

A modeling approach to cyber threat mitigation

Andrei Chiş^{1,†}, Oliviu Ionuţ Stoica^{2,†} and Ana-Maria Ghiran^{1,*,†}

¹ Babeş-Bolyai University, Faculty of Economics and Business Administration, Cluj-Napoca, Romania

² MassMutual, Cluj-Napoca, Romania

Abstract

Over the past decade, the security issues that are threatening IT systems worldwide gained increased attention. This was due to several factors and affected both enterprises and individuals. In case of enterprises, there is a popular trend among companies to give up on-premises solutions in favor of using cloud services. For both enterprises and individuals, another influential and decisive factor is the imposed legislation (ADPPA in U.S. or GDPR in EU) with respect to data privacy. Given these circumstances, more people/stakeholders should be involved in devising the security of IT systems who should be acquainted with “secure by design” principles. Given that not many of them are specialists in cyber security a solution that would help them in this matter is needed. This paper presents an approach to mitigate the cyber security threats at design phase of a system. Moreover, it can also be used in auditing an existing system. The main idea is to leverage knowledge that is expressed as diagrammatic models (e.g., dataflow diagrams or threat models created with a domain specific modeling language), which can be understood by all stakeholders of a system, both technical and non-technical.

Keywords

security, privacy, dataflow, threat, modeling

1. Introduction

Nowadays, organizations must recognize the inevitability of cyber security incidents and prepare themselves to effectively respond to them. In addition to the increased number of incidents, organizations must also deal with security regulations and new reporting requirements regarding data privacy and their ability to protect customers’ data. Moreover, companies strive to satisfy increasingly higher customer expectations which involve not only delivering the right service or product but also ensuring an adequate infrastructure that enables a prompt and trustful response.

The need for speed and availability of information forced companies to change their information systems from their on-premises solution to external service providers known as cloud services. This creates various benefits for companies, for their clients and their

BIR-WS 2024: BIR 2024 Workshops and Doctoral Consortium, 23rd International Conference on Perspectives in Business Informatics Research (BIR 2024), September 11-13, 2024, Prague, Czech Rep.

* Corresponding author.

† These authors contributed equally.

✉ andrei.chis@econ.ubbcluj.ro (A. Chiş); oliviu.stoica@stud.ubbcluj.ro (O. I. Stoica);

anamaria.ghiran@econ.ubbcluj.ro (A. M. Ghiran)

0000-0003-0173-7250 (A. Chiş); 0000-0001-7890-9386 (A. M. Ghiran)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

employees as it is available at any time and any place. But this trend of using cloud services brings up the need to enforce data loss prevention techniques, imposing policy controls for cloud services. Some of the policy and security measures are implemented by cloud service providers but many of them remain to be handled by the companies contracting such services.

The recommended approach is to develop information systems considering “secure by design” principles. Detecting security issues in early IT system’s design enables cost-efficient fixes. But whether there is the case of migrating an on-premise system to a cloud environment or auditing an existing system on a cloud environment, more people should be involved than the technical team: employees from business’ departments, managers, suppliers and even clients or end users of a system could provide valuable knowledge of how their data should be secured and identifying the risks they might be exposed to. Therefore, a simple domain specific language must be available to be employed by any stakeholder of the system ensuring the knowledge transfer among them.

The remainder of the paper is structured as follows: Section 2 provides an introduction to relevant concepts like Data Flow Diagrams (DFDs) and Threat Modeling followed by a short presentation of related works and an overview of our solution; Section 3 describes a proof-of concept of the presented solution and lastly, we summarize our contributions in conclusions.

2. Problem statement and background

Today a wide range of security tools are available that can be used to scan a system for vulnerabilities and perform an analysis to detect possible mitigation strategies. These tools are indeed extremely efficient in identifying vulnerabilities; however, such tools can only scan a system for known vulnerabilities. These vulnerabilities are publicly described in vulnerability databases, for example NVD (National Vulnerability Database) [1], which represent valuable knowledge sources for everybody, regardless of their intent.

While the security specialists need to identify and address every possible vulnerability of a system, a malicious actor only needs to identify one vulnerability of a system and that system gets compromised. Considering this, the addressed question is whether it is enough to scan for known vulnerabilities (which is done mainly after the system is implemented).

Having this in mind, a better approach is to put prevention first place, more specifically, to develop systems driven by secure by design principles.

