Systematic Risk Assessment of Cloud Computing Systems using a Combined Model-based Approach

Nazila Gol Mohammadi[1], Ludger Goeke[1], Maritta Heisel[1] and Mike Surridge[2]
[1]*Working Group Software Engineering, University of Duisburg-Essen, Oststr. 99, Duisburg, Germany*
[2]*IT Innovation, Southampton, U.K.*

Keywords: Context Analysis, Risk Assessment, Threat and Control Identification, Cloud Computing Systems.

Abstract: Data protection and a proper risk assessment are success factors for providing high-quality cloud computing systems. Currently, the identification of the relevant context and possible threats and controls requires high expertise in the security engineering domain. However, consideration of experts' opinions during the development life-cycle often lacks a systematic approach. This may result in overlooking of relevant assets or missing relevant domain knowledge, etc. Our aim is to bring context analysis and risk assessment together in a systematic way. In this paper, we propose a systematic, tool-assisted, and model-based methodology to scope the context and risk assessment for a specific cloud system. Our methodology consists of two parts: First, we enhance the initial context analysis necessary for defining the scope for risk assessment, and second we identify relevant threats and controls during design- and deployment-time. Using the context model, and design-time system model, we further refine the gathered information into a deployment model. All steps of our methodology are tool supported and in a semi-automatic manner.

1 INTRODUCTION

A proper information security risk management is a key success factor for providing secure cloud computing systems. Recently, a significant number of security incidents have been reported. Such incidents can lead to strong consequences for cloud service providers not only financially, but also in terms of reputation loss.

The management of information security risks requires activities to identify and control information security risks for the assets of an organization. These assets include, among others, processes and information/data that are essential for providing the business of an organization.

The establishment of a context for the information security risk management is an important activity as the context defines which assets are relevant for the risk management. Because of the complex architecture of cloud computing systems, the number of participating stakeholders and relevant regulations, the context establishment is a difficult task that carries the danger that critical information is overlooked.

Subsequent to context creation, risk assessment is a crucial task. The risk assessment identifies the risks that assets may be compromised due to an information security incident. To this end, the threats to the assets have to be identified. The large number of threats for different types of assets and the rapidly changing threat landscape make the threat identification a difficult task, and some threats can easily be overlooked. Moreover, dependencies between assets give rise to potential attack paths (using sequences of threats to individual assets), and also secondary effect cascades. Either can also be easily missed, leading to errors in the assessment of risks from the identified threats.

Finally, although standardized approaches require the context for risk management to be established first, in practice the use of third party software, services, and cloud resources means that the context for the risk analysis is often determined only just before a system is deployed in the cloud. Therefore, we aim to bring the context analysis and a risk assessment approach together to provide systematic guidance in identifying the relevant context information and performing threat and control identification as early as possible, i.e. during requirements engineering and design-time.

In this paper, we present a methodology that allows to systematically identify relevant context information and after that the identification of threats and

Mohammadi, N., Goeke, L., Heisel, M. and Surridge, M.
Systematic Risk Assessment of Cloud Computing Systems using a Combined Model-based Approach.
DOI: 10.5220/0009342700530066
In *Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) - Volume 2*, pages 53-66
ISBN: 978-989-758-423-7

controls. Our methodology follows a model-based approach. This ensures consistency and traceability, and provides unambiguous representations of risk factors and assumptions that can be used to facilitate communication between stakeholders. We provide tools supporting the use of each model, and these tools guide risk analysts through the different steps of our methodology and supports their execution, thus limiting the manual effort to apply the methodology. Furthermore, one of the tools provides a possibility to verify the model which helps to detect errors in the application of the methodology.

For the context establishment, we provide a graphical context pattern with an underlying context model that defines types of information that are relevant for the context of cloud computing systems and a tool supporting application of this context pattern. For the risk assessment itself, we provide a different graphical modelling approach that focuses on the composition of the system in terms of assets and their interdependencies. The tool in this case supports creation of the system model, and used with a knowledge base of security threats and controls, allows auto-generation of a threat catalogue for the modelled system. The tool then allows controls to be specified and risk levels automatically calculated. The knowledge base includes security threats, compliance threats (models of selected requirements under the GDPR), and also potential modelling errors which are treated as threats and highlight where aspects of the modelled system may be underspecified or inconsistent.

In our methodology the risk model is usually developed first, capturing the expected use of software and services from their suppliers. The context model is captured later by the operator of the system, encoding information about the intended deployment, which is then fed into the risk model which can then be used to complete the risk assessment procedure. The created models document the assumptions made by stakeholders at each step and facilitate communication. These models, with a suitable formatting, can provide documentation required for other purposes, e.g. for ISO 27001 audits, or to generate notices or policies on the use of personal data within the system.

The remainder of the paper is structured as follows: In Section 2, we briefly introduce the underlying concepts. Section 3 presents the contribution of this paper by outlining the objectives, the models used and our proposed methodology. Section 4 describes the application of the methodology with an example. We discuss related work in Section 5 and conclude our paper with an outlook on future research directions in Section 6.

2 BACKGROUND

In this section, we briefly introduce the fundamental concepts for our risk assessment methodology.

Data Protection Goals. Data protection goals are the following: *confidentiality* involves preventing unauthorized access to data, *integrity* preventing unauthorized or accidental alteration or corruption of data, *availability* ensuring authorized access to data is possible within an acceptable time, *unlinkability* separating privacy-relevant data from other data, *transparency* ensuring all involve parties understand the privacy-relevant data handling processes, and *intervenability* allowing involved parties to interfere with processing to make corrections and prevent inappropriate processing (Zwingelberg and Hansen, 2011).

The General Data Protection Regulation (European Union, 2016) obliges anyone collecting or processing data within the European Union to take responsibility for managing risks to Personally Identifiable Information (PII). Article 25 in particular stipulates that the data controller shall "both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organizational measures" to protect PII, "taking into account the state of the art, the cost of implementation and the nature, scope, context, and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing". In other words, data protection goals should be achieved by assessing risks during the design and operation of a system for processing PII, and introducing measures to manage those risks.

This poses considerable challenges for the reasons discussed above. A data controller should assess and manage risks to personal data, but they depend on suppliers of software and services, and in cloud based applications, even the management of resources used to implement the data processing system. It is difficult for a data controller to identify potential threats that should be managed, and ensure that the "technical and organizational measures" in place are sufficient to manage risks from those threats. This becomes more challenging when one considers that it is not enough to do this during the design of the system, but also during its operation. In principle, one must reassess risks whenever the system changes, or whenever new types of threats are discovered.

ISO 27001. The normative ISO 27001 (ISO/IEC 27001, 2017) standard specifies "the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of an organization". Since these requirements are defined in a generic way, the standard can be applied to organizations of any business area. Annex A lists information security control objectives and controls that should be considered and used where relevant in systems falling within the scope of the ISMS. ISO 27001 also requires the use of a risk assessment procedure to determine the appropriate level of risk treatment using security controls from Annex A. The scope of the ISMS should be defined by a Statement of Applicability.

