2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing

Model-Based Cybersecurity Assessment with NESCOR Smart Grid Failure Scenarios

Sumeet Jauhar*, Binbin Chen*, William G. Temple*, Xinshu Dong*,
Zbigniew Kalbarczyk†, William H. Sanders†, David M. Nicol†

*Advanced Digital Sciences Center, Illinois at Singapore, Singapore
{sumeet.j, binbin.chen, william.t, xinshu.dong}@adsc.com.sg
†University of Illinois at Urbana-Champaign, IL, USA
{kalbarcz, whs, dmnicol}@illinois.edu

***Abstract*—The transformation of traditional power systems to smart grids brings significant benefits, but also exposes the grids to various cyber threats. The recent effort led by US National Electric Sector Cybersecurity Organization Resource (NESCOR) Technical Working Group 1 to compile failure scenarios is an important initiative to document typical cybersecurity threats to smart grids. While these scenarios are an invaluable thought-aid, companies still face challenges in systematically and efficiently applying the failure scenarios to assess security risks for their specific infrastructure. In this work, we develop a model-based process for assessing the security risks from NESCOR failure scenarios. We extend our cybersecurity assessment tool, CyberSAGE, to support this process, and use it to analyze 25 failure scenarios. Our results show that CyberSAGE can generate precise and structured security argument graphs to quantitatively reason about the risk of each failure scenario. Further, CyberSAGE can significantly reduce the assessment effort by allowing the reuse of models across different failure scenarios, systems, and attacker profiles to perform "what if?" analysis.**

I. Introduction

Modernized electric power grids or smart grids incorporate information and communication technologies for improving power system control, monitoring, and response. However, the rapid digitization also exposes smart grids to various cyber threats. Neutralizing cyber threats in smart grids has become an urgent task, both for utility companies testing and deploying security mechanisms in their systems, and for the research community developing advanced solutions to combat emerging cyber threats. As part of this process, (cyber)security and risk assessment have become essential to support secure system design and deployment [1], [2], [3].

The electric sector failure scenarios and impact analyses, compiled by the US National Electric Sector Cybersecurity Organization Resource (NESCOR) Technical Working Group 1 [4], are a prominent example of the push toward more comprehensive and rigorous security assessment in the industry. The NESCOR failure scenarios describe specific types of undesirable cyber incidents and their impacts, as well as the vulnerabilities and potential mitigations associated with the failures. These scenarios, created by a coordinated community effort from grid operators, security consultants and regulators, provide an invaluable thought-aid for utilities to map to their systems, and identify relevant or similar threats to be addressed. However, companies still face challenges to systematically and efficiently apply these failure scenarios to assess the security risks of their specific infrastructure setups.

Model-based security assessment methods can potentially be used to streamline this process. A significant amount of research effort has been devoted to formalize the description of systems, attackers, and other security-related information, and to automate the assessment processes as much as possible. However, there are a number of practical challenges in model-based security assessment, including *who creates the model?* and *how well does the model reflect the real system?* In the smart grid setting, we believe the NESCOR failure scenarios provide a promising foundation for model-based security assessment, because they describe realistic incidents that are of concern to the industry, and do so in a sufficient level of detail to allow model creation. However, to the best of our knowledge, no model-based security assessment tool exists today that can support the process of assessing a NESCOR failure scenario on a specific smart grid system.

In this work, we address this gap by integrating the NESCOR failure scenarios into a model-based security assessment process. We achieve this by extending and adapting the CyberSecurity Argument Graph Evaluation (CyberSAGE) software tool, which was developed in our previous research [5]. The goal of this work is two-fold: 1) to study the feasibility of applying a model-based approach to assess realistic threat scenarios in smart grids; 2) to demonstrate the benefits of having more structured, formalized, and mechanized approaches for assessing the security of smart grids.

