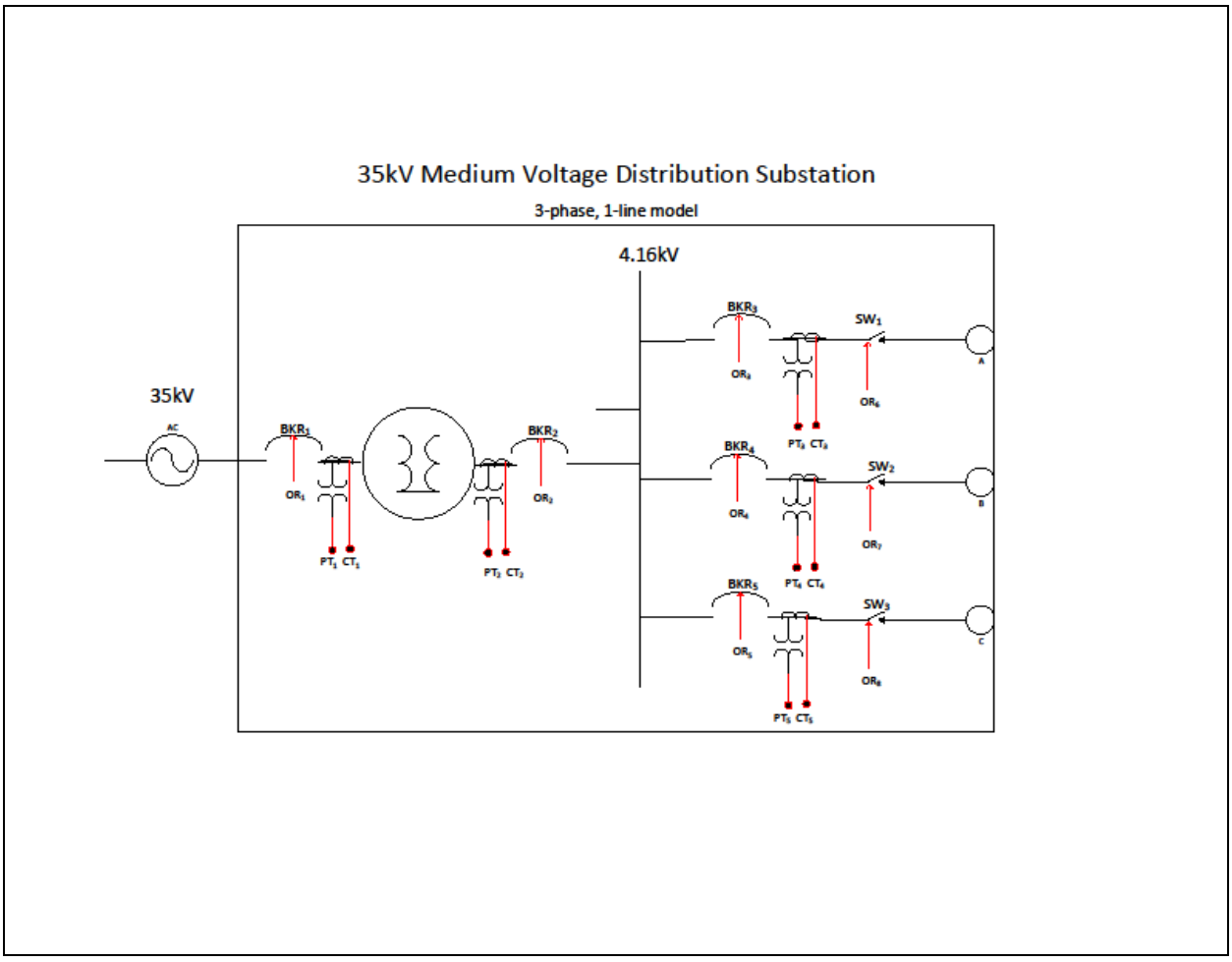

Figure 5:1 - Power Distribution Substation Diagram

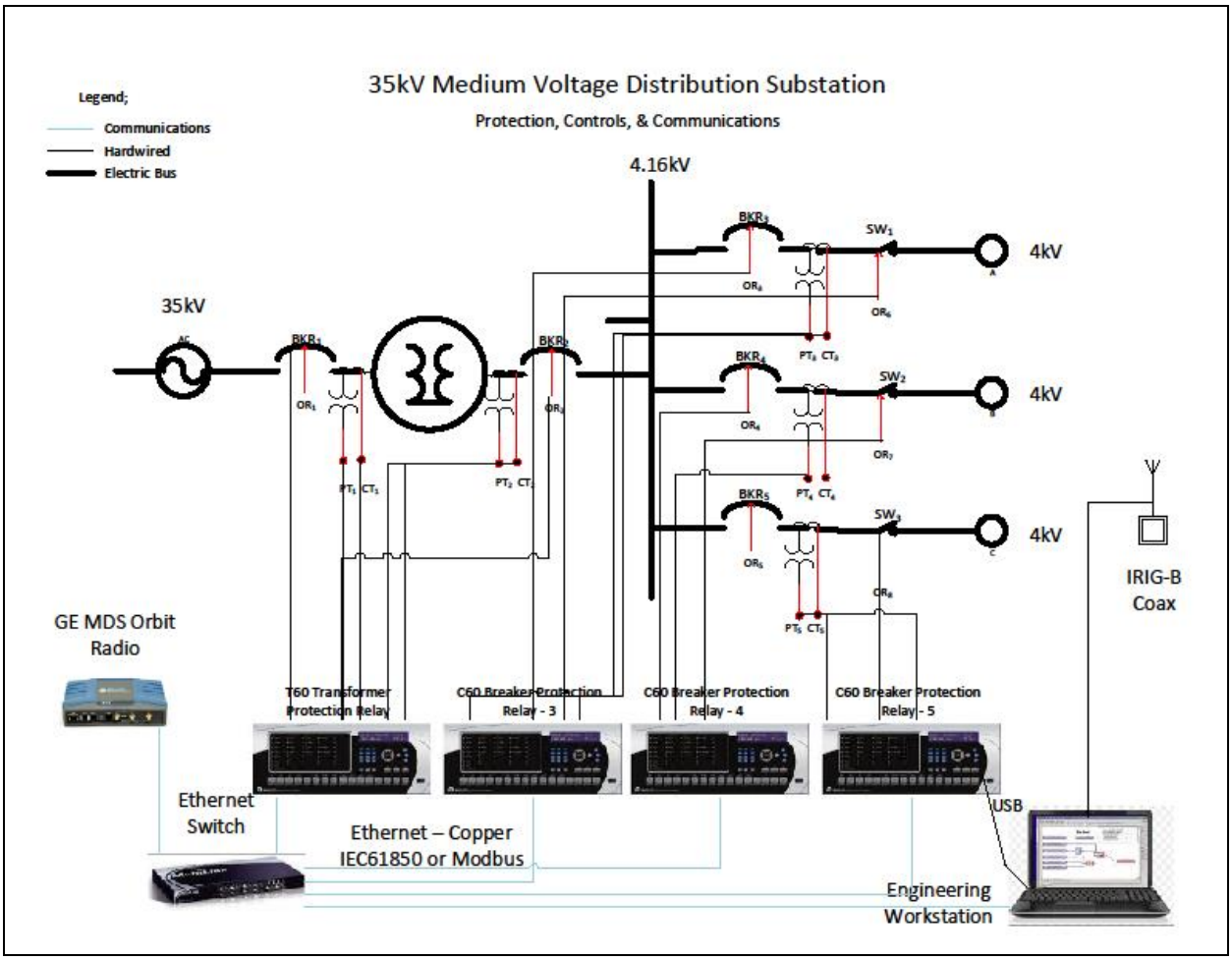

Figure 5:2 - Power Distribution Substation Diagram with Control System

This power distribution control system was then modeled using Möbius ADVISE incorporating the meta modeling tool. Characterization of avenues of attack were programmed using details collected from internet sources, GE engineers experienced in the domain, and examples available from Mobius documentation.