A typical response to GDPR Article 25 by data controllers is to use ISO 27001 accredited suppliers of software and operators of services and cloud resources used in creating the personal data processing system. In doing this, the data controller must determine from the Statement of Applicability whether the software or services they are using are covered by an ISMS, and whether the risk management criteria and assumptions used in those ISMS are sufficient for the data controller's purposes. Complex interdependencies exist between these externally sourced assets, making it extremely difficult to do this in practice.

ISO 27005. The ISO 27005 (ISO/IEC 27005, 2018) standard provides informative guidelines for conducting an information security risk management in an ISO 27001 compliant ISMS. Here, the different guidelines give support for the application of an information security risk management in the different phases.

An iteration of the information risk management process starts with a context establishment, which includes definition of the scope and boundaries of the system in which risks are assessed, and selecting: i) the risk management approach and assessment if the necessary resources for the risk management are available, ii) risk evaluation criteria for the assessment of information security risks (e.g. value of assets for the business processes of an organization), iii) impact criteria for assessing the impact (e.g. monetary loss) if a security property of an asset gets compromised, and iv) risk acceptance criteria, with respect to a scale of risk levels.

After the context establishment, the risk assessment starts with the risk identification, which involves the following processes: 1) identification of the assets within the defined scope, which may be difficult if some assets are supplied or operated by third parties, 2) identification of threats for the identified assets, which is often easier for the suppliers or operators than for the data controller, 3) identification of existing controls that are already implemented, which is at least partly determined by the suppliers, especially for assets implemented by software, 4) identification of vulnerabilities regarding the set of identified assets, which depends on the suppliers or operators of the assets, and 5) identification of the damages and consequences to an organization if a security property of an asset gets compromised.

The next step is risk analysis, which involves comprehending the nature and level of risk posed by the identified threats. This is found by first estimating the likelihood of each threat, given the presence of controls and vulnerabilities, and the impact of the threat given the harm caused should its consequences occur. ISO 27005 defines two categories of assets: i) primary assets, i.e. information and business processes; ii) supporting assets on which the primary assets rely, including hardware, software, and non-technical elements such as system administrators, etc. For primary assets, the impact is based on the direct consequences to the organization (or in the case of PII, to the data subject). For secondary assets, the impact of any compromise depends also on the interdependencies between assets. This is usually difficult to determine, as it is easy to overlook dependencies especially when the assets are supplied or operated by third parties.

The last step is risk evaluation, in which the acceptance criteria are applied to determine whether risk levels are acceptable. If not, one must select a risk treatment which may include avoidance (not using the system or function in which the risk arises), transfer (assigning responsibility to another party, although this is not permitted under the GDPR), or risk reduction (introducing more security controls to reduce the likelihood and/or impact of the associated threat).

3 RISK ASSESSMENT AND EVALUATION METHODOLOGY

In this section, we describe our methodology, which considers risk assessments at different phases in the lifecycle of a cloud-based software/service. Our methodology provides an underlying conceptual model, a knowledge base for the provision of a risk assessment and guidelines for the different steps of a risk assessment. It also includes tasks for the evaluation of the implemented software/service and its deployment by performing penetration tests.

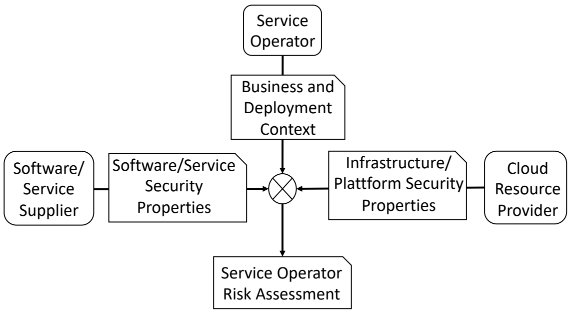

Service Operator
Business and Deployment Context
Software/ Service Supplier
Software/Service Security Properties
Infrastructure/ Plattform Security Properties
Cloud Resource Provider
Service Operator Risk Assessment

Figure 1: Organizational Dependencies in Risk Assessment.

3.1 Objectives of Our Methodology

Today, cloud-based application software and services are often designed and implemented by specialist suppliers, independently from their eventual operator(s). The former hold most of the knowledge required to analyze information security risks, and are in a position to introduce controls to address many of those risks. Yet they lack information about the deployment environment and so cannot perform a complete risk assessment, which ultimately must be done by the service operator. This leads to dependencies between different organizations in a risk assessment, as shown in Figure 1. This shows the involved stakeholders: the software/service supplier, the service operator who manages the deployed system, and the provider of the underlying cloud resources on which the system is deployed. The service operator is responsible for the final risk assessment, and needs information from the other two stakeholders. This is not addressed explicitly by ISO 27005 as it describes the procedure to be used within a single organization.

To solve these challenges, our methodology has the following objectives: 1) provision of a concept for a risk assessment that considers different lifecycle phases of a software/service; 2) provision of guidelines for the execution of the different risk assessment tasks; 3) provision of a knowledge base that provides necessary information for the execution of risk assessment tasks; and 4) provision of guidelines for the evaluation of the software/system by performing penetration tests.

Our approach depends on the creation and use of models to define the system in which risks are being assessed and to support identification and analysis of risks within that system. Models are used because they provide three significant benefits compared to other methods: 1) models provide an unambiguous description of the system, the assumptions made about potential sources of threats (e.g. attackers) and the presence of security controls, etc., 2) models allow automated threat identification and risk analysis, which is both repeatable and less likely to overlook important risks, and also faster and better suited to meet the GDPR obligation to frequently reassess risks during the operation of a personal data processing system, 3) the automated risk analysis can incorporate the analysis of attack paths and secondary effects, making it much easier to specify the impact of threat consequences, especially for threats to secondary assets.

3.2 Underlying Models and Patterns

Context Modelling. The context of a system represents a specific part of the system environment. This part is relevant for the definition of a system and understanding of its requirements (Pohl, 2010). A system context includes stakeholders, technical components, processes, events, and regulations (e.g. laws) that are relevant for the system (Pohl, 2010).

A context model is a general structural description of a particular type of system including its environment (cf. (Context-Patterns, 2018)). It enables the description and representation of a concrete system that represents an implementation of the system type as documented in a model (with a graphical representation).

System Modelling. Modelling of cloud systems is still in its infancy. Traditional modelling approaches are insufficient as they focus mainly on the technical aspects of a system, and system functionalities. Furthermore, they do not include information on the deployment, and relation between different types of technical parts of systems, e.g. hosts of applications. These models are designed to help system designers to graphically identify and analyze the threats that can arise in a system (Broy et al., 2012; Lock and Sommerville, 2010).