To this end, we have studied one class of NESCOR failure scenarios (Distributed Energy Resources), and modeled them using an extended version of the formalism proposed in [6]. More specifically, we model each failure scenario using a mal-activity diagram similar to those proposed in [7], and formally represent the relevant vulnerabilities, mitigations, and attacker properties. The result of the security assessment is a *security argument graph* — a graphical representation that connects mal-activity process with relevant system components and threat agents. The *security argument graph* is generated automatically by our CyberSAGE tool based on the mechanisms introduced in [6]. The graph is also used to provide a quantitative risk assessment for the NESCOR scenario based on user-provided system and attacker properties. Our model-based assessment process and the accompanying CyberSAGE tool makes it effective to perform "what if?" analysis for varying system settings and attacker profiles. Furthermore, CyberSAGE can significantly reduce the analyst's assessment effort by allowing the reuse of models across different failure scenarios, systems, and attacker profiles.

978-1-4673-9376-8/15 $31.00 © 2015 IEEE
DOI 10.1109/PRDC.2015.37

II. Background

Cybersecurity for smart grid systems has become a major concern over the last few years and it has become essential for utility companies to understand the latest threats and conduct thorough risk assessment for their systems. In this section, we introduce the NESCOR failure scenarios for unfamiliar readers and discuss model-based security assessment, to provide a foundation for the subsequent discussion of model-based cybersecurity assessment with the failure scenarios.

A. NESCOR Failure Scenarios

NESCOR Technical Working Group 1 has spent years developing and refining a set of electric sector failure scenarios and impact analyses [4]. The NESCOR failure scenarios consist of 111 unique cyber-incidents that could negatively impact an electric utility and cover a number of smart grid domains:

- Advanced Metering Infrastructure (32 scenarios)
- Distributed Energy Resources (25)
- Wide-Area Monitoring, Protection, and Control (11)
- Electric Transportation (16)
- Demand Response (7)
- Distribution Grid Management (16)
- Generic (4)

Each NESCOR scenario consists of a written **description**, a list of **relevant vulnerabilities**, a list of **impacts**, and a list of **potential mitigations** [8] (see example in Section III). The scenarios are intended to help utility companies conduct risk assessment [9], as well as to improve cybersecurity awareness during procurement/planning and to assist with training. They have also drawn research interest [10], [11], [12].

While the types of failures considered in the NESCOR scenarios are far-reaching and clearly defined, the process of analyzing a specific system for a scenario-based risk assessment is effort intensive and largely subjective. New tools are needed to help utilities maximize the benefits of NESCOR's efforts. Such tools would reduce human effort, promote repeatability of results, and clarify the assumptions and information that serve as the starting point for an assessment. For those tasks, the field of model-based security assessment has much to offer.

B. Model-Based Cybersecurity Assessment

Over the years, the cybersecurity research community has been working to develop various model-based tools to support security and risk assessment. The model-based tools available today include UML-based approaches [13], petri-net-based approaches [14], attack-tree-based approaches [15], and hybrid approaches combining different inputs [5]. Regardless of the modeling formalism, such tools share many common objectives and features. For example, many support a quantitative evaluation of system security using various metrics. The stated aim is often to support decision making and various comparative analyses.

In this paper, we focus on the CyberSecurity Argument Graph Evaluation (CyberSAGE) approach and software tool. CyberSAGE utilizes a broader assortment of security-related input information than many tools, while automating certain aspects of the security assessment process. Together, those features lend themselves to the analysis of NESCOR failure scenarios, which is a largely manual process today. The CyberSAGE methodology uses workflow models to define the scope of the security assessment: for example, allowing a utility to assess the security of its smart meter reading processes [16], or a distribution grid measurement and control process [6]. The assessment connects different types of information, including:

- **Goal:** a system-level property or requirement for the specified workflow (e.g. availability).
- **Workflow:** a model of the actors and interactions occurring in the system (e.g., UML activity diagram).
- **System:** a model describing the system's devices, connections, and configurations.
- **Attacker:** a model describing the skills, resources, and knowledge of the attacker under consideration.

Each of these elements are combined to form a *security argument graph* that visually represents potential attacks on the system components implementing a workflow. The structure of this graph determines dependency relationships that can be used to calculate system-level metrics from low-level data. The graph itself is created automatically, provided the above inputs are present. This is done through a library of *extension templates* that determine how various pieces of information are connected. For example, there could be a rule connecting a workflow step by a certain type of actor (e.g., intelligent electronic device) to a specific device in the network (e.g., EV charging station). More details may be found in our previous work [6].

