Defining a Security-Oriented Evolution Scenario for the CoCoME Case Study

Roman Pilipchuk, Stephan Seifermann, Emre Taspolatoglu
{pilipchuck, seifermann, taspolat}@fzi.de
FZI Research Center for Information Technology, Karlsruhe

Abstract

Information systems are subject to continuous change. In order to conduct empirical research on methods for software evolution, CoCoME was developed as a community-driven case study system. It is, however, not suitable for the validation of security-related approaches, as neither security nor privacy have been addressed in any evolution scenario. We elicited 53 security requirements coming from law, security guidelines and known threats. In this paper, we present three out of twelve security requirement categories including one representative requirement and share our experience in building the foundation for a security-oriented evolution scenario. Researchers in the field of secure software evolution can validate their approaches using this future evolution scenario.

1 Introduction

Companies operate information systems over decades. These systems are subject to continuous change. According to Lehman [1], this process of updating various parts of the system for different reasons is called software evolution. Case studies enable validating the effectiveness of software evolution methods. These case studies evolve through joint collaboration in certain communities. CoCoME, the Common Component Modeling Example, is such a community case study for collaborative empirical research on software evolution approaches [2]. The resulting case study platform consists of three parts: artifacts like requirements and source code representing the information system, an evolution scenario describing the evolution subject, and interconnected activities implementing one or more evolution scenarios [5]. CoCoME is a comprehensive case study representing a retail system of a supermarket as illustrated in Figure 1. It started as a monolithic system consisting of one enterprise, with an enterprise server and database, a chain of stores, where each store server is connected to the enterprise server, and each store server containing a cash-desk line, where each cash-desk PC is connected to the store server. Cashiers scan products at the cash desk. Customers pay by credit card or cash.

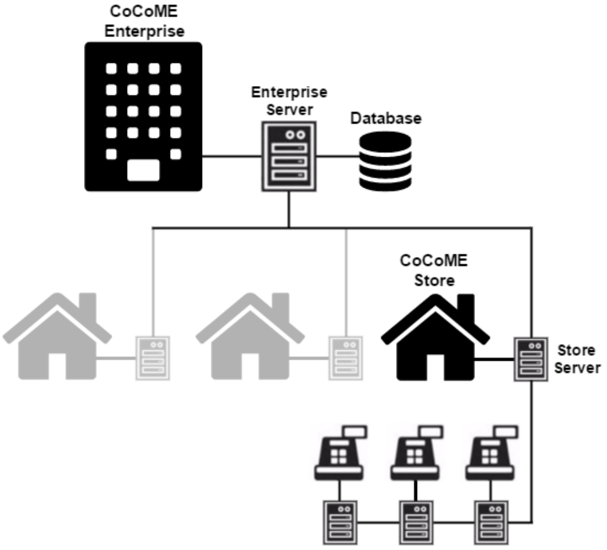

Figure 1: Simplified CoCoME architecture

One evolution scenario [9] for the previously closed system is migrating the enterprise server and database into the cloud and introducing a web service for customers. The business goal, is to reduce operating costs and establish a flexible adaptation and reconfiguration of these system parts to meet new customer requirements. With this evolution scenario, use cases change along with system boundaries. Thus, CoCoME becomes a more open system, as now several system parts interact with other information systems in the internet. This not only raises the potential attack surface, but also paves the way for new threats, for example data breaches in cloud systems [11]. The bigger attack surface, law compliance, and security of critical business data require dedicated handling of security. CoCoME, however, does not address security aspects such as confidentiality, integrity, and privacy in any evolution scenario. Thus, validation of security-related approaches is not possible yet.

In order to make CoCoME representative in terms of security and privacy, a security evolution scenario has to be defined. This evolution scenario has to start with business goals, define corresponding security requirements and develop use cases on this basis. To address security systematically in all system parts of CoCoME, the security requirements have to be elicited from various sources like known attacks on web services, security guidelines for information systems and cloud, as well as laws concerning trading systems. As there are, neither a security evolution scenario nor ready to use security requirements for CoCoME, we started to define a security evolution scenario.

