A Modeling-Driven Threat Analysis and Risk Assessment for Software-Defined Network Architecture

Mariyam Ouaissa[1*], Mariya Ouaissa[2], Zineb Nadifi[1], Sarah El Himer[3], Yassine Al Masmoudi[4], Ali Kartit[1]
[1]LTI, Chouaib Doukkali University, El Jadida, Morocco
[2]LISI, Cadi Ayyad University, Marrakech, Morocco
[3]Sidi Mohamed Ben Abdellah University, Fes, Morocco
[4]Laboratory of Geosciences and Environment Technics, Faculty of Science, El Jadida, Morocco
E-mail : ouaissa.mariyam@ucd.ac.ma, m.ouaissa@uca.ac.ma, nadifi.Z567@ucd.ac.ma, sarah.elhimer@usmba.ac.ma, almasmoudi.y@ucd.ac.ma, kartit.a@ucd.ac.ma
*Corresponding author

Keywords: SDN, SDN controller, threat modeling, DFD, STRIDE, risk assessment

Received: April 12, 2025

In recent years, the networking field has been marked by the emergence of a new technology that seems set to revolutionize everything in our infrastructure. Software-Defined Networks (SDN), which present a new approach to networking with a different architectural philosophy. These changes in network infrastructure simplify equipment and make it independent. SDN is not a single solution or product, but rather a collection of innovative technologies that enable centralized control of network resources, improved programmability and orchestration of these resources, and virtualization by decoupling them from the physical elements of the network. However, increasing centralization raises serious security and privacy concerns, exposing networks to vulnerabilities including unauthorized access, data breaches, and malware infections. This article examines security concerns in the deployment of SDN in a campus network environment using a structured Threat Modeling approach based on the STRIDE methodology. Following a methodical methodology, we describe the SDN use case and illustrate its architecture using a Data Flow Diagram (DFD) to identify essential assets and communication flows in each zone of the SDN architecture. We then examine potential hazards to each zone, assess the risks, and recommend appropriate mitigation strategies. Our method is to improve the security and dependability of SDN, a technology that provides substantial benefits in terms of flexibility, scalability, and network management but is vulnerable to a variety of cyber threats if not properly guarded. The combination of STRIDE threat modeling and CVSS scoring enables a comprehensive understanding of vulnerabilities in SDN environments, prioritizing mitigation efforts based on exploitability and impact. Specific outcomes include T11, rated as a Critical risk with a CVSS score of 9.8, and T5 and T4 reaching Extreme levels in the 5x5 matrix. This approach helps to address high-risk threats while ensuring a balanced security strategy across varying threat levels.

Povzetek: Opisano je ogrodje za modeliranje groženj in oceno tveganja v SDN-omrežjih z uporabo STRIDE, CVSS 3.1 in 5×5 matrike za celovito, podatkovno podprto varnostno strategijo.

1 Introduction

Software-Defined Network (SDN) represents a major advancement in network management and configuration. Unlike traditional network architectures, where controllers and switching devices are tightly integrated, SDN separates the control plane from the data plane [1]. The data plane consists of network devices such as switches, while flow management and decision-making are handled by the controller, which is the brain of the network. This controller, the smartest part of the network, makes decisions and manages traffic, including packet dropping. This separation enables centralized and scheduled network management, providing increased flexibility to adjust policies and configurations in real time [2].

However, despite its many benefits, SDN system introduces significant security challenges. SDN networks are inherently vulnerable to various threats due to their centralized control plane, the communication between controllers and network elements, and their reliance on programmable interfaces. Attackers can exploit these vulnerabilities to manipulate network traffic, launch Denial-of-Service (DoS) attacks, or gain unauthorized access to critical infrastructure [3]. Ensuring the security, integrity, and confidentiality of SDN networks is therefore a major concern. Moreover, traditional security mechanisms designed for conventional networks are often inadequate for SDN environments. The separation of the control and data planes, along with the open nature of SDN protocols such as OpenFlow, creates new attack vectors that require specialized security approaches [4].

Threat modeling is a crucial methodology for proactively identifying and mitigating security risks in

SDN networks. By systematically analyzing the components, data flows, and potential attack vectors, threat modeling helps security professionals anticipate threats and implement appropriate countermeasures [5].

In this paper, we propose a structured approach to threat modeling in SDN environments. This paper proposes a structured threat identification and risk quantification framework for SDN based on STRIDE, CVSS 3.1 scoring, and a 5×5 risk matrix to guide mitigation strategy selection. Our methodology consists of several steps, namely: defining the SDN-based network infrastructure and identifying vulnerabilities, generating data flow and process flow diagrams, identifying threats, and applying risk assessment to evaluate their impact on confidentiality, integrity, availability, and other security objectives if the identified vulnerabilities are exploited. The threats are then categorized, and corresponding mitigation strategies are proposed. The structure of this paper is as follows: Section 2 provides an overview of SDN and its vulnerabilities. Section 3 discusses the threat modeling methodology and tools. Section 4 presents the proposed methodology. Section 5 details the results along with a discussion. Finally, conclusions are drawn in Section 6.

2 Related work

Table 1 presents a comparative analysis of recent works on SDN security, highlighting the threat models used, SDN layers analyzed, evaluation metrics, and the overall scope of each study. This comparison underscores the novelty and completeness of our approach in addressing both qualitative and quantitative aspects of SDN threat assessment

Given the importance, topicality, and richness of the subject, it has been the subject of various researches and articles, and has been tackled from different nails and approaches. The study in [6] presented a security evaluation framework for SDN architectures in data center environments. It uses STRIDE and PASTA models to identify threats and vulnerabilities across all SDN layers. The framework includes CVSS-based risk assessment, attack modeling in a Mininet-ONOS testbed, and mitigation strategies using specific and centralized countermeasures.

Authors in [7] conducted a Systematic Mapping Study (SMS) to investigate how SDN controllers enhance security in IoT networks. It analyzes 33 studies, focusing on types of SDN controller architectures, security issues, and mitigation techniques. The study finds centralized controllers most used but highlights their limitations. It also identifies gaps and future research directions in securing SDN-IoT environments.

The research in [8] proposed a comprehensive threat model for SDN by identifying four key use cases where security breaches can occur: applications attacking controllers, inter-controller attacks, controllers attacking switches, and switches attacking controllers. It analyzed each scenario, compared them with traditional networks, and provided tailored countermeasures.

