Systematic Identification of Security Goals and Threats in Risk Assessment

Daniel Angermeier, Alexander Nieding, Jörn Eichler
Fraunhofer AISEC

Assessing security-related risks in software or systems engineering is a challenging task: often, a heterogeneous set of distributed stakeholders create a complex system of (software) components which are highly connected to each other, consumer electronics, or Internet-based services. Changes are frequent and must be handled efficiently. Consequently, risk assessment itself becomes a complex task and its results must be comprehensible by all actors in the distributed environment. Especially, systematic and repeatable identification of security goals and threats based on a model of the system under development (SUD) is not well-supported in established methods. Thus, we show how the systematic identification of security goals as well as threats based on a model of the SUD in a concrete implementation of our method Modular Risk Assessment (MoRA) supports security engineers to handle this challenge.

1 Introduction

Security risk assessment in software or systems engineering is a challenging task. We developed the method Modular Risk Assessment (MoRA) [EA15] to tackle this challenge. Our method features a modular structure, supports a unified method framework, well-defined artifacts as interfaces between activities, and different guidelines as well as catalogs to implement the method in a specific domain and environment. Identification of security goals and threats is one of the key challenges within the application of any security risk assessment method and vital to the validity of the results. Thus, we present an implementation of MoRA which supports systematic identification of security goals and threats based on a hierarchical model of the system under development (SUD), where security goals represent the combination of a security goal class (confidentiality, integrity, availability, ...) with an asset of the SUD (e.g., integrity of the asset "billing function" for an online shop).

The remainder of this publication is structured as follows: In Section 2, we elaborate on related work. Section 3 highlights how we systematically identify security goals and threats based on a model of the SUD. We use Section 4 to describe an approach for a systematic derivation of relationships between security goals. We give an example for an application of our method in Section 5. Finally, we conclude in Section 6.

2 Related Work

Several comprehensive standards and publications with strategies for the assessment and the management of security risks exist. ISO 31000 [ISO09] combined with ISO 27005 [ISO08] form a framework for the management of risks with a focus on information security. Nevertheless they do not explicate applicable methods for the identification of security goals and threats. Risk analysis based on a baseline protection approach is defined in the standards BSI 100-1 [BSI08a] and 100-3 [BSI08c] by the German Federal Office for Information Security. These standards are quite specific on the identification of security goals but do not elaborate on threat identification. Furthermore, they are not designed for the application in a development environment. In this section, we briefly introduce several approaches presenting concrete methods for a systematic threat identification and subsequently for risk evaluation with the claim to produce repeatable and traceable results.

Attack trees are seen as a viable option to identify and represent threats against a system. Consequently, some authors discuss options on the systematic or even automatic generation of attack trees. Exemplary, Vigo et al. [VNN14] propose an approach to automatically infer attack trees from a process algebraic specification. Knowledge of cyber-physical systems and attackers is recorded in algebraic models and attack trees are computed. Aside from threat trees, Microsoft's STRIDE Threat Model [Sho14] provides a structured, systematic approach to threat modeling. Based on data flow diagrams (DFDs) and a defined set of possible threats which can be applied to certain elements of a DFD, this approach offers a low-formalism approach to identify threats systematically.

In an extension to this, Roy et al. [RKT12] unify attack and defense trees to support risk estimation. An attack countermeasure tree combines defense mechanisms, attack scenarios, probabilistic risk

values, and prioritization of attack events as well as countermeasures.

As another example, Weldemariam et al. [WV11] propose a methodological approach to procedural security analysis. After building and reasoning on an extended system model, possible attacks are identified and related to affected assets and their properties (e.g., their value to the organization), analyzed, and evaluated to produce sets of security requirements, which establish a certain level of protection.

Most of the academic approaches highlighted in this section feature a high grade of formalism (and are, consequently, very demanding in terms of depth of analysis and respective effort) or are limited to a distinct aspect of risk assessment procedure (e.g., to threat analysis). By contrast, the approach we outline in the following section aims to be applicable in a broad area of potential settings with results that are easy to comprehend by domain experts.

3 Systematic Identification of Security Goals and Threats

MoRA features four core activities in its method framework: "Document SUD", "Determine Protection Needs", "Analyze Threats", and "Analyze Risks", each supported by a set of guidelines and further artifacts.

Some preparative activities prepare the method for its application in a specific domain: MoRA relies on an *assessment model* and a set of *catalogs* to homogenize assessments within the domain of application. Thus, the assessment model and the catalogs represent a common basic understanding of all actors regarding critical aspects of risk assessment.

