International Journal of Information Security (2023) 22:1713–1729
https://doi.org/10.1007/s10207-023-00713-y

REGULAR CONTRIBUTION

Yet another cybersecurity risk assessment framework

Mathias Ekstedt[1] · Zeeshan Afzal[1] · Preetam Mukherjee[1,2] · Simon Hacks[3] · Robert Lagerström[1]

Published online: 14 July 2023
© The Author(s) 2023

Abstract
IT systems pervade our society more and more, and we become heavily dependent on them. At the same time, these systems are increasingly targeted in cyberattacks, making us vulnerable. Enterprise and cybersecurity responsibles face the problem of defining techniques that raise the level of security. They need to decide which mechanism provides the most efficient defense with limited resources. Basically, the risks need to be assessed to determine the best cost-to-benefit ratio. One way to achieve this is through threat modeling; however, threat modeling is not commonly used in the enterprise IT risk domain. Furthermore, the existing threat modeling methods have shortcomings. This paper introduces a metamodel-based approach named Yet Another Cybersecurity Risk Assessment Framework (Yacraf). Yacraf aims to enable comprehensive risk assessment for organizations with more decision support. The paper includes a risk calculation formalization and also an example showing how an organization can use and benefit from Yacraf.

Keywords Threat modeling · Enterprise IT risk · Risk assessment · Attack tree

1 Introduction

With the increased pervasiveness and complexity of the IT infrastructure induced by the digitization, the importance of cybersecurity management also increases. From an enterprise management point of view, the cybersecurity responsibility has landed on its own role, the chief information security officer (CISO), and largely falls under the challenge of risk management. Bottom line, the CISO must determine what security controls should be applied in an IT infrastructure so that business risks and costs are minimized at the same time. At her disposal, the CISO has a multitude of methods, frameworks, and standards to support the work, but an integration of these to achieve an overall assessment is still missing, leaving practitioners with the task of managing heterogeneous sources and analyses. Risk management in general (not only focusing on cyberrisk) is a well-established field with numerous application areas and methods, mature enough to have its own ISO/IEC standards on risk analysis [13] as well as on information security management [15]. Cybersecurity risk management has become increasingly important in the more general strive of business and IT alignment. For instance, The Open Group, which developed The Open Group Architecture Framework (TOGAF) to target the many facets of business and IT alignment challenge, released a report for risk analysis[1] based on the FAIR approach [10].

In parallel to the developments in the enterprise cybersecurity management domain, the software engineering community has witnessed a corresponding increase in attention on the topic of security, resulting in the emergence of a community around the concept of threat modeling. This movement is perhaps most clearly represented by Microsoft's work on developing the secure development life cycle (SDL) in its organization during the early 2000 s [2] in combination with Shostack's book [38], which describes the STRIDE method.

Even though these two communities seem to have separate histories, they are in many respects similar and seem to move closer to each other over time, for instance with movements such as Dev(Sec)Ops .[3] A key difference is that threat modeling is focused on identifying vulnerabilities in soft-

✉ Mathias Ekstedt
mekstedt@kth.se

✉ Zeeshan Afzal
zafzal@kth.se

[1] KTH Royal Institute of Technology, Stockholm, Sweden

[2] Digital University Kerala, Thiruvananthapuram, India

[3] Stockholm University, Stockholm, Sweden

[1] https://publications.opengroup.org/c13g.

[2] https://www.microsoft.com/en-us/securityengineering/sdl.

[3] https://www.devsecops.org/.

ware system designs so that they can be eradicated as part of the development process, while the enterprise security risk management community additionally is interested in understanding the business consequence of vulnerabilities being exploited thus framing the interest as a risk. Threat modeling is clearly a model-based approach and, as indicated, at least parts of the enterprise security risk management are too. Interestingly, the combination and synergy of these two fields of threat modeling and enterprise risk management are still relatively unexplored, and in this work we aim to take a step to bridge this gap.

In particular, our **goal** with this work is to merge two strongholds from the two communities: the model-based security analysis from the threat modeling community and the quantitative risk assessment calculations from the risk management community. We set the following concrete **objectives** for this work.

1. to propose a metamodel for risk-based threat modeling.
2. to provide a risk calculation framework.

The most prominent work with a somewhat similar agenda is found in the method Process for Attack Simulation & Threat Analysis (PASTA) [25]. The presented approach is named Yet Another Cybersecurity Risk Assessment Framework (Yacraf) and the **novelty** in this approach is that we combine a metamodel with tailored logic for risk assessment calculations into a unified framework. This enables us to take the structure and architecture of IT systems and their context into account in the risk assessment. For example, the overall risk of having a highly severe software vulnerability in the system depends on its location; is it residing in a machine facing the internet or in a machine deep in the network hierarchy; is it an end target of a cyberattack that can cause negative business impact; or is it merely a stepping stone toward some other end goal. To simply base, a risk assessment on the patch level of machines misses this key condition.

In general, our approach adheres to the common view that *risk* is a function of *threat*, *vulnerability*, and *impact*, found for instance in FAIR [10] and PASTA [25]. However, in other details, our Yacraf metamodel differs. The presented metamodel provides transparency in how to argue around the value of different parameters in the risk assessment equation. It is still difficult to make assessments on all the individual parameters in the equation, but with a model supporting the assessment we will be able to be more clear and explicit around many complex phenomena related to the risk score. At the same time, the provided risk calculation framework supports consistency to the risk calculations by stipulating a certain way of doing the risk calculation given a certain model. In terms of delimitation, our work is focused on malicious cyberthreats, meaning that we assume an active involvement of some threat actor (an attacker) as well as an attack vector traversing IT systems. Furthermore, Yacraf is only focused on supporting cyberrisk assessment; the larger challenge of risk management is not covered.

The rest of the article is structured as follows: Next, we cover related work in terms of threat modeling frameworks and evaluate them with respect to our scope of model-based cybersecurity risk assessment. Afterward, we describe our main artifact, the Yacraf metamodel and its integrated risk assessment framework. Next we present an illustrative case of how the metamodel is instantiated and used to derive quantitative risk assessments. Afterward, we provide a discussion and some details on practical experiences of using Yacraf in real-life organizations. The article ends with a conclusion section.

2 Related work

This section presents the related work. As the goal of this work is to integrate model-based security analysis from the threat modeling community with quantitative risk assessment calculations from risk management, the section begins with an overview of various threat modeling and risk assessment methods currently available. Later, in Sect. 2.2, we will compare these different approaches, in order to motivate the need for our approach which is then detailed in Sect. 2.3.

2.1 Threat modeling and risk assessment methods

A multitude of methods exist for conducting threat modeling and risk assessment and management [9]. STRIDE [38] considers possible threats while a product or system is under development. The method involves creating a model of the system using data flow diagrams (DFDs) and then considers different threats that can impact each part of the model. The threats are generally known and relate to the method name, STRIDE, which stands for Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. DREAD [37] is a modified STRIDE approach developed by Microsoft to evaluate threats. It refers to five categories; Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability. It proposes a different method for threat assessment where values are assigned to different categories, allowing for an average value to be calculated to represent the overall risk. Another threat modeling framework is LINDDUN [8], which is designed to find privacy related design flaws.

Other approaches from the software engineering community are focused on model-based security analysis. These kinds of methods can further be grouped into two, i.e., those who support automatic analysis and those who do not. The first group includes approaches such as UMLsec [18, 19], SecureUML [3, 4], SECTET [1, 12], or STS-ml [28]. These approaches allow to specify a system as a set of components and interactions between them. Further, security properties such as constraints, requirements, or threats are added to the models, which are used to analyze the system automatically to enable formal reasoning and make deductions about its security. The second group of approaches does not allow an automated analysis. Examples for these are CORAS [21], secureTROPOS [26], and SecDSVL [2].

Another group of model-driven security approaches rely on attack trees and/or graphs. These techniques help visualize and model different attack steps (and their dependencies) taken by an attacker to compromise a system. The concept of attack trees was established by Schneier [33, 34], formalized by Mauw and Oostdijk [23], and extended to include defenses by Kordy et al. [20]. To avoid building new attack graphs for each system, domain-specific attack languages such as the Meta Attack Language (MAL) [17] may be employed. A wide range of tools [27] have been developed based on attack trees and graphs. The work in [30] makes an attempt to combine attack trees with STRIDE in an attempt to quantify threats. For all five categories of STRIDE threats, attack trees are developed. CVSS [4] scoring method is employed to introduce the risk value for the vulnerabilities. The method also allows for quantifying the overall system-wide risk resulting from the component attack trees and their combination. In general, assessment methods based on attack trees do not always consider the business consequence or risk arising from potential breaches. In addition, such approaches normally do not relate attack vectors with the system architecture (except for MAL) and often base their quantification on the vulnerability severity alone with no direct consideration of threat capability.

For the enterprise risk management domain, FAIR [10] provides a set of tools for understanding, measuring, and analyzing information risks to support managers to make better business decisions by understanding their organizational risk. It covers areas such as risk theory, risk calculation, scenario modeling, and communicating risk within the organization. It provides a comprehensive risk quantification approach combining estimates of threat actors, vulnerabilities, and incident impact. [42].

PASTA [25] is a risk-centric threat modeling framework that was designed for IT, security, compliance, and risk leaders who want to mitigate risks and to support them in understanding the causal threat factors. To achieve this, PASTA presents a step-by-step approach that focuses on business impact, threat research, and countermeasures that reduce risks. These steps are performed in an iterative process and can be aligned to different frameworks.