The work in [9] presented a model-based method for simulating and analyzing threats in in-vehicle networks, focusing on modern vehicles increasingly integrating IoT and SDN technologies. The approach involved capturing network data, creating a data model in DBC format, synthesizing a functional model using SysML, and translating it into a detailed simulation model. The method included steps for model validation and threat analysis by simulating various attack types such as DoS, frame injection, and fuzzing.

The authors in [10] introduced a systematic framework for modeling and analyzing the security of SDN systems. The authors proposed a novel graphical security model called Threat Vector - Hierarchical Attack Representation Model (TV-HARM), which captured complex attack paths and threat combinations in SDN environments. To assess the security posture, the framework employed three types of metrics: Network Centrality Measures, Vulnerability Scores and Attack Impact Metrics.

This present article provides a modeling-driven threat analysis and risk assessment framework tailored for SDN architectures. Compared to existing literature, this paper offers a more comprehensive and actionable approach to securing SDN architectures. While prior works have typically focused on threat identification using models such as STRIDE or custom use-case scenarios, they often fall short in translating these threats into quantifiable risk levels. This paper bridges that gap by integrating STRIDE for systematic threat modeling, CVSS for standardized quantitative risk scoring, and a 5×5 risk matrix for intuitive risk visualization and prioritization. The approach enables comprehensive analysis across the SDN application, control, and data planes, supporting informed decision-making for securing SDN deployments.

3 Background

In this section, we present the architecture of software-defined networks and their components, followed by a discussion of the security challenges associated with this architecture.

3.1 SDN architecture

The SDN architecture is composed of several layers and interfaces, as illustrated in Figure 1 [11], [12].

Table 1: Comparative analysis of existing works on SDN security

Reference	Model Used	SDN Layer Analyzed	Metrics Employed	Scope
Ivkić et al. [6]	STRIDE, PASTA	All SDN Layers	Threat mapping, risk classification	Data Center SDN
Oredola & Ashraf [7]	Systematic Mapping Study (SMS)	Control Plane	ML performance metrics (Precision, Recall), Packet Analysis	SDN-IoT
Sharma & Tyagi [8]	Custom Use-Case-Based Threat Model	All SDN layers	Comparative analysis, attack taxonomy	SDN architecture
Lekidis et al. [9]	Stochastic Model Checking with CTMCs (PRISM tool)	Control and Application	Reaction time, Network cost, Attack success probability	SDN controller
Eom et al. [10]	TV-HARM (Threat Vector-Hierarchical Attack Representation Model)	Control and Data Network	Centrality Measures, Vulnerability Scores, Attack Impact Metrics	SDN environments

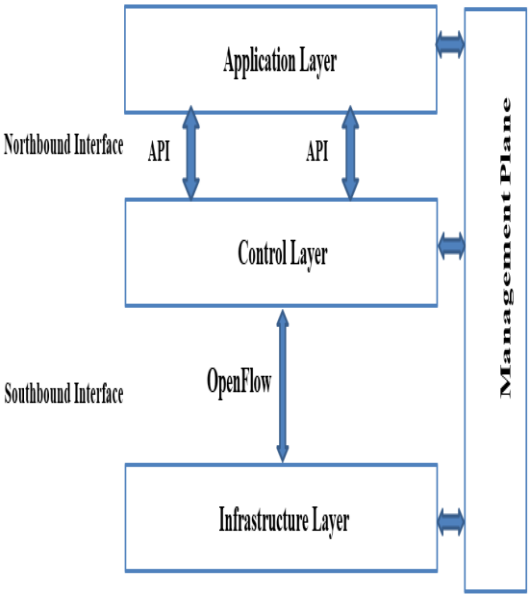

Figure 1: Architecture of SDN network

- **Infrastructure layer**

The infrastructure layer is made up of interconnected network pieces. Data links enable interconnection inside the infrastructure. Each connection connects a port on one network device to another's port. A network element computes, stores, and forwards network traffic. Network parts include virtual or hardware switches, routers, middle boxes, and other network components. They feature open programmable interfaces. The control layer uses these interfaces to reprogram their behavior and retrieve their local states. They deal directly with network traffic, performing forwarding, routing, caching, packet inspection, and other networking functions.

- **Southbound API**

The southbound API provides an open, customizable interface between the infrastructure and control layers. It facilitates interactions between the two layers by providing all required abstractions and protocols. It has two primary roles. It gives the control layer abstractions for interacting with network elements. Furthermore, it executes control layer directives on network elements. This standardized, open interface enables the control layer to reprogram the infrastructure layer's forwarding behavior and determine its status. The access plane layer is typically configured using flow rules. Flow rules consist of a set of matching fields linked with actions, as well as instructions indicating how the network element should carry out these actions. Flow rules are stored in OpenFlow tables on the network elements.

- **Control layer**

The control layer is considered the network's brain. It is a conceptually centralized entity that uses interfaces, network operations, network status, and a development environment to govern network element behavior. It abstracts network infrastructure, analyzes network status, and generates network knowledge. Network programs employ these abstractions to reprogram network components. Furthermore, SDN controllers transform application policies into low-level rules. They verify that the rules installed in network elements are consistent and valid. They also monitor network elements and traffic by collecting network status, network events, metrics, and other information.

- **Northbound API**

The northbound API provides a configurable interface between network controllers and applications. It enables the latter to communicate with the control plane and distribute its requirements throughout the network architecture. This abstraction enables network applications to fully leverage SDN capabilities such as global awareness, programmability, automation, and policy deployment. It provides network applications with universal data models, network states, management information, and other capabilities. These programs use the interface to change the behavior of network nodes and update the network state.

- **Application layer**

The application layer is located at the top of the SDN architecture. It includes the network applications. It abstracts the business logic, requirements, and goals of network stakeholders. This layer frequently communicates with third-party apps, administrators, developers, and other stakeholders. It converts these interactions into network policies and tactics. It then delivers them to the control layer via the northbound interface in order to configure or gather status of the network infrastructure.

- **Management layer**

The management layer coordinates the configuration and orchestration of software and hardware resources across the other SDN layers. It also communicates with administrators, allowing them to control and monitor SDN layers. The Manager looks after the SDN software and hardware resources. It delivers them in the right hardware and configurations. It modifies them based on network traffic and business logic. It manages their entire lifecycle. It tracks the use, operation, and quality of SDN resources. It gathers domain knowledge from every controller. It combines this knowledge with monitoring data to create a complete picture of the network.

3.2 Vulnerabilities in SDN

SDN offer numerous advantages over traditional networks. The separation of the control and data transport layers, as well as the centralization of network control in an SDN controller, enables more efficient and consistent network management. Thanks to their programmability, SDN networks allow administrators to dynamically define and adjust network policies, facilitating the automation of network functions and reducing operating costs [13].

