Developing Models for Physical Attacks in Cyber-Physical Systems

Carmen Cheh
Department of Computer Science
University of Illinois
Urbana, Illinois
cheh2@illinois.edu

Ken Keefe
Information Trust Institute
University of Illinois
Urbana, Illinois
kjkeefe@illinois.edu

Brett Feddersen
Information Trust Institute
University of Illinois
Urbana, Illinois
bfeddrsn@illinois.edu

Binbin Chen
Advanced Digital Sciences Center
Singapore
binbin.chen@adsc.com.sg

William G. Temple
Advanced Digital Sciences Center
Singapore
william.t@adsc.com.sg

William H. Sanders
Department of Electrical and
Computer Engineering
University of Illinois
Urbana, Illinois
whs@illinois.edu

ABSTRACT

In this paper, we analyze the security of cyber-physical systems using the *ADversary VIew Security Evaluation* (ADVISE) meta modeling approach, taking into consideration the effects of physical attacks. To build our model of the system, we construct an ontology that describes the system components and the relationships among them. The ontology also defines attack steps that represent cyber and physical actions that affect the system entities. We apply the ADVISE meta modeling approach, which admits as input our defined ontology, to a railway system use case to obtain insights regarding the system's security. The ADVISE Meta tool takes in a system model of a railway station and generates an attack execution graph that shows the actions that adversaries may take to reach their goal. We consider several adversary profiles, ranging from outsiders to insider staff members, and compare their attack paths in terms of targeted assets, time to achieve the goal, and probability of detection. The generated results show that even adversaries with access to noncritical assets can affect system service by intelligently crafting their attacks to trigger a physical sequence of effects. We also identify the physical devices and user actions that require more in-depth monitoring to reinforce the system's security.

KEYWORDS

cyber-physical systems; physical attack; attack graph; ontology

1 INTRODUCTION

In order to analyze the security risk of cyber-physical systems, designers have often focused on modeling the cyber aspect of such systems by enumerating the possible attacks that can be used to compromise the system and cause a cascading impact in the physical world. However, physical attacks on cyber-physical systems that target devices are just as concerning. The Metcalf power station sniper attack [16, 19] is a prime example of such physical attacks. Further, physical attacks coupled with a synchronized cyber effort are commonly overlooked by threat analyses. Therefore, in this paper, we focus on modeling of physical attacks and their consequences, and apply our approach to a railway station case study to provide insights into the security of the system.

The *ADversary VIew Security Evaluation* (ADVISE) meta modeling approach [4] uses a variant of attack graphs to perform a formal, repeatable, and auditable analysis of system designs. Attack graphs are used to analyze the system by enumerating all the possible attack steps that an adversary may take to achieve a goal [9, 10]. However, traditional attack graphs do not evaluate useful probabilistic metrics like the most common attack vector for a given adversary, expected time to complete an attack, and expected cost to the defender. The ADVISE approach builds a richer attack graph, termed the *attack execution graph* (AEG), which models additional details about attacks in order to facilitate a more comprehensive and quantitative analysis of an adversary's attack path.

The process of building such attack graphs, however, requires a lot of manual effort by an expert practitioner. The practitioner needs to enumerate the possible attack steps for each system component and connect those attack steps together. To reduce the practitioner's effort, the ADVISE meta modeling approach relies on an ontology that represents a high-level system model using familiar atomic elements and semantic relationships among them. The ontology allows the practitioner to specify attack steps just once during the construction of the ontology; subsequently, the only changes that a practitioner needs to make are updates to the system model instance. The attack graph is automatically generated by reasoning on the system model instance that is based on the ontology [4].

In this work, we constructed an ontology of cyber-physical system components and relationships, and a set of atomic attack steps that apply to those ontology elements. This ontology is the first step towards formally defining a library of physical attacks that can be reused by other system case studies. We then evaluate the effects of

CPS-SPC'17, November 3, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5394-6/17/11…$15.00
https://doi.org/10.1145/3140241.3140249

physical attacks on a railway transit system by using the ADVISE meta modeling approach. Given the cyber-physical architecture of our studied system, we constructed a system instance diagram composed of instances of components and relationships from the ontology. We used the ADVISE Meta tool to reason on that system instance diagram and automatically generate an AEG [12, 13]. We also specified information about the threat models and quantitative metrics that we want to calculate. Given that information, the tool simulates an adversary's decision-making and outputs the calculated metrics. From the simulation results, we identify (1) which adversaries pose the biggest threat to the system, (2) which attacks are most likely to happen, and (3) what additional defenses should we deploy to identify and prevent certain attacks.