This paper presents a solution based on conceptual models for identifying, communicating and understanding threats and mitigations of a system at design time. Our approach uses data flow diagrams to represent the flow of data through business processes, threat modelling to describe possible risks associated with the components of a system. The created models can be integrated with other architectural descriptions of the system enabling a better understanding of their interconnections.

In [2], authors have studied the importance of cyber security and established some parameters of cyber security: threat identification, vulnerability identification, access risk exploration, creating a contingency plan, respond to cyber security incident. The key points

in cyber threat mitigation are “vulnerability identification” and “threat identification” – what the system exposes versus what the system is exposed to.

To identify vulnerabilities, one needs to have knowledge of how the information flows through a system or a cluster of systems. For this, a good representation of how data travels is required. A widely accepted solution is a modeling representation based on a data flow diagram (DFD) [3].

There are established threat methodologies like STRIDE [4], that use DFDs when designing a system to identify those threats that violate security requirements like confidentiality, integrity, availability, authentication and non-repudiation [5]. However, DFDs are mainly used at design time of a systems (as the next sections shows) and lack an explicit connection with other data (i.e. representations of data to be used or processed).





Our approach distinguishes from other approaches that use modeling techniques to represent the security threats and mitigations for a system by enabling the created models to be linked with other data elements that are only available at run time.

2.1. Data Flow Diagrams

A data flow diagram (DFD) [6] is a graphical representation of an IT system but in relation with business processes. It shows the flow of data through different components of the system as well as their interactions. Although DFDs have not been standardized, adopters of DFDs have consistently employed similar concepts in their implementations.

There are four basic concepts in a data flow diagram and their most used graphical representation is shown in Table 1.

Table 1
Data flow diagram concepts and graphical representations

Symbol	Concept Name
	External Entity, User or system
	Data Flow
	Process
	Data store

The adopted definitions of the data flow diagram components are:

A *process* is an activity or a function which transforms data, and it is performed for a specific business-related reason. A *data flow* is a link or connector data between processes, data stores, systems, users or other kind of external entities. A *data store* is a collection of data or information that is stored in a physical device. An *external entity* can be a user, a

person, a system, an organization, or any other kind of entity that is external to the system and interacts with it. Data flow diagrams are widely used in secure by designed analysis, especially in *threat modeling*.

Collecting and storing information in conceptual models a manager, which normally is not a cyber security expert, can conduct an audit of the system with the cyber security department or external security specialists at much ease and speed. Security specialists can provide technical information about the system (e.g. what security measures are needed) while other business executives can provide information about business strategies, business wise or enterprise IT architecture.

Instead of DFD for system's representation, one can use BPMN (Business Process Model and Notation) [7], which also provides symbols for specifying the flow of activities in a process and it includes support for so-called data objects and data store references. However, some authors [8] differentiate between DFD and BPMN as the former is more concerned in capturing the data movement (hence is more suitable to be used during the analysis phase of the systems development life cycle- SDLC), while the latter is more appropriate in describing the activities that need to be executed in a process (hence it is more suitable to be used during the design phase of SDLC).

2.2. Threat modeling

While DFDs provide insights into how data flows through a system, they might not be sufficient on their own to comprehensively address security concerns.

Threat modeling can be seen as an engineering technique that helps to identify, communicate and understand threats and mitigations within the context of protecting a system and its information [9]. A threat model is a structured process with four main objectives: identify security requirements, pinpoint security threats and potential vulnerabilities, quantify threats and vulnerabilities and prioritize remediation methods. It can be seen as a structured representation of all the information that affects the security of a system. In essence, it is a view of the system and its environment through the lens of security. There are various methodologies for conducting threat modeling [10], each with its own strengths and weaknesses.

STRIDE [4] is a threat modeling methodology developed by Microsoft that identifies six types of threats, that provide the name of the methodology:

- Spoofing identity: when an agent impersonates somebody else, for example when using the authentication credentials of someone else
- Tampering with data: when an attacker changes the data during its transit over the network or when the data is at rest on disk storage or memory.
- Repudiation: when an actor denies actions in a system.
- Information disclosure: when an attacker violates confidentiality, getting access to information without authorization or stealing information.
- Denial of service: when an attacker is exhausting a system's resources to interrupt its availability.

- Elevation of privilege: when actions that are not authorized are performed in a system.