Furthermore, the increased flexibility of SDN networks enables faster response to changing network and application needs, accelerating time to market for new applications. Improved responsiveness to problems and outages also contributes to higher network availability. SDN optimizes performance by directing traffic more efficiently and avoiding bottlenecks (Figure 2).

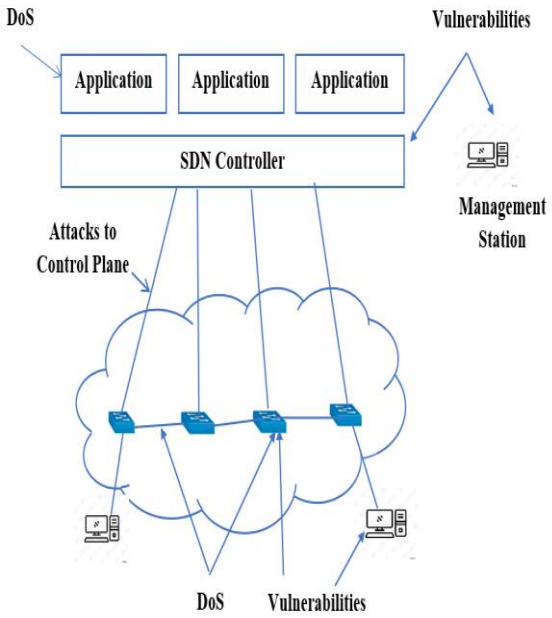

Figure 2: Vulnerabilities in SDN architecture

- **Between the network device and the host**

There are various ways to maliciously exploit network devices in the SDN data plane. A malicious host can attack any SDN switch or controller by generating forged network packets.

- **On the network device**

In SDN networks, the controller installs flow rules in a switch's flow tables, either proactively or reactively. However, switches have a limited number of flow table entries, posing a challenge in managing flow rules efficiently. Since SDN separates the control plane from the data plane, a critical security issue for OpenFlow switches is differentiating between legitimate and malicious flow rules. Additionally, the number of flow entries a switch can handle is a limiting factor. OpenFlow switches must buffer data flows while awaiting flow rules from the controller, making them vulnerable to flooding attacks. Due to resource constraints, an OpenFlow switch can struggle to buffer large volumes of unwanted UDP or TCP traffic, increasing its susceptibility to such attacks.

- **Between network devices**

SDN switch link communications are not encrypted, creating a vulnerability that enables attackers to intercept transmitted data, compromising network security.

- **Between controller and network device**

In SDN, if a switch fails to receive flow routing instructions from the controller—whether due to a controller failure or disconnection—the data plane becomes inoperable. This makes the link between the switch and the controller a prime target for attackers, necessitating strong security measures. Additionally, OpenFlow 1.3 does not natively encrypt communications between the controller and network devices using SSL or

TLS, nor does it enforce controller-level switch authentication. This lack of encryption and authentication exposes the entire SDN network to security threats, including unauthorized data access, man-in-the-middle attacks, eavesdropping, and manipulation of flow rules [14].

- **Between controllers**

Links between SDN controllers are exposed to various security threats. To ensure redundancy, controllers exchange update information; however, the absence of encryption on these links allows attackers to intercept and manipulate critical updates, compromising network integrity and security.

- **On the controller**

SDN threats arise from the separation of the control and data planes, as well as the reliance on a logically centralized controller. As the core component responsible for network control, the SDN controller is a prime target for attacks. Applications running on top of the controller can introduce security risks if not properly managed. To mitigate these risks, the controller must implement strong authentication and authorization mechanisms for applications, ensuring they are segregated based on their security impact before accessing network information. Additionally, a compromised workstation within the same network as the SDN controller can facilitate direct attacks, potentially leading to full network compromise. Furthermore, attackers can exploit the controller's ability to interpret and enforce network policies to orchestrate large-scale attacks. Given its central role in managing traffic, the SDN controller can also become a performance bottleneck, further impacting network resilience [15].

- **Between applications and the controller**

Unlike the standardized Southbound API, the Northbound API lacks a universal standard and has inherent security weaknesses. The absence of robust trust mechanisms in the Northbound API, which facilitates communication between applications and the controller, introduces potential vulnerabilities. Weak authentication methods and inadequate authorization controls can enable attackers to impersonate legitimate applications or gain unauthorized access, posing a significant security risk [16].

- **On the application**

Attackers often exploit applications to gain control over an SDN controller, a risk amplified by the fact that both applications and controller software are typically hosted on the same physical system. This allows malicious code to be injected into the control software through the Northbound API, which remains a vulnerable point of entry. The combination of untrusted applications and inherent API weaknesses enables an attacker to compromise the controller, ultimately allowing them to manipulate network rules and take control of the entire SDN infrastructure.

4 Threat modeling

Recent high-profile security breaches have demonstrated that reactive security solutions are inadequate. Proper threat modeling could have avoided some of these incidents [17],[18]. Various threat modeling tools are available, including DREAD, PASTA, OWASP Threat Dragon, and STRIDE [19].

- **DREAD**

The DREAD threat model is a risk assessment system that enables businesses to measure, compare, and prioritize the risk of security threats. The term DREAD stands for Damage, Reproducibility, Usability, Affected Users, and Discoverability. Each component contributes to a thorough assessment of potential security vulnerabilities, allowing teams to determine informed resource allocation and mitigation measures. DREAD, which was initially established as part of Microsoft's Security Development Lifecycle (SDL), has since become a widely adopted approach across a variety of sectors. Although Microsoft has since embraced alternative threat modeling methodologies, DREAD remains relevant due to its simplicity and practical application in a wide range of settings.

- **PASTA**

The PASTA abbreviation stands for Process for Attack Simulation and Threat Analysis. PASTA is a seven-step threat modeling methodology that integrates business objectives and technical requirements to deliver a comprehensive risk assessment of potential threats. Unlike other threat modeling methodologies, which may focus solely on technical vulnerabilities, PASTA adopts a comprehensive approach that considers both business effect and technological concerns. This comprehensive approach makes it especially effective in company situations where security decisions must be consistent with business objectives. The PASTA methodology is iterative and flexible, allowing organizations to tailor it to their own requirements while retaining a structured approach to threat assessment. By emphasizing risk-based analysis, PASTA assists organizations in prioritizing security investments and focusing on protecting their most valuable assets.

