Modeling Data Protection Vulnerabilities of Cloud Systems using Risk Patterns

Published in the Proceedings of the 10th System Analysis and Modeling Conference (SAM), pp. 1-19, 2018

Alexander Palm, Zoltán Ádám Mann, and Andreas Metzger

paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen, Essen, Germany

Abstract. Ensuring the protection of sensitive data is important for the adoption of cloud services. Cloud systems are becoming increasingly complex and dynamic, leading to various potential scenarios for attackers to get access to sensitive data. To handle such data protection risks, the concept of risk patterns was introduced previously. A risk pattern models a structural fragment of cloud systems that should not appear in the running system because it would lead to high data protection risks. At deployment and at run time, graph pattern matching and dynamic re-configuration methods can be used to ensure that the run-time model of the cloud system contains no instance of the risk patterns.
The previous work left it open, however, how and to what extent real data protection vulnerabilities can be modeled in the form of risk patterns. Therefore, this paper focuses on the design of risk patterns based on vulnerabilities described in the literature. Based on an analysis of 87 papers, we determined 45 risk patterns. Our findings (i) demonstrate that risk patterns can indeed capture many of the vulnerabilities described in the cloud literature, (ii) give insight into the typical structure of risk patterns, and (iii) show the limits of the applicability of the risk pattern approach.

Keywords: Cloud computing, Data protection, Privacy, Run-time model, Risk pattern

1 Introduction

Cloud computing is increasingly popular, thanks to the benefits it brings to both providers and users of cloud services. However, outsourcing sensitive data to the cloud puts the data at a risk, which many users of cloud services are not ready to accept [16].

Data protection in the cloud is hard because cloud systems are increasingly complex and dynamic. They consist of many different physical and virtual machines, as well as various applications and their software components, all of which

interact and may dynamically reconfigure during run time [1, 9, 15, 30]. In addition, a multitude of stakeholders may be involved, such as service consumers, cloud providers, data subjects, data controllers, and actual end users. Due to such complex interactions, a cloud system may expose vulnerabilities that enable attackers to gain access to sensitive data stored in the cloud. Moreover, since the attributes and interactions of the cloud entities continuously change, *data protection vulnerabilities* may arise during operation. By data protection vulnerability, we mean the possibility of unauthorized access to sensitive data. This is not the same as a system vulnerability (e.g., if a vulnerable system neither stores nor has access to sensitive data, then there is no data protection vulnerability), but system vulnerabilities may lead to data protection vulnerabilities which put sensitive data at risk.

To identify and mitigate data protection risks in complex and dynamic cloud systems, we have introduced the concept of risk patterns in our earlier work [26]. That approach was based on two types of artefacts:

- A model of the – current or planned – configuration of the cloud system, including infrastructure elements, middleware, applications, data, and the involved actors;
- A set of *risk patterns*, which describe cloud configurations that would cause too high risks of data protection violation and hence must be avoided.

For modeling the configuration of the cloud system, a meta-model was proposed [17]. When a cloud system is to be deployed, the system designer creates the model of the planned configuration as an instance of the meta-model. When the configuration changes during the deployment process or later during the operation of the system, the model is updated accordingly, so that it always reflects the current state of the cloud system and can be used as a run-time model.

Risk patterns are expressed in a domain-specific language based on the same meta-model as the cloud model. Risk patterns model fragments of a cloud system by specifying the presence or absence of certain entities, attributes, or relations. Risk patterns capture forbidden fragments of a cloud system model that would exhibit overly high data protection risks. During deployment and at run time, the model of the cloud system is checked for the existence of fragments corresponding to risk patterns. If an instance of a risk pattern is found in the cloud model, a potential data protection vulnerability is identified, which may be mitigated with appropriate changes of the deployment or by run-time adaptation.

Our previous work [26, 17] evaluated the risk pattern approach using two example risk patterns. The evaluation showed that, if the relevant data protection vulnerabilities are captured in the form of risk patterns, then these risk patterns can indeed be used to detect and mitigate the data protection risks during deployment and at run time. The prerequisite is a catalog of risk patterns capturing the relevant data protection vulnerabilities. Our previous work did not address in detail how risk patterns can be devised, leaving several questions open:

- Is it feasible to model a broad range of real data protection vulnerabilities in the form of risk patterns?

– What is the typical size and structure of risk patterns? (This is important as it impacts the applicability of graph pattern matching algorithms in terms of their computational complexity (which in turn is not part of this paper))
– For which kinds of data protection vulnerabilities is the risk pattern approach appropriate?

This paper seeks to answer these questions by gaining experience with modeling risk patterns. Specifically, we review 87 papers from the cloud security literature (which were collected in a previous survey [3]) and identify the ones that describe relevant vulnerabilities in sufficient detail. Then, we devise risk patterns for the vulnerabilities described in these papers. This results in a total of 45 risk patterns.

Our findings show that most of the vulnerabilities that were described in sufficient detail in the respective papers could indeed be captured by appropriate risk patterns, thus demonstrating the general applicability of the risk pattern approach. All identified risk patterns share the same high-level structure and consist of 6 to 10 entities. This suggests that graph pattern matching can indeed be efficiently used to find risk patterns in cloud models. Also some limitations of the risk pattern approach are uncovered, relating to both the types of vulnerabilities that can be captured (e.g., vulnerabilities resulting from human and social aspects are not appropriate) and the underlying cloud meta-model (a very fine-grained meta-model can lead to a proliferation of many similar risk patterns to capture essentially the same vulnerability).

The remainder of this paper is organized as follows. In Section 2 we review the meta-model underlying the risk pattern approach. Section 3 then gives an overview of the methodology used to define the risk patterns and Section 4 presents the structure of risk patterns. In Section 5 we describe the risk patterns that we derived from the literature. Section 6 summarizes the lessons learned during the process, while Section 7 describes related work and Section 8 concludes the paper.