An analyst can assign a *set of susceptible threats for each element of the DFD*. For example:

- Spoofing threats are expected to be added for Processes and External Entities components in a DFD
- Tampering threats are affecting Processes, Data Stores and Data Flows,
- Repudiation threats should be defined for Processes, External Entities and Data Stores,
- Information Disclosure threats need to be added for Processes, Data Stores and Data Flows
- Denial of Service threats need to be added for Processes, Data Stores and Data Flows
- Elevation of Privilege threats must be identified for Process components.

In order to easily identify concrete threats for each category, the analysts use the predefined catalogue of security threat trees that is provided by the STRIDE methodology. Then, for each threat an appropriate security risk level must be determined in order to be able to sort them and come up with proper countermeasures.

STRIDE categorizes the threats from the attacker point of view, as opposed to identification and categorization from a defensive perspective which is the focus of Application Security Framework (ASF) [11]. The threats in ASF are Authentication, Authorization, Configuration Management, Data Protection in Storage and Transit, Data/Input Validation, Error Handling & Management, Session Management, Auditing & Logging. From a defensive perspective, these might be considered as threats due to weaknesses that they can introduce in a system. Similarly to STRIDE, each of the threats in the ASF framework might have several mitigation techniques.

The modelers can choose to apply any threat modeling methodology, without being restricted to those previously enumerated. For instance, they might select a more specialized one like LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness and Non-compliance) [12] which is focused on a particular field from cyber security, namely data privacy and confidentiality.

2.3. Related works

Arwa et al. compared [13] data flow diagrams and use case diagrams and concluded that data flow diagrams are more powerful and can be easily included in the object-oriented approach. Use case diagrams can be employed as a first draft between system analysts and customers, then the system analyst can switch to data flow diagrams as a formal modeling of the system.

In another publication [14], the authors investigated whether data flow diagrams are enough for conducting threat modeling. They concluded that although DFDs could be more easily adopted in practice as they employ very few concepts, they lack specialized notions about security concepts, data elements, abstraction level and deployment information. They advocate for the need of a dedicated integrated language for threat modeling. The threat

modeling tool should provide a sufficiently complex language and level of support but, in the same time, should have the “ease of adoption” capability.

Sion presented [15] solution-aware data flow diagrams for security threat modeling. They stated that many current techniques enumerate numerous non applicable threats while they should focus on selecting those that are strictly related to the technological context or the domain (i.e. threat modeling in the software development should take in consideration the already implemented security solutions in the system). On the other hand, having too much information about the domain can mislead the analyzer and can make the engineer biased. To overcome misleading by biasing, they proposed a constant re-assessment of the threats.

Analys on DFDs diagrams could support information flow control or access control and Seifermann et al. [16] proposed an extended DFD syntax that could be used to model both of them. However, their work focused on capturing the logic and less on the visual representation.

An approach related to ours in the sense that considers security by design principles is that of [17]. They provide a set of models annotated with security flaws and propose an automated approach to perform inspection using model query patterns.

Our previous work [18] also considered the possibilities of analysis and detection of vulnerability patterns using knowledge graphs derived from DFD descriptions. While our prior research paid particular attention in identifying semantic relationships and patterns on the generated knowledge graph making it susceptible to machine processing and automated reasoning, in this paper we aim to address the support provided to the human security analysts which requires enhanced visual representations

2.4. Solution overview

The proposed solution is to create a domain specific modeling method. Besides the modeling language for describing concepts needed to capture the security issues, we defined a functionality to calculate the security score of the modelled system. Our proof of concept considers a business-oriented use case given by an online shop, which is very popular among enterprise systems.

Our modeling language groups the new concepts into two types of models. The first category of models includes concepts of the data flow diagram (*Data Flow Diagram Model*). The second model type provides a structural view, inspired from the mind maps, and it describes a threat methodology or a security framework methodology (*Threat Mitigation Model*).

In this paper we demonstrate how model driven development can be used together with data flow diagrams and threat modeling methodologies to mitigate cyber threats and do a security analysis of a system.

There are several threat methodologies that are widely accepted and commonly used in cyber security. In our solution we did not want to limit to a specific one, rather to allow the modeling of any methodology – we provide the possibility to create any methodology for various systems and business cases that exist, choosing STRIDE for our use case.

To implement our proposal, a metamodeling approach was chosen, building upon the foundation laid by ADOxx meta modeling platform [19]. This allows us to create the

concepts and constraints of our two model types by defining a domain specific modeling language. The metamodel is displayed in Figure 1, presenting the basic concepts and their relationships grouped by model types.

