Systematic Literature Review of Threat Modeling Concepts

Pedro Alfradique Lohmann[a], Carlos Albuquerque[b] and Raphael Machado[c]
Computer Science Institute, Fluminense Federal University (UFF), Niterói, Brazil

Keywords: Threat Modeling, Risk Management, Cyber Security, Information Security, Software Design, Secure Development Life-Cycle (SDLC), Secure Software Development Lifecycle (S-SDLC), Privacy Engineering, Privacy by Design, Data Protection, Systematic Literature Review.

Abstract: Threat Modeling (TM) has increased its relevance in cybersecurity risk management applied to software development, allowing developers to proactively identify and mitigate threats from various sources. In the present work, we execute a systematic literature research (SLR) on TM applied to cybersecurity. Population, Intervention, Comparison, Outcomes, Context (PICOC) criteria were used to define a research formula that was executed in three relevant digital libraries and was submitted to inclusion and exclusion criteria and a rigorous quality assessment, resulting in 16 papers that answered four research questions, which deeply defined key elements of TM, process steps, TM relation with risk management process existing in ISO 27005 and future perspectives for TM. This contribution supports the understanding of TM and its practical application when considering different existing models into real application development.

1 INTRODUCTION

Nowadays, as technology maturity increases, threats follow this evolution, not considering just phishing and massive storage attacks, but also companies' infrastructure, applications, databases, and servers. Also, considering cloud computing expansion, the attack surface has expanded, challenging information security professionals to protect asset information. Threat Modeling (TM) allows security professionals to identify security vulnerabilities, determine risk, and identify mitigations[1], and have been supporting companies' security journey in the evolved contemporary tech world.

The relevance of TM has been increasing. A systematic literature review (SLR) was performed (Xiong and Lagerström, 2019), assessing 54 papers and dividing them in three separated clusters: (1) articles making a contribution to TM, (2) articles using an existing TM approach, (3) introductory articles presenting work related to TM process. As results, TM was defined, different TM methods were listed, kinds of systems applicable for TM, types of threats and attacks and future research directions, considering their SLR results.

The previous SLR study considered four databases of high relevance, three of which the present work also considered: IEEE, Scopus, and Web of Science. Exclusion criteria were similar, aiming to found article papers related to TM, in English language. However, the results of previous study were divided into three clusters: Application of TM, TM methods and TM process. The paper concluded that TM: Lacks common ground; Have numerous definitions; Is used in different ways; Its commonly applied manually, but with flexible form (graphical, formal, qualitative, quantitative); It can have a general or specific application domain; Have multiple validation methods.

Considering the abovementioned diversity of the TM subject, this study conducted a fully segregated SLR research. The contribution of the present paper aims to understand definitions regarding the key elements of TM, presenting the detailed steps of each procedural TM paper contributions, providing the TM

[a] https://orcid.org/0000-0002-2256-0450
[b] https://orcid.org/0000-0002-5644-4167
[c] https://orcid.org/0000-0003-3339-9735

[1] https://www.microsoft.com/en-us/securityengineering/sdl/practices#practice4

Lohmann, P., Albuquerque, C. and Machado, R.
Systematic Literature Review of Threat Modeling Concepts.
DOI: 10.5220/0011783000003405
In *Proceedings of the 9th International Conference on Information Systems Security and Privacy (ICISSP 2023)*, pages 163-173
ISBN: 978-989-758-624-8; ISSN: 2184-4356

relationship with risk management, and driving future research directions for TM. Also, it is relevant to mention that, according to our mapping, 72% of the result papers analysed in this work were not present in the first SLR due to the growth of research interest in TM in the last years.

2 METHODOLOGY DESCRIPTION

A systematic literature review (SLR) is a research methodology that intends to identify, evaluate, and interpret available research that are relevant to a particular research question, topic area or phenomenon of interest. The scope of the analysis is based on primary studies, which are empirical studies that investigates specific research question (Kitchenham and Charters, 2007). The goal of this study is to provide a SLR about the proposed subject, using the guidelines presented by (Kitchenham and Charters, 2007), which includes performing the three phases and its specific steps.

As the first step, the PICOC criteria (Population, Intervention, Comparison Outcomes, Context) was considered for defining the research questions. PICOC stands for *Population*, specifying population subject group of interest for the research, *Intervention*, defining the scope of interest in reviewing, *Comparison*, for comparison between different methodologies, procedures or studies, *Outcomes*, related to factors of importance for the research, *Context*, defining the context of the research (Kitchenham and Charters, 2007). PICOC elements were defined according to the following:

- **Population:** Threat Modeling
- **Intervention:** Risk Management
- **Comparison:** This paper intends to analyse the paper results, not to compare them
- **Outcomes:** Process
- **Context:** Cyber Security

A. Research Questions

Considering the abovementioned PICOC criteria, this research will consider the following questions:

- **RQ01:** What are the key elements of threat modeling?
- **RQ02:** Which steps are considered for threat modeling?
- **RQ03:** Which phases of the risk ISO27005 Risk Management process are addressed?
- **RQ04:** What are the future perspectives for threat modeling?

B. Search Method

The objective of this study is to search for primary studies related to the research questions. Therefore, to carry out a rigorous and auditable process, it is essential to systematically expose the steps of the search method, presenting the definitions for search terms, search string, search strategy and data sources.

