ThreatGPT: An Agentic AI Framework for Enhancing Public Safety through Threat Modeling

Sharif Noor Zisad
Department of Computer Science
University of Alabama at Birmingham
Birmingham, Alabama, USA
szisad@uab.edu

Ragib Hasan
Department of Computer Science
University of Alabama at Birmingham
Birmingham, Alabama, USA
ragib@uab.edu

Abstract— **As our cities and communities become smarter, the systems that keep us safe, such as traffic control centers, emergency response networks, and public transportation, also become more complex. With this complexity comes a greater risk of security threats that can affect not just machines but real people's lives. To address this challenge, we present ThreatGPT, an agentic Artificial Intelligence (AI) assistant built to help people whether they are engineers, safety officers, or policy makers to understand and analyze threats in public safety systems. Instead of requiring deep cybersecurity expertise, it allows users to simply describe the components of a system they are concerned about, such as login systems, data storage, or communication networks. Then, with the click of a button, users can choose how they want the system to be analyzed by using popular frameworks such as STRIDE, MITRE ATT&CK, CVE reports, NIST, or CISA. What makes ThreatGPT special is that it does not just provide generic information, it acts like a knowledgeable partner. Using few-shot learning, the AI learns from examples and generates relevant smart threat models. It can highlight what might go wrong, how attackers could take advantage, and what can be done to prevent harm. Whether securing a city's infrastructure or a local health service, this tool adapts to users' needs. In simple terms, ThreatGPT brings together AI and human judgment to make our public systems safer. It is designed not just to analyze threats, but to empower people to understand and act on them, faster, smarter, and with more confidence.**

Index Terms—**GPT, Threat Modeling, GEMINI, Public Safety, Agentic AI, STRIDE, MITRE ATT&CK, CVE, NIST, NVD**

I. Introduction

Public safety systems form the backbone of modern urban infrastructure. They play an important role in maintaining societal stability by protecting populations during emergencies and supporting essential services such as transportation, healthcare, communication, and power distribution [1]. These systems are becoming smarter every day, thanks to digital technology and connected devices [2]. But as they become more advanced, they also become more vulnerable to cyber threats [3]. A small security weakness in one part of a public system can cause big problems. An attacker might break into a traffic control system and cause chaos on the roads. A failure in communication tools could delay the emergency response. In some cases, private information could be stolen or systems could be shut down. These risks are real and can affect both the systems and the people who depend on them.

To protect public safety, we need to understand the threats before they happen. This process is called threat modeling [4]. It involves looking at the components of a system, figuring out what could go wrong, and finding ways to stop or reduce the impact of those problems. Many well-known models exist to help with this, such as STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) [5], MITRE ATT&CK [6], CVE (Common Vulnerabilities and Exposures) databases [7], NIST (National Institute of Standards and Technology) guidelines [8], and CISA (Cybersecurity and Infrastructure Security Agency) threat reports [9]. These models offer strong methods, but using them can be hard, especially for people who are not cybersecurity experts. Most threat modeling tools today are technical, rigid, or hard to customize. They do not easily fit different types of users or systems [10]. Many require deep security knowledge or manual work to apply the frameworks correctly. For a safety officer, developer, city planner, or healthcare manager, this can be overwhelming. There is a need for a tool that can make threat modeling easier, faster, and more human-centered.

To this end, we have developed ThreatGPT, an Agentic AI [11], [12] assistant designed to help people perform threat analysis in public safety systems. It uses the power of large language models to understand user input, analyze system components, and generate structured threat models based on popular frameworks. It not only gives general advice but also provides targeted, meaningful insights based on the system you describe. What sets ThreatGPT apart is its integration of Agentic AI capabilities with well-established cybersecurity frameworks to deliver a flexible, intelligent, and context-aware threat modeling experience. Unlike conventional tools that offer static checklists or require expert-level knowledge, ThreatGPT functions as an interactive assistant capable of understanding user-defined system components, guiding the analysis process, and adapting to varying levels of technical expertise. It leverages large language models and few-shot learning [13] to generate structured threat assessments while maintaining alignment with industry standard methodologies. This synthesis of generative AI [14] with proven cybersecurity knowledge ensures that ThreatGPT delivers accurate, relevant, and actionable threat models tailored to diverse public safety systems. By enabling users to engage in an intuitive dialogue rather than navigate complex tooling, ThreatGPT democratizes access to high-quality threat analysis and empowers stakeholders from technical teams to public safety officials—to proactively identify risks and strengthen system resilience.

