Cascading Threats in Critical Infrastructures with Control Systems

Sandra König
Austrian Institute of Technology
Sandra.Koenig@ait.ac.at

Stefan Schauer
Austrian Institute of Technology
Stefan.Schauer@ait.ac.at

ABSTRACT

Critical infrastructures (CIs) increase in complexity due to numerous dependencies on other CIs but also due to the ongoing digitalization in the industry sector. This yields an increased risk of failure of a single CI as the overall systems gets very fragile and sensitive to errors Failure of a single component may affect large parts of an infrastructure due to cascading effects. One way to support functionality of a CI is the use of Industrial Control Systems (ICS) that allow monitoring remote sites and controlling processes. However, this is an additional source for threats as recent cyber-attacks have shown. Further, the additional information for such cyber systems is often not efficiently combined with existing information on the physical infrastructure. We here propose a method to combine these two sources of information in order to estimate the impact of a security incident on CIs, taking into account cascading effects of threats. An implementation of the model allows simulation of the dynamics inside a CI and yields a record of the status of each asset of the CI. The way the assets change their states illustrates the consequences of an incident on the entire CI. Visualization of the results provides an overview on the situation of the entire CI at a certain point of time and a sequence of such visualization over an entire period of time illustrates the changes over time. The results from this analysis may be used to support security officers in analyzing the current (hybrid) state of their CI in case of an incident and thus increase the hybrid situational awareness.

Keywords

Cascading failures, industrial control systems, critical infrastructures, hybrid situational awareness.

INTRODUCTION AND RELATED WORK

Critical infrastructures (CIs) are growing in complexity, not to the least due to the increasing digitalization and stronger interconnections among them. In most of the CIs, Industrial Control Systems (ICSs) are used to allow monitoring physical systems and controlling processes within the infrastructure. Depending on the characteristics of the infrastructure, either Supervisory Control and Data Acquisition (SCADA) systems or Distributed Control Systems (DCSs) are used (Green et al. 2014). Critical infrastructures that span across a large area, e.g., water or electricity provider, typically use SCADA systems which are supervised in a large control center rather than several smaller decentralized centers (as in the case for DCSs). Besides such control systems, CIs typically collect information on their physical domain from video surveillance applications, access control systems, smoke detectors and other sensors. Information on the cyber domain contains data from Intrusion Detection Systems (IDSs), Security Incident and Event Management (SIEM) systems, firewalls or similar systems.

This information on the physical and cyber subsystems within a CI is not always used optimally as it is often treated independently for each of the domains. However, sophisticated attacks such as Advanced Persistent Threats (APTs), as Operation Aurora (Zetter 2010) or Stuxnet (Karnouskos 2011), often start in the cyber system and later spread to the physical system. In this context, an early step of such a sophisticated attack may be the manipulation of data collected by the SCADA system to pretend normal functionality while the actual physical process is interrupted or disrupted. Protecting a CI from these kinds of attacks requires combining information from both the physical and the cyber domain.

Several approaches exist to investigate cascading effects in CI. Petri nets have been applied (and tested) to fluid

critical infrastructures (Ghasemieh et al. 2013). An overview on failure analysis for power systems is given in (Guo et al. 2017). While most of these are domain specific, we aim for a generally applicable high-level model that needs to be refined when applied to a specific area.

More general models on cascading failures in CIs include Markov chain models (Rahnamay-Naeini and Hayat 2016; Wu and Chu 2017). Our choice of finite-state automaton (a.k.a. finite-state machines) rather than Markov chain models is due to their ability to react on external input.

A description of CIs through a network of deterministic finite-state machines is given in (Klemetti et al. 2016), who applied the model to an electric grid as well as mobile networks. The increasing uncertainty about sates over time is captured by associating a probability distribution to each automaton rather than using a probabilistic automaton.

In this paper, we propose a hybrid view that combines information from the physical and the cyber domain to support the analysis of cascading effects of an incident on the entire CI. The proposed automaton model represents the functionality of a CI's assets through different states and analyses how these states change due to the incident. The implementation of the model represents the network of interdependent physical and cyber assets as a graph and simulates the dynamics within the critical infrastructure using stochastic processes. We introduce the methodology with a focus on critical infrastructures that apply ICS to monitor their processes as an example of a cyber-physical system.