The outline of the rest of the paper is as follows. We explain the ADVISE meta modeling approach and the railway transit system case study in Section 2. In Section 3, we describe the ontology and other input components into the ADVISE Meta tool. Section 4 describes the system evaluation performed using the model described in Section 3. In Section 5, we detail the related work in the domain of cyber-physical system modeling for security analysis. Section 6 describes our plans for future work. We conclude in Section 7.

2 PRELIMINARIES

In this section, we give an introduction to the ADVISE tool and its evolution. We also describe our case study of the railway transit system that we use in this paper.

2.1 ADVISE Meta Modeling Approach

The ADVISE formalism provides an executable, quantifiable security model in the form of an *attack execution graph* (AEG). In addition to the typical attack graph, the AEG includes details about the attack in terms of its cost, time to complete, probability of success, and probability of detection. In recent years, the ADVISE tool has been extended to include a meta model [4], which contains a higher-level system diagram, adversary profiles, and security metrics. From an ADVISE meta model, the ADVISE Meta tool generates an AEG using an ontology of components, relationships, and atomic attack steps. The ontology serves as a knowledge base that describes the system entities and the definitions of attacks. Unlike vulnerability databases, ontologies allow practitioners to automatically derive where and how attacks may be applicable to a given system. The ADVISE Meta tool uses the ontology to reason about the entities in the system diagram to generate an AEG. In the next section, we will describe the components of the ADVISE meta model and how they are combined together to create the AEG.

2.2 Case Study: Railway Transit System

Attacks on railway transit systems can have significant impact, ranging from loss of service to derailment. For example, a teenager once rewired a television remote control to communicate with wireless switch junctions, causing the derailment of a train and injury of twelve people [1]. Railway staff also pose a risk to the system's security. That was illustrated by the 2006 case in which traffic engineers hacked into a Los Angeles signal system, causing major traffic disruption [6].

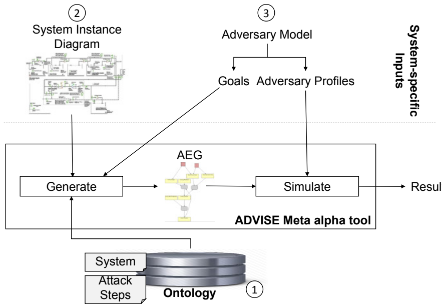

Figure 1: The general modeling approach. The numbered bubbles represent the order in which we describe the different inputs to the ADVISE Meta tool.

We have gained deep knowledge and understanding about the physical security challenges faced by railway transit system operators through a project partnership with a national transportation provider. In this paper, we focus on the physical security of a railway station. The railway station is a set of spaces containing devices. The devices control the movement of trains (e.g., signaling, traction power) and people (e.g., escalators, roller shutters). Other devices control the environment of the station and track (e.g., fan, water chiller). The environment of a space also affects the functioning of devices and movement of the people. For example, high temperatures can cause a device to overheat and trip, or cause a fire that will result in evacuation of the commuters.

The two main assets that an adversary would target are the trains and the commuters within a station. For example, affecting the movement of trains includes delaying trains to affect service revenue, and acceleration of trains to endanger human lives. Targeting the movement of people can cause mass evacuation and chaos, which can lead to significant injuries caused by a panicking crowd. Affecting the system environment might consist of creating fire or smoke within the station or tunnel. In the next section, we will define the adversary goal more specifically by relating it to properties of specific devices.

Our threat model involves the adversary's moving to rooms in the station to access devices. The person then performs actions on the devices. In the next section, we describe these actions as *attack steps* that are formally defined in an ontology. Within the same threat model, we can define different types of adversary profiles based on their skills, their system access, and their preferences for remaining undetected. We describe the different adversaries in more detail in the next section.