2 Cloud Meta-Model

In this section we briefly review the previously proposed meta-model [17]. The model of the cloud system, which plays a central role in the risk pattern approach, is an instance of this meta-model. Further, the risk patterns also reference entities, attributes, and relations from this meta-model.

The meta-model consists of the packages Actors and Assets (see Fig. 1). The Actors package defines the different data-specific roles (e.g., data subject, data controller) and cloud-specific roles (e.g., infrastructure provider) that a natural or legal person can have, and a trust relationship that can exist between different actors. An actor can also access and/or own assets, e.g. an actor can access a virtual machine.

The Assets package is further divided into the sub-packages Data, Applications, Middleware and Infrastructure. The elements necessary to model the data that has to be protected are given by the data sub-package. The main element

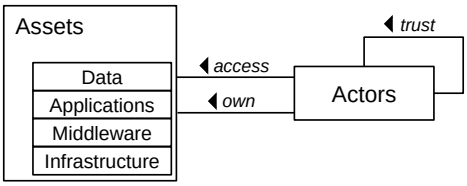


Fig. 1. Abstract view of the meta-model for cloud models [17]

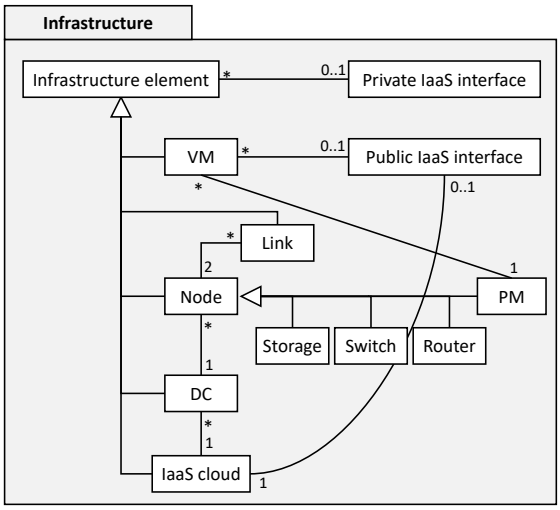


Fig. 2. Infrastructure sub-model of the meta-model for cloud models [17]

within this sub-package is the data object. Data can be stored in form of a stored data set or exchanged between different application components via a data flow element. The application sub-package comprises the elements needed to model software elements, like applications with different application components and connectors between them. Middleware elements, like web servers, application servers and database management systems are available in the middleware sub-package. The elements needed to model the infrastructure of a cloud system, like virtual machines (VMs), physical machines (PMs) and data centers (DCs) are given by the infrastructure sub-package. As an example, Fig. 2 shows the contents of the infrastructure sub-package. The full meta-model is shown in the Appendix.

3 Methodology

In this section we describe the methodology that we used to derive a catalog of risk patterns.

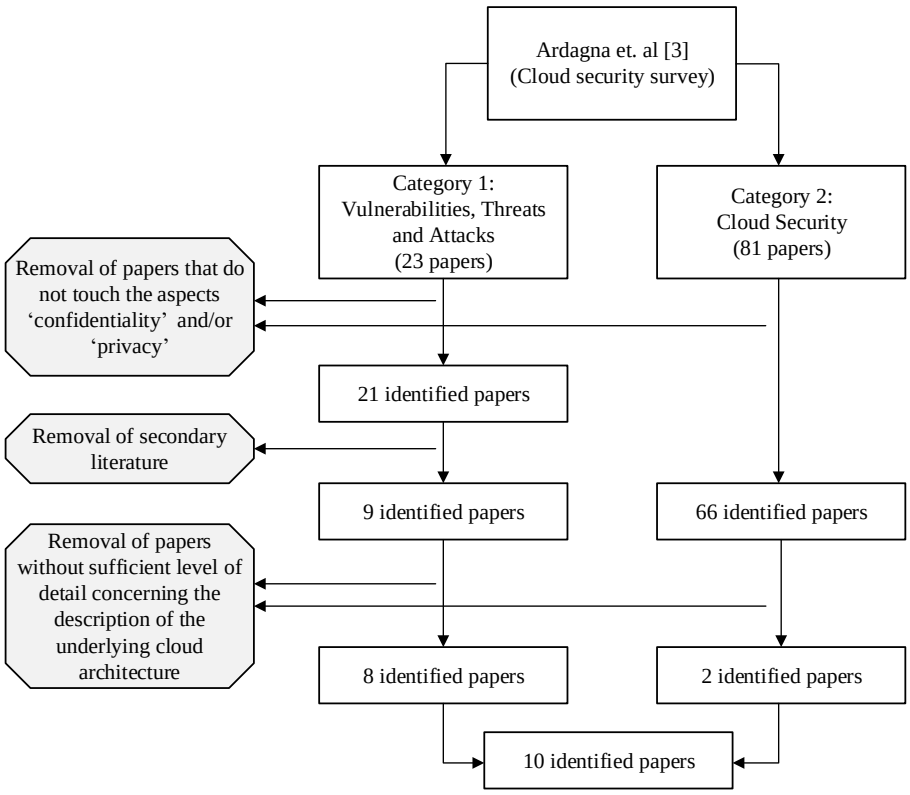


Fig. 3. Overview of the literature analysis

The starting point was a survey about cloud security [3]. From the categories defined in that survey paper, we focused on the categories 'vulnerabilities, threats and attacks' and 'cloud security', and analyzed all papers in those categories which mentioned the aspects *confidentiality* or *privacy*. In the publications of the first category, different attacks on cloud systems are described. Nearly all papers of this category (21 out of 23) touch the aspects confidentiality or privacy. Publications of the second category focus on security solutions and do not mention confidentiality or privacy to the extent the publications of the first category do (66 of 81 papers were considered relevant).

In the next step, we removed the secondary literature from the first category, leaving 9 out of 21 papers of this category. We analyzed all remaining papers for their level of detail concerning the description of an attack on or a vulnerability of a cloud system and the underlying cloud architecture. Only two papers of the category 'cloud security' described the underlying problem of the security solution in sufficient detail, so 64 papers were removed. In the category 'vulnerabilities, threats and attacks', 8 out of 9 papers were detailed enough.