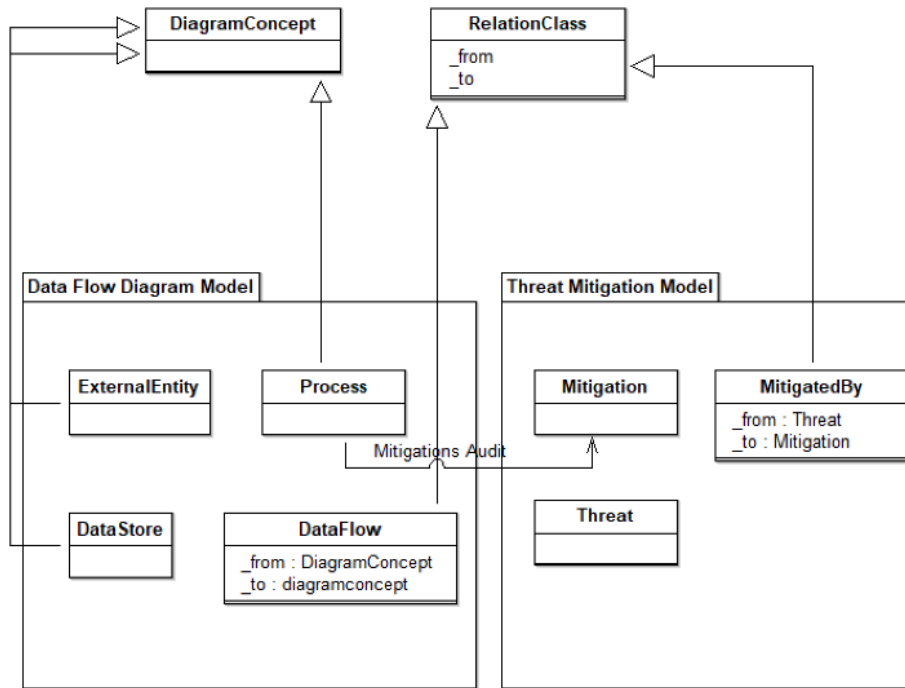


Figure 1: Metamodel of the proposed domain specific modeling language.

The functionality for calculating the security scores for the business process described in the Data Flow Diagram, given the threat methodology described in Threat Mitigation model is implemented as a functionality in the modeling tool via AdoScript [20].

Figure 2 shows an excerpt from this script. The algorithm reads the model’s content and for all objects of type “Process” retrieves the values from the attribute “Mitigations”. Each one is a hyperlink to an object described in the Threat Mitigation Model (STRIDE in our case). Also, all objects of type “Threat” are retrieved and a map with key/value pairs is created. This allows us to quickly asses whether for a specific process, the recorded mitigations are enough by taking into account the recommended mitigation methodology for a specific threat – in our example, described in the next section, the DoLogin process has 2 mitigations (Appropriate authentication and Don’t store secrets) which are among the mitigations endorsed in the Threat Mitigation Model for the Spoofing Identity Threat: therefore, 2 out of 3 (visible in Fig. 5 and 6).

3. Proof-of-concept

In this section we present a running example for the modeling solution that we described above. The first step is to choose an existing methodology and based on it to create a Threat

Mitigation Model. After that, the modeler will describe in a Data flow Diagram the most important entities in a system and how data travels among them.

