International Journal of Information Security (2022) 21:509–525
https://doi.org/10.1007/s10207-021-00566-3

REGULAR CONTRIBUTION

A risk-level assessment system based on the STRIDE/DREAD model for digital data marketplaces

Lu Zhang[1] · Arie Taal[1] · Reginald Cushing[1] · Cees de Laat[2] · Paola Grosso[1]

Published online: 14 September 2021

Abstract
Security is a top concern in digital infrastructure and there is a basic need to assess the level of security ensured for any given application. To accommodate this requirement, we propose a new risk assessment system. Our system identifies threats of an application workflow, computes the severity weights with the modified Microsoft STRIDE/DREAD model and estimates the final risk exposure after applying security countermeasures in the available digital infrastructures. This allows potential customers to rank these infrastructures in terms of security for their own specific use cases. We additionally present a method to validate the stability and resolution of our ranking system with respect to subjective choices of the DREAD model threat rating parameters. Our results show that our system is stable against unavoidable subjective choices of the DREAD model parameters for a specific use case, with a rank correlation higher than 0.93 and normalised mean square error lower than 0.05.

Keywords Risk assessment · STRIDE/DREAD model · Robust · Resolution · Digital data marketplace

1 Introduction

Sharing and utilising others' data can generate great value and improve collaborations among parties [1]. Digital data marketplace (DDM) is a distributed data trading platform that supports data and/or computes asset sharing and federation among consortium members to achieve a common goal [2]. An application area in which this concept is taking off is aviation. Multiple airline companies may share their aircraft data to predict the necessity of the air plane maintenance by training a machine learning model. They delegate their applications to a DDM infrastructure for better security and sovereignty. It is obviously a basic necessity for any DDM customer, such as for example an airline, to estimate the guaranteed security level of such digital infrastructures. To solve this problem, we propose a new system to assess the remaining risk of a specific application after applying security countermeasures of an existing infrastructure. The evaluation results can be used as guidelines to rank available DDM infrastructures in terms of guaranteed security.

There are studies proposing methodologies to assess the security levels provided by digital infrastructures, e.g. clouds [3–5]. These works only estimate the total security strength provided by the infrastructures according to the applied security countermeasures. They do not consider the influence of concrete applications on the obtained security level. Different applications may have different threats and the severity level of each threat may also change with applications.

The Microsoft STRIDE/DREAD model applies *risk attributes*, e.g. Damage and Affected Users, to measure the *likelihood* and impact of exploiting a vulnerability. Most recent work use the STRIDE/DREAD model to rank threats based on their severities. More recently, this has been proposed in IoT frameworks [6,7], and in cloud environments [8]. However, we adopt the model to compute the relative importance of each threat [9]. We also propose the new *risk attributes* for the DREAD model to fit the context of a DDM use case and define more fine-grained definitions of these attributes and their corresponding levels in our system to gain more objective assessment results.

Our system identifies threats semi-automatically by splitting the input application into transaction lists, assigns severity weights of each threat and estimates the final risk

✉ Lu Zhang
l.zhang2@uva.nl

[1] Multicale Networked Systems (MNS) Lab, University of Amsterdam, Amsterdam, The Netherlands

[2] Complex Cyber Infrastructure (CCI), University of Amsterdam, Amsterdam, The Netherlands

exposure after applying security countermeasures in the available digital infrastructures.

We additionally present a method to validate the stability and resolution of our ranking system with respect to subjective choices of the DREAD model threat rating parameters. The numerical values of *risk attributes* assigned to threats cannot be constant values during the life span of the system applying the model. Also, the choice of numeric values is not sufficiently objective. It is therefore important to analyse the stability and sensitivity of the STRIDE/DREAD model due to subjective choices of parameters in a real world use case.

To quantify the robustness of our system for different values of risk parameters, we use three metrics: two well-known, normalised mean square error (NMSE) and Kendall's Tau and one we define ourselves, granularity [10,11]. The metric granularity provides us insights into the resolution. Our experimental results show that our risk assessment methodology is stable to subjective choices of the *risk parameters* and able to provide sufficient resolution to discriminate the severity of real-world threats in general. Additionally, we observe that methodology performance is highly dependent on the application scenarios and corresponding threat databases.

The main contributions of our work are:

- We propose a threat-oriented risk assessment system that can quantitatively evaluate the remaining risk for data exchange applications that can be provided by DDM digital infrastructures.
- We investigate the robustness and resolution of the STRIDE/DREAD model in our proposed risk assessment system with respect to subjective choices of the risk parameters.
- We demonstrate the robustness and suitability of our risk assessment system for adoption in DDM infrastructures.

2 Related work

Recent research assesses the security provided by digital infrastructures, e.g. clouds. Zhang et al. propose an approach to assess the security of a cloud platform, but it only focuses on individual threats separately and provide only qualitative evaluation results [12]. Luna et al. propose a methodology in [3] to assess the security level of a Security Service Level Agreement (SecSLA) with respect to customers' requirements. The work allows cloud clients to compare SecSLAs provided by different cloud service providers (CSP) and aims to provide costumers with a general view of security coverage of the provided infrastructures. The security controls of the cloud infrastructures are classified based on the cloud control matrix (CCM) taxonomy, which makes the proposed system very difficult to migrate to another application context, e.g. Digital Data Marketplaces. Shaikh and Sasikumar advocate a similar security evaluation methodology of SLAs in [4] by using a trust model. Different security countermeasures are chained according to the taxonomy defined in this trust model. The system measures the security strength from multiple dimensions and computes a trust value. Nevertheless, these authors fail to consider that the vulnerabilities, normally varying with each application, have a strong influence on the effectiveness of applied security countermeasures. Sen and Madria propose an off-line risk assessment framework in [5] to evaluate the security level of an application for a specific CSP. They first identify the threats for a given application and estimate how much risk can be mitigated with the CSP's infrastructures. However, the system treats all the identified threats with equal severities and this is not what happens in real-world scenarios.

There are multiple risk management frameworks for information systems. The ISO provides standards with which an information system can gain adequate security. The standards describe the control objectives, required security controls and guidelines. The ISO standards are widely used as certifications for companies to verify the security of their information systems and promote customer's trust [13]. The National Institute for Standards and Technology (NIST) cybersecurity framework also offers guidance to facilitate risk management within specific organisations [14]. This framework aims to keep an information system safe by identifying security gaps. OCTAVE is also a risk-based assessment and planning process. It identifies the infrastructure vulnerabilities and develops protection strategies in design [15]. However, all the work mentioned above focuses on risk management while establishing a digital infrastructure. After risk analysis, the output of those frameworks are implementation requirements, e.g. security countermeasures, user action guidance's, for a single information system. Our proposed framework aims to choose the most secure digital infrastructures in an application-based manner with risk scores. CORAS is a model-driven risk assessment framework. It identifies the threats of a use case, assess the risk of each threat and develop treatments [16]. But it does not consider the relative importance of each threat and does not provide a total risk score for ranking different DDM digital infrastructures.

The Microsoft STRIDE/DREAD model provides a threat modelling approach and assesses a single threat risk by proposing attributes measuring difficulties of exploiting the vulnerability [9]. Most studies of the STRIDE/DREAD model focus on risk evaluation of an individual threat and provide threats ranking regarding their risk [17,18]. In [8], Anand et al. use the STRIDE/DREAD model to assess and prioritise threats in a cloud environment. They adapt the original risk parameters to the cloud environment and assign *impact factors* to each threat category. However, their model does not fit the context of DDMs, where applications are modelled as workflows and trust among collaborating parties

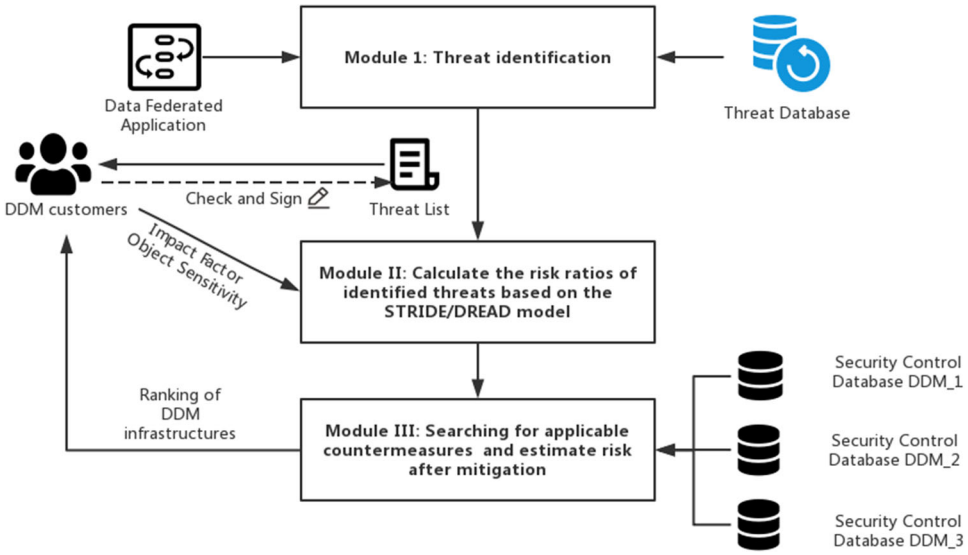


Fig. 1 Architecture of our application-based risk assessment system

plays a vital role. Moreover, all the studies in [8,17,18] just inherit the numeric values of *risk attributes* from the original Microsoft STRIDE/DREAD model without validating the objectivity of the choices.