The paper is structured as follows. A hybrid situational awareness model is introduced that combines and extends existing cyber and physical situational awareness to obtain a holistic view on the CI. Further, a model for characterizing the cascading effects of an incident in the CI is described and its use for impact estimation is sketched. The approach is illustrated with a small example and the article finishes with concluding remarks.

HYBRID SITUATIONAL AWARENESS

The current state of the cyber and physical domain is conveniently described by a *cyber situational awareness (CSA)* and a *physical situational awareness (PSA)* system, respectively. The term situational awareness is generally understood as a complete description of the environment including information on potential threats. To this end, it combines several existing sources, e.g., it incorporates information from cameras or access control systems for the physical domain or technical vulnerability databases for the cyber domain. Both CSA and PSA report events that happen in the corresponding system and issue an alarm if such an event threatens the system. A cyber alarm may be a detected virus in the system (say, a malware) while a physical alarm may be a fire in a building that is part of the CI. Further, both CSA and PSA observe numerous events that do not threaten the system directly because they are legitimate activities or do not have a high criticality.

When the information from the CSA and PSA are combined, they build up a *hybrid situational awareness (HSA)* that not only contains information from both subsystems but also combines the two sources of information to obtain new insights about the current status of the CI (Schauer et al. 2019). To achieve that, the HSA links events and alarms from the cyber and physical domain and is able to identify combinations of those events that may indicate a threat to the CI. Furthermore, it also shows which cascading effects might occur within the entire infrastructure due to such an (cyber, physical or hybrid) alarm by providing a list of assets which might be affected by the incident. For example, consider the following two events: an employee leaves the area of the CI in the evening, which is monitored by the access control system at the main gate; second, his credentials are (correctly) used to log in to a computer somewhere inside the CI, which is monitored by a host-based IDS. Both events individually are not of special interest, since they are legitimate in the physical and the cyber domain, respectively, and will not trigger an alarm in the CSA or PSA. However, the combination of both events within a short period of time indicates that there might be a problem, since the same employee cannot be at two places at (roughly) the same time. Such combinations can be caught by the hybrid view of the HSA and are therefore called *hybrid threats* or *hybrid alarms* as they threaten the system but only show up in the hybrid view.

The identification of such hybrid alarms is often difficult as it depends on the context if a combination of two events represents a threat to the infrastructure or not. Therefore, such an identification can be achieved using a rule based system which either defines a set of valid (i.e., a whitelist) or invalid (i.e., a blacklist) combinations of events. However, such an automated detection of correlations between cyber and physical events may only yield candidates for hybrid alarms but expert knowledge is required to decide if this is indeed an alarm or not.

In short, the benefits of a HSA as described here include the following:

- Awareness is increased and operators gain an overview on their cyber-physical system
- Implicit and potentially unknown interdependencies become clear
- Decision making for port operators is facilitated when information about incidents can be estimated and compared

One way to identify hybrid alarms is the correlation engine introduced in (Schauer et al. 2019). The potential cascading effects on the entire CI is analyzed based on a graphical representation of the CIs assets and a stochastic model that describes the propagation through this graph. In the following section, the threat propagation engine is described in more detail.

HYBRID THREAT PROPAGATION

A formal description of a CI is often based on a graph representation where the nodes correspond to single assets of the CI (e.g., a server) and edges to dependencies between them. Our approach is also graph based and additionally, nodes are classified as either cyber or physical, depending on the system they belong to. Directed edges between two assets indicate that one depends on the other. These dependencies include the ones among the cyber and physical assets, respectively, but also interconnections between the two domains. Particularly these interconnections are often not explicitly given. Hence, when building up this asset graph, it is important to discuss with experts who are familiar with the different parts of the CI, i.e., the physical and the cyber assets, to obtain a comprehensive and in-depth description of the entire system. For example, when examining the situation of a water provider it is necessary to talk to employees who know the water distribution network and with others familiar with the SCADA system.