- **OWASP Threat Dragon**

OWASP Threat Dragon is a threat modeling tool designed to create threat model diagrams within the secure development lifecycle. Aligned with the principles of the Threat Modeling Manifesto, it helps document potential threats, define mitigation strategies, and visually represent threat model components and attack surfaces. Available as both an online and desktop application, Threat Dragon facilitates comprehensive threat analysis and security planning.

- **STRIDE**

STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. STRIDE is commonly used in

cybersecurity to analyze potential security risks in applications, networks, and systems [20].

Each strategy aims at a specific perspective and will be more relevant and effective in some contexts than others; this paper will focus on STRIDE as a methodology. Microsoft developed the STRIDE threat model, which has emerged as one of the most effective models for proactive security planning. The STRIDE is a systematic approach to security that encourages development teams to think like hackers in order to defend their systems before they are breached [21]. Unlike DREAD, which primarily focuses on scoring and prioritizing threats based on impact and exploitability without offering a systematic method for discovering them, STRIDE enables a comprehensive mapping of threats to specific elements within the SDN system. While PASTA offers an attacker-centric and risk-driven methodology suitable for enterprise-level threat modeling, it requires extensive contextual and business-driven inputs, which can be complex and less adaptable in SDN infrastructure scenarios. Similarly, OWASP Threat Dragon, although user-friendly and valuable for visual modeling, is primarily a tool rather than a full framework, and it often depends on the underlying threat model being applied—such as STRIDE itself. Therefore, STRIDE was chosen for its clarity, ease of integration with data flow diagrams, and its alignment with technical threat categorization, making it particularly effective for identifying and structuring threats in programmable, software-driven networks like SDN. The STRIDE model divides threats into six categories, each addressing a different component of software security risk (Table 2).

Table 2: STRIDE model threat and security objective violation

Threat	Security objective violation
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of Service	Availability
Elevation of privilege	Authorization

Spoofing: Consider digital identity theft. This entails mimicking another user or system component in order to obtain illegal access. Spoofing attacks exploit authentication methods, allowing hackers to impersonate genuine users or devices.

Tampering: Tampering refers to the unlawful modification of data or code. Such assaults might jeopardize data integrity by modifying files, databases, software code, deployment pipelines, or memory in live systems. Tampering with any system carries significant hazards, particularly when data accuracy is crucial for decision-making.

Repudiation: Threats of repudiation take advantage of accountability gaps. This type of security danger happens when a user or system refuses to complete a certain task, such as a transaction. This threat takes advantage of a lack of non-repudiation measures in software systems, making it harder to hold parties accountable for their behavior.

Information disclosure: This is the unintended disclosure of confidential or sensitive information to unauthorized people. This could be due to insufficient encryption, inappropriate access controls, or vulnerabilities in web applications.

Denial of Service: This type of security threat attempts to disrupt service availability by overloading the system with excessive requests or exploiting system weaknesses. DoS attacks make systems unavailable to legitimate users and disrupt company operations.

Elevation of privilege: This happens when a hacker gains unauthorized access, typically by exploiting a system vulnerability. This can result in administrative control over a system, allowing the attacker to install malware, change system settings, or view sensitive data.

5 Proposed methodology

In this section, we introduce the steps of our proposed methodology, including the DFD diagram, threat identification, risk assessment, and threat mitigation.

5.1 Data flow diagram

A graphical representation of the SDN architecture is called a Data Flow Diagram (DFD). Understanding the sequence of information exchanges and, more importantly, identifying the vulnerabilities and threats associated with the architecture's assets is made easier by simplifying the processes using the DFD. This aids in proposing a mitigation and correction plan that may involve reducing, eliminating, or avoiding the threat altogether to better manage its potential impact if exploited [22]. To construct and analyze the DFD, we used the Microsoft Threat Modeling tool (MTM) with Azure Threat Modeling Tool (ATMT) version 1.0.0.33, which is a security design analysis tool developed by Microsoft. It supports STRIDE-based threat identification by automatically generating potential threats based on defined elements such as processes, data stores, data flows, and external entities. This process aids in proposing a mitigation and correction plan that may involve reducing or eliminating the threat entirely or avoiding it to better manage the potential impact in the event that it is exploited.

In Figure 3, rectangular shapes with solid black borders represent key network components, such as user hosts, SDN infrastructure devices (e.g., switches and host devices), and software entities. The circular shape

represents the orchestration and management application, which serves as an intermediary layer for communication between the user's host and the SDN controller. Green rounded rectangles indicate the flow of communication between components, distinguishing requests and responses. The red dotted box outlines the Network Domain, which includes essential SDN data plane elements such as the switch and host device. These components interact directly under the supervision of the SDN controller, which resides outside the network domain, representing the centralized control plane. Starting from the top left, the human user interacts with the user's host, which then communicates with the orchestration and management application. This application forwards requests to the SDN controller, which then controls and manages the data flow to the switch and host device within the network domain. This layout visually represents the logical separation between the application, control, and data planes, central to the SDN architecture.

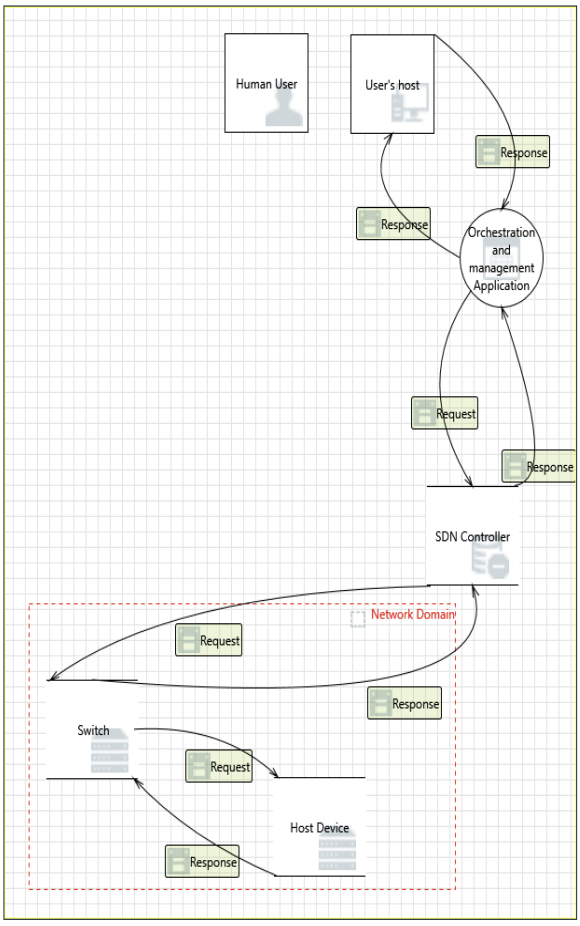

Figure 3: Data flow diagram of SDN network using Microsoft Threat Modeling tool

5.2 Threat identification

Threat identification comes next once the threat modeling approach is used, as seen in Figure 3.

