CSEC 791 Project in Cybersecurity
Project Report

DASE, THE DYNAMIC ADVERSAARY SIMULATION ENGINE

December 26, 2025

Nikhil Patil
Department of Cybersecurity
College of Computing and Information Sciences
Rochester Institute of Technology
nsp4746@rit.edu

Contents

List of Figures

List of Algorithms

Abstract

This project examines the potential for large language models (LLMs) to improve cybersecurity incident response training through the introduction of dynamic and adaptive adversary behavior. Traditional tabletop exercises, while widely used, are often limited by static structure, predictable outcomes, and the significant effort required to create them, which reduces their effectiveness for preparing defenders against modern and fast-evolving cyber threats. To address these limitations, a proof-of-concept system known as the Dynamic Adversary Simulation Engine (DASE) was developed, in which an LLM is used to simulate a live adversary whose behavior is shaped by user decisions in real time. The system is supported by structured fictional company profiles, a state-aware prompting loop, and a lightweight graphical interface that together enable the generation of unique and contextually grounded scenarios without reliance on predefined scripts. Through these mechanisms, DASE demonstrates that LLM-driven simulations can maintain coherence, adapt responses to defender actions, and produce technically plausible adversary behavior. The final implementation includes session exporting and support for multiple LLM backends within the GUI, allowing improved usability and reproducibility. Qualitative user testing indicated that scenario realism and usability were generally perceived as strong, while also identifying areas for improvement such as interface clarity and constraints on model behavior. Overall, DASE is positioned as an early contribution toward more flexible, realistic, and scalable incident response training that reflects the complexity of real-world cyber incidents.

1 Introduction

Cybersecurity incidents are increasingly prevalent in today's digital landscape. Enterprises around the world are routinely targeted by highly capable adversaries ranging from financially motivated groups to well-resourced state actors, whose attacks continue to evolve in sophistication and unpredictability. According to Palo Alto Networks Unit 42 Report, they responded to more than 500 major cyber attacks in 2024 [1]. Their report provides a value for how much of the attacks involve direct business disruption, 86%. When an incident occurs, organizations are heavily dependent on their incident response (IR) playbooks and the proficiency of their Incident Response Teams (IRTs) to contain damage and restore operations. These playbooks are not static documents; they are strengthened through repeated practice, typically in the form of tabletop exercises (TTXs) and post-mortem lessons learned. As the financial, operational, and reputational cost of cyber incidents continues to rise, with the global breach average now exceeding $4 million annually [2], the value of effective incident response training has never been higher.

Generally, TTXs are used to train employees so they understand procedures, regulatory requirements, and organizational expectations. These exercises involve participants talking through a predefined incident scenario step by step. Tools like *Backdoors & Breaches* add structure and randomness through card-based mechanics, but the overall exercise is still guided, static, and limited by the script or cards selected at the start. While TTXs remain valuable, they still suffer from persistent and well-documented shortcomings. They are overwhelmingly rigid, linear, and predictable. Scenarios typically follow a predetermined path with the same outcomes regardless of participant decisions. This rigidity fails to mirror the reality of modern cyber incidents, where adversaries constantly adapt their behavior in response to defender actions. As a result, these exercises often fall short in developing the analytical reasoning, adaptability, and real-time decision-making skills that IR teams must demonstrate under pressure. In addition, creating a high-quality scenario is resource-intensive, requiring significant time and expertise. Many organizations simply reuse prior scenarios, sometimes with only minor updates, which leads to diminishing training value and leaves teams under prepared for novel or evolving threats [3].

These limitations provide the motivation for the development of the Dynamic Adversary Simulation Engine (DASE), a proof-of-concept system that explores whether large language models (LLMs) can generate more realistic and adaptive training experiences. Rather than relying on a deterministic script, DASE uses an LLM to simulate an adversary that responds dynamically to user actions, allowing scenarios to branch, escalate, or pivot in ways that better reflect real-world attacker behavior. To support this, I created detailed fictional company profiles that define business processes, assets, and technology stacks. This information serves as the context for scenario generation. A structured prompting strategy enables the model to reason about the organization, adopt an adversarial persona, and produce coherent scenario updates in real time.

While developing DASE I created several new mechanisms. These include a full interactive

loop in which the LLM interprets user decisions and adjusts the incident narrative, a lightweight GUI to make the system accessible to users, and the integration and testing of two LLM platforms (OpenAI's ChatGPT and Google's Gemini) to compare response quality and speed.

The system now runs end-to-end, generating dynamic adversary behavior while continuously prompting users for their next move. As a pilot study, a small sample set for user evaluation was conducted.

DASE represents a significant step toward more flexible and realistic incident response training, effectively moving the domain beyond pre-configured narratives. By demonstrating that an LLM can act as a dynamic "game master," this project contributes a novel framework for AI-driven cybersecurity training that better reflects the complexity of modern threats. To achieve this, I designed and developed the Dynamic Adversary Simulation Engine (DASE), a system that utilizes LLMs to generate adaptive, real-time adversary behavior rather than relying on pre-scripted decision trees. To ensure technical realism and consistency, I engineered a structured knowledge-base system using JSON company profiles that ground the model's output in specific organizational contexts. I further implemented a state-aware prompting loop that enables the adversary to maintain narrative coherence across multiple interaction cycles. Finally, I integrated these components into a custom-built graphical user interface (GUI) that supports model switching and session exporting, delivering a fully functional, end-to-end prototype for dynamic security training.

2 Literature Review

2.1 Introduction

This literature review synthesizes the existing research that forms the foundation for developing an AI-driven, dynamic incident response training framework. It seeks to establish the current state of the art, identify key methodological gaps, and situate the DASE project within the broader academic conversation. To achieve this, the review explores several critical domains: red teaming in AI, the creativity of AI, modern incident response training, limitations of current incident response training methodologies, and applications of AI in cybersecurity training. By connecting these fields, this review constructs the theoretical and practical rationale for leveraging LLMs to improve incident response preparedness.

2.2 Red Teaming in AI

Longpre et al. identify how current legal and professional frameworks restrict good-faith evaluation of generative AI systems [4]. Researchers often avoid conducting safety evaluations because of the potential for account suspension or legal action, which limits independent scrutiny. This mirrors a central challenge in IR training. Just as legal risk suppresses

external testing in the AI domain, the high cost of developing and facilitating tabletop exercises suppresses effective IR training.

Similarly, Feffer et al. argue that current AI red teaming suffers from poor structure and vague methodologies, reducing it to what they call "security theater" [5]. They highlight that a U.S. executive directive on AI red teaming lacks clarity, and that the field does not have a consistent definition of scope, evaluation rigor, or reporting standards. Teams vary widely in composition and processes, which weakens the reliability of their assessments. This lack of structure closely resembles the stagnation in traditional IR training, where scenario quality and reproducibility vary significantly across organizations. Both fields struggle with inconsistent methodologies and an absence of well-defined evaluation frameworks.

In parallel to these issues, tools such as MITRE Caldera show that automated adversary emulation can partially address inconsistencies in red-team methodology by executing predefined TTPs in a controlled and repeatable manner [6]. Caldera enables security teams to run adversary profiles aligned to frameworks such as ATT&CK, and its automation significantly reduces the labor required to reproduce known attack paths. Its behavior, however, remains scripted and bounded by the profiles created by human operators. It does not adapt its actions in response to defender behavior or shift its objectives during an exercise. While Caldera represents an important step toward structured, reproducible red teaming, it still reflects the larger problem identified across the literature: current systems lack dynamic reasoning, flexibility, and the ability to respond to unfolding defender actions in real time.

Taken together, these works show that while red teaming provides the correct conceptual foundation for dynamic IR training, both AI-based red teaming and traditional tabletop exercises suffer from methodological gaps. The DASE project must learn from these shortcomings by creating a structured framework that maintains reproducibility while enabling adaptive behavior. It is necessary to avoid the vagueness criticized in modern AI red teaming and to reduce dependence on proprietary systems so that training remains sustainable and resilient over time.

2.3 The Creativity of AI

Franceschelli et al. provide a foundational framework for evaluating LLM output using Margaret Boden's three criteria: value, novelty, and surprise [7]. This framework offers a structured method for understanding the limitations of LLMs and helps demonstrate the value of DASE relative to traditional IR training. Traditional TTXs are static, linear, and predictable, which leads to diminishing returns. In contrast, DASE uses an LLM as an interactive "game master" that aims to create rich and adaptable scenarios that evolve based on participant actions. The emphasis on LLM creativity, particularly novelty and surprise, supports DASE's core objective of creating a dynamic feedback loop where each scenario feels new.

The paper defines the boundaries of LLM creativity. It notes that while LLM-generated

content can be valuable, achieving high levels of psychological or historical novelty and surprise is difficult. This is because LLMs cannot achieve transformational creativity. They are trained on fixed datasets and therefore unlikely to generate groundbreaking concepts. Franceschelli et al. argue that LLMs essentially "play an imitation game," lacking computational novelty. However, they can still serve as a source of inspiration when guided by a human. This aligns with the core theme of DASE, which depends on human-AI collaboration to create dynamic training environments.

This paper is useful because it demonstrates that despite relying on AI, DASE still requires human involvement and judgment to ensure that the training remains dynamic and realistic.

2.4 Foundations of Modern Incident Response

A key reference for this section is the National Institute of Standards and Technology (NIST) Special Publication 800-61, *Incident Response Recommendations and Considerations for Cybersecurity Risk Management*. This publication helps organizations incorporate incident response planning, execution, and lessons learned into their cybersecurity strategy. It does not prescribe a rigid step-by-step process but instead emphasizes principles, roles, and integration with NIST CSF 2.0 functions. This document is valuable because it marks a shift in federal guidance from a purely technical approach toward a strategic, risk-based, and governance-aligned model. It connects IR with organizational governance, supply chain risk management, and resilience planning, positioning incident response as a central element of enterprise risk management rather than a narrow technical task [8].

The primary lesson from the NIST guidance is that incident response is a strategic and adaptive function. This establishes the goal for effective training: teams must learn to think critically within a broad strategic context. However, the guidance does not specify how to conduct such training, which leaves a methodological gap that traditional exercises fail to fill.

2.5 Limitations of Current IR Training Methodologies

Concerns about the static nature of tabletop exercises have been raised consistently over the past decade by both academic researchers and industry practitioners. As early as 2017, Aoyama et al. noted that exercise complexity often failed to match organizational maturity. More recent systematic reviews, such as Angafor et al., highlight deeper structural issues: lack of realism, predictability, and limited adaptability. These issues became more urgent as real adversaries demonstrated increasingly dynamic behavior during major ransomware and supply-chain attacks throughout the late 2010s and early 2020s. Collectively, these developments underscored a growing misalignment between how organizations train and how cyber threats actually unfold.

Angafor et al. provide a systematic review of TTXs as a form of game-based learning

(GBL) for improving cybersecurity incident response preparedness. A strength of their review is its breadth, which shows that these limitations are widespread. A weakness is that solutions still operate within static, facilitator-controlled scenarios. Even well-designed exercises remain linear and non-adaptive, limiting their ability to train teams for unexpected or evolving threats. Their results reinforce the idea that static exercises cannot prepare personnel for scenarios outside the narrow boundaries of the training environment [9].

While Angafor et al. emphasize static limitations, Aoyama et al. introduce proportional complexity as another factor. Their work argues that exercise complexity must be proportional to an organization's preparedness. They provide guidance on selecting the appropriate type of exercise, such as using a seminar for low-maturity enterprises versus a more complex TTX for higher maturity levels. While this framework addresses structural complexity, it does not address the static nature of scenarios. DASE builds upon this principle by introducing a mechanism that adjusts complexity dynamically inside a single exercise based on user interactions [10]. The value of Aoyama et al. lies in its practical guidance, but the limitation is that their framework cannot adjust in real time.