This evolution scenario defines how CoCoME will address security and privacy issues. Our goal was not to provide a complete set of security requirements but to define requirements representative for the European Union (EU) which establish a minimum level of security for the overall CoCoME system. We analyzed the German law for IT-supported accounting systems (GoBD [4]), the security guidelines for information systems (IT-Grundschutz [3, 7]) based on ISO 27001 and 27002, an EU-approved cloud certification guideline (Cloud Security Certification (CSC) TÜV [12]), and some favorable and known attacks on web services [13, 15]. Except of GoBD, all sources are intuitively representative for the EU. Therefore, we generalized the GoBD requirements by replacing concrete requirements such as the retention time of data with placeholders. After we had elicited requirements from these sources, we checked their applicability on CoCoME and dropped requirements which were not reasonable. For instance, we omitted organizational requirements regarding filling a vacancy. For each remaining requirement, we mapped use cases and CoCoME system parts that have to change. Requirement categories structure elicited security requirements regarding their similarity, always keeping source traces.

The resulting security requirement list consisted of twelve categories and 53 requirements. Eight of these 53 requirements were not assigned to a specific category, as there was no other similar requirement for each of them. The security requirements serve as a foundation for the development of a security evolution scenario for CoCoME. This evolution scenario will establish security and privacy in CoCoME enabling us and other researchers to use CoCoME as a case study for the validation of security-related approaches. Further, the introduced categories can be reused for example in other trading systems as a starting point and structure for the elicitation of security requirements.

In this paper, we share our experience in defining a CoCoME security evolution scenario and exemplify its usage. Section 2 reports on three representative security requirements from different categories. A discussion of our methodology and our experiences takes place in Section 3. Section 4 concludes the paper.

2 Excerpt of Security Requirement Categories

Because trading systems such as CoCoME have a variety of stakeholders, different sources serve as a basis for the elicitation of security requirements. These sources, as described in the previous section, correspond to different aspects of security. After elicitation, the requirements are categorized into several categories, based on their similarities such as goals, applications, etc. For the sake of brevity, we only describe representative requirements that show major aspects of the three categories shown in Table 1. Each of these categories combines results from various sources.

Category	**Requirement**
Confidential Communication	Secured transfer of sensitive information
Access Control	Access control system
Data Separation	Data validation and sanitization

Table 1: Summary of suggested categories and representative requirements.

2.1 Confidential Communication

Secure communication summarizes requirements on information exchange between system parts regarding confidentiality and integrity. In this context, confidentiality means that no unauthorized party can intercept communication and gain knowledge about transmitted information apart from size and frequency of messages. Integrity means that no third party can alter communication unnoticed. Out of five requirements, two requirements describe general secure communication. The remaining three requirements define specific communications to be secured.

A typical requirement is to ask for “secured transfer of sensitive information”. The meaning of security matches the definition above. Sensitive information is not clearly specified because the requirement originated from the cloud certification guideline of TÜV Rheinland [12], which does not target a specific domain such as point-of-sale systems. For CoCoME, a) the information about the customers and users of the system, as well as their credentials and payment data, b) sale-related communication, and c) information about ordering or exchange of products is sensitive. To be more general, all communications between coarse-grained components are affected. Losing security properties for one of these communications might lead to serious impact on the business.

The secure communication requirements allow us to validate approaches that address confidentiality and integrity for communication links. The approaches can perform very basic security tests such as detecting anti-patterns of unencrypted links in a way threat modeling would do. More sophisticated approaches that, for instance, determine security properties of certain data classes can use the requirements that are more specific about the information to protect. The latter is subject to our current research [10].

2.2 Access Control

The category access control describes a group of similar security requirements which describe how the access to the information systems data, resources and services shall look like and how this access shall be managed. Precisely, this category contains requirements which pertain the access restriction from inside and outside, as well as the system that enforces access control. Requirements concerning appropriate authorization and therefore authentication, like user authentication mechanisms, as well as management and administration of all needed information, belongs also to this category. Eleven requirements, one from the law GoBD, seven from the security guideline IT-Grundschutz and three from the security guideline CSC TÜV are part of this category.