The graph representation is further extended such that each node (i.e., each asset) can be in one of several different states. The state of an asset changes either due to an external incident or due to a change state of an asset it depends on. In both cases, this state change is triggered from outside and does not necessarily happen in constant time intervals. Thus, we describe the dynamic of a single asset by a finite-state automaton (Paz and Rheinboldt 2014) rather than a Markov chain where changes happen at predefined points in time. The dependencies between two assets are manifold and complex as they depend on many factors such as the state of all assets they depend on (directly or indirectly) as well as external influences (such as weather). In order to capture these phenomena, we choose a probabilistic automaton that predicts the next state through a probability distribution over the set of all possible outcomes rather than a precise prediction.

Cascading effects usually have a trigger starting the process. In our case, such a trigger is an incident that causes an alarm reported by CSA, PSA or HSA that is correspondingly called cyber, physical or hybrid alarm (in the following, we will consider these alarms as a start of the cascading while keeping in mind that these are triggered by an incident or an unusual combination of events). In our approach, an alarm has attributes such as name or criticality to describe its nature. Depending on its type, an alarm may influence the dependencies between two assets (e.g., two neighboring rooms are strongly dependent in case of a fire alarm but to a lower degree in case of burglary). In order to capture this dynamic, we choose to model assets as probabilistic Mealy automaton (Paz and Rheinboldt 2014; Rabin 1963) as these also provide output after each state change. More explicitly, the inner structure of an asset looks as displayed in Figure 1.

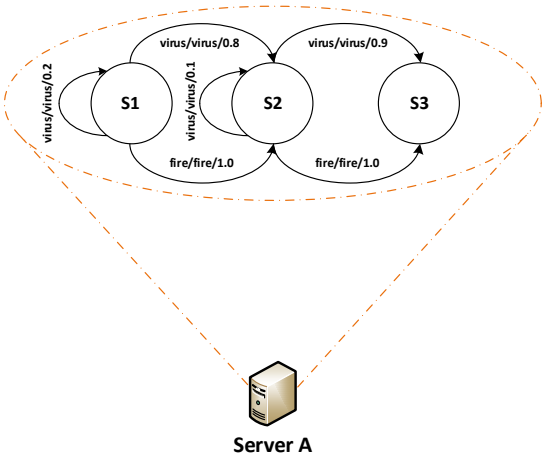

Figure 1. Representation of an asset as a probabilistic Mealy automaton

The asset can be in one of finitely many states (e.g., states s_1, s_2, s_3 as in Figure 1) and changes its state with a certain probability. This state change is due to an input and yields an output as a response to the state change. This is indicated by adding a triple of the form input/output/probability to the edges that describe potential state changes. For a Mealy automaton the output does not only depend on the current state of the asset but also on the input. This allows to forward information on an alarm from one asset to other assets depending on it. A more formal description of this model is given in (König et al. 2019), we here focus on its applicability to the security of CIs.

Dynamics inside a CI

Once the behavior of a single asset is modeled, the dynamics inside a CI can be described when the dependencies between the assets are known. Identification and modeling of dependencies inside a CI are challenging but a necessary part of risk assessment process (Schaberreiter et al. 2013). In our setting, dependencies between assets depend on static factors such as the type of an asset but also on dynamic factors such as the current state of the asset that affects the other one or the type of alarm. As an example, consider a building that is affected by a fire. The room where the fire breaks out may "inform" its neighboring rooms about the fire (i.e., the alarm) as well as how much it is affected by it (i.e., its current state). Both pieces of information influence the state change of the neighboring rooms. The effect on the entire CI due to a fire may be smaller (given that fire safety regulations are working) than, for example, in case of electricity outage where typically an entire floor, the whole building or the entire CI is affected. However, the degree of damage for a specific room also depends on the state of assets it depends on, e.g., if a small fire only affects a PC it is less likely to affect neighboring rooms than if the entire room is on fire.

Using probabilistic automaton to describe the behavior of the assets, the dynamics inside the CI become stochastic as well. This fits into the current line of research where complex systems are described through stochastic processes (König and Rass 2018; Rahnamay-Naeini and Hayat 2016; Wang et al. 2012). Such models are able to take into account the unpredictability of systems where an exact description is hardly possible due to the high complexity as well as due to unknown or non-deterministic factors. Still they provide a formal description of the system that allows an analysis of the average behavior of the system.