For this example, the chosen threat methodology, STRIDE, is modeled in Fig. 3.

```
1 CC "Modeling" GET_ACT_MODEL
2 CC "Core" GET_CLASS_ID classname:"Process"
3 CC "Core" GET_ATTR_ID classid:(classid) attrname:"Mitigations Audit"
4 CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(modelid) classname:"Process"
5
6 SET message:""
7 SET mapsInitialized:0
8 SETL tMapScore: (map())
9 SETL tMapMaxScore: (map())
10 SET processObjIds:(objjids)
11
12 FOR processId in:(processObjIds)
13 {
14   CC "Core" GET_OBJ_NAME objid:(VAL processId)
15   SET message:(message + objname + " has the following security scores:\n")
16   CC "Core" GET_ALL_REC_ATTR_ROW_IDS objid:(VAL processId) attrid:(attrid)
17   IF (mapsInitialized = 0)
18   {
19     SET firstrow:(token(rowids,0," "))
20     CC "Core" GET_INTERREF objid:(VAL firstrow) attrname:"Mitigation"
21     CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(tmodelid) classname:"Threat"
22     SET threatIds:(objjids)
23
24     FOR t in:(threatIds)
25     {
26       CC "Core" GET_OBJ_NAME objid:(VAL t)
27       SETL tMapScore[objname]:0
28
29       CC "Core" GET_CONNECTORS objid:(VAL t) out
30       SET max:0
31       FOR id in:(objjids)
32       {
33         SET max:(max +1)
34       }
35       SETL tMapMaxScore[objname):(max)
36     }
37     SET mapsInitialized:1
38   }
39
40   FOR j in:(rowids)
41   {
42     CC "Core" GET_INTERREF objid:(VAL j) attrname:"Mitigation"
43     CC "Core" GET_CONNECTORS objid:(tobjid) in
44
45     SET connectorIds:(objjids)
46     FOR k in:(connectorIds)
47     {
48       CC "Core" GET_CONNECTOR_ENDPOINTS objid:(VAL k)
49       CC "Core" GET_OBJ_NAME objid:(ffromobjid)
50       SETL tMapScore[objname):(tMapScore[objname] + 1)
51     }
52   }
53
54   FOR t in:(threatIds)
55   {
56     CC "Core" GET_OBJ_NAME objid:(VAL t)
57     SET message:(message + objname + " --> " + (STR tMapScore[objname]) + "/" + (STR tMapMaxScore[objname]) + "\n")
58   }
59   SET message:(message+"\n")
60 }
61
62 CC "AdoScript" VIEWBOX text:(message) title:"Security scores per process"
```

Figure 2: AdoScript functionality to compute security scores for business processes.

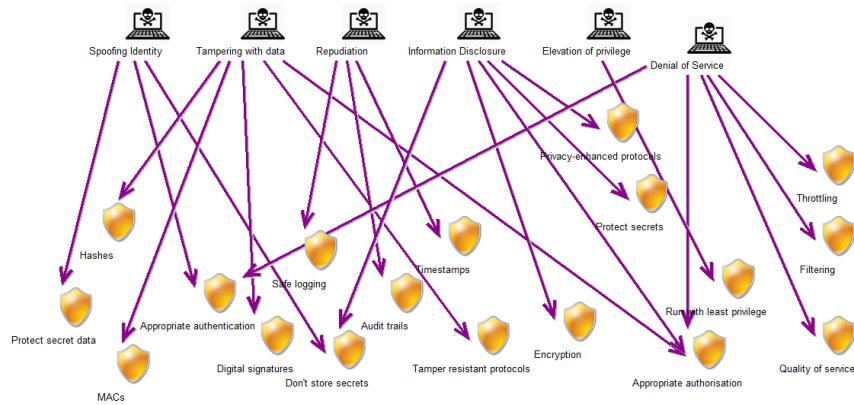


Figure 3: Threat Mitigation Model inspired from STRIDE methodology.

In the next step we create a Data Flow Diagram inspired by the following business use case: a shop’s employee inserts a list of products into the shop’s product database, afterwards a customer logs into the shop’s online platform and browses for products to place an order which will be saved into the shop’s order data base; finally, a manager reads the placed orders from the database. The Data Flow Diagram describing this use case can be observed in Figure 4.

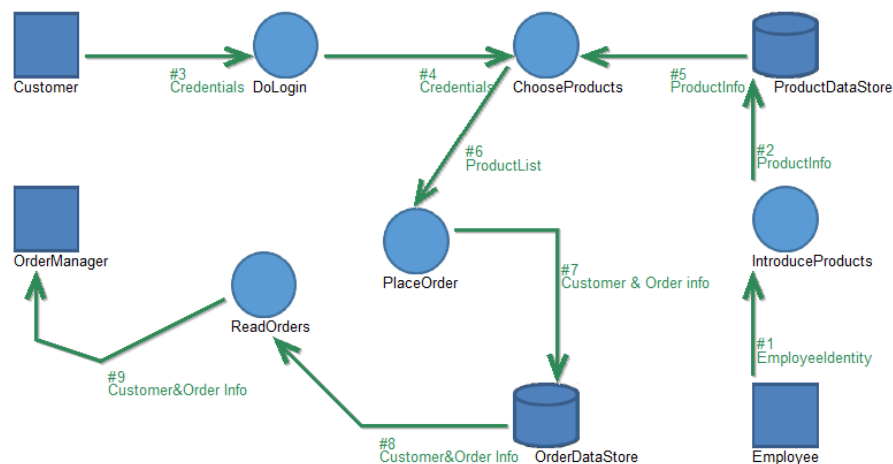


Figure 4: Data Flow Diagram Model describing our example of business use case.