TRIKE [32] also approaches threat modeling from a risk management perspective. It begins with the definition of the system by enumerating and understanding the system elements such as actors, resources, intended actions, and rules, as well as building a DFD. This allows the analyst to identify threats that fall into one of two categories: elevation of privilege or denial of service. To assess the risk of attacks that may impact business objects, TRIKE uses a point-based scale for each action, based on its probability.

Octave [6] focuses on information object containers and identifies areas of concern and threat scenarios on those containers in the context of an organization. The probability (in ordinal scale) of threat occurrence and its consequence determine the risk. In Octave, identification of "means" by which a threat actor exploits areas of concern or threats is questionnaire based. Overall, this method uses worksheet-based simple calculation for quick assessment of risk.

Lastly, there exist a wide range of general risk assessment methods. ISO 31000 [13] provides guidelines and principles on risk management. ISO 27005 [16], NIST 800-30 [31] discuss information security risk assessment and management. MONARC is inspired by ISO/IEC 27005 and offers a qualitative methodology for risk analysis [22]. Some other useful risk assessment methods include CRAMM [40] and FRAAP [29]. A more hands on frameworks is presented by the Committee of Sponsoring Organizations of the Treadway Commission (COSO): the Enterprise Risk Management-Integrated Framework [7].

2.2 Comparison of methods

As discussed before, there exist a number of threat modeling and risk assessment methods. The decision of which method to use for a particular organization or system is a complex task that is further complicated by the fact that there is no perfect method [36]. Different methods are designed with different points of view and often address different goals. Some methods are focused purely on design or software level, while others on IT assets or business objects, attackers, risks, and their impacts. The methods also differ in the level of detail. Consequently, some methods are better suited for a particular type of threat and system and are thus less effective for others.

To motivate the need for another model-based risk assessment framework, as proposed in this paper, we compare a number of existing methods. It is not possible to cover all possible methods; hence, the scope is set to the most well known methods and the list of such methods is inspired by previous works [35, 39, 45]. In total, seven approaches are considered. These are compared with each other using three different indicators, as discussed below. It should be noted that the presented comparison is merely a collation of related work and not a systematic literature review.

The first indicator is *scope* and is inspired by previous work in [36]. By *scope*, we mean the perspective or point of

[4] https://nvd.nist.gov/vuln-metrics/cvss.

view for which the method was developed. *Scope* is further defined by two subindicators, namely; *approach* and *goal*. *Approach* refers to the focus of a method, as different methods might have unique strategies. Some are focused on the design stage or the architecture of a system, while others on modeling IT assets or business objects, attackers, and the consequences. The method's *goal* could also differ and be of importance. Table 1 compares the different methods based on this indicator.

The second indicator is the *level of detail* (completeness), or the depth covered by a framework. As mentioned earlier, *risk* is commonly considered to be a function of *threat*, *vulnerability*, and *impact*. Therefore, a comprehensive risk analysis framework should include all three conceptual domains. However, different methods apply modeling at different granularity and stages. Some methods only focus on vulnerabilities and perform modeling of either the business objects and/or the IT assets, while others consider the threat actor and the impact too. By asset modeling, in this context and the rest of the paper, we mean identifying and modeling (all) the IT assets of the organization's infrastructure and their internal communications to identify vulnerabilities and attack surfaces. Another group of methods model threat actors by representing potential attackers to the organization and modeling their abilities and possible attack vectors. Finally, potential loss from possible attacks in a specific organizational scenario is estimated by some methods as part of impact or consequence modeling.

The third and final indicator is the *type of assessment* employed by a method. Some methods allow numeric and quantitative risk assessment, while others only enable a qualitative assessment. It should be noted that a quantitative risk assessment can still include some underlying qualitative input parameters. Another important differentiation between different models is whether they provide an explicit and consistent metamodel for cyberrisk assessment. Table 2 shows the comparison of different methods based on indicator two and three and their relevant subindicators.

2.3 Analysis and motivation

From the comparison in the tables above, we can see that risk analysis and threat modeling methods differ in scope, the aspects of risk they cover, and the assessment type. Furthermore, the methods also differ in the techniques they use for modeling different aspects of risk. STRIDE [38] provides a good means for identifying software-based security threats during product development by employing DFDs. However, it provides no means for evaluating those threats and the eventual risk. DREAD [37] can be used together with STRIDE to provide a way to evaluate the identified threats using a score for each threat, but the approach lacks detailed assessment and impact modeling.

Octave [6] focuses on organizational and business objects and starts with creating object profiles using worksheets. It identifies and assesses organizational threats and the resulting operational risk by using threat trees rather than focusing on specific systems. Thus, the technological risks are not considered in this method. The risk impact is measured with respect to different risk measurement criteria, viz. Reputation/customer confidence, Financial, Productivity, Safety and health, and Fines/legal penalties. However, not much attention is given to the capability of threat actors.

CORAS [21] enables risk analysis by performing modeling using the Unified Modeling Language (UML) to model organizational assets, their dependencies, and connections. It uses threat diagrams to model the threats, vulnerabilities, threat scenarios, and their relations along with the unwanted incidents and assets. Furthermore, brainstorming is used to estimate the consequence and impact of the identified unwanted incidents which makes the method manual and does not allow for an automated analysis. Lastly, this method does not give much consideration to the threat actors capability and the effort they might spend. Although CORAS provides a metamodel for the risk modeling language syntax, it falls short in describing how to derive the risk assessment from the model. ISMS-CORAS [5] is an extension to CORAS method that enriches the threat actor analysis provided by CORAS and provides diagrams and templates to support documentation requirements of the ISO 27001 standard. It enables classification of attacker types, templates for attacker description, and attacker overview diagrams to facilitate the attacker identification. There is also support for identification of attacker motivation and entry points, and for modeling this information in the threat diagrams. ISMS-CORAS provides a metamodel and enables a detailed threat analysis, but the focus and aim of the method are actually to establish an ISO 27001 [14] compliant Information Security Management System (ISMS).

FAIR [10] puts its main focus on assets at risk, although it does not specify any technique to use to perform the modeling. For evaluating threats, classes of threat communities are listed in FAIR e.g., nations, insiders, cybercriminals, and malware. Characteristics of these different threat communities, viz. motivation, sponsorship, capabilities, concerns for collateral damage, are also listed. To measure the impact of risk, FAIR identifies six types of loss forms which are classified as direct or indirect impact. Overall, FAIR provides a comprehensive risk quantification approach and is mostly complete for risk estimation. However, this method is weaker on the specific support for cybersecurity and IT systems. Furthermore, it provides neither means for overall risk modeling nor risk treatment [42].

PASTA [25] covers all aspects of risk as shown in Table 2. In its step-by-step approach, PASTA uses DFDs for modeling assets. Furthermore, attack trees include threats, abuse

Table 1 Comparison according to indicator 1—*Scope*

Framework	Reference	Approach	Goal
STRIDE	[38]	Software-centric	Identify threats
DREAD	[37]	Software-centric	Evaluate threats
OCTAVE	[6]	Asset-centric	Organizational risk
ISMS-/CORAS	[5, 21]	Asset-centric	Risk assessment
FAIR	[10]	Asset-centric	Risk assessment
PASTA	[25]	Risk-centric	Risk assessment
TRIKE	[32]	Asset-centric	Risk assessment

Table 2 Comparison according to indicator 2 and 3—*Level of detail* and *type of assessment*

Method	Business object modeling	Asset modeling	Threat actor modeling	Impact modeling	Metamodel	Quantitative	Qualitative
STRIDE		✗					
DREAD		✗					✗
OCTAVE	✗		✗	✗			✗
ISMS-/CORAS	✗	✗	✗	✗	✗	✗	✗
FAIR		✗	✗	✗		✗	
PASTA		✗	✗	✗		✗	✗
TRIKE		✗	✗	✗			

cases, and attack patterns and are used to evaluate and model different threat scenarios. Within an attack tree, threats are mapped to the use cases identified in asset modeling. Finally, to measure and model the impact of risk, attack patterns are mapped to different possible impacts. In PASTA, the terminology is not always consistent and concise. Further, it does not incorporate different attack behaviors and their different goals. Lastly, it focuses mostly on the process and falls short on the modeling itself [39].

TRIKE [32] suggests the use of DFDs as part of business modeling and considers threats in two possible categories. It also has some sort of impact consideration by using a point scale based on probability. However, the method lacks the details required of a quantitative risk assessment method and is not well documented.

Concluding from our analysis, no existing threat modeling or risk assessment method is flawless. Among the considered methods, CORAS or ISMS-CORAS, FAIR, and PASTA stand out as they cover all aspects of risk and enable a quantitative risk assessment. Moreover, CORAS is the only method that provides a metamodel although that model is limited to the language syntax. However, even these methods fall short in providing a modeled approach and lack details about threat actors and considered systems. ISMS-CORAS makes the CORAS threat analysis more detailed but focuses more on establishing and documenting an ISO 27001 compliant ISMS. To summarize, to the best of our knowledge, no existing quantitative risk assessment method provides a modeled approach that is consistent and comprehensive in terms of covering all aspects of risk with sufficient details and considering system architecture.