3 MODELING APPROACH

The central component of our approach uses the ADVISE Meta tool which takes in a number of user-specified inputs, and outputs an AEG. Figure 1 illustrates this process. First, we construct an ontology that models (i) the cyber-physical system entities and relationships, and (ii) the attack steps that can be performed on those entities. This ontology is generic and can be applied to any cyber-physical system. Next, we construct a system instance diagram that describes the specific case study in terms of the types of entities and relationships that are present in the railway station. Those types of entities and relationships are given by the ontology. Then, we define the threat model by enumerating the adversary profiles and defining the adversary goals. The tool then uses the ontology to reason on the system instance diagram, generating an AEG that leads to the specified goals. It also simulates the decision-making of each adversary profile by using a discrete-event, game-theory-based algorithm, and outputs the chosen attack path. We analyze those attack paths to obtain insights regarding the system security. We now describe each of the user-specified inputs.

3.1 Ontology

The ontology consists of two components: (1) a representation of cyber-physical systems, and (2) a library of attack steps that apply to that representation. We first start with a general conceptual understanding of the systems that we want to model, that will then drive the creation of classes and relations in the ontology.

Conceptually, we describe the system in terms of three non-disjoint domains: *physical*, *cyber*, and *information*. We distinguish among the three domains because they represent the levels at which human cognition perceives information [5]. The physical domain consists of devices and spaces. Devices are located in spaces. The cyber domain consists of hosts and network topology. Correspondingly, the hosts are devices (physical domain) that contain control software that transmit and receive digital data, and the network topology consists of cables connecting hosts that allow the transmission of bits. Lastly, the information domain consists of digital data such as control data and logs. These digital assets are represented as bits of information. By describing the system using the three domains, we can define actions (attack steps) that affect the system at different levels of abstraction. We can also drill down into low-level details about the effects of those actions.

We translate our general conceptual understanding into the ontology, shown graphically in Figure 2. The ontology defines component classes and kinds of relations that are definable between instances of the classes. The component classes use inheritance to describe specialization of types. The ontology's inheritance allows us to automatically infer information regarding child classes by deductive reasoning. More specifically, child classes inherit relationships and properties (attacks) from their parent classes [11].

The ontology also consists of generic attack steps that can be applied to the classes. By defining a library of attack steps, we use the ontology to reason on a system instance diagram and automatically enumerate which attack steps can be applied to the system entities. In particular, we use the ontology's multiple inheritance feature to derive all the entities that are vulnerable to a specific attack. We illustrate this feature with a small example. In our ontology in Figure 2, we can define an attack **A** that is applicable to any physical device. We model this by attaching **A** to the class **Device**. From the inheritance structure, this implies that **A** is applicable to all subclasses of **Device**, including **ProgrammableDevice** (or cyber hosts). This simplifies the job of creating an attack graph, since the practitioner does not need to recall that a host is also a **Device** and thus susceptible to attacks on physical devices. The ontology thus allows us to generate a complete AEG with respect to the defined attack steps, removing the chance for manual human error.

In Table 1, we define attack steps that can be applied at the level of abstraction corresponding to each of the three domains that we defined earlier. Some of the attack steps (**Overheat**, **PowerOff**, **AffectsPower**, and **AffectsTemperature**) are not actual attacks performed by an adversary, but instead model the physical consequences of certain actions.

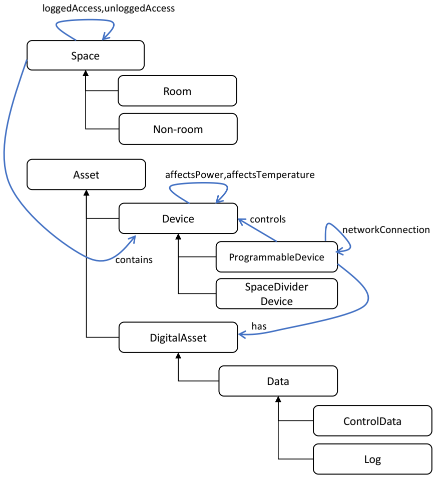

Figure 2: Our constructed ontology. Black boxes are component classes. Arcs with filled arrowheads represent inheritance, and labeled arcs are relationships.