The meta model constructed is shown in the figure 5.3.

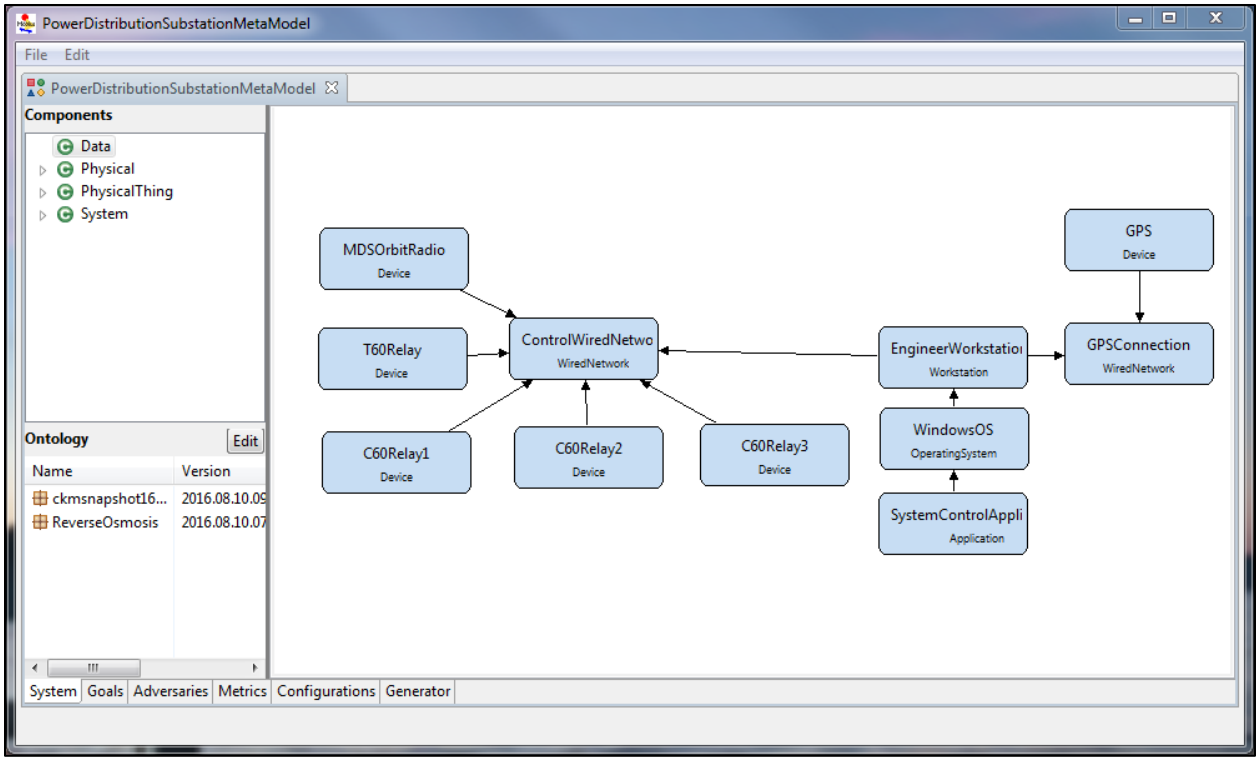

Figure 5:3 - Power Distribution Substation ADVISE Meta Model

The meta-model ontology that was used in the construction of the ADVISE Meta Model did not included physical components of power distribution substation and detailed connections, for example functions internal to the breaker protection controllers. As shown in the model representation, at this level of detail these aspects are hidden.

> ***Recommendation #16:*** *Provide a brief description for each component within the ontology, its attributes, and a description/meaning of each parameter.*

As with the previous models, once the ADVISE Meta Model was created, a Rewards Model was added, followed by a Study for configuration of Simulation.