In our methodology, a system model represents a system in which risks are being assessed in terms of assets and dependencies. A system model is constructed from the asset classes defined in a knowledge base, also called a domain model since it describes generic asset types in a given domain, such as cloud services. To model cloud applications, the domain model includes technical assets (physical or virtual devices and software components such as web applications, databases, etc), logical assets that represent capabilities or dependencies arising from asset interactions, and social or physical assets including physical spaces and human stakeholders. The domain model also contains models of potential threats to those assets, and security controls that could be

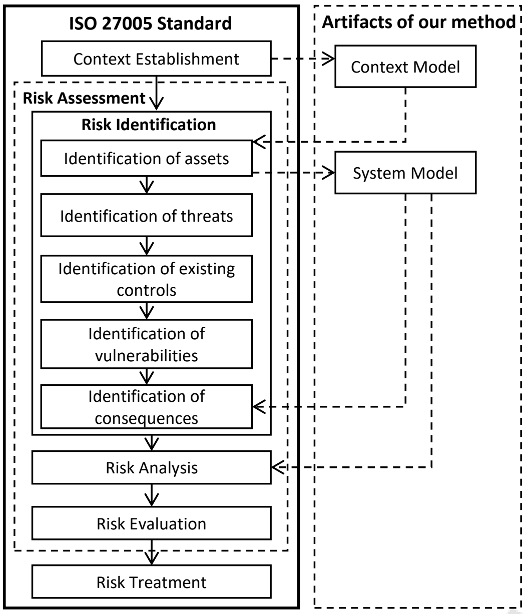


Figure 2: Our Approach and Relation to the Steps in ISO 27005 (ISO/IEC 27005, 2018).

used to protect the assets.

The design-time system model is normally created by the software/service developers, describing the expected arrangement of services and their relationships to an assumed cloud deployment environment. The deployment model is created by the service operator, based on the design-time model but characterizing the actual cloud deployment environment.

3.3 Our Methodology

Our methodology is based on ISO 27005, and covers the definition of the context, identification of assets, threats and existing security controls, and the consequences and impact of threats, and the subsequent analysis of risks from these threats. The models described above are used to support these processes as shown in Figure 2.

The types of assets to be considered in cloud applications, divided into primary and secondary assets as defined by ISO 27005, are shown in Figure 3. This diagram shows only the asset types specified (asserted) explicitly by a user, which include primary assets in the form of *Information* and *Business Processes*, and a wide range of secondary assets including *Location*, *Network*, *Stakeholder*, *Hardware*, and *Software*. Location assets are used to denote physical locations, e.g., a company's building, from which we can infer risk levels from physical attacks or attacks involving local access to devices. Network assets represent different means by which devices can communicate, and allow inference of threats from network attacks at OSI Layer 2 (which may depend on the type of network) or Layer 3, as well as attacks on specific types of asset interactions (e.g. attacks on web communications). Devices provide the means to store and process data, and allow modelling of threats against both the devices and the supported processing. Thus a Host asset can *host* a Process asset (representing one or more executing software components), or *store* data assets representing information.

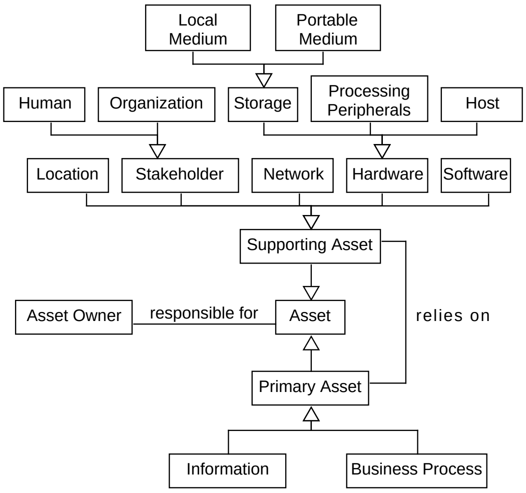


Figure 3: Conceptual Model for Asset Classes (Adapted from (Gol Mohammadi et al., 2019)).

Stakeholders can be used to represent both human and organizational stakeholders, and allow their relationships to asset to be captured. A human can *manage* devices or (implicitly) all devices at a location, *interactWith* processes which means the human has control over the process, and data assets can *relateTo* a human, indicating the data is Personally Identifiable Information and the human is the data subject. In the current version of the domain model, there are also specialized subclasses of information representing the various Special Categories of data defined by the GDPR Article 9, and subclasses of Human to represent adults and children, the latter being afforded extra protection under the GDPR Article 8.

Other assets including logical assets representing capabilities or dependencies within groups of assets are automatically inserted by the tooling, based on rules specified in the domain model itself.

The identification of threats, potential controls and threat consequences are all handled automatically by the tooling using knowledge encoded in the domain model. However, these steps are supported explicitly, and users still have to specify which of the potential controls are already present in the system design or the deployed system.

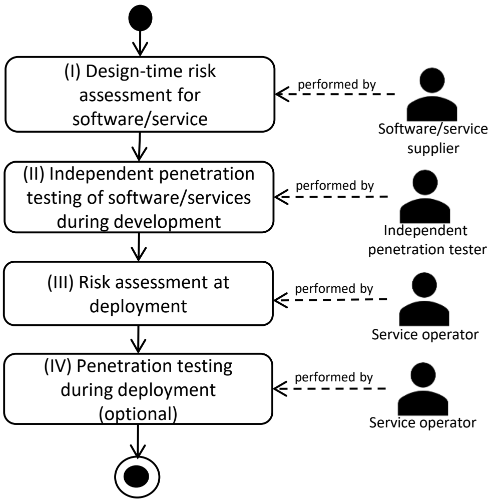

(I) Design-time risk assessment for software/service
performed by
Software/service supplier
(II) Independent penetration testing of software/services during development
performed by
Independent penetration tester
(III) Risk assessment at deployment
performed by
Service operator
(IV) Penetration testing during deployment (optional)
performed by
Service operator

Figure 4: Overall Approach of Our Methodology.

The domain model captures vulnerabilities as either intrinsic (purely internal) or extrinsic (externally exploitable) flaws in asset character or implementation (e.g. the potential presence of software vulnerabilities), or the absence of controls (e.g. failure to authenticate users of an asset). Because of this, identification of vulnerabilities is handled invisibly, and is not supported explicitly by the tooling. The user only has to define their level of trust that assets are free of intrinsic or extrinsic flaws.

3.3.1 Overview of Our Methodology

The overall procedure for using the methodology covering both the design time and deployment risk assessments is shown in Figure 4.

This shows our overall methodology with four main tasks that are performed during different phases. It starts with a risk assessment during the design/development of a cloud-based software/service that is performed by personnel of the software/service supplier ((I) in Figure 4). After the development of the software/service, it is tested by independent penetration testers to verify the results from the risk analysis ((II) in Figure 4). When the software/service is supplied to a certain organization for its deployment, a further risk analysis is performed that considers characteristics specific to the deployment environment ((III) in Figure 4). This risk analysis is executed by the service operator. If the service operator considers it as necessary and if it is feasible, further penetration tests should be performed on the software/service within its actual deployment environment.

