A Cross-Layer Security Analysis for Process-Aware Information Systems

Maria Leitner[1], Zhendong Ma[1], Stefanie Rinderle-Ma[2]
[1]Austrian Institute of Technology,
Digital Safety & Security Department
[2]University of Vienna, Faculty of Computer Science,
Research Group Workflow Systems and Technology
{maria.leitner, zhendong.ma}@ait.ac.at
stefanie.rinderle-ma@univie.ac.at

Abstract

Information security in Process-aware Information System (PAIS) relies on many factors, including security of business process and the underlying system and technologies. Moreover, humans can be the weakest link that creates pathway to vulnerabilities, or the worst enemy that compromises a well-defended system. Since a system is as secure as its weakest link, information security can only be achieved in PAIS if all factors are secure. In this paper, we address two research questions: how to conduct a cross-layer security analysis that couple security concerns at business process layer as well as at the technical layer; and how to include human factor into the security analysis for the identification of human-oriented vulnerabilities and threats. We propose a methodology that supports the tracking of security interdependencies between functional, technical, and human aspects which contribute to establish a holistic approach to information security in PAIS. We demonstrate the applicability with a scenario from the payment card industry.

Keywords: Information Security; Process-aware Information Systems; Cross-layer Analysis; Human-oriented Security Analysis;

1 Introduction

Humans are the weakest link when it comes to security in everyday business processes (e.g., [26] and [35]). For example, the 2014 data breach report of [40] shows that as of 2013, more than 85% of data breaches stem from external agents, more than 10% from insiders and around 1% from business partners. Research shows that for example insiders (i.e., employees) intentionally circumvent security measures for three reasons ([22]): First, employees *lack* *security-awareness* and have no incentive to apply security-compliant behavior. Secondly, employees skip security procedures in order to fulfill their job effectively. The cost of adherence to compliance is too high. Lastly, compliance is *impracticable* and employees choose different ways to fulfill their jobs. However, this non-compliant behavior increases the likelihood for vulnerabilities because it is hard to control (see [5] and [42]). While business still keeps running, such behavior is also difficult to detect until an incident occurs.

Business process can be seen as a vehicle to integrate the human, functional, and technical perspective in an enterprise. As humans are an essential part for fulfilling and managing business processes [20], it is important to incorporate them when it comes to security in business processes. Although security in Process-aware Information Systems (PAIS) is a interdisciplinary research field that faces many challenges such as human orientation or an agreement of technology, it often centers on the specification of access control mechanisms (cf. [24]). However, these mechanisms are often neither evaluated with users nor directly visible to the user and find ways to bypass it (e.g., [5] and [2]). Most research that incorporates human involvement focuses on the design and specification of security requirements in process modeling languages via extensions. For example, modeling extensions for UML or BPMN (e.g., [45] and [34]) exist that represent specific restrictions such as privacy, integrity, or confidentiality requirements on tasks. However, the concrete implementation on the technical level or user studies that evaluate the e.g., comprehensibility or cognitive effectiveness are often missing.

In this paper we aim to address two main research questions: (a) how to conduct a cross-layer security analysis that couples security concerns at business process layer as well as at the technical layer; and (b) how to include human factor into the security analysis for the identification of human-oriented vulnerabilities and threats. With question (a), we target to identify security concerns that *go beyond the business process* and to investigate technical aspects and human aspects that contribute to a thorough and systematic security analysis. Question (b) aims to provide a *holistic view* that shows that human involvement is as important as technical aspects when it comes to securing business processes.

In order to address questions (a) and (b), security is investigated as a cross-layer concern in PAIS architectures and a multilayer PAIS architecture model is provided that incorporates humans as essential factor spanning across the business process, system design, and implementation layers. The multilayer PAIS model is used as an input to generate a method for a human-oriented security analysis that contains four steps: business process analysis, function mapping, technical analysis, and human factor analysis. Moreover, we analyze and evaluate our method with a scenario from the payment card industry. With this examination, we show that humans are a key factor in the design, development, and execution of secure business processes. The approach follows the conceptual framework set out in [10].

The method proposed in this paper contributes to security engineering in PAIS but also to other research disciplines (e.g., human-centric PAIS [20]). We envision that it can be used for systematic, risk-driven security analysis of PAIS and guiding the design and development of the system.

The paper is structured as follows: Section 2 analyzes security in business processes based on a multilayer model that incorporates technical and human aspects. Section 3 describes the method for the human-oriented security analysis with an application scenario from the payment card industry. The main findings, limitations, and impact on research and practice are outlined in Section 4. Section 5 describes related work and Section 6 concludes the paper.

2 Security in PAIS Architectures

Security in business processes is an interdisciplinary field that links security mechanisms from the business process management and security research community (cf. [24]). When considering security from a holistic, engineering point of view, we have to assess not only the system and the architecture, but also the human-related aspects of the business process. To establish a structured view of the PAIS architecture, three layers that are relevant to security analysis can be identified (e.g., [3]):

- the **business process layer** represents the functional organization,
- the **system design layer** centers on the specification of the system architecture and its components,
- and the **implementation layer** incorporates the technologies utilized to implement the system.

In addition, a fourth **human layer** is considered orthogonally to the other three layers. In the following, we will discuss each layer in more detail.