1) *Search Terms*: Based on the research questions and the PICOC criteria, the next step was to define the search terms, as well as to identify its related synonyms.

Table 1: Search terms and Synonyms.

Criteria	Terms	Synonyms
Population	Threat Modeling	Threat Model - Threat Intelligence - Threat Trees - Threat Analysis - Attack Simulations
Intervention	Risk Management	Security Risk Assessment - Secure Configuration – Requirements - Requirements Engineering - Security Requirements Elicitation - Anti-Requirements - Secure Software Engineering - Secure Software - Security Testing - Security Risk – Measurement - Security and Privacy Controls
Comparison	-	-
Outcomes	Process	Framework - Method
Context	Cyber Security	Security - Computer Security - Information Security – Vulnerability – Data Security – Cloud-Security – Security Architecture – Cyber-Attacks – Attack – Adversary – Data Protection by Design – Privacy by Design – Privacy Engineering – System Analysis – Design – Software Design – Solution Design – System Model – Software Development – Secure Development Life-Cycle (SDLC)

2) *Search String*: The search terms were used to generate generic search string (presented in Table 2), which was applied to the Title, Abstract and Keywords in each data source to gather primary studies.

Table 2: Generic Search String.

(("Threat Modeling" OR "Threat Model" OR "Threat Intelligence" OR "Threat Trees" OR "Threat Analysis" OR "Attack Simulations") AND ("Risk Management" OR "Security Risk Assessment" OR "Secure Configuration" OR "Requirements" OR "Requirements Engineering" OR "Security Requirements Elicitation" OR "Anti-Requirements" OR "Secure Software Engineering" OR "Secure Software" OR "Security Testing" OR "Security Risk" OR "Measurement" OR "Security And Privacy Controls") AND ("Process" OR "Framework" OR "Method") AND ("Cyber Security" OR "Security" OR "Computer Security" OR "Information Security" OR "Vulnerability" OR "Data Security" OR "Cloud-Security" OR "Security Architecture" OR "Cyber-Attacks" OR "Attack" OR "Adversary" OR "Data Protection By Design" OR "Privacy By Design" OR "Privacy Engineering" OR "System Analysis" And "Design" OR "Software Design" OR "Solution Design" OR "System Model" OR "Software Development" OR "Secure Development Life-Cycle (SDLC)"))

3) *Search Strategy*: The search strategy is the way that the studies are retrieved, aiming to cover as much as possible the current literature. For this work, the adopted search strategy was *database search* in digital libraries and search engines, pointed out in the next section.

4) *Data Sources*: The chosen data sources were *Web of Science* [2], Scopus [3] and IEEE [4], considering certain quality-related criteria such as relevance in the computer field, publishing regularity, ease of use, filter variety and full text of papers available for members of the academia.

C. Selection Criteria

Regarding identified result studies into the digital databases deriving from the search string, it was defined that a criteria should be considered, to find the most relevant papers. The results from the three databases were consolidated and filtered, removing duplicates and them, being classified as “Accepted” or “Rejected”, meaning that the studies are relevant to respond the research questions and the opposite, respectively. The acronyms in Table 3 stands for Inclusive Criteria (IC) and Exclusion Criteria (EC).

Table 3: Inclusion/Exclusion criteria.

IC01	The study is about TM applied to cyber security
IC02	The study was published in a Journal
IC03	The study is in the field of computer science
IC04	The type of the document is article
IC05	The abstract of the paper contains the term “Threat Modeling”
IC06	The study was published in English
IC07	The journal has at least 2 articles published regarding this paper scope
EC01	The study is secondary or tertiary
EC02	The study is of the grey literature
EC03	The study is an index, preface, tutorial or editorial, lecture or summary of a conference workshop
EC04	The study is not related to computer science
EC05	The study is not related to TM applied to cyber security

After analysing the results, the paper “Assessing IoT enabled cyber-physical attack paths against critical systems” was removed due to EC05 criteria.

D. Quality Assessment

Having selection criteria defined and sufficient PICOC keywords groups covering all relevant studies, the quality of the result has to be assessed (Kitchenham and Charters, 2007). To achieve this, our study considered journals with good impact rate, having an average of 3.38, according to Table 4:

[2] https://clarivate.com/webofsciencegroup/solutions/web-of-science/

[3] https://www.scopus.com/

[4] https://www.ieee.org/

Table 4: Quality Assessment.

Paper	Journal	Imp. Factor
(Bedi and others, 2013)	Software: Practice and Experience	3.34
(Pereira-Vale and others, 2021)	Computers and Security	5.1
(Sabbagh and Kowalski, 2015; Shi and others, 2022)	IEEE Security and Privacy	3.5
(Nyambo, et. al, 2014)	International Journal of Computing and Digital Systems	1.01
(Alwaheidi and Islam, 2022; Mauri and Damiani, 2022)	Sensors	3.84
(Elahi and others, 2021; Uzunov and Fernandez, 2014)	Computer Standards and Interfaces	4.94
(Hacks and others, 2022; Xiong and others, 2022)	Software and Systems Modeling	2.82
(AlFedaghi and Alkandari, 2011)	International Journal of Digital Content Technology and its Applications	Discon-tinued
(Wijesiriwardana and others, 2020; Yeng and others, 2020)	International Journal of Advanced Computer Science and Applications	1.09
(Girdhar and others, 2022)	IEEE Access	3.47
(Zeng and others, 2022)	IEEE Transactions on Network and Service Management	4.75

E. Data Extraction

Some data were defined to be extracted from the studies, which were (1) Title, (2) Author, (3) Publication Year, (4) Publication Type, (5) Publisher, (6) Source.