C. Other Approaches for Cybersecurity Assessment

There are a number of standards and guidelines that are relevant to security and risk assessment for smart grid systems. The National Institute of Standards and Technology (NIST) has released a couple of documents that provide comprehensive guidelines on the requirements for and properties of secure smart grid systems [3], [17]. It also highlights the importance and desirable goals of security and risk assessment. Similarly, the North American Electric Reliability Corporation (NERC) guidelines elaborate further on detailed aspects that a security assessment needs to cover for smart grid [18].

A number of industry standards and best practices, including those from NIST and NERC, may be assessed using the Cyber Security Evaluation Tool (CSET) [19] from ICS-CERT. This questionnaire-based tool helps with the generation of compliance documentation and helps to couple the general standards with a user-specified network diagram. Finally, more relevant to this paper, EPRI has developed a Microsoft Excel toolkit to support the evaluation of NESCOR failure scenarios [20]. We discuss the merits of our approach compared to the EPRI toolkit as part of our evaluation in Section IV.

III. From Scenarios to Model-based Assessments

While the NESCOR scenarios are an invaluable thought-aid and education tool, we see the potential benefits of systematizing the knowledge embedded in the NESCOR documents and using them as a basis to conduct model-based assessments. We develop an extension of our workflow-oriented security assessment approach [6], [16] to realize this vision. This section describes the steps and intuition behind our method.

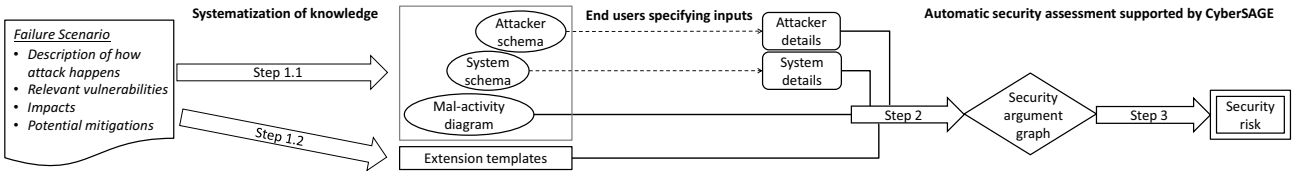

Fig. 1: The main steps of our approach in conducting model-based security assessment with NESCOR failure scenarios.

A. Running Example: NESCOR Scenario DER.1

We use the first distributed energy resource failure scenario (DER.1), *Inadequate Access Control of DER Systems causes Electrocution*, throughout the rest of this paper to illustrate our model-based security assessment method and results. We reproduce DER.1 below from [4] in slightly abridged form for readers' easy reference.

Description: *The DER owner fails to change the default password or not set a password for the DER system user interface. A threat agent (inept installer, hacker, or industrial spy) gets access through the user interface and changes the DER settings so that it does not trip off upon low voltage (anti-islanding protection), but continues to provide power during a power system fault.*

Relevant Vulnerabilities:

- Physical access may be obtained by unauthorized individuals to DER settings through the system UI
- Default password is not changed for the DER system
- System permits unauthorized changes to anti-islanding protection due to poor configuration design
- Commands or other messages may be inserted on the network between the user interface and the DER system, that result in unauthenticated changes to sensitive parameters

Impacts:

- System suffers physical damage due to feeding into a fault
- A utility field crew member may be electrocuted
- The utility experiences damage to its reputation

Potential Mitigations:

- Authenticate users for all user interface interactions
- Change default access credentials after installation
- Enforce limits in hardware to prevent equipment damage
- Train personnel on secure networking requirements
- Require management approval for critical security settings

B. Model-Based Security Assessment Process

Our approach models the information captured in NESCOR failure scenarios to assess the security level of a specified system against a specific attacker. Fig. 1 summarizes the main steps of our approach. We start by the systematization of the knowledge embedded in the failure scenarios into individual models (step 1.1) and their logical relationships (step 1.2). An end user then can specify the details about her system and the concerned attacker. The CyberSAGE tool then supports the automatic generation of a security argument graph (step 2) and quantitative evaluation of the security risk (step 3). We now discuss each step in more detail using the DER.1 example.