The following DFD elements can be identified:

External entities:

- Customer: Represents the customer who interacts with the online platform to place orders.
- Employee: Represents the shop’s employee who provides the list of products to be inserted.

- OrderManager: Represents the manager who reads the placed orders from the database.

Processes:

- IntroduceProducts: Represents the process where the shop's employee inserts a list of products into the product database. This process takes data input (list of products) and stores it in the product database (Data Store 1: Product DataStore).
- DoLogin: Represents the process where a customer logs in; it takes input the user's credentials and allows access on the platform
- ChooseProducts: Represents the process where a customer browses for products
- Place Order: Represents the process where a customer places an order through the online platform. This process takes input (order details) and stores the order information in the order database (Data Store 2: Order DataStore).
- Read Orders: Represents the process where a manager reads the placed orders from the order database. This process retrieves data (placed orders) from the order database (Data Store 2: Order DataStore).

Data Stores:

- Product DataStore (Data Store 1): Stores the list of products inserted by the employee.
- Order DataStore (Data Store 2): Stores the placed orders made by customers.

Data Flows:

- From Employee to IntroduceProducts: Represents the flow of the list of products from the employee to the process of inserting products.
- From Customer to DoLogin: Represents the order details flow from the customer to the Login process.
- From DoLogin to Choose Products: Represents the flow of data after the customer's login, the verification of credentials and the obtained authorization
- From Choose Products to Place Order: Represents the order details flow from the choice of the customer to the process of placing orders.
- From Place Order to Order Database: Represents the flow of placed orders data from the "Place Order" process to the order database.
- From Order Database to Manager: Represents the flows of placed orders data from the order database to the manager for reading purposes.

On each process, the modeler can select some mitigations similarly as in an audit (Figure 5).

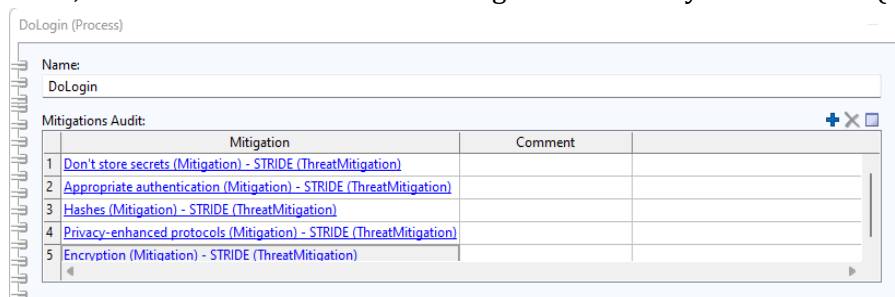


Figure 5: Mitigations for the “DoLogin” process.

Having these models created and the mitigation analysis performed, the security score of the modeled system can now be calculated through the implemented script. A sample of the result is presented in Figure 6.

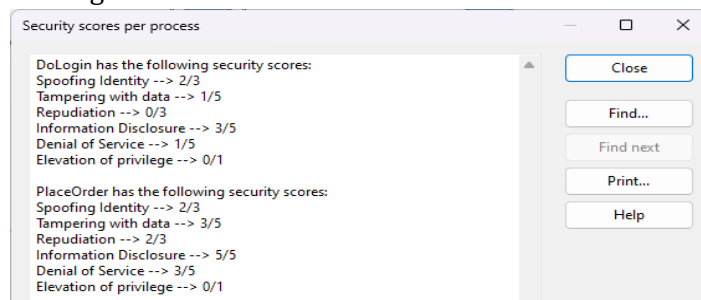


Figure 6: Security scores calculated for the modeled system.