F. Pilot

After defining the elements for the planning phase, a pilot was performed, executing the search string in each digital library. The results were consolidated, and all appeared keywords were analysed, being included in the search string for results improvement.

3 RESULTS AND DISCUSSIONS

A. Selection of Studies

Initially, the search string conducted in the data sources returned 29 results which, after removing duplicates, left 17 unique papers for analysis. Then, the including and exclusion criteria were applied in two reading stages: (1) Title, abstract and keywords; and (2) Introduction and conclusion. In this step, the number of the papers was the same, as the search string had already specified deeply directed the results into the most relevant ones.

B. Data Extraction

This section presents the extracted data in a summarized way, through graphs and descriptive texts. Data Extraction has shown that the digital library with most results was Scopus, which returned 62% of the studies, while IEEE returned 17% and Web of Science 20%.

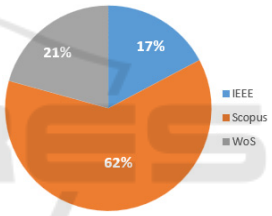

Figure 1: Overview of the results based on digital libraries.

Also, analysing the years of the results, it is possible to check that the threat modelling subject increased its research in the last two years.

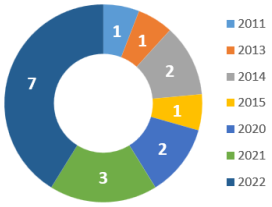

Figure 2: Overview of the publications per year.

C. Addressing the Research Questions

RQ01 - What are the key elements of threat modeling? (Bedi and others, 2013) brings honeytokens to identify real unidentified threats during TM exercise, in a three-phased process. It is defined that TM provides a structured way to secure software design by allowing security designers to accurately estimate the attacker's capabilities in respect of known threats (Swiderski and Synder, 2005).

(Shi and others, 2022) defines TM as a structured process for identifying and understanding potential threats as well as developing and prioritizing mitigations, so that valuable assets in the system can be protected, also guiding the investment on system security.

(Nyambo, et. al, 2014) uses the Microsoft Threat Model STRIDE to identify security threats on Livestock Data Center (LDC), as a case study application of the *methodology for security threat analysis and requirements specification in web/mobile applications development*. STRIDE stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of Privilege, which are considered the attacker goals and the ones that guide the TM exercise (Abi-Antoun, 2010).

(Sabbagh and Kowalski, 2015) analyses TM applied to software supply chain (SSC). The paper defines TM as an ontological analysis of threats that aims to study the existence and nature of the threats, expressing and capturing as many threats as necessary to manage exiting risks and take appropriate countermeasures. Moreover, the paper brings ISO 27005 definition of threat, which is defined as a potential cause of an incident that might result in harm to systems and organizations. Also, they classify the threats in SSC as Social Threats, related to human errors or behaviours such a supplier denying having sent a software product, and Technical Threats, related to hardware, operating systems and application, for instance, individuals being able to insert malicious code over the network due to suppliers' security flaws, resulting in defects in the delivered software product. The countermeasures proposed for each threat are also classified as Social and Technical.

(Mauri and Damiani, 2022) defines TM as a process for reviewing the security of a system, identifying critical areas, and assessing the risk associated with them, allowing for profiling and prioritizing problems as well as potential mitigation. The paper proposes a methodology for assessing security in Artificial intelligence (AI) and Machine Learning (ML) based systems. Microsoft STRIDE[5] threat model is adapted to support this specific field. The paper shares an overview of other relevant TM methods, such as PASTA (Morana and Uceda Vélez, 2015) and OCTAVE (Oladimeji and others, 2006), but affirms that STRIDE is the most mature one.

(Uzunov and Fernandez, 2014) presents a combination of values of threat libraries and taxonomies and propose a two-level pattern-based taxonomy for development of distributed systems, encompassing both threats to a system and the ones corresponding to countermeasures realizations, called meta-security threats. The article says that TM is a systematic approach to introduce several security measures resulted of analysing the potential attacks or threats to a system in each context, during the requirements analysis stage, design stage, or both, determining the risks and likelihoods and how they could be potentially mitigated. Also, conducting TM requires sound knowledge of system technical domain and sufficient security expertise to consider both generic and specific attacks. Finally, threat libraries, if used, enhances the efficacy of TM process, as the common threats are considered.

(Hacks and others, 2022) considers (Shostack, 2014) definition of TM, which is a process that supports the secure design of systems by easing the understanding of the system complexity, as well as identifying and modelling potential threats, weaknesses and vulnerabilities to the system or individual components. As TM exercise generally results in many threats modelled, keep tracking which parts of each one was tested became complex. Considering this, the paper presents an automated way of developers to check the quality of the results of the TM exercise, improving them. It is augmented that combining threat models can help defining the countermeasures for identified threats, but also reinforces that each existing threat model addresses specific threats. Moreover, it is shared that several threat models do not provide the necessary coverage of potential attacks but focus on attacker's capabilities through different metrics.