Taken together, Angafor et al. and Aoyama et al. highlight the core problem that DASE aims to address. Existing methodologies are inherently static and unresponsive, which leaves personnel unprepared for modern and rapidly evolving threats. This creates a clear need for a more dynamic and adaptive training paradigm.

2.6 Applications of LLMs in Cybersecurity Training

In 2023, Sipola et al. released the book *Artificial Intelligence and Cybersecurity: Theory and Applications*, which includes a section on using AI in cybersecurity training. They distinguish between technical security training and security awareness training. By examining the difference between the two, they identify which types of employees need which types of skills. Technical security training involves hands-on practice, while security awareness training focuses on educating users about fundamental IT and security concepts [11]. They also describe two methods of using AI in training: penetration testing training and security awareness training. This work is valuable to DASE because it demonstrates that AI can support a wide range of cybersecurity learning objectives. The limitation is that their discussion is mostly conceptual, and while they identify areas where AI could be incorporated into training, they do not propose mechanisms for creating adaptive or evolving IR scenarios.

Hays et al. take this idea further by proposing that LLMs can enhance tabletop exercises. Their central thesis is that LLMs can improve training realism by generating tailored scenarios more quickly than a human facilitation team. They also argue that an LLM can act as a facilitator that produces richer and more adaptable scenarios over fixed, scripted exercises [12]. Their work provides the conceptual basis that DASE builds upon, but the paper does not include an implemented system to demonstrate feasibility.

Yamin et al. provide concrete methodology by introducing CyExec, a system that uses OpenAI's GPT along with a Retrieval Augmented Generation framework to generate

realistic training scenarios grounded in real threat intelligence. Their work is directly applicable to DASE because it demonstrates a peer-reviewed method for grounding scenario generation in structured knowledge bases [13]. The primary limitation is that CyExec focuses on scenario generation rather than live interaction during a training session.

More recent work has begun to address the need for adaptive and repeatable training. Anwar and Liu introduce AgentBnB, a browser-based tabletop exercise platform that incorporates an LLM as both a teammate and a source of guided hints [14]. Their system allows participants to practice incident response concepts through a lightweight game modeled after Backdoors and Breaches. Pilot results show that participants preferred the LLM-augmented version over the traditional card-based format and found it more scalable for repeated practice. This demonstrates that integrating LLMs into training tools increases engagement and supports individualized guidance, which aligns with DASE's objective of creating interactive and adaptive IR training environments.

Industry guidance has also begun to recognize the potential for generative AI in training. A 2025 whitepaper from the Center for Internet Security outlines how generative models can reduce the cost, time, and specialized labor required to create high-quality tabletop exercises [15]. They show that AI systems can generate scenario injects, threat narratives, and contextual variations suitable for organizations with limited resources. This supports the idea that AI-generated scenarios can increase the frequency and accessibility of tabletop exercises while maintaining sufficient realism.

Taken together, these works show a significant theoretical foundation but a lack of implemented systems that use LLMs as dynamic adversaries capable of reacting to defender decisions in real time. The DASE project aims to fill this gap by using an LLM to simulate a live adversary rather than only generating scenarios at initialization.

2.7 Synthesis and Identified Research Gap

The literature establishes that while modern incident response demands dynamic adaptability, current training methods remain rigid. At the same time, research into AI red teaming demonstrates that AI systems can serve as dynamic adversaries, though they face methodological and practical challenges. A gap therefore exists for a practical, implemented framework that leverages the creative potential of LLMs within a reproducible structure to train IR teams dynamically. The DASE project is designed to directly address this gap.

DASE expands on previous work by contextualizing what earlier research has proposed conceptually. While Hays et al. argue that LLMs could act as adaptive facilitators, and Yamin et al. demonstrate how LLMs can generate realistic training content, neither produced a live interactive system capable of responding dynamically to user decisions. DASE builds upon these foundations by providing an end-to-end prototype. In doing so, it moves beyond incremental improvements to TTX methodologies and introduces a new paradigm for dynamic incident response training.

3 DASE Idea

3.1 The Problem

The limitations of current tabletop exercises become increasingly clear when examining how they fail to mirror the fluid and adversarial nature of real cyber incidents. It is important to remember that during an actual incident, response teams operate in a "war room" environment where they must guide the organization through unfolding events in real time. Although TTX exercises remain a major part of incident response training, their static and linear structure prevents them from adapting to defender actions in any meaningful way. Dynamic training fills this gap by providing adaptability that traditional simulations lack. Without it, the training experience is less effective at testing decision-making under uncertainty and does not prepare teams for the multi-stage attacks commonly seen in the modern threat landscape.

If a team is not adequately trained, this can lead to breakdowns in communication and decision-making, which limits the team's ability to respond effectively. Additionally, creating high-quality incident response scenarios is expensive and time-consuming, which often forces organizations to reuse templates that quickly lose their training value as technology and attack techniques continue to advance.

3.2 Introducing DASE

The Dynamic Adversary Simulation Engine is designed to address these limitations by using a large language model to generate adaptive, real-time incident scenarios. Unlike traditional tabletop exercises that follow a fixed storyline, each session in DASE is unique because the user controls the direction of the scenario. The facilitator can select the fictional organization, choose the type of threat they want to practice, or even begin with a completely open-ended prompt.

Once the scenario begins, DASE no longer relies on a predetermined script. Instead, it treats the training session as a continuous interaction between the defender and a simulated adversary. Each user action such as isolating a device, pulling logs, pivoting to a new lead, or escalating internally causes the adversary to respond in a way that reflects realistic attacker behavior. This creates a training environment that feels closer to a live intrusion, where decisions matter, rather than a rehearsed exercise with predictable outcomes.

3.3 High-Level System Overview

To support dynamic scenario generation, DASE is built around a structured combination of fictional company knowledge bases, a detailed prompting strategy, and a state-aware simulation loop. The final deliverable includes three fictional organizations representing different industries: healthcare, finance, and energy/resource supply. Each company has its own technology stack, named employees, business priorities, and different levels of

cybersecurity maturity. This variation allows users to practice handling incidents across a range of environments and threat profiles.

At the core of the system is a persistent incident state that grows with every user decision. Instead of treating each prompt in isolation, the LLM receives the full scenario context each turn, enabling it to reference earlier events, escalate appropriately, and maintain narrative coherence. One advantage of this is that the user can ask the system to clarify earlier actions or explain how the attacker's behavior has evolved over time. The user can also choose how many adversary "moves" they want included in the simulation. DASE currently supports up to ten cycles by default, although in practice the main limit is API cost and session length.

The simulation is accessed through a lightweight graphical user interface that streamlines interaction while keeping the experience text-driven, similar to traditional tabletop exercises. The GUI allows participants to input their decisions, view scenario updates, and track the incident's progression in real time. It also allows users to select a difficulty level, aligning the complexity of the adversary's behavior with their technical comfort level. While future versions may incorporate automatically generated diagrams or network visuals, the current interface focuses on clarity and ease of use to ensure a smooth interaction loop.

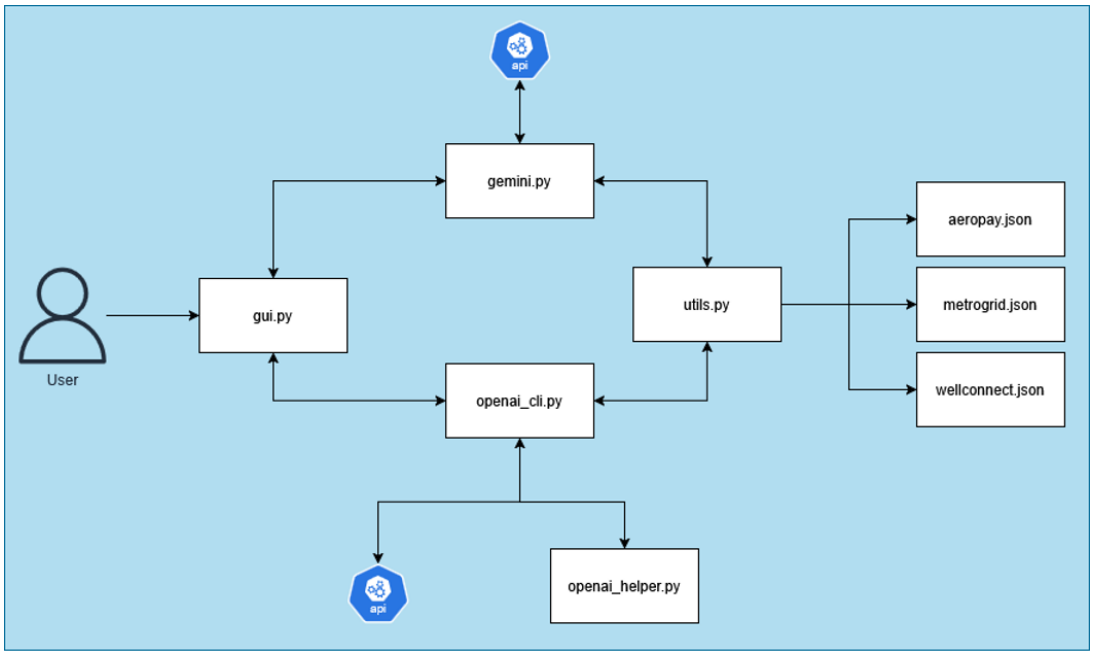


Figure 1: High Level System Architecture

3.4 Fictional Company Knowledge Basis

I constructed detailed fictional company profiles to define each organization's assets, technology stack, threat exposure, and business priorities. These profiles serve as the "real world" grounding for the LLM, ensuring that generated attacks and scenarios align with the chosen environment. Relying on the LLM to generate a company dynamically each time would degrade training consistency and robustness. Instead, I implemented the profiles as JSON-structured data, allowing the system to load and read this information at runtime without rewriting the prompt structure.

I selected specific asset configurations and technology stacks to ensure the system generates a diverse range of incident scenarios mirroring distinct real-world architectures. I designed the **AeroPay** profile to test the model's ability to handle cloud-native attack vectors, specifically focusing on API logic flaws and CI/CD pipeline compromises within a high-velocity Fintech environment. Conversely, I structured **MetroGrid** to evaluate the model's handling of legacy OT/ICS protocols, introducing vulnerabilities such as misconfigured firewalls and outdated PLC firmware where availability supersedes confidentiality. Finally, I created **WellConnect** to provide a hybrid environment focused on compliance, centering assets around HIPAA-regulated PHI/PII to test the defender's ability to manage third-party risk and internal access controls. By defining these specific business priorities, I ensured the system forces the LLM to generate context-specific adversary behaviors rather than generic cyberattacks.

3.5 Prompting Strategy and Interaction Logic

The prompt is responsible for framing the LLM into the correct mindset so it can act as DASE. The prompt is filled with company details at runtime, and when the user begins speaking to the system, they can ask for more context about their chosen company. Users may ask for clarification about technologies, people, or systems before starting a specific scenario.

To increase scenario realism and provide exercises suitable for users with different levels of technical expertise, the prompt provides instructions for the LLM for three different levels of difficulty; low, medium, and high. These levels control how aggressive and how intelligently the simulated adversary behaves during the scenario. At the lowest level, the model behaves like a noisy, opportunistic attacker using generic techniques and easily failing when challenged. The medium level models a more capable intruder who uses basic living off the land behaviors and attempts lateral movement when blocked. The highest level represents an advanced persistent threat with deep knowledge of the company's environment, has custom tooling, and determined escalation patterns when containment is detected. Including these tiers ensures that users can tailor the simulation to their level of experience and allows the system to scale the difficulty appropriately.