From this overview, it is clear that our work covers a currently unexplored area. Firstly, it specifically caters to DDMs and their customers. Secondly, it assigns severity weights for identified threats, with the modified STRIDE/DREAD model, to achieve more objective risk assessment results. Thirdly, we investigate the robustness and resolution of our proposed system against subjective choices of risk parameters in the original STRIDE/DREAD model with real-world security threats.

3 System architecture

A digital data marketplace (DDM) is a digital infrastructure that facilitates secure data exchange and federation. For instance, different DDM parties may want to gather their local data together and run a machine learning (ML) algorithm on their joint data, so that they can gain benefits from a more accurate prediction model. In the DDM community, there might be multiple DDM infrastructures with well-implemented security countermeasures and devices. DDM customers delegate their data federation applications to one of the DDMs for better security governance. Currently, there are two primary typical DDM applications. One is training disease diagnosis models in the health-care field; another one is to predict air plane maintenance necessity for airline companies.

Different data exchange applications suffer from different vulnerabilities. Likewise, different DDM infrastructure providers apply varying sets of security countermeasures. When deploying an application, these varying threats and countermeasures contribute to different final risk levels depending on the DDM it runs in. Our risk assessment system is designed collaboratively to increase the transparency and boost the trust of DDM customers to DDM providers.

The risk assessment is performed by a broker, who is essentially a trusted third party and closely cooperating with DDM customers and providers. The system estimates the risk level of all DDMs with respect to an application and provides a ranking of these DDMs to a DDM customer.

Figure 1 shows the architecture of the system. A collaboration of DDM customers first feed their applications, which is actually a list of transactions, into the risk assessment system. Module I identifies corresponding threats of the input application automatically by using a pre-constructed *Threat Database*. The *Threat Database* is constructed a priori by identifying a wide range of threats for typical data exchange applications in DDMs. The *Threat Database* can be updated during run time of the system, because new threats may occur and some existing threats may become obsolete. The list of identified threats is sent to the DDM customer and each collaborating party checks this threat list. They sign the list if they agree, or go into a negotiation phase if they disagree. Only with all the signatures from the collaborating parties, module II of the risk assessment system will process the approved threat list.

Module II estimates the risk level of each threat in the list with the modified STRIDE/DREAD model from Microsoft [9]. This model considers the possibility of an attack occurrence using five *risk attributes* and also the impact of each threat regarding the concrete application. DDM customers also provide *impact factors* and *object sensitivity* as inputs to module II. The *impact factors* reflect how the DDM customer perceives the influence of certain threats for their application.

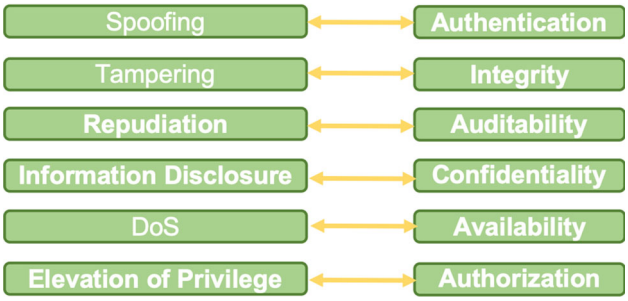


Fig. 2 Correspondence between the threat categories in the STRIDE model (left) and the security features (right)

The *object sensitivity* reflects the sensitivity of the shared data of the application perceived by the DDM customer.

Module III matches the threats with corresponding security countermeasures provided by individual DDM providers. This module determines the risk reduction level of each threat provided by different DDMs and calculates the total remaining risk to this application. Finally, this module provides the DDM rankings back to the DDM customers.

4 Module I: application-oriented threat identification

4.1 Mapping between Microsoft STRIDE model and security features

The STRIDE model is a threat modelling tool developed by Microsoft for analysing security flaws for cyber-security systems [9]. It groups threats into six categories: Spoofing (S), Tampering (T), Repudiation (R), Information disclosure (I), Denial of service (D), and Elevation of privilege (E) [9]. All the identified threats for a data-exchange application belong to at least one of these categories.

We define a mapping of the threat categories in the STRIDE model onto more generally understood security features, see Fig. 2 [19]. So it is more intuitive and comfortable for the DDM customers to consider the impacts of each threat category for their applications. In this way, the DDM customer does not need to have background knowledge of the STRIDE model. A threat may have distinct risks for different applications because applications may have various security goals. For example, the threats belonging to the category of Information Disclosure damage the confidentiality of the shared data rather than integrity.

4.2 Applications of DDM-governed data exchange

As shown in Fig. 1, DDM customers provide their application to the risk assessment system. In the DDM infrastructure, it is normal to use transaction lists to represent a data federation application. In this module, the input DDM application is split into multiple transactions. An example transaction list is shown below, with DO, CO, DP, AP representing data objects, compute objects, data providers, algorithm providers, respectively.

Transactions of an example DDM application
Trans 1: Third-party accesses DO from DP via remote mounting
Trans 2: Third-party accesses CO from AP via direct transfer
Trans 3: Third-party processes CO on DO with feature multi-tenancy, generating IRO
Trans 4: AP accesses IRO via direct transfer
Trans 5: AP processes on a third party

We characterise each transaction as an attribute tuple:

$$\langle stage,\ source,\ target,\ object,\ feature\rangle$$

Source and *target* are consortium parties of a DDM. *Feature* describes important aspects for threat identification. Direct transfer and remote mounting are two *features* for a transaction with *stage* transmission. For instance, the attribute tuple of transaction 1 becomes $\langle$transmission, DP, third party, DO, remote mounting$\rangle$.

4.2.1 Object sensitivity

The risk assessment system requires *object sensitivity*. It determines the potential damage of a threat. The *object sensitivity* depends on an individual application. For example, data objects in health-care applications usually are more sensitive than others because they may contain private information of patients.

4.2.2 Impact factors

Due to the concrete security goal of an application, the impact of each threat category in the STRIDE model varies. The DDM customers are required to assign *impact factors* for each threat category based on its corresponding security feature. According to the work in [8], there are five levels, which are critical (1), high (0.75), medium (0.5), low (0.25), and none (0), to scale the *impact factor*. The *impact factor* indicates the degree of concern of DDM customers about each threat category. A DDM customer supplying the *impact factor* critical asks for the greatest concern and priority for a specific threat, and with *impact factor* none the DDM customer has no concern about a given threat.

4.3 Threat modelling

Here, we introduce a general methodology for identifying threats for applications in DDMs. Every application can be split into a sequence of transactions, each of which can be represented by a 5-tuple. The threats of each transaction can be identified primarily based on its *stage* and *feature*. The threats of an application are the union set of threats for all its transactions. In addition, DDMs, as distributed platforms for data federation applications, are based on virtualisation technologies for better isolation. We consider common vulnerabilities of virtualisation when modelling threats for a given application. For instance, threats caused by the multi-tenancy *feature* [20].

We classify the threats of a DDM application into three *stages*. Stage I is data in storage, and the main concern is confidentiality, availability, privacy of asset objects in storage. Stage II is data in transmission, which is related to issues such as end-to-end communication security. Stage III is data in execution, and it focuses on whether the procession by an algorithm on the data complies with the agreed policies.

There are some threats mainly depending on the attribute *stage* of the transaction. For example, the threat of 'data object leakage during end-to-end transmission' exists in nearly all transactions with *stage* of transmission. Attacks like 'man-in-the-middle' and 'eavesdropping' may exploit these threats. Similarly, the threat of 'malicious compute objects during execution' is also common for transactions with *stage* attribute of execution.

However, some threats are dependent on distinct *features* of a transaction. For instance, mounting a local file may give a third-party sufficient permission to suffer from the threat of data object tampering. The *feature* multi-tenancy indicates data objects are processed individually in separate containers on the same physical third-party platform. An example threat for this *feature* is the 'denial-of-service attack' by one of the malicious co-tenant containers.

According to the approach introduced before, we conduct threat modelling for DDM applications semi-automatically according to a pre-defined dynamic threat database.

Figure 3 shows the screenshot of a pre-constructed SQL threat database for a DDM use case. Each a priori identified threat has nine different attributes, namely *threatName, stage, category, archetype, DP, AC, SL, AU, and ID*.