Table 1. Overview of relevant literature

Publication	Category	No. of risk patterns
Somorovsky et al. [29]	Control Interfaces	12
Aviram et al. [4]	Side Channel	24
Godfrey & Zulkernine [11]	Side Channel	
Green [12]	Side Channel	
Okamura & Oyama [19]	Side Channel	
Ristenpart et al. [22]	Side Channel	
Zhang et al. [32]	Side Channel	
Rocha & Correia [23]	Privilege Exploitation	7
Sedayao et al. [27]	Privilege Exploitation	
Bernsmed et al. [5]	Service Mistrust	2

In the end, 10 papers remained that served as a basis for the modeling of risk patterns. An overview of these publications is given in Table 1 and an overview of the literature analysis is given in Fig. 3.

After the analysis of the literature, we derived risk patterns based on the relevant excerpts of cloud architectures described in the selected publications. The modeling of a risk pattern was done in three steps:

1. Analysis and description of the attack
2. Identification of the underlying system vulnerability
3. Identification of the relevant paths within the meta-model

After the analysis and description of the attack, the main goal was to identify the specific system vulnerability exploited by the attack. This includes the identification of elements of the meta-model suitable to model this vulnerability and in particular its attack point. After this, the sensitive data that should be protected are modeled and in the third step the relevant paths connecting the sensitive data with the attack point are identified.

4 Structure of Risk Patterns

A risk pattern is a sub-structure of a cloud system configuration, which threatens the protection of sensitive data and therefore has to be avoided [26]. A risk pattern can typically be divided into three parts (see Fig. 4):

- Part (a) represents the *personal data* that need to be protected. These data are always modeled by the same elements of the data package: a data record which is part of a stored data set, and an actor who is the data subject that the data belong to.
- Part (b) represents the *attack point* of the system vulnerability: the point of the configuration through which an attacker gets access to the system. This part of the risk pattern depends on the type of the modeled attack.

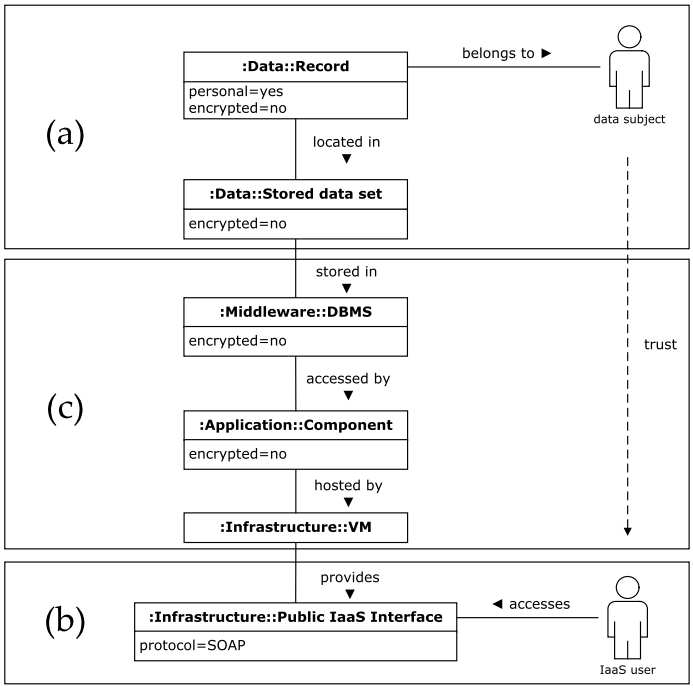


Fig. 4. An example risk pattern, structured into three parts

– Part (c) contains two *connections* between part (a) and part (b). One connection is between the sensitive data and the attack point, the other one between the actors. The possibilities for the first connection are determined by the underlying meta-model and depend on the 'distance' (within the meta-model) between parts (a) and (b). The second connection is always a mistrust relation between the data subject and the attacker. (Note: A dashed line implies that the according relation must not exist, whereas a solid line implies that the according relation must exist)

The attributes of the entities also play an important role. Sometimes vulnerabilities only differ in some attributes.

5 The Devised Risk Patterns

In this section we present how we modeled different types of vulnerabilities from the literature by different categories of risk patterns. The categorization is based on the different attack points of the risk patterns. Overall the risk pattern catalog includes 45 risk patterns in four categories. For reasons of space we include here only some examples. The full risk pattern catalog is available under https://zenodo.org/record/1324125#.W2A2mrhCREY.

5.1 Category: Control Interfaces

The first category of our catalog comprises risk patterns modeling attacks on control interfaces. Control interfaces are interfaces which give users the opportunity of maintaining their resources. The maintenance of resources includes the instantiation, starting and shut-down of virtual machines. Although such interfaces are protected with measures like authorization and signatures, still vulnerabilities exist. To provide the underlying basics for the definition of the risk patterns of this category, we first introduce an attack scenario targeting a vulnerability of a control interface, before we then describe the risk patterns derived from this scenario.

Underlying attacks. The attack scenarios which serve as a baseline for the definition of the risk patterns of this category are described in [29]. To provide the possibility of maintaining resources, Amazon Web Services (AWS) provides mainly two interfaces: a SOAP interface and a Web interface.

The SOAP interface is based on the Simple Object Access Protocol (SOAP) which uses an X.509 certificate for the identification of the user and an XML-based signature to enable authentication and prove the integrity of a message. Furthermore the SOAP messages themselves are based on XML. The authors of [29] proved that SOAP is vulnerable to so-called signature wrapping attacks. To perform a signature wrapping attack on SOAP, the attacker has to intercept a SOAP message exchanged between the user and the interface. Then the attacker can add an additional message body to the intercepted message and reuse the signature. This enables the attacker to perform arbitrary operations on the SOAP interface, because only the body referenced in the signature is verified for integrity, but the additional body is interpreted.