A thorough threat report was produced for every DFD component by using the MTM tool's STRIDE threat modeling technique. All of the dangers that were found were then individually recorded in sub-section 5.1 of the findings. These risks illustrate how different threats can compromise particular components. Additionally, we described the SDN assets affected by each STRIDE threat and how they relate to violations of security requirements. We also examined which threats could result in attacks after classifying all of the threats found in each SDN zones using the STRIDE approach.

5.3 Risk assessment

A crucial part of threat modeling is risk assessment, which helps organizations efficiently prioritize and address possible risks. Threat modeling is an organized method for enhancing system security by methodically identifying and evaluating hazards [23]. The CVSS and the 5x5 risk matrix are two popular techniques for measuring and visualizing risks. A standardized framework for assessing vulnerability severity based on variables including effect, exploitability, and environmental changes is offered by CVSS. In addition, the 5x5 matrix provides a simple and easy-to-use tool for decision-making by visualizing risks by classifying them based on likelihood and impact. By combining these techniques, security teams may strike a compromise between practical prioritization and quantitative accuracy, guaranteeing that mitigation efforts concentrate on the most serious risks [24]. This article explores how CVSS and the 5x5 matrix can be integrated, emphasizing how they work together to create a thorough framework for risk assessment.

5.4 Threat mitigation

Proposing suitable mitigation strategies comes after assessing risks and detecting threats in SDN network. Threat mitigation is the process of lowering or getting rid of possible hazards in a system. We examined a number of current methods in order to create effective mitigation measures. Based on these investigations, we chose the best solutions to safeguard the SDN network from these possible dangers, as covered in section 5.3.

6 Results and discussion

In this section, we detect threats and test the risk assessment using the STRIDE technique through experiments using the MTM tool. As was previously said, STRIDE uses the use case to map and classify the threats that have been identified. We applied the STRIDE threat modeling approach in our SDN architecture to systematically identify security vulnerabilities across the control, data, and application planes. Based on the findings, we proposed mitigation strategies to enhance the resilience of the SDN framework against potential attacks.

6.1 Threats identification

In this part, we discuss all the threats identified by STRIDE tool with respect to each zone of the SDN architecture (Table 3).

Table 3: Threat identification following STRIDE approach

SDN zones	STRIDE	Threats	Mapped DFD Element
SDN network element	Spoofing	**T1:** An attacker may spoof the Host Device, resulting in erroneous data being transmitted to the switch.	Host device
		T2: An attacker may impersonate Switch, causing data to be written to the attacker's target rather than Switch.	Switch Input
	Repudiation	**T3:** Switch states that it did not write data received from another entity over the trust boundary.	Data Handling
SDN controller	Spoofing	**T4:** An attacker may impersonate the SDN Controller, resulting in erroneous data being supplied to the attacker's target.	Controller Input
		T5: An attacker may impersonate the SDN Controller, resulting in false data being supplied to the orchestration and management applications.	Controller Data
		T6: An attacker may spoof the SDN Controller, causing data to be written to the attacker's target rather than the SDN Controller.	Controller Output
	Repudiation	**T7:** SDN Controller asserts that it did not write data received from another entity across the trust boundary.	Trust Boundary
	Information Disclosure	**T8:** Improper data protection on SDN Controllers can allow an attacker to view information that was not intended for dissemination.	Sensitive Data Flow
	Denial of Service	**T9:** An external agent restricts access to a data store on the other side of the trust boundary.	Data Store Access
		T10: An external agent disrupts data flow across a trust border in either direction.	Data Flow Disruption
SDN application	Spoofing	**T11:** An attacker may fake a user's host, resulting in unauthorized access to the Orchestration and Management Application.	User Access
	Tampering	**T12:** The web server 'Orchestration and Management Application' may be vulnerable to a cross-site scripting attack since it does not sanitize untrusted input.	Web Server Input
		T13: The web server 'Orchestration and Management Application' may be vulnerable to a persistent cross-site scripting attack since it does not sanitize data store 'SDN Controller' input and output.	Data Interaction
	Elevation of Privilege	**T14:** Orchestration and Management the application may be able to spoof the context of the user's host in order to gain further privileges.	User Context

6.2 Risk assessment

In order to preserve resources and assets, lower financial losses, enhance decision-making, and other goals, risk assessment entails evaluating the threats found through threat modeling, quantifying the risks, and implementing mitigation strategies.

- **CVSS calculator 3.1**

The Common Vulnerability Scoring System (CVSS), which assigns a risk score on a range of 0 to 10 in ascending order based on the severity and effect of the vulnerability, can be used to evaluate the risks associated with the threats and vulnerabilities mentioned above.

Table 4 presents how the score is divided according to the severity and the criticality hierarchy of the vulnerability.

Table 4: Allocation of vulnerability severity scores according to CVSS.

Severity	Score
None	0
Low	0.1 ➔ 3.9
Medium	4.0 ➔ 6.9
High	7.0 ➔ 8.9
Critical	9.0 ➔ 10.0

Following CVSS v3.1 calculator, the score is calculated by calling up the following parameters in Figure 4 and Table 5.

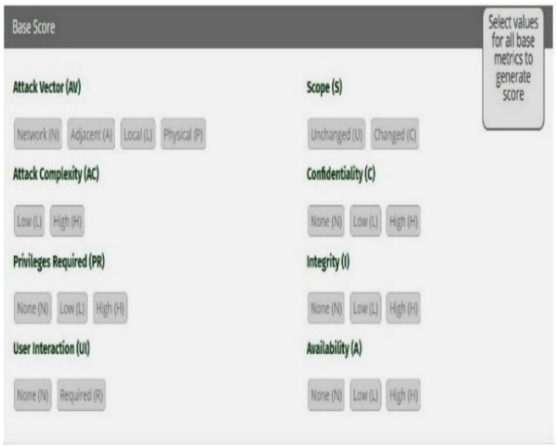

Figure 4: Common vulnerability scoring system version 3.1 calculator base score

- **AV (Attack Vector)**: N (Network), A (Adjacent), L (Local), P (Physical)
- **AC (Attack Complexity)**: L (Low), H (High)
- **PR (Privileges Required)**: N (None), L (Low), H (High)
- **UI (User Interaction)**: N (None), R (Required)
- **S (Scope)**: U (Unchanged), C (Changed)
- **C (Confidentiality Impact)**: N (None), L (Low), H (High)
- **I (Integrity Impact)**: N (None), L (Low), H (High)
- **A (Availability Impact)**: N (None), L (Low), H (High)

The bar chart in figure 5 illustrates the frequency and severity of CVSS-evaluated threats across different SDN architecture zones: Network Elements, Controller, and Applications.