Unlike traditional AI that only responds to commands, agentic AI is designed to act more like a helpful teammate. It can take initiative, ask questions, guide users, and adapt based on context. Instead of waiting for perfect instructions, an agentic AI works with the user—thinking along the way. In the case of ThreatGPT, the AI does not just answer queries, it walks with the user through the full process of threat modeling, suggesting next steps, clarifying missing parts, and making sure nothing important is left out. This makes the experience more natural and supportive, especially for users without deep technical knowledge.

Every public safety system possesses a unique architecture, context, and risk profile. For instance, one user may describe the infrastructure of a hospital's IT network [15], while another may focus on a smart traffic control system. Predefining and hardcoding threat modeling approaches for every potential use case is impractical and limits the flexibility of conventional systems. To address this variability, few-shot learning plays an important role. It enables the AI model to generalize from a small number of user-provided examples, allowing it to adapt dynamically to new system descriptions and generate relevant threat models without requiring extensive retraining or large labeled datasets.

Contribution: The contributions of this paper are as follows:

1) We trained an LLM with STRIDE, MITRE ATT&CK, CVEs, NIST, CISA report to develop ThreatGPT, an Agentic AI system capable of generating extensive threat models for any system specified by the user.
2) We demonstrated how ThreatGPT can support proactive threat analysis in public safety systems by helping users identify risks based on system components.

Organization: The structure of this paper is as follows: Section I covers the importance of threat modeling in ensuring public safety. Section II discusses the background of Agentic AI, Threat Modeling, and the framework we used to train our model, and Section III reviews the related works. In Section IV, the methodology of our proposed model is outlined along with the methodology. The experiments and the results are reviewed in Section V. Finally, Section VI presents the conclusion and discusses future directions.

II. Background

A. Agentic AI

Agentic Artificial Intelligence [11], [12] is a new kind of AI that can take action on its own. It does not wait for step-by-step commands. Instead, it understands the user's goal and helps complete tasks by reasoning and adapting. Unlike traditional AI, which only responds to direct input, agentic AI can guide users, ask questions, and suggest next steps. It acts more like a helpful assistant than just a tool. This makes it useful for solving problems that are complex or not clearly defined. The characteristics of Agentic AI [11] are as follows:

1) Goal driven: These systems are designed with a clear objective in mind and consistently work toward achieving it. Unlike reactive models that only respond to specific prompts, a goal-driven agent proactively pursues outcomes, making decisions and taking actions that align with its overall mission, even if the user does not explicitly request intermediate steps.

2) Planning and Reasoning: This model is capable of creating multi-step plans to achieve its goals. It does not just perform isolated tasks. It logically reasons through the problem, identifies the necessary steps, organizes them in sequence, and adapts its actions as it progresses, much like a human planning a project or solving a complex challenge.

3) Autonomy: One of the core features of Agentic AI is the ability to operate independently without needing constant human instructions. Once given an initial task or goal, an autonomous agent decides what to do next, manages sub-tasks, and resolves issues it encounters, much like a self-driving car making decisions on the road.

4) Adaptability: It can modify its actions and strategies based on new information or changing circumstances. If an obstacle arises or conditions change, the AI doesn't freeze or fail. Instead, it intelligently adjusts its plan to continue pursuing its goal, ensuring resilience in dynamic environments.

5) Memory and State: It maintains memory of past interactions, decisions, and actions to inform its future behavior. This memory enables it to learn from previous experiences, avoid repetitive mistakes, build context over time, and deliver smarter, more consistent results instead of treating every request as completely new and isolated.

Agentic AI works by combining language understanding, memory, and learning. It can follow long conversations and remember what was said earlier. It can also handle missing or unclear information. Some agentic AI systems use few-shot learning [13], where they learn from just a few examples. Others use reinforcement learning [16], where they get better by learning from feedback. These abilities help the AI improve over time and adjust to different users and situations.

This kind of AI is helpful in many real-world areas, such as healthcare, education, and public safety. In these fields, people often deal with systems that are large and complex. They need help making safe and smart choices. Agentic AI can support them by giving useful advice, checking for errors, and offering guidance. It helps users focus on what matters without needing deep technical knowledge. By doing so, agentic AI makes technology more human-friendly and easier to trust.

B. Threat Modeling

Threat modeling is the process of finding and understanding possible security problems in a system [4]. It helps people think about what could go wrong before it actually happens. By looking at the different parts of a system, we can find weak spots that an attacker might try to use. This process helps in building safer and more secure systems. It is an important part of designing systems that protect both data and people.