The Web interface enables an attacker to perform a so-called script injection attack on the cloud interface. As this attack is also founded in the underlying protocol (HTTP), the derived risk patterns are analogous to those of the aforementioned attack. Based on these attack scenarios, the usage of SOAP and HTTP can be considered as a data protection vulnerability of a cloud system and therefore is modeled as a risk pattern.

Risk pattern definition. The attack point of the risk patterns of this category is shown in Fig. 5 (Note: this is an excerpt of the corresponding risk pattern shown in Fig. 4). The interface is modeled as a 'Public IaaS Interface' entity from the Infrastructure package of the meta-model. The IaaS user accessing the interface may behave as the attacker of the attack scenario described above, thus gaining unauthorized access to sensitive data. The protocol of the interface can be identified by the attribute 'protocol', which is 'SOAP' in Fig. 5.

The meta-model allows multiple possibilities for connecting the 'Public IaaS Interface' of Fig. 5 with the sensitive 'Data Record', i.e., multiple possibilities for the attacker to actually access sensitive data, depending on the specific cloud configuration. The risk pattern shown previously in Fig. 4 is one possibility,

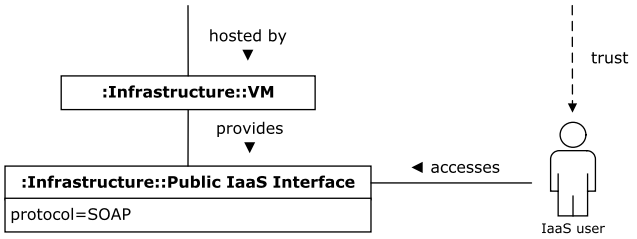


Fig. 5. Excerpt of a risk pattern of the category 'control interfaces'

in which the access takes place through a chain of a virtual machine (VM), an application component, and a database management system (DBMS). The meta-model allows five further possibilities, described by the following chains of assets:

- VM → application component → local database
- VM → application server → application component → local database
- VM → DBMS
- VM → application server → DBMS
- VM → application server → application component → DBMS

These risk patterns capture different cloud configurations that exhibit a similar data protection vulnerability.

5.2 Category: Side Channels

The second category of our catalog includes risk patterns modeling side channel attacks. Side channels are based on a shared resource (e.g. CPU cache) which enables data leakage or is misused for communication between two virtual machines that are co-located on the same physical machine but belong to different users.

Underlying attacks. Side channels can have two consequences: they can be misused for communication between otherwise isolated VMs [19, 22] or for data leak [12, 32, 11, 22]. Side channels always rely on multi-tenancy, which means that a physical machine is shared among different users.

How such co-location can be accomplished in Amazon EC2 is described in [22]. An attacker can instantiate lots of VMs of the same instance type and inside the same availability zone as the victim's VM. Doing this, there is a high probability that one of the instantiated VMs is on the same physical machine as the victim's one. The probability can even be increased if the instantiation process is launched right after the victim's instance is re-instantiated, because following Amazon EC2's VM placement strategy, physical machines with free capacity are filled first before new physical machines are started. After co-location is achieved, the way is cleared for one of the following attacks or techniques.

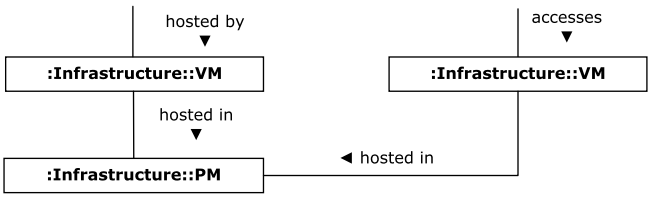


Fig. 6. Excerpt of an abstract risk pattern of the category 'side channels'

In [19] a technique called CCCV for the use of the CPU load as a side channel is described. The technique is possible if virtual CPUs (VCPUs) of different VMs share the same physical CPU. Assuming a spyware was injected into the victim's VM, the CPU load can be manipulated, so that data can be transfered adhering to the protocol described in [19]. This protocol is based on the fact that a high CPU load of one VM affects the performance of co-located VMs, and can thus be observed by them. A different attack where CPU load is misused as a side channel for communication is also described in [22].

The Prime+Trigger+Probe (PTP) technique uses a shared cache as side channel [22, 11]. As its name implies, PTP comprises three phases. Within the 'prime' phase, the attacker fills all lines of the shared cache and measures the time needed to read each of these lines. In the following 'trigger' phase, the attacker hands over the control of the shared cache to the victim's VM. The victim's VM then may change some lines of the cache and hand over control back to the attacker's VM. The change of cache lines results in cache misses when the attacker probes the cache during the 'probe' phase. This technique can be used to communicate between the two VMs [22] when a change of the cache is interpreted as the sending of a '1' and no change of the cache is interpreted as a '0'. Furthermore, this technique can be used to extract sensitive data such as private keys [12], because the attacker can possibly determine which operations were carried out by the victim's VM based on the changes of the access times of different cache lines and also based on which cache lines have been changed. A slightly different technique is described in [32] and a scenario stating possible consequences of such an attack is described in [4].

Risk pattern definition. As side-channel attacks always rely on a shared resource used by two co-located VMs and only differ in the type of resource used and in nuances of hardware settings, we introduced *abstract risk patterns* which can be made concrete through attributes of the concerned elements depending on the specific vulnerability that should be exploited. The attack point of the abstract risk patterns (and therefore also of all concrete risk patterns of this category) comprises two VMs being hosted on the same physical machine (see Fig. 6). The actor accessing one of the VMs (and thus serving as attacker) is not trusted by the data subject. Attributes used to concretize abstract risk

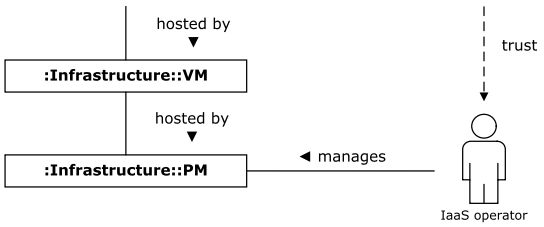


Fig. 7. Excerpt of a risk pattern of the category 'privilege exploitation'

patterns include 'hypervisor' to specify a certain hypervisor and 'cpu-scheduling' to specify a certain kind of CPU scheduler.

