Estimating Cascading Effects in Cyber-Physical Critical Infrastructures

Stefan Schauer[1][0000−0003−4446−9081], Thomas Grafenauer[1][0000−0002−6911−5610], Sandra König[1][0000−0003−2881−4519], Manuel Warum[1][0000−0002−9001−0696], and Stefan Rass[2][0000−0003−2821−2489]

[1] AIT Austrian Institute of Technology GmbH {stefan.schauer, thomas.grafenauer.fl, sandra.koenig, manuel.warum}@ait.ac.at
[2] Alpen-Adria-Universität Klagenfurt stefan.rass@aau.at

Abstract. Nowadays, critical infrastructures operate a large number of highly interdependent, cyber-physical systems. Thus, incidents can have far-reaching cascading effects throughout the entire infrastructure, which need to be identified and estimated to realize a proper risk management. In this paper, we present a formal model to describe the propagation of a threat through the various physical and cyber assets within a critical infrastructure and the cascading effects this has on the entire infrastructure. We further show, how this model can be implemented into a prototypical tool, which allows to efficiently simulate the cascading effects of a given incident on the entire network of the infrastructure's cyber-physical assets. The functionalities of the tool are demonstrated using a small demo set-up of a maritime port infrastructure. In this set-up, four incident scenarios both from the physical and cyber domain are simulated and the results are discussed.

Keywords: threat propagation · cascading effects · simulation framework · risk estimation.

1 Introduction

Due to the ongoing digitalization in the industrial sector, the interconnections and interdependencies among the physical and cyber systems within today's critical infrastructures (CIs) have increased drastically. A large number of the systems required for the delivery of the CIs service are connected to, controlled by or operated by cyber systems for reasons of efficiency and convenience. Hence, a clear distinction between physical systems and cyber systems, or rather between the Operation Technology (OT) network and the Information and Communication Technology (ICT) network is no longer possible. It has been shown in the past years, that these extensive interconnections give malicious parties and cyber criminals the opportunity to hack and compromise crucial systems without big technological or financial effort. Moreover, the impairment of such a system has wide-spreading effects within the CI itself but also for other CIs and the society as a whole due to these interconnections and interdependencies.

For example, the Ukrainian power provider has been hacked twice in 2015 and 2016 [8, 7]. The main entry point for the hackers were manipulated Word documents [8] and as a result, large areas of the country have been without power for several hours. In 2017, the WannaCry and (Not-)Petya malware has infected millions of systems in various sectors, e.g., in hospitals [2] or in maritime port infrastructures [25], with an impact of over 300 million dollars for the logistics company Moeller-Maersk alone [6]. In this case, one of the starting points for the infection was a malicious update of an accounting software in the Ukraine [6]. These are just a few examples which show, how easily a compromised or malfunctioning system can have large cascading effects on the connected infrastructure. In this paper, we present a prototypical tool, which allows to identify and estimate the potential cascading effects an incident at some system within an organization can have on the overall infrastructure. The tool is part of a larger system, which implements the concept of a Hybrid Situational Awareness (HSA) [29] and builds upon a stochastic process modelling the dependencies among the various systems within a CI. The main advantage of this approach (and thus also of the tool presented here) is that no difference is made between the physical and the cyber domain, i.e., between the ICT and the OT network. Therefore, the tool is able to indicate the effects of a physical incident on the cyber domain and vice versa. The stochastic process used in the approach models each system within the CI as a probabilistic Mealy automaton [15]. In this way, the different operational states as well as the non-deterministic nature of the spreading of an attack can be modelled. The tool as well as the mathematical approach have been developed as part of the H2020 project SAURON with a focus on critical port infrastructures; however, both can be applied analogously on any other type of CIs.

2 Related Work

Among the first models of cascading effects was the Cross Impact Analysis (CIA) [9] that allows describing how dependencies between events affect future events. An extension is the Cross Impact Analysis and Interpretative Structural Model (CIA-ISM) [1] which applies in emergency management to analyze the effects between critical events and to obtain a view on future consequences. These models can be seen as predecessors of contemporary stochastic models.

Cascading effects are of particular interest in the context of security of CIs as first introduced by [18] due to their high importance for society [17, 21]. Cascading effects can occur both in a network of interconnected CIs [22, 14] and inside a complex CI that contains several subsystems as, for example, in power systems [10]. A general overview and a detailed comparison of different methods is given in [22]. Due to recent incidents already mentioned in the Introduction, research also focuses on targeted attacks on CIs, e.g., on power systems [13, 3].