Threat modeling is useful in many fields, such as finance, healthcare, and public safety. It helps both experts and non-experts understand risks in simple ways. However, doing it manually can take a lot of time and effort. It often requires special knowledge and technical skills. That is why many researchers and developers are now exploring how artificial intelligence can support and improve the threat modeling process, making it faster, easier, and more accessible for everyone. The components of a threat model [17] are:

1) Assets: They are the critical components within a system that require protection from potential threats. These can include sensitive data, software applications, network infrastructure, intellectual property, and user credentials. Identifying assets is essential in threat modeling, as it helps prioritize security measures around the most valuable and vulnerable elements of the system.

2) Entry Points: These are the interfaces or access channels through which users, including potential attackers, can interact with a system. Examples of entry points include web portals, APIs, network ports, wireless connections, and physical access points. Properly identifying and securing entry points is crucial to prevent unauthorized access and reduce the system's overall attack surface.

3) Attacker Model: It describes the characteristics, capabilities, and potential behaviors of adversaries who may target the system. This model considers factors such as the attacker's motivation, resources, technical expertise, and level of access. A well-defined attacker model helps security analysts anticipate possible attack vectors and design more defensive strategies.

4) Threats and Vulnerabilities: Threats are potential events or actions that could harm a system, such as data breaches, denial of service attacks, or unauthorized privilege escalation. Vulnerabilities are the specific weaknesses or flaws, such as software bugs, misconfigurations, or insecure interfaces, that threats exploit. Understanding the relationship between threats and vulnerabilities is fundamental to assessing system risks and prioritizing mitigation efforts.

5) Mitigation strategies: These involve implementing technical and procedural controls to reduce the risk from the identified threats and vulnerabilities. Examples include encryption, access control mechanisms, multi-factor authentication, intrusion detection systems, and regular software patching. Effective strategies help to minimize the likelihood of successful attacks and the potential impact on system operations.

C. Threat Model Framework

A threat model framework is a structured methodology used to systematically identify, evaluate, and address potential security threats and vulnerabilities within a system [18]. It provides a set of guidelines, processes, and classification schemes that help security analysts and system designers anticipate how an attacker might compromise assets, exploit entry points, and attack the system. The frameworks we used to develop our model are as follows.

1) STRIDE: It is a popular threat modeling framework created by Microsoft [5]. It helps identify six types of security threats: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each category focuses on a different kind of attack or weakness in a system. STRIDE is useful because it gives a simple way to think about how a system can be attacked. It works well during the design phase, helping analysts find threats early.

2) MITRE ATT&CK : It is a large, open-source knowledge base of real-world cyber attacks [6]. It shows the different steps attackers take, from getting access to stealing data. The framework is based on real tactics and techniques used by hackers. Each entry explains how an attack works and what tools may be used. MITRE ATT&CK helps security teams understand attacker behavior and plan stronger defenses.

3) CVE: Common Vulnerabilities and Exposures (CVE) is a list of publicly known security issues [7]. Each CVE is a unique ID linked to a specific weakness in software or hardware. These IDs make it easy for people around the world to talk about the same problem. CVEs are widely used in tools, reports, and threat models to identify risks in systems. They are maintained by the MITRE Corporation [19] with support from the U.S. government.

4) NIST: The National Institute of Standards and Technology (NIST) provides guidelines and standards for cybersecurity [8]. NIST helps organizations build strong and secure systems. Its publications, like the NIST Cybersecurity Framework [20], are widely used in both the government and private sectors. NIST offers best practices on risk management, access control, encryption, and more. These guidelines are helpful during system design, development, and security evaluation.

5) CISA: The Cybersecurity and Infrastructure Security Agency (CISA) is a U.S. government agency that protects critical infrastructure [9]. It publishes regular reports on threats, vulnerabilities, and response strategies. CISA shares alerts, best practices, and risk assessments to help public and private organizations stay secure. Its reports are trusted sources for current cybersecurity trends and threats. CISA also works with other agencies to respond to major cyber incidents.

6) NVD: The National Vulnerability Database (NVD) is a detailed resource that builds on CVEs and is managed by NIST [21]. It adds more information to each CVE, such as severity scores, fix status, and impact details. NVD helps security professionals understand how serious a vulnerability is and how it affects systems. It is often used in automated security tools for scanning and threat detection.

III. Literature Review

In this section, we will discuss some recent works using AI for threat analysis.

Hamza et al. highlighted the growing need for proactive cyber-risk management in a world where threats are constantly evolving [22]. Traditional methods like rule-based and signature-based detection are no longer enough, as they struggle to handle new and advanced attacks. They explored how Generative AI models, such as GANs and VAEs, can be used to simulate realistic and unseen cyberattack scenarios, helping organizations test and improve their defense systems. This shift from reactive to proactive threat modeling allows for better preparation against complex threats such as advanced persistent threats (APTs). They also emphasized the value of combining conventional security methods with AI-driven techniques to build adaptive and resilient cybersecurity systems.