The assessment model primarily contains rules how to estimate the impacts of violations of security goals as well as instruments to estimate the required attack potential to execute an attack or to overcome protective measures. For example, to support impact estimation, a list of damage criteria maps potential damages (e.g., "loss of 10.000 - 50.000 $") to damage potentials (e.g., "moderate"). This helps focus impact estimation on domain facts ("is the damage between 10.000 and 50.000 $?") instead of personal opinions ("I think the damage is moderate"). While this particular approach has been derived from the standard BSI 100-2 [BSI08b], other strategies to determine the need for protection are possible.

The aforementioned catalogs entail generalized but pre-evaluated elements used in the method, such as threats and controls. Their purpose is to aid the user in the process of determining what to protect, how to attack the elements in need of protection and how to protect the SUD against it. The threat catalog for example describes threat classes - recurring threats generalized for the analysis of different SUDs with a common baseline estimation of the required attack potentials for the execution of these threats.

The activity *Document SUD* provides the basis for MoRA's model-based approach, where security engineers and domain experts decompose the SUD into functions, data, components, and connections. The model contains several relations between these elements, forming a graph: elements can be refined into sub-elements of the same kind. Functions describe behavior and functionality provided by the SUD and require data, components, and their connections to be executed. Connections link components to each other. Finally, Data is stored on components or transmitted using these connections. For example, a hardware component can be decomposed into a CPU, flash memory and other components. If necessary, components (including their connections) and data can be further refined into lower levels of granularity. Relations between components are thereby usually modeled on the same level of granularity. Following these strategies, this activity creates a unified representation for the SUD which supports both tracing of changes and systematic identification of security goals and threats.

The next activity *Determine Protection Needs* systematically identifies security goals for the SUD based on the model: we combine each element of the model of the SUD with each of the *security goal classes* in the assessment model (e.g., confidentiality, integrity, availability, authenticity), resulting in *potential* security goals (e.g., "confidentiality of data *private key*", but also "confidentiality of data *public key*"). For each of these potential security goals, we determine damage potentials based on the assessment model. All potential security goals with non-zero damage potential represent actual security goals, i.e., properties of the SUD that require protection. For example, "confidentiality of patient data" in a medical system handling patient data represents a security goal, as a violation results in loss of privacy. The relationships between the elements of the SUD also imply relationships between the security goals. For example, the integrity of a function depends on the integrity of data elements required by the function as well as on the integrity of components processing the function. This approach derives the security goals and their relations from known information, namely the SUD, the security goal classes, the damage criteria, and the MoRA method itself. We will go into detail in Section 4. This can also be used to inherit the estimation of single damage criteria or whole damage potentials to associated security goals. Consequently, the results are traceable, comprehensible, and systematically identified. Furthermore, changes to the SUD can easily be traced and integrated into the risk assessment.

Once the security goals are identified, we switch from the defender's to the attacker's perspective to identify potential threats to the SUD's security goals in the activity *Analyze Threats*. Again, a systematic approach is applied: For all security goals, we identify threats based on the model of the SUD and a cata-

log of possible threats. To identify applicable threats, we consider the security goal's class and its relation to the model of the SUD. For example, to identify threats to the security goal "confidentiality of patient data", we consider all relationships to other security goals and elements of the SUD. If the transmission of "patient data" over connection X is part of the model of the SUD, then the threats "information disclosure: eavesdropping on a connection" and "information disclosure: eavesdropping as Man-in-the-Middle" are applicable to the combination of "confidentiality" and "interface". Likewise, threats "information disclosure: reading from memory" on each of the components linked by connection X must be considered. The use of "tags" may also help in identifying appropriate threats: annotating components, connections and data elements with (technology) tags (e.g. "LTE, "Ethernet", "IEEE 802.11" ...) further narrows down the selection of suitable threat classes. For example, we can estimate the required attack potential to execute a threat based on the combination of a set of risk factors (such as required expertise, time, knowledge ...). This approach originates from the Common Methodology for Information Technology Security Evaluation[1] and shows one possible strategy in evaluating the likelihood of an attack. This evaluation can be simplified through estimates taken from the associated threat class. Additionally, known vulnerabilities of the SUD modify this estimation by reducing the required attack potential accordingly.

Identifying security goals and threats separately yields an important benefit: the model of the SUD and the security goals provide solid ground based on the domain experts' knowledge of the SUD and its environment, while threats are identified based on experience from the security domain. All aspects (i.e., SUD, goals, and threats) can be updated independently at first and necessary changes can be propagated systematically in consequence.