2.1 Business Process Layer

The business process layer centers on the processes when organizing and managing work in an organization (cf. [11]), i.e. the functional organization. This includes not only the process participants but also the outcomes and the customer values of processes. Business processes can be found in every business domain such as in banks, hospitals, or retail stores. Attacking a business process requires full or at least partial knowledge of the *participants* in a process and on the *activities* performed.

2.2 System Design Layer

System design defines system architecture, system components and their interfaces, and data. Putting together, they realize the specification of business processes. System design can be done at different levels of abstraction, from conceptualization and modeling of systems to selection of concrete technologies and system artifacts. System design follows engineering principles which usually includes analysis, design, and verification.

Generally, a system architect needs to consider common ICT components such as hardware, software, database and networks. Within the system architecture, software components such as applications and services perform various actions defined by a business process. During the design, the input/output to the software components and their behaviors are considered to a great extent to ensure that all components will behave according to the design. Also considered are the component-component and component-user interactions.

With respect to information security, all of the variants at the system design layer can have profound security implications on business process security, as well as on system implementation. To achieve security by design, a set of security controls need to be analyzed and planned for securing business processes (cf. [24]). Besides, security testing can be carried out at design time (cf. [28]).

2.3 Implementation Layer

The implementation layer is concerned with the technologies utilized to implement the system design and covers an extensive set of technologies (e.g., web services, programming languages), PAIS products (e.g., SAP, ERP, or Oracle), and protocols (e.g., transport protocols, networks) which can be utilized to implement and execute business processes. Hence, the underlying technology is a key factor when it comes to security in PAIS. For example, a web service-based application system can be attacked with e.g., denial of service, XML, or SOAP attacks (cf. [19, 41]). With the increase in use of different technologies and system complexity, the number of vulnerabilities increases and therefore the attack surface and attack vector.

Depending on the technology used, different outsiders might be more interested or motivated to attack. For instance, a web-based application might be an easier target for non-professional (e.g., script kiddies) and professional hackers than a closed system (which might be only interesting for hired professionals).

2.4 Human Layer

In business processes, manual (i.e. performed by a human) and automated tasks (i.e. performed by a machine or program) are distinguished. Typically, humans access PAIS based on their roles that represent job functions or organizational affiliations ([13]). To clearly identify human involvement and its impact on security, we analyze human activities based typical roles they take in business processes (see [11, 43, 21, 32]). Typically, managers (i.e. *chief process officer*, *business engineers*, and *process owners*) use dashboards or control panels to evaluate and track business processes (such as handover time between tasks). *Process designers*, *system architects*, and *developers* are responsible for the design and implementation of business process models and systems and the set up of security controls. *Process participants* and *knowledge workers* actively participate in business processes. *Subjects* participate passively in the business process (e.g., as a patient in a medical procedure). *Business partners* take part in a business process choreography where public views on the internal processes are visible to the partners. Each business partner has to comply with a defined set of rules or a set of contractual agreements (see [43]). This implicates also security-related policies ([6, 17].

Employee, Business partner, Contractor, Nation state, Professional hacker, Non-professional hacker is a list of attacker roles adapted from commonly used attacker types in security literature and practice, and the threat agents library defined in [7]. From an information security point of view, humans are the main actor that passively or actively influences the security of a system. Generally, humans have two types of "negative" influences on security: they either introduce vulnerabilities in terms of flaws or mistakes in the design, implementation, configuration, and operation of the system ([23]), or pose threats as attackers to exploit the vulnerabilities and comprise the security of the system ([26, 31]). Obviously, different roles typically connected to a business process are able to introduce vulnerabilities and launch attacks. For example, social engineering attacks are well known for acquiring passwords or other information from humans in order to gain access to information systems ([15]). Hence, common tasks in risk assessment are to analyze and estimate how likely a part of the system could be attacked, by whom, and with what means. Although actual attacks are often done by crafted input, computer instructions or programs (e.g. malware), the origins of the attacks can always be linked to humans. Therefore, identifying potential attackers and their roles is crucial for security analysis of PAIS.

3 Cross-Layer Security Analysis

This section proposes a methodology for cross-layer security analysis of business processes including a human-oriented perspective. The methodology is based on the PAIS architecture (cf. Section 2) and divided into four steps. Each step analyzes one layer, i.e., business process analysis, function mapping, technical analysis, and human factor analysis. The novelty of the methodology is its holistic nature. Each of the layers contains different and relevant information for security analysis. Moreover, the interrelations between the layers are an interesting source for security analysis. Take for example data flow. There is a process data flow which connects tasks at the business process level by their input and output parameters. In addition, data is exchanged between the process tasks (business process layer) and the invoked applications (system layer), e.g., web services. Based on the methodology, security of process-oriented applications can be investigated. It can be also used for security compliance checking.

In the following, we will use a running example to examine each step of the analysis.