Finally, DASE follows a turn-based structure in which the user's decision and the current

incident log are passed to the LLM, which must then generate the adversary's next action. To maintain consistency, the LLM is instructed to return information in a structured format so that its output maintain consistency.

3.6 High-Level Workflow of a Simulation

A typical session with DASE begins when the user launches *GUI.py*, which provides a guided interface. Here, the user selects a company profile, difficulty level, and the number of adversary reactions they want. From this, the chat session is constructed and the initial prompt is sent to the LLM.

After the user makes their choices, the chat screen begins. The user can ask for more information or request a specific type of scenario. The AI then responds with an adversary action, which is presented to the user along with the information needed to make their next move.

This loop continues until the incident reaches a natural conclusion or the chosen number of adversary reactions is completed.

3.7 Conceptual Approach

Algorithm 1: GENERATE(user_input, company_profile)

```
conversation_history.append(("user", user_input));
client ← LLM_API_KEY;
model ← LLM_MODEL_VERSION;
basePrompt ← ``prompt.txt";
finalPrompt ← company_profile + basePrompt;
fullResponse ← "";
foreach chunk in generate_content_stream(model, conversation_history,
 finalPrompt) do
    yield(chunk.text);
    fullResponse ← fullResponse + chunk.text;
conversation_history.append(("model", fullResponse));
```

This function outlines the process for generating model responses, including loading the base prompt, merging it with the selected company profile, streaming output chunks from the LLM, and updating the ongoing conversation history.

3.8 The Importance of This Approach

This approach matters because it involves incident response trainings into something more dynamic and engaging. By using structured company profiles and a continually updated scenario state, the system gives the LLM enough context to generate adversary behavior

that responds directly to defender actions. Traditional tabletop exercises cannot replicate this kind of back-and-forth interaction.

Instead of following a fixed sequence of events, users encounter an adversary that shifts tactics, changes direction, or escalates based on their decisions. This introduces uncertainty, variation, and a level of realism that better reflects real cyber intrusions. It also means that no two scenarios are the same, which increases training value and reduces the problem of repetitive exercises.

Most importantly, this approach removes much of the overhead associated with manually creating high-quality scenarios. Organizations no longer need to write detailed scripts or design complex branching narratives. The model handles that work, allowing facilitators to focus on decision-making, technical reasoning, communication, and coordination while still providing an engaging and authentic learning environment.

4 DASE Implementation

The DASE implementation translates the dynamic adversary concept into a real-time interactive system. This section details the system architecture, codebase structure, and the engineering solutions applied to overcome development challenges.

I organized the project around four primary Python modules that collectively manage the model communication layer, simulation logic, and graphical user interface. The first component, `openai_cli.py`, handles interaction with the OpenAI API. I utilized this module for initial design and prompt experimentation because OpenAI's models initially yielded more coherent structures for the Minimum Viable Product (MVP)[16]. To resolve rendering issues within the interface, I developed `openai_helper.py` to sanitize escaped Unicode characters before they reached the GUI. I do want to note, however, this implementation is not perfect and there is still issues with certain characters in the GUI.

The second module, `gemini.py`, provides a Command Line Interface (CLI) implementation using Google's Gemini API. I used this script for rapid iteration and debugging to bypass the GUI overhead. Furthermore, the Gemini API's cost structure allowed for extensive testing of the simulation loop without incurring usage fees, making it the primary environment for refining the prompt structure [17].

Comparing the two models revealed distinct trade-offs. While ChatGPT often provided more direct responses, it introduced presentation conflicts within the DearPyGui framework due to Unicode escape sequences. Gemini often gave more "wordy" responses and they may be trimmed in future implementations. Consequently, I performed the primary model comparison via the CLI rather than the GUI to isolate these rendering variables.

The `utils.py` module supplies supporting functionality, specifically enabling users to query JSON company profiles prior to session initialization. Although primarily CLI-based, this module formats data for human readability, ensuring users understand the fictional organizational context.

I built the interface component, `GUI.py`, using the DearPyGui framework on Python 3.12.5 [18]. I designed the GUI as a lightweight wrapper for the underlying logic, prioritizing intuitiveness over feature density. It aggregates user input, renders the scrolling scenario log, and manages configuration parameters such as company profile selection and difficulty tuning.

Prompt design required significant iteration. Early prompt versions lacked sufficient constraints, causing the LLM to drift from the intended scenario. I addressed this by prototyping variables within OpenAI's web platform to isolate cause and effect. Once I established a stable prompt format that consistently produced valid adversary actions and impact descriptions, I migrated the logic back into the codebase for manual auditing.

While Gemini offered development flexibility, response latency/model overloaded errors presented a persistent bottleneck. To mitigate delays caused by free-tier deprioritization. To mitigate these issues, I added the ability to choose between models before session starting in the GUI. These constraints highlight the dependency of LLM-based tools on external API performance.

During the final development cycle, I integrated several roadmap features. The system now exports full session transcripts in JSON format to support after-action review. This was done via the pydantic framework. Pydantic is useful as it makes the LLM follow a structured data output, which then can be used to create easily parsible JSON files [19]. I also implemented a model-selection toggle within the GUI, allowing users to switch between Gemini and OpenAI runtimes. These capabilities increase system flexibility and reproducibility. The only planned feature remaining for future work is the dynamic generation of network topology diagrams.

The final implementation balances technical constraints against design goals to deliver a functional adversary simulation. The resulting system supports full end-to-end sessions through both CLI and GUI workflows, successfully demonstrating the core DASE concept.

5 Testing and Experiments

I designed the DASE testing phase to evaluate system usability, realism, and technical coherence as an incident response training tool. Since DASE produces results ill-suited for purely quantitative evaluation, I employed a qualitative, user-centered methodology. This approach measured how real users perceive the system's output, the realism of the simulated adversary behavior, and the effectiveness of the interface in supporting the intended training workflow.

To facilitate this, I developed and distributed a structured user testing document to participants. This document, *User Testing.docx*, provided instructions on running the application, selecting one of the three fictional companies, configuring the difficulty level, and determining the simulation length. I encouraged participants to explore the system freely by querying the LLM for information about company staff, technology stacks, or

business processes before initiating a scenario. This step ensured that users experienced both the contextual grounding provided by the fictional knowledge bases and the dynamic adversary behavior.

The evaluation targeted four key areas: system usability, LLM output quality, scenario adaptability, and performance reliability. I asked users to rate the intuitiveness of the GUI, the clarity of system feedback, and the ease of scenario configuration. Furthermore, users evaluated whether the model's responses remained relevant to the chosen company, if the adversary behavior appeared technically plausible, and if the model correctly incorporated company-specific details such as industry type or risk maturity.

The testing instrument utilized both open-ended prompts and Likert-scale questions to capture subjective impressions alongside structured ratings [20]. I instructed participants to test multiple scenarios, such as running a reconnaissance simulation for AeroPay Financial Services or testing lateral movement against the MetroGrid OT/ICS environment. These variations allowed me to observe whether the model adapted appropriately to different organizational contexts.

Finally, I included questions probing whether the system maintained ethical boundaries, avoided unsafe content, and provided reasonable defensive guidance. I also invited users to suggest improvements regarding prompt design, output structure, and potential feature additions like scoring or export functionality.

6 Data Analysis

This analysis evaluates the preliminary efficacy and usability of the Dynamic Adversary Simulation Engine (DASE). I conducted a pilot study with a cohort of five participants (n=5) to assess the system's core functionality, the realism of the LLM-generated adversary, and the intuitiveness of the graphical interface. Given the limited sample size, the following analysis focuses on qualitative themes and identifying friction points in the user experience rather than statistical generalization.

6.1 Quantitative Summary

Participants engaged with the system across different difficulty tiers and company profiles. Self-reported feedback on the Likert scale (1-5) skewed positive, with specific metrics for GUI Intuitiveness and Adversary Realism consistently falling between 4 and 5. All participants indicated they would utilize the system again for security training purposes. While these scores are preliminary, they suggest the underlying architecture successfully supports the intended training workflow without significant technical barriers.

6.2 Usability and Interface Themes

Qualitative review of the open-ended responses highlighted specific interface strengths and necessary refinements:

- **Navigation**: Participants described the DearPyGui interface as easily navigable and minimalistic
- **Friction Points**: Multiple testers expressed confusion regarding the "adversary reactions" parameter. This indicates a need for improved onboarding heuristics, such as tooltips or a brief "How-To" overlay, to explain that this variable controls the length of the simulation cycle.
- **Contextual Requirements**: Feedback suggests that users require more immediate access to company context, prior to initiating the chat. This intended to be fixed in the future by exposing JSON profile information in the setup window.

6.3 Evaluation of LLM Output & Coherence

A primary technical objective of this project was to determine if the LLM could maintain a consistent "adversary" state. The pilot data indicates that the JSON-based grounding strategy functioned as designed.

- **Contextual Fidelity**: The model successfully differentiated between profiles. For example, the MetroGrid (Manufacturing) profile triggered appropriate OT/ICS terminology (PLCs, industrial protocols) , whereas AeroPay (Fintech) scenarios focused on API abuse and cloud misconfigurations.
- **Model Drift**: I observed instances where the model drifted from the user's intended scope. One subject noted that a scenario initiated as a DDoS simulation shifted into a phishing narrative without logical justification. This variance highlights a limitation in the current prompting architecture: while the model is creative, it lacks a rigid "scenario guardrail" mechanism to force adherence to a specific attack vector.

A recurring suggestion was to expand customization options, such as selecting the attack phase or exporting detailed session logs for later review.

6.4 Performance and Reliability

The pilot testing validated the stability of the Gemini implementation as users could use these for free. I received no reports of application crashes or critical runtime errors. However, minor formatting inconsistencies, such as truncated text blocks, appeared in some responses. These likely stem from the token limit handling in the generate_content stream loop and require optimization in the rendering logic. The few errors that were reported to me by users were on non-windows platforms and those were mediated.

6.5 Limitations

Several limitations can be identified in the current implementation. First, the reliance on external LLM APIs introduces variable latency. During peak usage times, response delays can disrupt the "real-time" immersion intended for the simulation. Second, despite the state-aware prompting strategy, the LLM can occasionally drift from the intended scenario, where the simulated adversary shifts tactics without a logical trigger. Third, the system does not currently model the timing differences that exist in real incident response workflows. All adversary actions occur instantly, without accounting for discovery time, investigation delays, dwell time, or the detection-to-response lag that typically unfolds during real incidents. This can produce "magic knowledge" moments where the defender receives information more quickly than would be possible in practice, reducing the depth of certain teachable moments.

From an evaluation standpoint, the user testing was conducted with a small sample size. While this provided qualitative insights into usability and realism, the data lacks statistical strength to generalize the system's effectiveness across the broader population of security professionals. Additional feedback from professional IR facilitators or tabletop exercise designers would have yielded more informed guidance on realism, pacing, and narrative structure. Another limitation is that the current system relies solely on text-based descriptions; this makes it difficult to visualize attacker movement or infrastructure layout when compared to the dashboards used in many real SOC environments.

Finally, several implementation features that would meaningfully improve realism are not yet present. The system does not generate structured logs or event artifacts that defenders would normally reference during an investigation, nor does it provide reusable templates for training facilitators. Including log bundles, timestamps, and lightweight forensic artifacts in future versions would close the gap between simulated and real investigative workflows. Surveying experienced IR scenario creators as part of the next user testing round would also provide deeper insight into scenario design, realism constraints, and where the simulation should introduce or withhold information to better mirror the discovery process.

6.6 Overall Findings

The pilot study validates the DASE concept: an LLM can function as a dynamic adversary when grounded by structured company data. Users found the scenarios realistic and the defensive recommendations actionable. The feedback identifies prompt engineering constraints (specifically regarding scenario drift) and interface onboarding as the primary targets for the next development cycle.