This motivates the need for another approach, as described in this paper. This approach named Yet Another Cybersecurity Risk Assessment Framework (Yacraf) will allow to enrich the enterprise IT risk domain by the means of threat modeling. Yacraf leverages the benefits and combines the two domains of model-based security analysis and quantitative risk assessment. Furthermore, it goes one step further and integrates a model-based quantitative risk assessment in an explicit metamodel that provides more decision support than any other existing method. The result is also expected to be more realistic as the risk assessment provides additional resolution by considering the structure and architecture of IT systems and their contexts. Yacraf is comprehensive and is intended to be used by end-user IT organizations as a tool in their cyber risk analysis. To put Yacraf into context, we classify it according to the three presented indicators (*Scope*, *Level of detail*, and *Type of assessment*) above. Yacraf's goal is to provide a quantitative risk assessment with an approach that is focused on different IT components (assets). Moreover, it covers all three conceptual domains and enables modeling for assets, threats, and the impact. Finally, Yacraf provides a metamodel and a quantitative assessment.

3 The framework

In this section, we present our proposed framework. It consists of a metamodel that defines what needs to be modeled in order to perform cybersecurity risk assessment, as well as a formalism for deriving the risk assessment from a model instance.

3.1 Metamodel

The metamodel, presented in Fig. 1, describes classes, their associations, and class attributes. It is conceptually divided into the three risk assessment domains: *vulnerability*, *threat*, and *impact*.

3.1.1 Vulnerability

The `Asset` is the central element of our metamodel. The `Asset` is used as an abstract class representing any kind of IT related component. The idea is then to specialize `Assets` into any class that make sense for the system domain at hand. Since this will vary, we here introduce exemplary classes that have been inspired by (but not identical to) DFDs often found in the threat modeling community. Moreover, when `Assets` are refined, additional class associations are (normally) added. In our metamodel, we do not elaborate on all possible associations we could or would like to include for different subassets. Instead, we introduce an `Asset` self-association as a place holder for all of these. Our ambition is thus to be flexible with the exact design of the metamodel for system modeling, because this will vary in practice. The only strict requirement we introduce is that systems are represented by some sort of `Asset` and that these `Assets` can be associated in various ways. As we will see later, we can afford this flexibility as the exact system meta model does not impact the security assessment.

In order to be able to assess the vulnerability dimension of the risk assessment, we provide the class `Vulnerability`, which is related to `Asset`. A `Vulnerability` can be exploited, which we describe by an `Attack event`, but also protected with `Defense mechanisms`. With `Attack events` and `Defense mechanisms`, we also introduce the notion of attack trees [33] and attack-defense trees [20]. Generally, a `Vulnerability` can be understood as a composition of `Attack events` and `Defense mechanisms`. In turn, this also means that the relation between `Vulnerability` and `Asset` is a derived relationship depending on the `Attack event` and `Defense mechanism` relationships to the `Asset`.

3.1.2 Threat

`Attack events` are executed by an `Attacker`. To express the planning behind attacks, we facilitate `Abuse cases`. Analogous to use cases, an abuse case is a set of actions representing some complete system interaction, but as opposed to use cases with a malicious intent [24]. In our metamodel, these actions are the attack events; thus, an `Abuse case` is a composition of a number of `Attack events`. As the `Attack events` are ordered in graphs, some of them would constitute the attack surface (the `Attack events` without parents) and there would be at least one end goal (the `Attack events` without children). Accordingly, an `Abuse case` has a derived relation to `Assets`, as `Attack events` are also targeting them. Finally, an `Abuse case` is related to exactly one `Attacker`, and these two classes jointly represent antagonists and their expected behavior, i.e., the threat.

3.1.3 Impact

To understand the impact of a risk, the concept of `Loss events` is crucial. In general, the discussion of causes and consequences within the risk analysis field is diverse. Here, we approach this distinction with the assumption that `Attack events` are happening to the IT or cyberdomain and `Loss events` relate to the business or physical context that the cyber `Asset` is connected to. An `Attack event` is thus causing a `Loss event`. For instance, the confidentiality breach of some customer records is an attack that lacks any inherent consequence; this is instead captured as a `Loss event` such as regulatory fines, lost reputation, or customer privacy exposure.

This example raises also the demand for different `Actors`. Our metamodel prescribes that every `Loss event` must be related to exactly one `Actor`. In other words, for every loss there must be someone who suffers from it. Similar to `Abuse cases`, `Loss events` have a derived relation to `Asset` as `Attack events` also are targeting them.

With this everything is now covered that is strictly needed to support the risk assessment. However, the challenge of identifying `Loss events` is tightly related to understanding the business or physical process supported by the IT systems. This is the topic of business and process modeling and analysis. For these activities, there exist a large and wide variety of methods and models ranging from general-purpose approaches such as Business Process Model and Notation (BPMN)[5] to models designed to simulate properties of physical systems[6] to simulation tools developed for specific domains, such as power grids.[7] As these modeling approaches vary greatly, similar to the system domain discussed above, we opted to introduce only an example skeleton for business modeling consisting of `Use cases` and `Business goals`. With these classes, it is possible to represent how the user interacts with IT systems, as is com-

[5] https://www.omg.org/spec/BPMN/2.0/.

[6] E.g., the OpenModelica community, https://www.openmodelica.org.

[7] E.g., tools from OPAL-RT https://www.opal-rt.com/power-systems-overview/.

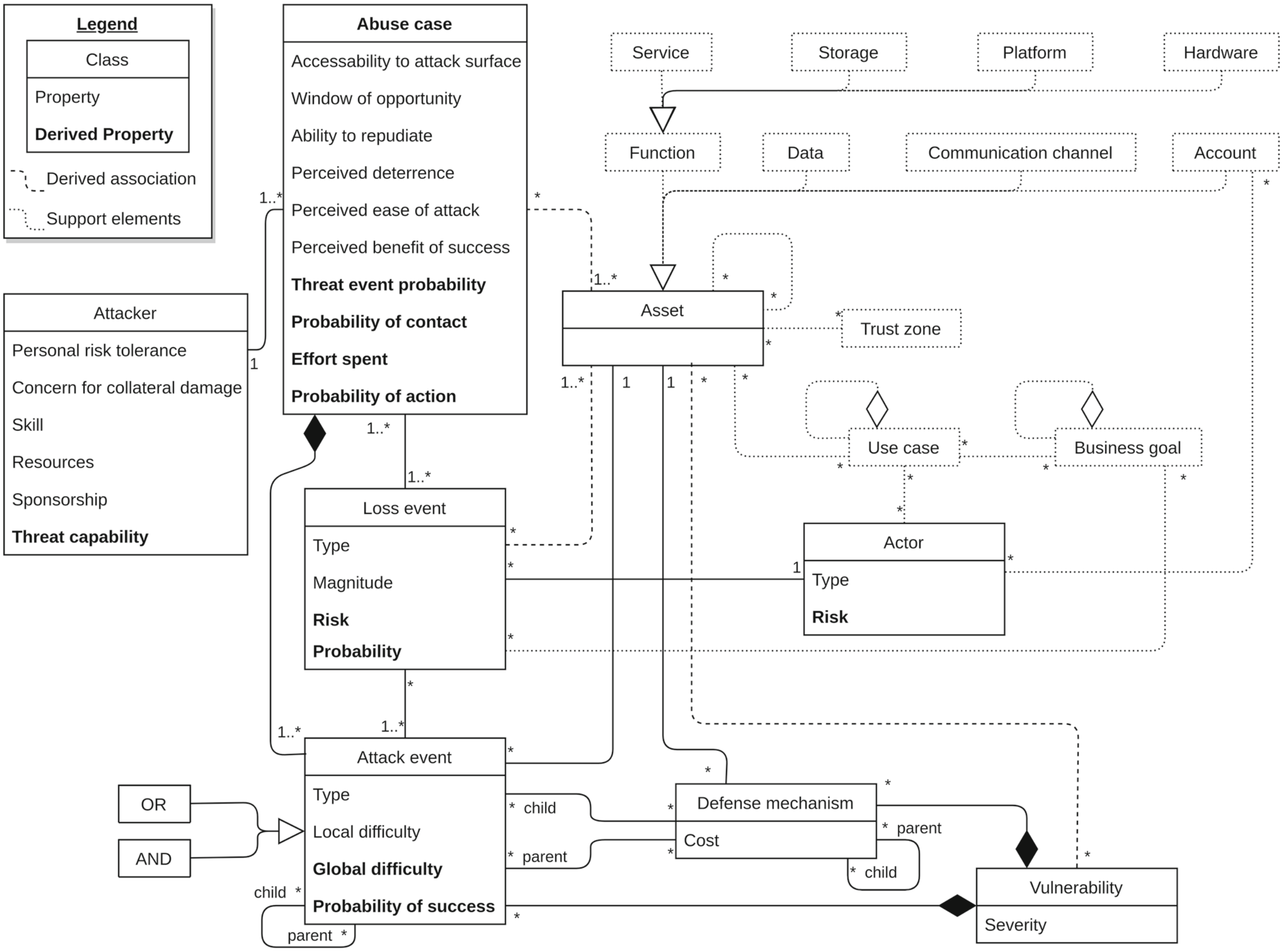

Fig. 1 Yacraf metamodel

mon in software modeling, and for what purposes. We add a relation between Business goal and Loss event to emphasize the fact that the latter typically violate the former to be considered as a loss. Similar to the system modeling part of the metamodel, we note that the exact structure of the business and process metamodel does not impact the risk assessment per se; it is merely a mechanism to help identifying and structuring Loss events.[8]

3.2 Risk assessment calculations

Thus far, we have presented classes and their relations of the meta model. In Fig. 1, there are also a number of class attributes that enable the risk assessment along the categories of vulnerability, threat, and impact. However, our suggested risk assessment calculation method is more complicated than multiplying these three values. At large, it follows the risk assessment calculation method as presented in FAIR [10]. We have reused the level of abstraction for the risk assessment as well as the assumptions about what type of input is needed for the assessment. Even though the method is driven by a quantitative risk paradigm where the core is constituted of mathematically formalized equations, we also acknowledge that the input values are qualitatively estimated or assumed forming a hybrid approach as a whole.