Cascading effects in interconnected networks (sometimes called "network of networks") have been investigated based on Bayesian networks to model and analyze dependencies [4] as well as physical models such as percolation [5, 16] to evaluate the propagation of failures or on topological properties and network analysis [23]. In our work presented here, we apply a model that extends the existing percolation approach by taking into account the dynamics inside a single component [15]. Generally, cascading effects are not exactly predictable due to the many factors that come into play which makes exact models infeasible. Further, consequences of incidents are often influenced by human actions (in particular for targeted attacks such as malware attacks), which further increases the uncertainty. Based on this insight, a large number of stochastic models has been developed.

Classical Markov models hinge on a high complexity of the state space and ask for specification of many transitions probabilities. One way to handle the complexity of Markov models is to use an abstract state space whose states only contain information relevant for the system dynamics [28]. The approach is extended by the Interdependent Markov Chain (IDMC) model [27] that allows describing cascading failures in interdependent infrastructures. Therein, each system is described by a discrete-time Markov chain and the chains are coupled to capture the interdependencies between them.

Markov chains with a memory are used in [33] to describe situations where the transition probabilities do not only depend on the current state but also on the past ones. A conditional Markov transition model has been applied to describe cascading failures in electric power grids [32] and the transition probabilities are derived from a stochastic for flow redistribution. Moreover, time-dependent Markov chains have been used to model system behaviour and account for dynamic aspects of attack strategies to quantify risks to data assets [31]. These phenomena are also captured by our model since the (probabilistic) automata describing a component can be seen as a representation of Markov chains and transitions may be time-dependent. However, our model explicitly allows for reactions to input signals, which is not possible for Markov chain models that update their states at fixed time intervals. Another probabilistic approach to model cascading effects in CIs is application of branching processes that are typically used to describe growth of a population. The applicability of branching processes on modeling cascading effects is investigated in [26].

A necessary input to all spreading models is a complete identification of dependencies. This is a challenge on its own as many dependencies are not visible at first sight. Interconnections between infrastructures are investigated with a probabilistic model in [19] and empirical findings are presented in [20]. An overview on existing modeling approaches and open research questions concerning CI interdependencies in urban areas is given in [11]. A method to identify and describe service failure interdependencies is introduced in [30].

3 Threat Propagation Model

The model we are using to describe the propagation of a threat through the various assets within a CI and the cascading effects it has on this infrastructure applies concepts from graph theory, automaton theory and stochastic processes.

It is part of a larger project, i.e., the SAURON framework [29], which provides comprehensive situational awareness for the physical and the cyber parts of a critical port infrastructure to its operators. Therefore, the SAURON framework integrates a Physical Situational Awareness (PSA), a Cyber Situational Awareness (CSA) and a Hybrid Situational Awareness (HSA) system. Whereas the PSA and the CSA are more or less standard components which are building upon existing tools and frameworks tailored to the requirements of port infrastructures, the HSA is developed from scratch integrating the threat propagation model and represents the main innovation in the project. In detail, the HSA works on events detected and alarms triggered by both the PSA and the CSA upon incidents happening in the respective domains. In this way, the HSA connects the information coming from the PSA and the CSA and brings together the two usually separated domains.

The establishment of such a hybrid view on an infrastructure's asset architecture is achieved by using two main modules, i.e., the Event Correlation Engine (ECE) and the Threat Propagation Engine (TPE). As the name already indicates, the ECE focuses exclusively on events, i.e., general occurrences, detected by the PSA and the CSA with the main goal to identify inconsistencies among those events. The TPE deals with alarms, i.e., critical incidents, and is responsible for the identification of the potential cascading effects of such incidents. To achieve that, the TPE builds upon a graph representation of all the assets, physical and cyber, given in the infrastructure. The assets are represented by the nodes of this graph, whereas each node can be in one of several operational states (cf. Section 4.1 for further details). The edges in this graph represent the different dependencies between the various assets.

As already mentioned above, the TPE processes the alarms triggered by the PSA or the CSA. An alarm usually involves one specific asset and, depending on the type of incident, changes the operational state of this asset. This represents a reduction of the service or the capacity of this asset due to the incident happening. Based to the connections among the asset in the graph, i.e., the dependencies, the state change of one asset might also affect the operational state of all the subsequent assets depending on it and so forth. In this way, the cascading effects of the incident, which triggered the alarm, on the entire infrastructure are described by the TPE. As a result, the TPE delivers two lists: one list containing the most critical assets affected by the alarm and a second list describing the assets that will be affected immediately in the next step.