> ***Recommendation #17:*** *It is not clear from compilation error messages what in particular is wrong within the parameters and files created within the Möbius tool. Debug could be aided by a form of translation between the cpp compiler and the Möbius tool. Information could be provided to indicate the error in the Möbius model producing the error in the cpp file. Alternatively, could the Möbius tool provide its own error handling before passing code to the cpp compiler?*

5.2 Test Scenario 2 -- Reverse Osmosis Water Treatment System

A second test scenario was constructed for a portable water treatment system which differed substantially from the power substation in that it is primarily a mechanical system. The goal was to model

physical exploits such as flow interruptions or contaminant injections that would destroy the system, in addition to intrusions to the computer control system.

The system consists of:

1. (4) input sediment filters,
2. reverse osmosis filter for dissolved inorganic solids,
3. (2) Carbon adsorption VOC filters,
4. Control system,
5. engineering workstation for instrumentation, configuration, and diagnostics,
6. pressure, flow, and temperature sensors,
7. RO pump and valves,
8. diverting pump and valve,
9. inlet and outlet pumps,
10. storage tank,
11. Optional wired internet connection to engineering workstation.

The system block diagram is shown in figure 5.4.

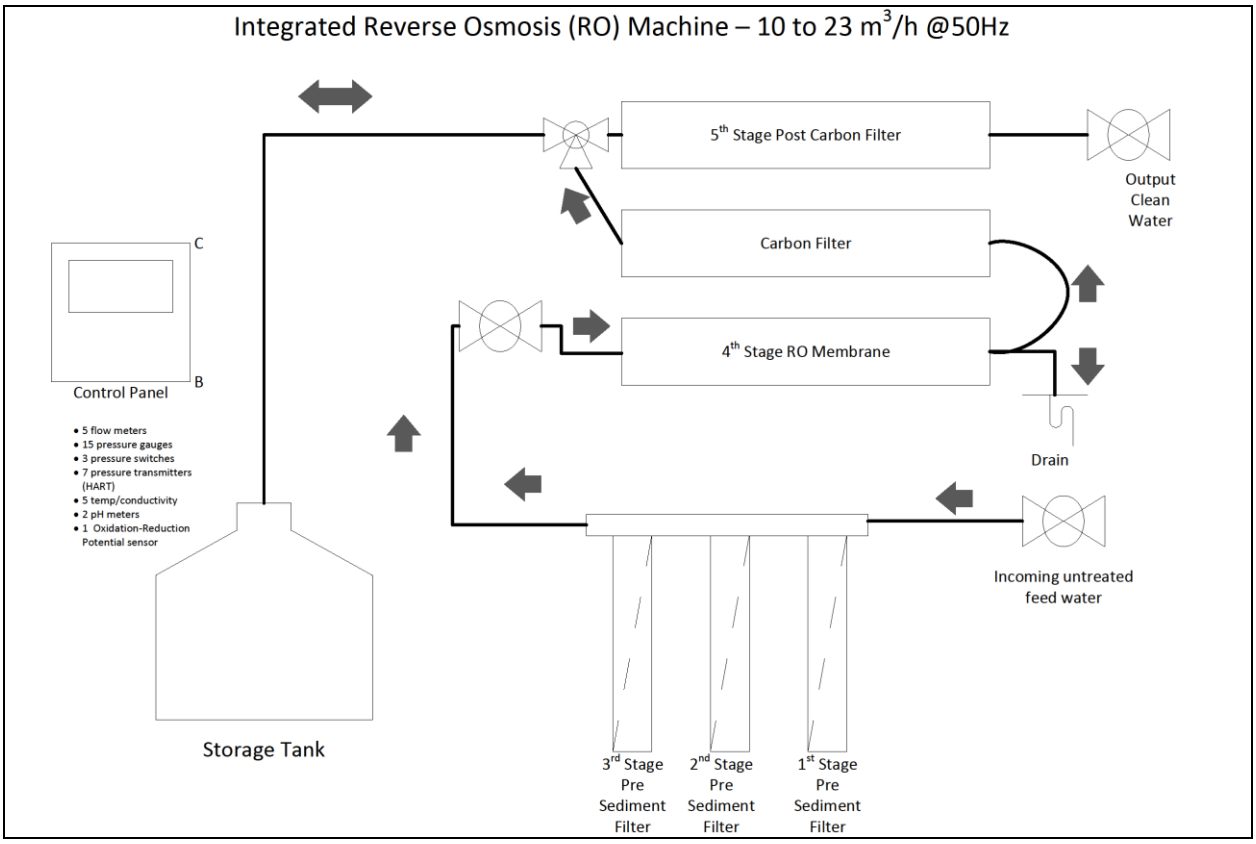

Figure 5:4 - Reverse Osmosis Water Treatment System Schematic

The mechanical aspects of this system were not accommodated by the meta-model ontology used in the substation. This required new ontology components to constructed and allowed evaluation of meta-model development functions within the tool. The rest of the model development – creation of the meta model, the rewards model, the study, and the simulation -- was relatively the same as the substation test scenario and the tutorials provided by the Möbius team.

The ADVISE Meta Model used to describe this system is shown in figure 5.5. Note that the model does not include the storage tank or the controller. It only models the physical attributes of the system including only the basic linear flow arrangement of the filtration system.