5.3 Category: Privilege Exploitation

The third category of our catalog comprises risk patterns modeling privilege exploitation attacks. In privilege exploitation attacks, an administrator abuses their privileges to get access to sensitive data [23, 27].

Underlying attacks. Administrators of cloud systems normally have no rights to log on to client VMs. However, an administrator with root privileges can generate memory dumps of client VMs for troubleshooting. The administrator can also misuse this opportunity to extract private data (e.g., cryptographic keys) from such memory dumps [23]. Although data might be stored encrypted on permanent storage, the administrator can get access to the cleartext if the memory dump is generated at the right moment (i.e. when data are decrypted for processing). Because private keys are often stored as ASN.1-objects, an attacker just needs to search for typical byte sequences of ASN.1-objects within the memory dump. This attack becomes more difficult if secure hardware is used that prevents the memory from being dumped. In this case, an attacker may trigger a VM relocation first, and perform the attack when the VM is relocated on a physical machine which is not using this kind of hardware. A similar kind of attack is also possible on storage devices [27].

Risk pattern definition. The attack point of the risk patterns of this category is shown in Fig. 7. As all attacks of this category are based on an administrator abusing their privileges to access sensitive data, this situation is modeled in the risk patterns of this category. More specifically, an untrusted IaaS operator managing the physical machine on which the VM with the personal data of the data subject is hosted is accessing these data.

5.4 Category: Service Mistrust

The fourth category of our catalog comprises risk patters modeling the problem of mistrust within service compositions. Because service compositions consist of

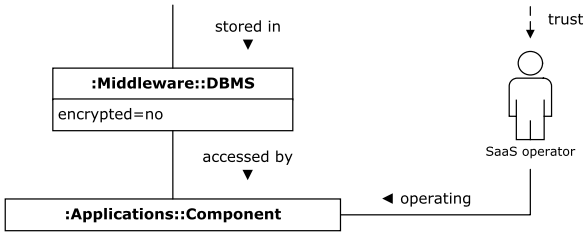


Fig. 8. Excerpt of a risk pattern of the category 'service mistrust'

different services which are composed in a hierarchical fashion and belong to different providers, a data subject may not trust – and may not even know – some of the involved providers. An untrusted provider getting access to sensitive data constitutes a data protection risk.

Underlying attacks. Bernsmed et al. [5] consider a scenario where a service provider might access personal data processed by its service. In service compositions, the situation of trust becomes more complex. Although there may be a chain-of-trust between the participating providers of a service compositions, a data subject whose data is processed by the composed service, does not necessarily trust all of the involved providers. The situation becomes even worse if the data subject does not even have knowledge about the participating providers. Therefore this situation is a data protection vulnerability and can be modeled as a risk pattern.

Risk pattern definition. The attack point of the risk patterns of this category is shown in Fig. 8. The situation is modeled by a SaaS (Software as a Service) operator that is responsible for the operation of an application component – which represents a service in a possibly larger composition of services – and therefore can access sensitive data processed by this application component.

6 Lessons Learned

Through the modeling of real data protection vulnerabilities in the form of risk patterns, we have gained insight into both the applicability of the risk pattern approach and the characteristics of typical risk patterns.

6.1 Applicability of the Risk Pattern Approach

Since we managed to represent several real data protection vulnerabilities in a natural way in the form of risk patterns, we can state that *the risk pattern approach is appropriate for modeling data protection vulnerabilities.* In particular,

quite complex attack scenarios spanning multiple cloud layers could be modeled, and also very different kinds of attacks.

That said, it is important to note that the attacks described in several papers could not be reasonably modeled in the form of risk patterns. In some cases, this was due to a lack of detail about the vulnerabilities in the respective papers (cf. Fig. 3), because those papers focused primarily on describing data protection techniques against some classes of attacks, rather than describing specific vulnerabilities in detail. However, there were also cases where the non-applicability of the risk pattern approach had other reasons, and these reasons shed some light on the limits of the applicability of the approach:

- Some vulnerabilities are not technical, but stem from human or organizational factors, e.g., lack of security training for personnel (e.g. the vulnerability described in [7] lies is the careless behaviour of users not cleaning their Amazone Machines Images of passwords before making them available for others). In contrast, risk patterns are appropriate for capturing forbidden socio-technical configurations that are at least partly *technical* in the way they expose data.
- Several papers focus on attacks and not on vulnerabilities, e.g., describing several ways an attacker could exploit some basic vulnerability. Risk patterns are not meant to capture specific attacks, but rather configurations that lead to high *risks* of successful attacks (therefore the risk patterns stemming from different side-channel attacks (cf. Section 5.2) only slightly differ.). Thus, the use of risk patterns is more proactive than, for instance, intrusion detection techniques [28].
- Several papers focus on system vulnerabilities which do not necessarily imply data breach (e.g. attacks compromising the availability of resources, cf. [6, 13]). Although risk patterns could in principle also be used to model such system vulnerabilities, our focus was on *data protection* vulnerabilities, thus rendering some attacks irrelevant.
- Some papers describe vulnerabilities that are not cloud-related. While risk patterns could in principle also be used in other contexts, the currently used meta-model is *cloud-specific*, hence we only considered cloud-related vulnerabilities. However, we decided to differentiate between cloud-related side-channel attacks and the more general virtual machine escape (cf. [21]). The latter was not cloud-related for us and therefore excluded.
- Some papers describe unlikely attacks, i.e., attacks that work only under strong assumptions about the possibilities of the attacker. Risk patterns are supposed to be created in the course of risk assessment, covering the configurations that are considered to be *too risky* – not necessarily all configurations that might allow some attack under unlikely conditions. This also applies to some of the configurations we modeled in this paper.