4 Model Implementation

4.1 Formal definition

The formal description of the threat propagation model is essentially a network of coupled probabilistic Mealy automata (following [15]): at a high level, consider a set V of assets forming nodes of a graph $G = (V, E)$, where a directed edge $u \to v \in E$ models a dependency of asset v on asset u (e.g., u may be a supplier for v). Each node $v \in V$ undergoes an evolution over time, switching between different states of operation, ranging from "fully operational" until "outage". The transition from one state to another is triggered by notifications (messages) exchanged between assets, and is in general probabilistic. That is, if asset u experiences troubles and changes its state from "fully operational" to "partly affected", it notifies its dependent asset (in our example node v if there is an edge $u \to v$) about this change. The dependent asset v, in turn, may not deterministically react upon this, but may change its state probabilistically based on this information. If v undergoes its own state transition (caused by the change of u before), it acts in the same way as u and notifies its descendants (in the asset network G) about its new status, which lets them react likewise.

Probabilistic Mealy automata are a natural description of this intuition, since they process and emit symbols, which are the messages received from other automata. The simulation model thus has two levels (cf. also Fig. 1): (i) the *outer model*, which is a humble directed graph $G = (V, E)$, and (ii) a set of *inner models*, one for each $v \in V$, which are Mealy automata over a common set S describing the states of operation. Each Mealy automaton v describes the transition from state $s_1 \in S$ to state $s_2 \in S$ by the triplet $\langle$incoming message m, outgoing notification m', probability $p\rangle$, with the semantics that upon receiving the message m from a parent node/asset of v in G, with probability p, there will be a change into state s_2, upon which another notification m' is published to other assets that depend on this node (in G). With probability $1 - p$, the automaton v will remain in its state (possibly also notifying other assets about this fact, if the transition from s_1 to s_1 is defined accordingly).

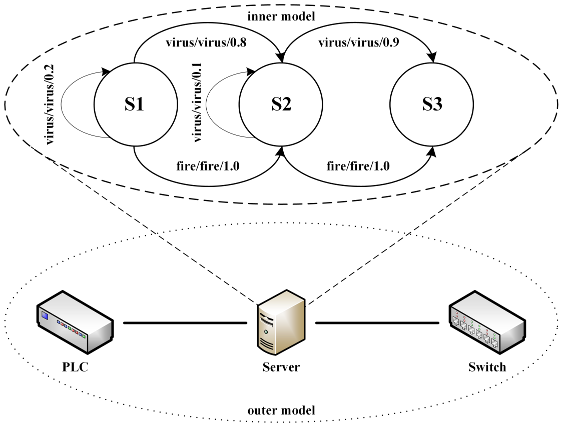

Fig. 1. Schematic illustration of the inner and out model.

Further, an element of time herein can added, if we replace the probability p by some time-dependent value $p(t)$, and let the automata undergo transitions without external forces (this is the behavior of a Markov chain, and as such different to the Mealy automaton model; hence distinguishes the model from a Markov chain in this aspect). We do not explore this possibility further here and leave it for future work.

4.2 Reference Implementation

The model described in Section 4.1, has been implemented as a Node.js module using the TypeScript language and is callable through a HTTP-based interface [3]. In addition, a browser-based Single Page Application has been developed to enable users drawing and connecting their own physical or cyber assets. This includes relevant information about each asset such as probabilistic state transitions or which types of alarms affect it. In the current version of the tool, this information needs to be filled in by the user; in particular, the probabilities for all state transitions are chosen based on the experience and knowledge of the infrastructure's experts. It is planned in the future to support the user with these inputs, e.g., by using a small sample set for the probabilities of state transitions, to reduce the effort required for setting up the model. Aside from simulations, all other processing is done in-browser, meaning that it is not reliant on network connectivity.

After a user has modeled his own environment and assets, the underlying implementation allows triggering any of the defined alarms on any asset that responds to it. While the modelling aspects are handled by the web application, any actual calculations related to simulations are offloaded to the HTTP-based application programming interface (API). If the user triggers such an alarm, a configurable amount of simulations are run on the server side, an ordered set of changes will be returned, and the web application aggregates it into distributions. Information valuable to determine appropriate courses of actions can be gleaned from this, including which assets are likely to be affected next or what is the worst possible outcome if no mitigating measures are undertaken.

