Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 961

Integrated Security Framework: Towards a Holistic Approach for Analysis, Simulation and Management of System Security Features

Yuan Gao[1], Robert Fischer[1], Simon Seibt[2], Mithil Parekh[1] and Jianghai Li[3]

Abstract: The increasing cyber threats require quick action from security experts to protect their industrial automation control system (IACS). For fulfilling the requirement, we propose to divided the classic cyber security analysis scope into three separated, yet interconnected domains: Threat, System and Security. Thus different groups of security professionals can work independently, and are not required to have the knowledge about the full scope. In addition, we proposed an asset-centric system architecture model to enable the modeling and simulation of attacks according to publicly known threats and vulnerabilities. Analysis based on the generated attack/defense trees can assist to manage and continuously monitor the deployed security controls. The proposed approach with tool supports reduces the workload of security experts as well as the incidents response team (IRT) towards an adaptive defense manner.

Keywords: threat model, asset management, attack/defense tree, adaptive defense

1 Introduction

The chaos raised by the ransomware attack WannaCry [Ma17] confirmed once again to the public that our cyberspace of both personal and industrial systems, are relatively vulnerable to the day-to-day increased cyber threats. One major reason for the obfuscation of the border between general IT security and industrial security is the introduction of commercial-off-the-shelf (COTS) hardware and software. Involving COTS in an industrial automation control system (IACS) can help to save the budget while reduce the time-to-market. However, meanwhile, COTS lack strict and extensive security evaluation procedures and therefore, are considered of making IACS more vulnerable. Furthermore, the intentions and actions targeting at critical IACS (e.g. power plants and energy grids) are increasing [Co16][PSF17]. For assuring that IACS can survive among the increased threats, beyond defense-in-depth [Wa16], security information and event management (SIEM) systems are required for the continuously monitoring purpose [GX16]. In addition, based on the system monitoring, adaptive defense [De14] using temporary controls can help to block ongoing attacks thus to buy time for the incidents response team (IRT) to

[1] Otto-von-Guericke University Magdeburg, Research Group Multimedia and Security, Universitätsplatz 2, 39106 Magdeburg, yuan.gao@ovgu.de
[2] Nuremberg Institute of Technology, Department of Computer Science, Keßlerplatz 12, 90489 Nuremberg, seibtsi47063@th-nuernberg.de
[3] Tsinghua University, Institute of Nuclear and New Energy Technology, Beijing, China, lijianghai@mail.tsinghua.edu.cn

doi:10.18420/in2017_99

take appropriate countermeasures.

However, compared to the increased threats, nowadays security teams for IACS lack the capabilities for performing quick, dynamic, situational reactions. First of all, traditional security analysis and mitigation processes are costly while hardly to reuse, because they use top-down approaches while top levels have little knowledge about the security relevant details. Secondly, industrial security professionals cannot benefit from the IT security development. On one side, the threats against general IT systems are classified and analyzed with the assumption to deploy controls on similar and scalable systems. On the other side, industrial systems are designed to be heterogeneous for achieving different functionality. Thus security professionals need to know the full analysis scope from threats to the target system architecture, which means a huge workload. At the same time, the industrial field lack of reusable, generalized, and easily extendable/adaptable tool supports.

For bridging general IT-security threats to a target IACS while managing correspondent controls adaptively, this work contributes the following points:

- Dividing the full analysis scope into three separate domains: Threat, System and Security. The three domains are interconnected to each other to achieve the security requirements of IACS.
- Proposing the system architecture model in an asset-centric manner to address the correct level of relevant security details.
- Distinguishing external/internal security forces work for an IACS. Thus different groups of security professionals might work independently in different domains.
- Combining the threat model and the system architecture model using attack/defense trees. A prototype is designed and it is expected to support IRT to manage continuous monitoring while to achieve adaptive defenses.