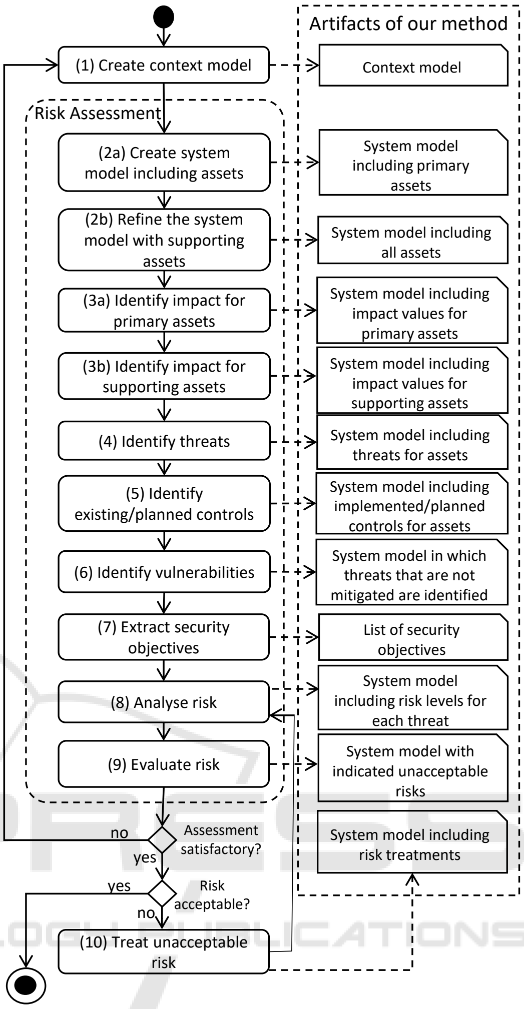

Artifacts of our method
(1) Create context model
Context model
Risk Assessment
(2a) Create system model including assets
System model including primary assets
(2b) Refine the system model with supporting assets
System model including all assets
(3a) Identify impact for primary assets
System model including impact values for primary assets
(3b) Identify impact for supporting assets
System model including impact values for supporting assets
(4) Identify threats
System model including threats for assets
(5) Identify existing/planned controls
System model including implemented/planned controls for assets
(6) Identify vulnerabilities
System model in which threats that are not mitigated are identified
(7) Extract security objectives
List of security objectives
(8) Analyse risk
System model including risk levels for each threat
(9) Evaluate risk
System model with indicated unacceptable risks
no
Assessment satisfactory?
yes
System model including risk treatments
yes
Risk acceptable?
no
(10) Treat unacceptable risk

Figure 5: Overview General Risk Assessment.

General Risk Analysis of Our Methodology. The risk assessment steps (I) and (III) in Figure 4 are described in more detail in Figure 5, which shows the steps to be performed and the input and/or generated output of each step. In this section we focus on the general procedure, without considering if the risk assessment is performed during the design time or deployment. Specific variants used in each of the lifecycle phases are described later in Section 3.3.2.

The steps are as follows:

(1) Create Context Model. Before the actual risk assessment, the scope and boundaries for the risk analysis have to be specified. The scope represents the starting point for the identification of the assets of an organization that are relevant for the risk assessment. For establishing the context, the characteristics regarding the business of an organization are considered. These characteristics are represented by the following information: 1) relevant legal, regulatory and contractual requirements; 2) relevant internal stakeholders that are interacting with the provided service(s); 3) relevant locations; 4) high-level assets that represent business processes of an organization and information that is related to the business processes; 5) interfaces (cf. (ISO/IEC 27005, 2018), Section 7.3, p. 7f).

In our methodology the scope is specified by creating a context model (see Figure 5, step 1). To support this step, we provide a context pattern for cloud computing scenarios (see Section 3.2). This pattern can be instantiated for the considered cloud computing scenario. Our context pattern specifies the relevant types of context information for cloud computing scenario. During its instantiation, the pattern elements of appropriate types are instantiated. In this connection, the main business processes are represented in the context pattern by instantiated cloud computing services of the levels Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and/or Software as a Service (SaaS). The context pattern instance also includes the processed data. Furthermore, relevant legislations and regulations in the form of appropriate indirect stakeholders, direct stakeholders of the considered cloud computing service and any relevant locations are identified within the pattern. Strictly speaking, the data contained in a context pattern instance is an asset that is normally identified during the identification of assets. Our methodology identifies the data already within the context establishment because the types of processed data are necessary for deriving relevant regulations.

The output of this step is a context model in the form of an instance of our context pattern.

(2a) Create System Model Including Primary Assets. Based on the scope that is defined by the instantiated context pattern in step 1, the assets that are considered within the risk assessment (see Figure 5, steps 2a and 2b) are identified. Regarding assets, the ISO 27005 distinguishes between primary assets and supporting assets. Primary assets comprise the business processes that are contained in the scope as well as the processed information and data. Supporting assets are those elements on that the primary assets rely (e.g. software, hardware, network). The current step (see Figure 5, step 2a) relates to the identification of primary assets by creating a system model (see Section 3.2). Accordingly, processes and the processed data are modelled in this system model. The modelled processes are less abstract than the business processes in the context pattern instance from step 1. The system model defines different types of processes that represent running applications that may process, store or exchange data. Therefore, a process in the system model contains always a supporting asset in the form of a software. This software has to be taken into account when a process is considered during the risk assessment. Relations between different processes are also modelled in a system model. Regarding the data, the system model distinguishes between data with normal protection needs and sensitive data that requires high protection needs. The data shall accord to the data in the context pattern instance. Jurisdictions for the data processing are also modelled in a system model. These jurisdictions shall correspond to identified legislations in the context pattern instance.

The output of this step is a system model that contains primary assets and jurisdictions for the data processing.

(2b) Refine the System Model with Supporting Assets. This step (see Figure 5, step 2b) identifies the supporting assets for the primary assets from step 2a (see Figure 5, step 2a). The following types of supporting assets specified in ISO 27005 are considered: 1) software; 2) hardware; 3) networks and their components; 4) personnel; and 5) sites (cf. (ISO/IEC 27005, 2018), Annex B.1.1, p. 28).

Our methodology extends theses types of supporting assets by *public spaces* and *private spaces*. These spaces represent physical locations where devices may be located, and subnets (notably radio subnets) may be accessed. There is a specialized private space called a *DataCentre* which represents a data center containing network assets and devices to avoid having to include these explicitly in the system model. Virtual hosts and virtual networks can be provisioned at a data center, without worrying about the mapping to physical hardware.

The identified supporting assets are included into the system model that has been created during step 2a. Within the system model, hardware, networks, and network components are represented by different types of *network assets*. For a network asset associations to primary assets in the form of hosted processes and processed, stored, and/or exchanged data are identified. Associations that represent connections to or interactions with other network assets are also modelled, and are used to determine attack paths and secondary effect cascades during the later identification of threats for the identified assets in step 4 (see Figure 5, step 4). Associations with human assets (representing user roles) are used to model threats involving malicious users (insider attacks), and threats from user error, such as phishing attacks.

Relationships with relevant jurisdictions are also taken into account, primarily to allow the detection of cross-border data transfers and potential compliance threats, e.g. where nationally regulated data (as defined in the GDPR Article 9.4) moves across a border.

The output of this step is the system model that includes all primary and supporting assets.