(AlFedaghi and Alkandari, 2011) analyses key aspects of Microsoft Software Development Lifecycle (SDL)[6]. It is presented that TM examines a system from the attacker's perspective and based in the assumption that an attack comes during interaction with the system (Abi-Antoun and others, 2007) e (Abi-Antoun, 2010). Also, it shares the Microsoft definition of TM, consisting in a systematic process used to identify threats and vulnerabilities in software, being considered a form of risk analysis, having 5 components, existence, capability, history, intentions and targeting. It is highlighted that TM is different from attack modelling, which concentrates on nature of an attack, not threats conducting them.

[5] https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach

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

An example presented of it is the attack tree, that is done separate from TM, meaning that the threat could execute the mapped attack tree. Finally, it says that DFDs are used for performing TM, and that they are composed by potential threat targets, such as data sources, data flows and interactions, and symbols, such as External entity (rectangles), process (circle), multiple process (double circumference circle), Data store (upper and lower edge box), Data flow (arrows) and Privilege Boundary (dotted line).

(Xiong and others, 2022) proposes a TM language named enterpriseLang, for enterprise security based on MITRE Enterprise ATT&CK Matrix, which describes adversary behaviours to measure the resilience of an enterprise against various cyber-attacks. The paper defines TM as an approach for identifying main assets within a system and threats to these assets (Xiong and Lagerström, 2019), being coupled with attack simulations to evaluate its security and discovery of strengths and weakness in software applications. Also, TM methods can be categorized into manual modelling, automatic modelling, formal modelling (mathematical models) and graphical modelling (attack trees, attack and defence graphs, or tables) (Xiong and Lagerström, 2019). From the system evaluation perspective, the system architecture is represented and analysed, potential security threats are identified and appropriate mitigation techniques are selected (Dhillon, 2011; Frydman and others, 2014). Regarding application development, TM is used to assist engineers to identify potential security threats associated with a software product (Kiesling and others, 2016).

(Alwaheidi and Islam, 2022) delivers a data-driven threat-analysis (d-TM), specifically for cloud-based systems, across all cloud service provider threat layers. STRIDE threat model was considered an effective technique for effective measure for safeguarding cloud systems (Morana and Uceda Vélez, 2015), helping identifying attack vectors impact, before its occurrence, and its vulnerabilities. PASTA was also mentioned as relevant, with its seven stages covering a complete cyber security posture in the cloud (Sequeiros and others, 2020). Attack trees are mentioned as relevant, providing schematic representation of how assets in a cloud system might be attacked in the form of a tree (Sequeiros and others, 2020). DFDs were used to represent the relationships between systems and endpoint assets. As knowledge bases for cloud TM, MITRE CWE[7], MITRE CAPEC[8] and NIST SP-800-53[9] are considered.

(Wijesiriwardana and others, 2020) proposes a knowledge-modelling based approach to semantically infer the associations between architectural level of security flaws and code-level bugs, combining TM, static code analysis and exploiting knowledge bases to infer relationships between flaws and bugs. In the TM related work, some approaches that considers architectural risk analysis are mentioned, with identification and mitigation trees forming a knowledge base called attack patterns (Frydman and others, 2014); Other tool was developed a CAPEC attack patterns, allowing developers to think like attackers, mapping threats in all SDLC phases and checking it against each STRIDE category (Yuan and others, 2014); A practical approach was used STRIDE to detect vulnerabilities and mitigations in software. Those tree approaches work only in the design phase, not linking threats with source code level bugs (Berger and others, 2016).

(Yeng and others, 2020) compared TM methods to determine their suitability for identifying cloud related threats, in the healthcare context. The paper defines TM as the use of methods to help in thinking, identifying and enumerating possible risks and threats, helping in the identification of the lack of security controls for mitigating risks (Alhebaishi and others, 2019; Amini and others, 2015; Cheng and others, 2012, p. 202; Hong and others, 2019; Yahya and others, 2015; Zimba and others, 2016). Also, threat models mentioned, such as Attack Tree (structured way of describing threats and vulnerabilities of a system, presenting possible attack paths from attackers) (Shostack, 2014; Zimba and others, 2016), Attack Graph (graphical view of attack paths, considering attack point to attack target and network information dependency interactions) (Cheng and others, 2012), Attack Surface (software features that may have vulnerabilities) (Amini and others, 2015), Practical Threat Analysis (identification of assets and their related values, mapping the potential damage to be caused by adversaries, identifying vulnerabilities, assessing the risks of threats and defining risk mitigations strategies to the system), Threat Model Framework for Personal Network (description of all devices in network from user perspective; gathering of network requirements from use case diagram, network architecture, environments and technologies; definition of data

[7] https://cwe.mitre.org/
[8] https://capec.mitre.org/
[9] https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

flows using UML sequence diagrams, actors and devices; identification of assets and its related threats; identifying vulnerabilities and related risks; rating of the vulnerabilities and threats), STRIDE (threats related to STRIDE acronym), TM in pervasive computing (identification of cloud computer user roles; identification of security domains and interaction dynamic; mapping of trust levels; vulnerability identification and risk evaluation), LINDDUN (identification of privacy threats) (Gholami and Laure, 2016).