The web application does not rely on the user drawing their environment within the application. Instead, such environments can be imported from and exported to disk using a human-readable, JSON-based text file. The object schema contained within these network files is strongly based on the visualization library vis.js and requires only a set of nodes composed of unique identifiers and Mealy automaton transition quintuples, and a set of directed edges composed of unique identifiers as well as which two nodes are connected. The operational state of each asset is expressed as an integer of the domain $\{1, 2, 3\}$.

Once a series of simulations for a given scenario has been calculated, the API responds with an ordered set of changes that occurred for each simulation. This also includes a verification, i.e., which transitions have been triggered by which event. As multiple simulations are run, different outcomes are generated and aggregated for further evaluation by human operators. Simulations can either be run until they no longer trigger any transitions within the Mealy automata, or until a predefined logical time has been reached. The minimum number of

[3] Online available at https://atlas.ait.ac.at/sauron

iterations for the simulation required to obtain a statistically significant result can be determined in (at least) two ways: first, conditional on the evolution being only degenerative, i.e., without recovery or repairs, we have all statuses evolve monotonous and bounded, so the status of each CI will necessarily lead to convergence. Consequently, we can stop as soon as the change to the overall status falls below some (small) threshold. The same holds if we treat the simulation outputs as samples from a time series. On these, we may apply a palette of convergence diagnostics known from the field of Markov chains (e.g., [24]). Alternatively, if we want to conduct statistical tests based on simulation results, we can consider each simulation output as one sample from an experiment, and ask for the number of such samples required for statistical significance. Software for statistical power analysis (e.g., G*Power [12]) can do this.

The main output of the application is twofold: the "State Distributions" aggregate each asset's final state distribution across all simulations and the "Worst and Average Outcomes" provides an overview on which assets are at risk after a certain amount of time in a worst case and an average scenario. This information can be directly used for incident handling, i.e., to evaluate the potential consequences when a specific alarm is triggered and show them to the security operator using the tool.

To test the performance of the simulation core functionality, we tracked the processing time for different model configurations (see Table 1) where each test consists of 1000 simulation runs and the initial event was randomly chosen. The smallest model needed half a second to simulate whereas the biggest one needed approximately 3 minutes and 20 seconds. This can further be enhanced since the performance is currently limited to Node.js single threaded language JavaScript but can be improved by using worker threads.

Test-Nr	Nodes	Edges	Symbols	Transitions	Simulation-Time
1	100	200	1	2-3	509 ms
2	500	1200	1	2-3	5254 ms
3	1000	3000	1	2-3	23191 ms
4	1000	3000	5	10-15	32825 ms
5	1000	10000	5	10-15	199592 ms

Table 1. This table shows the performance of the simulation core for different model sizes.

5 Scenario Description

To showcase the application, we created an example scenario of a simplified port infrastructure with four physical areas: two office buildings with four rooms each, a truck gate that monitors incoming and outgoing traffic and an area where liquid natural gas (LNG) is stored (see Figure 2). The rooms of the office buildings again contain different ICT assets such as workstations and servers; more important rooms (i.e., B1-4, B2-3 and B2-4) are secured by access control systems. The LNG area and the truck gate have surveillance cameras and access controls installed. All cameras and access controls are connected via the OT network; similarly, all workstations are connected in the ICT network. Further, there are three physical servers located in the infrastructure: one of them is responsible for database management, another one provides the customer relationship management (CRM) services and the access control (ACC) and the third runs the services for surveillance (SUR) and freight management (FRM). The connections between these assets were set according to their physical and/or cyber based dependencies to each other. Hence, the three servers are physically connected to both the ICT and the OT network, since they provide databases and services for the working stations and access to OT devices like cameras and access controls.

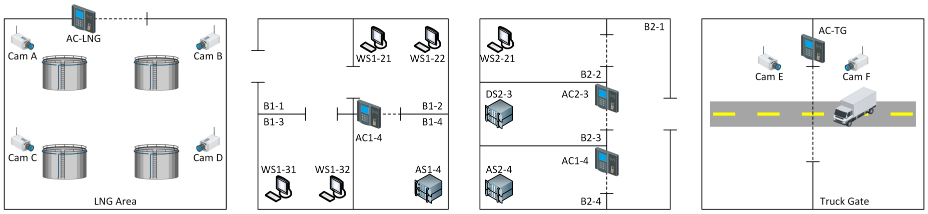

Fig. 2. Schematic set-up of the port infrastructure in our scenario

To model cascading effects, transitions need to be defined for each threat at the corresponding assets. For our scenario, we consider a physical threat, i.e., a fire, and a cyber threat, i.e., a system gets hacked and compromised. In case of a fire, only physical assets have transitions to react to the alert; cyber based assets do not actively react on the input "fire" and thus are not directly affected. However, they can still be affected by the threat because of their dependencies to physical assets. A fire usually reduces the operational state of a physical asset to "outage", i.e., the asset shuts down or is destroyed. To model this effect, the physical assets transform the input fire to the output "offline" or "destroyed" which is then sent to all dependent assets. In a similar way, the cyber threat of hacking a system can also affect physical assets: a room can become "compromised" if the access control system is hacked, it does not provide sufficient protection any more.