(3a) Identify Impact for Primary Assets. The impact measures the consequences if an asset gets compromised by a threat. In this step (see Figure 5, step 3a) the impact for primary assets is determined. The knowledge base within our methodology contains predefined default values for the impact level of each type of compromise and asset type. These default levels are mostly quite low except for sensitive data classes, because in practice this is appropriate for secondary assets, and they can be of any type apart from data.

(3b) Identify Impact for Supporting Assets. This step (see Figure 5, step 3b) determines the impact values for supporting assets. However, this step is usually not required, because our model-based approach makes it possible to automatically detect and analyze attack paths and secondary effect cascades. Since the main business value of a secondary asset is to support primary assets, it is often sufficient to assume there is no direct impact from a compromise in a secondary asset other than the propagation of consequences to primary assets.

The output of step (3) is the system model including the specification of impact for (mainly primary) asset compromises.

(4) Identify Threats. In this step, possible threats to the assets in the system model are identified (see Figure 5 step 4). To this, a catalogue of threats that refer to the different types of assets is provided. This catalogue is generated using the threat identification rules incorporated into the knowledge base. If new types of attacks are discovered, which are not merely the old attacks exploiting some new programming error, the knowledge base must be updated.

The output of this step is the system model including a threat catalogue describing threats to each system asset.

(5) Identify Existing/Planned Controls. Controls are measures that have been or are planned to be realized for assets to decrease the likelihood that an asset gets compromised by certain threats. Thus, the identification of existing/planned controls is a part of the risk assessment (see Figure 5, step 5). Our methodology supplies a catalogue that contains controls for the different types of assets. This catalogue is generated based on the knowledge base. During the control identification, the existing and/or planned controls can be selected from the set of controls that are associated to a certain type of asset. Within our methodology, controls that could address a particular threat are combined as so called *control strategies*. When all the controls needed for a control strategy are selected, the effect is to reduce the likelihood of a threat causing the expected consequences. Each control strategy has a strength parameter which determines the maximum residual threat likelihood when the control strategy is selected. If new security measures are developed to protect certain types of assets, these should be added to the knowledge base.

The output of this step is the system model in which the appropriate implemented and/or planned controls are specified at each system asset.

(6) Identify Vulnerabilities. As discussed above, in our methodology this step is implicit, so no user action is needed. A vulnerability associated with flaws in assets is modelled in terms of asset trustworthiness, while the lack of security measures is captured by the presence (or in this case lack) of the relevant controls.

(7) Extract Security Objectives. This step considers the extraction of security objectives (see Figure 5, step 7) for the considered software/service. It is not adapted from the ISO 27005, but is included because security objectives can be used as an input to any processes used to verify security properties, including penetration tests that may be applied in steps (II) and (IV) in Figure 4. They can also be used to formulate a designated level of security for a software/service to third parties (e.g. cloud service providers). The security objectives are derived from the impact factors specified in step (3), and the threats that potentially cause high impact asset compromises as determined from step (5).

The output of this step is a list of security objectives.

(8) Analyze Risk. In the risk analysis (see Figure 5, step 8), for any identified threat regarding each asset the risk level is assessed. A risk level is a quantitative value that represents the risk that an asset gets compromised by the considered threat. It is assessed based on values that have been identified during the previous steps of the risk identification (see Figure 5). Within our methodology, risk levels are determined for any threat. A risk level is determined based on the likelihood that a threat occurs and has the expected consequences, in the presence of selected controls, and the impact of these consequences taking account of attack paths and secondary effects.

The output of this step is the system model in which risk levels are assigned to the threats to each asset.

(9) Evaluate Risk. During the risk evaluation (see Figure 5, step 9), it is evaluated if the determined risk levels from the risk analysis are acceptable referring to the defined risk criteria. In our current domain model, risk levels are expressed using a 5-point scale from *Very Low* to *Very High*, and we suggest that on this scale a *Medium* level risk might be acceptable for a short time, but normally the worst case risk level from any threat should be *Low* or *Very Low*. However, this depends on the risk appetite of the user, who must make a decision after step 9, as shown in Figure 5. If any the level of any risk is too high, it must be treated in a further step.

The output of the risk evaluation is the system model in which unacceptable risk levels are indicated.

After the risk evaluation, it has to be evaluated if the previous results of the risk assessment are meaningful (see Figure 5). If this is not the case, the scope of the risk assessment has to be adjusted and the steps of the risk assessment have to be repeated. If the results are meaningful, the approach continues with the treatment of unacceptable risks. Our domain model and tooling provides some assistance with this, as potential system modelling errors are also modelled as threats, so if there are any *modelling error* threats in the generated catalogue, it may mean assets or relationships have been overlooked, and the model may need to be revised and the process from step (2) repeated.

(10) Treat Unacceptable Risk. The risk treatment (see Figure 5, step 10) includes all risk levels, which have been determined during the risk evaluation, that are not acceptable. In our methodology (see Figure 5 step 10), additional controls for the corresponding asset can be selected with the objective to reduce the risk level. At this, appropriate controls that are provided by the knowledge base are selected or new controls are added to the knowledge base and are selected afterwards. After the selection of controls, the steps for the risk analysis and risk evaluation for the concerned assets are repeated to examine if risk treatment has been resulted in acceptable risks. If this is not the case for certain assets, these assets must undergo a further risk treatment and so on.

It is also possible to use other risk treatments, such as avoiding risk by not providing certain features and corresponding assets by the software/service any more. The concerned assets are removed from the system model, or (if the removal is only temporary) marked as disabled (this is included in the domain model as a potential security control). Outsourcing the risk is also possible.

The output of this step is an updated system model with the selected risk treatments.

3.3.2 Variants

(I) Risk Analysis at Design-Time/Development. At design time, the above procedure is used by the developer of the software/service, ideally during its design but certainly before offering it for use in a specific application (see Figure 4 step I). The following discussion just considers specific differences compared to the general risk assessment (see Figure 5).

The context model created during the context specification contains all context information that is relevant for the software/service at the design time and has to be considered during the development of the software/service. Additionally, context information can define requirements to a potential deployment environment that have to be fulfilled by a corresponding cloud infrastructure (e.g. computing center shall be located in the European Union because the General Data Protection Regulation shall apply).

Within the identification of assets (see Figure 5 steps 2a and 2b), the primary assets represent the processes of the software/service that host parts of the software or applications of the service for processing, storing, and/or transferring data. The affected data itself is also considered as primary asset. The supporting assets include components that are necessary for the provision of the software/service. The software/service developer may not know the precise specification of some of these assets, but can encode their assumptions by their choices at this stage. For example, they may know that the service will be deployed at a data center, and assume that processes will run and data will be stored on virtual hosts. In other cases, the model should be as general as possible, e.g. if it is possible for one person to fulfil two roles, the roles should not be combined because in general, two different people may be involved. Other aspects may be omitted altogether, and left for the service operator to define, e.g. the jurisdictions in which services will be deployed, and from where they may be accessed.