Step 1: Systematization of knowledge: This step consists of two sub-steps as described below.

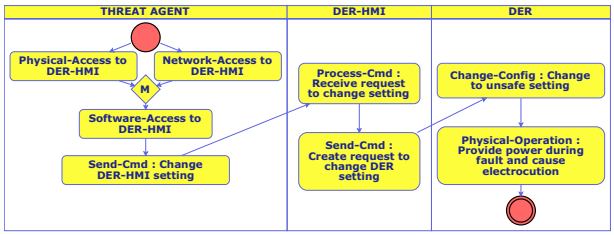

Fig. 2: DER.1 mal-activity diagram (CyberSAGE screenshot).

Vulnerabilities	Attacker property		p
Weak authentication	IT skills	H	0.8
		M	0.4
		L	0.1
Unchanged default password	IT skills	H	0.99
		M	0.95
		L	0.9

TABLE I: An example probability table (for *S*oftware Access).

Step 1.1 (Representation of information): This sub-step formalizes different aspects embedded in failure scenarios. Specifically, we represent the textual description of how a failure happens as a mal-activity diagram (similar to [7]). This is because the NESCOR descriptions are from an attacker's perspective and hence it is natural to model the malicious intentions of the attacker. However such a representation does not allow us to discover new attack paths into the system. We use the vulnerabilities and mitigations to define schemas of system/attacker properties. As we will show in Section IV, many properties defined in our schemas (like security controls of a device) are applicable across multiple failure scenarios. We also associate the impacts with the mal-activity diagram. CyberSAGE supports the creation of mal-activity diagrams, and the customization of system and attacker schemas.

For the DER.1 example, we begin by developing a mal-activity diagram (see Fig. 2) to show how a threat agent's actions may influence the normal processes in the system. The threat agent first tries to gain either physical or network access to the DER-HMI, then changes the settings. This malicious activity results in the system taking an unsafe action. We also use the scenario description to identify key system components (e.g., DER-HMI and DER equipment). We associate each type of system with a schema of properties, which are derived based on the vulnerabilities and mitigations in the NESCOR document. Example properties include *restrict-physical-access*, *authenticate-user*, etc. We also define an attacker schema to capture attacker properties that affect the outcome of individual attack steps, such as *access*, *domain knowledge*, *IT skills* and *malicious intention*.

Step 1.2 (Extension templates): After formalizing different aspects of information (mal-activity diagram, system, and attacker schemas), we must systematize the logical relation-

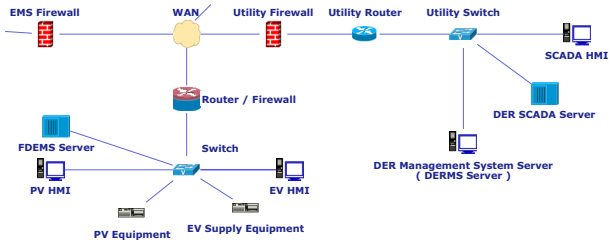

Fig. 3: System topology modeling (CyberSAGE screenshot).

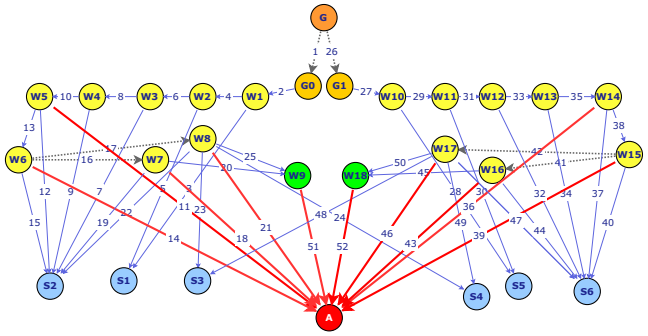

Fig. 4: A security argument graph generated by CyberSAGE, for assessing DER.1 over example system of Fig. 3.

ships among them. This is done using *extension templates* as we defined in our previous work [6]. Intuitively, an extension template is a formal reusable rule for connecting a security-related statement or claim with relevant supporting arguments, which also carries the logic about how numerical evidence associated with supporting arguments affect metrics associated with the higher level statement. Our CyberSAGE tool supports the description of extension templates as rule files.