The applicability of the approach is also strongly related to how cloud configurations can be modeled by using the types from the meta-model described in [17]. The experience with using the meta-model has shown that it is indeed

a solid basis for modeling complex cloud configurations. In particular, the possibility of the meta-model to combine different hardware and software entities, data, and actors in a single model has proven invaluable, since all risk patterns span several of these categories. Some small extensions to the meta-model were also necessary, e.g., some new attributes and relationships between certain assets and actors had to be introduced (e.g. to model the direct access of an IaaS user to a virtual machine or a DBMS running directly on a virtual machine) . This is normal; project-specific tailoring of the meta-model was also envisaged in [17].

6.2 Characteristics of Risk Patterns

As shown in Section 4, risk patterns have a common structure. All risk patterns that we devised follow this same structure. From a graph-theoretic point of view, a risk pattern defines a path from an attacker to the sensitive data (the path through which the attacker may be able to access the data), plus an additional path between the same two vertices encoding that the data belong to a data subject who does not trust the attacker. This means that instead of a general graph pattern matching problem as suggested in [26], only subgraphs of very limited structure (cycle graphs) must be searched for in the cloud model, which may require significantly less computation.

For assessing the computational implications, it is also an important finding that the risk patterns are quite small: all the devised risk patterns consist of 6 to 10 entities, with exactly two connections per entity.

Another aspect is the number of risk patterns. In particular, a single system vulnerability leads to multiple risk patterns: if an asset is compromised from which there are k different kinds of paths to the sensitive data, then potentially k risk patterns are needed to capture all possible data protection vulnerabilities stemming from the same system vulnerability (and it is possible to use different data protection mechanisms to protect each of those paths). It is important to note that the number k of different kinds of paths from an asset to the data depends on the meta-model and especially the possible connections between the different entities of it. With the meta-model used in this work, $k \leq 6$. Moreover, for some system vulnerabilities, indeed 6 different risk patterns were needed; for some other system vulnerabilities, a lower number of risk patterns was sufficient. If the meta-model were refined with further types and relationships, this could lead to higher values of k and thus to a proliferation of similar risk patterns (i.e. they only differ in the possible paths between the compromised asset and the sensitive data). Therefore, the level of detail of the meta-model constitutes an important trade-off between the accuracy of modeling cloud configurations and the effort for modeling the risk patterns.

7 Related Work

We discuss the work most relevant to ours along two aspects: (1) data protection risks of cloud services and (2) model-based approaches for cloud security and privacy.

7.1 Data Protection Risks of Cloud Services

Risk management covers the process of describing, detecting and mitigating risks. So far, only few frameworks for risk management of services have been presented [18].

Djemame et al. [8] propose a risk assessment framework for cloud computing which is designed to help cloud users and cloud providers assess risks during service deployment and operation. This approach focuses on the relationship between service providers and services. However, they do not state how risks may be monitored during operations. This is where risk patterns can help by specifying what cloud configurations to look for during operations to determine risky situations.

Meszaros and Buchalcevova [18] present a framework for online service risk management. They consider similar assets to ours and present a risk and threat model as basis. They focus on risk assessment and mitigation and propose techniques for risk monitoring. Our approach can be considered complementary to their work as our risk patterns capture specific configurations of cloud services and systems that would lead to high data protection risks.

Several authors have analyzed specific data protection risks in the context of cloud computing and services. Paquette et al. [20] analyzed the risks of cloud computing, focusing on the context of governmental use of cloud computing. Fernandes et al. [10] surveyed security issues in cloud computing as a potential source for data protection risks. These insights provide an important source of input for our approach as they help defining and specifying risk patterns by taking important data protection concerns into account. Our approach can be seen as a vehicle for capturing and utilizing this kind of knowledge.

7.2 Model-Based Approaches for Cloud Security and Privacy

Apvrille and Roudier introduced Attack Graphs based on the SysML-Sec framework [2]. Similar to risk patterns, also attack graphs are visual representations of security threats. However, attack graphs are used to model malicious attacks on – especially embedded – systems, whereas risk patterns encode cloud configurations that can potentially lead to data protection issues. That is, an attack graph models all the details of an attack, including the tools used and activities performed by an attacker, whereas a risk pattern models only the cloud configuration that could potentially be exploited, thereby enabling more general preventive measures. Attack graphs were applied to model a single attack, whereas we compiled a catalog of 45 risk patterns.

Watson and Little [31] introduced an approach to reason about the deployment of a distributed system and its impact on security. They state that not all deployment problems can be solved during design time, so run-time reasoning is needed. In contrast to our risk patterns, their approach requires the assignment of security levels to all assets, which can be difficult in some settings. In fact, risk patterns could help here: the number of risk patterns in which a given asset type appears may be used as an indication of the security requirements of that asset.

Beyond the types of entities considered in that paper, we also explicitly consider actors. As shown in our paper, actors are important for accurately determining data protection concerns.

Similarly to our approach, the work of Schmieders et al. also applied model-based adaptive methods to data protection in the cloud [24, 25]. That work, however, is limited to one specific type of privacy goals: geo-location constraints. Our work, in contrast, addresses data protection goals in a much broader sense.

Kritikos and Massonet proposed a domain-specific modeling language for modeling security aspects in cloud computing [14]. This includes security controls, security properties, security metrics, and security capabilities. In contrast, our work focuses on modeling the typical assets of cloud systems and their relationships, which are the possible attack surfaces and make up the configurations that may lead to data protection violations.