Implementing this stochastic model yields empirical data on the behavior of the assets that allows prediction about the future development of the CI. If enough information on the CI is available, e.g., for a specific sector such as smart grids, simulation approaches may be used as an alternative to a formal mathematical description of the dynamics (Findrik et al. 2016; Jaromin et al. 2013; Song et al. 2016). If available, testbeds may be used to simulate different scenarios (Green et al. 2016) and gain some additional insight in the dynamics of a system.

Overall State of a CI

The condition of a CI can be described through the level of functionality of its components. Clearly, if all components were in a bad state, the CI will not be able to provide the expected output while everything works fine if all components were in a good state. However, things are less clear in situations where some assets are not fully operating: Is the CI still able to provide the desired output if some components fail or only work to a certain degree? In order to answer this question, we choose a set of possible states, say "properly working", "limited functionality", and "failure" and let each asset of the CI be in exactly one of these states. For a formal analysis the states may be represented by numeric values, say 1 (properly working), 2 (limited functionality), and 3 (failure). A visualization of the states of all assets provides a basic overview on the overall state of the CI, e.g., assets colored in green (state 1), orange (state 2) and red (state 3), demonstrates where a problem affects the CI the most. The overall state of the CI may be measured by counting the number of assets in the worst state. However, not all assets are of the same importance to the CI operator, so one may only care about the output of the CI, e.g., measure the quality or quantity of the provided water.

IMPACT ESTIMATION FOR CASCADING THREATS

Based on the threat propagation model described above, it is possible to simulate the behavior of all assets of the CI (in the physical as well as in the cyber domain) after an incident. The state changes of the assets indicate the development of a threat and highlight its cascading behavior over different subsystems. The spreading process is reproduced by defining a set of states and transmission probabilities such that the probabilistic Mealy automatons simulate the behavior over time. Recording the state changes of all assets over time (yielding a time series for each asset) indicates how the spreading evolves as the states represent how strong assets are affected.. Depending on the considered scenario, it may be necessary to distinguish several alarms coming in (e.g., from different assets) which can be solved by a prioritization in terms of criticality or time. A basic implementation

can be done in a software such as MATLAB where the prioritization of alarms can be implemented as a FIFO (First In, First Out) queue.

Mimicking the cascading of threats based on the described model requires a comprehensive description of the CI. This particularly includes a list of all the relevant assets within the cyber and physical system including their interdependencies as well as the connections between these two systems. Then, the set of states of each asset needs to be defined as well as the words used for communication between the assets (i.e., the input and output alphabets of the Mealy automatons) that describe the considered alarms. This information may be collected in discussion with (risk) managers who typically have a good overview on the entire system.

After the general specification of the CI, more situation specific aspects need to be taken into account. For each pair of states, the state transitions function and output function need to be formalized. The output function assigns a deterministic output to every pair of input and state which corresponds to the message sent to the dependent asset. In the probabilistic setting, the transition function assigns a probability distribution over all possible states to each combination of an input and a state. The next state is then chosen according to this probability distribution. This more detailed information is collected in discussions with domain experts as knowledge of the processes is required to estimate transmission likelihoods or to predict the most likely next state. Asking for exact estimates of probabilities may be avoided by rather asking for the most likely value combined with an assurance and estimate a probability distribution from these inputs (König and Rass 2018).

With this general and specific information, it is possible to simulate the consequences of an alarm affecting one or more assets by implementing the random behavior of the assets. Recording the state changes over time provides empirical data that can be analyzed statistically. First, it is possible to predict the overall state of the CI by estimating how many assets fail on average (i.e., are in the worst state). Second, one can look at single assets and estimate the likelihood that it is affected or fails. This may be interesting for core assets that should be particularly protected but also provides an overview on state of the CI. To this end, the graphical representation of the infrastructure is extended such that each node is assigned a color depending on its average state, e.g., green if the likelihood of failure is low and red if it is high.

