ThreatFinderAI: Automated Threat Modeling Applied to LLM System Integration

Jan Von der Assen[1], Alberto Huertas[1], Jamo Sharif[1], Chao Feng[1], Gérôme Bovet[2], Burkhard Stiller[1]
[1]Communication Systems Group, Department of Informatics, University of Zurich UZH, CH–8050 Zürich, Switzerland
[vonderassen, huertas, cfeng, stiller]@ifi.uzh.ch, jamo.sharif@uzh.ch
[2]Cyber-Defence Campus, armasuisse Science & Technology, CH–3602 Thun, Switzerland, gerome.bovet@armasuisse.ch

***Abstract*—Artificial Intelligence (AI) is a rapidly integrated technology, significantly contributing to advancements like 6G. However, its swift adoption raises considerable security concerns. Large Language Models (LLMs) pose risks such as spear phishing, code injections, and remote code execution. Conventional threat modeling, used in secure software development, faces challenges when applied to AI systems, as existing methodologies are designed for traditional software. Furthermore, AI-specific threat modeling research is sparse and lacks approaches providing practical support or automation. Thus, this demo paper presents *ThreatFinderAI*, an asset-centric threat modeling and risk assessment framework. *ThreatFinderAI* fulfills seven steps aligned with AI system design and transforms AI threat and control knowledge bases into a queryable knowledge graph for automated asset identification and threat elicitation. It also proposes business impact analysis and expert estimates for AI threat impact quantification. In the demonstration, *ThreatFinderAI* is illustrated by securing a customer care application relying on LLMs. Through this, it is demonstrated how the proposed framework can be used to identify relevant threats and practical countermeasures and communicate strategic risk.**

***Index Terms*—Threat Modeling, AI Systems, Large Language Models, AI Security**

I. Introduction

Artificial Intelligence (AI) is a disruptive technology being integrated into various key domains, from healthcare to communication systems such as 6G [1]. The rapid adoption of AI raises significant security concerns. In the case of Large Language Models (LLMs), studies highlight different threats, including spear phishing, code injections, and remote code execution [2]. Similar vulnerabilities are noted in Machine Learning, Federated Learning, or Computer Vision [3].

Therefore, the continuous integration of AI in society necessitates addressing security concerns. One effective method from conventional application security is threat modeling, which is employed in secure software development and risk assessment. However, existing methodologies are inadequate for AI systems due to their conventional software system focus and lack of practical cybersecurity approaches [2], [3].

Thus, this demo paper introduces a user-friendly and automatic asset-centric threat modeling and risk assessment framework called *ThreatFinderAI*. The methodology of *ThreatFinderAI* includes seven steps aligned with AI system design procedures. It transforms community-driven knowledge bases on AI threats and controls into a queryable knowledge graph, enabling automated asset, threat, and control elicitation through a custom stencil library. *ThreatFinderAI* also proposes a business impact analysis method using Monte Carlo simulations and expert estimates for AI threat impact quantification [4]. This demonstration showcases *ThreatFinderAI* in a practical scenario: ensuring security-by-design for a conversational customer care application aiming to integrate LLMs. Here, it is demonstrated how *ThreatFinderAI* can be used for three tasks: to identify relevant threats, discover practical controls, and finally, to assess and communicate strategic risks about the LLM integration project. A video presentation of the demonstration is available in [5].

The remainder of this document is structured as follows. Section II introduces the architecture behind the *ThreatFinderAI* approach, which serves as a baseline for this demonstration. Next, Section III walks the reader through the steps that the demonstration will entail. Finally, Section IV summarizes this work while outlining future work.

II. *ThreatFinderAI*

The design of *ThreatFinderAI* is shown in Fig. 1. As can be seen, the architectural design is composed of ten key components that are proposed to fulfill a common asset-centric threat modeling methodology [4]. Firstly, the *Security Objective* component allows users to indicate the key security principles that should be considered while generating the threat model. *ThreatFinderAI*() offers confidentiality, integrity, availability, authorization, or non-repudiation as principles that could be selected. At this stage, it is essential to understand the business relevance of the system to be developed. This is achieved by defining a specific architecture aligned with a business mission and identifying a key security requirement and architectural asset.