We also share FAIR's probabilistic approach to risk assessment to capture the often overlooked but important dimension of uncertainty. However, a few things have been modified and added in relation to FAIR: 1) We have altered attributes framed as frequencies into probabilities to simplify the calculations, similar to [41]. 2) We have removed second order impacts for simplification as our focus is on cyber. Instead, we do not limit the impact reasoning only to intra organizational phenomena. 3) Most importantly, we have expanded and nuanced the assessment of vulnerabilities to encompass a more elaborate assessment of system architectures assisted

[8] Of course, this is only true for the abstraction level we have chosen for the metamodel. With a more refined metamodel, we could for instance imagine that we derived the Loss event magnitude from the business model, for instance, with simulation tools as the aforementioned.

by attack graphs. Doing so, the risk calculations fit onto an explicit metamodel and, thus, constitute an important contribution of our work.

Figure 2 provides a structural overview of how the attributes found in the metamodel in Fig. 1 are aggregated into an overall risk value. Next, we will describe all the attributes and then continue with a formal description of our suggested risk assessment calculation.

Before going into details of the risk assessment's calculation, we provide an overview. Beginning at the top of Fig. 2, *Risk* is associated with both `Actor` and `Loss event`. As indicated earlier, we consider *Risk* fundamentally belonging to `Loss events`, but it can also be aggregated to a total value or exposure to individual `Actors`. On `Loss event`, we also assign *Probability* and *Magnitude*. The `Loss event` probability depends on the *Threat event probability (TEP)*, i.e., that the attack is attempted, a property placed on the `Abuse case` and the *Probability of success (PoS)* of an `Attack event`. The TEP in turn depends on the `Abuse case` properties *Probability of Action (PoA)* and *Probability of Contact (PoC)*. The PoS depends on the one hand of how difficult it is to succeed with the full attack vector, encoded in the property *Global difficulty* found on `Attack event`, and the other hand on the expected *Effort spent* on the `Abuse case` (composing that `Attack event`). Next, we describe the calculations bottom-up.

The level of risk an entity is willing to assume in order to achieve a potential desired result.

PoA represents the probability that an attacker will perform the attacks in an abuse case. To quantify this value is clearly challenging and is dependent on a great number of properties. We have chose to reuse factors mentioned in FAIR for threat profiling as a representative set of independent causal factors for this. These are firstly *Risk Tolerance (RT)* and the *Concern for Collateral Damage (CfCD)* of the `Attacker`. RT is about how concerned attackers are about getting caught. CfCD is used to measure the tolerance for damaging unintended targets. Of course, the dependency is only present for the particular `Attacker` associated with the `Abuse case` (recall that our metamodel prescribes a multiplicity of exactly one `Attacker` per `Abuse case`). Secondly, the PoA depends on four properties of the `Abuse case` itself: *Ability to Repudiate (AtR)*, *Perceived Deterrence (PD)*, *Perceived Ease of Attack (PEoA)*, and *Perceived Benefit of Success (PBoS)*. By AtR, we mean if it is possible to deny an attack. PD represents the expected consequences of getting caught. PEoA is a measure of how easy it is for the attack to happen. PBoS is used to quantify the expected benefit of a successful attack.

We consider the PoA to be a probability distribution that represents the uncertainties of the assessed scenario, and as indicated in Fig. 2 the value of PoA needs to be estimated qualitatively based on the underlying factors. More formally then, the set of all the abuse cases is represented as *AbuseCase* and set of all the attackers is represented as *Attacker*.

Next, as an `Abuse case` is related to one `Attacker` only, $executedBy : AbuseCase \rightarrow Attacker$ is the mapping representing the `Attacker` related to an `Abuse case`.

For an $abuseCase \in AbuseCase$,

$$\begin{aligned} &PoA_{abuseCase} = f_{PoA}(RT_{attacker},\\ &\quad CfCD_{attacker}, AtR_{abuseCase},\\ &\quad PD_{abuseCase}, PEoA_{abuseCase}, PBoS_{abuseCase}) :\\ &\quad executedBy(abuseCase) = attacker \end{aligned}$$

f_{PoA} is the function aggregating all the component factors to compute PoA.

Probability of Contact (PoC) on the other hand is the probability that the attacker has access to the attack surface related to the `Abuse Case` (i.e., its `Attack events` without parents). This can be estimated from the `Abuse Case`'s specific *Window of Opportunity (WoO)* and *Accessibility to Attack Surface (AtAS)*. We also assume that the AtAS also will depend on the `Attacker`'s *Threat Capability (TC)*. Again the PoC is probability distribution qualitatively estimated from these underlying causal factors.

$$PoC_{abuseCase} = f_{PoC}(WoO_{abuseCase}, AtAS_{abuseCase})$$

The *Threat Event Probability (TEP)* of an `Abuse Case` is determined by the PoA and PoC of the same `Abuse Case`.

$$TEP_{abuseCase} = PoA_{abuseCase} \times PoC_{abuseCase}$$

`Attack events` exploit `Vulnerability`(ies) in `Assets`, and `Defense mechanisms` are put in place to protect against such exploitation. The set of all possible `Attack events` is represented as *AttackEvent* and the set of all the `Defense mechanisms` is represented as *Defense*. Existence of `Defense mechanism`(s) increases the level of protection of the `Asset` by increasing the difficulty for an `Attacker`. Possible `Defense mechanism`(s) against an `Attack event` can be identified as $parent\_def : AttackEvent \rightarrow 2^{Defense}$ to represent the mapping of a `Attack event` to a set of `Defense mechanism`(s). $2^{Defense}$ represents the powerset of `Defense mechanisms`. $active : Defense \rightarrow \{true, false\}$ is a function for identifying the set of existing activated `Defense mechanism`(s). One `Defense mechanism` may depend on other `Defense mechanism`(s) for its existence; this can be formally

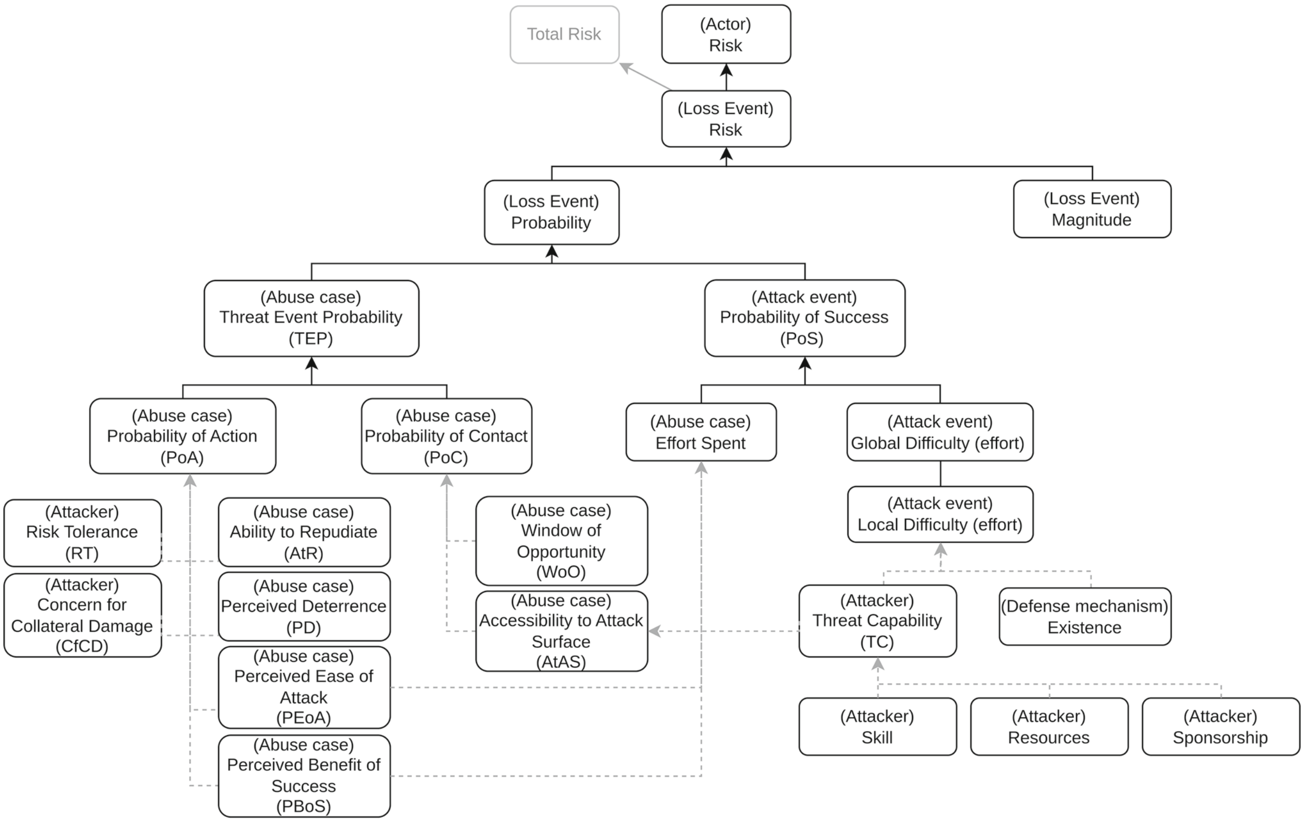
Total Risk
(Actor) Risk
(Loss Event) Risk
(Loss Event) Probability
(Loss Event) Magnitude
(Abuse case) Threat Event Probability (TEP)
(Attack event) Probability of Success (PoS)
(Abuse case) Probability of Action (PoA)
(Abuse case) Probability of Contact (PoC)
(Abuse case) Effort Spent
(Attack event) Global Difficulty (effort)
(Attack event) Local Difficulty (effort)
(Attacker) Risk Tolerance (RT)
(Attacker) Concern for Collateral Damage (CfCD)
(Abuse case) Ability to Repudiate (AtR)
(Abuse case) Perceived Deterrence (PD)
(Abuse case) Perceived Ease of Attack (PEoA)
(Abuse case) Perceived Benefit of Success (PBoS)
(Abuse case) Window of Opportunity (WoO)
(Abuse case) Accessibility to Attack Surface (AtAS)
(Attacker) Threat Capability (TC)
(Defense mechanism) Existence
(Attacker) Skill
(Attacker) Resources
(Attacker) Sponsorship