Example 1 (Credit Card Blockage and Renewal) *Figure 1 displays a credit card blockage and renewal request (using BPMN notation, cf. [32]). The process*

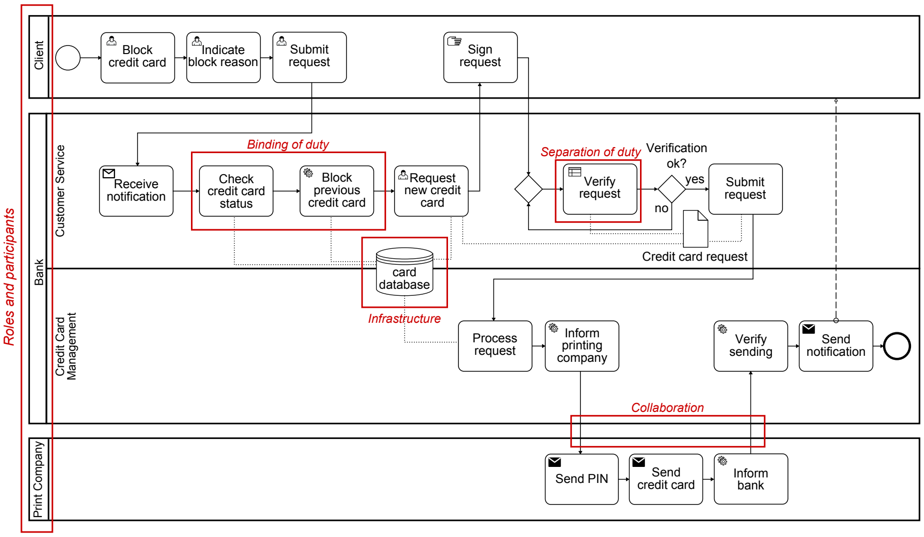

Figure 1: Example 1: Credit Card Blockage and Renewal

contains several participants: a client that has a credit card, a bank that issues credit cards, and a printing company that prints and sends out the cards. In the example process, a client orders the blockage of a credit card due to card theft or other reasons. Therefore, the client requests the suspension of the previous card and orders a new credit card by signing a request form. The customer service receives the notification of the blocked card, checks the credit status, and blocks the credit card. Furthermore, a binding of duty (BoD) constraint is imposed on the `Check credit card status` *and* `Block previous credit card` *tasks, i.e. both tasks have to be performed by the same person. To verify the request, two employees of the customer service have to check and sign the request. This procedure is often called "four-eyes-principle" or separation of duty (SoD) constraint (see [4]). After a successful verification, the request is submitted to the credit card management department that processes the request and informs the printing company. After several working days later the client receives a personal identification number (PIN), a credit card, and an additional notification letter by mail (with several days difference between the letters).*

3.1 Business Process Analysis

Analyzing the business process layer identifies the following information: process tasks as functional building blocks, process participants and their roles, security-related constraints, information about the system infrastructure, and possible business collaborations. Let us illustrate this based on the running example:

(a) *Identify participants and their roles:* Usually, organizational information is (partly) available in business process models. Depending on the notation, it might be represented in a different way. In BPMN as standard notation, pools represented the different involved participants (organizations, partners), specifically, client, bank, and printing company in Figure 1. Furthermore, *roles of process participants* are specified as swimlanes, here the customer service and the credit card management for the bank. If information on participants and roles is not (sufficiently) available, it can be acquired via, e.g., interviews (cf. [24]).

(b) *Analyze security-related constraints* such as Separation of Duty (SoD) and Binding of Duty (BoD) which are realized by access control mechanisms that perform authentication and authorization of both users and programs when they access data or use computer resources. In the business process in Figure 1, a BoD constraint is defined for tasks `Check credit card status` and `Block previous credit card` to ensure that the same person performs both tasks. In addition, the task `Verify request` contains a business rule that states that two different people have to verify the request. This SoD constraint is imposed to prevent error and fraud (see [4]).

(c) *Denote system infrastructure:* The business process layer partly indicates the *IT infrastructure* behind the business process. It can be seen from Figure 1 that a *card database* exists that stores the credit card data. Furthermore, the figure contains automated tasks that connect, for example, the bank with the printing company.

(d) *Register business collaborations:* The usage of pools in the figure shows that this business process is a collaboration and therefore might require security measures (e.g., encryption or digital signatures).

3.2 Function Mapping

The function mapping investigates the functional building blocks (i.e. tasks) of a business process and their incorporation in the system design architecture. Note that we are aware that large-scale business processes can contain a vast amount of tasks and the complexity and effort of such analysis would increase quickly. However, in order to minimize the complexity, we recommend to perform the functional mapping and further steps only on tasks that can be classified as security-critical[1].

Assume that task `BlockPreviousCreditCard` has been identified as being security-critical. Figure 2 displays its function mapping. The main purpose of the task is to block the credit card and write the required data to a card database. The first layer in Figure 2 displays the constraints at business process layer that can be derived based on the business process model in Figure 1. In

[1] In order to determine which tasks are security-critical, for example, the method proposed by [39] can be followed where the decision bases on a discussion between business and security analyst.