8 Conclusions and Future Work

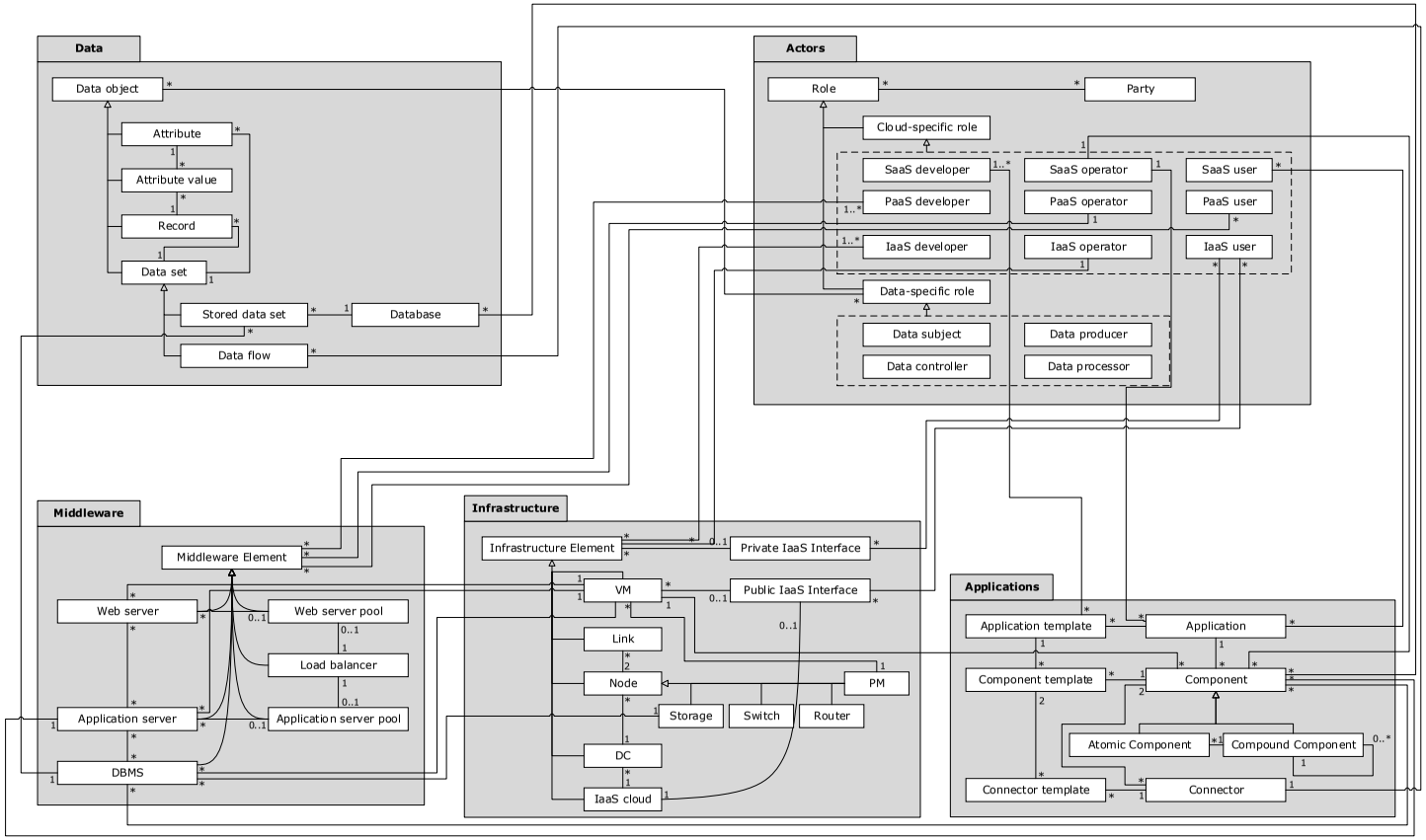
In this paper, a catalog of risk patterns was elaborated on the basis of vulnerabilities described in the literature. The results demonstrate that our previously proposed risk pattern approach [26] in combination with the meta-model described in [17] is capable of modeling typical data protection vulnerabilities. The present work also sheds light on the limits of the applicability of the risk pattern approach and the typical characteristics of risk patterns.

Several directions for future work remain. First, the syntax and semantics of the *language* of risk patterns should be defined formally. The catalog of risk patterns elaborated in this work is an important input for the formal definition of the language, as it shows the different constructs that must be supported. Second, the *process* of devising risk patterns should be formalized based on the experience reported here, and then validated by using it to create further risk patterns. Third, an efficient *algorithm* should be devised and implemented to search for risk patterns in the cloud model, using the gained insights about the structure and size of risk patterns. Fourth, it should be investigated how the expressive power of the language could be increased by introducing *wildcards* or other mechanisms to compactly represent families of related risk patterns.

Acknowledgment. This work was partially supported by the European Union's Horizon 2020 research and innovation programme under grant 731678 (RestAssured).

Appendix

The following picture shows the underlying meta-model (without the packages 'Goals & Metrics' and 'Mechanisms'):


Data
Data object
Attribute
Attribute value
Record
Data set
Stored data set
Database
Data flow
Actors
Role
Party
Cloud-specific role
SaaS developer
SaaS operator
SaaS user
PaaS developer
PaaS operator
PaaS user
IaaS developer
IaaS operator
IaaS user
Data-specific role
Data subject
Data producer
Data controller
Data processor
Middleware
Middleware Element
Web server
Web server pool
Load balancer
Application server
Application server pool
DBMS
Infrastructure
Infrastructure Element
Private IaaS Interface
VM
Public IaaS Interface
Link
Node
PM
Storage
Switch
Router
DC
IaaS cloud
Applications
Application template
Application
Component template
Component
Atomic Component
Compound Component
Connector template
Connector