Fig. 2 Risk assessment calculation overview. (Dashed lines indicates qualitative aggregations and solid lines indicates quantitative aggregations.)

represented as $parent\_def : Defense \rightarrow 2^{Defense}$. Therefore, $parent\_def$ maps $(AttackEvent \cup Defense) \rightarrow 2^{Defense}$.[9]

Abuse cases, which are compositions of various Attack events, initiates from the *attack surface* and ends at *target* Attack event(s). $constituent : AbuseCase \rightarrow 2^{AttackEvent}$ maps an Abuse case to the set of constituent Attack event(s). In the context of an Abuse case, an Attack event can have parent Attack event(s). For a particular Attack event in an Abuse case, the set of parent Attack events can be represented as $parent\_att : constituent(abuseCase) \rightarrow 2^{constituent(abuseCase)}$. Similar to the $parent\_def$, $parent\_att$ can also be used to map $Defense \rightarrow 2^{AttackEvent}$. It will represent the Attack event(s) which can exploit Vulnerability(ies) in a Defense mechanism. Therefore, $parent\_att : (AttackEvent \cup Defense) \rightarrow 2^{AttackEvent}$.

Abuse cases are created for executing one or more target Attack event(s) which will generate loss for the company. Other Attack events, which are part of the Abuse case, are helping the Attacker to execute the target Attack event(s). $targetAtt : AbuseCase \rightarrow 2^{constituent(abuseCase)}$ represents the mapping between an Abuse Case and one or more target Attack event(s) where, $abuseCase \in AbuseCase$. Target Attack event(s) have no child Attack event(s) therefore, for any $attackEvent \in constituent(abuseCase)$, $parent\_att(attackEvent) \notin targetAtt(abuseCase)$.

Effort required by an Attacker to successfully execute an Attack event, irrespective of the prerequisite Attack events in the Abuse case, we have termed *Local difficulty* for the Attack event, whereas *Global difficulty* for the Attack event is the required effort from the Attacker to successfully execute the Attack event along with all the prerequisite Attack events within the context of the particular Abuse case. The calculation of *Global difficulty* given here is similar to the standard calculation for attack graphs or attack trees [17].

Local difficulty for an Attack event depends on Attackers *Threat capability (TC)*. The value of TC

[9] We note here that the Trust zone does not really play an active role in the risk assessment per se. Similar to the Vulnerability, we want to represent an Asset's potential misplacement in an inappropriate Trust zone in terms of Attack events it enables. From an attack-centric world view, a Trust zone then represents an area where we expect low effort to move around for the Attacker, and we would expect a high effort to cross a trust boundary. The Trust zone thus represents more of a requirement than an inherent system property that can be used for the risk assessment.

depends on factors such as *Skill*, *Resources*, and *Sponsorship* of the Attacker.

$$TC_{attacker} = f_{TC}(Skill_{attacker}, Resources_{attacker}, Sponsorship_{attacker})$$

[10]

TC of an Attacker and the existence of Defense mechanism(s) can be combined together to compute the *Local difficulty* associated with an Attack event for the Attacker in the context of a particular Abuse case.

$$\begin{aligned} LocalDifficulty_{attackEvent} &= f_{LD}(active(defense), \\ &\quad TC_{attacker}) \\ &\quad where, defense \in parent\_def(attackEvent), \\ &\quad executedBy(abuseCase) = attacker, \\ &\quad attackEvent \in constituent(abuseCase) \end{aligned}$$

After computing the *Local difficulty* for an Attack event, one can compute its *Global difficulty* by considering all the prerequisite Attack events to be executed by the Attacker in the Abuse case. Therefore, computation of *Global difficulty* for the parent Attack events (immediate prerequisites) is necessary to compute *Global difficulty* for a child Attack event.

The Attack event can be of two different types, *AND* or *OR*. An Attack event of type *AND* needs all of its parent Attack events to be executed before its execution, whereas an Attack event of type *OR* needs at least one of its parents to be executed before its execution. Type of an Attack event can be formally represented as $type : attackEvent \rightarrow \{AND, OR\}$.

The computation of *Global difficulty* to execute *attackEvent,* i.e., $GlobalDifficulty_{attackEvent}$ for an Attacker in the context of an Abuse case, is presented in Algorithm 1.

Other than *Global difficulty*, *Expected effort spent* by an Attacker to execute an Abuse case is a factor for determining the *Probability of success (PoS)*. We assume *Expected effort spent* on an Abuse case by an Attacker to be determined from the Abuse case specific *Perceived Ease of Attack (PEoA)* and *Perceived Benefit of Success (PBoS)*. Functional composition of these qualitative factors can be represented as

$$ExpectedEffortSpent_{abuseCase} = f_{Effort}(PEoA_{abuseCase}, PBoS_{abuseCase})$$

Algorithm 1 Global difficulty calculation

```
Require: attackEvent: Attack event for which Global Difficulty have to be determined, Abuse case: Global difficulty will be determined in the context of a particular Abuse case, $attackEvent \in AttackEvent_{abuseCase}$, Local difficulty values for Attack events
Ensure: Global difficulty for attackEvent
GlobalDifficulty(attackEvent)
if $parent\_att(attackEvent) \neq \emptyset$ then
    if $type(attackEvent) = OR$ then
        $GlobalDifficulty_{attackEvent} \leftarrow Min(GlobalDifficulty(parent\_def(attackEvent))) + LocalDifficulty_{attackEvent}$
    else if $type(attackEvent) = AND$ then
        $GlobalDifficulty_{attackEvent} \leftarrow \sum(GlobalDifficulty(parent\_def(attackEvent))) + LocalDifficulty_{attackEvent}$
    end if
else
    $GlobalDifficulty_{attackEvent} \leftarrow LocalDifficulty_{attackEvent}$
end if
return $GlobalDifficulty_{attackEvent}$
```

Next, we assume that an Attacker will succeed to execute the target Attack event when she spends more *Effort* than the calculated *Global difficulty* of the target Attack event in the context of the concerned Abuse case.

ExpectedEffort–Difficulty Ratio (EDRatio) can be computed as

$$EDRatio = \frac{ExpectedEffortSpent_{abuseCase}}{GlobalDifficulty_{targetAtt(abuseCase)}}$$

Probability of Success (PoS) is a cumulative distribution function on probability density function *EDRatio*.

PoS with respect to a target Attack event is

$$PoS_{attackEvent} = \int_{-\infty}^{x} EDRatio(u)du$$

Loss events occur due to the execution of Attack events. These Attack events are target Attack events of different Abuse cases. The set of all the Loss events is represented by *LossEvent*.

Loss Event Probability (LEP) for a specific Loss event can be computed from all the Attack events which can trigger it. $cause : LossEvent \rightarrow 2^{AttackEvent}$ represents the mapping between Loss events and the set of loss causing Attack events. *LEP* depends on the *Probability of Success (PoS)* for the causal Attack events and *Threat Event Probability (TEP)* of all the Abuse cases related to those Attack events. If a Loss event is triggered from a causal Attack event which belongs to a single Abuse case, then

$$LEP_{lossEvent} = TEP_{abuseCase} \times PoS_{attackEvent}$$

[10] *Accessibility to Attack Surface (AtAS)* is used for PoC calculation with respect to an Abuse case. AtAS can be computed from the TC of the Attacker associated with the Abuse case. $AtAS_{abuseCase} = f_{AtAS}(TC_{attacker})$ where, $executedBy(abuseCase) = attacker$

$$where, cause(lossEvent) = \{attackEvent\},$$
$$attackEvent \in targetAtt(abuseCase)$$

There are n such Events which can trigger a specific Loss event. An event triggering a Loss event can be represented as

$$E_i = (attackEvent, abuseCase) \quad i : 1 \to n$$
$$where, attackEvent \in AttackEvent,$$
$$attackEvent \in targetAtt(abuseCase)$$

therefore,

$$P(E_i) = TEP_{abuseCase} \times PoS_{attackEvent}$$

LEP for a Loss event can be computed by considering all the related Events together, as given below (formulation for the probability of union of multiple events is used here),

$$\begin{aligned}\mathrm{LEP}_{\mathrm{lossEvent}} &= P(\bigcup_{i=1}^{n} E_i)\\ &= \sum_{i=1}^{n} P(E_i) - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} P(E_i \cap E_j)\\ &\quad + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=i+2}^{n} P(E_i \cap E_j \cap E_k) - \ldots\\ &= \sum_{i=1}^{n} (-1)^{i+1} \sum_{\substack{i_1, i_2, \ldots, i_k:\\ 1 \le i_1 < i_2 < \ldots < i_k \le n}} P(E_{i_1} \cap E_{i_2} \cap \ldots \cap E_{i_k})\end{aligned}$$