Table 1: List of attack steps defined in ontology

Domain	Attack Step	Ontology Class
Physical	Move	Space
	Damage	Device
	DirectControl	Device
	Overheat	Device
	PowerOff	Device
	AffectsPower	Device
	AffectsTemperature	Device
Cyber	Login	ProgrammableDevice
	RemoteControl	Device
	NetworkMove	ProgrammableDevice
Information	DelMovementToken	Data
	AddMovementToken	Data

3.2 System Instance Diagram

We constructed a *system instance diagram* that represents our railway station case study. The system architecture and the relations between the devices are modeled using the constructs we defined in our ontology. We chose a set of representative rooms shown below:

- **PSC:** Houses cyber hosts that station operators use to monitor and control the devices in the station.
- **Server Room:** Houses the *Programmable Logic Controller*s (PLCs) that control the operation of other devices, and the *Building Access Control* (BAC) server that manages accesses to the doors and roller shutters.
- **Traction Power (Power Room):** Houses the equipment that controls the power to the track.
- **Water Chiller (Environment Control Room):** Houses equipment that controls the temperature in the station.

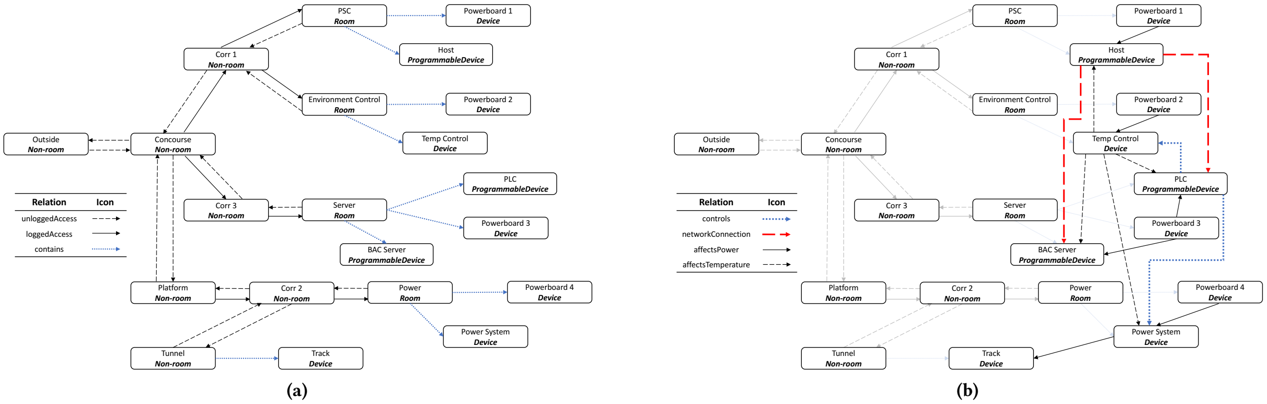

Figure 3: Two views of the system instance diagram. The boxes represent the entities in the railway station. Each box has a unique name and belongs to a class in the ontology. (a) The physical station architecture. (b) The relations between the devices.

Table 2: The access elements possessed by different adversary profiles at the start of the simulation.

			Physical Access							Asset Access								
			Corr 1	PSC	Env Ctrl Rm	Corr 3	Server Rm	Corr 2	Power Rm	PB 1	Host	PB 2	Temp Ctrl	PB 3	PLC	BAC Server	PB 4	Power Sys
Outsider																		
Privileged Insider			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Insider	Cleaner		✓	✓	✓	✓	✓	✓	✓									
	Station Op		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓			✓
	Maintenance Staff	Env Ctrl	✓		✓							✓	✓					
		Power						✓	✓								✓	✓
		Access Ctrl				✓	✓									✓		
		Server				✓	✓							✓	✓			

3.3 Adversary Profiles and Goals

We now define the adversary profiles and their goals. Each attack step requires some preconditions to be satisfied before it can be attempted, e.g., the adversary's skill in an activity, or the adversary's access permissions to a system component. So depending on the adversary profile, some of the attack steps that we defined earlier can or cannot be attempted. The accesses that an adversary may have are (1) permissions to perform the **Move** attack step to enter a **Space** , and (2) permissions to perform the **DirectControl**, **RemoteControl**, **Login**, or **NetworkMove** attack steps to control an **Asset**. We define three types of adversary profiles: an **Outsider** adversary with skills to break door access locks, a **PrivilegedInsider** adversary who has gained access to all types of legitimate access cards, and an **Insider** adversary. Table 2 shows the accesses that those adversaries possess.