2 Related Work

Security analysis processes and associated risk managements are addressed in different international standards, like ISO/IEC 27005 [II11] and IEC 62443 [IE15]. The latter industrial standard series proposed a zone-conduit model for guiding the segmentation of security zones while monitoring their communications through conduits. The UK's national standard proposed the Domain Based Security (DBSy) Model to assist the analysis [HM09]. However, the DBSy Model is a conceptual model represents the security consideration in a draft way [GP16]. In general, all these three standards analyze the target system from top levels without touching the correct level of security relevant details of IACS. Furthermore, without a proper system architecture model, the created triads of "threat-attack-vulnerability" are text-based and duplications may exist. For analyzing the security posture of a small scale system, like an android kernel [LH15], it is obvious that

the authors can be ¨full-stack" engineers who know everything about the system thus no explicit system architecture modeling is required. Compared to this, Mao et al. employed a mathematical model of power systems to analyze whether their operations are secure [MI14]. However, the method they named as "trend security analysis" used to indicate abnormal system behaviors which have no clue to triggering reasons or possible attackers. A multi-layered security analysis framework was proposed in [Ha15]. This work proposed a reference model for representing a cyber-physical system (CPS) while considering attack based on that model. However, the proposed framework does not involve the asset management that can help to track product-specific vulnerabilities. A more comprehensive framework that considers common threats/vulnerabilities as well as the system architecture at the same time is expected.

Fischer et al. proposed different groups of entities for modeling attacks against IACS [Fi16]. Their work provides a solid foundation for the system architecture model discussed in this paper. Meanwhile, digital libraries for common threats and vulnerabilities, like common attack pattern enumeration and classification (CAPEC) [Ba08] as well as common vulnerabilities and exposures (CVE) [MG02] are available as the sources. The attack/defense tree described by Kordy et al. in [Ko13] is a suitable data structure for threat modeling. In addition, the Automation Markup Language (AutomationML) [IE14b] provides the foundation for building the system architecture model of IACS. With proper extensions, it can be used for creating the threat model as well as modeling security controls. AutomationML was originally designed to provide a neutral data format for data exchange within an industrial working environment. Thus it was designed to incorporate with additional industrial data formats: such as COLLADA for 3D data [IP12] and PLCopen for program logic [IE14a]. Based on these state-of-the-art technologies, this paper intended to propose a holistic security management approach for IACS with the identified proper level of security details.

3 Proposed Approach

In the proposed approach, as shown in Figure 1, we divide the full security analysis scope into three domains: Threat Domain, System Domain and the Security Domain. Each domain has its specific tasks and a separate group of security professionals work on it. Threat Domain is created from security threats/vulnerabilities known to public and keeps tracking new appeared security accidents all over the world. Vice versa, the system domain consists of the system architecture model of the target IACS and its most important functions as the protection goals. After that, a group of security experts work intensively on Security Domain to bridge the previously created two domains to meet business, regulatory as well as standardization requirements by defining the threat model, performing risk analysis, managing security controls and continuously monitoring the target system. Data exchanges between the three domains can be categorized into two workflows: The setup workflow creates the security domain from the beginning. In other words, when an organization first time decides to perform a security analysis on the target

system, the setup workflow serves for that purpose to create all relevant entities within the three domains. The second workflow, or we call it the run-time workflow, aims to support IRT to manage the security domain by continuously monitoring deployed controls and warning for appeared new threats.

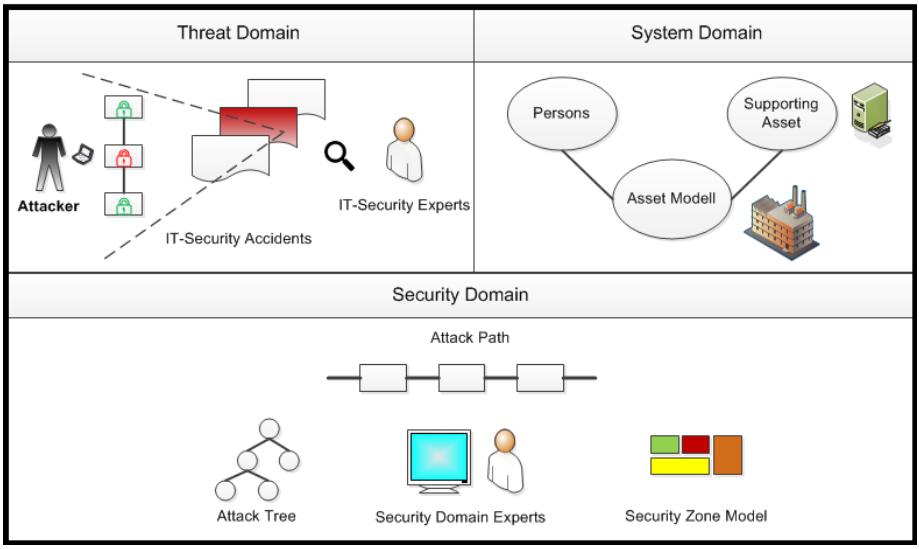

Fig. 1: The three domains of the security analysis scope.