6 Results and Discussion

For our first showcase, we simulate two fires happening in different buildings, i.e., in room B1-1 (cf. Figure 3) and B2-1 (cf. Figure 4) and compare the outcome of 1000 simulation runs each. The grey-scale in the figures represent the average state of the CI's assets (white = state "1" to dark grey = state "3"). In Scenario 1, a fire starts in room B1-1. Figure 3 shows the average state of each asset over all simulation runs. The fire caused an outage of all workstations and the server in building B1. Two applications and services run on this server: CRM and ACC. Since the ACC service and application is now offline, all the access controls (triangles) to sensitive areas are affected and don't operate any more. As a result, the LNG area, the truck gate as well as two rooms in building B2 can be accessed without any authorization required.

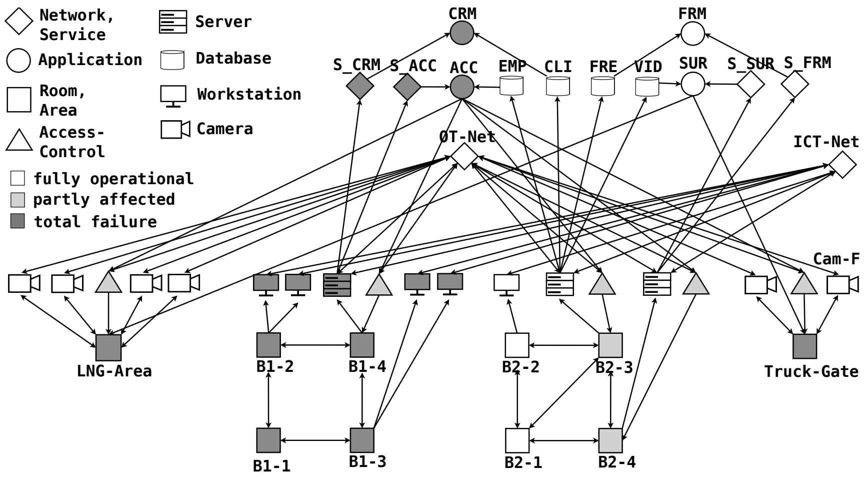

Fig. 3. Average case result of Scenario 1 (fire in room B1-1).

In Figure 4, the fire starts in room B2-1 but does not affect all adjacent rooms equally. The room B2-3, which contains the database server, is only affected partially because it is more resistant to fire (modelled with a transition probability of only 20%) and is damaged (i.e., changes to state "3") in only 44% of the simulations. As a result, the database server in room B2-3 is less affected than the application server of the adjacent room B2-4. Due to the damage of the application server, the SUR and FRM services are shut down, too. Hence, although the cameras and access controls are still intact, security operators are no longer able to access the video stream. Therefore, the LNG area and the truck gate are partially affected and not considered secure any more.

When comparing both scenarios, we see that due to the fire resistance of room B2-3, fewer assets are in the most severe state "3" and more are only partially affected (i.e., in state "2") in Scenario 2. This indicates that using fire resistant material will lower the overall risk for the CI. However, when computing the average operational state of the entire CI (i.e., the arithmetic mean over the states of all assets), we can see that the difference is not significant (1.77 for Scenario 1 and 1.64 for Scenario 2). Nevertheless, when looking at the worst case (which is not depicted here due to space limitations), the average operational state for the entire CI is much worse compared to Scenario 1, i.e., 2.02 for Scenario 2 versus 1.86 for Scenario 1. That is, the CI is expected to be partly affected but likely remain running (“fully operational” $= 1 < 1.86 < 2 =$ “partly affected”) in Scenario 1, as opposed to be probably affected if not endangered to be out of order (“partly affected” $= 2 < 2.02 < 3 =$ “outage”).