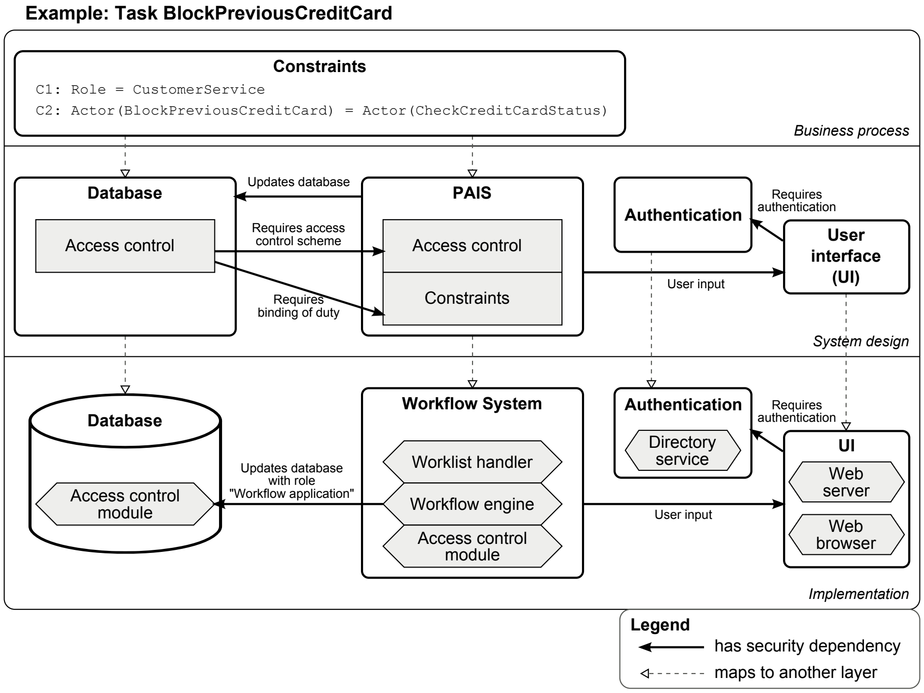

Figure 2: Function Mapping and Technical Analysis of Task `BlockPreviousCreditCard`

particular, constraint $C1$ assigns a role to the task and $C2$ defines a BOD constraint (see previous section).

Based on these requirements, we can adopt the function into the system design layer where the systems are outlined and their interconnections. In particular, a database, a PAIS, an authentication module, and a user interface (UI) are identified. The PAIS and database contain modules (e.g., access control) that should align with the constraints defined in the business process layer. It can be seen from Figure 2 that depicting the security-aware components and their interactions just for a single task is challenging as it includes different software components that interact with each other.

3.3 Technical Analysis

Based on the system design in the function mapping, the technical analysis is concerned with the deployment and execution of a single task in a business process.

Continuing with Example 1, the system infrastructure is explicitly defined in the implementation layer in Figure 2. In particular, the UI and the authentication service enforce a proper authentication and authorization of users. Furthermore, a workflow system interacts with a database in order to store the processed data. Note that in this example, the role accessing the database is not *CustomerService* but is *WorkflowSystem*. Hence, the access control restrictions $C1$ and $C2$ defined in the business process layer are eventually enforced in the workflow system. The user role information is typically not transmitted to the database application.

The actual realization of the functions and system depends on the existing enterprise IT environment, available technologies and resources. The result is very likely to be a combination of vendor-specific BPM software and customized software implementations. Figure 3 shows an example of the data flow diagram (DFD) of the system that is often used in threat modeling (cf. [38]) for identifying potential IT security threats. All interactions with the user happen on the *Browser*. A *Web server* responds to user requests and data input with constructed web content. The *Authentication Server* handles a user's login requests to the system. The *LDAP Directory* stores the user directory information. The *BPM Server* is where the workflow engine locates which handles the workflow and access control and constraints specified in the BPM rules in the *BPM Repository*. The *Business Application Server* hosts various applications or web services that are invoked by the *BPM server*. The applications on the *Business Application Server* read and write to the *Card database* through the *Data Management System* which provides access control to the *Card Database*, among other functions.

The access control restrictions $C1$ and $C2$ are thus the function mappings of the requirements to various components and sub-systems. However, this imposes several risks for security: First of all, the constraints defined on the business process layer and drafted on the system design layer are not consistently transferred to the implementation layer. At implementation level, the access control restrictions such as $C1$ and $C2$ are not enforced in the database. The access control restrictions are specified and enforced in the workflow system but the database does not distinguish the updates from the workflow system; they are all made by the role *WorkflowSystem*. Hence, a human-specific verification of the data entries in a database cannot be performed – only based on the logs of the workflow system. Furthermore, this opens the possibility to access the database given access control restrictions (such as role assignments and constraints) in the workflow system were overcome.

From an information security point of view, more security risks need to be considered. The system might be distributed geographically and across different trust domains. For example, the main system can be within a *Internal Trust Boundary*, the *Web server* needs to be placed in a *Controlled zone* (e.g. a Demilitary Zone (DMZ)) because it might be subject to various networked attacks originated from the external entities from the internet. The DFD in Figure 3 identifies the trust boundaries, data flow, and entry points. In a systematic security analysis, each of the elements need to be considered for security risks. This means even though the access control constraints from the business process layer can be achieved by the functions provided by applications and services such as authentication and workflow engine, any compromise at the system or code level can easily subvert these attempts. Therefore, the functional mapping