The *stage* describes in which stage this threat occurs. The *category* refers to the threat categories in the STRIDE model, as discussed in Sect. 4.1. It indicates to which category this threat belongs. *DP, AC, SL, AU, ID* are assigned values for the *risk attributes* for the given threat. The *archetype* describes the collaborating relationships among DDM members and each application follows at least one *archetype* [2]. The concrete *archetypes* and the complete version of the threat database can be found in github. [1]

5 Module II: risk assessment of an individual threat

Once threats have been identified by the methodology described in Sect. 4 and approved by all collaborating parties, this module computes the application-dependent *risk ratio* of each threat with the modified Microsoft DREAD model. The DREAD model is commonly used to rank individual threats based on their severities. In module II, we adopt the concept of the DREAD model to compute the relative importance of each threat according to the estimated risk level. Furthermore, we redefine five *risk attributes* to fit the context of DDM applications and increase objectivity in the assessment procedure.

5.1 Original DREAD model

The original DREAD part of the STRIDE/DREAD model proposed by Microsoft is used to assess and rank threats in terms of their risk [9]. It defines five *risk attributes* to estimate the probability of an exploitation of a vulnerability from distinct aspects. These attributes are Damage (D), Reproducibility (R), Exploitability (E), Affected users (A), and Discoverability (Di) [21].

- Damage (D): How much are the assets affected?
- Reproducibility (R): How easily the attack can be reproduced?
- Exploitability (E): How easily the attack can be launched?
- Affected users (A): What's the number of affected users?
- Discoverability (Di): How easily the vulnerability can be found?

Each *risk attribute* is scaled into three qualitative levels as high, medium, and low. Due to the property of a concrete threat, one of the three qualitative levels can be assigned for each *risk attribute*. All the five aspects need to be considered to assess the risk of a threat. The threat risk ranges from 0 to 10, and the DREAD model uses three integers 0, 5, and 10, to represent the three corresponding levels numerically. In the STRIDE/DREAD model, we represent a threat by the following five-tuple $(D_{t_i}, R_{t_i}, E_{t_i}, A_{t_i}, Di_{t_i})$ with $D_{t_i}, R_{t_i}, E_{t_i}, A_{t_i}, Di_{t_i} \in \{0, 5, 10\}$ of numeric numbers. The risk, represented as a metric called *risk score* $rs(t_i)$, is quantified as an average of the numeric values of those five *risk attributes*:

[1] https://github.com/kelsey-1015/DL4LD.

$$rs(t_i) = \frac{1}{5}(D_{t_i} + R_{t_i} + E_{t_i} + A_{t_i} + Di_{t_i}) \quad (1)$$

According to the *risk scores* of the threats, the DREAD model can rank all the threats regarding their risk.

However, the description of each risk parameter is obscure and there are no concrete definitions of each level for the original DREAD model. This probably increases the degree of subjectivity when assessing the risk level of a single threat with the original DREAD model.

5.2 Modified DREAD model for DDMs

We redefine five *risk attributes* and corresponding risk levels to better meet the requirement of the DDM applications. For example, we address the importance of monitoring and potential trust among collaborating parties in a DDM instance. Table 1 shows the defined *risk attributes* and qualitative descriptions of three scaled levels.

Damage Potential (DP) describes the damage caused if a threat occurs. The assets of DDM applications are data objects, compute objects and intermediate results objects, which we have discussed in Sect. 4.2. The *object sensitivity* assigned by the DDM customer determines the corresponding level of the *risk attribute* Damage Potential (DP). For some threats like encryption key leakage during exchange, the DP is always set as the highest level regardless of the *objective sensitivity* of the application. In Fig. 3, we use "TOP" to represent such threats for attribute DP.

Accessibility (AC) describes who can perform attacks to exploit a threat. If collaborating members of a DDM can only perform the attacks, the attribute is scaled as low due to the mutual trust among them. If a third party of an application can also exploit the threat, i.e. more risk is included, AC is scaled as medium. The highest risk occurs if one entity can perform this attack, including malicious parties outside the DDM.

Skill level (SL) defines what skills are needed to exploit this threat. The probability is much lower if this exploitation requires complex programming or hacker skills. The risk is the highest, scaled as high, if it just requires simple tools or even a web browser.

Affected users (AU) is scaled into different levels according to how many collaborating parties are affected if a threat occurs.

Intrusion detectability (ID) describes how easy monitoring tools can detect the exploitation of this threat. A threat is more severe if its exploitation is more difficult to detect, which indicates a higher success rate of attacks and more resulting damage.

Security experts can determine these *risk attributes* a priori and reference information can be found in some public vulnerability databases, for instance, CAPEC [22]. Damage

Table 1 *Risk attributes* of modified DREAD model and corresponding qualitative descriptions of three levels

Risk attributes	Damage potential (DP)	Accessibility (AC)	Skill level (SL)	Affected users (AU)	Intrusion detectability (ID)
Low	Low data sensitivity	By collaborating parties	Advanced skills	One party member	Detectable without monitoring
Medium	Medium data sensitivity	By collaborating parties or any trusted third party	Malware existing in using attack tools	Internet or Partial party members	Detectable by monitoring
High	High data sensitivity	By outsiders of DDMs	Simple tools	All party members	Very hard to detect even by monitoring

ThreatName	Stage	Category	Archetype	DP	AC	SL	AU	ID
IP spoofing	II	S	ALL	SO	H	M	H	M
Identity spoofing: via remote data access	III	S	IV, V, VII	SO	H	L	M	H
Insecure data deletion	III	ID	ALL	SO	M	L	M	H
Malicious compute: Data Disclosure	III	ID	ALL	SO	L	H	H	M
Unauthorized disclosure: Eavesdropping	II	ID	ALL	SO	H	H	M	H
Weak Access Control	I	ID	ALL	SO	H	H	L	H
Malicious compute: high result correlation	III	ID	III	SO	L	H	H	M
Encryption Keys Leakage during exchange	II	ID	ALL	TOP	H	L	H	H

Fig. 3 A screenshot of a SQL threat database for a DDM use case

potential (DP) and affected users (AU) are application-dependent and subjective in nature. Currently, we use 0, 5, and 10 to represent the three *risk attribute* levels numerically. We further discuss the influence of other numeric representations on the stability and resolution of our methodology.

Integrating the application-dependent *impact factors* described in Sect. 4.2.2, we calculate the risk score $rs(t_i)$ of a threat t_i as the product of a *likelihood LH* and an *impact factor IF*:

$$rs(t_i) = LH(t_i) \cdot IF(t_i) \tag{2}$$

The *likelihood* $LH(t_i)$ and the *impact factor* $IF(t_i)$ are obtained as follows:

$$LH(t_i) = \frac{1}{5}(DP_{t_i} + AC_{t_i} + SL_{t_i} + AU_{t_i} + ID_{t_i}), \tag{3}$$

where DP_{t_i}, AC_{t_i}, SL_{t_i}, AU_{t_i}, ID_{t_i} denote the numeric values of the five *risk attributes* in Table 1 for threat t_i, and $IF(t_i)$ equals to the *impact factor* of the threat category in the STRIDE model that threat t_i belongs to.

We must observe that the *likelihood* LH is a linear combination of the five *risk attributes*. By the choice of a linear combination, Microsoft treats all attributes equally.

According to Eq. (3), we can compute the *risk score* of each threat for the application, which represents the risk of each threat. A higher *risk score* indicates a more dangerous threat for the concrete application.

To determine the relative importance, we define a *risk ratio* *rr* of each threat t_i in the threat list of the application. This is calculated as follows:

$$rr(t_i) = \frac{rs(t_i)}{\sum_{t_i \in T} rs(t_i)}, \quad \text{with} \quad \sum_{t_i \in T} rr(t_i) = 1, \tag{4}$$

where $rs(t_i)$ denotes the *risk score* of threat t_i, T denotes the threat list of the application identified by module I.

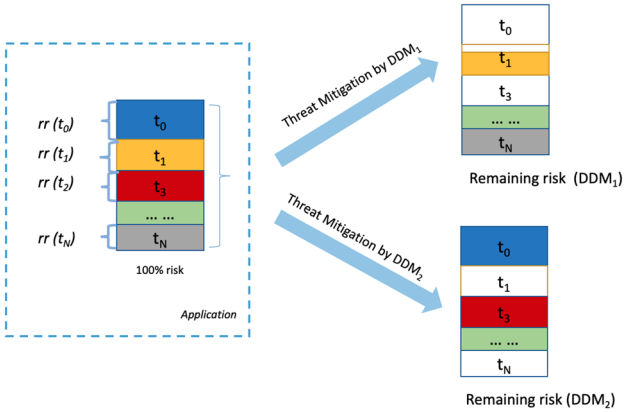


Fig. 4 Functionality of module III. On the left side, we see the input of module III, the identified threats of an application with corresponding *risk ratios*. On the right side, we see the remaining risk of each threat after applying security countermeasures by the DDMs. White areas indicate zero risk, coloured areas indicate the remaining risk

6 Module III: risk mitigation and risk-level evaluation

Module III of the risk assessment system matches security countermeasures to identified threats for an application, computes the mitigation level of each threat and calculates the total remaining risk of the application.