Finally, we assess risks based on the estimated attack potentials for identified threats and the damage potentials of security goals they threaten in the activity *Analyze Risks*, again according to the assessment model.

4 Systematic Derivation of Relationships for Security Goals

In this section, we consider two sorts of relationships for security goals, based on the model of the SUD and the security goal classes: first, security goals can support each other, inheriting damage potentials from supported security goals. For example, security goal 1 *availability of the database server* supports security goal 2 *availability of the function patient data retrieval* in an SUD where the patient data is stored on the database server. Thus, security goal 1 inherits the damage potentials of security goal 2. Second, we consider relationships between attacks on elements of the SUD and associated security goals. For example, an attack on the database server's availability in the previous example will impact all availability-related security goals for functions executed on the server, data processed (sent, received, or stored) by the server, or the server itself. For the security goal classes and the "supports-relationship" between security goals, the following assumptions are made:

- A security goal of class confidentiality is restricted to supported security goals of class confidentiality
- A security goal of class availability is restricted to supported security goals of class availability
- A security goal of class integrity is not restricted by default, as, e.g., a manipulated device can become unavailable or leak data

We follow a similar approach for threats: a threat threatens one or more security goal classes and applies to an element of the SUD, which implicitly targets a potentially virtual security goal for the SUD, that is, even if no security goal for the availability of a connection is defined, all affected security goals can be derived for the threat by assuming that security goal existed.

In addition, the following inputs are processed for the SUD:

1. All elements in an element's hierarchy
2. All functions an element is mapped to
3. All data produced, received, or stored by a component
4. All data transmitted over a connection
5. All connections linked to a component

Please note that all derived relationships represent suggestions which must be checked carefully by the analyst. Nevertheless, the systematic derivation of these relationships both improves completeness and reduces complexity, as the analyst can focus on a small set of relevant security goals. The hierarchy (1) is a natural starting point for consideration: If the confidentiality of data X is broken, it is plausible to assume that this also holds true for all data contained in X. (2) is an explicitly modeled aspect of the SUD: Data, components, and connections support functions. Thus, if the integrity of a component is affected, then the integrity of all functions (partially) computed on that component is affected as well. Additionally, the availability and the confidentiality of these functions are also threatened. (3) and (4) follow the same idea: data is handled by components or transmitted over connections; therefore, security goals for these components or connections also affect all security goals of

[1] https://www.commoncriteriaportal.org/cc/

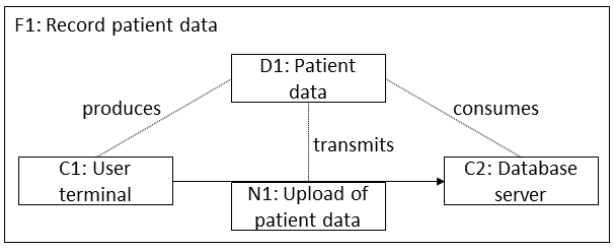

Figure 1: Depiction of the SUD

related data. Finally, (5) represents the fact that connections depend on the components which establish those connections.

5 Example application

In this section, we show a very simple example for the application of our method for a patient data recording system with an artificial assessment model.

Our assessment model for this example encompasses the damage potentials "Low", "Medium", and "High". The model also contains the following damage criteria and associated damage potentials:

- DC1: Delayed treatment of patient $\mapsto$ Medium
- DC2: Incorrect (and potentially dangerous) treatment of patient $\mapsto$ High
- DC3: Financial loss less than 1.000$ for the clinic $\mapsto$ Low
- DC4: Privacy violation $\mapsto$ Medium

The SUD, as depicted in Figure 1 consists of the function "F1: Record patient data", the data "D1: Patient data", the components "C1: User terminal (produces D1)" and "C2: Database server (consumes D1)", and the connection "N1: Upload of user data D1 from C1 to C2 via mobile communication". All of the data, components, and connections are mapped to F1. We construct potential security goals for all elements except the connections, as we can create virtual security goals for the connections on demand as described in the previous section. Figure 2 shows this process. Next, we assign damage criteria from our example assessment model. For example, we get the following mapping for the function's security goals:

- Availability of F1 $\mapsto$ DC1 (Medium), DC3 (Low), because the medical examiner will not be able to retrieve patient data from the database, resulting in a delayed treatment and less productivity. Overall, this assigns the damage potential "Medium" to this security goal
- Confidentiality of F1 $\mapsto$ DC4 (Medium), as patient data is involved
- Integrity of F1 $\mapsto$ DC1 (Medium), DC2 (High), DC3(Low), DC4 (Medium), as the manipulated function may even result in wrong treatment for a patient.