The division of domains helps to refine a security analysis activity into different modules with specific focuses. This enables different groups of security professionals working independently. Thus an individual security professional does not need to know everything about the target IACS while workflows might be optimized. During the setup workflow, Threat Domain in general and System Domain for a specific IACS can be built up in parallel. Particularly, System Domain enables the navigation from the security concerns on top-levels down to low-levels of tangible assets and their run-time environment. The navigation bridges deployed security controls and the affected assets to support a continuous monitoring and even an adaptive defense on the target IACS.

3.1 Threat Domain

This domain takes publicly known security accidents and attacks as references for creating the security architecture to protect the target IACS. Obviously the amount of considered historical attacks can reach a huge number. Fortunately, nowadays they are categorized and maintained by security experts and published as digital libraries. Within the work of this paper, we selected CAPEC, as mentioned in Section 2, as the source of commonly used attack methods. In additional, CVE tracks security vulnerabilities of products, which can be exploited by attacker, from different vendors [MG02]. Aiming to identify security

threats while to address proper controls effectively, we also took Threat Assessment & Remediation Analysis (TARA) [Wy12] into account.

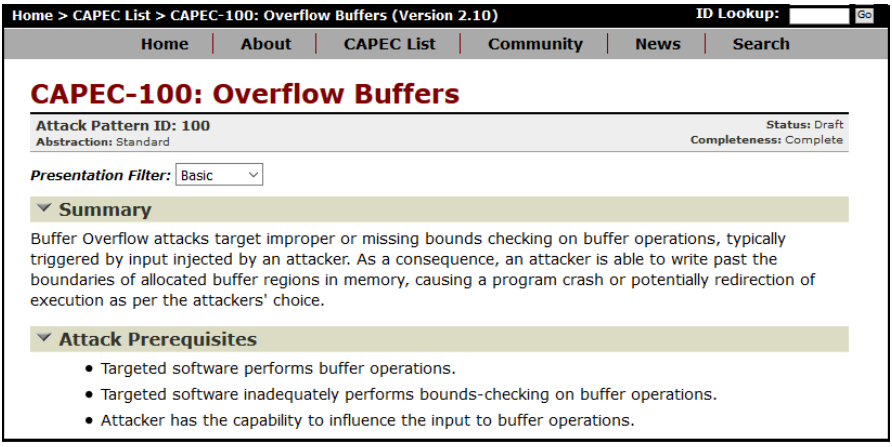

Fig. 2: CAPEC-100: Overflow Buffers.

Figure 2 shows the basic view of a well-known attack pattern with the id CAPEC-100: *Overflow Buffers* from CAPEC. An attacker can exploit a vulnerable function call by crafting the malicious input to crash the software or to execute unauthorized program. CAPEC describes this kind of attack and its prerequisites in plain text. Before adding CAPEC-100 into the threat domain, it will be modeled with additional properties to ease the later using in Security Domain. Formula (1) shows the 3-tuple for modeling a threat. *AffectedAssets* is the list of possible affected asset types so that later relevant threats can be filtered according to the assets contained by the target IACS. An asset type is an abstract type without vendor specified details (e.g. a workstation). *Rules* are required when a single attack performs on multiple assets at the same time.

$$Threat := \langle ID, AffectedAssets, Rules \rangle \quad (1)$$

The tuple for the example in Figure 2 is <CAPEC-100, ANY, NULL> where the universe symbol ANY means all assets (with software components) might be attacked using this pattern. NULL means it works directly on the affected asset. An additional example is CAPEC-285: *ICMP Echo Request Ping* which exploits the ping response to discover hosts within the network. Here *AffectedAssets* contains but not limited to PCs/Servers, PLC Communication Modules, routers or any network enabled device with an IP address. Compared to this, CVE tracks vulnerabilities in a product-base way, like CVE-2016-9159 for SIMATIC S7-300/400 PN CPUs, which are widely used PLCs. Thus a CVE vulnerability has affected single type of asset inherently. Threats described in TARA reference attacks in CAPEC, so we can easily model them according to relevant CAPEC items with a proper defined attribute *Rules*.