We have studied all 25 DER examples, for which we look beyond the templates from our previous work [6] and define 15 new extension templates. For the DER.1 example, 7 out of these 15 are applicable: *Software Access*, *Network Access*, *Physical Access*, *Send Command*, *Process Command*, *Change Configuration* and *Physical Operation*. As an example, the logic in the *Software Access* extension template specifies that *access to software on a specific device* depends on: 1) its previous activity step, i.e., access to the corresponding device (either physically or through network), and 2) the relevant system properties (e.g., the access control mechanisms for the particular software), and 3) the requirement for the attacker (e.g., access to credential information and IT skill level).

Our *Software Access* extension template also quantifies the probability of exploiting vulnerabilities to gain software access as a function of system and attacker properties. For example, Table I shows that an attacker can gain software access by exploiting either *weak authentication* or *unchanged default password*. It specifies different values of probability p for successfully exploiting some vulnerability, given a certain attacker property. The example values in the table show that an attacker with high IT skill can exploit the weak authentication with probability $p = 0.8$, while an attacker with low IT skill can only do so with probability of $p = 0.1$. On the other hand, exploiting the *unchanged default password* vulnerability is less sensitive to an attacker's IT skill level. In the example, $p \geq 0.9$ even for an attacker with low IT skill.

Step 2: Security argument graph generation: The first step distils the knowledge embedded in a failure scenario by breaking different pieces apart and identifying their logical relationships. Step 2 uses this information to examine a specific system, by creating a security argument graph that connects security-related information to reason about the security level of the studied system. As opposed to the examples in our earlier work [6], [16] where the security argument graph is to argue about a positive goal (e.g., system availability), in the NESCOR failure scenario analysis we use the threat agent's goal (i.e., to cause the failure to occur) as the focal point to generate the security argument graph. To generate the security argument graph, end users need to first specify the details about their system (see an example system in Fig. 3) based on the schema from step 1.1 and also indicate how the actors in the mal-activity diagram (Fig. 2) map to the components in the system topology (Fig. 3). Based on these inputs, the CyberSAGE tool can automatically apply suitable extension templates from step 1.2 in a recursive manner to integrate the mal-activity diagram, system, and attacker information and generate a security argument graph.

In the following, we consider DER.1 over the example system in Fig. 3, which consists of a photovoltaic (PV) generator and an electric vehicle (EV) charging station. Our assessment aims to examine the risk of failure in either system. We use the CyberSAGE tool to map the mal-activity actors DER-HMI and DER to two pairs: PV HMI and PV equipment, and EV HMI and EV equipment, respectively. We also map the initial starting position of attacker to WAN.

Fig 4 shows the generated security argument graph. From top to bottom, the G nodes represent the attacker's goal, the yellow W nodes correspond to steps in the mal-activity diagram (with the special green nodes indicating the starting states), the blue S nodes represent the system components, and finally the A node represents the attacker.

As shown, the goal node G (the failure scenario occurs) holds if either $G0$ or $G1$ holds, which are two sub-goals that correspond to the failure of the PV and EV system respectively. This OR relationship is represented by the dotted edges 1 and 26. We follow the $G0$ branch (PV system) to explain the graph. For attack to accomplish goal $G0$, the last activity step, i.e., physical-operation at DER device (see Fig. 2), as represented by node $W1$ should occur. Node $W1$ depends on its previous activity step (i.e., change-config at DER, represented by node $W2$) and properties of the corresponding component (the PV equipment, represented by node $S1$). By automatically and repeatedly applying suitable extension templates, the CyberSAGE generated graph further traces back to earlier steps in the mal-activity diagram. Note that node $W6$, which represents software-access to PV HMI, depends on the accomplishment of either physical access step (node $W7$) or network access step (node $W8$) over PV HMI. Node $W6$ also depends on specific properties (e.g., default password vulnerabilities) of PV HMI (node $S2$) and specific properties (e.g., IT skill) of the attacker (node A), the latter dependency is highlighted by the red edge 14. With CyberSAGE tool, one can click on each node or edge to show and edit corresponding detailed properties, such as relevant system vulnerabilities, required attacker properties, etc. For node $W8$ (network-access to PV HMI), since we assume the attacker's entry point is from

external network (WAN), CyberSAGE automatically includes critical components (i.e., router / firewall as represented by node $S3$, and switch as represented by node $S4$) on the network path into the generated security argument graph. Finally, the edges from the green nodes $W9$ and $W18$ to node A capture the attacker's intention to initiate such an attack.