Table 5: Risk management table (CVSS v3.1 base score)

Threat	AV	AC	PR	UI	S	C	I	A	Score	Severity
T1	N	L	N	N	U	L	L	N	6.5	Medium
T2	N	L	N	N	U	N	H	N	7.5	High
T3	N	H	L	N	U	N	L	N	3.1	Low
T4	N	L	N	N	U	H	N	N	7.5	High
T5	N	L	N	N	U	L	L	N	6.5	Medium
T6	N	L	N	N	U	N	H	N	7.5	High
T7	N	H	L	N	U	N	L	N	3.1	Low
T8	N	H	L	N	U	H	N	N	5.3	Medium
T9	N	L	N	N	U	N	N	H	7.5	High
T10	N	L	N	N	U	N	N	H	7.5	High
T11	N	L	N	N	U	H	H	H	9.8	Critical
T12	N	L	N	R	C	L	L	N	6.1	Medium
T13	N	L	N	R	C	L	L	N	6.1	Medium
T14	N	L	L	N	U	H	H	H	8.8	High

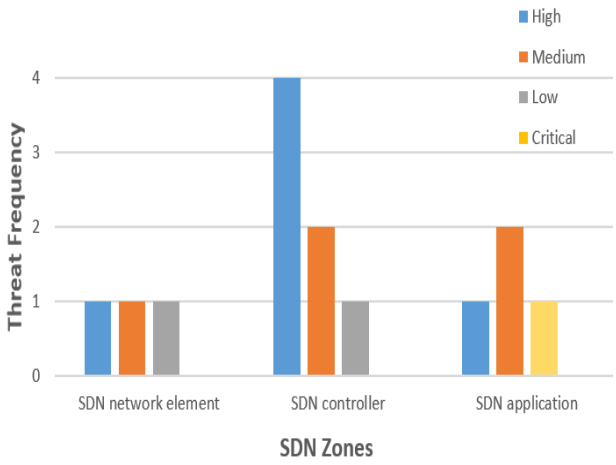

Figure 5: Frequency and severity of threats by SDN zones

The SDN Controller zone emerges as the most targeted, with several High severity threats, including spoofing and DoS attacks (T4–T10). The Application zone, though having fewer threats overall, contains the most severe one (T11 - Critical), involving user impersonation. Interestingly, while the Network Element zone hosts fewer threats, it includes Medium and High severities, suggesting vulnerabilities in identity verification mechanisms. The chart underscores the SDN Controller's critical role in the network and the need to harden it against identity- and access-based threats. The Application zone also requires attention, particularly regarding secure development practices to avoid injection-based vulnerabilities.

The severity distribution histogram across STRIDE categories reveals that Spoofing and DoS are the most critical threat categories in the SDN architecture, with Spoofing showing the highest frequency and including multiple High and Critical severity threats, indicating significant risks related to identity impersonation and unauthorized access. DoS threats also rank high in severity, highlighting vulnerabilities in system availability and resilience. In contrast, other categories such as Repudiation, Information Disclosure, Tampering, and Elevation of Privilege appear less frequent and mostly of Low to Medium severity, suggesting they are comparatively less impactful in the analyzed context. This distribution underscores the need to prioritize robust authentication, controller validation, and availability safeguards in SDN security strategies (Figure 6).

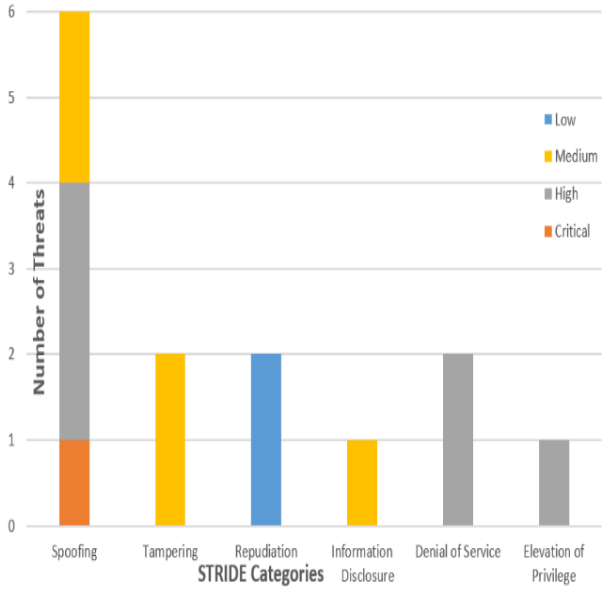

Figure 6: Severity distribution across the STRIDE categories

- **5 by 5 Matrix**

Prior to undertaking the evaluation, assets should be identified and prioritized using the 5 by 5 Risk Matrix, a helpful tool for risk assessment that combines threat impact and likelihood ranked from low to extreme. The likelihood is the possibility that the risk will materialize, whereas the impact is the severity of the consequences if the risk materialized. According to the matrix in Figure 7, the risk is color-coded in green, yellow-low, orange, and red, and can be classified as low, medium, high, or extreme using the formula:

Risk rating = Impact * Likelihood

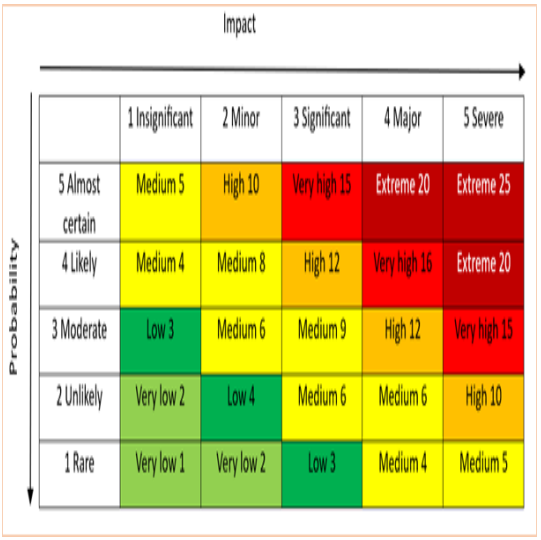

Figure 7: 5 by 5 risk matrix

Based on the 5x5 matrix, a threat assessment was carried out, and summarized in the table 6.