In summary, security professionals in Threat Domain model threats/vulnerabilities from public digital libraries and keep tracking updates from them. The 3-tuple representations will ease the searching and filtering of threats that later happen in Security Domain, corresponding to the asset-centric system architecture model (System Domain). The typical assets like server, gateway etc. introduced by Fischer et al. [Fi16] contribute to the general asset types. It is clear that additional sources, like U.S. National Vulnerability Database as well as ETSI Information Security Indicators [RG15], can be involved into the threat domain. Even individual claims or cyber security news can be taken into account too. For example, the increasing news about a current spreading ransomware can warn IRT to check whether their IACS is under the threat and to take immediate reactions when applicable. In particular, Threat Domains tracks the threats and vulnerabilities in general. Thus it does not rely on a specific target IACS. Once Threat Domain was created, it can be used for analyzing any target IACS. In addition, Threat Domain must not cover every single threats or vulnerability from those sources. We started our work from a very small filtered group. However, it should be complemented and kept updated in the future. Furthermore, against a specific target IACS, the coverage of considered issues in Threat Domain can be used as a measurement for its security posture.

3.2 System Domain

Compared to the tracked threats in general, the system domain consists of the system architecture model of an target IACS. The security analysis on an IACS is normally performed in a top-down approach, as stated in our previous work [Wa16] as well as in the industrial standard [IE15]. In the top-down approach, a complex system and its protection goals are divided into groups (e.g. hardware, software etc.) and sub-systems thus they can be easily conquered. However, the top levels lack of knowledge about those affected tangible assets (e.g. a workstation within the system). Top-down approaches are suitable for understanding the overall risks in a systematic way while it is difficult relying on them to deploy and to manage security controls. Thus alternatively, System Domain creates the system architecture model in a bottom-up manner. The process starts from modeling the centric assets which are relevant for major functionality of the target IACS, which means they are also important for security analysis. For example, a centric asset can be a sensor, an actuator or a computer with software to control the sensor/actuator. Network devices, which might be compromised to affect the major functionality, belong to centric assets too. Particularly, these assets will be assigned an unique id for later referencing in the security domain. Furthermore, the model was extended by attaching different perspectives to address required security details. The concept perspective is a reorganization of the groups of basic elements proposed by Fischer et al. [Fi16] in an asset-centric manner. These perspectives contain but are not limited to:

- **Functional Perspective** describes the specific functionality of the IACS.
- **Network Perspective** describes the network connections.

- **Data Perspective** takes the dataflow and configurations within the system into account.
- **Person Perspective** cares about individuals who (potentially) have access to one or multiple assets.
- **Resource Perspective** represents the capacity of one asset or a predefined constraint. Exceeding the capacity/constraint might affect the relevant functionality.

Within this paper, we mainly consider the functional perspective and the network perspective. The former perspective describes the important functionality of the target IACS. An important functionality can be one of the safety-critical functions or those are important for the business purpose, e.g. transferring electricity generated by a power plant into the energy grid. Failing the functionality will bring financial lost and possible safety impacts (e.g. turbine cooling) to the power plant. The items in the functional perspective are listed according to their priorities that are decided by the domain experts of the organization. Formula (2) shows the 4-tuple of a function. *AssetList* contains all involved assets. Not like the abstract asset type in the universe ANY used in Threat Domain, here an asset is a tangible one that belongs to one abstract asset type. *AssetProperty* describes associated asset details (e.g. vendor, series, configurations etc.) and further security relevant details of the function, e.g. the communication between involved assets (network perspective). In particular, network devices are part of centric assets instead of part of the network perspective so that we can model all compromising happen on centric assets rather than within any perspective. Persons can be treated also as assets. Alternatively, they are modeled separately as the person aspect, since one person might have multiple roles, e.g. as an employee and an attacker at the same time. The bottom-up process enables the asset-centric model to be independent from Threat Domain and Security Domain, since the asset model only relies on the target IACS.

$$\textit{Functionality} := \langle \textit{ID}, \textit{Description}, \textit{AssetList}, \textit{AssetProperty} \rangle \tag{2}$$

$$\forall\, a \in \textit{AssetList} \Rightarrow \exists\, t \in \text{ANY}: a \rightarrow t$$

3.3 Security Domain

This is the core domain where security experts create the “threat-attack-vulnerability” triads iteratively. On one side, publicly known threats/vulnerabilities are ready for referencing. On the other side, important functionality (protection goals) identified in System Domain are listed in the functional perspective. Then security experts work on the protection goals in the order of their priorities. For each protection goal, identified threats/vulnerabilities will be filtered according to relevant IACS assets. Then based on the filtered results, the security experts work out attack paths against a protection goal. As Formula (3), an attack path targets specific *Asset* and contains an ordered list of 2-tuple of connected assets as well as deployed attack methods from Threat Domain. The affected