(Pereira-Vale and others, 2021) says that TM means defining a threat model and being able to enumerate threats for a given system.

(Girdhar and others, 2022) presents existing TM techniques for the automotive sector, specifically in cyber-physical system (CPS) of a connected and automated vehicle (CAV) ecosystem, with an extensive application of STRIDE threat model, which have been widely applied in IT industry. The paper defines DFD as a graphical representation of the data flow, composed of various entities.

(Zeng and others, 2022) proposes a vulnerability risk prioritization system named LICALITY, aiming to capture attacker preference on exploiting vulnerabilities through a TM method, which identifies threat attributes of the given historical threat and exploits record of the network and neuro-symbolic model that learns such threat attributes. It also says that TM aims to provide a systematic analysis of potential threats and vulnerabilities in a system, being a process for capturing, organizing, and analysing all the information that affects the security of a system, identifying vulnerabilities and suggesting defences against them (Drake, 2022). Also, it defines threat model as a structure representation of an attack graph enumerating known vulnerability that attackers may exploit in a system (Drake, 2022).

When applied for risk assessment context, risk metrics are created based on attackers' motivation, capability, and corresponding vulnerabilities (Bromander and others, 2016). Also, TM is a step of the LICALITY system, which considers 3 arguments for TM: (1) attackers experiences of software services have attributes on assessing the likelihood of exploitation; (2) Vulnerabilities access complexity and impact features have attributes on assessing the criticality of exploitation; (3) exploit records in the wild have attributes on assessing the likelihood of exploitation (Zeng and others, 2022).

(Elahi and others, 2021) proposes a semi-quantitative approach for TM and risk analysis of intelligent mobile cloud computing applications (IMCCAs), which defines quantitative risk scores for each threat. The paper says that TM is a recommended method for modeling attack/defense scenarios in software applications to assess their risks (Souppaya and Scarfone, 2016).

RQ02 - Which steps are considered for threat modeling?

Table 5 shows all papers that presented a methodology that contributes to TM application were classified as "Procedural", and others as "Conceptual". The conceptual ones contain papers that presented methodologies, but for different purposes other than TM. For RQ02, only procedural papers were analysed.

Table 5: Paper contributions classification.

Paper	Contribution
(Alwaheidi and Islam, 2022; Bedi and others, 2013; Elahi and others, 2021; Girdhar and others, 2022; Mauri and Damiani, 2022; Nyambo, et. al, 2014; Shi and others, 2022; Wijesiriwardana and others, 2020; Xiong and Lagerström, 2019; Yeng and others, 2020)	Procedural
(AlFedaghi and Alkandari, 2011; Hacks and others, 2022; Pereira-Vale and others, 2021; Sabbagh and Kowalski, 2015; Uzunov and Fernandez, 2014; Zeng and others, 2022)	Conceptual

(Bedi and others, 2013) defines a threat-oriented security model which consists of 3 phases: (1) Identification of known and unknown threats, (2) analyse the identified threats attack paths, trough threat trees, (3) use meta-agents combined with fuzzy inference systems to monitor identified threats using security baseline and applying actions according to risk level, saving the system from being compromised.

(Shi and others, 2022) presents six steps for TM, mentioning that they are defined to be iteratively applied during the SDLC: (1) *Define security requirements*, including relevant standards and functionalities; (2) *Model the system*, consisting in the creation of a graph representing the artifacts as nodes, grouped in trust zones, and data flows between the nodes. The model can be written using specific syntax or can be a diagram form, such as data flow diagram (DFD), which should contain system entities, events, and system boundaries, having 4 types of elements,

processes, data flows, data stores and external entities; (3) *Identify threats based on the system model*, using threat knowledge data bases, such as STRIDE[10], Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE) and Common Attack Pattern Enumeration and Classification (CAPEC); (4) *Evaluate the identified threats based on multiple aspects, such as accessibility to attackers, attack complexity, privileges required and so on,* quantifying the severity of the threats trough scoring systems, such as Common Vulnerability Scoring System (CVSS) and Common Weakness Scoring System (SWSS); (5) *Mitigate potential threats according to the security requirements*, considering in which specific system the threat resides and considering general suggestions provided by knowledge bases, such as CWE and CAPEC; (6) *Validate that the threats are mitigated after applying mitigation techniques*, testing the applied security controls. OWASP Application Security Validation Standard is recommended.

(Nyambo, et. al, 2014) defines that the identification of security threats paused by an application consists in three steps: (1) *Application decomposition*, defined by a presentation of how application works, interacts with users and which assets attackers might be interested in. Approaches like Unified Modelling Language class, Entity relationship and Data Flow Diagrams (DFD) can be used in the decomposition; (2) Determination and ranking of threats, defining a specific that model, describing identified threats e associated threat categories and calculating their impact over the application; (3) Determination of countermeasure and mitigation, step which were not explored by the paper.