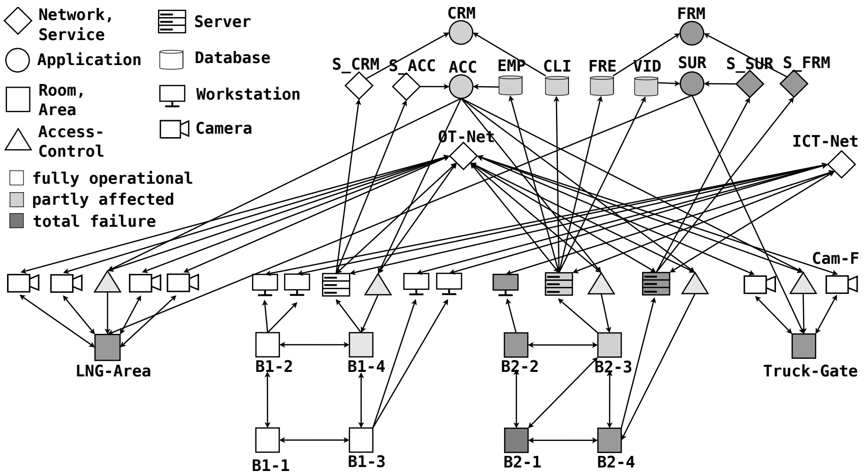

Fig. 4. Average case result of Scenario 2 (fire in room B2-1).

In our second showcase, we assume that the camera “Cam F” at the truck gate is hacked by a malicious party and we again compare two scenarios with 1000 simulation runs: Scenario 3, where there is no special security layer in the OT network, and Scenario 4, where cyber security measures are implemented to protect the OT network. Due to these security measures, we assume that the chance for the OT network to be compromised when one of the devices in the network gets hacked is only 30%, whereas without this security layer, the OT network is always compromised. In Scenario 3, the compromised camera “Cam F” caused many other devices in the OT and ICT network to be compromised (cf. Figure 5), which has a crucial effect on the surveillance of the LNG area and the truck gate (in our showcase, compromised assets are in state “2”, since they are still functional but might send or display false information). For this case, the average operational state for the entire CI is 1.47, i.e., more likely fully functional with some (smaller) chance of being partly affected (1.47 is closer to state 1 than to state 2).

When looking at Scenario 4, the devices connected to the OT network are not as easily compromised, leaving them in a much better operational state. Accordingly, also the average operational state for the entire CI is better, i.e., 1.16. The advantage of implementing security measures on the network level are even more visible, when we observed the likelihood of individual assets to change

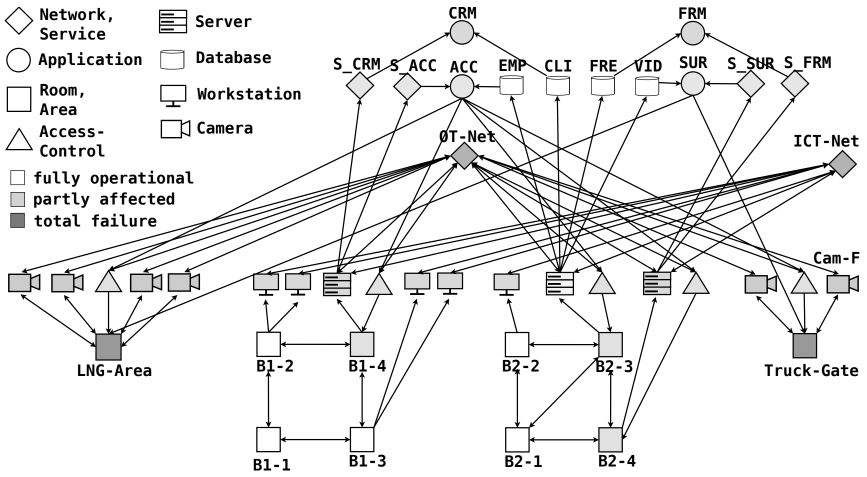

Fig. 5. Average case result of Scenario 3 (insecure OT network).

their operational state directly after "Cam F" has been compromised (i.e., after the initial step of the simulation). Table 2 shows the top five assets, which are most likely to change their operational state within the next three simulation steps. From this table, we can see that the likelihood of the top two assets ("OT Network" and "LNG Area") is reduced drastically. Hence, the results from the simulation can also be used to evaluate the implementation of new security measures or mitigation actions at different assets in the CI.