Function can be omitted to be compatible with a low critical threat or an advanced persistent threat (APT). One attacker can start from one asset and go via assets according to the ordered list to reach the target *Asset* where he can interfere the affected protection goal. Compared to the kill-chain concept, attack paths focus on the reachability from the attacker to the target asset represented by the system architecture model.

$$AttackPath := < Asset, Function, \{<Asset, Threat>\}> \quad (3)$$

Each attack path can be used for describing one row in the risk assessment table so that based on it, security experts can estimate the impact and the likelihood of the risk [GP16]. How to determine the priority of a risk is out of the scope of this paper. However, attack paths need to be improved for supporting the risk assessment. On one side, two attack paths share the same ordered list of assets can be different when they employ different attack methods. On the other side, different attack paths can have the same sub-path. Both cases will cause the risk table increasing exponentially and make the following analysis and mitigation phase difficult. Thus reusing the same sub-path as well as related attack methods are meaningful. As a more compact data structure, we introduced attack/defense trees into Security Domain. Since assets are involved in attack paths can be identified by their unique Ids, attack trees can be created by union the same part of attack paths. Security experts can implement and deploy security controls to mitigate existing risks. Each security control can also be mapped to a correspondent attack tree thus the tree is adapted as an attack/defense tree. Meanwhile, the tree-like data structure enables the quantified analysis for identifying the security bottle-neck of the system where a security control can gain the maximum effect. An example of the attack/defense tree is illustrated with a running example in Section 4.

3.4 Security Professionals and Data Exchange

Security professionals and experts working within the three domains can be divided into 4 groups: The first group contains the security experts who work for maintaining those digital libraries. They work completely outside the organization (IACS) and have no knowledge about a specific IACS. The second group are domain experts work in the system domain. They belong to the organization and understand the business and functional goals of their IACS very well. They mainly work on maintaining the system architecture model and must not be professional on security topics. They also need to have meeting with the third group of security experts to determine whether attack paths are reasonable for their IACS. As stated before, the third group of security experts work in the security domain to bridge known threats to the target IACS and design proper security controls. They can be employees of the organization or external consultants. They are all professional and proficient in up-to-date security topics. If they are not employees of the organization, their knowledge about the target IACS should be constrained. Actually, by constraining information exchange between domains, security features and system details are separated so that with appropriate access controls, the risk of data leakage will be reduced. The last group of security staff work in Security Domain for the organization as

IRT members. They are not as proficient as the third group of experts in security topics. However, they will be involved in the design of security controls since later they will take the responsibility to continuously monitoring the deployed controls with the help from SIEM systems and take quick reactions to appeared threats or ongoing attacks, before senior security experts (internal and external) are in place.

The dataflow between Threat domain and Security domain is unidirectional from digital libraries to the security domain within the organization. The data exchange between Security Domain and System Domain are bidirectional. On one side, identified attack paths and implemented security controls need to be tested within System Domain. On the other side, the real-time status of security controls need to be transferred to the Security Domain so that IRT can monitor them and take reactions when needed. There is no data exchange between the Threat Domain and System Domain.

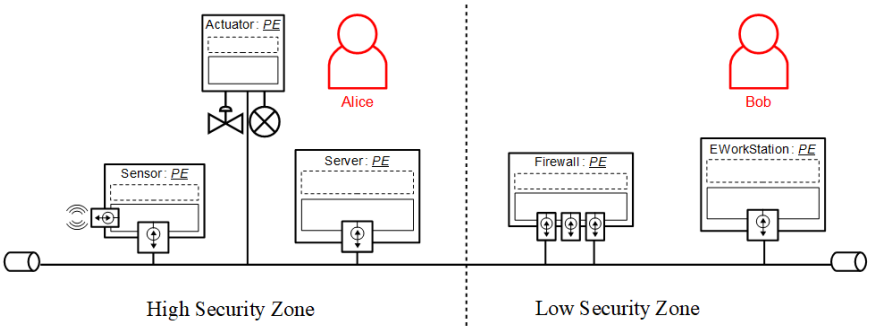

Fig. 3: A running example: pressure control system.