As shown in Fig. 4, the input of module III is a list of threats with corresponding *risk ratios* $rr(t_i)$. Those threats constitute the original 100% risk of the application without any mitigation from DDMs and the proportion of each threat is equal to its *risk ratio* calculated by Eq. (4). According to information of security countermeasures provided by DDMs, this module ranks DDMs regarding total remaining risk for the application.

6.1 Security countermeasures matching and threat mitigation

As shown in Fig. 1, DDM providers publish the *CM Database*, a database of supporting security countermeasures. The risk assessment system accesses each *CM Database*

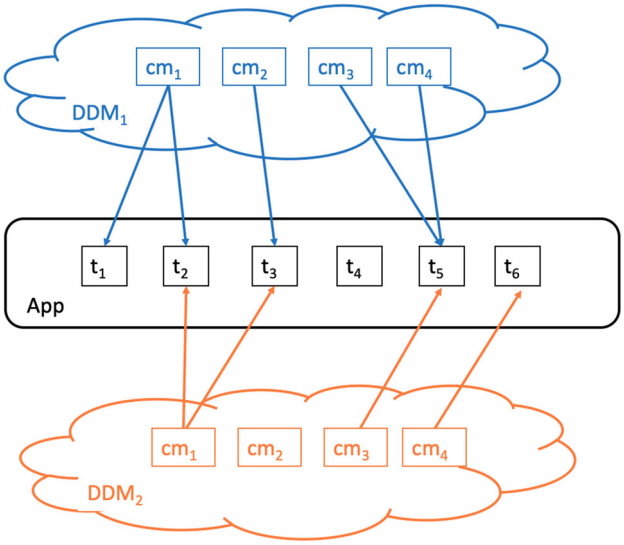


Fig. 5 Threats mitigation by security countermeasures of each DDM provider. This results into a mitigation list of original threats for each DDM

and matches suitable security countermeasures for each threat of the application. Figure 5 illustrates the matching procedure. The module checks both the feasibility and necessity of applying a security mechanism to an application. Necessity indicates whether a security countermeasure can mitigate one or multiple threats identified for the application. Feasibility means whether a security countermeasure can fit the data type or data volume of the shared objects of the application. For instance, the watermarking techniques are only applicable to data objects of images. In Fig. 5, an arrow from cm_j to t_i indicates countermeasure cm_j is both feasible and necessary to apply to the application to mitigate threat t_i. The matching from security countermeasures to threats can be one-to-one (1–1), one-to-multiple (1–N) or multiple-to-one (N–1). A one-to-multiple mapping indicates a security countermeasure is capable to mitigate multiple threats. A multiple-to-one mapping means multiple security countermeasures apply to only one threat.

Every DDM_k applies a mitigation factor $f_{m;k} : CM_k \times T \rightarrow [0, 1]$ by multiplying the $rs(t_i)$ of each $t_i \in T$ with $f_{m;k}(cm_j, t_i)$ for all $cm_j \in CM_k$, if cm_j does not apply to threat t_i, we define $f_{m;k} = 1$, i.e. it leaves $rs(t_i)$ unchanged; if cm_j can fully mitigate threat t_i, we define $f_{m;k} = 0$.

The mitigation factor $f_{m;k}$ is a measurement for the reduction in *likelihood* after applying a security countermeasure to a threat. For instance, it is much more difficult to perform an *eavesdropping* attack after end-to-end encryption than on plaintext. For a single threat, two factors influence the risk of a threat, *likelihood LH* and *impact factor IF*, according to Eq. (3). The impact stays the same and the *likelihood* is reduced by $f_{m;k}(cm_j, t_i)$. That's why $f_{m;k}(cm_j, t_i)$ is serving as a scale factor of original threat risk score, subject to constraint $0 \leq f_{m;k}(cm_j, t_i) \leq 1$.

Security countermeasures $cm_j \in CM_k$ and identified threat t_i jointly determine the value of $f_{m;k}(cm_j, t_i)$. In DDM applications, monitoring techniques usually play a vital role to detect policy breaches. Hence, we classify the security countermeasures into two categories, namely prevention countermeasures and detection countermeasures. Prevention countermeasures are those security mechanisms aiming to stop an attack from occurring and prevent a policy breach, e.g. data access control and cryptographic mechanisms. Detection countermeasures are those aiming to detect any attacks or policy breaches during the data exchange procedures, e.g. system call monitoring. The *mitigation factors* in our risk assessment system for countermeasures that apply to a threat are defined as:

$$f_{m;k}(cm_j, t_i) = \begin{cases} 0, & \text{if } t_i \text{ is prevented by } cm_j \\ R_d, & \text{if } t_i \text{ is detected by } cm_j \end{cases}$$

R_d denotes the real time detection rate of the applied monitoring technologies. R_d is provided in the DDM countermeasure database offered by the DDM providers. Normally, DDM providers can achieve the estimated detection rate from IDS designers according to experimental evaluations. It is also possible for DDM providers to adjust R_d of a concrete countermeasure based on the historical data when apply to other DDMs.

For security countermeasures that prevent a threat, we assume the threat can be adequately mitigated and set the value as 0. It is also possible to recalculate *risk attributes* after applying the countermeasure according to Table 1 and determine the corresponding $f_{m;k}(cm_j, t_i)$. For security countermeasures that detect an intrusion, the *mitigation factor* is equal to the accuracy rate, denoted as R_d, of the implemented monitoring detection and algorithm. The value of R_d is typically gained with the historical data.

If multiple countermeasures are matched to a single threat, we need to consider interactions and redundancy among those security countermeasures when determining the joint mitigation level. The multiple security countermeasures are chained and the *joint mitigation factor* is calculated as:

$$F_{m;k}(t_i) = \prod_{j=1}^{N_k} f_{m;k}(cm_j, t_i) \tag{5}$$

$F_{m;k}(t_i)$ is the joint mitigation factor of threat t_i, $F_{m;k}(t_i) \in [0, 1]$. N_k denotes the total number of security countermeasures for a threat t_i in CM_k and $f_{m;k}(cm_j, t_i)$ denotes the mitigation factor of countermeasure cm_j to threat t_i.

6.2 Total risk level of an application

The remaining risk of a threat after mitigation by DDM_k is computed as:

$$rr_{remain;k}(t_i) = rr(t_i) \cdot F_{m;k}(t_i) \tag{6}$$

$rr_{remain;k}(t_i)$ denotes the remaining risk of threat t_i after applying security countermeasures of DDM_k; $rr(t_i)$ denotes the original *risk ratio* of threat t_i.

The *risk level RL* of an application A provided by DDM_k is calculated as the summation of the remaining risk $rr_{remain;k}$ of all threats:

$$RL(A, DDM_k) = \sum_{t_i \in T} rr_{remain;k}(t_i) \tag{7}$$

Module III of the risk assessment system computes the risk levels for potential DDM providers and provides the rankings to DDM customers.

7 System stability due to subjective choices

Most risk assessment systems suffer from the problem of being too subjective. In this section, we investigate how the system ranking results fluctuate due to the subjective choices of the parameters. We call this the *stability* of the risk assessment system.

The subjective choices mainly occur in module II. As mentioned in Sect. 5, the STRIDE/DREAD model maps the three qualitative levels of each *risk attribute*, namely low, medium, and high, into three numeric values [0, 5, 10] with a bijective function. A function $f : X \rightarrow Y$ is bijective, if for all $y \in Y$, there is a unique $x \in X$ such that $f(x) = y$. The numeric combination of [0, 5, 10] indicates an equal risk increase between adjacent qualitative levels for all *risk attributes*, which fits the majority of risk assessment scenarios. However, it is also possible and reasonable to adopt other numeric values, e.g. [0, 1, 2] or even [1, 3, 8] entailing non-equalised risk increase. In the following, we will explore two questions: (i) To which degree can numeric values be chosen objectively depending on system physical effects? (ii) How these chosen numeric values relate to the system output, which is the risk rankings of DDMs.

7.1 Physical effect of value vectors

We put the numeric values in a three-dimensional vector and name it as a *value vector*.

Every threat has a tangible effect on the system. With physical effect, we mean the measurable effects of the threat *risk attribute* values on the components in DDMs. Different *value vectors* express different physical effects. A *value vector* determines the quantitative risk increase between subsequent qualitative levels of each *risk attribute*, as explained in Table 1. For instance, the *risk attribute* accessibility has three levels, which are only by consortium parties, by both consortium parities and third party and by outsiders. If the system adopts a *value vector* [0, 5, 10], it means the risk level increases in equal steps as increasing qualitative levels. However, a *value vector* [1, 3, 8] implies that there is a higher risk increase from medium to high than from low to medium. This higher increase is because an attack from outsiders is considered more serious.