Mohamed et al. proposed a comprehensive framework for cyber threat detection by leveraging AI, natural language processing (NLP), and malware analysis [23]. Their approach aimed to detect threats like the Follina vulnerability [24] by integrating multiple analytical techniques. The study showed that while AI and NLP are effective, they face challenges with unstructured data and high computational demands.

Malatji et al. examined the multifaceted dimensions of AI-driven cyberattacks, offering insights into their implications and mitigation strategies [25]. The study discussed the benefits and pitfalls of AI technologies in cybersecurity, emphasizing the need for robust defense mechanisms. A key limitation was the fast evolution of AI, outpacing current security measures and requiring constant adaptation.

Roshanaei et al. explored strategies for enhancing cybersecurity through AI and machine learning [26]. The study emphasized real-time threat detection and predictive capabilities, highlighting the importance of integrating AI into security operations. However, the authors emphasized the ethical and legal concerns associated with AI deployment, highlighting the potential risks of over-reliance on automated systems without adequate human oversight.

Schmitt focused on AI-based cyber threat detection to protect digital infrastructures [27]. The research evaluated machine learning classifiers for anomaly-based malware and intrusion detection, addressing challenges in deploying AI-enabled cybersecurity solutions. While the study provided valuable insights into integrating AI into security systems, it also recognized the limitations in adapting these models to the dynamic nature of cyber threats and the need for continuous learning mechanisms.

Based on the literature review presented above, it is evident that most existing research focuses on utilizing machine learning models to detect and analyze threats across various systems. To the best of our knowledge, there has been no prior work specifically dedicated to developing an AI agent for automated threat modeling. Therefore, our study is the first to introduce an agentic AI approach tailored for threat modeling, to enhance cybersecurity and ensure public safety.

IV. METHODOLOGY

In this work, we designed and developed an Agentic AI system to facilitate intelligent, context-aware, and security-oriented query handling. Our proposed Agentic AI system comprises four primary layers: the Command-Line Interface (CLI) Layer, the AI Agent Layer, the Knowledge Base/Training Dataset Layer, and the Pretrained Large Language Model (LLM) Layer. The system's workflow is presented in Figure 1.

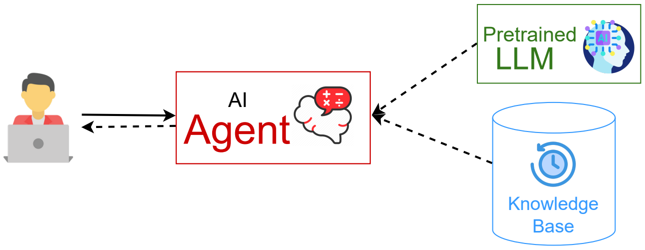


Fig. 1. ThreatGPT System Workflow

The CLI Layer provides a lightweight interface for users to efficiently submit queries and receive outputs.

At the core of the system is the AI Agent Layer, which is responsible for managing the overall workflow. This layer handles the preprocessing of user inputs, determines the appropriate action strategy, formulates structured prompts for the language model, and postprocesses the outputs to ensure coherence, accuracy, and relevance. The agent dynamically decides whether to interact directly with the pretrained language model or to retrieve supplementary information from the knowledge base, depending on the nature of the query and the contextual requirements.

The Knowledge Base and Training Dataset Layer functions as a centralized repository of systematically compiled information, encompassing static datasets, user-specific data, and specialized resources such as threat modeling frameworks, including STRIDE, MITRE ATT&CK, CVEs, NVD, and NIST standards. This layer plays a critical role in enhancing the system's ability to ground the outputs of the large language model (LLM) in authoritative and domain-specific security knowledge, thereby improving the contextual relevance, accuracy, and reliability of the generated threat models.

The Pretrained LLM Layer integrates the Gemini API [28], a high-performance language model that generates natural language responses based on the agent's prompts. The system ensures high-quality, context-sensitive, and factually grounded output by using the Gemini model. The architecture of the system is presented in Figure 2.