In case the analysis of the current situation has indicated problems (e.g., an important asset is likely to fail, or several assets are likely to end in a state of reduced operation), additional simulations allow testing potential countermeasures. For example, if a sensitive asset is likely to fail, the simulation may be rerun with an increased protection of the asset or a second asset in place which is organized as a backup. Moreover, if a simulation indicates that a large part of a CI is likely to be affected by an alarm additional preventive measures on an organizational level may be required to reduce the expected damage. In this sense, the proposed threat propagation method is not only a tool to analyze the impact of cascading effects on a CI but may also be used to investigate various methods to reduce the expected damage. In general, the security officers monitoring the physical and the cyber domain within a CI (e.g., within a port) will instantiate the model. Hence, it is understood as a collaborative approach with data coming from experts in their respective fields; however, the collaboration is realized in physical meetings and discussions rather than in an online tool.

EXAMPLE

Consider a water provider whose main aim is availability of water with a focus on the quality of drinking water. To reach this goal, a SCADA system is applied to ensure water production, purification and distribution. In this example, the physical domain contains assets such as pumps, wells, water plants and the water distribution system equipped with sensors and Programmable Logic Controller (PLCs). Further, the cyber domain contains assets such as laptops that allow remotely controlling pumps, the SCADA server, the communication system (including switches) and other systems. An extract of this system is displayed in Figure 2. Each component is descried through a Mealy automaton (as shown in Figure 1) but we hide the detailed description to focus on the dependencies between components.

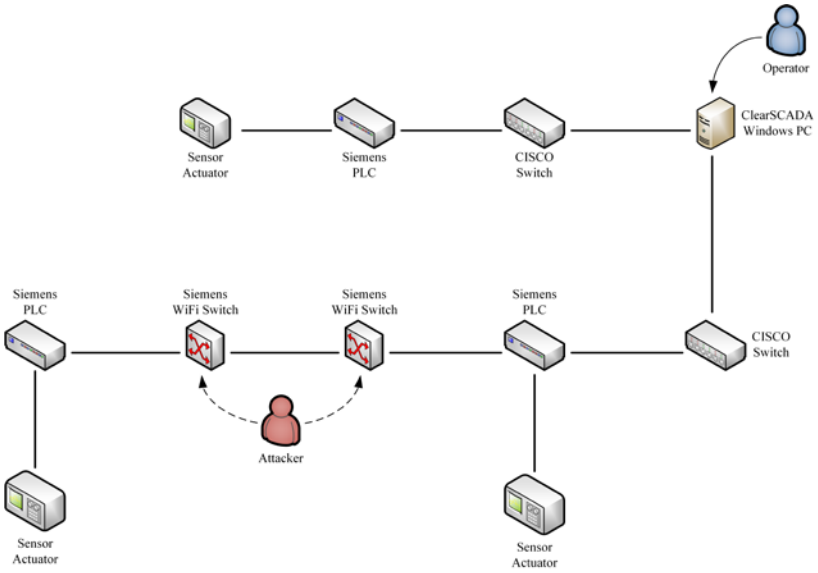

Figure 2. High-level representation of a part of a water provider (automaton details hidden)

An APT (Advanced Persistent Threat) attack on this system, as described in (Gouglidis et al. 2018), typically consist of several steps involving both the cyber and the physical system. Prior to the actual attack, information on the company is collected for a targeted attack, e.g., by social engineering. Then, a device from the cyber network connected to the WiFi network is compromised. This allows launching an attack that causes failure of some controllers (physical assets) and might further give access to the internal company network (cyber asset). Having access to a management server enables the attacker to modify controller logic to cause failure of controllers or switches in the physical network. Depending on the importance of the manipulated devices, this may cause significant damage to the CI operator.

Analyzing cascading effects in this network with our approach involves the following steps:

- Identify all important assets (physical and cyber) and specify the corresponding probabilistic automatons. In particular, define the set of potential states, the transition regimes and the notification behavior (the output function) for each asset.
- Identify all dependencies between the assets both inside the corresponding domains as well as between domains
- Identify all possible alerts (physical, cyber and hybrid) that may cause cascading effects

For the considered attack, the first two steps yield the graph model of the CI as shown in Figure 2 (the graphical representation quickly gets complex so that only a subnetwork of the CI is depicted). For each alert it is now possible to simulate the effects on the entire CI and represent the resulting states of the assets graphically.