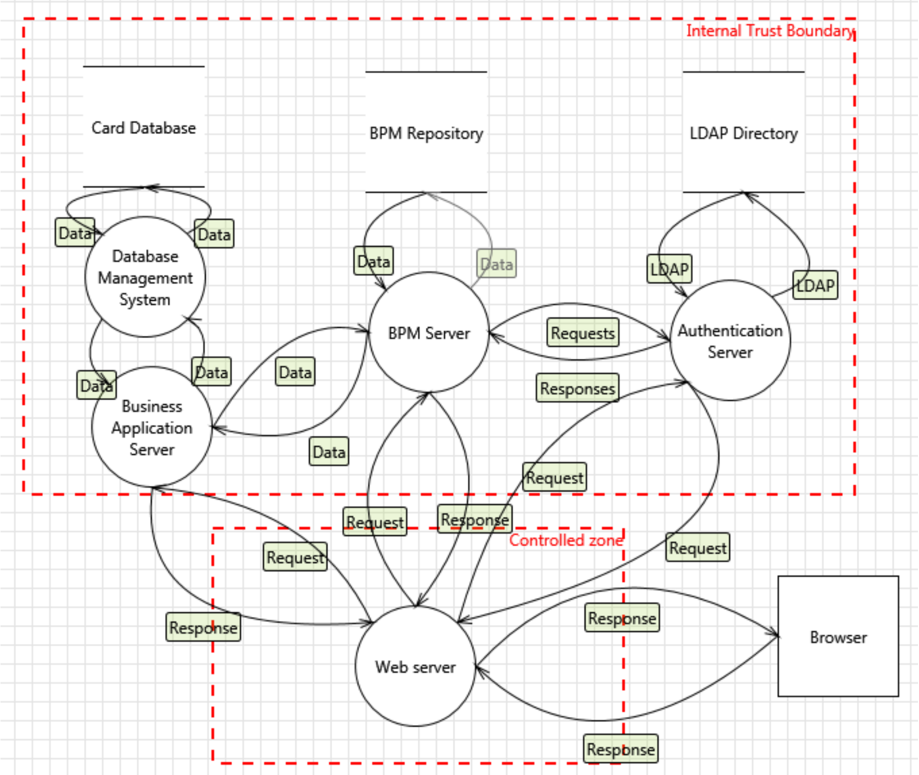

Figure 3: Data flow diagram of system architecture

provide a means for consider business process security and information security at the same time.

3.4 Human Factor Analysis

Table 1 shows example risks and solutions that can be drawn based on the human-oriented cross-layer analysis. Some of the solutions are common security practice. However, other solutions raise interesting new research questions, e.g., consistency checks of access constraints implemented at different layers.

The human factor is analyzed in terms of the roles (cf. Section 2.4). We construct the analysis along the components identified through the functional mapping for task `BlockPreviousCreditCard` as depicted in Figure 2. Specifically, a role might introduce a vulnerability or pose a threat for the component (cf. "negative influences" of humans on security as outlined in Section 2.4).

First of all, as we analyze a specific process task, the roles that are responsible for the entire BPM project, i.e., the *BPM group*, the *Business Engineers*, the *Chief process officer*, and the *Process owners* (cf. Section 2.4) are not considered because they are not directly involved in the business process as shown

Table 1: Human-oriented Security Analysis: Example Risks and Solutions

Role	B	S	I	V	T	Security risk	Solution
Process designers	☒			☒		misconfiguration of access constraints	consistency/compliance checks
	☒			☒		uncontrolled adaption of access constraints/business process	implement change control mechanisms
	☒			☒		uncontrolled administrative changes	implement change control mechanisms
	☒			☒		miscommunication with stakeholders	regular workshops
			☒	☒		errors in workflow design	correctness and compliance checks
Developers			☒	☒		misconfiguration at application level	follow security best practices and conduct security audit
			☒	☒		use default password	change password in configuration
			☒	☒		deviate from access constraints as specified at process level	conduct cross-layer consistency checks
			☒	☒		introduce vulnerability in inhouse developed application code	follow security best practices and life cycle
System architects		☒		☒		insufficient security controls at different parts of the system	defense in depth
		☒		☒		incorrect definition of trust boundary	conduct thorough security analysis and using reference security architecture
		☒		☒		insufficient protection against (rare) threats	defense in depth; additional security controls for high value assets
		☒		☒		introduce vulnerabilities in the system design enabling user to bypass BPM rules	conduct cross-layer consistency checks
Knowledge workers / Process participants			☒	☒		use weak password	security training and technology support for password management
			☒		☒	exploit logical flaws in the workflow	correctness and compliance checks, anomaly detection at BP level
			☒		☒	inject incorrect data into the process data context	data flow checks, implement integrity checks
			☒		☒	inject incorrect data into the database (via the PAIS)	data flow checks, consistent integrity checks between PAIS and database
Outsiders			☒		☒	launch denial of service attack	design and implement resilience measures such as time out or upper limit for number of instances
			☒		☒	social engineering and malware-based attack	security training, change management, and security controls

B: Business Process Level; S: System Design Level; I: Implementation Level;
V: Vulnerability, T: Threat

in Figure 2. Further on, task `BlockPreviousCreditCard` neither requires any interaction with *Subjects* nor with *Partners*. Hence, all these roles are not considered within the analysis.

At design time, mostly the technical staff such as system architects or developers are involved because they might introduce errors or make bad design decisions. For example, as software-intensive systems, PAIS often use and integrate off-the-shelf products. Due to complexity and knowledge gap, sometimes software and their security features are not correctly configured, thus becomes vulnerable to malicious attacks. For example, not changing the default authorization value of a SAP service might lead to unrestricted access for all authenticated users (see [30]). A tiny implementation mistake in the OpenSSL library (i.e. the Heartbleed bug) allows an attacker to read sensitive information in a server's memory including the server's private key and other users' passwords (cf. [8]).