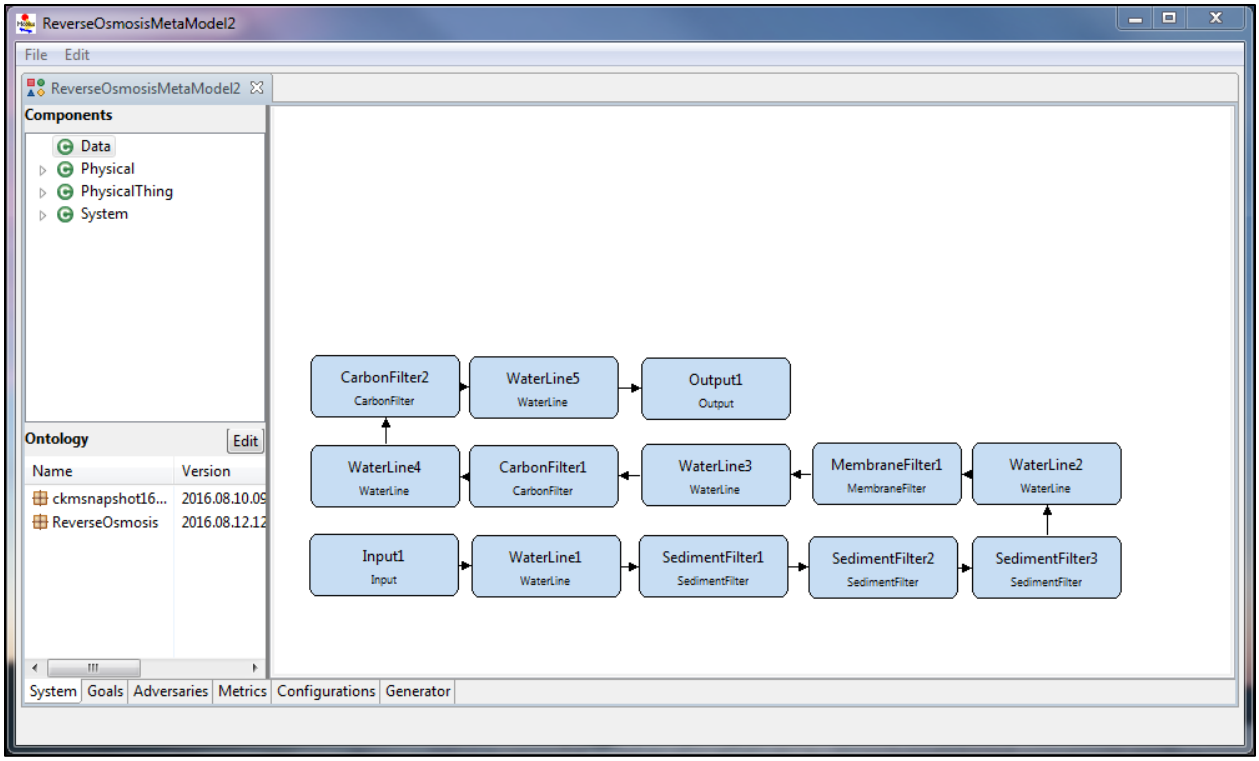

Figure 5:5 - Reverse Osmosis Water Treatment ADVISE Meta Model

In order to incorporate an ADVISE meta-model a system ontology was created. Ontology development was found to be less intuitive. The methodology was very much different than the attack models and development was more difficult than expected, especially with much less example code. Consequently, it was viewed that confidence in the accurate representation of the system was reduced and confidence in the validity of the resultant attack model was reduced.

6 Conclusion

Model development using the Mobius tool for both ADVISE and ADVISE meta-models was found to be straight forward and efficient through adoption of aspects of the Mobius bank robbery and networked computer intrusion examples. Two test scenarios modeling attacks on existing GE systems, an electric power distribution substation and a portable water treatment facility were developed, one using the meta-model ontology and the other using lower-level ADVISE modeling. Recommendations regarding the usability of the tool were recorded for these exercises.

Following development of the GE test scenarios, the simulation process and analysis of simulation results for these models were examined. Questions and recommendations regarding further refinement of the meta-model and its execution were recorded. Validation of the model and verification of the simulation were areas that elicited questions concerning the operation of the tool. Proposals for additional tool functionality to improve the process and increase tool usefulness were documented. Further investigation of simulations using the GE test scenarios, model refinement, and examination of model validation would help answer some of these questions.

7 Recommendations

Recommendation #1: *Add the ability for the user to specify of additional ontology components supporting new or additional specific domain knowledge.*

Recommendation #2: *Additional documentation of the ontology and its components would be helpful. More complex and complete ontologies could be produced for more accurate system models.*

Recommendation #3: *Provide a clearer representation of the results. Currently, results are listed in the order defined in the Rewards Model. A clearer way might be, columns from left to right, Time, Reward Variable 1, Reward Variable 2, etc. In addition, list mean values and confidence intervals in the appropriate Reward Variable column.*