Identified controls (see Figure 5 step 5) for the primary assets and supporting assets that are part of the software/service shall be implemented during its development. The controls for the other supporting assets represent planned controls that shall be implemented by a potential deployment environment. The software/service developer can only implement controls that are embedded in their components, but may at this stage specify their assumptions or requirements on which other controls shall be used.

The same applies to security objectives. Security objectives for primary assets and secondary assets as a part of the software/service shall be implemented during the development. The security objectives for the other supporting assets are planned and shall be implemented by a potential deployment environment.

(II) Independent Penetration Testing of Software/Services. This task (see Figure 4 step II) includes the specification and execution of penetration tests regarding the software/service. These tests are performed on exposed software/service endpoints and interfaces to confirm that:

- the strength of security controls is sufficient, e.g. encryption algorithms and key lengths
- potential vulnerabilities cannot be exploited, e.g. using injection attacks, cross-site scripting attacks, etc.

Test scenarios may be based on the risk model from the design time risk analysis (see Figure 4 step I). For example, if a software asset was specified as highly trustworthy, one can perform a *what if* analysis by reducing the trustworthiness level of this asset, and seeing which types of threats then lead to high risk levels. The tests should then check for flaws that could enable those types of threats to be carried out.

(III) Risk Assessment at Deployment. The risk assessment in this task considers the deployment of the software/service in an actual cloud environment (see Figure 4 step III). This risk assessment is performed based on the context model and system model from the risk assessment at design time (see Figure 4 step I). In the following, only specific differences compared to the general risk assessment (see Figure 5) are considered.

The context model from the context definition (see Figure 5 step 1) is expanded by context information that is specific to a certain deployment environment (e.g. cloud provider, location of computing center, service level agreements).

In the system model, further supporting assets (see Figure 5 step 2b) that are specific to the deployment environment are identified (e.g. components that are used for virtualization) and have not been known at design time. Additionally, supporting assets that have been represented during design time in an abstract way are concretized (e.g. the concrete type of a virtual machine or router is specified). Because all primary assets have been identified during the risk assessment at design time, there is no need for a further identification during deployment.

For the new added supporting assets impacts are assessed (see Figure 5 step 3b), threats and counteracting existing controls are identified (see Figure 5 steps 4 and 5), vulnerabilities are identified (see Figure 5 step 6), security objects are extracted (see Figure 5 step 7), and a risk analysis (see Figure 5 step 8), a risk evaluation (see Figure 5 step 9) and if necessary a risk treatment are performed (5 step 10).

Regarding concretized supporting assets it has to be checked if further threats have to be considered (see Figure 5 steps 4). If this is the case, for these threats existing mitigating controls have to be identified (see Figure 5 steps 5) and the remaining steps for the risk assessment (see Figure 5 steps 6 - 9)) and if necessary a risk treatment (see Figure 5 step 10) have to be performed.

With regard to planned controls for supporting assets, that have been identified during design time, it has to be checked if these controls are actually implemented by the deployment environment in a sufficient way (see Figure 5 step 5).

(IV) Penetration Testing during Deployment. In some cases, the service operator may wish to conduct additional penetration tests to verify security in the specific operational deployment environment. This may be necessary where the software/service supplier cannot arrange penetration testing in a sufficiently realistic environment. This may arise if:

- the service operator wishes to deploy software/services in an environment that is unlike that envisaged by the software/service supplier; or
- the deployment environment is only accessible to the service operator, e.g. because it uses a hybrid cloud composed partly of in-house resources.

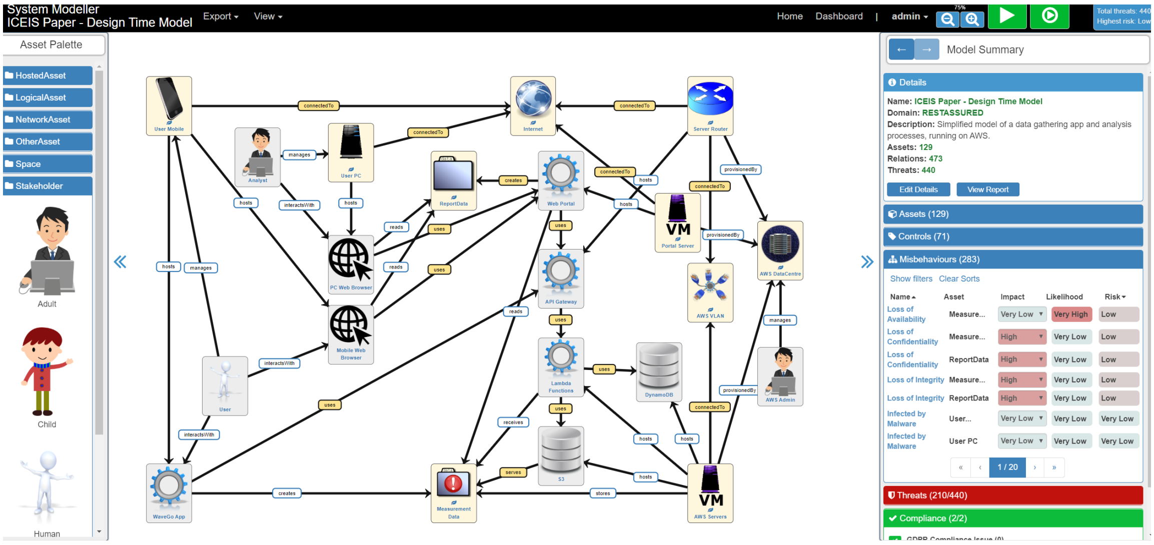
In these cases, if the service operator needs to verify security properties, they should carry out their own tests, where permitted by the cloud (or other) resource providers.

Any test scenarios are based on the results from risk assessment at deployment (Figure 4 step III).

4 APPLICATION EXAMPLE

The methodology was tested with a commercial software and service supplier, one of the partners in the collaborative project in which it was developed. The company involved is an SME, specializing in the provision of applications in health and social care mainly for local government organizations. They provide software in two ways: by deploying it in the cloud and offering it as software as a service (SaaS), or by supplying it to service operators who deploy it in their preferred environment.

To avoid any potential ethical issues, the validation exercise reported here is based on a generic data collection and analysis application, in which a user collects data using an app running on their smart phone, and sends it to be stored and analyzed by services running in the cloud. Results of the analysis are made available via a web portal. The SME often uses this design, and depending on the application it may be used to capture personal data related to the phone



Model Summary

Details

Name: ICEIS Paper - Design Time Model
Domain: RESTASSURED
Description: Simplified model of a data gathering app and analysis processes, running on AWS.
Assets: 129
Relations: 473
Threats: 440

Edit Details View Report

Assets (129)

Controls (71)

Misbehaviours (283)

Show filters Clear Sorts

Name	Asset	Impact	Likelihood	Risk
Loss of Availability	Measure...	Very Low	Very High	Low
Loss of Confidentiality	Measure...	High	Very Low	Low
Loss of Confidentiality	ReportData	High	Very Low	Low
Loss of Integrity	Measure...	High	Very Low	Low
Loss of Integrity	ReportData	High	Very Low	Low
Infected by Malware	User...	Very Low	Very Low	Very Low
Infected by Malware	User PC	Very Low	Very Low	Very Low

« ‹ 1 / 20 › »

Threats (210/440)

Compliance (2/2)

GDPR Compliance Issue (0)

Figure 6: Design Time Model of a Generic Data Collection and Analysis Application.

user, and to support self-analysis by that user or analysis by others, e.g. health and safety professionals.