The set of Use cases is represented as *UseCase*. $impact : LossEvent \to 2^{UseCase}$ represents the mapping between Loss events and the set of impacted Use cases. Generally, the *Magnitude* of a Loss event ($Magnitude_{lossEvent}$) can be derived back to its *Impact* on the Use cases (which in turn may impact Business goals).

$$Magnitude_{lossEvent} = f_{Mag}(Impact(lossEvent))$$

In the context of an IT system, the *Risk* for a Loss event can be calculated from *LEP* and *Magnitude*.

$$Risk_{lossEvent} = LEP_{lossEvent} \times Magnitude_{lossEvent}$$

Addition of individual *Risk* values will compute the *Total Risk*

$$TotalRisk = \sum_{lossEvent \in LossEvent} Risk_{lossEvent}$$

There are different Actors related to an IT organization. The set of all the related Actors can be presented as *Actor*. Loss events are also associated with Actors. $loss : Actor \to 2^{LossEvent}$ represents mapping between Actors and Loss events. Computation of *Actor specific Risk* can be calculated as follows:

$$Risk_{actor} = \sum_{lossEvent \in loss(actor)} Risk_{lossEvent}$$

4 An illustrative example

In this section, an hypothesized video streaming company is presented. The company permits authenticated users to access the online video storage. Anyone can become a user of the video streaming company by paying the subscription fees. Part of the assets of the company and their associations are presented in Fig. 3 according to the metamodel described earlier. We present the example by using the metamodel domains vulnerability, threat, and impact.[11]

4.1 Vulnerability

Assets for the example video streaming company as shown in Fig. 3 can have vulnerabilities. A list of possible vulnerabilities in the assets are listed in Table 3. Severity values of the vulnerabilities are classified as high, medium, and low. CVSS [12] and CWSS [13] are the sources to find out vulnerability severity. Possible defenses against the vulnerabilities can be found out in the mitigation techniques listed in MITRE ATT&CK. [14] The difficulty values associated with the vulnerabilities can be taken from CVSS and CWSS or a method such as the one proposed in [46] can be employed for this purpose.

Identification of vulnerabilities on assets enables us to identify possible attack events. Various attack events can be causally related and can form attack graphs as shown in Figs. 4 and 5. The attack events are marked in colors to show their correspondence with the assets in Fig. 3. Defenses on the organizational assets can be used to protect vulnerabilities from exploitation. All the possible defenses are shown in the attack graphs given in Figures.[15]

[11] The example is not motivated nor empirically underpinned; it is only there to illustrate the Yacraf per se. The syntax used in the models does not follow any common strict notation policy for its mapping to the metamodel semantics, but we believe that it is intuitive enough to keep the reader out of confusion. Finally, the example illustrates only deterministic risk calculations to avoid too many details.

[12] https://www.first.org/cvss/.

[13] http://cwe.mitre.org/cwss/.

[14] https://attack.mitre.org/mitigations/.

[15] Effects of other possible defenses, if any, are directly included while calculating the difficulty values for the attack events.

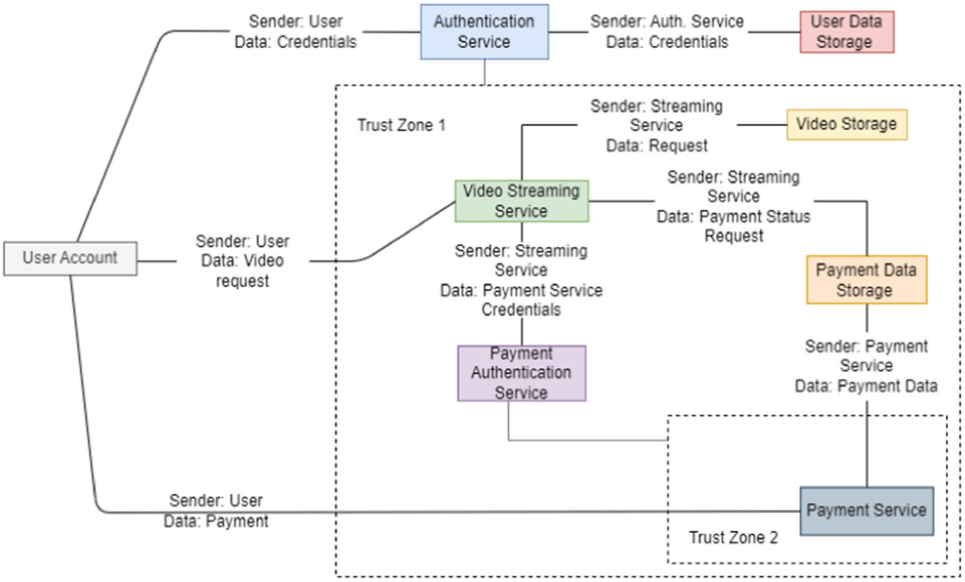

Fig. 3 Assets of the example video streaming company

Table 3 List of vulnerabilities

Asset	Vulnerability	Severity	Defense
Authentication service	Authentication bypass by spoofing (CWE-290)	Medium	Multi-factor authentication (M1032)
User data storage (MySQL 5.5.0)	Buffer overflow (CVE-2013-1492, CWE 119)	High	Update software (M1051)
Video storage (MongoDB Server v4.4)	Read Overrun (CVE-2020-7928)	Medium	Update software (M1051)
Video streaming service (nginx v1.15.5)	Memory disclosure (CVE-2018-16845)	Medium	Restrict file and directory permissions (M1022)
Payment authentication service	Origin validation error (CWE-346)	High	Change software configuration (M1054)
Payment data storage (postgreSQL 9.5.2)	NULL pointer dereference (CVE-2016-5423, CWE-476)	Medium	Update software (M1051)

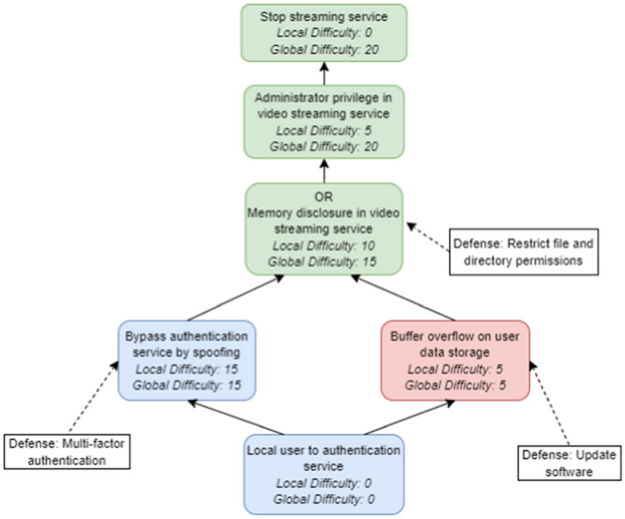
Stop streaming service
Local Difficulty: 0
Global Difficulty: 20
Administrator privilege in video streaming service
Local Difficulty: 5
Global Difficulty: 20
OR
Memory disclosure in video streaming service
Local Difficulty: 10
Global Difficulty: 15
Defense: Restrict file and directory permissions
Bypass authentication service by spoofing
Local Difficulty: 15
Global Difficulty: 15
Buffer overflow on user data storage
Local Difficulty: 5
Global Difficulty: 5
Defense: Multi-factor authentication
Local user to authentication service
Local Difficulty: 0
Global Difficulty: 0
Defense: Update software

Fig. 4 Possible attack graph for the hacktivist

4.2 Threat

An organization can be attacked by various types of attackers or threat agents. We have identified two potential attackers on the assets of the video streaming company given in Table 4. Qualitative values for various attributes of these two attackers are also listed in the table. Difficulty of attack varies depending on the capability of different attackers and also on the existence of defense mechanisms. Two attack graphs shown in Figs. 4 and 5 are created with respect to two different types of attackers, hacktivists and organized crime groups.

These potential attackers can perform different abuse cases as shown in Table 5. Calculation for all the different factors related to the abuse cases is shown. *PoC*, *PoA*, and *Effort spent* are calculated from the qualitative values of the component factors. Global difficulty values are computed from the related attack graphs. *TEP* and *PoS* are calculated as mentioned in the *Risk assessment calculations* in Sect. 3.2.

4.3 Impact

The use cases and the business goals with respect to the example company are shown in Fig. 6. The ultimate goal for the company is to become market leader in video streaming sector. To fulfill this goal, the company has two sub-goals: 1) satisfying existing customers and 2) to increase the number of customers. We have identified some use cases related to

Table 4 Attacker characteristics

Attacker	Risk tolerance	Concern for collateral damage	Skill	Resources	Sponsorship	Derived threat capability
Hacktivist	Medium	Medium	Medium	Medium	Low	Medium
Organized crime group	High	Medium	High	High	Medium	High

Table 5 Abuse cases

Attacker	Hacktivist	Hacktivist	Organized group	Organized group
Abuse case	Illegal access to user data storage	Block video streaming	Illegal access to user data storage	Bypass payment authentication service
Accessibility to attack surface (derived from threat capability in Table 4)	Medium	Medium	High	High
Window of opportunity	Medium	Low	High	Low
*Probability of contact (PoC)	0.4	0.2	0.8	0.4
Risk tolerance (Table 4)	Medium	Medium	High	High
Concern for co-lateral damage (Table 4)	Medium	Medium	Medium	Medium
Ability to repudiate	Medium	Low	High	Low
Perceived deterrence	Medium	High	Low	High
Perceived ease of attack	Medium	Low	High	Low
Perceived benefit of success	Low	Medium	Low	High
*Probability of action (PoA)	0.025	0.008	0.17	0.018
*Threat event probability ($TEP = PoC \times PoA$)	0.01	0.0016	0.136	0.0072
*Effort spent	3	8	3	19
*Global difficulty	7	20 (Fig. 4)	4	28 (Fig. 5)
*Probability of success (PoS)	0.43	0.4	0.75	0.68
$TEP \times PoS$	0.0043	0.00064	0.102	0.0049

*notation indicates the computed parameters

the business goals. These use cases are implemented by company assets and actors. In Fig. 6, color codes for the assets are the same as those used for Fig. 3.

