1. Rationale

In the development of Cyber-physical systems (CPS), security becomes an important property. More and more CPS are connected to the internet and, therefore, the attack surface of such systems grows. CPS often operate in safety-critical contexts, e.g., cars or critical infrastructure. Hence, securing the system is essential to guarantee the safety of the system and, therefore, a method for developing CPS secure-by-design is needed. Since CPS interacting with their physical environment, the method should also take this special nature of CPS into account.

We need a method that allows us to model, analyze, and verify security requirements for CPS. Since CPS have to satisfy safety requirements, MDSD became one leading paradigm for developing such systems. Thus, it seems natural to apply a method for security requirements of such systems to these already used functional models, e.g., system architecture, behavior description, allocation, and deployment.

There are several threat modeling and security analysis approaches. The question is, to which extent are these approaches suitable for the requirements of model-driven development of CPS? In particular, we focus on the question to which extent is the hardware/platform of CPS considered in model-driven security approaches. Thus, we conduct a structured literature review in this area to a) find out, which requirements of CPS are satisfiable by existing model-driven methods, b) to which extent current approaches consider the platform of the system, and c) where are current research challenges. Thus, the main focus of the survey is to find out, which approaches does consider both software description ("cyber"-layer) and platform description ("physical"-layer).

Requirements

We defined a set of seven requirements a platform-aware approach should satisfy. For this, we conducted an unstructured literature search. We use existing surveys (see appendix) found in this initial search for creating our requirements.

We define the following requirements, which are to our opinion the most important. However, we do not claim the list to be complete.

- R1 – Support different system layers: One key aspect of CPS is that they consist of different layers, i.e., a cyber layer for the distributed aspect of the system and for communication, as well as a physical layer for interaction with the physical world via sensors and actuators. Since in this survey we focus on the use of the platform, this requirement has some significance. R2 – Formal methods and formal models: Security-by-Design stems from applying formal analyses and model transformations to enable consistency between all development steps. Thus, the method has to provide i) formal models for the system and the threats/attacks, ii) formal analyses on these models, and iii) consistency (correct refinement) of the models to source code of the system.
- R2 – Phases of the SDLC: Security-by-Design means to consider security in all phases of the SDLC. Thus, all phases of the SDLC should be covered. First of all, the system has to be designed, e.g., by modeling the software architecture and the target platform. Furthermore, potential threats and attacks have to be specified, e.g., by a threat model. Additional phases are the application of formal analyses, (automatic) threat mitigation, deployment, or runtime analyses like monitoring or simulation. It is important that each phase is clearly defined and has specified input and output artifacts.

- R3 – System-of-Systems: Modern CPS are complex systems that consist of several sub-systems. To handle the development of such system-of-systems, the method has to provide a structured approach, e.g., to allow compositional analyses, it has to provide the specification of a hierarchical system specification. In large systems, not all subsystems are developed by the same provider. Thus, complex systems often integrate third-party resources. Hence, in the best case, a method is able to process both fully known parts and only partially known (or even unknown) parts of the system.
- R4 – Threat Model: Since the threat model defines potential threats to the system, it is important that such a model is sufficiently expressive; it should cover as many kind of threats as possible. In the best case, the threat model is extensible, allowing one to define new threats. We do not restrict the threat model to the application layer but also consider threats regarding the platform and hardware.
- R5 – Formal Methods: Security-by-Design is aided by applying formal analyses and model transformations to enable consistency between all development steps. Thus, the method has to provide formal models for the system and the threats and attacks as well as formal analyses on these models.
- R6 – Refinement: An approach has to provide a correct synthesis to source code, i.e., a source-code generation that preserves the analysis results, e.g., by generating secure source code for the platform. In the best case, a full code generation for the is provided but also partial code generation for the security implementations might be sufficient, e.g., if the code for secure communication is generated. Other code generations are possible, e.g., automatically generated test cases or static code analyses.
- R7 – Supported Requirements: Typically, beside functional requirements and security requirements also non-functional requirements are essential for CPS, since CPS are often restricted by resource constraints, e.g., restricted memory or computing power. Restricted computational power and timing constraints get important properties and make the secure development of CPS more difficult, e.g., when encryption is needed. Also, such systems have to formally show that specific (hard real-time) safety-requirements are fulfilled. Hence, an approach has to provide the specification and/or verification of such non-functional requirements. In the best case, an approach has to allow the specification and verification of functional, non-functional, and security requirements.

2. Research Questions

With our survey we tend to answer the following research questions:

Q1: To which extent do model-driven security approaches for CPS consider the platform?

Q2: Which of the other stated requirements (cf. R2-R7) do these approaches fulfill?

Q3: What are current challenges and open research questions in the area of such platform-aware approaches for model-driven security?

3. Search Strategy

For searching, we are going to use the following online libraries:

- ACM Digital Library: https://dl.acm.org
- IEEE Explore: http://ieeexplore.ieee.org
- Springer Link: http://www.springerlink.com
- Science Direct: https://www.sciencedirect.com

We define three set of keywords we see as important:

1. Secure, Security, Threat, Attack
2. CPS, Cyber-Physical System, embedded+distributed
3. Model-Driven, MDSD, MDE, AOM, Aspect Oriented, Method+Model

We will search for publications that have at least one keyword of (1.) and (2.) or at least one keyword of (2.) and (3.): (1 AND 2) OR (2 AND 3).

Since we figured out problems using the automatic search when using too many keywords (e.g., SpringerLink provided wrong results for some uses of the AND operator), we decided to apply the search for each combination manually in each library and merged all results per library using Mendeley Desktop. Please finde the search strings for each library in the file SearchStrings.txt.

4. Study Selection Criteria

To set the focus on mature approaches only, we focus on approaches published as

- Conference papers, Journal articles, and book chapters only

We exclude articles from the initial set, when:

- Short papers / work in progress papers
- Domain-specific (not related) context, e.g., cloud or web services
- Informal approaches (like brainstorming, models for visualization only…)
    - Informal threats
    - Informal functional model
    - No model-driven approach
- Only one layer supported

- Only one step of the SDLC considered, e.g., only threat modeling without a functional model
- Not written in English
- Not accessible

5. Inclusion and Exclusion Strategy

After the initial search in the online libraries, we will use a step-wise strategy applying at each step our exclusion criteria:

1. Exclusion by format
2. Exclusion by title
3. Exclusion by abstract
4. Exclusion by full-text
5. Following relevant references (snowballing depth: 2-3)

Finally, we are going to cluster publications describing the same approach.

6. Data Extraction

We created a data extraction form which is provided as a template in a separate file „DataExtractionTemplate.xlsx“

Appendix - Other Surveys:

Uzunov, A. V., Fernandez, E. B., & Falkner, K. (2012). Engineering Security into Distributed Systems: A Survey of Methodologies. *Journal of Universal Computer Science*. http://doi.org/10.3217/jucs-018-20-2920

Nguyen, P. H., Klein, J., Traon, Y. Le, & Kramer, M. E. (2013). A Systematic Review of Model-Driven Security. In *2013 20th Asia-Pacific Software Engineering Conference (APSEC)* (Vol. 1, pp. 432–441). http://doi.org/10.1109/APSEC.2013.64

Shafi, Q. (2012). Cyber Physical Systems Security: A Brief Survey. In *Computational Science and Its Applications (ICCSA), 2012 12th International Conference on* (pp. 146–150). http://doi.org/10.1109/ICCSA.2012.36

Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., & Halgand, Y. (2015). A survey of approaches combining safety and security for industrial control Systems. *Reliability Engineering & System Safety*, *139*, 156–178. http://doi.org/10.1016/j.ress.2015.02.008