One typical requirement is called "access control system" and is derived from the IT-Grundschutz of the Federal Office for Information Security (BSI) [3]. It demands individually designed access rights for all roles (e.g. cashier, store manager) in CoCoME. The design of access rights has to be aligned with the role's tasks to allow for performing all its tasks without any disruption. It also demands establishing of a system for management and enforcement of access rights over all various CoCoME information systems.

Every security-related approach with focus on access control will benefit from the realization of this requirement in CoCoME. In our research, we focus on a specific access control paradigm, namely role-based access control. We elicit role models with a hierarchy containing access rights by analyzing business processes. Therefore, CoCoME needs dedicated access control to be used as a validation for our security-related approach. Once it will be established, we will use it as a case study for illustration and validation of the algorithm which elicits the roles and their access rights.

2.3 Data Separation

Data separation handles a broad context. It becomes especially vital with cloud-based applications, which leverage virtualized environments sharing the underlying resources. Data separation can be defined as the separation of information or resources from each other, such as personal or commercial data of one customer from another. Cloud applications are multi-tenant, which raises multiple questions: Are data and resources residing on the same virtual environment and on the same physical infrastructure safe? For instance, the iObserve approach [8] addresses this issue implied by an evolution scenario regarding database migration and privacy laws. The approach determines the physical location of the database and checks if the database remains in a permitted set of locations. As you see, the aspect of data separation is more relevant to data privacy or compliance, and there are regional as well as international institutions trying to regulate such issues or introduce countermeasures like separation or geo-localization of private data [14], [6].

Data separation is a countermeasure for several attacks that become possible by introducing the pick-up shop [9] to CoCoME, where customers make purchases over the internet and pick up their merchandise later at the store. Attackers can leverage the new interface, i.e. the according web service, for threatening the CoCoME system. Data separation has to be strongly considered in this area. So, one concrete requirement of the data separation category is defined as the tuple of "data validation and sanitization". Assume an attacker with capabilities to issue an XSS attack, e.g. persisting cross-site scripting, which is one of the most common input vulnerabilities in web-based applications [16]. Without going further into detail, a very conceptual mechanism to avoid easy persistent cross site scripting is data separation. For example, keeping untrusted data separate or at least validated from the active browser content is not the separation of any data, but decoupling of data and applications. An example would be an interpreter behind the web client that needs proper input data for further processing. This is a different issue from multi-tenancy, since it is not just addressing the cloud-based application logic of CoCoME, but is also relevant for other components in its architecture, such as the pick-up shop components representing the web interface.

Elicitation of requirements based on some known attacks is also a bit different than elicitation from existing laws and regulations, which are well-documented, maintained and generally accepted. Deriving possibilities of attacks from use cases and system properties and using them as necessary information sources for deciding the requirements can be error-prone, compared to two other approaches. Still, interpreting the attackers as the "malicious" users of the system and their attacks as the "miss" use cases, has the advantage of being the lowest-level approach to the CoCoME cloud application. This and the necessary modeling of threats are subject to our research as well.

3 Discussion

We want to discuss the completeness and representativeness of our results, as well as share our experience a) in using the types of sources mentioned above, and b) in using the methodology to mine the resulting requirements and categories.

The most important properties of our results are completeness and representativeness. We did not focus on completeness intentionally. Therefore, the results cannot be complete with respect to possible sources. We, however, argue that this is not necessary or even possible. A system can never be completely secure. Therefore, even complete requirements cannot make a system completely secure, which invalidates the benefit of completeness. In contrast, we focused on representativeness, which makes resulting requirements and evolution scenarios applicable for more approaches. We consider our results representative because we covered various source categories: general implementation guidelines cover technical lowlevel requirements, certification guidelines cover high-level technical requirements, and basic security requirements (IT-Grundschutz) cover the state of the art. We even considered laws and derived technical requirements. The correctness of requirements stems from the correctness of our sources. The categorization done by three people can be considered correct.