(Mauri and Damiani, 2022) uses (Myagmar and others, 2005) definition, which considers that a typical TM process consists in five steps: (1) Objectives Identification – Status the security properties the system should have; (2) Survey – Determines the system's assets, their interconnections and connections to outside systems; (3) Decomposition – Selects the assets that are relevant to the security analysis; (4) Threat Identification – Enumerates threats to the system's components and assets; (5) Vulnerabilities Identifications – Examines identified threats and determines if known attacks show that overall system is vulnerable to them.

(Xiong and others, 2022) defines some steps for the implementation of the proposed meta-model: (1) Load enterpriseLang in a simulation tool called securiCAD; (2) Create system model by specifying its assets, associations, and adversaries' entry point; (3) Execute attack simulations over the system model, automatically receiving the vulnerabilities and possible mitigation strategies, due to ATT&CK knowledge base incorporated into enterpriseLang, on its original implementation.

(Alwaheidi and Islam, 2022) considers four phases for TM exercise, in cloud context: (1) Data collection, aiming to understand business processes, services and infrastructure assets; (2) Data analysis, identifying and extracting data levels and constructing data-flow diagrams; (3) Threat analysis, identifying weaknesses and related threats, composing the threat profile and threat priority list; (4) Threat mitigation, defining controls and identifying security assurance level for each control.

(Wijesiriwardana and others, 2020) defines three steps for TM: (1) Decomposition, consisting in understanding the application and how it interacts with external entities, producing DFDs; (2) Determine and Rank Threats, stablishing a threat categorization methodology, covering both attackers and defensive perspectives; (3) Countermeasures and mitigation, for ranked threats.

(Yeng and others, 2020) defines that TM for cloud computing in healthcare should have the following characteristics (Amini and others, 2015; Malik and others, 2008; Shostack, 2014): (1) Identifying and classifying assets; (2) Identifying users and Threat agents; (3) Establishing Trust level and Users role; (4) Identifying Security Domain; (5) Identifying vulnerabilities; (6) Identifying Threats, (7) Ranking and Measuring vulnerabilities; (8) Ranking and measuring threats; (9) Identifying countermeasures, (10) defining new assets threats or vulnerabilities. Also, it is reinforced that this whole process should ongoing. The paper also identifies each step over the presented TM methods, in a consolidated table view.

(Girdhar and others, 2022) defines a framework for investigation of cyberattacks-related accidents, in connected and automated vehicles (CAV), containing 3 phases, (1) analysis of cyberattack-induced CAV accident; (2) STRIDE threat modeling; (3) Potential cybersecurity measures.

(Elahi and others, 2021) defines 8 steps for semi quantitative TM and risk analysis: (1) Characterize the system; (2) Identify the attack vectors/threat

[10] https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

conditions; (3) Classify attack vectors/threat conditions; (4) Assess the impact of the threats; (5) Determine the likelihood of threats; (6) Compute gross risks; (7) Identify security controls and their effectiveness; (8) calculate net risk.

RQ03 - Which phases of the ISO27005 Risk Management process are addressed?

ISO27005:2022 defines 7 phases, which were originated from ISO31000 and adapted into information security context: (1) Internal and External context establishment, criteria and scope; (2) Risk Identification, covering assets, threats, vulnerabilities, consequences; (3) Risk Analysis, delimiting the consequences and likelihood of each risk, having its risk level; (4) Risk Evaluation, comparing with defined context criteria; (5) Risk Treatment, with the options of modify, retain, avoid or share the risks, defining a risk treatment plan; (6) Communication and Consultation, involving stakeholders for decision making (7); Monitoring and Review, in ongoing basis over related assets, impacts, threats, vulnerabilities, likelihoods of occurrence.

According to the above definitions, TM steps of the papers classified as "Procedural" in RQ02 were compared against each phase of the risk management process. As a result, it was understood that all procedural papers covers ISO27005 phases 1,2,4,5, but phase 3 were considered only in (Elahi and others, 2021). It is interesting to highlight that severity and threat ranking is addressed in the papers, but likelihood of the threat is present in just one paper, despite others mentions about creation of attack trees and DFD during context validation.

RQ04 - What are the future perspectives for threat modeling?

Future perspectives presented by selected papers were classified and seven groups: (1) Integration with cybersecurity standards or knowledge bases, to amplify the scope of analysed threats and vulnerabilities, (2) Integration with other threat models, (3) Incorporation of legal and regulatory requirements into the threat model methodology, (4) Application of the presented methodology, tool or framework, for validation, improvement and extension, (5) Automation of the presented method or some phases of the assessment, (6) Creation of benchmarks for quantitative evaluation of threat model tools and (7) Definition of relevant research areas on development of secure application, as presented by Table 6:

Table 6: Future Perspectives.