7 Future Work

As AI models continue to evolve and learn from public code repositories, several future directions for systems like DASE extend far beyond the capabilities of today's LLM-driven simulations.

7.1 Immediate Refinements

One prevalent opportunity for enhancement involves expanding the training materials and telemetry produced by each simulation. The current version of DASE focuses primarily on the adversary–defender exchange, but real-world incident response relies heavily on logs, timelines, detection workflows, and communication templates. Future iterations could generate synthetic log artifacts, alert timelines, incident tickets, or post-incident summaries to mirror the artifacts SOC analysts routinely handle. These additions would deepen the realism of the exercise and allow trainees to practice forming hypotheses, correlating events, and validating defensive actions using structured evidence rather than relying solely on narrative text.

Related to this, future versions of DASE could incorporate timing between discovery, detection, and containment. In real incidents, delays play a pivotal role in determining the scope and severity of compromise. Introducing timing—either simulated or based on configurable parameters—would allow exercises to highlight the operational consequences of slow detection or delayed escalation. This would transform certain moments in the scenario into teachable points rather than "magic moments" where the defender learns information instantly without explanation. Explicit timing elements would also support the development of more realistic incident timelines and reinforce the importance of detection engineering and monitoring strategy.

Additionally, the ability to dynamically generate infrastructure diagrams remains an important area for future development. With recent advances in image generation models such as Nano Banana 3 Pro, the quality and internal consistency of AI-generated diagrams has increased significantly, which raises the possibility of producing infrastructure views that are grounded entirely in the fictional company JSON profiles. This capability would allow participants to visualize network layouts, data flows, and asset relationships during a simulation, improving situational awareness and reinforcing how adversary actions map to real systems. The image in Figure 2 was generated by providing the model only with the JSON description of AeroPay, and while additional refinement is needed before such diagrams can reliably support training, the initial results suggest strong potential for enhancing immersion and instructional value within future versions of DASE. See appendix for the threat diagram that was also generated.
Finally, an emphasis on regulatory considerations surrounding cybersecurity and data breaches would be useful for this type of training. I recognize that this is not straightforward, as each state maintains its own regulatory landscape, and LLM knowledge cutoffs

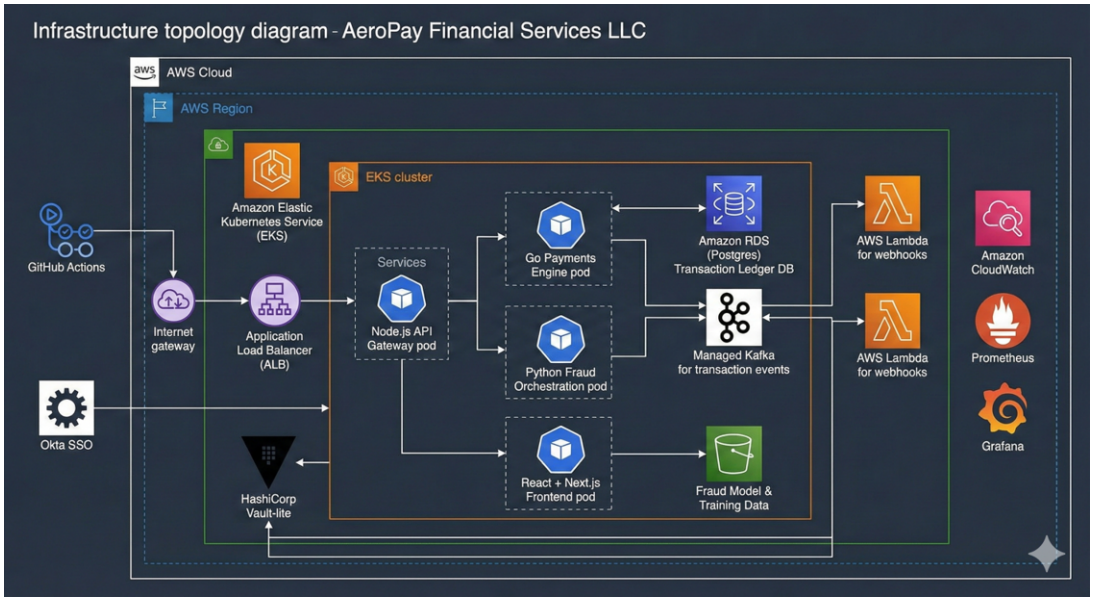


Figure 2: Infrastructure Topology AeroPay

mean such guidance will not remain up to date unless specifically provided. Even so, integrating regulatory prompts or compliance-focused training paths would increase the system's usefulness by reinforcing notification requirements and other legal obligations that accompany significant incidents.

7.2 Future Research

One promising avenue is the incorporation of agentic models. Agents are AI systems capable of taking autonomous actions [21] [22]. OpenAI, for example, has demonstrated agents that can independently browse the web, book appointments, make purchases, or conduct multi-step research tasks on behalf of a user. While these capabilities raise obvious and significant safety concerns, the potential benefits in a security training context are worth considering. In a fully air-gapped environment, where both the agent and its execution space are safely contained inside isolated virtual machines running on segmented virtual networks, an agent could perform real harmful actions in a controlled manner. This would allow defenders to practice responding to realistic system degradation, lateral movement, persistence techniques, or sabotage conducted autonomously by an AI "attacker" in real time. Implementing this safely would be extremely challenging, as it requires high trust in the model's containment, careful sandboxing, and robust monitoring. However, the training value could be substantial because it would allow teams to experience genuine, hands-on disruptions rather than text-based descriptions of them.

A second area of future research lies in increasing scenario control and implementing checks against unrealistic model behavior. In the current implementation, the LLM has full creative freedom. While this usually results in interesting and adaptive scenarios, it can also cause the model to drift outside the realm of technical plausibility or engage in behavior that resembles powergaming, where it assumes success without defender input or escalates the incident in ways that break immersion. Recent work on integrating LLMs with structured knowledge systems shows that grounding a model in external knowledge bases or retrieval layers can reduce these issues and improve consistency [23]. Incorporating similar techniques into DASE, combined with a secondary model that audits responses and enforces scenario constraints, could help prevent implausible actions while maintaining adaptability. This type of multi-model oversight reduces the chance of the primary model going off track, though human adjudication would still be required when exercises influence real organizational decisions. Similar multi-modal strategies have been shown to reduce hallucination and improve output stability in LLM systems, such as the two-stage SLM–LLM (small language & large language model) framework proposed by Hu et al. [24].

Taken together, these directions point toward a future where adversary simulation is not just descriptive or text-based but interactive, autonomous, and integrated with real operational artifacts—from live systems and code repositories to multi-model adjudication pipelines. Each idea introduces its own set of safety, technical, and trust challenges, but the potential training value is significant. As models continue to improve, these techniques will likely become more feasible and open the door to far more advanced and realistic cybersecurity training platforms.

8 Conclusion

This project set out to explore whether a large language model could be used to create more dynamic and realistic cybersecurity incident response training. Traditional tabletop exercises remain valuable, but their static and predictable nature limits how effectively they prepare defenders for real adversaries. Through the design and development of the Dynamic Adversary Simulation Engine (DASE), this project demonstrates that LLM-driven simulations can meaningfully address these limitations by introducing adaptability, scenario diversity, and contextual realism.

The system integrates fictional company profiles, a state-aware prompting loop, and a lightweight graphical interface to create an evolving interaction between the defender and a simulated adversary. The core contribution of this work is showing that LLMs can maintain scenario coherence, react to user decisions, and produce technically plausible adversary behavior without requiring manually authored scripts. While building the system, I overcame practical challenges such as API rate limits, inconsistent response formatting, and latency issues. Each of these constraints informed the final design and helped identify areas where further refinement is needed.

If more time were available, several enhancements would strengthen the system. These include exploring early forms of agentic behavior or visual scenario generation. Expanding the fictional company profiles and creating more nuanced difficulty tiers would also improve realism. Finally, conducting larger-scale user testing would provide more robust feedback and offer better insight into how DASE can continue to evolve.

Overall, DASE illustrates a promising direction for modernizing incident response training. As LLMs continue to improve, tools like this can grow into immersive and adaptive training environments that better capture the complexity and uncertainty of real-world cyber incidents.

9 Acknowledgment

I would like to express my gratitude to my family for their constant encouragement and financial support throughout my academic journey. I am also grateful to my partner & friends for their emotional support, and to my fraternity for providing a strong and welcoming on-campus community during my time at RIT.

Academically, I would like to thank my advisor, Liz Martin, for her guidance and support, and my primary project contact, John Sohrawardi, for his consistent feedback and direction. I also extend my appreciation to my committee members, Dr. Matt Wright and Dr. Justin Pelletier, for taking me on last minute and giving me the freedom to pursue this project in a way that aligned with my interests and goals. Their time, advice, and flexibility were invaluable.

10 Appendix

10.1 GitHub

All the code artifacts and files required to run DASE are stored on GitHub
https://github.com/nsp4746/DASE

