Threat Analysis in Practice – Systematically Deriving Security Requirements

Conference Paper · November 2018

DOI: 10.1007/978-3-030-03673-7_25

Markus Fockel
Fraunhofer IEM
20 PUBLICATIONS **67** CITATIONS
SEE PROFILE

Masud Fazal-Baqaie
Next Data Service AG
34 PUBLICATIONS **83** CITATIONS
SEE PROFILE

Threat Analysis in Practice – Systematically Deriving Security Requirements

Markus Fockel, Sven Merschjohann, and Masud Fazal-Baqaie

Software Engineering and IT Security, Fraunhofer IEM, Paderborn, Germany
{firstname}.{lastname}@iem.fraunhofer.de

Abstract. With the growing number of incidents, the topic security gains more and more attention across all domains. Organizations realize their lack of state-of-the-art security practices, however, they struggle to improve their software lifecycle in terms of security. In this talk, we introduce the concept of security by design that implements security practices within the whole software lifecycle. Based on our practical experience from industry projects in the regulated industrial automation and unregulated classical IT domain, we explain how to perform a threat analysis and how to integrate it into the software lifecycle.

Keywords: Threat Analysis · Security by Design · IEC 62443

1 Introduction

Across all domains cyber-security gains more and more attention as the number of reported incidents is increasing. Due to the growing degree of software within products and services, the potential attack surface increases while at the same time attacks are getting more professional. In order to counter this disturbing trend, organizations across all domains need to improve on their security measures. Indeed, in our daily practice, we often observe the lack of state-of-the-art security practices and thus a lot of potential for improvement. Some industries react by establishing security standards to enforce the use of proper security practices. For example, the standard IEC 62443 [2] regulates the security for industrial automation and describes process requirements for security. But also in the classical IT domain where no such standard exists the awareness for the lack of security practices is growing. In this talk, we introduce the concept of security by design that integrates security practices within the whole software lifecycle (Section 2). We highlight "threat analysis" as its first step to identify security requirements. Based on our practical experience from industry projects in the industrial automation and classical IT domain, we explain how to perform threat analysis and how to integrate its results into the software lifecycle for reaching security by design and compliance with standards (Section 3). We demonstrate the threat assessment based on STRIDE [3], and explain that adjusting the used tools to the context of the organization ensures an efficient analysis and process workflow. We conclude the talk with our assessment of the benefits of threat analysis (Section 4).

2 Security by Design

In order to systematically develop secure products and services, security must be emphasized throughout the whole software lifecycle, such that the results are secure by design. Thus, every step of a common software lifecycle is enhanced with security practices as illustrated in Figure 1. The shown high-level process can be further tailored and refined towards specific domains and system classes [1] and many aspects also correlate with security standards for specific domains, e.g., IEC 62443 for industrial automation.

The very first security practice is the *threat analysis*. It is used to identify the system's assets, its security objectives, threats to it and associated risks. Using these results of the threat analysis a *secure design* can be created, which specifically targets the identified threats with appropriate countermeasures on an architectural level. This ensures that the risk of the threats is reduced to an acceptable level and the product design is secure. However, the realization can still introduce vulnerabilities, e.g., due to flaws in the code like buffer overflows. Therefore, *code analysis* and code reviews need to be applied to ensure secure code. In addition, each identified threat should be the basis for a test case, such that its mitigation can be verified. In practice, many vulnerabilities are caused by faulty deployments. Hence, the process demands the *enforcement* of secure configurations. Lastly, the process also needs to cover the long-term security by implementing a functional *patch management*, in order to quickly fix vulnerabilities after the product or service has been released.

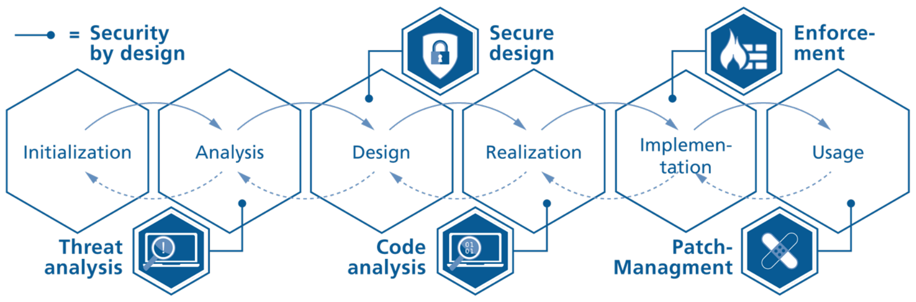

Fig. 1. Security by Design: Security practices throughout the whole software lifecycle

3 Security Practice: Threat Analysis

As a first step towards security by design, we apply threat analysis and integrate it into existing development processes. We integrate it, for example, by integrating the tool used for threat modeling with existing application lifecylce management (ALM) tools. Consequently, identified and modeled threats can be refined as and traced to requirements and test cases in the ALM tooling. In the following, we illustrate the five steps of threat analysis and provide practical advice based on several projects across various domains.