References

1. Ahvar, E., Ahvar, S., Mann, Z.Á., Crespi, N., Garcia-Alfaro, J., Glitho, R.: CACEV: a cost and carbon emission-efficient virtual machine placement method for green distributed clouds. In: IEEE International Conference on Services Computing (SCC). pp. 275–282. IEEE (2016)
2. Apvrille, L., Roudier, Y.: SysML-Sec attack graphs: compact representations for complex attacks. In: International Workshop on Graphical Models for Security. pp. 35–49. Springer (2015)
3. Ardagna, C.A., Asal, R., Damiani, E., Vu, Q.H.: From security to assurance in the cloud: A survey. ACM Comput. Surv. 48(1), 2:1–2:50 (2015)
4. Aviram, A., Hu, S., Ford, B., Gummadi, R.: Determinating timing channels in compute clouds. In: Proceedings of the 2nd ACM Cloud Computing Security Workshop, CCSW 2010, Chicago, IL, USA, October 8, 2010. pp. 103–108 (2010)
5. Bernsmed, K., Jaatun, M.G., Meland, P.H., Undheim, A.: Thunder in the clouds: Security challenges and solutions for federated clouds. In: IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom). pp. 113–120. IEEE (2012)
6. Booth, G., Soknacki, A., Somayaji, A.: Cloud security: attacks and current defenses. In: 8th Annual symposium on information Assurance (ASIA'13). pp. 56–62 (2013)
7. Bugiel, S., Nürnberger, S., Pöppelmann, T., Sadeghi, A.R., Schneider, T.: Amazonia: when elasticity snaps back. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011. pp. 389–400 (2011)
8. Djemame, K., Armstrong, D., Guitart, J., Macias, M.: A risk assessment framework for cloud computing. IEEE Transactions on Cloud Computing 4(3), 265–278 (2016)
9. Elrotub, M., Gherbi, A.: Virtual machine classification-based approach to enhanced workload balancing for cloud computing applications. Procedia Computer Science 130, 683–688 (2018)
10. Fernandes, D.A.B., Soares, L.F.B., Gomes, J.V.P., Freire, M.M., Inácio, P.R.M.: Security issues in cloud environments: a survey. Int. J. Inf. Sec. 13(2), 113–170 (2014)
11. Godfrey, M., Zulkernine, M.: A server-side solution to cache-based side-channel attacks in the cloud. In: IEEE Sixth International Conference on Cloud Computing (CLOUD). pp. 163–170. IEEE (2013)
12. Green, M.: The threat in the cloud. IEEE Security & Privacy 11(1), 86–89 (2013)
13. Jensen, M., Schwenk, J., Gruschka, N., Iacono, L.L.: On technical security issues in cloud computing. In: IEEE International Conference on Cloud Computing, CLOUD 2009, Bangalore, India, 21-25 September, 2009. pp. 109–116 (2009)
14. Kritikos, K., Massonet, P.: An integrated meta-model for cloud application security modelling. Procedia Computer Science 97, 84–93 (2016)
15. Mann, Z.Á.: Multicore-aware virtual machine placement in cloud data centers. IEEE Transactions on Computers 65(11), 3357–3369 (2016)
16. Mann, Z.A., Metzger, A.: Optimized cloud deployment of multi-tenant software considering data protection concerns. In: Proc. of the 17th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing. pp. 609–618. IEEE Press (2017)
17. Mann, Z.A., Metzger, A., Schoenen, S.: Towards a run-time model for data protection in the cloud. In: Modellierung 2018. pp. 71–86. Gesellschaft für Informatik e.V. (2018)

18. Meszaros, J., Buchalcevova, A.: Introducing OSSF: A framework for online service cybersecurity risk management. Computers & Security 65, 300–313 (2017)
19. Okamura, K., Oyama, Y.: Load-based covert channels between Xen virtual machines. In: Proceedings of the 2010 ACM Symposium on Applied Computing. pp. 173–180. ACM (2010)
20. Paquette, S., Jaeger, P.T., Wilson, S.C.: Identifying the security risks associated with governmental use of cloud computing. Government Information Quarterly 27(3), 245–253 (2010)
21. Pearce, M., Zeadally, S., Hunt, R.: Virtualization: Issues, security threats, and solutions. ACM Computing Surveys (CSUR) 45(2), 17 (2013)
22. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. pp. 199–212. ACM (2009)
23. Rocha, F., Correia, M.: Lucy in the sky without diamonds: Stealing confidential data in the cloud. In: IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W). pp. 129–134. IEEE (2011)
24. Schmieders, E., Metzger, A., Pohl, K.: Architectural runtime models for privacy checks of cloud applications. In: Proceedings of the Seventh International Workshop on Principles of Engineering Service-Oriented and Cloud Systems. pp. 17–23 (2015)
25. Schmieders, E., Metzger, A., Pohl, K.: Runtime model-based privacy checks of big data cloud services. In: International Conference on Service-Oriented Computing. pp. 71–86 (2015)
26. Schoenen, S., Mann, Z.Á., Metzger, A.: Using risk patterns to identify violations of data protection policies in cloud systems. In: 13th International Workshop on Engineering Service-Oriented Applications and Cloud Services (WESOACS). (2017)
27. Sedayao, J., Su, S., Ma, X., Jiang, M., Miao, K.: A simple technique for securing data at rest stored in a computing cloud. In: IEEE International Conference on Cloud Computing. pp. 553–558. Springer (2009)
28. Shameli-Sendi, A., Cheriet, M., Hamou-Lhadj, A.: Taxonomy of intrusion risk assessment and response system. Computers & Security 45, 1–16 (2014)
29. Somorovsky, J., Heiderich, M., Jensen, M., Schwenk, J., Gruschka, N., Iacono, L.L.: All your clouds are belong to us: security analysis of cloud management interfaces. In: Proceedings of the 3rd ACM Cloud Computing Security Workshop (CCSW 2011). pp. 3–14 (2011)
30. Toeroe, M., Pawar, N., Khendek, F.: Managing application level elasticity and availability. In: 10th International Conference on Network and Service Management. pp. 348–351 (2014)
31. Watson, P., Little, M.: Multi-level security for deploying distributed applications on clouds, devices and things. In: IEEE 6th International Conference on Cloud Computing Technology and Science. pp. 380–385 (2014)
32. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-VM side channels and their use to extract private keys. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. pp. 305–316. ACM (2012)