The design time model created by our partner for this application is shown in Figure 6.

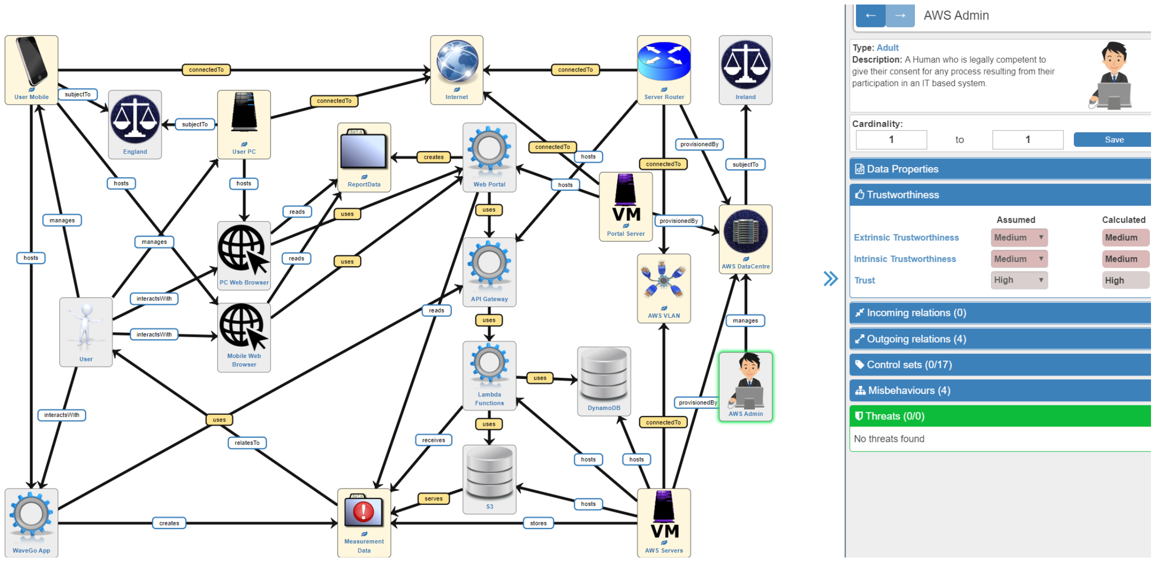
The user of the phone is in the lower-left quadrant, using the app to collect data and submit it via the API gateway process which is in the middle of the diagram. The analyst is shown top-left, and both they and the user have access to results from the data analysis via a browser running on their device. The analysis services are hosted by a data center, and use resources modelled on those supported by Amazon Web Services, although of course the cloud provider might not be Amazon.

The figure shows the model after the risk assessment procedure has been used to iteratively identify and address unacceptable risks. Various controls have been specified in the model, and at this stage risk levels are all Low or Very Low. At this stage the software/service provider may run penetration tests to verify their components, or have an independent penetration testing organization conduct such tests if their customers need independent verification of component security and trustworthiness.

A customer organization wishing to use the software would take the design time model as a starting point, including security control specifications and trustworthiness assumptions. They would then analyze the intended deployment context, and modify the risk model for their own purposes. The result of this process is shown in Figure 7.

The differences between these models reflect the specific deployment plans of the service operator. A major change is that the data collected by the user is now personal data relatedTo that user. The analyst role has also gone, and the user now manages both client devices and browsers, because in this case the intention is to provide a self-analysis service to users. Jurisdictions have been added showing that the operator is planning to deploy the service at a data center in one country, and serve clients in a second country. Finally, the operator has a lower opinion of their cloud service provider than the software/service developer has of Amazon, which they modelled by reducing the trustworthiness of the sysadmin role as shown.

These changes lead to two main differences in the assessment of risks. Firstly, a number of GDPR compliance threats will have been added to the threat catalogue representing technical requirements from the GDPR (in addition to the requirement to assess and manage risks). These include the need for a legal basis for processing personal data, which the operator decided to do by consent from the user. This means the process of registering to use the app must include a suitable notice allowing the user to express (or withhold) their consent. Secondly, because the cloud sysadmin role is considered only moderately trustworthy, risk levels from insider attacks are much higher, leading the operator to specify that trusted hardware must be accessible by the provisioned VMs so sensitive processes can be protected even from the system manager. If this is not available, they may decide to use their own in-house facilities instead of running this application in the cloud.



AWS Admin

Type: Adult
Description: A Human who is legally competent to give their consent for any process resulting from their participation in an IT based system.

Cardinality: 1 to 1

Data Properties

Trustworthiness

	Assumed	Calculated
Extrinsic Trustworthiness	Medium ▾	Medium
Intrinsic Trustworthiness	Medium ▾	Medium
Trust	High ▾	High

Incoming relations (0)

Outgoing relations (4)

Control sets (0/17)

Misbehaviours (4)

Threats (0/0)

No threats found

Figure 7: Deployment Model for the Data Collection and Analysis Application.

5 RELATED WORK

There is a recent case study on threat modeling that revealed the effectiveness and efficiency of threat modeling in large scale applications (Stevens et al., 2018). The participants reported that threat modeling supports the communication in teams and the development of mitigation strategies. Since we provide a tool-support methodology and graphical representations of the models, we improve the scalability for larger applications.

The ISO 27005 standard considers three major steps for risk identification and treatment: (i) identifying assets, (ii) identifying threats and evaluate their risks for the assets, and (iii) identifying controls to protect the assets against threats. Existing risk management methods mostly do not consider explicitly documenting and modeling the assets of the system under consideration.

Lund et al. propose the CORAS risk management process (Lund et al., 2011). The authors describe the process of identifying threats as an interactive brainstorming session. By considering our methodology, the threat and control identification is systematic and even can be used for structuring the brainstorming sessions, if necessary. Furthermore, before threat and control identification, using our methodology the context and boundaries for performing risk assessment is specified. We also provide more systematic way, for identifying asset in a model name design-time system model that include primary and supporting assets as described in ISO standard.

The ProCOR method (Wirtz et al., 2018) describes a risk management process which combines problem frames with CORAS. In this method, the identification of threats relies on external knowledge and lacks of a systemization. The same authors propose another method for identification of assets (Gol Mohammadi et al., 2019). The approach considers a problem frame model for the identification of relevant assets. In contrast to their work, we propose a combined approach with context modelling and a risk assessment method which enhance the asset identification. Furthermore, the threats and control identification are not only based on the expertise of security engineers, but also tool support with a comprehensive knowledge base of possible threats has been considered.