At runtime, the actual users of PAIS and outsiders can introduce vulnerabilities as well as threats. For example, the access point of knowledge workers or process participants is typically the PAIS. From this point, they can access and execute business process activities. Knowledge workers have an extensive knowledge of the business processes and systems and might seek possibilities to circumvent security controls or exploit knowingly deactivated controls (cf. [12]). Reconsidering the Heartbleed bug (see [8]), also a hacker as an outsider can exploit the vulnerability and pose threats to the systems, e.g. by using stolen credentials to log into the system to compromise the whole security controls. Besides, insecure day-to-day operations by humans can lead to total failure of security protections. Social engineering, which targets humans in daily operations is proved to be quite effective to compromise many high secure systems [16].

Moreover, the attacks at the implementation can impact other layers such as the system design or the business process layer. For example, if the Heartbleed bug affected the interaction between the database and the workflow system in Example 1, the PAIS and the business process itself might have been compromised. On the other hand, bad design choices by the technical staff can introduce vulnerabilities on the implementation level that knowledge workers are not aware of. For example, if the system architects do not remove the vulnerability introduced by Heartbleed, the knowledge worker would continue working which can yield data breaches. While each single role has a different intersection with the PAIS system, the security implementations can be interrelated. A vulnerability or a threat at one point within one layer might have profound impact or cascading effect on other points at the business process, system design, and implementation layer. Therefore, the presented methodology can bring a big picture of the security issues, which helps us to trace the security interdependency in PAIS at design and runtime and to identify new security issues, in order to find proper solutions.

4 Discussion

This section examines the main findings, impact on research and practice, and the limitations of this paper.

4.1 Main Findings

This paper provides interdisciplinary research that bridges the gap between BPM/ PAIS and information security research. As security is a cross-layer concern, this paper investigates how security is embedded in PAIS architectures. It provides a holistic approach addressing security concerns at the business process, system design, and implementation layer in a PAIS. Moreover, security aspects at each layer are analyzed under consideration of human aspects. Using roles to represent human factors in security analysis gives us a fine-grained grasp of human involvement in various aspect of PAIS. As our analysis shows, it helps to transform the abstract notion of "human" into concrete and specific entities such that we can have more precise analysis related to the security implications of specific part of the PAIS architecture.

The research conducted in this paper shows that security issues are tightly interrelated and the impact of a security attack might be difficult to comprehend without a holistic view on a system. For example, an attack on the implementation to compromise a password can jeopardize the enforcement of constraints defined in the business process layer, or a design error by a system architecture might introduce a vulnerability that creates an opportunity for a disgruntled employee to circumvent the organizationally posed security controls and constraints. Particularly in complex settings with hundreds or thousands of process tasks, a systematic and holistic methodology helps to connect the dots and identify non-obvious security links.

Security in PAIS is a complex issue. A common approach is to create layers of abstraction, e.g. concentrating issues related to business process, or to software security. Our analysis shows it is also important to consider issues at the interfaces and beyond the abstraction layers.

Overall, the methodology proposed in this paper can be used for

- security analysis
- (security) compliance checks
- security trainings

These usage scenarios will be detailed in the next section.

4.2 Impact on Research and Practice

A main contribution of this paper is the methodology for a cross-layered and human-oriented security analysis. Researchers and practitioners may use this analysis as input for further security evaluations such as risk or threat analysis. For example, the human factor analysis derives a list of internal or external attackers that can be used as a list of “suspects” for the IT risk management (e.g., [37, 33]). The methodology can also serve as input for developing checklists for systematic (security) compliance checks. Moreover, this approach can serve as basis for several research directions; the methodology contributes not only to the security engineering of PAIS but also to research on human-centric PAIS (see [20]). For instance, the training of process participants can raise security awareness, can prevent misuse or errors, and can increase the knowledge of workers in order to fulfill their job functions successfully.

Based on the methodology, security-awareness trainings for different target groups such as business partners, developers, or managers can be developed. Furthermore, the result of the analysis can be used for the design and development of security controls. In addition, practitioners may use this review for revisiting security measures in PAIS, e.g., by reevaluating existing security measures with regard to human factors.

4.3 Limitations

This paper incorporates roles from the BPM and the information security area. However, we did not include roles from the risk management disciplines (e.g., [37, 33]) that focus on risk analysis such as employees/ users or chief security officers. One reason is that the classification in this paper centered only on roles from the business process community. It should be noted that some roles from risk management overlap with the roles outlined in Section 2.4.

The methodology is evaluated with an application scenario. The scenario should provide an initial guidance how to apply the methodology. However, the applicability of the methodology is not restricted by any means: For example, the number of tasks used in the function mapping and technical analysis, the level of detail for the human factor analysis, or the number of vulnerabilities and threats assessed can be adapted accordingly to the scenario.

5 Related Work

Humans are an important factor when it comes to information security. Research investigates human users from many perspectives such as trusting, decepting, and hacking humans (cf. [9, 26, 31]). For example, ways to socially engineer and deceive users in order to acquire information are shown in [26]. Furthermore, humans might circumvent security measures (see [22, 2]). Other publications such as [35, 23] analyze the outcomes of humans errors and use it to provide usable and effective security measures. Typically, dependencies can be used in information security to identify and analyze relations such as between assets and attackers ([36]) or between objectives and patterns ([18]). In this paper, we analyze dependencies that go beyond the business process in a multilayer model.