Table 6: Risk rating based on 5 by 5 risk matrix

Threat	Likelihood	Impact	Risk rating
T1	Likely (4)	Major (4)	Very High (16)
T2	Likely (4)	Severe (5)	Extreme (20)
T3	Moderate (3)	Minor (2)	Medium (6)
T4	Almost Certain (5)	Major (4)	Extreme (20)
T5	Almost Certain (5)	Severe (5)	Extreme (25)
T6	Likely (4)	Severe (5)	Extreme (20)
T7	Moderate (3)	Minor (2)	Medium (6)
T8	Likely (4)	Significant (3)	High (12)
T9	Unlikely (2)	Major (4)	Medium (6)
T10	Unlikely (2)	Major (4)	Medium (6)
T11	Almost Certain (5)	Major (4)	Extreme (20)
T12	Likely (4)	Significant (3)	High (12)
T13	Likely (4)	Significant (3)	High (12)
T14	Likely (4)	Severe (5)	Extreme (20)

Because the SDN system requires a high level of vigilance regarding data confidentiality, integrity, and availability—given the critical nature of the information circulating within the architecture and the potential consequences of compromise—any incorrect decision-making or unauthorized access could lead to severe network disruptions. As a result, the risk assessment of threats is often rated very high, whether using the CVSS framework or a 5x5 risk matrix, thereby increasing the overall risk level.

6.3 Threat mitigation techniques

It's time to provide a list of mitigation strategies to lessen the potential harm that could result from one of the threats being exploited after they have been recognized and categorized by zones using the STRIDE approach.

- **SDN network element**

Implement standard mutual authentication using CA-signed certificates or pre-shared keys to prevent spoofing of switches and other elements in both the control and data planes. Use secure boot and firmware validation to ensure the integrity of devices. Enable packet-level logging and auditing with timestamps and source identifiers to detect anomalies. To manage resource use, apply rate-limiting on packet-in messages from switches to the controller to prevent control plane flooding.

- **SDN controller**

Secure the SDN controller by enforcing TLS-based mutual authentication for all communication with switches and applications. Apply role-based access control (RBAC) to restrict administrative privileges and prevent unauthorized actions. Use encryption with secure key management for sensitive data exchanges and configuration files. Isolate critical modules and northbound applications using containerization or sandboxing techniques. Monitor for anomalies such as excessive flow table misses using Intrusion Detection Systems (IDS). Enforce timeout-based flushing of unused flow rules and resource caps per switch or application to mitigate denial-of-service (DoS) risks and preserve controller availability.

In terms of practical deployment, ONOS supports mitigations through its SecurityMode configuration for mutual TLS, Intent Framework for flow rule enforcement and application isolation, and the ability to implement rate-limiting via custom intents. Floodlight enables secure communication through mutual TLS, enforces flow rules, and implements RBAC for access control. It also supports API security and application isolation. OpenDaylight, with its AAA security plugin, provides RBAC, secure REST APIs, mutual TLS, and Karaf-based application isolation, offering a robust platform for enforcing security controls. Ryu, using OpenFlow TLS authentication, allows flow rule enforcement, rate-limiting, and secure RESTful API access, making it a versatile controller for securing SDN environments.

- **SDN application**

Strengthen application security by enforcing mutual authentication and message signing between applications and the controller. Implement strict API access policies to limit interaction only to authorized modules. Utilize application whitelisting and run-time monitoring to detect unexpected behavior. Use rate-limiting and resource quotas on northbound requests to prevent application-induced overload. Isolate applications using virtual machines or containers, and enforce code-signing and verification before deployment.

6.4 Discussion

The results from the STRIDE and CVSS 5x5 matrix highlight a range of threats across different SDN zones, with varying levels of risk severity. The CVSS scores provide a nuanced view of the potential impact and exploitability of each threat, allowing us to prioritize mitigations effectively. Threats such as T11, with a critical CVSS score of 9.8, point to vulnerabilities with a significant and broad impact on confidentiality, integrity, and availability, indicating an urgent need for countermeasures. Similarly, threats like T2, T4, and T6, with high severity scores (7.5), suggest vulnerabilities that could lead to severe data corruption or denial of service, requiring immediate attention but possibly with less widespread consequences than T11. On the other hand, threats such as T3 and T7, with low CVSS scores (3.1), may have limited consequences and are lower priority for remediation, yet they still need to be addressed as part of a comprehensive security strategy. The CVSS severity classification—ranging from Low to Critical—guides the prioritization of resources, ensuring that high-risk vulnerabilities are handled first. By combining both STRIDE threat modeling and CVSS scoring, organizations can better understand the threat landscape in SDN environments and make informed decisions about which vulnerabilities to address based on their exploitability, impact, and potential consequences.

7 Conclusion

This paper highlights the security threats that can compromise an SDN-based network architecture. Given the critical role of SDN in managing and optimizing modern network infrastructures, any compromise in its security can lead to severe disruptions, including unauthorized access, traffic manipulation, and large-scale network failures. This underscores the importance of modeling threats before deploying an SDN architecture to systematically assess potential risks and develop effective mitigation strategies. Our work provides a structured approach to threat identification, risk assessment, and countermeasure implementation in SDN environments. By leveraging the STRIDE framework, we systematically analyze vulnerabilities specific to SDN, particularly those affecting the control plane, data plane, and communication channels. The proposed methodology offers valuable insights for securing SDN networks and serves as a reference for further research in this domain.

While this study presents a structured threat modeling and risk assessment framework for SDN using STRIDE, CVSS 3.1, and a 5×5 risk matrix, several limitations exist. First, the model has not been empirically validated through real-world penetration testing or deployment in a live SDN environment. Second, the use of STRIDE is inherently static, lacking adaptability to evolving threats or real-time risk propagation. Third, the framework does not model dynamic or intelligent threat actors capable of adjusting their strategies. Future research could address these gaps by integrating AI-based threat prediction techniques, aligning the analysis with the MITRE

ATT&CK framework to improve operational relevance, and enabling real-time DFD generation from SDN controller logs to maintain an up-to-date security posture.