A popular goal of such an attack is the SCADA server that allows the manipulation of switches and thus enabling denial-of-service attacks (DoS). Alternatively, technical vulnerabilities of logically connected devices such as sensors or cameras may be exploited to change their behavior or modify data such that wrong measurements or surveillance data are reported. For illustration purpose we consider an attacker who gains access to two switches through the WiFi. This may affect connected devices such as PLCs (Programmable Logic Controllers) which in turn may compromise other dependent devices. The propagation is illustrated in Figure 3 where affected devices are marked red.

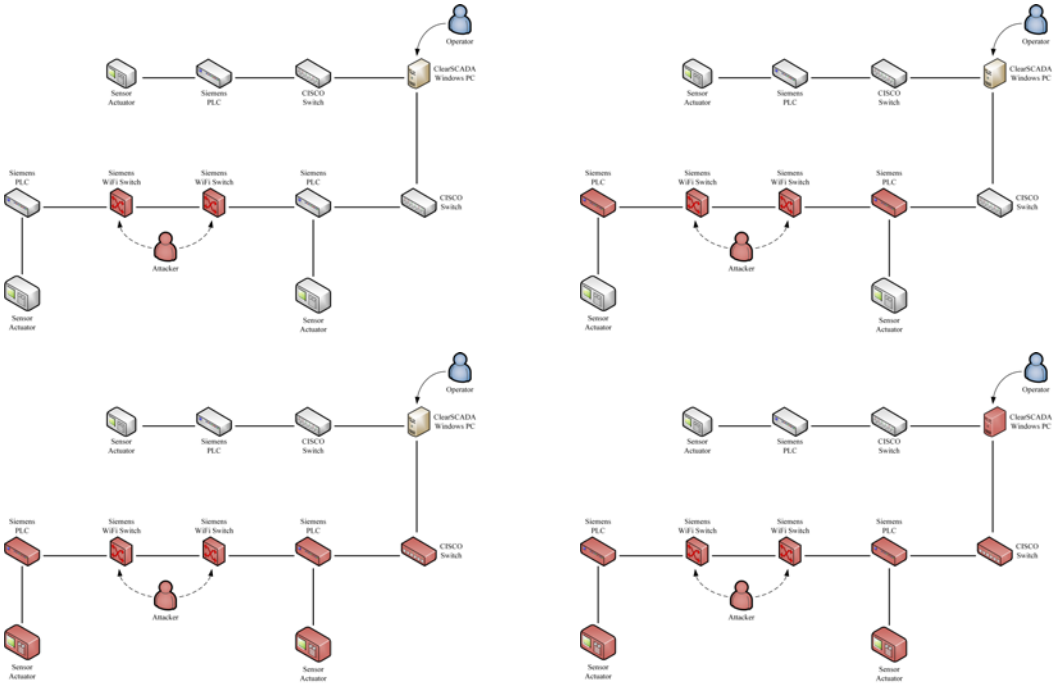

Figure 6. Cascading effects of an attack on two switches of a water provider

CONCLUSION

Investigating cascading effects of an incident within a CI requires the combination of available information on both the physical domain and a cyber domain (including control systems such as a SCADA system). A formal analysis of the CI based on a graph model where nodes are represented as probabilistic Mealy automatons allows investigating the behavior after an incident that occurs in the CI. This model considers different levels of functionality of each component while taking into account the unpredictability of complex systems. Since it combines information from CSA and PSA, the approach presented here yields a hybrid view on the infrastructure and is able to identify threats that are not recognized if the two domains are treated as independent. Simulations based on this model allow estimating the level of functionality of each component within the entire system. This provides an overview on the state of the considered CI after an incident, taking into account cascading effects.

The model described in this paper aims to find a balance between a detailed description of the CIs assets and interdependences and feasibility of implementation. The current description is limited in terms of the source of information used. Extensions may take into account social components (addressing the well-known statement that humans are the weakest link in a security chain) or more technical details. Future directions of research may also focus on the identification of hybrid threats and ways to estimate the impact for a specific type of CIs in more detail.

ACKNOWLEDGMENT

This work was supported by the European Commission's project SAURON (Scalable multidimensional situation awareness solution for protecting European ports) under the Horizon 2020 framework (Grant No. 740477).