Potential loss events caused by attack events are presented in Table 6. Impacted use cases and suffered actors due to the loss events are also listed in the table. Company is treated as an actor suffering from all the loss events.

4.4 Risk assessment

Risk values computed by the Yacraf risk assessment method proposed in this paper are shown in Table 6. Estimated loss event magnitude values are multiplied with the loss event probability values (derived from Table 5) to compute the actor and loss event specific risk. Total risk or actor specific risk can be computed by summing up the individual risk values.

While computing the *Local difficulty* values given in Figs. 4 and 5, it is assumed that all the defense mechanisms are active. With all activated defenses in Fig. 4, the Calculated Risk on the actor *Company* for the loss event *Videos_unavailable* will be €3200. If there are no active defenses in the attack graph, the local difficulty values for the attack events a) *Bypass_authentication_service_by_spoofing*, b) *Buffer overflow_on_user_data_storage*, and c) *Memory_disclosure_in_video_streaming_service* will become 5, 2, and 5, respectively. The *Global difficulty* value for the target attack event, i.e., *Stop_streaming_service*, will become 12. In this scenario, the Calculated Risk on the actor *Company* for the loss event *Videos_unavailable* will be €5500.

Table 6 Loss events

Target attack event	Access user data	Stop streaming service	Stop streaming service	Change payment data
Loss event	User data leaked	Videos unavailable	Videos unavailable	Payment data changed
Impacted use cases	Accessing videos	Streaming videos	Streaming videos	Making payments
Loss event type	Financial loss, legal issues	Productivity loss, loss of reputation, loss of competitive advantage	Productivity loss, financial loss	Financial loss, legal issues, loss of reputation
Suffered actor	Company	Company	User	Company
Magnitude	€100,000	€5,000,000	€1,000	€10,000,000
Abuse case	Illegal access to user data storage (by Hacktivist and Organized group)	Block video streaming (by hacktivist)	Block video streaming (by hacktivist)	Bypass payment authentication service (by Organized group)
*Loss event probability (LEP) (from Table 5)	0.107	0.00064	0.00064	0.0049
*Risk ($LEP \times magnitude$)	€10,700	€3200	€0.64	€49,000

*Notation indicates the computed parameters

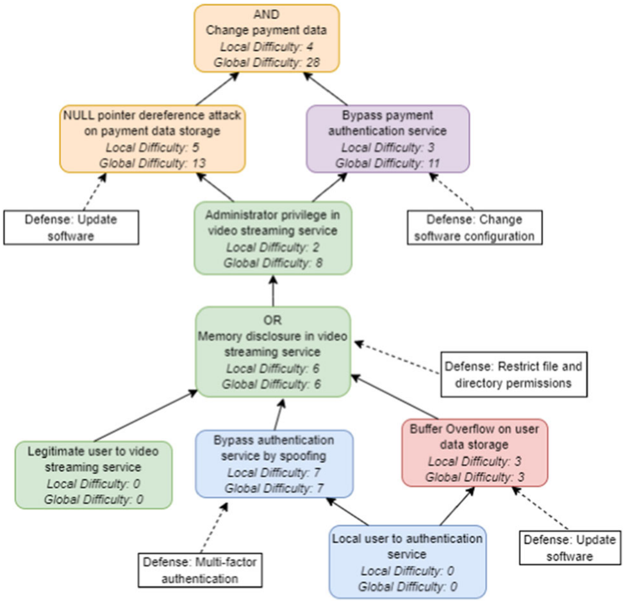
AND
Change payment data
Local Difficulty: 4
Global Difficulty: 28
NULL pointer dereference attack on payment data storage
Local Difficulty: 5
Global Difficulty: 13
Bypass payment authentication service
Local Difficulty: 3
Global Difficulty: 11
Defense: Update software
Administrator privilege in video streaming service
Local Difficulty: 2
Global Difficulty: 8
Defense: Change software configuration
OR
Memory disclosure in video streaming service
Local Difficulty: 6
Global Difficulty: 6
Defense: Restrict file and directory permissions
Legitimate user to video streaming service
Local Difficulty: 0
Global Difficulty: 0
Bypass authentication service by spoofing
Local Difficulty: 7
Global Difficulty: 7
Buffer Overflow on user data storage
Local Difficulty: 3
Global Difficulty: 3
Defense: Multi-factor authentication
Local user to authentication service
Local Difficulty: 0
Global Difficulty: 0
Defense: Update software

Fig. 5 Possible attack graph for the organized crime group

At this point, if *Restrict_file_and_directory_permissions* defense mechanism is activated, the global difficulty for the *Stop_streaming_service* will become 17. Calculated Risk will become €3750.

Any risk assessment method should satisfy the general intuitions about risk. From this example, we saw that the risk value is increasing when defenses are removed and is decreasing with the addition of defense mechanisms. It can be said that the Calculated Risk for this example is perfectly following the general intuitions regarding risk values.

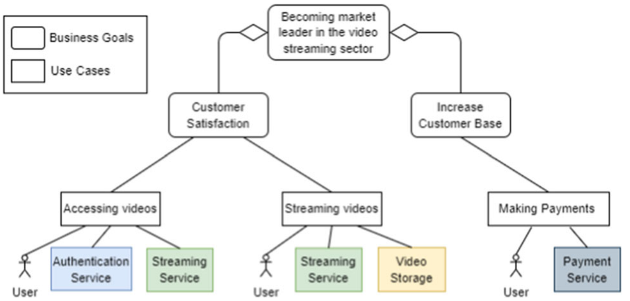

Fig. 6 Example business goals and use cases

4.5 Recommended defense mechanisms

Let us assume no defense mechanisms are enabled, for example, in Fig. 4. As given above, the *Global difficulty* value for the target attack event, i.e., *Stop_streaming_service*, is 12. If *Multi-factor_authentication* is the only activated defense mechanism, global difficulty for the *Stop_streaming_service* will remain unaltered, i.e., 12. In another scenario, if *Restrict_file_and_permissions* defense mechanism is activated, the global difficulty for the *Stop_streaming_service* will become 17. Calculated Risk on the actor *Company* for the loss event *Videos_unavailable* in these two separate scenarios would be €5500 and €3750. Implementing *Restrict_file_and_directory_permissions* defense will generate a better risk reduction compared to *Multi-factor_authentication*. Therefore, between these two available options we would recommend the first defense mechanism to the company. Depending on the security budget, a company may decide on a set of defense mechanisms to reduce the risk as cost-effectively as possible.

In general, analysis of the cost-effectiveness of defense mechanisms is not a trivial task, and it might contain a wide variety of mitigations, including both technical and organizational measures. We also acknowledge that this challenge lies both in estimating cost *and* effectiveness. Yacraf does not address these challenges per se, it only supports consistent and transparent analysis of the estimated values, so a complete discussion on this estimation problem is beyond the scope of this paper.

5 Discussion and practical experiences

Our objective with this work was to provide an explicit metamodel for risk-based threat modeling along with a risk calculation framework. The presented framework Yacraf uses features and best practices from well known methods such as FAIR [10] and PASTA [25] and further extends them. The metamodel is aligned with common approaches and/or tools from the threat modeling community (e.g., attack graphs) with some influences from enterprise architecture. The risk assessment also follows the general intuitions about risk as discussed earlier. In this way, we merged the model-based security analysis from the threat modeling community and the quantitative risk assessment calculations from risk management community. As part of the future work, we aim to put forward a tool to support the overall process of using our method.

We also have real world experience of using Yacraf. During 2020 we conducted three case studies in three different organizations, implementing our metamodel to assess their risks. Organization A is one of the leading developers of housing and residential areas in a European region. Its operations focus on new production of homes in attractive locations, with emphasis on expanding metropolitan areas and university towns. The annual revenue is approximately €1.5 billion, and the company has 2,600 employees. Organization B is one of the largest private real estate companies in a European country. They rent apartments and premises. Furthermore, they build both rentals and condominiums; 300 employees are responsible for a yearly revenue of €11 million. Organization C is a leading provider of facility management services in a European region, offering all the facility management services necessary for a company or public body to work smoothly and efficiently. Its yearly revenue of €950 million is generated by more than 10,000 employees.

The cases were executed by students, three different groups, as a part of their thesis projects [11, 43, 44]. Both senior researchers and company representatives were highly involved in the work. These studies were the first phase in a larger research project and the results served as input to the second phase, in which we are now. Based on the results of the Yacraf studies, we are currently conducting penetration tests of critical assets at the three companies. There are nine penetration testing sub-projects being guided based on the output of the Yacraf phase one cases, from industrial control systems used for electricity, heat, water, and ventilation in the buildings, to smart locks used for entering the facilities.

Our evaluation of the case studies hinted positive aspects of the use of our metamodel. It was perceived as easy to learn and understand. The applicants highlighted that the visualization of threats using graphs eased the understanding and that the different parts nicely fit together. The modularity also fosters the allocation of different tasks to various experts in the organization leading to a more efficient use.