The choice of *value vector* should, in the first place, be determined by how the risk is supposed to increase between subsequent qualitative levels. We can classify those *value vectors* into two categories, namely evenly spaced and non-evenly spaced *value vectors*. Evenly spaced *value vectors* indicate equal steps in risk increase between adjacent levels. If we represent a *value vector* as $[v_{i,1}, v_{i,2}, v_{i,3}]$, an evenly spaced *value vector* is a three-term arithmetic progression $v_{i1} = a$, $v_{i2} = v_{i1} + \delta$, $v_{i3} = v_{i1} + 2\delta$. These evenly spaced *value vectors* are more interesting for us because they fit for most scenarios and share the same physical effect of the original *value vector* from the Microsoft STRIDE/DREAD model. Non-evenly spaced *value vectors* include some distortion and have different steps between neighbouring *risk attribute* levels. If one opts for an evenly spaced *value vector*, there are still many choices having the same physical effect, e.g. [0, 5, 10] versus [0, 1, 2]. The decision of which one to choose exactly turns to be subjective. So it is important to validate the methodology stability with distinct *value vectors* of similar physical effect. Particularly, we would like to investigate the system stability for the DL4LD use case.

We define a metric *Spreading Level SL* to characterise different *value vectors*. Those *value vectors* indicating similar physical effect should have the same *SL*. The *spreading level* of a *value vector* $\mathbf{v}_i = [v_{i,1}, v_{i,2}, v_{i,3}]$ is calculated as:

$$SL(\mathbf{v}_i) = (v_{i,2} - v_{i,1}) - (v_{i,3} - v_{i,2}) \tag{8}$$

7.2 Metrics definition

In this section, we further explore how the subjectively chosen parameters, *value vectors*, influence the output of the risk assessment system.

As introduced in Sect. 5, module II computes the application dependent risk of each threat with the modified STRIDE/DREAD model and calculates the *risk ratios* all the threats in the approved threat list. Obviously, both the threat risk and *risk ratios* are varying with the chosen *value vector*. Those fluctuated *risk ratios* further flow into module III in the system for threat mitigation. The security countermea-

sures and *risk ratios* jointly determine the DDM rankings. In the ideal scenario, the risk assessment system would always generate the same ranking for DDMs for a given application regardless of subjective choices. Absolute values of *risk ratios* play a vital role.

Also, most users of the DREAD/STRIDE model, or our modified version, are focused on the rankings of threats in terms of their risk. We expect stable ranking results for those value vectors with the same physical effect. So we investigate the variance of threat risk rankings caused by a subjectively chosen *value vector*.

Two metrics are adopted to quantify the variance of *risk ratios* with various *value vectors*: Kendall's Tau and normalised mean square error (NMSE).

7.2.1 Kendall's Tau

We are able to rank the threats in terms of risk according to their *risk ratios*. Kendall's Tau is one of the commonly used metrics to measure the similarity of two rankings [11]. We use it to measure the stability between threat rankings of different adopted *value vectors*.

The definition is as follows:

$$\tau(T_x, T_y) = \frac{\#\text{conc pairs}(T_x, T_y) - \#\text{disc pairs}(T_x, T_y)}{\binom{N}{2}}, \tag{9}$$

where T_x represents a threat ranking according to *risk ratios* with *value vectors* $\mathbf{v_x}$ and T_y represents a threat ranking according to *risk ratios* with *value vector* $\mathbf{v_y}$, and N denotes the total number of threats in the list. This leads to a set of $\binom{N}{2}$ pairs. For any pair of *value vectors* $\mathbf{v_x}$ and $\mathbf{v_y}$, we calculate the *risk ratios* for the N threats: $(rr_x(t_i), rr_y(t_i))$, where $rr_x(t_i)$ is the *risk ratio* of threat t_i in T_x with $\mathbf{v_x}$ and $rr_y(t_i)$ is the *risk ratio* of threat t_i in T_y with $\mathbf{v_y}$. *#conc pairs* denotes the number of threat pairs that are concordant in both rankings T_x and T_y, and *#disc pairs* denotes the number of threat pairs that are discordant in both rankings. Threats t_i and t_j are considered a concordant pair if $rr_x(t_i) \leq rr_x(t_j), rr_y(t_i) \leq rr_x(t_j)$. Otherwise, they are considered as a discordant pair.

7.2.2 Normalised mean square error

We choose the metric normalised mean square error (NMSE) to quantify the variance of *risk ratios* due to different *value vectors* [10]. The reason we choose NMSE rather than other metrics are twofold. On the one hand, NMSE is sensitive to outliers. On the other hand, the results are not influenced by absolute values after normalisation. The definition of NMSE is as follows:

$$RR_x = \{{rr_1}^{(x)}, {rr_2}^{(x)}, {rr_3}^{(x)}, \ldots, {rr_N}^{(x)}\} \tag{10}$$

$$\overline{RR_x} = \frac{1}{N}\sum_i rr_i^{(x)} \tag{11}$$

$$\overline{RR_y} = \frac{1}{N}\sum_i rr_i^{(y)} \tag{12}$$

$$NMSE(RR_x, RR_y) = \frac{1}{N}\sum_N \frac{(rr_i^{(x)} - rr_i^{(y)})^2}{\overline{RR_x}\cdot\overline{RR_y}} \tag{13}$$

RR_x denotes the *risk ratios* of N threats using *value vector* $\mathbf{v}_x$. The *risk ratio* of the ith threat with *value vector* $\mathbf{v}_x$ is denoted by ${rr_i}^{(x)}$. $\overline{RR_x}$ denotes the average of all *risk ratios* in RR_x.

8 Experimental validation of system stability

In this section, we validate the stability of the risk assessment system. Here, we focus on the stability of *risk ratios* because they influence the stability of DDM rankings of the risk assessment system. We compute and analyse the values of Kendall's Tau and NMSE of varying *value vectors* under different experimental settings.

8.1 Experimental design

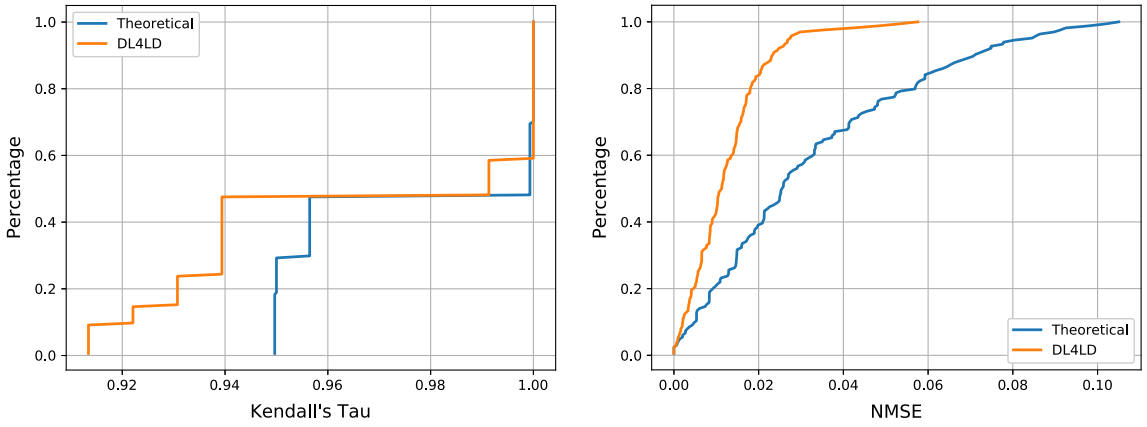
In the experimental validation, we consider *value vectors* $\mathbf{v}_i = [v_{i,1}, v_{i,2}, v_{i,2}]$ with $v_{i,j} \in \{0, 1, \ldots, 10\}$. We construct a set V_{total} of 165 different *value vectors*. In particular, the *value vector* used by the original Microsoft DREAD model is called the baseline *value vector* $\mathbf{v}_{base}$, in our case $\mathbf{v}_{base} = [0, 5, 10]$.

8.1.1 Experiment A

In this experiment, we aim to explore the sensitivity of the threat risk rankings to the applied *value vectors* in a general sense. We compute the two metrics, Kendall's Tau and NMSE, between *risk ratios* for any *value vector* $\mathbf{v}_i$ in V_{total} and for $\mathbf{v}_{base}$, i.e. $\tau(T_x, T_y)$ and $NMSE(RR_x, RR_y)$ where $\mathbf{v}_x \in V_{total}$ and $\mathbf{v}_y = \mathbf{v}_{base}$. This results in a set of Kendall's Tau values and a set of NMSE values. The size of each set is equal to the size of V_{total}.