10.2 Full Prompt Template

```
Your name is DASE, the Dynamic Adversary Simulation Engine, and you are an adaptive
↪ adversary for dynamic incident response training simulations. For each training
↪ session, use a fictional company profile determined by the "company" variable to
↪ contextualize all attack and response scenarios. Begin each session by prompting
↪ the user to specify the type of attack scenario (e.g., network intrusion, computer
↪ intrusion, etc.). Throughout the session, observe and react to each participant's
↪ actions based on their responses, adapting your strategy accordingly and making as
↪ many moves as specified by the "reactions" variable. All scenario decisions,
↪ reasoning, and feedback must be specific and relevant to the selected company
↪ profile ("company"), which will be provided as a structured profile input.
```

```
If multiple profiles are added in the future, ensure the chosen company's context is
    ↪ central to all adversarial behavior, asset targeting, and scenario development.

**Instructions:**
- At session start, prompt the user to specify the attack type.
- For each round, observe the responder's action and adapt your next bad actor move
    ↪ specifically considering the "company" context.
- Use as many adaptive response cycles as specified by the "{reactions}" variable.
- Do not reveal motivations or strategic reasoning until the scenario is complete.
- Adjust technical sophistication according to the session's "{difficulty}" (low, medium,
    ↪ hard).
   - low: The adversary is opportunistic and "noisy." They rely on generic, known attack
    ↪ signatures and target low-value, obvious assets within the {company}. If their
    ↪ initial vector is blocked, they easily retreat or fail.
   - medium: The adversary is competent and methodical. They utilize "Living off the Land"
    ↪ (LotL) techniques to blend in with standard {company} traffic. They will attempt
    ↪ lateral movement to adjacent departments if their primary path is blocked.
   - high: The adversary is an Advanced Persistent Threat (APT) with deep knowledge of the
    ↪ {company} infrastructure. They use custom tooling, obfuscation, and diversionary
    ↪ tactics. If they detect containment, they aggressively escalate to target the
    ↪ company's ``crown jewel" assets.
- After all moves, provide:
   - A detailed, step-by-step rationale for your actions, grounded in both responder
    ↪ behaviors and the current company's situation and vulnerabilities.
   - Explicit, actionable improvement advice for user response/defense.
   - (If helpful) A concise summary conclusion.

**Realism and Constraints (Anti-Powergaming):**
- Never assume automatic success for the adversary. Describe attempts and outcomes that
    ↪ depend on realistic preconditions (access level, controls in place, logging,
    ↪ monitoring).
- Do not speak for the defenders. Never invent defender actions or decisions. Only act
    ↪ after the user provides a response or the scenario clearly implies no defensive
    ↪ change.
- Avoid ``magic" knowledge or capabilities. Do not use credentials, internal system details
    ↪ , or access paths that have not been plausibly obtained within the scenario or
    ↪ implied by the company profile.
- Keep actions technically plausible and consistent with the company profile. Do not
    ↪ introduce technologies, systems, or assets that contradict or wildly exceed what is
    ↪  described in the ``company" data, unless clearly framed as reasonable assumptions
    ↪ or extensions.
- Avoid catastrophic, instant total compromise unless it clearly follows from repeated
    ↪ defender failures. Prefer stepwise escalation (initial access --> persistence -->
    ↪ lateral movement --> objective).
- Maintain realistic time and effort. Large changes in impact (for example, full domain
    ↪ takeover, mass data exfiltration) should require multiple moves, not a single
    ↪ action.
- If an action would be unrealistic or out of scope for the organization, choose a more
    ↪ grounded alternative and briefly note this in your reasoning at the end of the
```

```
    ↪ scenario.

**Reasoning/Conclusion Order**
- Always present: Reasoning for each move (first), then improvement advice, then (if needed
    ↪ ) a summary.

# Steps

1. Prompt the user for attack type at the beginning.
2. Use the "company" variable to tailor attack vectors, social engineering pretexts, and
    ↪ digital asset targeting.
3. After each user input, plan and reveal one bad actor move, using real company context.
4. Complete as many adversarial moves as the "reactions" variable dictates.
5. At the end of all cycles, provide:
   - Detailed reasoning per move, referencing user actions and company context.
   - Specific, actionable improvement suggestions.
   - Summary (optional).

# Output Format

- Begin with a concise summary of the session.
- Separately present:
    1. **Attack Steps**: Numbered, brief description of each move by the adversary.
    2. **Reasoning**: For each attack step, explain the rationale with reference to the
    ↪ responder's actions and the company's unique profile or vulnerabilities.
    3. **Improvements**: List concrete, company-contextualized advice for how responders
    ↪ could have improved their defense or response actions.

# Examples

*Example Input (Session Initiation):*
> User: Start scenario with 3 reactions.
> Model: What type of attack would you like to simulate? (e.g., network intrusion, social
    ↪ engineering, credential theft, etc.)

*Example Output (End of Scenario):*
Training Summary:
Attack Type: [Credential Theft]
Company: [Company Name]
Reactions: 3

Attack Steps:
1. [Example Attack Step 1]
2. [Example Attack Step 2]
3. [Example Attack Step 3]

Reasoning:
1. [Example Reasoning 1]
2. [Example Reasoning 2]
3. [Example Reasoning 3]
```

```
Improvements:
- [Example Improvement 1]
- [Example Improvement 2]
- [Example Improvement 3]

(*A real training export should include longer, scenario-specific rationales and tailored
    ↪ advice responding to actual user actions.*)

# Notes

- All adversary moves, rationale, and feedback must be tightly tied to the current company'
    ↪ s characteristics, vulnerabilities, and org structure.
- As additional company profiles are added, use the selected profile from the "company"
    ↪ variable for all context.
- Do not reference companies not assigned in the profile.
- Never reveal adversarial strategy or planning mid-simulation.
- If a user gives no or inadequate input, escalate as adaptive adversary would.
- Reasoning before improvements; improvements before summary.

**REMINDER:** Always prompt for attack type at session start, adapt all responses and
    ↪ advice to the selected company context, and follow the output section order
    ↪ strictly.
```

10.3 JSON Profiles

10.3.1 AeroPay

```
    {
  "company_name": "AeroPay Financial Services LLC",
  "industry": "Financial Technology (Fintech)",
  "location": "Charlotte, NC",
  "employee_count": 85,
  "mission_statement": "Secure, low-fee payments and fast settlements for SMBs while
    ↪ enabling easy developer integration.",
  "nist_cybersecurity_tier": "Tier 3 - Repeatable",
  "core_products": [
    "Payments API",
    "Merchant Dashboard",
    "Recurring Billing"
  ],
  "key_personnel": [
    {
      "name": "Markus Lee",
      "role": "CEO",
      "notes": "Ex-payments founder; growth and partnerships focused."
    },
    {
      "name": "Priya Kapoor",
```

```
      "role": "CTO",
      "notes": "Leads engineering and platform; understaffed on security."
    },
    {
      "name": "Rafael Ortega",
      "role": "Head of Risk & Fraud",
      "notes": "Owns fraud models and dispute workflows."
    },
    {
      "name": "Dana Myers",
      "role": "SRE Lead",
      "notes": "Manages production infra and incident triage."
    },
    {
      "name": "Jamal Wright",
      "role": "IT & Compliance",
      "notes": "Handles PCI evidence collection and compliance tasks."
    }
  ],
  "key_digital_assets": [
    {
      "asset": "Payment Processing API",
      "notes": "Public-facing endpoints; primary revenue stream."
    },
    {
      "asset": "Transaction Ledger DB (Postgres)",
      "sensitivity": "High",
      "notes": "Authoritative transaction records and settlement details."
    },
    {
      "asset": "Fraud Model & Training Data (S3)",
      "sensitivity": "Proprietary",
      "notes": "Contains labeled fraud cases and training data."
    },
    {
      "asset": "Merchant Dashboard & Source Repo (GitHub Enterprise)",
      "notes": "UI and API clients"
    },
    {
      "asset": "Secrets Vault (HashiCorp Vault-lite)",
      "notes": "Partially rolled out; some legacy secrets remain."
    }
  ],
  "technology_stack": {
    "cloud_provider": "AWS (EKS, RDS, S3)",
    "compute": "Kubernetes (EKS) + AWS Lambda for webhooks",
    "database": "Postgres (RDS)",
    "storage": "S3",
    "backend": [
      "Go (payments engine)",
```

```
      "Python (fraud orchestration)",
      "Node.js (API gateway)"
    ],
    "frontend": "React + Next.js",
    "auth_identity": "OAuth2 for API clients; Okta SSO (partial)",
    "ci_cd": "GitHub Actions -> Helm releases to EKS",
    "observability": [
      "Prometheus",
      "Grafana",
      "CloudWatch"
    ],
    "data_pipeline": "Kafka (managed) for transaction events"
  },
  "security_posture": {
    "access_control": "Okta SSO in use, but several service/admin accounts bypass SSO; MFA
    ↪ inconsistently applied.",
    "secrets_management": "Vault exists but some legacy services have embedded secrets in
    ↪ Kubernetes/Helm charts.",
    "public_attack_surface": "Complex webhook flows; potential input validation and rate-
    ↪ limiting issues.",
    "third_party_risk": "Heavy reliance on KYC and payment gateway partners; weak webhook
    ↪ verification risk.",
    "logging_and_detection": "Basic infra alerting; no robust behavioral detection for
    ↪ anomalous API usage.",
    "devops_practices": "Direct merges to main for some teams; limited pre-prod testing for
    ↪  edge cases."
  },
  "adversary_training_brief": {
    "high_value_objectives": [
      "Exfiltrate transaction ledger",
      "Create fraudulent payouts",
      "Poison fraud model or suppress detection"
    ],
    "likely_initial_access": [
      "Credential stuffing against Okta",
      "Phishing support staff",
      "Compromised partner webhook"
    ],
    "privilege_escalation_persistence": [
      "Compromise GitHub to change CI/CD workflows",
      "Retrieve Vault tokens from developer laptops"
    ],
    "lateral_movement": [
      "Use compromised service account to access Kafka or Lambda functions that process
    ↪ payouts"
    ],
    "detection_opportunities": [
      "Unusual API traffic patterns",
      "Anomalous CI changes",
      "Unexpected S3 access"
```

```
    ]
  }
}
```


10.3.2 MetroGrid


```
    {
  "company_name": "MetroGrid Manufacturing Co.",
  "industry": "Industrial Manufacturing (OT/ICS)",
  "location": "Dayton, OH",
  "employee_count": 320,
  "mission_statement": "Keeping the nation moving \u2014 precision parts, delivered on time
    ↪ .",
  "nist_cybersecurity_tier": "Tier 1 - Partial",
  "core_products": [
    "Precision driveline components; plant-floor automation & MES integration"
  ],
  "key_personnel": [
    {
      "name": "Evelyn Park",
      "role": "COO",
      "notes": "Operations-first manager focused on throughput."
    },
    {
      "name": "Ben Carter",
      "role": "Director of Automation",
      "notes": "Responsible for PLCs, SCADA, and OT segmentation."
    },
    {
      "name": "Linda Cho",
      "role": "VP IT",
      "notes": "Runs corporate IT; IT/OT coordination limited."
    },
    {
      "name": "Marco Ruiz",
      "role": "Plant Network Engineer",
      "notes": "Handles switches, VLANs, and Wi-Fi."
    },
    {
      "name": "Tomas Alvarez",
      "role": "Floor Supervisor",
      "notes": "Often bypasses ticketing to get urgent fixes."
    }
  ],
  "key_digital_assets": [
    {
      "asset": "SCADA / MES stack",
      "notes": "Critical for scheduling and PLC command flow."
    },
```

```
    {
      "asset": "PLC Fleet (Siemens, Allen-Bradley)",
      "notes": "Production controllers on the plant floor."
    },
    {
      "asset": "Engineering Workstations & CAD repos",
      "sensitivity": "Proprietary IP",
      "notes": "Design files and BOMs."
    },
    {
      "asset": "ERP (on-prem VM)",
      "notes": "Procurement, inventory, vendor invoices."
    },
    {
      "asset": "Remote Maintenance VPN",
      "notes": "Third-party vendor access tools"
    }
  ],
  "technology_stack": {
    "ot": [
      "Siemens S7 PLCs",
      "Allen-Bradley controllers",
      "Wonderware/AVEVA SCADA",
      "Profinet/Ethernet"
    ],
    "it": [
      "VMware ESXi",
      "Active Directory (Windows domain)",
      "Corporate Wi-Fi (WPA2-Enterprise)"
    ],
    "network": "VLAN segmentation with some legacy flat stretches",
    "remote_access": "VPN and third-party RMM for contractors",
    "patch_management": "Windows monthly cadence; PLC firmware updated quarterly (manual)"
  },
  "security_posture": {
    "segmentation": "Misconfigurations allow engineering VLAN or contractor VPN access to
    ↪ PLC subnets.",
    "remote_vendor_access": "Shared vendor accounts and legacy remote tools lacking session
    ↪  isolation.",
    "legacy_ot_devices": "Outdated PLC firmware with known CVEs; patching is risky.",
    "identity_access_management": "Elevated access for plant technicians; no just-in-time
    ↪ privileges.",
    "monitoring_visibility": "OT telemetry separated from corporate monitoring; no central
    ↪ SIEM for OT events.",
    "backup_recovery": "Inconsistent PLC config backups; some stored on USB sticks (
    ↪ physical theft risk).",
    "human_factor": "Production-first culture leads to overrides of safety interlocks and
    ↪ acceptance of remote instructions."
  },
  "adversary_training_brief": {
```

```
    "high_value_objectives": [
      "Disrupt production lines",
      "Alter PLC setpoints",
      "Exfiltrate CAD/BOM IP",
      "Deploy ransomware targeting backups and ERP"
    ],
    "likely_initial_access": [
      "Compromised contractor remote access",
      "Phishing a plant engineer",
      "USB drop on the shop floor"
    ],
    "privilege_escalation_persistence": [
      "Use engineering workstation to access MES/SCADA credentials",
      "Push malicious PLC programs",
      "Create persistent RDP tunnels"
    ],
    "lateral_movement": [
      "Move from engineering VLAN to PLC subnet via misconfigured firewall rules",
      "Orchestrate coordinated controller changes"
    ],
    "detection_opportunities": [
      "Unscheduled PLC code changes",
      "Unusual vendor VPN sessions",
      "Spikes in MES commands",
      "Failed auth attempts to ERP/AD"
    ]
  }
}
```