Recommendation #4: *In ontology creation and entry of parameters such as attack cost or outcome probability, display upper/lower bounds accepted by the program.*

Recommendation #5: *Improve documentation or provide aids to entry of ontology object coding. The addition of code to object input boxes was confusing; function of resultant executable code was uncertain. Give a brief description on what impact the value has on the system.*

Recommendation #6: *Currently, rewards model variables can be added to a watch list and monitored at a specified interval. A custom range, or viewing for an entire simulation would be useful.*

Recommendation #7: *Currently, rewards variables are added to the watch list at creation. It would convenient to also specify ADVISE model elements for display.*

Recommendation #8: *Similar to rewards model auto generation (refer to recommendation #4), generate a range study that corresponds to the global variables defined in the ontology; a range specified by the tool that encompasses possible global variables values.*

Recommendation #9: *Improve model creation efficiency through increased use of the meta-model ontology. Much of the work for modeling creation was eased through the use of the ontology -- ADVISE model elements are defined in the ontology, all of the connections are defined, all of the coding is done within the ontology.*

Recommendation #10: *Provide a less flexible but easier method for model creation. Currently the user is required to place all of the logic code, initialization code and parameter code within the ontology. Provide more auto generation or base class definition or parameter defaults to enhance rapid model development.*

Recommendation #11: *Provide a default/standard simulation output so that models and studies can more easily be compared.*

Recommendation #12: *Provide headings for simulation outputs. Provide descriptions of simulation outputs meaning and how calculated.*

Recommendation #13: *Improve display of attack (simulation goal achieved) success details -- what path was taken, what accesses was gained, and what skills and knowledge were used. Output the path the attacker took and possibly the other paths that were attempted and how far the attacker proceeded on alternate paths.*

Recommendation #14: *Provide better indication in the output of whether the tool contains prebuilt (and somehow validated) system ontology components and adversary models.*

Recommendation #15: *Improve the format of the simulation output. An alternative format displays variables of interest at each time step.*