8.1.2 Experiment B

In this experiment, we aim to evaluate the fluctuations of threat risk rankings among *value vectors* of similar physical effect. According to the discussion in Sect. 7.1, those *value vectors* with similar physical effect should have the same *spreading level*. Hence, we partition all the *value vectors* in set V_{total} in groups with equal *SL*. We calculate the two met-



(a) Cumulative Distribution Function of Kendall's Tau values

(b) Cumulative Distribution Function of NMSE values

Fig. 6 CDF of Kendall's Tau values and NMSE between all *value vector* and the baseline *value vector* [0, 5, 10] for both the theoretical and the DL4LD threat database

rics, Kendall's Tau and NMSE, for any pair of *value vectors* in each equal *SL* cluster, i.e. $\tau(T_x, T_y)$ and $NMSE(RR_x, RR_y)$ for the *value vectors* $\mathbf{v}_x, \mathbf{v}_y \in V_{total}$ with $\mathbf{v}_x \neq \mathbf{v}_y$ and $SL(\mathbf{v}_x) = SL(\mathbf{v}_y)$. In this way, we can achieve the variation of system outputs due to the subjective choice of *value vectors*.

8.2 Experimental threat database

We need to construct proper threat databases to compute and analyse *risk ratios* of a threat set. For simulation purposes, the assigned values of the five *risk attributes* described in Table 1 can uniquely identify each threat. In the current experiment, we consider two threat databases, namely the theoretical threat database and the DL4LD threat database. The goal of the DL4LD project is to help the Dutch logistics sector with IT tools that promotes digital business processes, with particular support for the trustworthy sharing of sensitive data. The project DL4LD aims to facilitate secure and trustworthy data sharing with the concept of digital data marketplaces (DDM) [23].

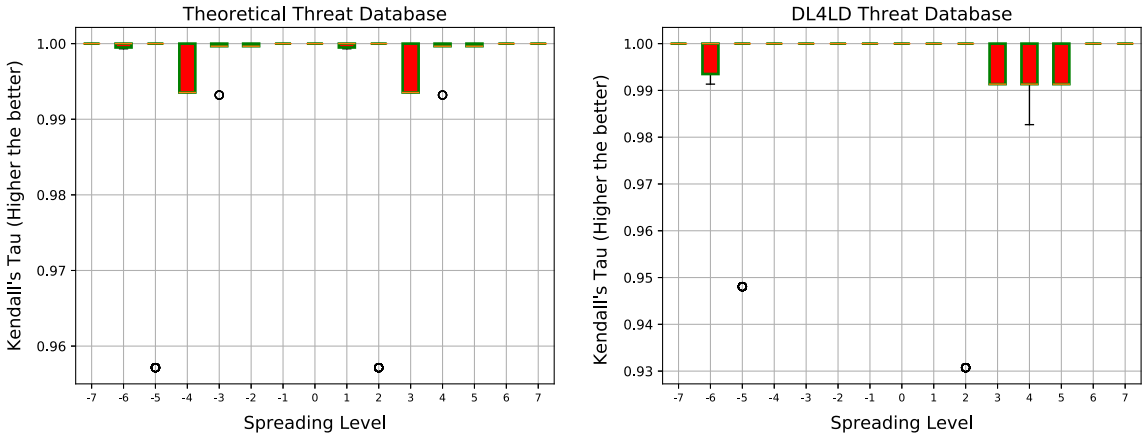
For the theoretical threat database, we consider all possible combinations of five *risk attributes*, each of which can be one of the three values in a *value vector*. The total number of threats in this database is 3^5 (243). Obviously, any real-world threat database, like the DL4LD threat database, is a subset of the theoretical threat database.

There are seven different archetypes defined for DL4LD data exchange scenarios [2]. We model the threats for those archetypes and construct the DL4LD threat database. There are in total 22 threats for all archetypes in the DL4LD threat database. For each threat, we read the related literature and determined the levels of five *risk attributes*.

8.3 Analysis of Kendall's Tau values

We compute Kendall's Tau values between threat risk rankings generated by the baseline *value vector* [0, 5, 10] and any arbitrary *value vector* in set V_{total} for both theoretical and DL4LD threat database. For each database, we rank all threats according to their *risk ratios* computed in Eq. (4).

Figure 6a shows the Cumulative Distribution Function (CDF) of those Kendall's Tau values for both theoretical and DL4LD threat database. For the theoretical threat database, all of the *value vectors* contribute to Kendall's Tau values higher than 0.95, and 50% of the *value vectors* have Kendall's Tau values higher than 0.99. We conclude that the threat risk ranking is almost stable for all *value vectors* in the theoretical database. For the DL4LD threat database, approximately 50% of the *value vectors* have Kendall's Tau values higher than 0.99, which is similar to the DL4LD use case. But another half have Kendall's Tau values between 0.91 and 0.94, which are lower than the minimum value for the theoretical threat database. The comparatively larger ranking variance for the DL4LD use case may be due to the characteristics of the DL4LD threat database. For two threats with higher diversity of *risk attributes* levels, their rankings are likely to flip with different *value vectors*. For instance, if we have two threats with *risk attributes* [L, L, L, L, L] and [H, H, H, H, H], the ranking will never alter no matter how you change the adopted *value vectors*, because [L, L, L, L, L] will always have the lowest *risk ratio* and [H, H, H, H, H] will always have the highest. The rank of threats with *risk attributes* [L, H, L, M, L] and [M, L, M, H, M] most likely will flip after changing the *value vectors*. A higher proportion of such sensitive threats exists in the DL4LD threat database than in the theoretical threat database.



(a) Box plots of Kendall's Tau in theoretical threat database

(b) Box plots of Kendall's Tau in DL4LD threat database

Fig. 7 All value vectors in set V_{total} are grouped with identical *spreading level* ranging from -7 to 7. Each value vector indicates a bijective mapping from qualitative levels of risk attributes to numeric representations. For each value vector cluster with identical *spreading level*, we compute Kendall's Tau values between any pairs of value vectors and plot them as a box

Figure 7a, b shows the box plot for Kendall's Tau values as a function of *SL* for the theoretical and the DL4LD threat database, respectively. The Kendall's Tau values are computed according to *Experimental Design B* in Sect. 8.1. All the *value vectors* in set V_{total} are grouped with equal *SL*. Each box depicts the Kendall's Tau values computed between all possible pairs of *value vectors* within an equal *SL* group.

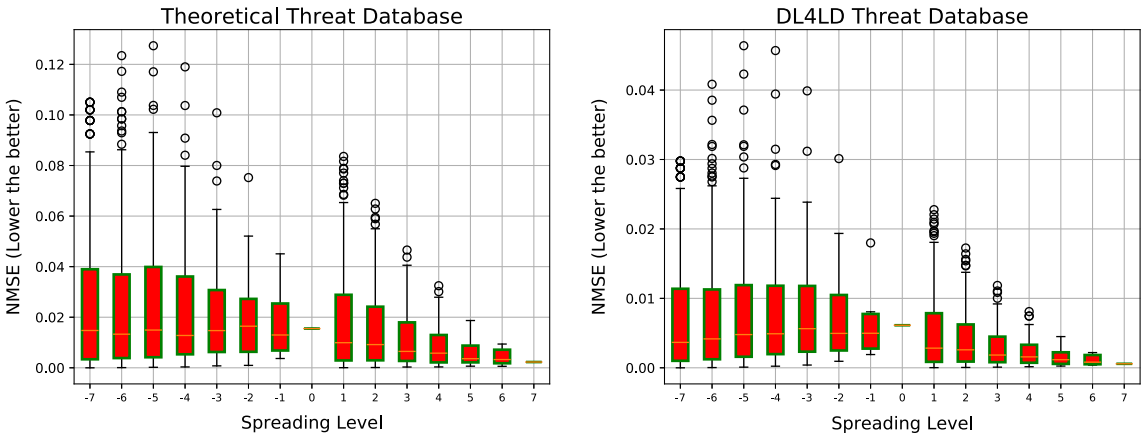
Figure 7a shows Kendall's Tau values among threats rankings for the theoretical threat database. We specifically focus on evenly spaced *value vectors* because they are most commonly used in reality. The Kendall's Tau values of evenly spaced *value vectors* ($SL = 0$) are all equal to 1. A subjectively chosen *value vector* with *SL* equals to 0 does not influence the risk ranking of all theoretical threats. Also, since all the real-world threat databases, e.g. DL4LD threat database, are a subset of the theoretical database, the Kendall's Tau values among evenly spaced *value vectors* should always be 1 for any threat database. The results illustrated in Fig. 7b confirm this conclusion. More generally, as shown in Fig. 7a, nine out of 15 boxes have all values extremely close to 1, whereas four boxes have a slightly higher degree of dispersion, but the minimums are still larger than 0.99. Only two outliers around 0.955 occur for boxes $SL = -5$ and $SL = 2$, respectively. Subjective choices of *value vectors* having the same *spreading level* do not cause the risk rankings to fluctuate. As the theoretical database includes any real-world threat database, we may expect similar high stability achieved in any other threat database, e.g. DL4LD. Figure 7b shows the box plots for the DL4LD threat database. Similarly, most *value vector* clusters have Kendall's Tau values very close or all equal to 1. But the worst case, the two outliers in boxes with $SL = -5, 2$, have comparatively higher variance than for the theoretical database.