7 Conclusion

In this paper, we presented a tool which allows simulating the propagation of a threat though a CI's asset network and estimating the cascading effects of that threat. This is achieved by using a formal model building on graph theory, automaton theory and stochastic processes. The main benefit of the approach compared to other methodologies in the literature (cf. Section 2) is that it combines assets from the physical and the cyber domain, integrates their interdependecies and thus provides a holistic (or hybrid) view on the cascading effects. Hence, the results can improve the CI operator's risk analysis and risk management processes. We demonstrated the functionality in four scenarios, describing the effects of physical and cyber threats on a simplified maritime port infrastructure. Based on these scenarios, it is easy to see that the model can be quickly adapted to integrate new security measures and estimate their effects for different incidents. However, one major drawback of the approach is that the transition probabilities need to be specified for each dependency (i.e., each node in the graph). For large networks, this can be a laborious task which involves expert opinions. Thus, a next steps is to integrate a methodology to formalize this process, making it more efficient and less time-consuming.

Insecure OT network			Secured OT network		
Asset name	State	Likelihood	Asset name	State	Likelihood
OT-Network	2	86.20%	Truck-Gate	2	59.80%
LNG-Area	3	85.50%	OT-Network	2	26.70%
Truck-Gate	3	78.90%	LNG-Area	3	26.30%
IT-Network	2	76.70%	Truck-Gate	3	25.00%
Camera-B	2	74.0%	IT-Network	2	23.80%

Table 2. This table shows the top five most likely assets to be affected within the next three simulation steps after “Cam F” has been compromised (with a 85% probability).

Acknowledgement

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

1. Bauls, V.A., Turoff, M.: Scenario construction via delphi and cross-impact analysis. Technological Forecasting and Social Change **78**(9), 1579 – 1602 (2011)
2. BBC News: NHS cyber-attack: GPs and hospitals hit by ransomware (2017), http://www.bbc.com/news/health-39899646
3. Bilis, E.I., Kroger, W., Nan, C.: Performance of electric power systems under physical malicious attacks. IEEE Systems Journal **7**(4), 854–865 (2013)
4. Burnap, P., Cherdantseva, Y., Blyth, A., Eden, P., Jones, K., Soulsby, H., Stoddart, K.: Determining and sharing risk data in distributed interdependent systems. IEEE Computer **50**(2), 72–79 (2017)
5. Carreras, B.A., Newman, D.E., Gradney, P., Lynch, V.E., Dobson, I.: Interdependent risk in interacting infrastructure systems. In: System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. pp. 112–112 (2007)
6. Cimpanu, C.: Maersk Reinstalled 45,000 PCs and 4,000 Servers to Recover From NotPetya Attack (2018), https://www.bleepingcomputer.com/news/security/maersk-reinstalled-45-000-pcs-and-4-000-servers-to-recover-from-notpetya-attack/
7. Condliffe, J.: Ukraines Power Grid Gets Hacked Again, a Worrying Sign for Infrastructure Attacks (2016), https://www.technologyreview.com/s/603262/ukraines-power-grid-gets-hacked-again-a-worrying-sign-for-infrastructure-attacks/
8. E-ISAC: Analysis of the Cyber Attack on the Ukrainian Power Grid. Tech. rep., E-ISAC, Washington, USA (2016), https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf
9. Gordon, T., Hayward, H.: Initial experiments with the cross impact matrix method of forecasting. Futures **1**(2), 100 – 116 (1968)
10. Guo, H., Zheng, C., Iu, H.H.C., Fernando, T.: A critical review of cascading failure analysis and modeling of power system. Renewable and Sustainable Energy Reviews **80**, 9–22 (2017)
11. Hasan, S., Foliente, G.: Modeling infrastructure system interdependencies and socioeconomic impacts of failure in extreme events: emerging R&D challenges. Natural Hazards: Journal of the International Society for the Prevention and Mitigation of Natural Hazards **78**(3), 2143–2168 (2015)