Several approaches tackle risk analysis for business processes. In [1], risk patterns for business processes are specified based on BPMN. These patterns partly incorporate the human aspect. The security-risk-oriented pattern 4 (SRP 4), for example, explicitly models the role of an attacker which accesses data in a business process in an unauthorized manner. [1], however, is confined to a set of five patterns that foresee possible attackers as explicit roles. By contrast, the approach at hand provides a systematic view on all layers of the system where potential attackers are not known beforehand. The approach presented by [39] analyzes a real-world example from the insurance domain with particular focus on the data flow. Hence, it investigates the connections between the tasks rather than the tasks. Such cross-task analyses, in general, can be applied complementary to the function and human-oriented analysis presented in this paper. Overall, we found that approaches for security in business processes often concentrate on one aspect such as data flow or security requirements.

A multilayer architecture in [44] contains business, process, integration, software, and technology architecture layers. Furthermore, the enterprise architecture is a cross-layer that represents aggregate artifacts and their relationships across all layers. In this paper, security is also seen as a cross-layer issue. Furthermore, the secure enterprise business model in [29] is based on a modern enterprise system architecture and contains three layers: First, the *business layer* specifies business processes and related organizational aspects. Secondly, the *application layer* specifies the services and data schemas for business process execution. Lastly, the *infrastructure layer* provides the software and hardware needed to automate the process execution (e.g., PAIS, databases, or operating systems). Furthermore, each layer can be split according to timing: design time, run time, and audit time. I.e. enterprise security and its measures is scattered across all layers and required at all times (design, run, and audit time). While the architecture in [29] centers on the technology and the execution time, our paper provides a methodology that combines technical and human aspects. We do not consider audit time in this paper since the respective analysis examines the security impacts on the systems during design- and runtime. To our best knowledge we are not aware of any other literature that investigates the human factor in business process security.

Furthermore, a case study on modeling secure transactions is shown in [14]. In particular, the authors use Secure Tropos (see [27]) to model dependencies between merchants that offer services, banks that requesting the service and cardholders that own the data. In this paper, we analyze security based on a payment card industry use case. However, we do not define specific requirements nor provide a methodology to define security-related artifacts. In fact, we use this as example to display that security is a issue across all layers. In addition, we analyze the dependencies between the levels in a multilayer architecture model. Furthermore in [25], the attentive listener (ATLIST) vulnerability analysis is provided. This method was mainly developed for service-oriented architectures that use a plethora of web technologies and SOA-specific standards and targets SOA-based business processes. Our methodology can be used to determine prerequisite artifacts for the ATLIST analysis.

6 Conclusion

Security is as good as its weakest link – the humans – in everyday business processes. So far, research has neglected to analyze the human factor when it comes to vulnerabilities and threats in PAIS. This paper proposes a cross-layer and human-oriented methodology for PAIS. The methodology examines business functionality beyond the business process layer and investigates technical and human aspects that are essential for a thorough examination. With an application scenario, we show that even the security analysis for a single business function results into an assessment of a multilateral interaction between multiple application components. Overall, the proposed methodology contributes to the security engineering of PAIS which can be used as a basis to conduct further security evaluations, systematic compliance checking, and target-group oriented security trainings. As a next step, the application of the analysis will be applied to selected real-world business process scenarios.

References

[1] Naved Ahmed and Raimundas Matulevičius. Securing business processes using security risk-oriented patterns. *Computer Standards & Interfaces*, 36(4):723–733, 2014.

[2] Steffen Bartsch and M. Angela Sasse. How users bypass access control - and why: The impact of authorization problems on individuals and the organization. In *ECIS 2013 Completed Research*, 2013.

[3] Frank Böhr, Linh Thao Ly, and Günter Müller. Business process security analysis–design time, run time, audit time. *it–Information Technology it–Information Technology*, 55(6):217–224, 2013.

[4] Reinhardt A. Botha and Jan H. P. Eloff. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal*, 40(3):666–682, 2001.

[5] Andy Briney. Security focused. *Information Security*, pages 40–68, 2000.

[6] B. Carminati, E. Ferrari, and P.C. K. Hung. Web service composition: A security perspective. In *Workshop on Challenges in Web Information Retrieval and Integration*, pages 248–253. IEEE, 2005.

[7] Timothy Casey, Patrick Koeberl, and Claire Vishik. Threat agents: A necessary component of threat analysis. In *Workshop on Cyber Security and Information Intelligence Research*, pages 56:1–56:4. ACM, 2010.

[8] Codenomicon Ltd. The heartbleed bug. Accessed: 2015-05-04.

[9] Carl Colwill. Human factors in information security: The insider threat – who can you trust these days? *Information Security Technical Report*, 14(4):186–196, 2009.

[10] Gurpreet Dhillon and James Backhouse. Current directions in is security research: towards socio-organizational perspectives. *Information Systems Journal*, 11(2):127–153, 2001.

[11] M Dumas, M. La Rosa, J. Mendling, and H. A. Reijers. *Fundamentals of Business Process Management*. Springer, 2013.

[12] Jeremy Epstein. Security lessons learned from société générale. *IEEE Security Privacy*, 6(3):80–82, 2008.

[13] L. Fuchs, G. Pernul, and R. Sandhu. Roles in information security - a survey and classification of the research area. *Computers & Security*, 30(8):748–769, 2011.