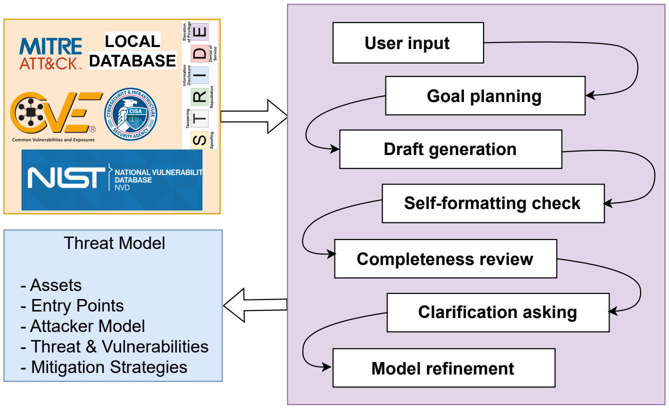


Fig. 2. System Architecture

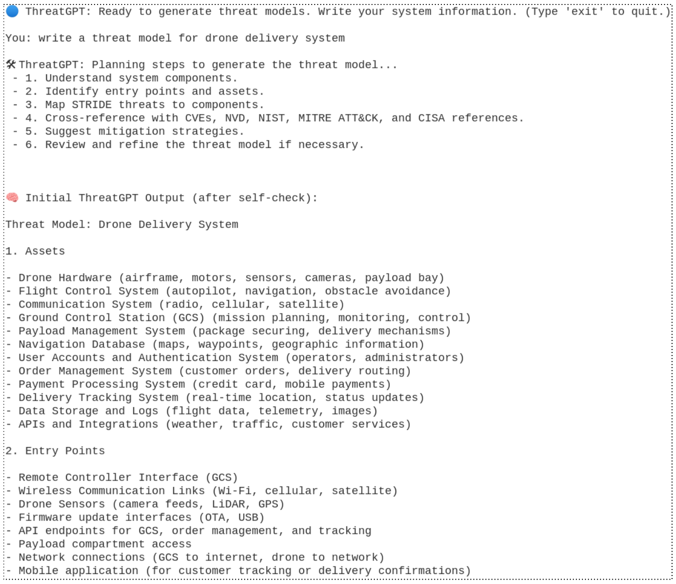
To train the model effectively, we utilized a few-shot learning approach [13] by providing it with more than 50 detailed examples of threat models. These examples incorporate widely recognized cybersecurity frameworks and standards. Additionally, guidelines from respected sources like NIST and CISA were also integrated. This comprehensive training enabled the Gemini language model to better understand cybersecurity terms, threat categories, and real-world scenarios, resulting in improved accuracy and more context-specific, reliable responses.

When a user submits a request to generate a threat model for a specific system, ThreatGPT initiates a systematic and autonomous workflow to generate high-quality threat models. It first recognizes its goal and develops a detailed multistep plan, encompassing asset identification, entry point analysis, threat mapping, vulnerability assessment, and mitigation suggestions. It proceeds to generate an initial threat model draft, ensuring adherence to the learned format by conducting an immediate self-verification and auto-formatting phase if necessary. Following this, ThreatGPT performs an internal completeness review to identify any missing or ambiguous security aspects. If gaps are detected, it autonomously formulates clarification questions to obtain critical information from the user without interrupting the overall workflow. Once clarifications are received, ThreatGPT refines the initial model by integrating the new information, enhancing threat coverage across the frameworks it learned. Finally, it delivers the polished threat model, ensuring that the output is accurate, comprehensive, and professionally structured without requiring extensive human intervention.

V. Experiment and Results

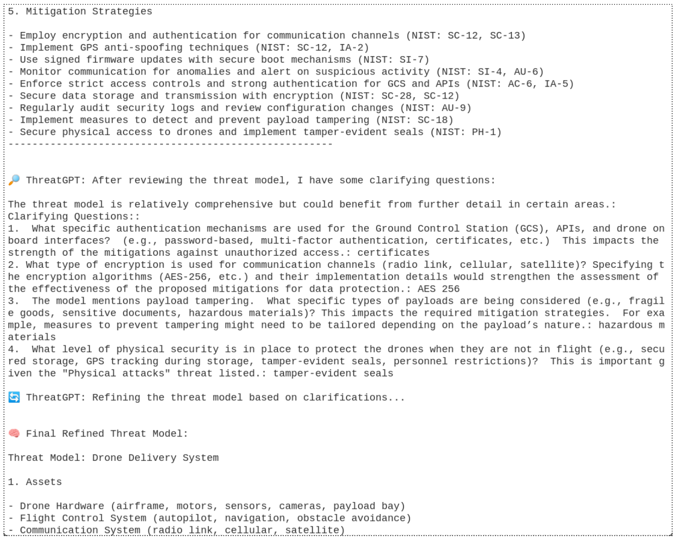
The experiments were conducted using the free version of the Google Gemini API [28], accessed through a local client setup. The system operated on an Ubuntu 20.04 LTS environment and was equipped with a Ryzen 7 5700U CPU with integrated graphics. The machine had 16 GB of RAM and a 1 TB SSD for storage. No external GPU acceleration was used. All client-side preprocessing, prompt construction, and postprocessing operations were performed using the local computational resources. The Gemini model itself remained hosted remotely on Google's servers, with local hardware primarily responsible for handling API interactions and lightweight processing tasks.