Yacraf has also been the backbone of Master level university course for four years (2019–2022) with about 25 students per year. We taught the framework in a set of traditional lectures and with specific Q&A sessions. The students then applied the framework on an organization of their choice (most created a fictive case and some used a real case). We (as teachers) gave them feedback on sketches developed during the course that eventually resulted in full threat models with accompanying risk analyses. The students were expected to spend 200 h in total. We collected feedback after the courses were finished as a part of the regular course survey. Lastly, we conducted workshops with a set of student volunteers to get a deeper discussion about the framework and their process of applying it.

6 Conclusion

This paper presents a model-based risk assessment approach named Yet another cybersecurity risk assessment framework (Yacraf). Yacraf allows a holistic risk assessments for organizations by combining the two domains of model-based security analysis and quantitative risk assessment. The core novelty of this approach, however, revolves around the introduction of an explicit metamodel for model-based cybersecurity risk assessment. This enables more transparent and structured decision support than other approaches. The paper includes a formalization of risk calculations and also an example instant of how an organization can make use of Yacraf. The paper also provides a short summary of practical experiences of using Yacraf in real-world organizations in case studies. These studies demonstrate the positive potential of using Yacraf.

Funding This work was partly funded by the Swedish Civil Contingencies Agency (MSB) through the Swedish national research Center on Resilient Information and Control Systems (RICS). Open access funding was provided by KTH Royal Institute of Technology.

Data availability No datasets were generated or analyzed during the study so data availability is not applicable to this article.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethical Standards This article does not contain any studies with human participants or animals performed by any of the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Alam, M., Breu, R., Hafner, M.: Model-driven security engineering for trust management in secret. J. Softw. **2**(1), 47–59 (2007)
2. Almorsy, M., Grundy, J.: Secdsvl: A domain-specific visual language to support enterprise security modelling. In: 23rd Australian Software Engineering Conference (ASWEC), pp. 152–161 (2014)
3. Basin, D., Doser, J., Lodderstedt, T.: Model driven security: from uml models to access control infrastructures. ACM Trans. Softw. Eng. Methodol. (TOSEM) **15**(1), 39–91 (2006)
4. Basin, D., Clavel, M., Egea, M.: A decade of model-driven security. In: Proceedings of the 16th ACM Symposium on Access Control Models and Technologies, pp. 1–10 (2011)
5. Beckers, K., Heisel, M., Solhaug, B., Stølen, K.: ISMS-CORAS: A structured method for establishing an ISO 27001 compliant information security management system. In: Heisel, M., Joosen, W., López, J., Martinelli, F. (eds) Engineering Secure Future Internet Services and Systems-Current Research. Lecture Notes in Computer Science, vol. 8431, pp. 315–344. Springer (2014)
6. Caralli, R.A., Stevens, J.F., Young, L.R., Wilson, W.R.: Introducing octave allegro: improving the information security risk assessment process. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, Technical report (2007)
7. Committee of Sponsoring Organizations of the Treadway Commission, et al (2004) Enterprise risk management-integrated framework: executive summary and framework. American Institute of Certified Public Accountants (AICPA)
8. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir. Eng. **16**(1), 3–32 (2011)
9. ENISA: Compendium of risk management frameworks with potential interoperability. Technical report, European Union Agency for Cybersecurity (2022)
10. Freund, J., Jones, J.: Measuring and Managing Information Risk. Butterworth-Heinemann, Waltham (2015). https://doi.org/10.1016/C2013-0-09966-5
11. Friman, N.: Security analysis of smart buildings. Bachelor thesis, School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology (2020)
12. Hafner, M., Breu, R., Agreiter, B., Nowak, A.: Sectet: an extensible framework for the realization of secure inter-organizational workflows. Internet Res. **16**(5), 491–506 (2006)
13. ISO: ISO 31000:2018 Risk management-Guidelines. Standard, International Organization for Standardization (2018)
14. ISO, IEC,: ISO/IEC 27001:2005 Information technology-Security techniques-Information security management systems-Requirements. Standard, International Organization for Standardization/International Electrotechnical Commission (2005)
15. ISO, IEC,: ISO/IEC 27000:2018 Information technology-Security techniques-Information security management systems-Overview and vocabulary. Standard, International Organization for Standardization/International Electrotechnical Commission (2018)
16. ISO, IEC,: ISO/IEC 27005:2018 Information technology-Security techniques-Information security risk management. Standard, International Organization for Standardization/International Electrotechnical Commission (2018)
17. Johnson, P., Lagerström, R., Ekstedt, M.: A meta language for threat modeling and attack simulations. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, pp 1–8 (2018)
18. Jürjens, J.: UMLsec: Extending UML for secure systems development. In: Jézéquel J, Hußmann H, Cook S (eds) UML 2002—The Unified Modeling Language, 5th International Conference, Dresden, Germany, 2002, Proceedings, Springer, Lecture Notes in Computer Science, vol. 2460, pp. 412–425 (2002)
19. Jürjens, J.: Secure Systems Development with UML. Springer, Berlin, Heidelberg (2005)
20. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of attack–defense trees. In: International Workshop on Formal Aspects in Security and Trust, pp. 80–95 (2010)
21. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis: The CORAS Approach. Springer, Berlin, Heidelberg (2010)
22. Mathey, F., Bonhomme, C., Rocha, J., Lombardi, J., Joly, B.: Risk assessment optimisation with MONARC. https://www.monarc.lu/assets/files/publications/2018-HACK.LU-CASES.pdf (2018)
23. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: International Conference on Information Security and Cryptology, pp. 186–198 (2005)
24. McDermott, J.P., Fox, C.: Using abuse case models for security requirements analysis. In: 15th Annual Computer Security Applications Conference (ACSAC 1999), 6–10 December 1999, pp. 55–64. AZ, USA, IEEE Computer Society, Scottsdale (1999)
25. Morana, M.M., Uceda Vélez, T.: Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. Wiley, Hoboken, New Jersey (2015)
26. Mouratidis, H., Giorgini, P., Manson, G., Philp, I., et al.: A natural extension of tropos methodology for modelling security. In: Proceedings Agent Oriented Methodologies Workshop, Annual ACM Conference on Object Oriented Programming, Systems, Languages (OOPSLA), Seattle (2002)
27. Noel, S., Elder, M., Jajodia, S., Kalapa, P., O'Hare, S., Prole, K.: Advances in topological vulnerability analysis. In: Cybersecurity

Applications Technology Conference For Homeland Security, pp. 124–129, (2009) https://doi.org/10.1109/CATCH.2009.19

28. Paja, E., Dalpiaz, F., Giorgini, P.: Modelling and reasoning about security requirements in socio-technical systems. Data Knowl. Eng. **98**, 123–143 (2015)
29. Peltier, T.: Information Security Risk Analysis. Auerbach Publications, Boca Raton (2010)
30. Potteiger, B., Martins, G., Koutsoukos, XD.: Software and attack centric integrated threat modeling for quantitative risk assessment. In: Scherlis WL, Brumley D (eds) Proceedings of the Symposium and Bootcamp on the Science of Security, Pittsburgh, April 19–21, 2016, pp 99–108. ACM (2016)
31. Ross, R.: Guide for conducting risk assessments. NIST Special Publication 800-30 Revision 1, National Institute of Standard and Technology (2012)
32. Saitta, P., Larcom, B., Eddington M.:Trike v1 methodology document (2005)
33. Schneier, B.: Attack trees. Dr Dobb's J **24**(12), 21–29 (1999)
34. Schneier, B.: Lies Digital Security in a Networked World, vol. 21, pp. 318–333. Wiley, New York (2000)
35. Shevchenko, N., Chick, T.A., O'Riordan, P., Scanlon, T.P., Woody, C.: Threat modeling: a summary of available methods. Carnegie Mellon University Software Engineering Institute, Technical report (2018)
36. Shevchenko, N., Frye, B.R., Woody, C.: Threat modeling for cyber-physical system-of-systems: methods evaluation. Carnegie Mellon University Software Engineering Institute, Technical report (2018)
37. Shostack, A.: Experiences threat modeling at microsoft. Technical report, Microsoft (2008)
38. Shostack, A.: Threat Modeling: Designing for Security. Wiley, Indianapolis (2014)
39. Tuma, K., Calikli, G., Scandariato, R.: Threat analysis of software systems: a systematic literature review. J. Syst. Softw. **144**, 275–294 (2018)
40. UK Government Central Computer and Telecommunications Agency (CCTA): CCTA risk analysis and management method (CRAMM). Technical report, CCTA (2003)
41. Wang, J., Neil, M., Fenton, NE.: A Bayesian network approach for cybersecurity risk assessment implementing and extending the FAIR model. Comput Secur **89** (2020)
42. Wangen, G., Hallstensen, C., Snekkenes, E.: A framework for estimating information security risk assessment method completeness. Int. J. Inf. Secur. **17**(6), 681–699 (2018)
43. Weigelt, C.: af Rantzien DFH A process for threat modeling of large-scale computer systems: A case study. Bachelor thesis, School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology (2020)
44. Wessman, L., Wessman, N.: Threat modeling of large-scale computer systems: implementing and evaluating threat modeling at company x. Bachelor thesis, School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology (2020)
45. Xiong, W., Lagerström, R.: Threat modeling—a systematic literature review. Comput. Secur. **84**, 53–69 (2019)
46. Xiong, W., Hacks, S., Lagerström, R.: A method for assigning probability distributions in attack simulation languages. Complex Syst. Inf. Model Q **26**, 55–77 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.