Recommendation #16: *Provide a brief description for each component is within the ontology, its attributes a description/meaning of each parameter.*

Recommendation #17: *It is not clear from compilation error messages what in particular is wrong within the parameters and files created within the Möbius tool. Debug could be aided by a form of translation between the C++ compiler and the Möbius tool. Information could be provided to indicate the error in the Möbius model producing the error in the .cpp file. Alternatively, could the Möbius tool provide its own error handling before passing code to the C++ compiler.*

8 Questions

Question #1: *Is there sufficient documentation for the tool? Documentation on the methods that are implemented and the modeling strategies was adequate. More documentation on entry fields would be helpful -- description of how used in calculations.*

Question #2: *How is the simulation output best tested? Given simulation output from the tool as probabilities, a real world example that validates the model would be useful but difficult to model. A representation of an existing system could be constructed; however, it would be difficult to estimate attacker skill and knowledge profiles. construct a system and then get an adversary matching the profile modeled to attack the system.*

Question #3: *Was the output from the tool useful? It appears the output from the tool is best used for model comparisons. For example, system with a certain level of firewall protection could be compared to another system with an estimated higher degree of firewall protection.*

Question #4: *Would it be useful to provide output to allow the user to track the progress of the attacker.*

Question #5: *What metrics are output? Probabilities of the current state of an element within the ADVISE model are output. These can be set to goal elements, representing the probability that a specific adversary achieves a particular goal, or to access points or knowledge elements. So the tool necessarily assesses the vulnerabilities of the system. With the vulnerabilities for the system defined within the ontology and the Möbius ADVISE meta model linked to these ontology components, the vulnerabilities of the system are already known to the user and the Möbius tool. The Möbius tool does not report the vulnerabilities a system, it only reports the probability of a particular element's state at an instance of time within the model.*

9 Next Steps

As an alternative to the approach taken in this evaluation, it might be valuable to examine tool usefulness from the perspective of a user experienced or knowledgeable in attack methods. Questions concerning how an example attack might progress in time were not explored due to lack of exposure to more sophisticated attack techniques, for example, how an attacker interacts with a system's defenses in order gain an advantage, and how this type of attack can be modeled in Mobius.

10 References

Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J. M., . . . Webster, P. (n.d.). *The Mobius Modeling Tool.* University of Illinois at Urbana-Champaign.

Courtney, T., Daly, D., Derisavi, S., Lam, V., & Sanders, W. H. (n.d.). *The Mobius Modeling Enviroment.* University of Illinois at Urbana-Champaign.

Feddersen, B., Keefe, K., Sanders, H. W., Muehrcke, C., Parks, D., Crapo, A., . . . Palla, R. (n.d.). *An Ontological Model for Constructing Mobius.* University of Illinois. Retrieved from https://www.perform.illinois.edu/Papers/USAN_papers/15FED01.pdf

Feddersen, B., Keefe, Ken, Sanders, H. W., Muehrcke, C., Parks, D., Crapo, A., . . . Palla, R. (2015). *Enterprise Security Metrics with the ADVISE Meta Model Formalism.* University of Illinois. Retrieved from https://www.perform.illinois.edu/Papers/USAN_papers/15FED02.pdf

Ford, D. M., Keefe, K., LeMay, E., Sanders, H. W., & Muehrcke, C. (n.d.). *Implementing the ADVISE Security Modeling Formalism in Mobius.* University of Illinois. Retrieved from https://www.perform.illinois.edu/Papers/USAN_papers/12FOR04.pdf

LeMay, E., Ford, D. M., Keefe, K., Sanders, H. W., & Muehrcke, C. (2011). Model-based Security Metrics using ADversary VIew Security Evaluation (ADVISE). *Eighth International Conference on Quantitative Evaluation of SysTems*, (pp. 1-10). Retrieved from https://www.semanticscholar.org/paper/Model-based-Security-Metrics-Using-ADversary-VIew-LeMay-Ford/a06ac40aecca2ffa7646d6767a7906a983fabf83/pdf