[14] Paolo Giorgini, Fabio Massacci, and John Mylopoulos. Requirement engineering meets security: A case study on modelling secure electronic transactions by VISA and mastercard. In *Conceptual Modeling*, pages 263–276. Springer, 2003.

[15] Sarah Granger. Social engineering fundamentals, part i: hacker tactics, 2001. Accessed: 2014-04-07.

[16] Christopher Hadnagy. *Social Engineering: The Art of Human Hacking*. Wiley, 2010.

[17] M. Hafner, M. Memon, and R. Breu. SeAAS-A reference architecture for security services in SOA. *Journal of Universal Computer Science*, 15(15):2916–2936, 2009.

[18] T. Heyman, R. Scandariato, C. Huygens, and W. Joosen. Using security patterns to combine security metrics. In *Availability, Reliability and Security*, pages 1156–1163, 2008.

[19] Meiko Jensen, Nils Gruschka, Ralph Herkenhöner, and Norbert Luttenberger. SOA and web services: New technologies, new standards - new attacks. In *European Conference on Web Services*, pages 35–44, 2007.

[20] Sonja Kabicher-Fuchs, Stefanie Rinderle-Ma, Jan Recker, Marta Indulska, Francois Charoy, Rob Christiaanse, Reinhold Dunkl, Gregor Grambow, Jens Kolb, Henrik Leopold, and Jan Mendling. Human-centric process-aware information systems (HC-PAIS). Technical report, arxiv.org, 2012.

[21] N. Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto. Web services choreography description language version 1.0. Technical report, 2005. http://www.w3.org/TR/ws-cdl-10/.

[22] I Kirlappos, S Parkin, and M. Angela Sasse. Learning from "shadow security": Why understanding non-compliance provides the basis for effective security. In *Workshop on Usable Security*, 2014.

[23] Sara Kraemer, Pascale Carayon, and John Clem. Human and organizational factors in computer and information security: Pathways to vulnerabilities. *Computers & Security*, 28(7):509–520, 2009.

[24] Maria Leitner and Stefanie Rinderle-Ma. A systematic review on security in process-aware information systems – constitution, challenges, and future directions. *Information and Software Technology*, 56(3):273–293, 2014.

[25] Lutz Lowis and Rafael Accorsi. Vulnerability analysis in SOA-Based business processes. *IEEE Transactions on Services Computing*, 4(3):230–242, 2011.

[26] Kevin D. Mitnick and William L. Simon. *The Art of Deception: Controlling the Human Element of Security.* John Wiley & Sons, September 2001.

[27] Haralambos Mouratidis and Paolo Giorgini. Secure tropos: A security-oriented extension of the tropos methodology. *International Journal of Software Engineering and Knowledge Engineering*, 17(02):285–309, 2007.

[28] Haralambos Mouratidis and Paolo Giorgini. Security attack testing (SAT)—testing the security of information systems at design time. *Information Systems*, 32(8):1166–1183, 2007.

[29] Günter Müller and Rafael Accorsi. Why are business processes not secure? In *Number Theory and Cryptography.* Springer, 2013.

[30] Mariano Nuñez Di Croce. Attacks to SAP web applications. BlackHat DC 2011 Briefings, 2011.

[31] Peter O. Okenyi and Thomas J. Owens. On the anatomy of human hacking. *Information Systems Security*, 16(6):302–314, 2007.

[32] OMG. Business process model and notation (BPMN) version 2.0. OMG Document formal/2011-01-03, Object Management Group, January 2011.

[33] Thomas R. Peltier. *Information Security Risk Analysis, Second Edition.* CRC Press, April 2005.

[34] Alfonso Rodriguez, Eduardo Fernandez-Medina, and Mario Piattini. A BPMN extension for the modeling of security requirements in business processes. *IEICE TRANSACTIONS on Information and Systems*, E90-D(4):745–752, April 2007.

[35] M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' - a Human/Computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–131, 2001.

[36] Reginald E. Sawilla and Xinming Ou. Identifying critical attack assets in dependency attack graphs. In *Computer Security*, pages 18–34. Springer, 2008.

[37] Gary Stoneburner, Alice Goguen, and Alexis Feringa. Risk management guide for information technology systems. NIST special publication 800-30, National Institute of Standards and Technology (NIST), July 2002.

[38] Frank Swiderski and Window Snyder. *Threat modeling*. Microsoft Press, 2004.

[39] Stefan Taubenberger, Jan Jürjens, Yijun Yu, and Bashar Nuseibeh. Resolving vulnerability identification errors using security requirements on business process models. *Information Management & Computer Security*, 21(3):202–223, 2013.

[40] Verizon. 2014 data breach investigations report. Technical report, 2014.

[41] A. Vorobiev and Jun Han. Security attack ontology for web services. In *Semantics, Knowledge and Grid, 2006*, pages 42–48, 2006.

[42] Richard Walton and Walton-Mackenzie Limited. Balancing the insider and outsider threat. *Computer Fraud & Security*, (11):8–11, 2006.

[43] M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.

[44] R. Winter and R. Fischer. Essential layers, artifacts, and dependencies of enterprise architecture. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW)*, pages 30–30. IEEE, 2006.

[45] C. Wolter, M. Menzel, and C. Meinel. Modelling security goals in business processes. In *Modellierung*, pages 197–212. GI, 2008.