The availability of "C1: User Terminal", by contrast, has no assigned damage potential, as the medical examiner can switch to another terminal with very little loss of time in our example. For the security goal

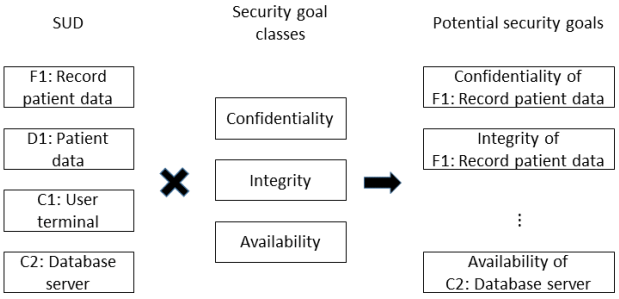

Figure 2: Generation of potential security goals

SG_{I-C1} "integrity of C1: User terminal", we can examine the supported security goals for our assessment: we know that data "D1: Patient data" is produced on C1, therefore, the confidentiality, availability and integrity of D1 is supported by SG_{I-C1}. Likewise, as D1 is mapped to F1, we can deduce that the security goals for D1 support the security goals for F1 according to the rules for security goal classes. Thus, SG_{I-C1} supports the "integrity of F1: Record patient data" among others, leading to an assigned damage potential "High". The same rationale concludes that manipulation of "C1: User terminal" violates the security goal "integrity of F1: Record patient data", assigning the damage potential "High" to that threat. Figure 3 shows the trace between the elements of the SUD used for this derivation of relationships based on our small exemplary SUD.

6 Summary and Conclusion

After a brief summary of the MoRA method, we sketched a systematic procedure to identify security goals based on a given modeling technique in combination with defined security goal classes and damage criteria. Thereafter, we showcased the identification

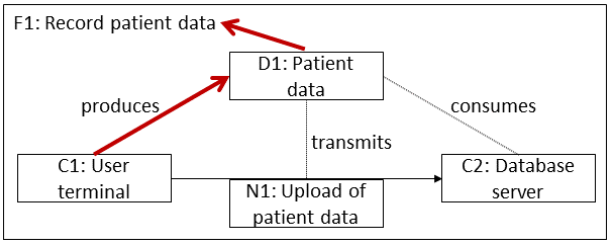

Figure 3: Relationships between elements of the SUD used to deduce relationships between security goals

of potential threats to the previously identified security goals by taking the information available in the system model into account and combining them with a provided threat catalog.

Early experiences from the application of this method and the presented guidelines show that our systematic and guided approach induces a good understanding of the subject matter and produces reproducible and comprehensible assessment results.

References

[BSI08a] BSI. Standard 100-1: Managementsysteme für Informationssicherheit (ISMS). *Bonn: Bundesamt für Sicherheit in der Informationstechnik*, 2008.

[BSI08b] BSI. Standard 100-2: IT-Grundschutz Vorgehensweise. *Bonn: Bundesamt für Sicherheit in der Informationstechnik*, 2008.

[BSI08c] BSI. Standard 100-3: Risikoanalyse auf der Basis von IT-Grundschutz. *Bonn: Bundesamt für Sicherheit in der Informationstechnik*, 2008.

[EA15] J Eichler and D Angermeier. Modular risk assessment for the development of secure automotive systems. *31. VDI/VW-Gemeinschaftstagung Automotive Security*, 2015.

[ISO08] ISO. Information technology — security techniques — information security risk management. ISO/IEC 27005:2008, International Organization for Standardization, Geneva, Switzerland, 2008.

[ISO09] ISO. Risk management — principles and guidelines. ISO 31000:2009, International Organization for Standardization, Geneva, Switzerland, 2009.

[RKT12] Arpan Roy, Dong Seong Kim, and Kishor S Trivedi. Attack countermeasure trees (act): towards unifying the constructs of attack and defense trees. *Security and Communication Networks*, 5(8):929–943, 2012.

[Sho14] Adam Shostack. *Threat modeling: Designing for security.* John Wiley & Sons, 2014.

[VNN14] Roberto Vigo, Flemming Nielson, and Hanne Riis Nielson. Automated generation of attack trees. In *Computer Security Foundations Symposium (CSF), 2014 IEEE 27th*, pages 337–350. IEEE, 2014.

[WV11] Komminist Weldemariam and Adolfo Villafiorita. Procedural security analysis: A methodological approach. *Journal of Systems and Software*, 84(7):1114–1129, 2011.