Step 3: Security metrics calculation: The final step of our security assessment method uses the generated security argument graph to quantitatively evaluate the risk of the NESCOR failure scenario. The evaluation is based on the logical relationships among different nodes, as defined by the corresponding extension templates. Specifically, each mal-activity step in the graph will be associated with some probability that is computed based on both the relevant vulnerabilities and the concrete properties of the concerned attacker (see Table I for an example). CyberSAGE then aggregates the probability values for these individual events through the logical operators AND/OR to calculate the value for the goal node in the graph.

We specified an example set of quantitative relationships in our extension templates. In Section IV we provide more details about DER.1 evaluation results based on our example extension templates. A security analyst can customize the evaluation logic by editing corresponding extension templates.

IV. Evaluation

This section reports the overhead (or the effort required) and benefits of applying our model-based assessment method with NESCOR failure scenarios. Our evaluation is based on an extended version of our CyberSAGE tool, which allows automatic generation of security argument graphs and automatic evaluation of security metrics like failure probabilities.

A. Effort Required to Apply a Model-based Approach

With the support of our CyberSAGE tool, we have modeled all 25 NESCOR failure scenarios under the DER category. As discussed in Section III (see Fig. 1), applying our approach requires manual effort to model the mal-activity diagrams and system / attacker schemas (step 1.1), and to distil the extension templates (step 1.2). One researcher spent around 1 hour to formalize each of the 25 different mal-activity diagrams based on the description of each individual scenario. Also we distil the system components involved in these scenarios into 10 different types, including DER device, DER Human-Machine-Interface (HMI), server, networking devices, etc. We define for each type a corresponding schema of security related properties (e.g., access control, authentication, etc.), based on the vulnerabilities and potential mitigations mentioned in the scenarios. In total, less than 100 distinct properties are needed to model all devices mentioned in the 25 scenarios. Since the same system and attacker templates are shared across different scenarios, the modeling effort is effectively amortized.

After we had all 25 mal-activity diagrams in place, we conducted a 2-day internal workshop, identifying common attack steps shared by different scenarios and creating extension templates to describe them. We find that we only need 15 extension templates to cover all of the 25 DER failure scenarios. This is because many failure scenarios share similar types of attack steps (e.g., gaining network access, sending malicious commands). Furthermore, we find that most of the 15 extension templates we identified can be readily applied to failure scenarios in other NESCOR failure scenario categories, such as those related to Advanced Metering Infrastructure (AMI). Due to space limitations, we report the details of our 15 extension templates in our technical report [21].

	IT Skill	Domain knowledge	Access	Probability to launch (C / I / A) attack
Inept installer	Low	Low	Physical	0.1 / 0.1 / 0.1
Hacker	High	Medium	Remote	0.6 / 0.9 / 0.9
Industrial spy	High	High	Remote	0.9 / 0.6 / 0.6

TABLE II: Main properties of three attacker profiles.

Once the above knowledge systematization step has been done, an end user only needs to specify the details of her concerned system and attackers based on the defined schema. By following our method, this becomes a one-time effort, since the user inputs can be reused across different DER failure scenarios. Finally, once the user specifies the inputs, CyberSAGE automates the remaining steps, i.e., the generation of the graph and the evaluation of the security metrics.

Compared to the CyberSAGE tool, the EPRI Excel toolkit [20] places greater reliance on the analyst's judgment when evaluating the risks from failure scenarios. The toolkit allows the analyst to go through each failure scenario individually and rate the severity of vulnerabilities (low/moderate/high), the implementation level of mitigations (not/partially/largely/fully implemented), and threat and impact scores (on a scale of 0,1,3,9). Based on the total number of likelihood and impact points, the analyst then manually assigns a risk score of high/moderate/low. Since there is no explicit data input from or modeling of the system being assessed, the time needed in the EPRI Excel toolkit for assessing an individual scenario is similar or potentially less than CyberSAGE. However, when scaling up to dozens or hundreds of scenarios, a CyberSAGE assessment can be less ambiguous, more consistent, and simultaneously more efficient, since it supports longer-term reduction in effort from prior systematization.