The prompt and corresponding response samples generated by the AI model are presented in Figures 3 and 4. The AI model systematically identifies and organizes key components of the threat model, including Assets, Entry Points, Attacker Models, Threats and Vulnerabilities, and Mitigation Strategies for a given system. ThreatGPT first generates an initial threat model. Then it asks for clarification questions if it needs to refine the model. For illustration purposes, only the important portions of the complete response are shown in the figures.



```
🔵 ThreatGPT: Ready to generate threat models. Write your system information. (Type 'exit' to quit.)

You: write a threat model for drone delivery system

🛠ThreatGPT: Planning steps to generate the threat model...
 - 1. Understand system components.
 - 2. Identify entry points and assets.
 - 3. Map STRIDE threats to components.
 - 4. Cross-reference with CVEs, NVD, NIST, MITRE ATT&CK, and CISA references.
 - 5. Suggest mitigation strategies.
 - 6. Review and refine the threat model if necessary.



🧠 Initial ThreatGPT Output (after self-check):

Threat Model: Drone Delivery System

1. Assets

- Drone Hardware (airframe, motors, sensors, cameras, payload bay)
- Flight Control System (autopilot, navigation, obstacle avoidance)
- Communication System (radio, cellular, satellite)
- Ground Control Station (GCS) (mission planning, monitoring, control)
- Payload Management System (package securing, delivery mechanisms)
- Navigation Database (maps, waypoints, geographic information)
- User Accounts and Authentication System (operators, administrators)
- Order Management System (customer orders, delivery routing)
- Payment Processing System (credit card, mobile payments)
- Delivery Tracking System (real-time location, status updates)
- Data Storage and Logs (flight data, telemetry, images)
- APIs and Integrations (weather, traffic, customer services)

2. Entry Points

- Remote Controller Interface (GCS)
- Wireless Communication Links (Wi-Fi, cellular, satellite)
- Drone Sensors (camera feeds, LiDAR, GPS)
- Firmware update interfaces (OTA, USB)
- API endpoints for GCS, order management, and tracking
- Payload compartment access
- Network connections (GCS to internet, drone to network)
- Mobile application (for customer tracking or delivery confirmations)
```

Fig. 3. Initial Threat Model



```
5. Mitigation Strategies

- Employ encryption and authentication for communication channels (NIST: SC-12, SC-13)
- Implement GPS anti-spoofing techniques (NIST: SC-12, IA-2)
- Use signed firmware updates with secure boot mechanisms (NIST: SI-7)
- Monitor communication for anomalies and alert on suspicious activity (NIST: SI-4, AU-6)
- Enforce strict access controls and strong authentication for GCS and APIs (NIST: AC-6, IA-5)
- Secure data storage and transmission with encryption (NIST: SC-28, SC-12)
- Regularly audit security logs and review configuration changes (NIST: AU-9)
- Implement measures to detect and prevent payload tampering (NIST: SC-18)
- Secure physical access to drones and implement tamper-evident seals (NIST: PH-1)
------------------------------------------------------


🔎 ThreatGPT: After reviewing the threat model, I have some clarifying questions:

The threat model is relatively comprehensive but could benefit from further detail in certain areas.:
Clarifying Questions::
1.  What specific authentication mechanisms are used for the Ground Control Station (GCS), APIs, and drone on
board interfaces?  (e.g., password-based, multi-factor authentication, certificates, etc.)  This impacts the
strength of the mitigations against unauthorized access.: certificates
2. What type of encryption is used for communication channels (radio link, cellular, satellite)? Specifying t
he encryption algorithms (AES-256, etc.) and their implementation details would strengthen the assessment of
the effectiveness of the proposed mitigations for data protection.: AES 256
3.  The model mentions payload tampering.  What specific types of payloads are being considered (e.g., fragil
e goods, sensitive documents, hazardous materials)? This impacts the required mitigation strategies.  For exa
mple, measures to prevent tampering might need to be tailored depending on the payload’s nature.: hazardous m
aterials
4.  What level of physical security is in place to protect the drones when they are not in flight (e.g., secu
red storage, GPS tracking during storage, tamper-evident seals, personnel restrictions)?  This is important g
iven the "Physical attacks" threat listed.: tamper-evident seals

🔄 ThreatGPT: Refining the threat model based on clarifications...


🧠 Final Refined Threat Model:

Threat Model: Drone Delivery System

1. Assets

- Drone Hardware (airframe, motors, sensors, cameras, payload bay)
- Flight Control System (autopilot, navigation, obstacle avoidance)
- Communication System (radio link, cellular, satellite)
```