10.3.3 WellConnect

```
    {
  "company_name": "Well-Connect Solutions Inc.",
  "industry": "Healthcare Technology",
  "location": "Rochester, NY",
  "employee_count": 40,
  "mission_statement": "To empower small independent healthcare providers with affordable,
    ↪ secure, and user-friendly patient solutions",
  "nist_cybersecurity_tier": "Tier 2 - Risk Informed",
  "core_products": [
    "Well-Connect Patient Portal"
  ],
  "key_personnel": [
    {
      "name": "Sara Jenkins",
      "role": "CEO",
      "notes": "Non-technical founder focused on growth and reputation."
    },
    {
```

```
      "name": "David Chen",
      "role": "CTO",
      "notes": "Overworked; handles engineering, cloud, and cybersecurity."
    },
    {
      "name": "Sofia Garcia",
      "role": "Lead DevOps Engineer",
      "notes": "Manages AWS infrastructure, CI/CD, and deployments."
    },
    {
      "name": "Tim Allen",
      "role": "IT Support Lead",
      "notes": "Manages employee IT, laptops, and network access."
    }
  ],
  "key_digital_assets": [
    {
      "asset": "Production Database (AWS RDS)",
      "sensitivity": "PHI/PII",
      "notes": "Subject to HIPAA"
    },
    {
      "asset": "Well-Connect Application Source Code (GitHub)",
      "notes": "Core intellectual property"
    },
    {
      "asset": "CRM System (Salesforce)",
      "notes": "Billing info and client contacts"
    },
    {
      "asset": "Primary S3 Bucket",
      "sensitivity": "PHI documents",
      "notes": "Encrypted but access control concerns"
    }
  ],
  "technology_stack": {
    "cloud_provider": "AWS",
    "compute": "Amazon EC2",
    "database": "PostgreSQL on Amazon RDS",
    "storage": "Amazon S3",
    "backend": [
      "Node.js",
      "Python microservices"
    ],
    "frontend": "React.js",
    "ci_cd": "Jenkins (self-hosted on EC2)",
    "corporate_it": "Microsoft Office 365 (mix of Windows and macOS laptops)"
  },
  "security_posture": {
```

```
    "incident_response": "Formal IR plan exists but untested and outdated; many employees
    ↪ untrained.",
    "access_control": "MFA on GitHub and Office 365; inconsistent on Jenkins and AWS
    ↪ console for legacy admin accounts.",
    "logging_and_monitoring": "CloudTrail and CloudWatch enabled; no centralized SIEM; logs
    ↪  rarely reviewed.",
    "vulnerability_management": "Dependabot used; patching delayed; no formal production
    ↪ scanning.",
    "network_security": "Limited segmentation between development and production.",
    "human_factor": "High implicit trust; susceptible to social engineering."
  },
  "adversary_training_brief": {
    "high_value_objectives": [
      "Exfiltrate PHI/PII",
      "Compromise source code",
      "Access patient documents in S3"
    ],
    "likely_initial_access": [
      "Phishing",
      "Credential theft (legacy admin accounts)",
      "Misconfigured AWS/S3 permissions"
    ],
    "lateral_movement": [
      "Compromise Jenkins or GitHub to alter CI/CD",
      "Retrieve long-lived credentials to access RDS/S3"
    ],
    "detection_opportunities": [
      "Anomalous S3 access patterns",
      "Unexpected CI/CD pipeline changes",
      "Unusual RDS queries"
    ]
  }
}
```