Abuse frames describe problem-oriented patterns to analyze security requirements from an attacker's point of view (Lin et al., 2003). An anti-requirement is fulfilled when a threat initiated by an attacker is realized. Domains are considered as assets. The malicious machine of an abuse frame acts as the interface between the attacker and the asset domain. Comparable to problem frames, abuse frames are patterns to describe the typical behavior of attackers. To apply an abuse frame, it is composed of a base problem which is represented by a problem frame. Composing means to map domains from the base problem into the abuse frame. Based on the composed abuse frame, the attacker's behavior can then be further analyzed. Abuse frames consider problems in isolation, whereas we consider the system as a whole together with its context.

Haley et al. provides a problem-oriented framework to analyze security requirements (Haley et al., 2008). Using so called satisfactory arguments, the authors analyzes a system with regard to security based on its functional requirements. There is no specific way to identify threats. Instead, our methodology allows to analyze the system as a whole together with its context, i.e. to analyze context domains, assets, threats and controls in combination.

In the context of goal-oriented software-engineering, Secure Tropos allows to to model and analyze security requirements along with functional requirements (Mouratidis and Giorgini, 2007). Another goal-oriented approach is the usage of anti-models (van Lamsweerde, 2004). Currently, we do not consider goals in detail and do not provide any goal refinement process. Instead, we put a special focus on data protection requirements and the corresponding regulations in context model and data protection requirement in possible data flows which we analyze with regard to information security.

Opdahl and Sindre introduce misuse cases as an extension of use cases. A misuse case describes how a software shall not be used, i.e. how potential attackers may harm the software (Sindre and Opdahl, 2005). Misuse cases are a high-level description which does not provide any details about protection of assets, as we do in our approach.

Microsoft developed a method called STRIDE (Shostack, 2014). It is a popular security framework which is used to identify security threats. Using data flow diagrams for modeling the system and its behavior, threats are elicited based on existing threat categories: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of privilege. Each of these categories is a negative counterpart to a security goal.

The STORE method allows to elicit security requirements in the earliest phases of a software development process (Ansari et al., 2019). The authors follow a similar approach to first identify domain knowledge such as external entities and vulnerabilities. To identify threats, the authors propose a threat dictionary which contains a collection of previously identified threats. However, their work has no systematic context analysis and asset identification prior to threat identification.

The other innovation of our approach is that we further take that gained knowledge from threat and control identification to further enrich the context information of a cloud system. The revised version of context model is beneficial during the deployment-time. Furthermore, our approach is conform to ISO standard. We employ context and asset models in our methodology, i.e., by developing a tool-assist methodology that makes it possible to set up an context model and asset model (named in this paper, design-time system model). Furthermore, our methodology supports the automated identification of threats and controls based on these models.

6 CONCLUSION

This paper described a methodology for applying an ISO 27005 risk assessment procedure to cloud based applications in which the operator of a service obtains software/services from a third party developer, and deploys using resources from a separate cloud resource provider. The methodology is based on the use of models, and provides several benefits over conventional approaches:

- the models provide a means for the software/service developer to contribute to the risk assessment, and communicate their requirements and assumptions unambiguously and in a useful form;
- the models allow context to be captured and incorporated by the service operator, enabling an assessment of risks in specific circumstances;
- the models ensure consistency and traceability, allowing the relationship between security properties and controls and consequent risk levels to be found and used when formulating security verification tests;
- it is possible to automate the identification of threats and assessment of risk levels, making it less likely that any significant threats (including multi-step attack paths and secondary effects) will be overlooked;
- automation also reduces the time and cost to repeat a risk assessment, making it easy to address different applications based on the same software, or to analyze a system if there is a change in the system or in the threat landscape.

In future, we aim to exploit our models even further, moving beyond the deployment and facilitating run-time risk assessments, so providing a means to comprehensively address the requirement under the GDPR Article 25 for security by design and by default, not only during the design of a system for handling PII, but also “at the time of the processing itself”.

ACKNOWLEDGEMENTS

This work received funding from the EU's Horizon 2020 research and innovation programme under grant agreement 731678 (RestAssured). We thank Roman Wirtz for his help in preparing the final version of this paper.

REFERENCES

Ansari, M. T. J., Pandey, D., and Alenezi, M. (2019). STORE: security threat oriented requirements engineering methodology. *CoRR*, abs/1901.01500.

Broy, M., Cengarle, M. V., and Geisberger, E. (2012). *Cyber-Physical Systems: Imminent Challenges*, pages 1–28.

Context-Patterns (2018). Overview, http://context-patterns.info/index.html.

European Union (2016). General Data Protection Regulation. https://gdpr-info.eu.

Gol Mohammadi, N., Wirtz, R., and Heisel, M. (2019). Systematic asset identification and modeling during requirements engineering. In *14th Intl. Conf. on Risks and Security of Internet and Systems (CRiSIS)*.

Haley, C. B., Laney, R. C., Moffett, J. D., and Nuseibeh, B. (2008). Security requirements engineering: A framework for representation and analysis. *IEEE Trans. Software Eng.*, 34(1):133–153.

ISO/IEC 27001 (2017). Information technology - Security techniques - Information security management systems - Requirements.

ISO/IEC 27005 (2018). Information technology - Security techniques - Information security risk management.

Lin, L., Nuseibeh, B., Ince, D. C., Jackson, M., and Moffett, J. D. (2003). Analysing security threats and vulnerabilities using abuse frames.

Lock, R. and Sommerville, I. (2010). Modelling and Analysis of Socio-Technical System of Systems. In *Proc. of the 15th IEEE Intl. Conf. on Engineering of Complex Computer Systems*, ICECCS, pages 224–232.

Lund, M. S., Solhaug, B., and Stølen, K. (2011). *Model-Driven Risk Analysis - The CORAS Approach*. Springer.

Mouratidis, H. and Giorgini, P. (2007). Secure tropos: a security-oriented extension of the tropos methodology. *intl. Jour. of Software Eng. and Knowledge Eng.*, 17(2):285–309.

Pohl, K. (2010). *Requirements engineering: Fundamentals, principles, and techniques*. Springer.

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

Sindre, G. and Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. *Requir. Eng.*, 10(1).

Stevens, R., Votipka, D., Redmiles, E. M., Ahern, C., Sweeney, P., and Mazurek, M. L. (2018). The battle for new york: A case study of applied digital threat modeling at the enterprise level. In *27th USENIX Security Symposium, USENIX Security*, pages 621–637.

van Lamsweerde, A. (2004). Elaborating security requirements by construction of intentional anti-models. In *26th Intl. Conf. on Soft. Eng. (ICSE),*, pages 148–157.

Wirtz, R., Heisel, M., Borchert, A., Meis, R., Omerovic, A., and Stølen, K. (2018). Risk-based elicitation of security requirements according to the ISO 27005 standard. In *Evaluation of Novel Approaches to Software Engineering - 13th Intl. Conf., ENASE*, pages 71–97.

Zwingelberg, H. and Hansen, M. (2011). Privacy protection goals and their implications for eid systems. In *IFIP PrimeLife Intl. Summer School on Privacy and Identity Management for Life*, pages 245–260. Springer.