Fig. 4. Clarifying questions for refined threat model

We evaluated our model using three prompts. The simple prompt consists of the system name along with the detailed component names. It makes it easier for the model to understand and create the threat model. It does not need to think about the components. The compound prompt consists of the system name and some component names. The model needs to think about the detailed components and make a threat model for them. Finally, the complex prompt has only the system name. The model thinks about all the components and their interconnection to make the threat model. The sample prompts we used in each category are as follows:

Simple Prompt: Generate a threat model for a Drone Delivery Management System where customer orders are processed through a mobile application connected to a scheduling server. The server assigns delivery tasks to drones, which navigate autonomously using a GPS-based navigation system. Package status is updated in real time via the tracking module, and operational data from drones is continuously monitored through a cloud-based dashboard. Consider the interactions and data flow between these components when identifying assets, entry points, threats, vulnerabilities, attacker models, and mitigation.

Compound Prompt: Generate a threat model for a Drone Delivery Management System. The main components include the drone navigation system, delivery scheduling server, package tracking module, customer mobile application, and cloud-based monitoring dashboard.

Complex Prompt: Generate a threat model for a Drone Delivery Management System.

The time consumption for these three prompts is presented in Figure 5. Based on the figure, it is observed that the response time for each type of prompt remains within the range of 20 to 30 seconds. According to industry experts, the average time required for a human to generate a threat model for a specific system is over 40 hours [29], whereas our proposed model achieves this task within 30 seconds. Therefore, it can be concluded that our approach significantly reduces the time needed for threat model generation, offering a highly efficient alternative to manual methods. This rapid response capability is particularly critical for enhancing public safety, where timely

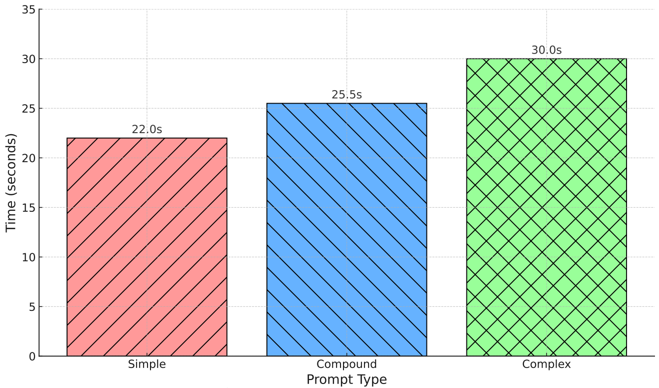


Fig. 5. Response time comparison

identification and mitigation of threats are essential to protecting critical infrastructure and ensuring the security of communities.

VI. Conclusion

In this study, we introduced ThreatGPT, an Agentic AI system designed to automate threat model generation using a few-shot learning approach with the Google Gemini API. By integrating over 50 compiled examples based on frameworks such as STRIDE, MITRE ATT&CK, CVE, NIST, CISA and NVD, ThreatGPT demonstrated strong performance in producing structured, contextually relevant, and accurate threat models. Experimental results show that ThreatGPT significantly reduces modeling time, achieving a generation time of approximately 30 seconds compared to the 40+ hours typically required by human experts. This rapid generation capability has important implications for public safety, where timely identification of risks is very important to protecting infrastructure, maintaining operational continuity, and enhancing community resilience.

Despite its advantages, the system presents certain limitations. Its outputs depend heavily on the quality and scope of the few-shot examples, and it may occasionally miss highly specific or emerging threats not represented in the training data. Furthermore, reliance on external API interactions can introduce latency, and the use of the free Gemini API limits access to advanced fine-tuning capabilities. Future work will focus on expanding the diversity of threat modeling examples, integrating retrieval-augmented generation for real-time access to live vulnerability databases, and enabling interactive refinement of threat models. These enhancements aim to further strengthen the robustness, adaptability, and applicability of ThreatGPT across cybersecurity and public safety domains.

References

[1] B. Friedman, "What is public safety?" *BUL Rev.*, vol. 102, p. 725, 2022.

[2] N. I. Chowdhury, M. A. Hoque, and R. Hasan, "An overview of cyber attacks and defenses on intelligent connected vehicles," *Computer and Information Security Handbook*, pp. 1481–1494, 2025.