We define three goals that the adversary wants to achieve: (1) killing power to the railway track (**TrackPowOff**), (2) locking the station exits (**LockExits**), and (3) controlling the temperature in the station (**TempCtrlOff**). The first goal affects the movement of trains (`Track`) and thus disrupts system service. The second goal affects the commuters' movements (`BAC Server`). Finally, the third goal affects the station's environment (`Temp Control`).

4 EVALUATION

We input our system instance diagram and adversary models into the ADVISE Meta tool, which uses our defined ontology to reason on the inputs to generate an AEG. We use the tool to simulate the actions taken by an adversary profile and calculate metric values for that set of actions. The metrics we define are (i) the total cost of performing attacks, (ii) the number of attack steps needed to accomplish a goal, and (iii) the fraction of attack steps that are likely to be detected. These metrics describe the different tradeoffs that an adversary will make when attacking the system. We analyze the resulting metric values to obtain insights regarding (i) the possible sequence of actions taken by an adversary, (ii) the assets that require further protection, (iii) the user actions that require further monitoring, and (iv) the set of adversaries that pose the highest risk to the system's security.

Experiment Setup. For each attack step, we defined (1) the cost for the adversary to attempt the attack step and (2) the probability of the attack step's being detected by a defender. These settings will impact the adversary's decision-making process when choosing between different attack paths. The cost of the attack step is quantified in terms of the expert knowledge and physical effort required to execute the attack. The cost is an integer score ranging from 1 to 10, where 1 represents the least effort and 10 represents the most effort. The probability that an attack will be detected by a defender is a floating value that ranges from 0 to 1. We assume that as a preliminary defense, the defender can monitor the devices through CCTV cameras in the rooms, and the state of the commuters by direct observation. Based on our understanding of the system and possible attacks, we set the parameters as shown in Table 3.

Table 3: The cost incurred by an adversary and detection probability of each attack step. Damaging powerboards have a cost of 8, and damaging other devices have a cost of 10.

	Move	Damage	DirectControl	RemoteControl	Login	AddMovement	DelMovement
Cost	1	8 or 10	5	4	3	5	5
Detection Prob	0	1	0	0	0	1	1

Table 4 shows the resulting metrics for each adversary profile (with the exception of the **PrivilegedInsider**). We discuss the insights garnered from the results, identifying areas in the system that require additional defenses.

Attack Paths. We can see from the simulation results that although the building access control system allows staff to move only into rooms within their job roles, staff are still capable of affecting other devices in different rooms to achieve the goal. In particular, the environment control staff can cut power to the track despite not having access to the power room or the server room that controls the device logic. This attack path may have been easily missed by a human analyst who did not consider the physical consequences of changing a device's temperature. Thus, it is important not to overlook noncritical staff members or assume that a malicious action will remain contained within a room. The effects of the noncritical staff member's actions, however, take much longer to propagate through the system than those of other adversaries. So the practitioner has more time to catch the attack before the adversary's goal is achieved. Thus, it is crucial to implement detection mechanisms that monitor changes in physical processes.

Attack Steps. We find from our simulation results that the **Damage** action is the most commonly used one by all the adversary profiles and often targets powerboards. Since the outsider, cleaner, and station operators have physical access to all rooms, they can perform the **Damage** action on the appropriate device to achieve any goal. Thus, the cost to damage equipment, specifically powerboards, should be increased via additional physical guards. We also note that none of the insiders were detected unless they physically damaged equipment. So a specialized detection mechanism is needed to distinguish between malicious actions and normal actions. In particular, we need to monitor physical movements, host logins, and actions performed using the control software.