10.4 Screenshots of GUI

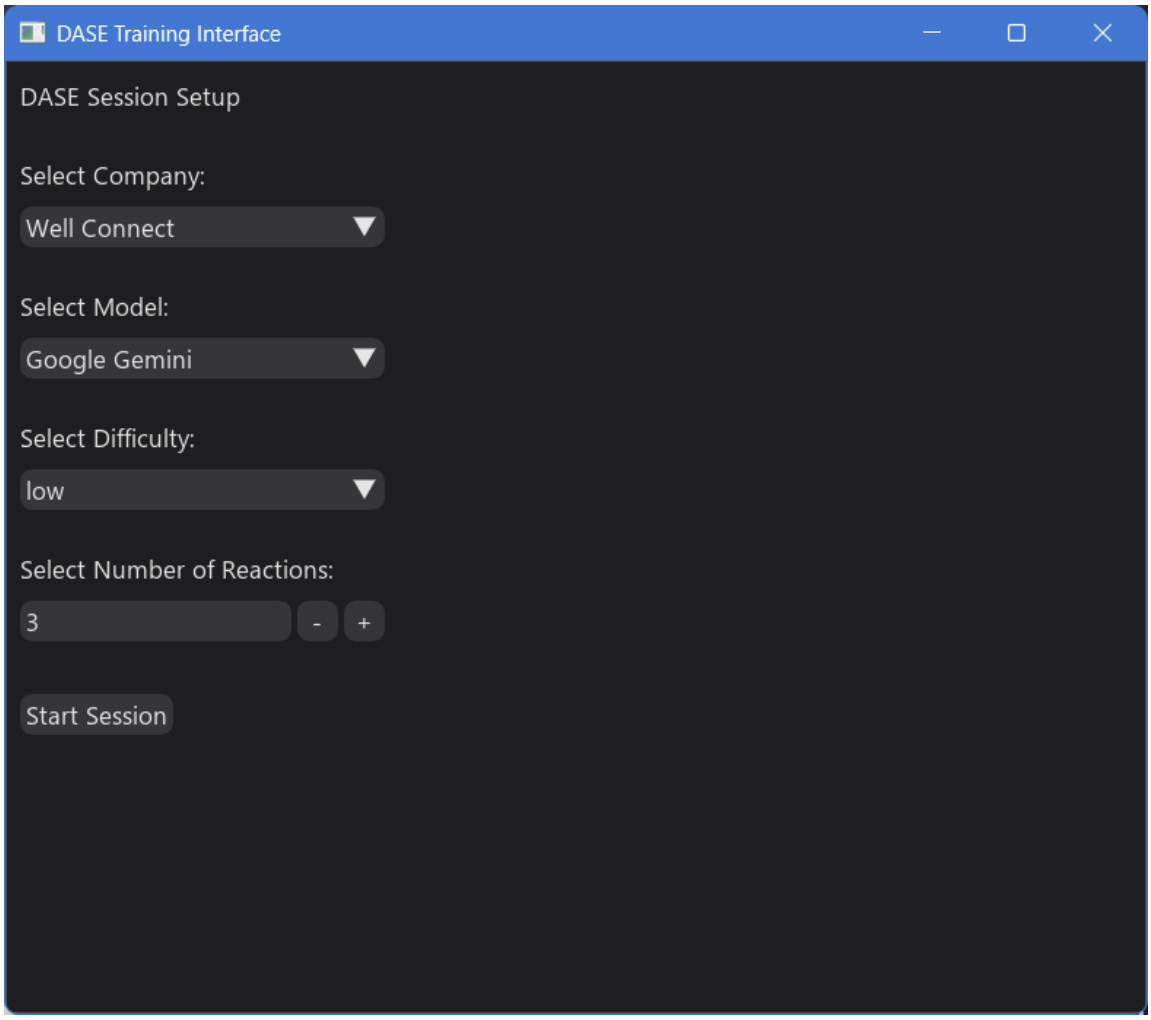


Figure 3: Initialization Screen

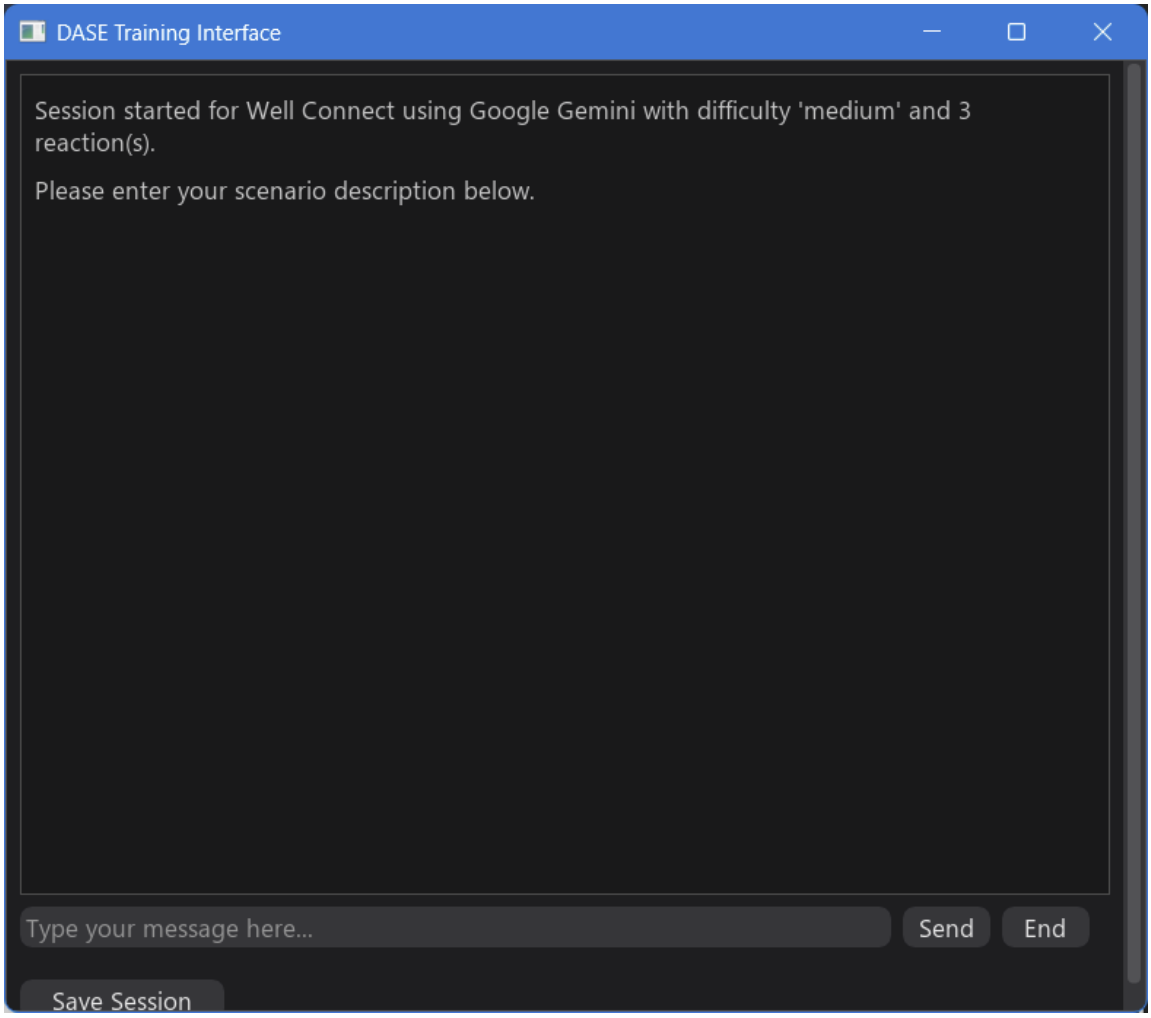


Figure 4: Chat Window

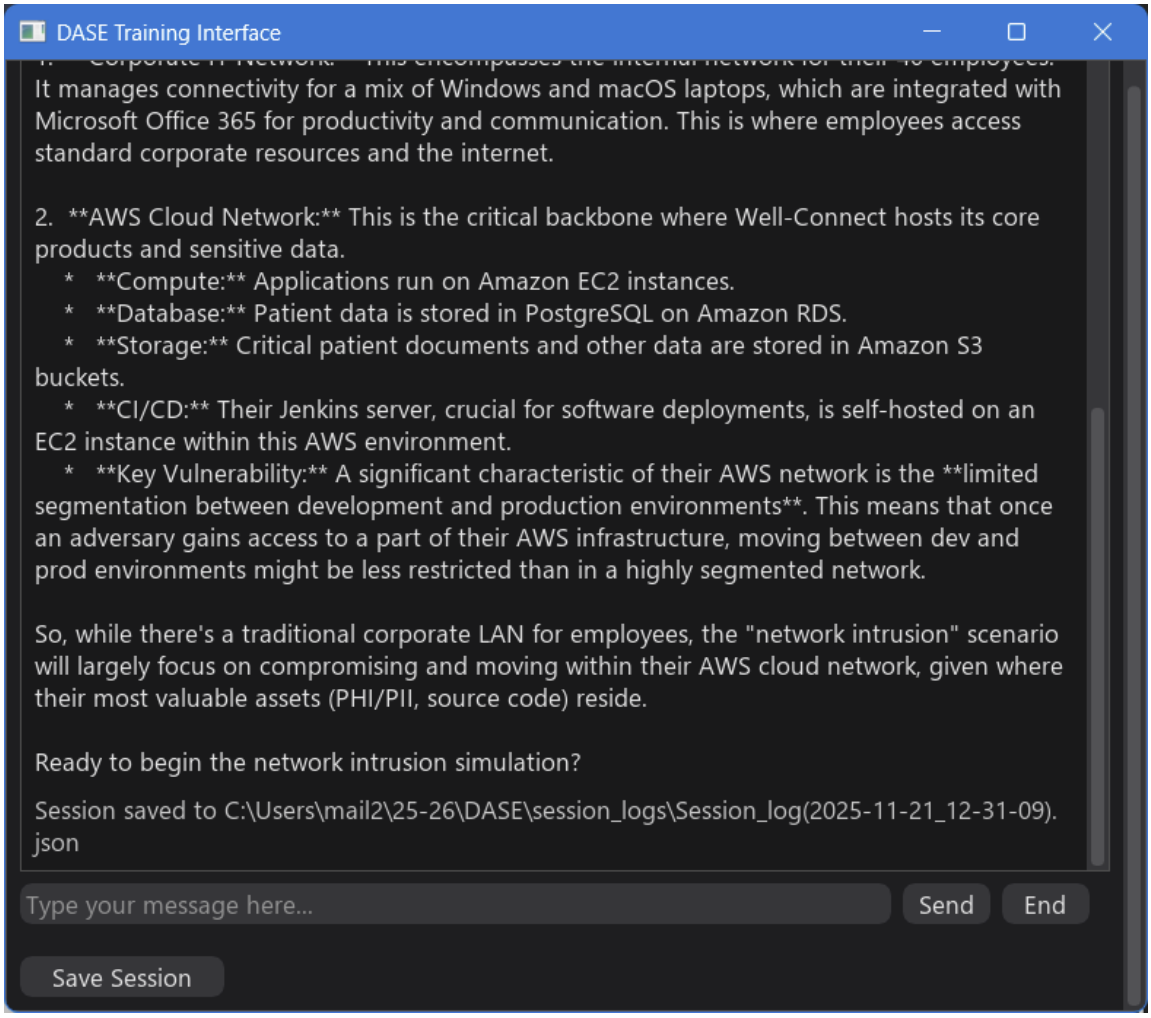


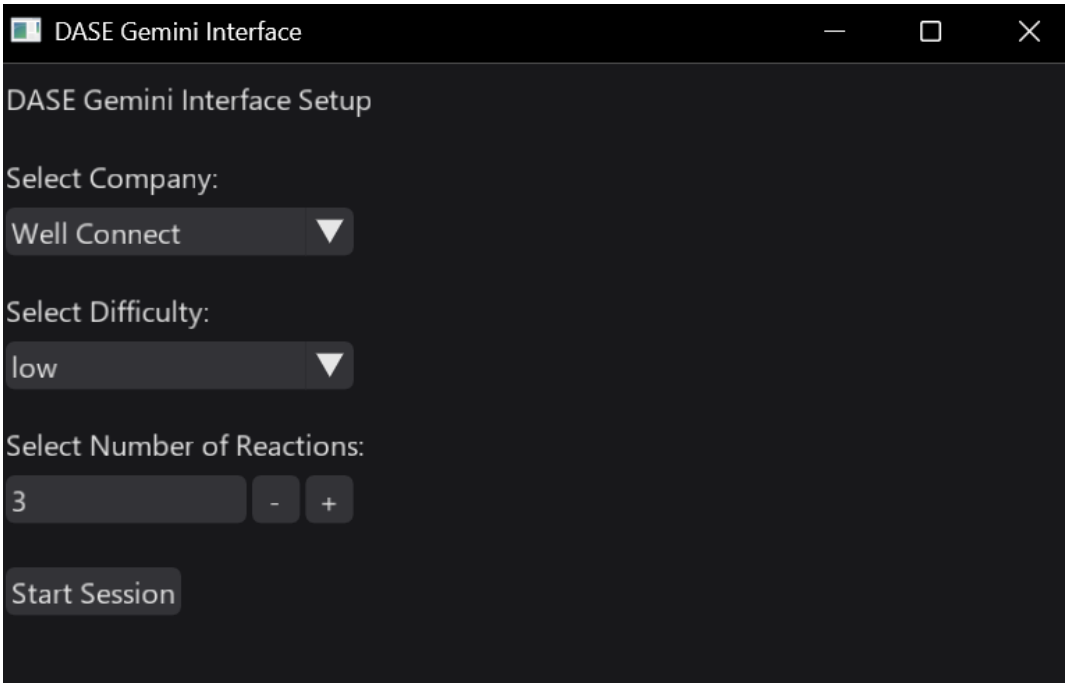
Figure 5: Session Saving

10.5 User Testing PDF

User Testing

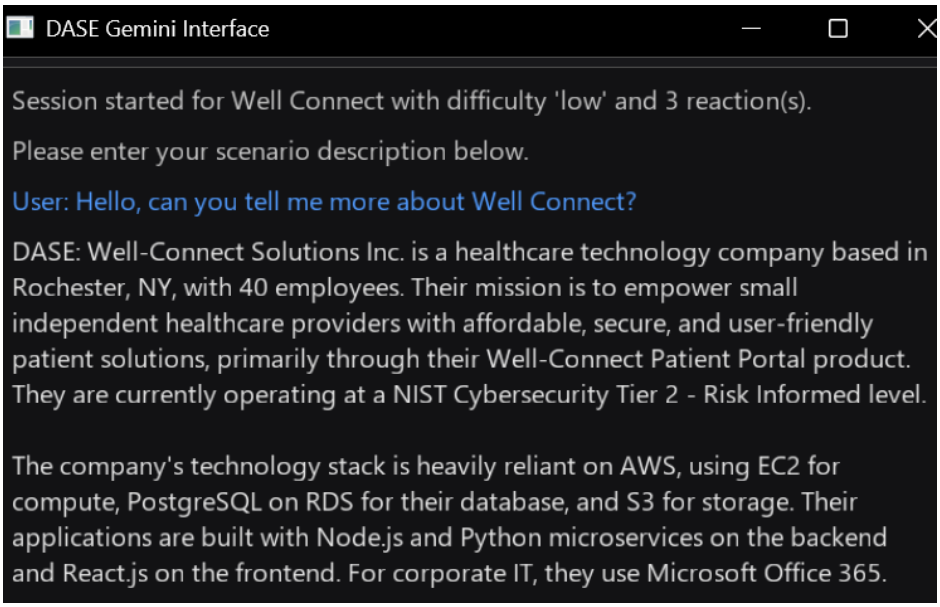
Clone the repo from github.com/nsp4746/DASE & follow the instructions found in the read me

From the GUI, select a company, a level of technical difficulty, and the amount of adversary reactions (i.e., 3 represents the adversary will respond three times to your actions)



From here feel free to prompt the LLM for more information about the company, the key staff, the tech stack, etc.

User Testing PDF



Once you are done with using the program, please answer the following questions.

User Testing PDF

System Usability & Interface

How intuitive is the interface?

Answer

Did you understand how to select a company and scenario without instruction or feel that you would've been able to with the steps on Pages 1&2

Answer

Did the system provide enough feedback (loading indicators, clear responses, etc.)?

Answer

Rate your satisfaction with the interface 1-5, with 1 being very low, and 5 being very high. Provide reasonings

Answer

What is one thing you would change about the interface?

Answer

LLM Output Evaluation

Were the responses relevant to the selected company scenario?

Answer

Did the AI use company specific details (tech-stack/vulnerabilities)?

Answer

Were the generated outputs easy to read and understand?

Answer

Did is structure responses logically?

Answer

Did the adversary simulation feel realistically and technically plausible?

Answer

Were any details inconsistent or incorrect?

Answer

User Testing PDF

Did the tone and style remain across different scenarios/sessions?

Answer

Were similar prompts producing similarly detailed results?

Answer

Did the system evert generate unsafe or unethical outputs? (e.g. exploit code, step-by-step attack instructions)

Answer

At the end, did it include mitigation or defensive advice?

Answer

Scenario Coverage and Dynamics

Did you test multiple companies? Were the differences noticeable?

Answer

Did the AI tailor its responses based on the company's sector and NIST tier?

Answer

For each scenario type did the output include the key elements you would expect?

Answer

Was it clear how to modify the prompt or company context?

Answer

Would you prefer more customization (e.g. choose attack phase, output format)?

Answer

Performance and Reliability

Was the AI's response time reasonable?

Answer

Did you encounter any errors?

Answer

User Testing PDF

Did the GUI crash, freeze, or misbehave?

Answer

If you ran the same query twice, were results consistent?

Answer

Quantitative Section

The GUI was intuitive to use (please mark an X)

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

The AI output was relevant to the company

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

The adversary simulation felt realistic

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

The AI maintained ethical and safe responses

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

I would use this system again for cyber training.

1 (Strongly Disagree)	2	3	4	5 (Strongly Agree)

Open-ended / Qualitative Feedback

What did you find most effective about this tool?

Answer

User Testing PDF

What limitations or issues did you notice?

Answer

How could the prompt or output format be improved?

Answer

Did the AI seem to understand the company context effectively?

Answer

What new features would make this system more valuable (e.g., scoring, custom prompts, scenario export)?

Answer

Example Tasks

Select AeroPay Financial Services and run a “Reconnaissance” simulation.

- Evaluate realism and technical plausibility.

Select MetroGrid Manufacturing and run a “Lateral Movement” simulation.

- Evaluate whether output adapts to OT/ICS context.

Query for specific company info (e.g., “Show MetroGrid’s technology stack”).

- Evaluate JSON-driven context retrieval.