1. Specify System and its Security Context In our projects, we often use the Microsoft Threat Modeling Tool (TMT) as tool support[1] to specify the system under development as illustrated in Figure 2. Here, elements within the system (circles) and external to the system (rectangles) as well as different zones of trust (denoted by red, dashed borders) define the system boundaries and which elements are more trustworthy than others. In addition, it is important to specify assumptions about security measures that are provided by the system's context (the so-called security context). The more security measures are taken by the system's context, the less measures need to be provided by the system itself. For instance, if we can rely on the operating system's user management, we do not need to develop our own login mechanism. It is crucial to document the assumptions about the security context.

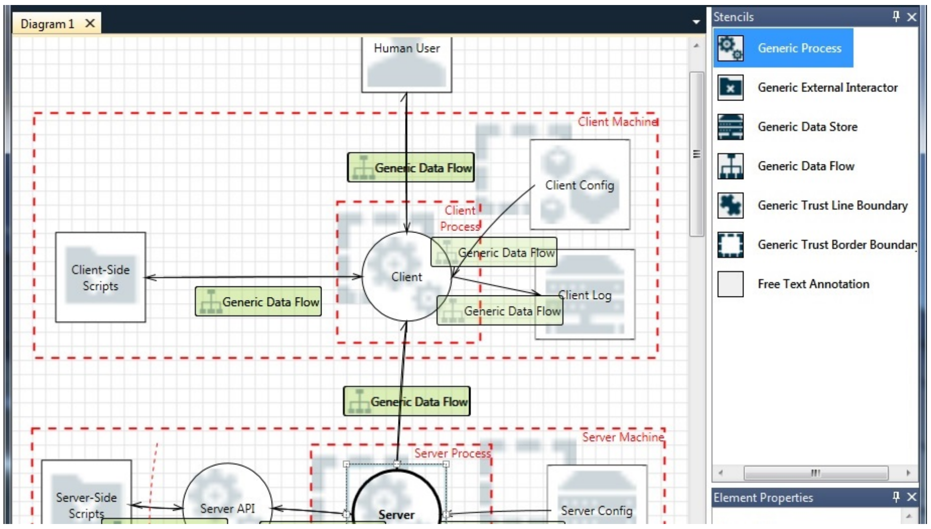

Fig. 2. A threat model in the Threat Modeling Tool

2. Determine Assets and Security Objectives In the second step, the assets of the system are identified. For all elements it is determined what security objectives need to be achieved and what needs to be safeguarded, e.g., "confidentiality of customer data in the database" or "availability of the server".

[1] https://www.microsoft.com/en-us/download/details.aspx?id=49168

3. Identify and Localize Threats In this step, possible threats for the parts of the system are identified. We use STRIDE [3] for categorization and systematically go through all communication flows that cross zones of trust to brainstorm about possible threats. While performing this step in workshops with system experts, it is helpful to have a list of possible threats as a starting point to foster discussions. TMT has an automated threat generation feature. However, it is important to customize its generation template to avoid superfluous generated threats, ultimately leading to frustrated participants.

4. Define Countermeasures For each threat that is identified, countermeasures have to be defined. This does not necessarily lead to implemented security mechanisms within the system. Countermeasures can also be taken care of by the system context beyond the system boundaries. Stakeholders may also choose to accept the risk by open threats without countermeasures. There is a n-to-m relation between threats and countermeasures, e.g., one threat can require a set of countermeasures to reduce the residual risk to an acceptable level.

5. Assess Risk of Threats In order to prioritize threat mitigation tasks and to assess the effectiveness of countermeasures, the risk associated with threats is assessed. For instance, IEC 62443 requires to assess a risk value for the unmitigated threat and a residual risk once countermeasures are in place. There are different ways to assess risk, for early threat prioritization, we use Protection Poker [4] as part of threat modeling workshops with system experts, as it produces quick results with low effort.

4 Conclusion

In order to counter the growing trend of security incidents, organizations need to improve their security measures. As illustrated, this can be achieved by systematically integrating security practices within the software lifecycle, thus achieving security by design. Based on the practical experience from various industry projects, we explained how to perform threat analysis as the first security practice in the process and how to integrate it into the lifecycle for compliance with standards.

References

1. Geismann, J., Gerking, C., Bodden, E.: Towards ensuring security by design in cyber-physical systems engineering processes. In: International Conference on Software and System Process (ICSSP 2018) (May 2018)
2. International Electrotechnical Commission (IEC): IEC 62443-4-1:2018: Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements (Jan 2018)
3. Shostack, A.: Threat Modeling: Designing for Security. Wiley (Feb 2014)
4. Williams, L., Meneely, A., Shipley, G.: Protection poker: The new software security "game". IEEE Security & Privacy **8**(3) (May 2010)