4 Exemplary Application

Figure 3 shows a running example following the modelling elements proposed in [Fi16] where "PE" indicates physical entities. One of the protection goal for the target IACS is its safety system. The system consists of three major components: a sensor for detecting the pressure of a container, an actuator (valve) for controlling the valve connected to the container, a server (automation computer) for executing the safety function. When the pressure within the container is too high, the server will send command to the actuator to open the valve. Thus, the inner pressure of the container will be reduced. In the next we setup the three domains briefly.

Threat Domain: a small group of threats are considered as an example (see Table 1). Their *Rules* attribute are omitted since all of them are NULL that means attacks take effect directly on assets without conditions.

ID	*AffectedAssets*	Name and Description

CPAEC-49	PC, Server, Door Locks…	Password Brute Forcing
CAPEC-100	ANY	Overflow Buffers
CAPEC-285	Server, Switch, Router…	ICMP Echo Request Ping

Tab. 1: Considered attacks in Threat Domain.

System Domain: We take the safety-critical function (pressure control) as the protection goal. The *AssetList* contains the related assets: {Sensor, Actuator, Server}. The attribute *AssetProperty* is not implemented here, but still we can read all system details from Figure 3.

Security Domain: By checking the intersections between the column *AffectedAssets* in Table 1 and *AssetList* in the system architecture model, various threats can be addressed: Targeting the actuator, Alice might connect her notebook (not illustrated in the figure) to the server and employ CAPEC-100 to inject the malicious code to block the valve. Alternatively, Bob can exploit the CAPEC-285 to find the Server thus to bypass the firewall and plays CPAEC-49 on the server to close the valve. An attacker can even attack the valve directly if it is a software embedded smart actuator. The relevant attack/defense tree is illustrated in Figure 4.

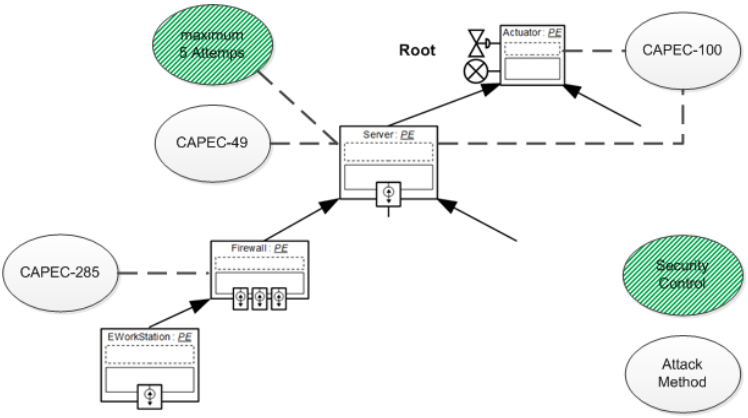

Fig. 4: An attack/defense tree example.

Knowing the two threats and the system model illustrated in Figure 4, IRT can deploy controls on the server and continuously monitor them for detecting ongoing attacks so that they can switch the valve into the manual mode to protect the system. In addition, when IRT is notified that there is a virus spreading outside the high security zone, they can take temporary control to block the ports used by the virus to protect the server. According to Figure 3, IRT can know from the system architecture model, the temporary control will not affect the safety function. So they will deploy the temporary control until senior security experts can provide a permanent solution, e.g. installing anti-virus software on the server.

5 Conclusion and Future Work

In this paper, we proposed a framework to separate the security workload for an IACS. The framework can benefit from publicly known threats and vulnerabilities while master the correct level of security details of the system architecture. Particularly, the data structure attack/defense trees are involved for easing the daily work of security professionals while providing the possibility for continuous monitoring and adaptive defense. The future work can focus on the improvement of the threat model used in the security domain. Currently the threat model focuses on functions availability and manipulations. One attack path starts from one asset and ends at the target asset. Conditional attacks or APT should be addressed in future work.

Note: Some of the above described modelling-analyses are being elaborated as part of AREVA's participation in the "SMARTEST" Cybersecurity Testing R&D with three German University partners, partially funded by German Ministry BMWi. In addition, the authors would like to thank for the project CRP J02008 funded by IAEA as well as the fertile discussion with Mr. Jackson Wynn from MITRE.

References

[Ba08] Barnum, S.: Common attack pattern enumeration and classification (capec) schema description, Cigital Inc, http://capec.mitre.org/documents/documentation/CAPEC/Schema_Description_v1, volume 4, 2008.