Our experiences in using the mentioned sources vary depending on the type of source. Working with sources considering technical aspects was straightforward: requirements regarding deployment or infrastructure already provide the system design granularity that matches the CoCoME granularity. We raised the abstraction level of requirements regarding implementation details such as encryption algorithms used or discarded them if this was not possible. This step makes the requirement elicitation fuzzier, but is necessary because technical requirements often target implementations rather than design. Compared to technical sources, we spent more effort in mining juristic sources. First, we dropped all organizational requirements such as hiring a data protection officer because these aspects are not covered by CoCoME yet. Second, we extracted technical requirements by understanding and interpreting laws. Checklists based on laws could save effort, but are not possible because laws usually have to be interpreted for a given use case. Thus, handling legal sources in a reproducible and systematic way is not possible.

The methodology we applied provided us with considerable results. We were able to mine requirements from various sources following a predefined process. In contrast to this, consolidating these requirements is pretty tough: even if we were able to define categories for requirements to group them according to overlapping concerns, summarizing a category is a highly creative process. The major challenge is finding a representation that covers all aspects of the input requirements and still keeps the summarized requirement checkable in the system to be implemented. We, however, argue that summarizing requirements is not always necessary. For defining an evolution scenario, simply ensuring that all requirements of a category can be realized is sufficient.

4 Conclusion

In this paper, we presented three out of twelve security requirement categories for CoCoME as well as the methodology for mining them and the contained requirements. We want to use the 53 security requirements for defining a CoCoME evolution scenario that provides a case study for validating security-related evolution approaches. We argued that the results are sufficient for this purpose. The major challenges in creating the categories were the handling of legal sources and summarizing requirements.

Researchers evaluating security evolution approaches, i.e. approaches that target access control, confidentiality, integrity, and authenticity in the architectural design phase, benefit from the requirements and the evolution scenario to be defined

Future work includes reviewing the consolidated requirements and releasing a technical report that covers all requirements as well as the definition of a CoCoME evolution scenario.

References

[1] M. M. Lehman and L. A. Belady. *Program evolution : processes of software change*. Academic Press, 1985.

[2] S. Herold et al. "CoCoME - The Common Component Modeling Example". In: *The Common Component Modeling Example*. Vol. 5153. 2008, pp. 16–53.

[3] German Federal Office of Information Security. "IT Security Guidelines - IT-Grundschutz in brief". 2012.

[4] German Federal Ministry of Finance. *Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff*. 2014.

[5] R. Heinrich et al. "A Platform for Empirical Research on Information System Evolution". In: *SEKE*. 2015, pp. 415–420.

[6] E. Schmieders, A. Metzger, and K. Pohl. "Architectural Runtime Models for Privacy Checks of Cloud Applications". In: *PESOS*. 2015, pp. 17–23.

[7] German Federal Office of Information Security. *IT-Grundschutz Catalogues*. https://www.bsi.bund.de/EN/Topics/ITGrundschutz/ITGrundschutzCatalogues/itgrundschutzcatalogues_node.html. Online; accessed 18.12.2016. 2016.

[8] R. Heinrich. "Architectural Run-time Models for Performance and Privacy Analysis in Dynamic Cloud Applications". In: *ACM SIGMETRICS* 43.4 (2016), pp. 13–22.

[9] R. Heinrich, K. Rostami, and R. Reussner. *The CoCoME Platform for Collaborative Empirical Research on Information System Evolution*. Tech. rep. 2016,2; Karlsruhe Reports in Informatics. KIT, 2016.

[10] S. Seifermann. "Architectural Data Flow Analysis". In: *Proceedings of WICSA*. 2016, pp. 270–271.

[11] Top Threats Working Group. *The Threacherous 12 - Cloud Computing Top Threats in 2016*. https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf. Online; accessed 18.12.2016. 2016.

[12] TÜV Rheinland i-sec GmbH. "Anforderungskatalog Certified Cloud Service v3.0". 2016.

[13] *CAPEC - Common Attack Pattern Enumeration and Classification*. https://capec.mitre.org/. Online; accessed 05.05.2016.

[14] *Multi Tenancy and Physical Security*. https://www.owasp.org/index.php/Cloud-10_Multi_Tenancy_and_Physical_Security. Online; accessed 20.12.2016.

[15] *OWASP Top Ten Project*. https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. Online; accessed 20.12.2016.

[16] J. Williams, J. Manico, and N. Mattatall. *XSS (Cross Site Scripting) Prevention Cheat Sheet*. https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet. Online; accessed 17.08.2016.