REFERENCES

Findrik, M., Smith, P., Kazmi, J. H., Faschang, M., and Kupzog, F. (2016). "Towards secure and resilient networked power distribution grids: Process and tool adoption." *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, Sydney, Australia, 435–440.

Ghasemieh, H., Remke, A., and Haverkort, B. R. (2013). "Survivability Evaluation of Fluid Critical Infrastructures Using Hybrid Petri Nets." *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, IEEE, Vancouver, BC, Canada, 152–161.

Gouglidis, A., König, S., Green, B., Rossegger, K., and Hutchison, D. (2018). "Protecting Water Utility Networks from Advanced Persistent Threats: A Case Study." *Game Theory for Security and Risk*

Management, S. Rass and S. Schauer, eds., Springer International Publishing, Cham, 313–333.
Green, B., Frey, S., Rashid, A., and Hutchison, D. (2016). "Testbed diversity as a fundamental principle for effective ICS security research." London.
Green, B., Paske, B., Hutchison, D., and Prince, D. (2014). "Design and construction of an Industrial Control System testbed."
Guo, H., Zheng, C., Iu, H. H.-C., and Fernando, T. (2017). "A critical review of cascading failure analysis and modeling of power system." *Renewable and Sustainable Energy Reviews*, 80, 9–22.
Jaromin, R., Mullins, B., Butts, J., and Lopez, J. (2013). "Design and Implementation of Industrial Control System Emulators." *Critical Infrastructure Protection VII*, J. Butts and S. Shenoi, eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 35–46.
Karnouskos, S. (2011). "Stuxnet worm impact on industrial cyber-physical system security." IEEE, 4490–4494.
Klemetti, M., Puuska, S., and Vankka, J. (2016). "Entropy as a metric in critical infrastructure situational awareness." E. M. Carapezza, ed., Baltimore, Maryland, United States, 98250K.
König, S., and Rass, S. (2018). "Investigating Stochastic Dependencies Between Critical Infrastructures." International Journal on Advances in Systems and Measurements, 11(3 & 4), 250–258.
König, S., Rass, S., Rainer, B., and Schauer, S. (2019). "Hybrid Dependencies between Cyber and Physical Systems." *accepted for publication*, London.
Paz, A., and Rheinboldt, W. (2014). *Introduction to Probabilistic Automata*. Elsevier Science, Burlington.
Rabin, M. O. (1963). "Probabilistic automata." *Information and Control*, 6(3), 230–245.
Rahnamay-Naeini, M., and Hayat, M. M. (2016). "Cascading Failures in Interdependent Infrastructures: An Interdependent Markov-Chain Approach." *IEEE Transactions on Smart Grid*, 7(4), 1997–2006.
Schaberreiter, T., Kittilä, K., Halunen, K., Röning, J., and Khadraoui, D. (2013). "Risk Assessment in Critical Infrastructure Security Modelling Based on Dependency Analysis." *Critical Information Infrastructure Security. CRITIS 2011.*, Lecture notes in computer science, Springer, Berlin.
Schauer, S., Rainer, B., Museux, N., Faure, D., Hingant, J., Rodrigo, F. J. C., Beyer, S., Peris, R. C., and Lopez, S. Z. (2019). "Conceptual Framework for Hybrid Situational Awareness in Critical Port Infrastructures." *Critical Information Infrastructures Security*, E. Luiijf, I. Žutautaitė, and B. M. Hämmerli, eds., Springer International Publishing, Cham, 191–203.
Song, J., Cotilla-Sanchez, E., Ghanavati, G., and Hines, P. D. H. (2016). "Dynamic Modeling of Cascading Failure in Power Systems." *IEEE Transactions on Power Systems*, 31(3), 2085–2095.
Wang, Z., Scaglione, A., and Thomas, R. J. (2012). "A Markov-Transition Model for Cascading Failures in Power Grids." *2012 45th Hawaii International Conference on System Sciences*, IEEE, Maui, HI, USA, 2115–2124.
Wu, S.-J., and Chu, M. T. (2017). "Markov chains with memory, tensor formulation, and the dynamics of power iteration." *Applied Mathematics and Computation*, 303, 226–239.
Zetter, K. (2010). "Google Hack Attack Was Ultra Sophisticated, New Details Show." *WIRED*.