[3] B. W. Wirtz and J. C. Weyerer, "Cyberterrorism and cyber attacks in the public sector: how public administration copes with digital threats," *International Journal of Public Administration*, vol. 40, no. 13, pp. 1085–1100, 2017.

[4] A. Shostack, *Threat modeling: Designing for security*. John wiley & sons, 2014.

[5] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Stride-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[6] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the mitre enterprise att&ck matrix," *Software and Systems Modeling*, vol. 21, no. 1, pp. 157–177, 2022.

[7] V. Yosifova, A. Tasheva, and R. Trifonov, "Predicting vulnerability type in common vulnerabilities and exposures (cve) database with machine learning classifiers," in *2021 12th National Conference with International Participation (ELECTRONICA)*. IEEE, 2021, pp. 1–6.

[8] R. Ross, "Managing enterprise security risk with nist standards," *Computer*, vol. 40, no. 8, pp. 88–91, 2007.

[9] I. Czarnowski, "A framework for the clustering and categorization of cisa reports," *Procedia Computer Science*, vol. 207, pp. 4369–4377, 2022.

[10] P. Theurich, J. Witt, and S. Richter, "Practices and challenges of threat modelling in agile environments," *Informatik Spektrum*, vol. 46, no. 4, pp. 220–229, 2023.

[11] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous intelligence for complex goals–a comprehensive survey," *IEEE Access*, 2025.

[12] Y. Shavit, S. Agarwal, M. Brundage, S. Adler, C. O'Keefe, R. Campbell, T. Lee, P. Mishkin, T. Eloundou, A. Hickey *et al.*, "Practices for governing agentic ai systems," *Research Paper, OpenAI*, 2023.

[13] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," *arXiv preprint arXiv:2203.04291*, 2022.

[14] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative ai," *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024.

[15] N. I. Chowdhury and R. Hasan, "Security analysis of cardiovascular implantable electronic device (cied) using a threat model-based approach," in *SoutheastCon 2023*. IEEE, 2023, pp. 81–88.

[16] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, 2012.

[17] S. N. Zisad and R. Hasan, "Towards a security analysis of radiological medical devices using the mitre att&ck framework," in *SoutheastCon 2024*. IEEE, 2024, pp. 1577–1582.

[18] D. J. Bodeau, C. D. McCollum, and D. B. Fox, "Cyber threat modeling: Survey, assessment, and representative framework," *Mitre Corp, Mclean*, pp. 2021–11, 2018.

[19] L. AMBROSE and K. ROGERS, "The mitre corporation," in *Ada: Experiences and Prospects: Proceedings of the Ada-Europe International Conference, Dublin, 1990*. Cambridge University Press, 1990, p. 193.

[20] G. B. White and N. Sjelin, "The nist cybersecurity framework," in *Research anthology on business aspects of cybersecurity*. IGI Global, 2022, pp. 39–55.

[21] H. Booth, D. Rike, and G. A. Witte, "The national vulnerability database (nvd): Overview," 2013.

[22] H. Rauf, S. I. H. Shah, T. Ali, H. Gul, and M. Soomro, "Using generative ai for simulating cyber security attacks and defense mechanisms: A new approach to ai-driven cyber threat modeling," *Spectrum of Engineering Sciences*, vol. 3, no. 3, pp. 361–381, 2025.

[23] N. Mohamed, "A comprehensive framework for cyber threat detection: leveraging ai, nlp, and malware analysis," *International Journal of Information Technology*, pp. 1–8, 2025.

[24] S. Regi, G. Arora, R. Gangadharan, R. Bathla, and N. Pandey, "Case study on detection and prevention methods in zero day attacks," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2022.

[25] M. Malatji and A. Tolah, "Artificial intelligence (ai) cybersecurity dimensions: a comprehensive framework for understanding adversarial and offensive ai," *AI and Ethics*, pp. 1–28, 2024.

[26] M. Roshanaei, M. R. Khan, and N. N. Sylvester, "Enhancing cybersecurity through ai and ml: Strategies, challenges, and future directions," *Journal of Information Security*, vol. 15, no. 3, pp. 320–339, 2024.

[27] M. Schmitt, "Securing the digital world: Protecting smart infrastructures and digital industries with artificial intelligence (ai)-enabled malware and intrusion detection," *Journal of Industrial Information Integration*, vol. 36, p. 100520, 2023.

[28] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[29] S. Compass, "Understanding the developer-centric threat modeling process," https://www.securitycompass.com/whitepapers/understanding-the-developer-centric-threat-modeling-process/, Mar 2025.