B. Assessment under Varying System and Attacker Settings

To provide a concrete example of the potential benefits from our approach, we describe how a security analyst can use CyberSAGE to study the probability for failure scenario DER.1 to occur under different system settings and different attacker profiles. In CyberSAGE each mitigation can independently reduce the probability of successfully exploiting the corresponding vulnerability. We start with a baseline setting where NESCOR mitigations for DER.1 are not in place and then progressively apply them one by one where each mitigation can reduce the probability of exploiting the corresponding vulnerability by 99%. We model three attacker profiles—inept installer, hacker, and industrial spy—as mentioned in the DER.1 scenario description. Table II summarizes the main properties of the three attacker profiles, including their level of IT skill and domain knowledge, their possible access levels to the DER system and their malicious intention, as described by the probability of launching attacks impacting confidentiality/integrity/availability. All the parameters specified in Table II can be fine-tuned by a security analyst.

CyberSAGE supports quantitative evaluation of the probability for a failure scenario to occur. This can be extended

	Inept installer	Hacker	Industrial spy
Baseline	1.5×10^{-2}	4.1×10^{-1}	4.2×10^{-1}
Baseline + Authenticate Users	9.3×10^{-3}	3.1×10^{-1}	3.7×10^{-1}
Baseline + Change default password	9.3×10^{-3}	3.1×10^{-1}	3.7×10^{-1}
Baseline + Enforce hardware limits	1.4×10^{-3}	3.0×10^{-2}	2.9×10^{-2}
Baseline + Secure network training	1.5×10^{-2}	1.0×10^{-2}	1.4×10^{-2}
Baseline + Require manager approval	5.5×10^{-3}	2.5×10^{-1}	3.4×10^{-1}
Baseline + All 5 mitigations	3.0×10^{-4}	3.1×10^{-4}	5.9×10^{-4}

TABLE III: Probability for DER.1 failure scenario to occur.

to a risk score for the failure scenario by multiplying the failure probability by an impact score provided during model creation. A user can easily vary system settings and attacker properties in CyberSAGE. The assessment result depends on the structure of the security argument graph, the system and attacker inputs, and the probability tables embedded in the relevant extension templates. Due to space limitations, we only provide an intuitive interpretation of the results in this section. Our technical report contains more detail about the system settings, attacker properties, extension template coefficients, including a preliminary sensitivity analysis of the input values.

As shown in Table III, under the baseline setting, for both the industrial spy and hacker profiles, there is a high probability that the attacker can break into the system by exploiting network-related vulnerabilities, ultimately causing failure DER.1 to happen. Note that, although an industrial spy is more capable than a hacker (as she has better domain knowledge and similar IT skill), her intent to cause an integrity-related failure scenario like DER.1 is lower than a hacker's. These factors are aggregated through the security argument graph and result in similar failure probabilities under the two attacker profiles.

The table also shows that CyberSAGE can differentiate among the impacts of various mitigations for different types of attackers. For example, the mitigation of enforcing hardware limits on the DER device has a bigger impact on preventing DER.1 failure scenario against all attackers. In comparison, training the personnel on secure networking requirements can reduce the failure probability for both the hacker and industrial spy settings, but not for the inept installer setting. This is because the installer gains software access through physical means instead of through network. The last row of the table shows that applying all mitigations on the system results in an even lower chance that the failure scenario can occur. Since applying all mitigations can be expensive, an analyst can use CyberSAGE to study the failure probability under different subset of relevant mitigations to identify more effective mitigations.

V. Conclusion

In this paper, we develop a model-based method for assessing the security risks of NESCOR failure scenarios. We extended our model-based security assessment tool CyberSAGE, to demonstrate the effectiveness of our proposed method. Incorporating realistic failure scenarios into model-based security assessment tools can aid security analysts to systematically and efficiently assess the security level of their systems. We will make our failure scenario models available through CyberSAGE tool portal [22]. We hope our work will facilitate the adoption of model-based security assessment in the smart grid industry.