Adversary Stealth. During an adversary's decision-making process, he or she may have a choice of several attack paths. An adversary chooses one of the attack paths based on his or her preference of either minimizing the cost of the attack path or reducing the risk of being detected by a defender. We want to investigate which attack paths (that correspond to using different access cards) an adversary will take, given differing levels of stealthiness.

In our previous work [2], we developed an intrusion detector for malicious movement that raises an alarm when a logged movement deviates from the normal behavior associated with the access card used. In that work, we simulated malicious movement for various insiders and calculated the percentage of malicious movements that were detected. That percentage is used as an estimation of the detection probability of the **Move** attack step for different maintenance staff. In particular, the probability of detecting malicious movement by server staff, environment control staff, and power staff are 0.5, 0.6, and 0.95 respectively.

We represent the adversary's stealthiness by a preference weight. A weight of 0 implies that the adversary does not mind being detected; a higher weight implies that the adversary cares more about remaining undetected. We vary the preference weight from 0 to 11 and observe which access cards the **PrivilegedInsider** uses to achieve the **TrackPowOff** goal.[1] The simulation results show that for weights between 0 and 5, the adversary uses the power staff member card to access the power room. If the weight is between 6 and 10, the environment control staff member card is used to access the environment control room. Finally, if the weight is above 10, the server staff member card is used to access the server room. This shows that the stealthiest adversaries, i.e., those with higher weights, will use a server staff's access card. We should thus focus our efforts on detecting the subsequent actions of that adversary that include logging into the hosts and using the control software to perform actions.

In conclusion, the adversaries that pose the biggest threat to the system in terms of shorter time (or path length) to achieve goals and lower detection probability are the server staff and station operators. When considered in combination, the outsider together with the access control staff may also pose a problem, since the access control staff member is able to grant access to the station.

5 RELATED WORK

Analyzing system security requires significant manual effort from the practitioner. One way to reduce the practitioner's burden is to use a common ontology or library that can be applied to similar systems. Ontologies have been used to build a knowledge base of vulnerabilities and known attacks [15, 20]. The tool $P^2CySeMoL$ [8] also builds up a meta model that is based on a library of attack steps and countermeasures. Well-established tools such as the TVA-tool [9, 10] use a predefined library of exploits. However, these ontologies and libraries focus solely on the cyber domain, and thus are lacking when applied to cyber-physical systems.

Chen et al. [3] analyzed a train control system and a mobile transportation app using attack trees and a Failure Modes, Vulnerabilities and Effects (FMVEA) analysis. They showed that the key challenges

[1] The preference weight can be any integer value, but we will see later that values above 11 do not affect the results.

Table 4: Calculated metrics for each adversary profile. The different possible attack steps (excluding actions that represent physical consequences) are listed. The goals are numbered: 1 for TrackPowOff, 2 for TempCtrlOff, and 3 for LockExits.

	Outsider	Cleaner	Station Op	Env Ctrl	Server	Power	Access Ctrl
Cost	10 or 12	10 or 12	9,10, or 12	7,10, or 12	7 or 9	7,10, or 12	7 or 9
Detect	3	1	0 or 1	0 or 1	0	0 or 1	0 or 1
Attack Path Length	7–15	7–15	7–15	8–15	6–8	6–8	6–9
Attack Steps	**Damage**: PB 2 or 4, TempCtrl, PowSys	**Damage**: PB 2 or 4, TempCtrl, PowSys	**Damage**: PB 2 or 4, TempCtrl, PowSys **Login**: Host **CtrlSoftware**: PLC **RemoteCtrl**: PowSys, TempCtrl	**Damage**: PB 2, TempCtrl **DirectCtrl**: PB 2, TempCtrl	**Damage**: PB 3 **Login**: PLC **RemoteCtrl**: PowSys, TempCtrl	**Damage**: PB 4, PowSys **DirectCtrl**: PB 4, PowSys	**Damage**: PB 3 **DirectCtrl**: BAC Server
Goals	1,2,3	1,2,3	1,2,3	1,2	1,2,3	1	3