References

[1] R. Chaudhary, F. S. Aujla, N. Kumar, and P. K. Chouhan, "A comprehensive survey on software-defined networking for smart communities", International Journal of Communication Systems, Vol. 38, No. 1, p. e5296, 2025. https://doi.org/10.1002/dac.5296

[2] M. Aldaoud, D. Al-Abri, M. Awadalla, and F. Kausar, "Leveraging ICN and SDN for future internet architecture: a survey", Electronics, Vol. 12, No. 7, p. 1723, 2023. https://doi.org/10.3390/electronics12071723

[3] Y. Maleh, Y. Qasmaoui, K. El Gholami, Y. Sadqi, and S. Mounir, "A comprehensive survey on SDN security: threats, mitigations, and future directions", Journal of Reliable Intelligent Environments, Vol. 9, No. 2, pp. 201-239, 2023. https://doi.org/10.1007/s40860-022-00171-8

[4] A. Liatifis, P. Sarigiannidis, V. Argyriou, and T. Lagkas, "Advancing sdn from openflow to p4: A survey", ACM Computing Surveys, Vol. 55, No. 9, pp. 1-37, 2023. https://doi.org/10.1145/3556973

[5] M. Erbas, S. M. Khalil, and L. Tsiopoulos, "Systematic literature review of threat modeling and risk assessment in ship cybersecurity", Ocean Engineering, Vol. 306, p. 118059, 2024. https://doi.org/10.1016/j.oceaneng.2024.118059

[6] I. Ivkić, D. Thiede, N. Race, M. Broadbent, and A. Gouglidis, "A Security Evaluation Framework for Software-Defined Network Architectures in Data Center Environments", arXiv preprint arXiv:2304.05776, 2023. https://doi.org/10.5220/0011988300003488

[7] C. Oredola and A. Ashraf, "A Systematic Mapping Study on SDN Controllers for Enhancing Security in IoT Networks", In: 2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Paris, France, pp. 317-324, 2024. https://doi.org/10.1109/seaa64295.2024.00056

[8] P. K. Sharma and S. S. Tyagi, "Security Enhancement in Software Defined Networking (SDN): A Threat Model", International Journal of Advanced Computer Science and Applications, Vol. 12, No. 9, pp. 208-217, 2021. https://doi.org/10.14569/ijacsa.2021.0120925

[9] A. Lekidis and I. Barosan, "Model-based simulation and threat analysis of in-vehicle networks", In: 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), Toulouse, France, pp. 1-8, 2019. https://doi.org/10.1109/wfcs.2019.8757968

[10] T. Eom, J. B. Hong, A. Seongmo, J. S. Park, and D. S. Kim, "A systematic approach to threat modeling and security analysis for software defined networking", IEEE ACCESS, Vol. 7, pp. 137432-137445, 2019. https://doi.org/10.1109/access.2019.2940039

[11] K. Kaur, V. Mangat, and K. Kumar, "A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture", Computer Science Review, Vol. 38, p. 100298, 2020. https://doi.org/10.1016/j.cosrev.2020.100298

[12] F. Zobary, "Optimizing SDN Controller to Switch Latency for Controller Placement Problem", Informatica, Vol. 48, No. 8, pp. 165-176, 2024. https://doi.org/10.31449/inf.v48i8.5846

[13] S. Aouad, I. El Meghrouni, Y. Sabri, A. Hilmani, and A. Maizate, "Security of software defined networks: evolution and challenges", Int J Reconfigurable & Embedded Syst ISSN, Vol. 2089, No. 4864, p. 4864, 2023. https://doi.org/10.11591/ijres.v14.i1.pp26-34

[14] J. Kim, M. Seo, S. Lee, J. Nam, V. Yegneswaran, P. Porras, G. Guofei, and S. Shin, "Enhancing security in SDN: Systematizing attacks and defenses from a penetration perspective", Computer Networks, Vol. 241, p. 110203, 2024. https://doi.org/10.36227/techrxiv.22065620.v2

[15] B. Ayodele and V. Buttigieg, "SDN as a defence mechanism: a comprehensive survey", International Journal of Information Security, Vol. 23, No. 1, pp. 141-185, 2024. https://doi.org/10.1007/s10207-023-00764-1

[16] M. Rahouti, K. Xiong, Y. Xin, S. K. Jagatheesaperumal, M. Ayyash, and M. Shaheed, "SDN security review: Threat taxonomy, implications, and open challenges", IEEE ACCESS, Vol. 10, pp. 45820-45854, 2022. https://doi.org/10.1109/access.2022.3168972

[17] A. R. Mahlous, A. R. (2023). "Threat model and risk management for a smart home IoT system", Informatica, Vol. 47, No. 1, pp. 51-64, 2023. https://doi.org/10.31449/inf.v47i1.4526

[18] W. Xiong and R. Lagerström, "Threat modeling–A systematic literature review", Computers & security, Vol. 84, pp. 53-69, 2019. https://doi.org/10.1016/j.cose.2019.03.010

[19] N. Naik, P. Jenkins, P. Grace, D. Naik, S. Prajapat, and J. Song, "A comparative analysis of threat modelling methods: STRIDE, DREAD, VAST, PASTA, OCTAVE, and LINDDUN", In: The International Conference on Computing, Communication, Cybersecurity & AI, London, UK, Cham: Springer Nature Switzerland, pp. 271-280, 2024. https://doi.org/10.36227/techrxiv.173014171.11449253/v1

[20] P. Das, M. R. A. Asif, S. Jahan, K. Ahmed, F. M. Bui, and R. Khondoker, "STRIDE-Based Cybersecurity Threat Modeling, Risk Assessment and Treatment of an In-Vehicle Infotainment System", Vehicles, Vol. 6, No. 3, pp. 1140-1163, 2024. https://doi.org/10.3390/vehicles6030054

[21] M. Ouaissa and M. Ouaissa, "Analyzing and Mitigating Attacks in IoT Smart Home Using a Threat Modeling Approach-Based STRIDE", International Journal of Interactive Mobile Technologies, Vol. 19, No. 2, 2025. https://doi.org/10.3991/ijim.v19i02.52377

[22] I. Hossain, N. I. Chowdhury, and R. Hasan, R. "How Secure is AI-based Coding? A Security Analysis Using STRIDE and Data Flow Diagrams", In: 2023 IEEE Virtual Conference on Communications (VCC), NY, USA, pp. 56-61, 2023. https://doi.org/10.1109/vcc60689.2023.10474718

[23] J. K. Debnath and D. Xie, "CVSS-based vulnerability and risk assessment for high performance computing networks", In: 2022 IEEE International Systems Conference (SysCon), Montreal, QC, Canada, pp. 1-8, 2022. https://doi.org/10.1109/syscon53536.2022.9773931

[24] A. Vaezi, S. Jones, and A. Asgary, "Integrating Resilience into Risk Matrices: A Practical Approach to Risk Assessment with Empirical Analysis", Journal of Risk Analysis and Crisis Response, Vol. 13, No. 4, pp. 252-272, 2023. https://doi.org/10.54560/jracr.v13i4.411