ID	Papers
1	(Girdhar and others, 2022; Hacks and others, 2022; Mauri and Damiani, 2022; Shi and others, 2022; Uzunov and Fernandez, 2014; Xiong and others, 2022; Yeng and others, 2020)
2	(Uzunov and Fernandez, 2014; Yeng and others, 2020)
3	(Yeng and others, 2020)
4	(AlFedaghi and Alkandari, 2011; Alwaheidi and Islam, 2022; Bedi and others, 2013; Elahi and others, 2021; Girdhar and others, 2022; Hacks and others, 2022; Nyambo, et. al, 2014; Sabbagh and Kowalski, 2015; Wijesiriwardana and others, 2020; Zeng and others, 2022)
5	(Alwaheidi and Islam, 2022)
6	(Shi and others, 2022)
7	(Pereira-Vale and others, 2021)

4 CONCLUSIONS

Regarding key elements of TM, it is understood that TM is a systematic approach for securing software and applications, performed during designing phase. Also, environment scope (on-premises or cloud) doesn't drive the analysis, being adapted for each scenario, provided that the data flow diagrams (DFD), as a tool for architecture description, represents the flows of information, entities, processes, data stores and privileged boundaries adequately. Threat models can be used to construct the DFDs, such as Attack Tree, Attack Graph, Attack Surface, Practical Threat Analysis, Threat Model Framework for Personal Network, STRIDE, Pervasive Computing and LINDDUN. To do so, sound knowledge of system technical domain and security expertise are mandatory.

In DFDs, attacker's capabilities can be analysed and mapped, along with their possible interactions within the system and existing elements. These interactions will be considered as potential threats, and they can enhance the TM process if complemented by threat libraries, which maps known threats for different scenarios that can affect the analysed system. After identifying threats and documenting, TM considers the definition of adequate safeguards and security controls for mitigation of the identified threats. This controls and action plans can be prioritized according to threat severity, which can be ranked based on their criticality.

This paper identified the key elements listed above in TM steps, as RQ02 was intended to consolidate. Generally, TM methodologies follows

(1) Understanding of the applications or asset's structure and how elements interact with each other, (2) Threat identification in this context, (3) Threat analysis, regarding its severity and impact in the established context, (4) Definition of countermeasures. Some authors consider risk calculation based on likelihood, others include technological elements to support threat identification, such as honeytokens, for unknown threats mapping based in real attack situations.

RQ03 allowed the study to confirm that TM is directly relate to risk management, as the integration between TM steps and ISO 27005 phases, although few of the papers includes likelihood factor in TM exercise. As R04 pointed for future research should consider having full integration between TM and risk management, enabling appropriate risk mapping and mitigation control of the identified threats and their related risks.

REFERENCES

Abi-Antoun, M., 2010, STRIDE-based security model in Acme: p. 16.

Abi-Antoun, M., Wang, D., and Torr, P., 2007, Checking threat modeling data flow diagrams for implementation conformance and security, *in* the twenty-second IEEE/ACM international conference – Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering - ASE '07: ACM Press, Atlanta, Georgia, USA, p. 393.

AlFedaghi, S., and Alkandari, A., 2011, On Security Development Lifecycle: Conceptual Description of Vulnerabilities, Risks, and Threats: International Journal of Digital Content Technology and its Applications, v. 5, p. 296–306.

Alhebaishi, N., Wang, L., and Singhal, A., 2019, Threat Modeling for Cloud Infrastructures: ICST Transactions on Security and Safety, v. 5, p. 156246.

Alwaheidi, M. K. S., and Islam, S., 2022, Data-Driven Threat Analysis for Ensuring Security in Cloud Enabled Systems: Sensors, v. 22, p. 5726.

Amini, A., Jamil, N., Ahmad, A. R., and Z`aba, M. R., 2015, Threat Modeling Approaches for Securing Cloud Computin: Journal of Applied Sciences, v. 15, p. 953–967.

Bedi, P., Gandotra, V., Singhal, A., Narang, H., and Sharma, S., 2013, Threat-oriented security framework in risk management using multiagent system: PROACTIVE RISK MANAGEMENT: Software: Practice and Experience, v. 43, p. 1013–1038.

Berger, B. J., Sohr, K., and Koschke, R., 2016, Automatically Extracting Threats from Extended Data Flow Diagrams, *in* Caballero, J., Bodden, E., and Athanasopoulos, E. eds., Engineering Secure Software and Systems: Springer International Publishing, Cham, p. 56–71.

Bromander, Jøsang, A., and Eian, M., 2016, Semantic Cyberthreat Modelling: Proc. STIDS, p. 74–78.

Cheng, Y., Du, Y., Xu, J., Yuan, C., and Xue, Z., 2012, Research on security evaluation of cloud computing based on attack graph, *in* 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems:, p. 459–465.

Dhillon, D., 2011, Developer-Driven Threat Modeling: Lessons Learned in the Trenches: IEEE Security and Privacy, v. 9, p. 41–47.

Drake, V., 2022, Threat Modeling: OWASP.

Elahi, H., Wang, G., Xu, Y., Castiglione, A., Yan, Q., and Shehzad, M. N., 2021, On the Characterization and Risk Assessment of AI-Powered Mobile Cloud Applications: Computer Standards & Interfaces, v. 78, p. 103538.

Frydman, M., Ruiz, G., Heymann, E., César, E., and Miller, B. P., 2014, Automating Risk Analysis of Software Design Models (M. Ivanovic, Ed.): The Scientific World Journal, v. 2014, p. 805856.

Gholami, A., and Laure, E., 2016, Advanced Cloud Privacy Threat Modeling:

Girdhar, M., You, Y., Song, T.-J., Ghosh, S., and Hong, J., 2022, Post-Accident Cyberattack Event Analysis for Connected and Automated Vehicles: IEEE Access, v. 10, p. 83176–83194.