For both the theoretical and the DL4LD use case, there is almost no or neglectable influence due to the subjective choices of *value vectors* having the same physical effect (*spreading level*).

8.4 Analysis of normalised mean square error (NMSE)

The metric NMSE describes the variance of absolute values of *risk ratios* with different *value vectors*, which have a direct impact on final DDM exposure rankings. To explore the general sensitivity of *risk ratio* values to varying *value vectors*, we compute NMSE values between the baseline *value vector* [0, 5, 10] and any *value vectors* in set V_{total}.

Figure 6b shows the cumulative distribution function (CDF) of NMSE values computed with all *value vectors* in set V_{total} for both the theoretical and the DL4LD threat database. For the theoretical threat database, approximately 50% of the *value vectors* result in an NMSE value smaller than 0.03 compared with the baseline *value vector*. An NMSE value of 0.03 means that the average shift between two data sets, *risk ratios* with two *value vectors*, is 3% of the product of mean values of the two data sets. For some specific *value vectors*, *risk ratios* vary unneglectable comparing to those computed with the baseline *value vector*. About 18% of *value vectors* in set V_{total} result in an NMSE value higher than 0.06, and the maximum value is 0.1. This is due to the nonlinear mappings from qualitative levels to quantitative numbers in module II. However, the *risk ratios* are much less sensitive for threats in the DL4LD use case. Also shown in Fig. 6b, approximately 95%



(a) Box plots of NMSE in theoretical threat database

(b) Box plots of NMSE in DL4LD threat database

Fig. 8 All *value vectors* in set V_{total} are grouped with identical *spreading level* ranging from −7 to 7. Each value vector indicates a bijective mapping from qualitative levels of risk attributes to numeric representations. For each value vector cluster with identical *spreading level*, we compute NMSE values between any pairs of *value vectors* and plot them as a box

of the *value vectors* in set V_{total} have NMSE values smaller than 0.03 for the baseline *value vector*. The maximum value of NMSE is only 0.06. One reasonable explanation is that absolute larger differences of *risk ratios* normally occur for threats with a smaller risk attributes level diversity, e.g. [L, L, L, L, L] and [H, H, H, H, H]. Such threats are not frequently included in the DL4LD threat database or any other real-world threat database. Hence, we may expect the risk assessment system is quite robust against subjective choices of *value vectors* for the majority of use cases.

Figure 8 shows the box plots of NMSE values as a function of *SL* for both the theoretical and the DL4LD threat database. For *value vectors* of the same *SL*, we calculate NMSE values of *risk ratios* with every two *value vectors* in the group.

We first analyse stability for evenly spaced *value vectors*. Shown in Fig. 8a, b, the dispersion degrees of boxes for $SL = 0$ are very small. The pairwise NMSE values among evenly spaced *value vectors* for both threat databases are concentrated in the medians of the boxes, which are about 0.015, and 0.008 respectively. There are no outliers of relatively higher NMSE values. These NMSE values imply that the system is highly stable to subjective choices for *value vectors* with linear mappings from *risk attribute* qualitative levels to numeric representations.

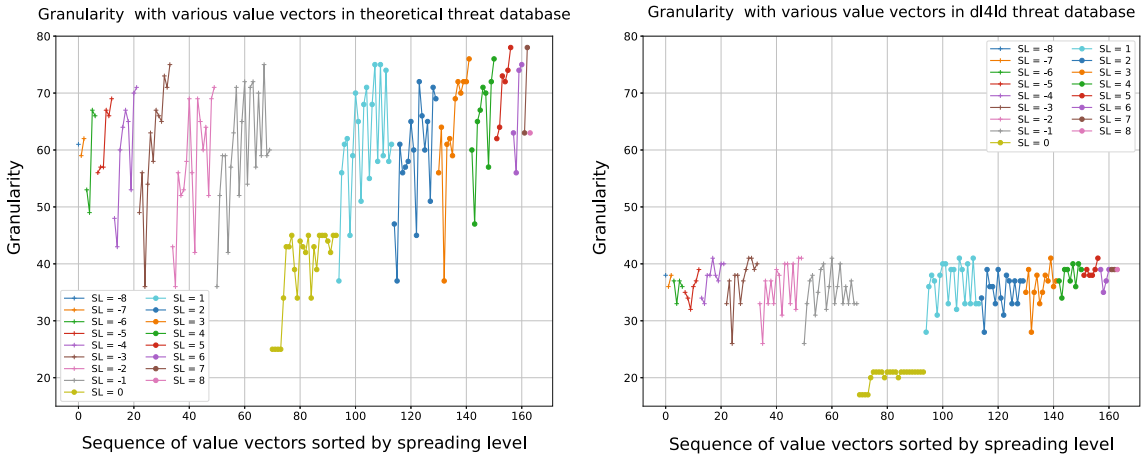
Figure 8a shows the box plots for the theoretical threat database. Each box has a relatively high degree of dispersion, and the median value is around 0.01. An NMSE value of 0.01 is quite acceptable and has a relatively small probability of causing a ranking flip for DDMs in the final output of our system. The NMSE values in Fig. 8a indicate that the ranking is stable for about 50% of *value vectors* for each equal *SL* cluster. However, outliers from 0.07 to 0.13 occur in most *value vector* clusters, especially for those with negative *SL* values. However, the system stability is much higher for the DL4LD use case shown in Fig. 8b. All the boxes for the DL4LD threat database have median values of 0.005, which are much smaller than that of the theoretical threat database. Furthermore, the outliers are much acceptable, with the maximum value smaller than 0.05. The DL4LD use case is very robust to *value vector* variance.

9 Experimental validation of system resolution

In this section, we aim to validate the achieved resolution of our methodology provided by the output of module II, which are *risk scores*. We define a metric of granularity to measure resolution quantitatively. We try to investigate how the chosen *value vectors* influence the system resolution for both theoretical and DL4LD threat database. In addition, we also explore whether the current methodology can provide sufficient granularity for identified threats in the DL4LD use case.

9.1 Definition of granularity and experimental design

The metric granularity aims to evaluate the resolution of our methodology in module II. Granularity is defined as *the total number of unique values of risk scores for a given threat database*. This metric describes the capability of distinguishing between threats in terms of assessed risk. It is usually not



(a) Granularity of threats in the theoretical database with various *values vectors*

(b) Granularity of threats in the DL4LD database with various *values vectors*

Fig. 9 Values of granularity with varying *value vectors* for both the theoretical and the DL4LD threat database. The value vectors are firstly sorted with increasing *spreading level*. For those with equal *spreading level*, the *value vectors* are in lexicographic sorted order

expectable that many threats result in the same risk level, which is equal to the value of computed *risk score*.

We adopt the same *value vector* set V_{total} as described in Sect. 8.1. We compute granularity with each *value vector* in V_{total} for both theoretical and DL4LD threat database. As mentioned in Sect. 8.1, the number of threats is 243 in the theoretical database and 22 in the DL4LD threat database. We also consider the influence of assigned *impact factor*, which has been explained in Sect. 4.2.2, on the resulting system resolution. Any of the five *impact factors* scales a threat and the *risk score* of that threat is scaled accordingly.

9.2 Analysis of granularity Values

Figure 9a, b shows the values of granularity with various *value vectors* in the theoretical threat database and the DL4LD threat database, respectively, with the *spreading levels* depicted in different colours.

Firstly, we investigate the relationship between achieved granularity and *spreading level* of *value vectors*. For both threat databases shown in Fig. 9, non-evenly spaced *value vectors* ($SL \neq 0$) normally gain much better resolution than evenly spaced ones ($SL = 0$). Also, the *value vectors* of different *spreading level* have a similar range of granularity for all non-evenly spaced *value vectors*. Based on the conclusion drawn in Sect. 8, those evenly spaced *value vectors* normally have comparatively higher stability. A higher resolution is achieved at the sacrifice of system stability.

The values of granularity fluctuate for *value vectors* of identical *SL*. We recommend DDM customers to choose a *value vector* with relatively high granularity and to avoid those with very low resolution. As shown in Fig. 9a, b, the relative relations of granularity values among same *value vectors* for both threat databases are similar. For each group with equal *SL*, there is a *value vector* resulting in a low granularity value. Those *value vectors* are sorted in lexicographic order for an equal-SL cluster. Hence, those *value vectors* with the first element as 0 contribute to relative worse resolution compared with those of similar physical effect. According to the above observation and discussion, our system can warn system users when they use such *value vectors*.