[Co16] Condliffe, J.: Ukraine's Power Grid Gets Hacked Again, a Worrying Sign for Infrastructure Attacks., https://www.technologyreview.com/s/603262/ukraines-power-gridgets-hacked-again-a-worrying-sign-for-infrastructure-attacks, accessed: 30/06/2017.

[De14] Deloach, S. A.; Ou, X.; Zhuang, R.; Zhang, S.: Model-driven, moving-target defense for enterprise network security, Models@run.time, pp 137-161, 2014.

[Fi16] Fischer, R.; Clausing, R.; Dittmann, J.; Ding, Y.: Industrie 4.0 Schwachstellen: Basisangriffe und Szenarien, Proceedings of DACH Security, 2016.

[GP16] Gao, Y.; Parekh, M.: Cybersecurity Risk Assessment using DBSy Models and Attack Trees, 47th Annual Meeting on Nuclear Technology: Workshop Preserving Competence in Nuclear Technology, 2016.

[GX16] Gao, Y.; Xie, X.: SIEM Framework for Policy-based Monitoring of SCADA Systems, Lecture Notes in Informatics (LNI), INFORMATIK 2016: Workshop New Security Standards for Industrial Automation and Control Systems (IACS/SCADA), 2016.

[Ha15] Hahn, A.; Thomas, R. K.; Lozano, I; Cardenasm A.: A multi-layered and kill-chain based security analysis framework for cyber-physical systems, International Journal of Critical Infrastructure Protection, volume 11, pp 39 – 50, 2015.

[Hm09] HMG: Information Assurance Standard No. 1, 2009.

[Ie14a] IEC: 62714-1: Engineering data exchange format for use in industrial automation systems engineering, Automation markup language, Part 1: Architecture and general requirements, 2014.

[Ie14b] IEC: 61131-10 (Draft): PLCopen XML Exchange Format, 2014.

[Ie15] IEC: Security for Industrial automation and control systems – Part 3-2: Security risk assessment and system design, 2015.

[II11] ISO, IEC: 27005 Information technology – Security techniques – Information security risk management, 2011.

[IP12] ISO, IEC: 17506: Industrial automation systems and integration, COLLADA digital asset schema specification for 3D visualization of industrial data, 2012.

[Ko13] Kordy, B.; Kordy, P.; Mauw, S.; Schweitzerm P.: ADTool: security analysis with attack/defense trees, International Conference on Quantitative Evaluation of Systems, pp 173 – 176, 2013.

[LH15] Liu, T.; Huuck, R.: Case study: Static Security Analysis of the Android Goldfish Kernel, International Symposium on Formal Methods, pp 589 – 592, 2015.

[Ma17] Lee Mathews: How WannaCry Went From A Windows Bug To An International Incident, https://www.forbes.com/sites/leemathews/2017/05/16/wannacry-ransomware-ms17-010/#27267a252609, accessed: 30/06/2017.

[MG02] Mell, P.; Grance, T.: Use of the common vulnerabilities and exposures (cve) vulnerability naming schema, national INST of standards and technology Gaithersburg MD Computer Security DIV, 2002.

[MI14] Mao, A.; Iravani, M R.: A trend-oriented power system security analysis method based on load profile, IEEE Transac. on Power Systems, volume 29, pp 1279 – 1286, 2014.

[PSF17] Perlroth, Nicole; Scott, Mark; Frenkel, Sheera: Cyberattack Hits Ukraine Then Spreads Internationally., https://www.nytimes.com/2017/06/27/technology/ransomware-hackers.html, accessed: 30/06/2017.

[RG15] Rennoch, A.; Gaudin, G.: ETSI ISG SI: Security Indicators Quick Reference Card (V 1.1.2), http://cloudsecurityalliance.it/wp-content/uploads/2015/01/isiQRC_V1_1_2_june2015.pdf, accessed: 05/07/2017.

[Wa16] Waedt, K.; Parekh, M.; Tong, X.; Gao, Y.; Ding, Y.; Xie, X.: Nuclear Safety and Risk-based Cybersecurity Testing, 47th Annual Meeting on Nuclear Technology, 2016.

[Wy12] Wynn, J.; Whitmore, J.; Upton, G.; Spriggs, L; McKinnon, D.; McInnes, R.; Graubart, R; Clausen, L.: Threat Assessment and Remediation Analysis Methodology Description, https://www.mitre.org/publications/technical-papers/threat-assessment--remediation-analysis-tara, accessed: 30/06/2017.