4. Conclusions

In this paper we presented a conceptual modeling approach for threat identification, choosing suitable mitigation techniques by calculating security scores according to domain specific methodologies. We used conceptual model representations to obtain an overview of the system and enable a security analysis: data flow diagrams have been leveraged in conjunction with existing threat modeling methodologies to perform analysis of the cyber system and identify weak points by generating a security score. The generated security score combined with the data flow diagram can be presented as an audit report understandable by all involved stakeholders, technical and non-technical. As future work, the presented solution can be extended by adding new functionalities for each process regarding the audit of external libraries or third party's applications. In a similar vein, our solution can be integrated with external tools specialized on scanning the vulnerabilities of the third party libraries used in the modeled system. Nevertheless, the metamodel can be supplemented with new domain specific concepts to allow modeling a security risk score methodology hence, an improved security analysis

Acknowledgements

This research used infrastructure acquired as part of the project POC/398/1/1/124155 - co-financed by the European Regional Development Fund (ERDF) through the Competitiveness Operational Programme for Romania 2014-2020.

References

- [1] H. Booth, D. Rike, G. A. Witte, The national vulnerability database (nvd): Overview, 2013. URL: <https://www.nist.gov/publications/national-vulnerability-database-nvd-overview>.
- [2] S. Ghuandare, A. Patil, R. Lad, Importance of Cyber Security, International Journal of Engineering Research & Technology, vol 8 – 05, 2020.

- [3] L. Sion, D. Van Landuyt, K. Wuyts, W. Joosen, Privacy risk assessment for data subject-aware threat modeling, In: IEEE Security and Privacy Workshops, pp. 64-71, 2019.
- [4] L. Kohnfelder, P. Garg, The threats to our products. Microsoft Interface, Microsoft Corporation, 33, 1999.
- [5] A. Shostack, Threat modeling: Designing for security. John Wiley & Sons, 2014.
- [6] P. G. Larsen, N. Plat, H. Toeteneel, A formal semantics of data flow diagrams. Formal aspects of Computing, 6, pp.586-606, 1994. doi:10.1007/BF03259387.
- [7] OMG - Object Management Group: Business Process Model and Notation, URL: <https://www.bpmn.org/> Accessed 2023/12/27.
- [8] G.M. Giaglis, A taxonomy of business process modeling and information systems modeling techniques. International Journal of Flexible Manufacturing Systems, 13(2), pp. 209-228, 2001.
- [9] T. UcedaVelez, M.M. Morana, Risk Centric Threat Modeling: process for attack simulation and threat analysis. John Wiley & Sons, 2015.
- [10] S. Hussain, A. Kamal, S. Ahmad, G. Rasool, S. Iqbal, Threat modelling methodologies: a survey. Sci. Int.(Lahore), 26(4), pp.1607-1609 Vancouver, 2014.
- [11] L. Conklin, V. Drake, S. Strittmatter, Z. Braiterman, Threat Modeling Process URL: https://owasp.org/www-community/Threat_Modeling_Process Accessed 2024/01/25.
- [12] K. Wuyts, L. Sion, W. Joosen, Linddun go: A lightweight approach to privacy threat modeling, in: IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 302-309, IEEE, 2020.
- [13] A. Y. Aleryani, Compative Study between Data Flow Diagram and Use Case Diagram, International Journal of Scientific and Research Publications, 6(3), pp.124-126, 2016.
- [14] L. Sion, K. Yskout, D. Van Landuyt, A. van Den Berghe, W. Joosen, Security threat modeling: are data flow diagrams enough?, in: Proceedings of the IEEE/ACM 42nd international conference on software engineering workshops, pp. 254-257, 2020.
- [15] L. Sion, K. Yskout, D. Van Landuyt, W. Joosen, Solution-aware data flow diagrams for security threat modeling. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 1425-1432, 2018.
- [16] S. Seifermann, R. Heinrich, D. Werle, R. Reussner, Detecting violations of access control and information flow policies in data flow diagrams. Journal of Systems and Software, 184, p.111138, 2022.
- [17] K. Tuma, L. Sion, R. Scandariato, K. Yskout, Automating the early detection of security design flaws, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 332-342, 2020.
- [18] A. Chis, I. Stoica, A. M. Ghiran, R. A. Buchmann, A Knowledge Graph Approach to Cyber Threat Mitigation Derived from Data Flow Diagrams, in: IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2024, Cluj-Napoca, Romania, 2024.
- [19] BOC GmbH, The ADOxx metamodeling platform, 2024. URL: <https://www.adoxx.org>. Accessed 2024/08/01.
- [20] BOC GmbH The AdoScript Programming Language, 2024. URL: <https://www.adoxx.org/live/adoscript-language-constructs> Accessed 2024/08/01.