Acknowledgments

This work is supported by Singapore's Agency for Science, Technology and Research (A*STAR) under a research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center. We also thank Prageeth Gunathilaka, Li Yuan and Qi Qu for their valuable contributions in tool development.

References

[1] ENISA, "Appropriate security measures for smart grids," 2012.

[2] R. Habash, V. Groza, D. Krewski, and G. Paoli, "A risk assessment framework for the smart grid," in *IEEE EPEC*, Aug. 2013.

[3] NIST: The Smart Grid Interoperability Panel - Cyber Security Working Group, "NISTIR 7628, Guidelines for Smart Grid Cyber Security, Revision 1," 2014.

[4] National Electric Sector Cybersecurity Organization Resource, "Electric sector failure scenarios and impact analyses," Electric Power Research Institute, Tech. Rep. 2.0, Jun. 2014.

[5] A. H. Vu, N. O. Tippenhauer, B. Chen, D. M. Nicol, and Z. Kalbarczyk, "CyberSAGE: A tool for automatic security assessment of cyber-physical systems," in *QEST*, Sep. 2014.

[6] N. O. Tippenhauer, W. G. Temple, A. H. Vu, B. Chen, D. M. Nicol, Z. Kalbarczyk, and W. Sanders, "Automatic generation of security argument graphs," in *PRDC*, Nov. 2014.

[7] G. Sindre, "Mal-activity diagrams for capturing attacks on business processes," in *REFSQ*, Jun. 2007.

[8] National Electric Sector Cybersecurity Organization Resource, "Electric sector failure scenarios common vulnerabilities and mitigations mapping," Electric Power Research Institute, Tech. Rep., Jun. 2014.

[9] A. Lee, "Integrating electricity subsector failure scenarios into a risk assessment methodology," Electric Power Research Institute, Tech. Rep., Dec. 2013.

[10] R. Abercrombie, B. Schlicher, and F. Sheldon, "Security analysis of selected ami failure scenarios using agent based game theoretic simulation," in *HICSS*, Jan. 2014.

[11] R. Berthier and W. H. Sanders., "Monitoring advanced metering infrastructures with amilyzer," in *C&ESAR*, Nov. 2013.

[12] C. Hawk and A. Kaushiva, "Cybersecurity and the smarter grid," *The Electricity Journal*, vol. 27, no. 8, pp. 84 – 95, 2014.

[13] T. Sommestad, M. Ekstedt, and H. Holm, "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," *Systems Journal, IEEE*, vol. 7, no. 3, Sep. 2013.

[14] E. LeMay, M. Ford, K. Keefe, W. Sanders, and C. Muehrke, "Model-based security metrics using ADversary VIew Security Evaluation (ADVISE)," in *QEST*, Sep. 2011.

[15] B. Kordy, L. Pietre-Cambacedes, and P. Schweitzer, "DAG-based attack and defense modeling: Don't miss the forest for the attack trees," *CoRR*, vol. abs/1303.7397, 2013.

[16] B. Chen, Z. Kalbarczyk, D. M. Nicol, W. H. Sanders, R. Tan, W. G. Temple, N. O. Tippenhauer, A. H. Vu, and D. K. Yau, "Go with the flow: Toward workflow-oriented security assessment," in *NSPW*, 2013.

[17] NIST: Joint Task Force Transformation Initiative, "Nist special publication 800-53, security and privacy controls for federal information systems and organizations, revision 4," 2013.

[18] The North American Electric Reliability Corporation (NERC), "Security guidelines for the electricity sector: Vulnerability and risk assessment, version 1.0," 2012.

[19] ICS-CERT, "CSET: The cyber security evaluation tool."

[20] EPRI, "Failure scenario based risk assessment toolkit for the electricity subsector," June 2014, prototype Version 0.3.

[21] "CyberSAGE NESCOR Tech Report," https://www.illinois.adsc.com.sg/papers/CyberSAGE_NESCOR_TechReport.pdf, Tech. Rep.

[22] "CyberSAGE," https://www.illinois.adsc.com.sg/cybersage/.