We further discuss the absolute values of granularity for both threat databases. As shown in Fig. 9a, evenly spaced *value vectors* have granularity between 25 and 45. For a threat database of 243 threats with five different *impact factors*, on average 27 threats may result in the same risk level even adopting the *value vector* with the highest resolution. The performance for non-evenly spaced *value vectors* is better with granularity values between 35 to 80. Nevertheless, there are still on average 15 different threats that are not distinguishable for their risk with the current methodology in the theoretical threat database. As indicated in Fig. 9b, the DL4LD use case performs very well regarding the small size of the threat database of only 22 threats, each of which may have five different *impact factors*. The granularity values are around 20 for evenly spaced *value vectors* and vary from 24 to 45 for non-evenly spaced ones. Compared to the theoretical threat database, the DL4LD threat database can achieve approximately half of the granularity with only one-tenth of the threats number. The discussion above indicates the system provides sufficient resolution to distinguish threats in the DL4LD use case.

10 Conclusions and future works

Customers of DDMs, or other digital infrastructures, need to know what is the risk level associated with running their applications in any specific DDM. Hence, we propose a broker-based risk assessment system to quantitatively assess the risk level. This system allows customers to rank available digital infrastructures in terms of guaranteed security and select the optimal one regarding their applications.

To increase transparency, the system runs on a trusted third party and collaborates interactively with all involved parties. It addresses the complexity of DDMs by considering a number of influencing factors, such as application *archetypes*, security requests of DDM customers, potential trust among collaborating parties, interactions of security countermeasures. Our proposed system considers the relative importance of each threat and is able to capture the dynamic feature of risk levels in data exchange applications in DDMs.

Furthermore, we validated the stability and resolution of the Microsoft STRIDE/DREAD model in our risk assessment system with a concrete DDM use case, DL4LD. Our experimental results show that subjective choices of users have a very subtle influence on the final DDM rankings of the system. In addition, the risk assessment system provides sufficient resolution and works very well in terms of stability for the DL4LD use case.

In our future work, we will further consider performance cost and optimise the matching procedure between countermeasures and threats in module II. We want to improve our risk assessment system to be adaptive. The applied countermeasures, monitors, can be distributed accordingly with the real-time measured risk level.

Acknowledgements This paper builds upon the work done within the Dutch NWO Research project 'Data Logistics for Logistics Data' (DL4LD, www.dl4ld.net), supported by the Dutch Top consortia for Knowledge and Innovation 'Institute for Advanced Logistics' (TKI Dinalog, www.dinalog.nl) of the Ministry of Economy and Environment in The Netherlands and the Dutch Commit-to-Data initiative (https://commit2data.nl/).

Funding This work is funded by The Dutch Research Council (NWO). The NWO grant number for the DL4LD-project is: 628.009.001.

Declarations

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

Appendix

Appendix A: Threat analysis of digital marketplaces (DDMs)

The identified threats and the corresponding information for data exchange applications in DDMs are shown in Table 2.

Table 2 Threat list of data federation applications in DDMs

Threat name	Stage	Category	Risk attributes				
			DP	AC	SL	AU	D
IP spoofing	II	S	SO	H	M	H	M
Identity spoofing: Remote Data Access	III	S	SO	H	L	M	H
Insecure data deletion	III	ID	SO	M	L	M	H
Malicious compute: Data Disclosure	III	ID	SO	L	H	H	M
Unauthorized Disclosure: Eavesdropping	II	ID	SO	H	H	M	H
Weak Access Control	I	ID	SO	H	H	L	H
Malicious compute: High Correlation of Input and Output Data	III	ID	SO	L	H	H	M
Encryption Keys Leakage during Exchange	II	ID	TOP	H	L	H	H
Cross-tenant Side Channel Attack	III	ID	SO	M	L	H	H
Management Interface Compromise	I, III	ID, T	SO	M	M	M	M
Isolation Failure: Poorly Separated Container Traffic	III	ID	SO	L	L	H	H
Isolation Failure: Cross Container Attack	III	ID	SO	M	L	H	H
Insecure Running Environment	III	ID	SO	M	L	H	H
Man-in-the-Middle	II	T	SO	H	M	M	L
Malicious compute: Tamper Processed Data	III	T	SO	L	H	H	L
Log Files Tampering: illegal members delete or modify log files	I, II, III	T	TOP	L	L	H	L
Data Leakage/Loss	I	T	SO	H	L	M	L
Not-trustable Computing Environment	III	T, ID	SO	M	M	H	L
Denial of Service (DoS) Attack by Co-tenant Containers	III	DoS	SO	L	H	H	L
Container Runtime Escape	III	EP	SO	L	M	H	M
Repudiation Attacks	II	R	SO	M	L	H	L
Insufficient Auditing	II	R	SO	L	H	M	H

For each threat, we assign corresponding *Stage*, *Category* and *Risk Attributes* according to literatures. 'H', 'M', 'L' represent 'High', 'Medium', 'Low' respectively. 'SO' stands for the 'sensitivity of the object'. 'TOP' stands for the highest level of 'sensitivity of the object'

References

1. Sagiroglu, S., Sinanc, D.: Big data: a review. In: 2013 International Conference on Collaboration Technologies and Systems (CTS), pp. 42–47. IEEE (2013)
2. Zhang, L., Cushing, R., Gommans, L., De Laat, C., Grosso, P.: Modeling of collaboration archetypes in digital market places. IEEE Access **7**, 102689–102700 (2019)
3. Luna, J., Ghani, H., Vateva, T., Suri, N.: Quantitative assessment of cloud security level agreements: a case study. In: Proc. of Security and Cryptography, pp. 64–73 (2012)
4. Shaikh, R., Sasikumar, M.: Trust model for measuring security strength of cloud computing service. Procedia Comput. Sci. **45**, 380–389 (2015)
5. Sen, A., Madria, S.: Off-line risk assessment of cloud service provider. In: 2014 IEEE World Congress on Services, pp. 58–65. IEEE (2014)
6. Shivraj, V., Rajan, M., Balamuralidhar, P.: A graph theory based generic risk assessment framework for internet of things (IoT). In: 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pp. 1–6. IEEE (2017)
7. Sicari, S., Rizzardi, A., Miorandi, D., Coen-Porisini, A.: A risk assessment methodology for the Internet of Things. Comput. Commun. **129**, 67–79 (2018)
8. Anand, P., Ryoo, J., Kim, H., Kim, E.: Threat assessment in the cloud environment: a quantitative approach for security pattern selection. In: Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication, pp. 1–8 (2016)
9. Miscosoft: Microsoft security development lifecycle (SDL). https://www.microsoft.com/en-us/securityengineering/sdl/ (2020)
10. Poli, A.A., Cirillo, M.C.: On the use of the normalized mean square error in evaluating dispersion model performance. Atmos. Environ. A. Gen. Top. **27**(15), 2427–2434 (1993)
11. Lindskog, F., McNeil, A., Schmock, U.: Kendall's tau for elliptical distributions. In: Credit Risk, pp. 149–156. Springer, Berlin (2003)
12. Zhang, X., Wuwong, N., Li, H., Zhang, X.: Information security risk management framework for the cloud computing environments. In: 2010 10th IEEE International Conference on Computer and Information Technology, pp. 1328–1334. IEEE (2010)
13. Disterer, G.: ISO/IEC 27000, 27001 and 27002 for information security management
14. Gordon, L.A., Loeb, M.P., Zhou, L.: Integrating cost–benefit analysis into the NIST Cybersecurity Framework via the Gordon–Loeb Model. J. Cybersecur. **6**(1), tyaa005 (2020)
15. Alberts, C., Dorofee, A., Stevens, J., Woody, C.: Introduction to the OCTAVE Approach. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, Tech. Rep (2003)
16. Den Braber, F., Hogganvik, I., Lund, M.S., Stølen, K., Vraalsen, F.: Model-based security analysis in seven steps-a guided tour to the CORAS method. BT Technol. J. **25**(1), 101–117 (2007)
17. Seifert, D., Reza, H.: A security analysis of cyber-physical systems architecture for healthcare. Computers **5**(4), 27 (2016)

18. Cagnazzo, M., Hertlein, M., Holz, T., Pohlmann, N.: Threat modeling for mobile health systems. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 314–319. IEEE (2018)
19. Lundgren, B., Möller, N.: Defining information security. Sci. Eng. Ethics **25**(2), 419–441 (2019)
20. Gao, X., Gu, Z., Kayaalp, M., Pendarakis, D., Wang, H.: ContainerLeaks: emerging security threats of information leakages in container clouds. In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 237–248. IEEE (2017)
21. STRIDE/DREAD, The DREAD approach to threat assessment. https://docs.microsoft.com/en-us/windows-hardware/drivers/driversecurity/threat-modeling-for-drivers (2020)
22. CAPEC: Common Attack Pattern Enumeration and Classification. https://capec.mitre.org/ (2020)
23. DL4LD: Data Logistics for Logistic Data. https://www.dl4ld.nl (2020)