to analyzing a cyber-physical system's security are identifying attacks from both the cyber and physical domain and tracing the consequences of attacks in the physical domain. For example, Marronne et al. [14] combined two *Unified Modeling Language* (UML) models, one addressing physical protection of a system and the other addressing cyber protection, to perform vulnerability analysis in critical infrastructure systems. They used a small use case of a railway trackside shelter to evaluate their approach. The TREsPASS project [18] developed an attack navigator [17] that takes in adversary profiles and a library of attack patterns to generate an attack tree. These attack patterns can span the cyber, human, and physical domains. Our approach also uses a high-level language to represent basic concepts and attack steps. However, the ontology that we built aims to address the physical consequences of attacks in far more detail. To address the effects of attacks in the physical domain, Peter et al. [7] extended attack graphs to model the details of physical processes in a system. They modeled the overheating of a transformer in a substation and show the key attack steps that need to be performed. However, their approach is very specific and difficult to apply to a diverse set of devices in a complex setting.

6 DISCUSSION AND FUTURE WORK

In this paper, we constructed an ontology for cyber-physical systems. The ontology also has attack steps that model the physical consequences of actions. Since the attack steps in the AEG are obtained from that ontology, the AEG is only as complete as the ontology. Therefore, we plan to extend our ontology to model various types of devices and possible attack steps and their consequences.

Furthermore, in this work, we model physical consequences as attack steps that are performed by an adversary. We plan to separate the physical consequences from the attack steps by composing the AEG with *stochastic activity networks* (SANs), fault trees, or hybrid attack graphs. Currently, it is difficult to perform such a composition with the current ADVISE tool. However, we envision that this feature will be part of future releases of the tool.

We also intend to extend our case study to model the full railway station. Since a good fraction of the rooms have similar devices and relationships among them, the ontology that we define here can be applied to those components, reducing the amount of modeling effort needed.

We envision that our approach can be used by railway transportation organizations or similar cyber-physical systems' organizations in order to provide insight into the threats posed by different adversary profiles and suggest countermeasures to put in place.

7 CONCLUSION

Designers often evaluate the security vulnerabilities of cyber-physical systems purely from the cyber perspective without taking into consideration the cascade effect of physical consequences or actions that can impact the system negatively. We used the ADVISE meta modeling approach to analyze a railway station. We constructed an ontology that describes the assets and spaces in the system, and physical relations between them. We also defined attack steps that model the physical effects, such as causing damage, device overheating, and powering off. Using the ADVISE tool, we generated an AEG for a small portion of the railway station and simulated different adversary profiles moving through the station. The results show that adversaries can intelligently target a device within their reach, causing a cascade that leads to a bad system state. When detection mechanisms are present in the system, the model shows how adversaries adjust their strategy in response, thus providing practitioners with insight into what other defense mechanisms are needed to harden the security of a system.

ACKNOWLEDGMENTS

This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate, and supported in part by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). This work was partly done when Carmen Cheh was a research intern at ADSC. We also want to thank the experts from SMRT Trains LTD for providing us data and domain knowledge. We thank Atul Bohara, Jenny Applequist, and the anonymous reviewers for their feedback on the paper.

REFERENCES

[1] Graeme Baker. 2008. Schoolboy hacks into city's tram system. http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html. (January 11 2008).

[2] Carmen Cheh, Binbin Chen, William G. Temple, and William H. Sanders. 2017. Data-Driven Model-Based Detection of Malicious Insiders via Physical Access Logs. In *Proc. 14th International Conference on Quantitative Evaluation of Systems*, Nathalie Bertrand and Luca Bortolussi (Eds.). Springer International Publishing, 275–291. https://doi.org/10.1007/978-3-319-66335-7_17

[3] Binbin Chen, Christoph Schmittner, Zhendong Ma, William G. Temple, Xinshu Dong, Douglas L. Jones, and William H. Sanders. 2015. Security Analysis of Urban Railway Systems: The Need for a Cyber-Physical Perspective. In *Proc. 33rd International Conference on Computer Safety, Reliability, and Security*, Floor Koornneef and Coen van Gulijk (Eds.). Springer International Publishing, 277–290.

[4] Brett Feddersen, Ken Keefe, William H. Sanders, Carol Muehrcke, Donald Parks, Andrew Crapo, Alfredo Gabaldon, and Ravi Palla. 2015. Enterprise Security Metrics with the ADVISE Meta Model Formalism. In *Proc. 9th International Conference on Emerging Security Information, Systems, and Technologies.* Venice, Italy, 65–66.