12. Heinrich-Heine-Universität Düsseldorf: G*power: Statistical power analyses for windows and mac, http://www.psychologie.hhu.de/arbeitsgruppen/allgemeine-psychologie-und-arbeitspsychologie/gpower.html, retrieved: 2019-08-21
13. Koc, Y., Warnier, M., Kooij, R.E., Brazier, F.M.T.: A robustness metric for cascading failures by targeted attacks in power networks. In: 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC). IEEE (2013)
14. König, S., Rass, S.: Investigating stochastic dependencies between critical infrastructures. International Journal on Advances in Systems and Measurements **11**(3&4), 250–258 (2018)
15. König, S., Rass, S., Rainer, B., Schauer, S.: Hybrid dependencies between cyber and physical systems. In: Arai, Kohei, B.R.K.S. (ed.) Intelligent Computing. CompCom 2019. Advances in Intelligent Systems and Computing. Springer, Cham (2019)
16. König, S., Schauer, S., Rass, S.: A stochastic framework for prediction of malware spreading in heterogeneous networks. In: Secure IT Systems. Proceedings of NordSec conference 2016, Oulu, Finland, pp. 67–81. Springer International (2016)
17. Kotzanikolaou, P., Theoharidou, M., Gritzalis, D.: Cascading effects of common-cause failures in critical infrastructures. In: Butts, J., Shenoi, S. (eds.) Critical Infrastructure Protection VII. pp. 171–182. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
18. Laprie, J.C., Kanoun, K., Kaniche, M.: Modelling Interdependencies Between the Electricity and Information Infrastructures. In: Saglietti, F., Oster, N. (eds.) Computer Safety, Reliability, and Security. pp. 54–67. Springer (2007)
19. Little, R.G.: Controlling cascading failure: Understanding the vulnerabilities of interconnected infrastructures. Journal of Urban Technology **9**(1), 109–123 (2002)
20. Luiijf, E., Nieuwenhuijs, A., Klaver, M., van Eeten, M., Cruz, E.: Empirical findings on critical infrastructure dependencies in europe. In: Setola, R., Geretshuber, S. (eds.) Critical Information Infrastructure Security. pp. 302–310. Springer (2009)
21. McGee, S., Frittman, J., James Ahn, S., Murray, S.: Implications of cascading effects for the hyogo framework. International Journal of Disaster Resilience in the Built Environment **7**, 144–157 (2016)
22. Ouyang, M.: Review on modeling and simulation of interdependent critical infrastructure systems. Reliability Engineering & System Safety **121**, 43 – 60 (2014)
23. Pagani, G.A., Aiello, M.: The power grid as a complex network: A survey. Physica A: Statistical Mechanics and its Applications **392**(11), 2688 – 2700 (2013)
24. Plummer, M., Best, N., Cowles, K., Vines, K.: Coda: Convergence diagnosis and output analysis for mcmc. R News **6**(1), 7–11 (2006), https://journal.r-project.org/archive/
25. PTI: New malware hits JNPT operations as APM Terminals hacked globally | The Indian Express (2017), http://indianexpress.com/article/india/cyber-attack-new-malware-hits-jnpt-ops-as-apm-terminals-hacked-globally-4725102/
26. Qi, J., Dobson, I., Mei, S.: Towards estimating the statistics of simulated cascades of outages with branching processes. IEEE Transactions on Power Systems **28**(3), 3410–3419 (2013)
27. Rahnamay-Naeini, M., Hayat, M.M.: Cascading failures in interdependent infrastructures: An interdependent markov-chain approach. IEEE Transactions on Smart Grid **7**(4), 1997–2006 (2016)
28. Rahnamay-Naeini, M., Wang, Z., Ghani, N., Mammoli, A., Hayat, M.M.: Stochastic analysis of cascading-failure dynamics in power grids. IEEE Transactions on Power Systems **29**(4), 1767–1779 (2014)

29. Schauer, S., Rainer, B., Museux, N., Faure, D., Hingant, J., Rodrigo, F.J.C., Beyer, S., Peris, R.C., Lopez, S.Z.: Conceptual Framework for Hybrid Situational Awareness in Critical Port Infrastructures. In: Luiijf, E., Zutautaite, I., Hämmerli, B.M. (eds.) Critical Information Infrastructures Security. pp. 191–203. Lecture Notes in Computer Science, Springer International Publishing (2019)
30. Seppänen, H., Luokkala, P., Zhang, Z., Torkki, P., Virrantaus, K.: Critical infrastructure vulnerability—a method for identifying the infrastructure service failure interdependencies. IJCIP **22**, 25–38 (2018)
31. Vasilevskaya, M., Nadjm-Tehrani, S.: Quantifying Risks to Data Assets Using Formal Metrics in Embedded System Design. In: Koornneef, F., van Gulijk, C. (eds.) Computer Safety, Reliability, and Security. pp. 347–361. Springer (2015)
32. Wang, Z., Scaglione, A., Thomas, R.J.: A Markov-transition model for cascading failures in power grids. In: 2012 45th Hawaii International Conference on System Sciences. IEEE (2012)
33. Wu, S.J., Chu, M.T.: Markov chains with memory, tensor formulation, and the dynamics of power iteration. Appl. Math. Comput. **303**(C), 226–239 (2017)