In the second step, the *Diagram Editor* component supports the architectural analysis. This involves contextually modeling the threat environment, whether designing a new architecture or creating a threat model for an existing system. *ThreatFinderAI* suggests using visual architectural modeling, verifying existing system diagrams, and ensuring the AI life cycle's stages are incorporated, as outlined by [6]. A diagram editor and procedure are included to model the entire attack surface, incorporating concepts such as processes, environments, data, and models. Each diagram item is categorized (*e.g.,* data, model, procedure, actor, infrastructure), ensuring machine-processable diagrams through annotated stencils.

978-3-903176-66-9

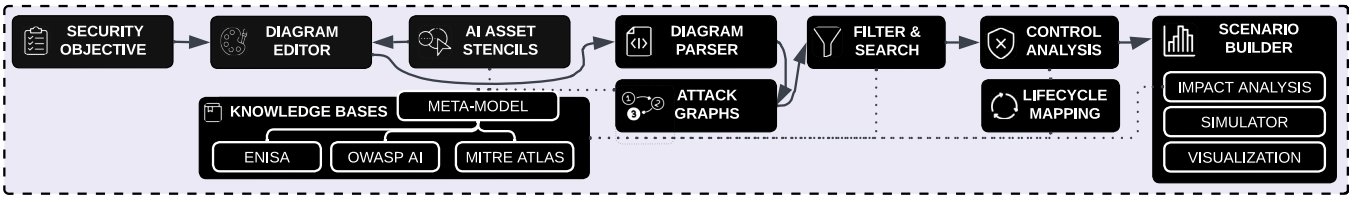

Fig. 1. Architectural Overview of *ThreatFinderAI*

The *AI Assets Stencils* component is used to elicit functional and data assets subject to security goals. This visual approach, familiar in software architecture development, allows for the automated compilation of assets from the diagram parser component and querying them against the meta-model. The *Knowledge Bases* component with identified assets serves as input for threat identification, which identifies related threat events. For instance, untrusted actor-provided training data may indicate vulnerability to a data poisoning attack. The methodology achieves threat identification through an attack graph querying knowledge bases such as ENISA (2020), OWASP AI Exchange, and MITRE ATLAS, transformed into a graph-based form and aligned to the meta-model.

Depending on the scenario and objective, not all threats are equally relevant, necessitating a threat analysis phase where users navigate and prioritize threats, particularly those related to the key asset or objective. These tasks are achieved by the *Filter and Search* component, yielding threats like data inference attacks, allowing users to filter the threat list based on life cycle, impact, or asset. Crucially, threats referring to the key security objective or asset are highlighted.

The *Control Analysis* component identifies technical, organizational, or strategic mitigation controls, querying knowledge bases through the meta-model, which correlates controls with the threat's life cycle stage. For instance, monitoring a model's usage is a countermeasure during the production stage.

Finally, the *Scenario Builder* concludes by supporting risk analysis. In risk assessment contexts, previous threat models can be reused to analyze strategic risks. Quantifying risks requires historical or expert-based data. Since AI security is a novel subset of cybersecurity, *ThreatFinderAI* relies on expert-based opinions. The methodology aids experts in visualizing and communicating AI threats' uncertainty by mapping business impacts to security properties. Monte Carlo simulations of the expert-estimated financial losses and occurrence distributions are used to model and quantify risk scenarios, providing metrics and visualizations to assess residual risk exposure.

Implementing the components involves a web-based solution with a graphical user interface using React.js. Users create a project, define the key asset and security goal, and model the architecture using the diagrams.net editor, which ensures no data leaves the browser. A bespoke XML-based stencil library and querying knowledge graphs facilitate asset elicitation and threat identification. Residual risks are quantified through Monte Carlo simulations in Python, with the results visualized on the front end. The source code is publicly available in [7].