[5] Jim Gash. 2012. Physical operating environment: How the cyber-electromagnetic environment fits. In *Canadian Military Journal*, Floor Koornneef and Coen van Gulijk (Eds.). Vol. 12. 28–34.

[6] Shelby Grad. 2009. Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced. http://latimesblogs.latimes.com/lanow/2009/12/engineers-who-hacked-in-la-traffic-signal-computers-jamming-traffic-sentenced.html. (1 December 1 2009).

[7] Peter J. Hawrylak, Michael Haney, Mauricio Papa, and John Hale. 2012. Using hybrid attack graphs to model cyber-physical attacks in the Smart Grid. In *5th International Symposium on Resilient Control Systems.* 161–164.

[8] Hannes Holm, Khurram Shahzad, Markus Buschle, and Mathias Ekstedt. 2015. P^2 CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language. *IEEE Transactions on Dependable and Secure Computing* 12, 6 (Nov 2015), 626–639.

[9] Sushil Jajodia, Steven Noel, Pramod Kalapa, Massimiliano Albanese, and John Williams. 2011. Cauldron mission-centric cyber situational awareness with defense in depth. In *Military Communications Conference.* 1339–1344.

[10] Sushil Jajodia, Steven Noel, and Brian O'Berry. 2005. Topological analysis of network attack vulnerability. In *Managing Cyber Threats.* Springer, 247–266.

[11] Markus Krötzsch, František Simančík, and Ian Horrocks. 2012. A Description Logic Primer. *Computing Research Repository* abs/1201.4089 (2012).

[12] Elizabeth LeMay. 2011. *Adversary-Driven State-Based System Security Evaluation.* Ph.D. Dissertation. University of Illinois at Urbana-Champaign, Urbana, Illinois.

[13] Elizabeth LeMay, Michael D. Ford, Ken Keefe, William H. Sanders, and Carol Muehrcke. 2011. Model-based Security Metrics using ADversary VIew Security Evaluation (ADVISE). In *Proc. 8th International Conference on Quantitative Evaluation of SysTems.* Aachen, Germany, 191–200.

[14] Stefano Marrone, Ricardo J. Rodriguez, Roberto Nardone, Francesco Flammini, and Valeria Vittorini. 2015. On synergies of cyber and physical security modelling in vulnerability assessment of railway systems. *Computers & Electrical Engineering* 47 (2015), 275–285.

[15] Sumit More, Mary Matthews, Anupam Joshi, and Tim Finin. 2012. A knowledge-based approach to intrusion detection modeling. In *Proc. IEEE Symposium on Security and Privacy Workshops.* IEEE, 75–81.

[16] Jose Pagliery. 2015. Sniper attack on California power grid may have been 'an insider,' DHS says. http://money.cnn.com/2015/10/16/technology/sniper-power-grid/. (17 October 17 2015).

[17] Christian W. Probst, Jan Willemson, and Wolter Pieters. 2016. The Attack Navigator (Invited). In *Graphical Models for Security - Revised Selected Papers*, S. Mauw, B. Kordy, and S. Jajodia (Eds.). Lecture Notes in Computer Science, Vol. 9390. Springer Verlag, Berlin, 1–17. https://doi.org/10.1007/978-3-319-29968-6_1

[18] The TREsPASS Project. 2017. Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security. https://www.trespass-project.eu/. (2017).

[19] Richard A. Serrano and Evan Halper. 2014. Sophisticated but low-tech power grid attack baffles authorities. http://www.latimes.com/nation/la-na-grid-attack-20140211-story.html. (11 February 11 2014).

[20] Blake Shepard, Cynthia Matuszek, C. Bruce Fraser, William Wechtenhiser, David Crabbe, Zelal Güngördü, John Jantos, Todd Hughes, Larry Lefkowitz, Michael Witbrock, Doug Lenat, and Erik Larson. 2005. A Knowledge-based Approach to Network Security: Applying Cyc in the Domain of Network Risk Assessment. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3.* AAAI Press, 1563–1568.