Hacks, S., Persson, L., and Hersén, N., 2022, Measuring and achieving test coverage of attack simulations extended version: Software and Systems Modeling.

Hong, J. B., Nhlabatsi, A., Kim, D. S., Hussein, A., Fetais, N., and Khan, K. M., 2019, Systematic identification of threats in the cloud: A survey: Computer Networks, v. 150, p. 46–69.

Kiesling, T., Krempel, M., Niederl, J., and Ziegler, J., 2016, A Model-Based Approach for Aviation Cyber Security Risk Assessment, *in* 2016 11th International Conference on Availability, Reliability and Security (ARES):, p. 517–525.

Kitchenham, B., and Charters, S. M., 2007, Guidelines for performing systematic literature reviews in software engineering:

Malik, N. A., Javed, M. Y., and Mahmud, U., 2008, Threat Modeling in Pervasive Computing Paradigm, *in* 2008 New Technologies, Mobility and Security:, p. 1–5.

Mauri, L., and Damiani, E., 2022, Modeling Threats to AI-ML Systems Using STRIDE: Sensors, v. 22, p. 6662.

Morana, M. M., and Uceda Vélez, T., 2015, Risk centric threat modeling: process for attack simulation and threat analysis: Wiley, Hoboken, New Jersey, 1 p.

Myagmar, S., Lee, A. J., and Yurcik, W., 2005, Threat Modeling as a Basis for Security Requirements: Proceedings of the IEEE Symposium on Requirements Engineering for Information Security, p. 9.

Nyambo, et. al, D., 2014, An Approach for Systematically Analyzing and Specifying Security Requirements for the Converged Web-Mobile Applications: International Journal of Computing and Digital Systems, v. 3, p. 207–217.

Oladimeji, E. A., Supakkul, S., and Chung, L., 2006, Security threat modeling and analysis: A goal-oriented approach, *in* ICSE 2006.:

Pereira-Vale, A., Fernandez, E. B., Monge, R., Astudillo, H., and Márquez, G., 2021, Security in microservice-based systems: A Multivocal literature review: Computers & Security, v. 103, p. 102200.

Sabbagh, B. A., and Kowalski, S., 2015, A Socio-technical Framework for Threat Modeling a Software Supply Chain: IEEE Security & Privacy, v. 13, p. 30–39.

Sequeiros, J. B. F., Chimuco, F. T., Samaila, M. G., Freire, M. M., and Inácio, P. R. M., 2020, Attack and System Modeling Applied to IoT, Cloud, and Mobile Ecosystems: Embedding Security by Design: ACM Comput. Surv., v. 53.

Shi, Z., Graffi, K., Starobinski, D., and Matyunin, N., 2022, Threat Modeling Tools: A Taxonomy: IEEE Security & Privacy, v. 20, p. 29–39.

Shostack, A., 2014, Threat Modeling: Designing for Security: Wiley, Indianapolis.

Souppaya, M., and Scarfone, K., 2016, Guide to data-centric system threat modeling: National Institute of Standards and Technology.

Swiderski, F., and Synder, W., 2005, Threat Modeling: Microsoft Press, Redmond , Washington, 288 p.

Uzunov, A. V., and Fernandez, E. B., 2014, An extensible pattern-based library and taxonomy of security threats for distributed systems: Computer Standards & Interfaces, v. 36, p. 734–747.

Wijesiriwardana, C., Abeyratne, A., Samarage, C., Dahanayake, B., and Wimalaratne, P., 2020, Secure Software Engineering: A Knowledge Modeling based Approach for Inferring Association between Source Code and Design Artifacts: International Journal of Advanced Computer Science and Applications, v. 11.

Xiong, W., and Lagerström, R., 2019, Threat modeling – A systematic literature review: Computers & Security, v. 84, p. 53–69.

Xiong, W., Legrand, E., Åberg, O., and Lagerström, R., 2022, Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix: Software and Systems Modeling, v. 21, p. 157–177.

Yahya, F., Walters, R. J., and Wills, G. B., 2015, Analysing threats in cloud storage, *in* 2015 World Congress on Internet Security (WorldCIS):, p. 44–48.

Yeng, P. K., D., S., and Yang, B., 2020, Comparative Analysis of Threat Modeling Methods for Cloud Computing towards Healthcare Security Practice: International Journal of Advanced Computer Science and Applications, v. 11.

Yuan, X., Nuakoh, E. B., Beal, J. S., and Yu, H., 2014, Retrieving Relevant CAPEC Attack Patterns for Secure Software Development, *in* Proceedings of the 9th Annual Cyber and Information Security Research Conference: Association for Computing Machinery, New York, NY, USA, p. 33–36.

Zeng, Z., Yang, Z., Huang, D., and Chung, C.-J., 2022, LICALITY—Likelihood and Criticality: Vulnerability Risk Prioritization Through Logical Reasoning and Deep Learning: IEEE Transactions on Network and Service Management, v. 19, p. 1746–1760.

Zimba, A., Hongsong, C., and Zhaoshun, W., 2016, Attack tree analysis of Man in the Cloud attacks on client device synchronization in cloud computing, *in* 2016 2nd IEEE International Conference on Computer and Communications (ICCC):, p. 2702–2706.