III. Setup and Demonstration

To demonstrate the *ThreatFinderAI* approach and its prototypical implementation, the respective components were deployed to a production environment. The frontend was deployed as a Cloudflare page, with strict access over HTTPS. The backend, used for performing Monte Carlo simulations, was deployed in a VM with 2 GiB of memory and 2 vCPUs. The RESTful API is terminated by an HTTPS-only tunnel. The publicly running instance can be accessed through [8].

In the demonstration, the presenter illustrates the full life cycle of the platform for a particular scenario inspired by a real-world use case, which is being experimented with by Swisscom, the largest telecommunication provider in Switzerland. For the demonstration, a fictitious organization has historically employed conversational agents to help customers answer questions about typical customer care aspects such as products, services, and bills. Due to their limited expressiveness and the wide attention to LLMs, the organization is intrigued whether an LLM-based system architecture could increase the resolution rate. It is assumed that the presenter is tasked with the security analysis of the scenario, through which the following case study questions should be answered:

- ***Q1)*** Which threats arise from the LLM integration?
- ***Q2)*** Which control mechanisms should be considered?
- ***Q3)*** What residual risks remain from the integration, and do they pose a strategic risk?

Thus, first, the presenter creates a new scenario in *ThreatFinderAI* for the previously summarized problem description. This entails the textual description of the project and the definition of a key security objective, which serves as an initial guidance for threat identification. Since the project is not supporting a critical business mission (*e.g.,* sales, network provisioning), *data* is defined as a key asset, and *confidentiality* is defined as a key security objective.

After the scenario is framed, the architectural modeling stage is demonstrated by manually creating a diagram in the platform. To model the system integration of the LLM, three trust boundaries are defined: The web application, which would be publicly accessible to customers; the private cloud of the company where knowledge about products and services is stored; and finally, the third-party LLM provider, which is accessible through an API. For each of these trust boundaries, critical components are modeled using the stencil library provided by *ThreatFinderAI*, including data assets (*e.g.,* customer questions, augmented context from customer forums

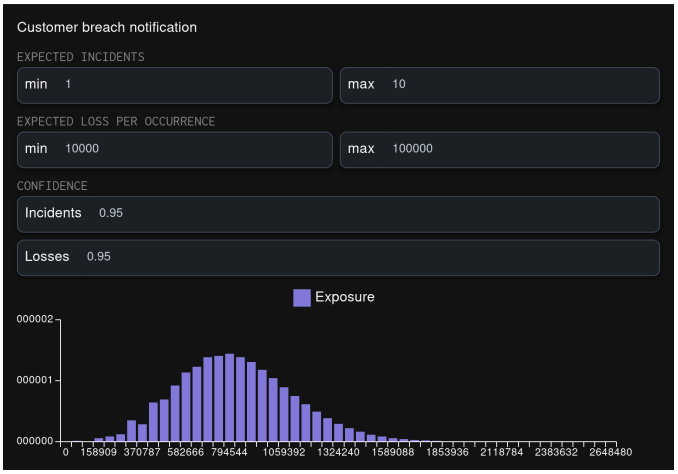

Fig. 2. Visual Risk-based Threat Quantification

and knowledge bases), models (*i.e.,* the third-party LLM), as well as processes and infrastructures (*e.g.,* data splitting, embedding generation, context retrieval, virtual clouds). Using the provided stencils, the architectural pipeline can be modeled holistically, starting from the customer question and concluding with the augmented prompt output.

After modeling the architecture, assets are automatically extracted from the diagram and used to query against the knowledge bases. Based on the asset's life cycle, category, and definition of the key asset, several threats are automatically proposed for the presenter. For example, *customer questions* are highlighted as a potential key asset. To define the threat model, the presenter can interactively filter based on the aforementioned aspects (*e.g.,* only focus on the inference or training stage) and explore the different knowledge bases (*e.g.,* contrast the ENISA report with the OWASP AI Exchange). After reviewing the suggestions, key threats, such as *leaking sensitive data through the prompt to the LLM provider*, are included in the threat model.

In the next stage, threat metadata (*e.g.,* target asset, stage) is used to query the knowledge graph for countermeasures. For each suggested countermeasure, a description and category (*e.g.,* development-time control, governance measure) are provided. The presenter selects controls that apply to the use case. For example, *data minimization* could be applied to the prompts to avoid the case where customers enter personal data into their prompts, which would be forwarded to the LLM provider and lead to a potential confidentiality breach.

At this stage, a technical threat modeling activity could be concluded since it yielded constructive steps (answering ***Q1*** and ***Q2***) to consider for the architectural design. However, to support the discussion of strategic cyber risks, the presenter continues to perform a risk analysis using the threat model. Specifically, the potential effects of a data minimization failure (*i.e.,* leaking personal data to the third-party LLM provider) on the business are demonstrated. To do so, the presenter selects the threat from the threat model, upon which the tool suggests a set of business impacts that could relate to the threat. For example, the *violation of a data protection regulation* is considered. To quantify its impact, the minimum and maximum estimates for the number of incidents and losses are defined. In the demonstration, it is modeled that between one and three incidents could arise over a long-term plan and that each incident could lead to losses between 10'000 and 100'000 USD. By means of the impacts defined and simulations performed, *ThreatFinderAI* is demonstrated to facilitate the discussion of the threat as a strategic risk using business terms (***Q3***).

IV. Summary and Future Work

This demonstration paper introduced *ThreatFinderAI*, an approach to visually model assets and automatically generate threat models while refining them in a guided manner. Furthermore, *ThreatFinderAI* supports control identification and the elicitation of business impacts stemming from the residual risk of the threat and related control measures. Finally, the approach supports threat quantification of business impacts and its visualization. The *ThreatFinderAI* approach was demonstrated for a particular flavor of AI-based system architectures, relying on LLM, which presents a particular threat model due to its complexity and emerging business models. More specifically, the demonstration showcased how a system architecture for a customer care application can be securely developed by identifying threats, mitigating them with control measures, and communicating residual risk.

Due to the modular architecture of *ThreatFinderAI*, further experiments are planned, such as integrating different LLM models as a reasoning agent to create threat models while qualitatively evaluating their performance using experts.

Acknowledgments

This work has been partially supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the CyberMind and RESERVE (CYD-C-2020003) projects and *(b)* the University of Zürich UZH.

References

[1] Nokia Corporation, "6G explained," January 2024, https://www.nokia.com/about-us/newsroom/articles/6g-explained, Last Visit July 10, 2024.

[2] A. Kucharavy, Z. Schillaci, L. Maréchal, M. Würsch, L. Dolamic, R. Sabonnadiere, D. P. David, A. Mermoud, and V. Lenders, "Fundamentals of Generative Large Language Models and Perspectives in Cyber-Defense," arXiv preprint 2303.12132, March 2023.

[3] C. Feng, A. H. Celdrán, J. von der Assen, E. T. M. Beltrán, G. Bovet, and B. Stiller, "Dart: A solution for decentralized federated learning model robustness analysis," *Array*, p. 100360, 2024.

[4] J. von der Assen, J. Sharif, C. Feng, C. Killer, G. Bovet, and B. Stiller, "Asset-centric Threat Modeling for AI-based Systems," pp. 1–8, September (To Appear) 2024.

[5] J. von der Assen and J. Sharif, "Demonstration: Threat Modeling using ThreatFinderAI ," 2024, https://youtu.be/cPt8cyAjtmQ, Last Visit July 10, 2024.

[6] European Union Agency for Cybersecurity (ENISA), "Securing Machine Learning Algorithms," 2021.

[7] J. von der Assen and J. Sharif, "ThreatFinderAI," 2024, https://github.com/jvdassen/ThreatFinder.ai, Last Visit July 10, 2024.

[8] ——, "ThreatFinder," 2024, https://threatfinder-ai.pages.dev/, Last Visit July 10, 2024.