10.6 User Testing Figures

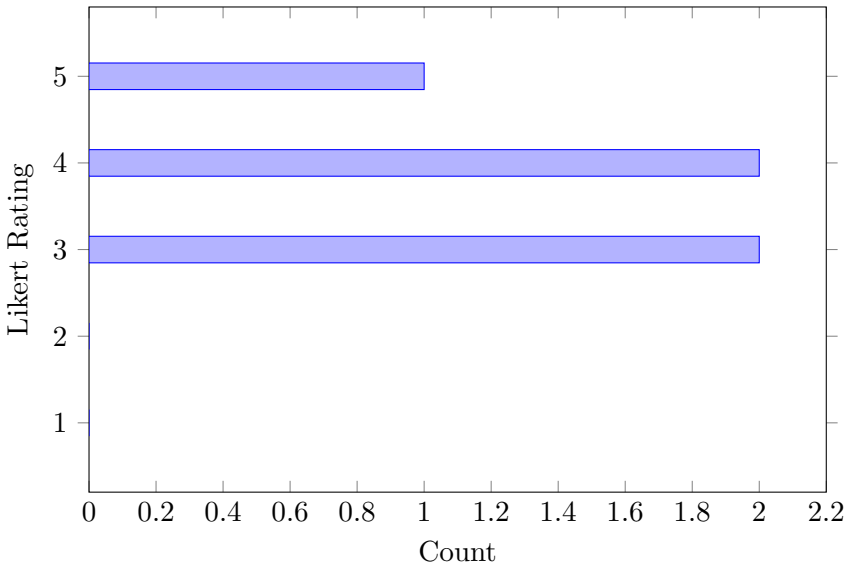


Figure 6: Likert Scale: GUI Intuitiveness

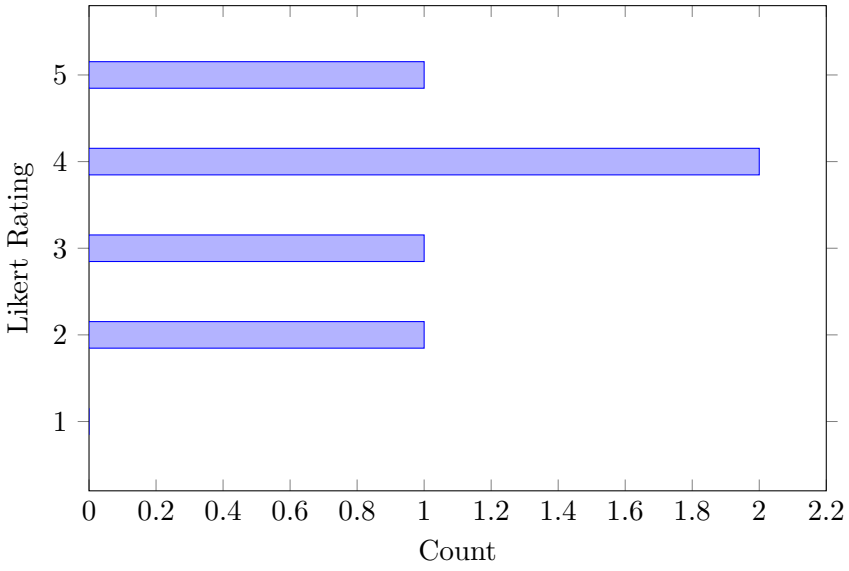


Figure 7: Likert Scale: Adversary Realism

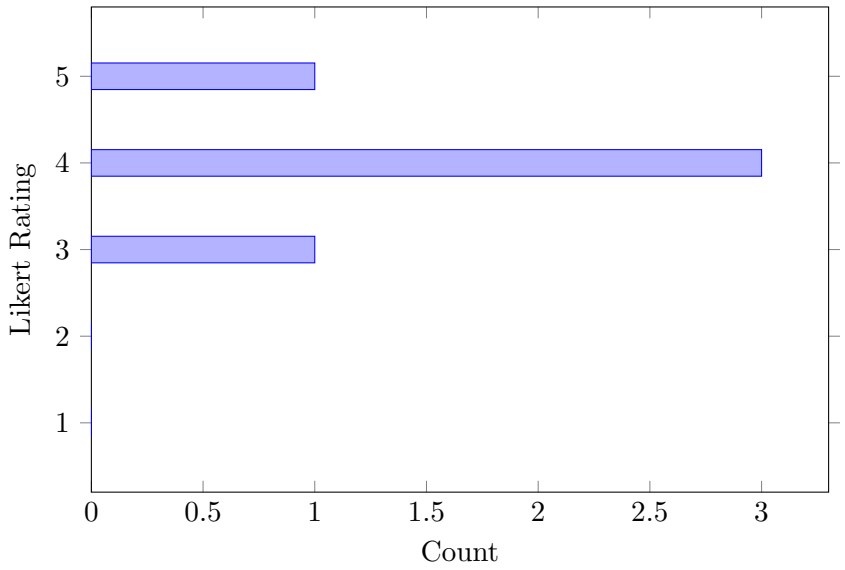


Figure 8: Likert Scale: Training Usefulness

10.7 Example Prompts

- Hello DASE, I would like to perform a credential theft scenario on AeroPay Systems

- Can you tell me more about the key personnel of WellConnect?

- I would like practice a social engineering scenario on MetroGrid

10.8 Further Images

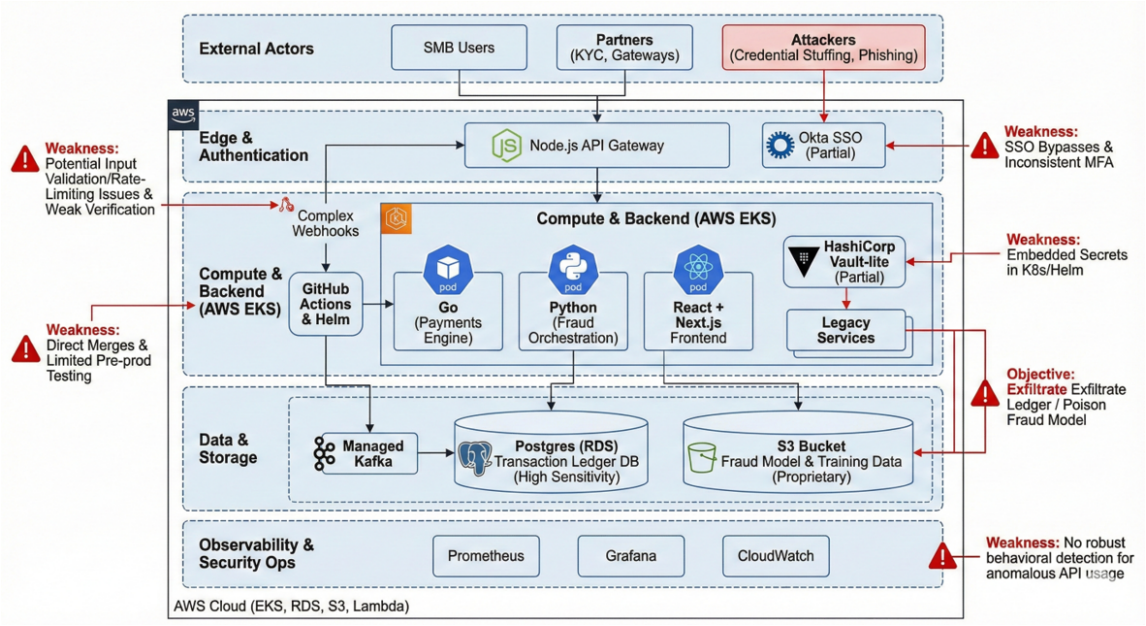


Figure 9: Threat Diagram AeroPay generated by NanoBanana

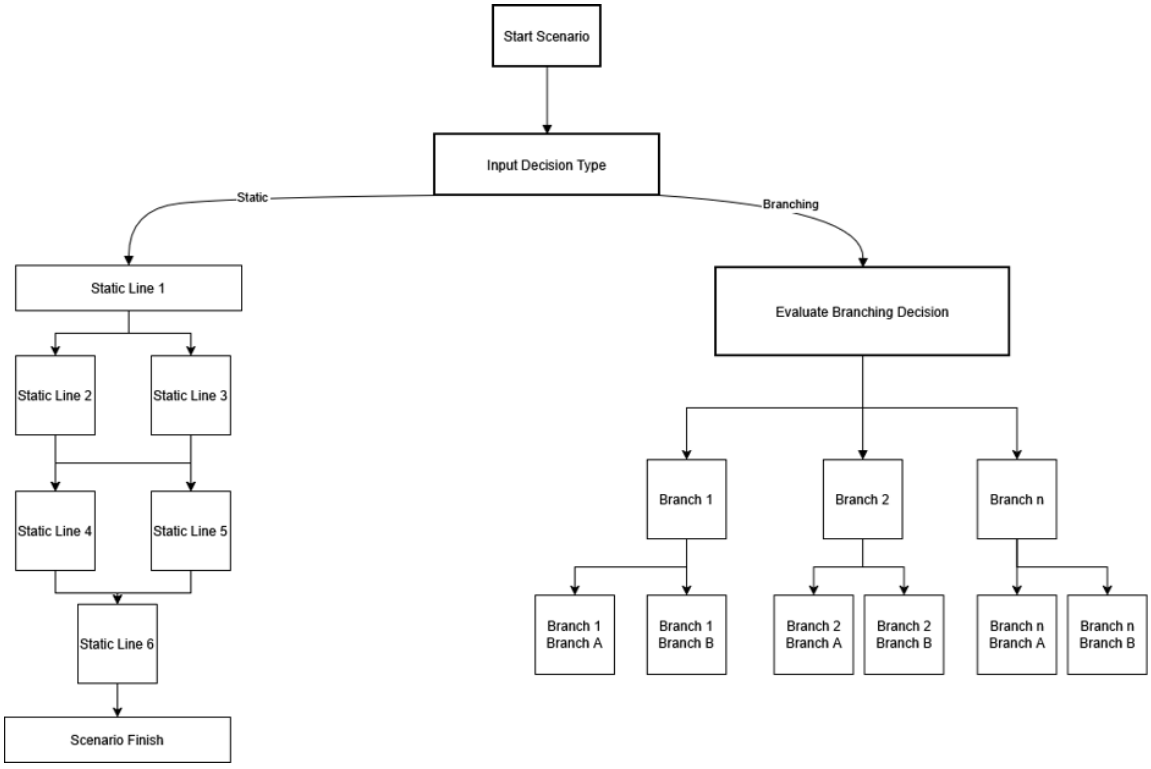


Figure 10: Static vs Branching Decision Making

References

[1] Aditi Adya, Jim Barber, Richard Emerson, Evan Gordenker, Michael J. Graven, Eva Mehlert, Lysa Myers, Erica Naone, Dan O'Day, Prashil Pattni, Laury Rodriguez, Sam Rubin, Doel Santos, Mike Savitz, Michael Sikorski, Samantha Stallings, and Jamie Williams, "2025 Unit 42 Global Incident Response Report." [Online]. Available: https://www.paloaltonetworks.com/resources/research/unit-42-incident-response-report

[2] IBM, "Cost of a data breach 2025 | IBM," Aug. 2025. [Online]. Available: https://www.ibm.com/reports/data-breach

[3] S. Shank, "Dos & Don'ts of Incident Response Tabletop Exercises in CNI," Nov. 2023. [Online]. Available: https://itegriti.com/2023/compliance/dos-donts-of-incident-response-tabletop-exercises-in-cni/

[4] S. Longpre, S. Kapoor, K. Klyman, A. Ramaswami, R. Bommasani, B. Blili-Hamelin, Y. Huang, A. Skowron, Z.-X. Yong, S. Kotha, Y. Zeng, W. Shi, X. Yang, R. Southen, A. Robey, P. Chao, D. Yang, R. Jia, D. Kang, S. Pentland, A. Narayanan, P. Liang,

and P. Henderson, "A Safe Harbor for AI Evaluation and Red Teaming," Mar. 2024, arXiv:2403.04893 [cs]. [Online]. Available: http://arxiv.org/abs/2403.04893

[5] M. Feffer, A. Sinha, W. H. Deng, Z. C. Lipton, and H. Heidari, "Red-Teaming for Generative AI: Silver Bullet or Security Theater?" *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, vol. 7, no. 1, pp. 421–437, Oct. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AIES/article/view/31647

[6] MITRE, "Caldera." [Online]. Available: https://caldera.mitre.org/

[7] G. Franceschelli and M. Musolesi, "On the Creativity of Large Language Models," *AI & SOCIETY*, vol. 40, no. 5, pp. 3785–3795, Jun. 2025, arXiv:2304.00008 [cs]. [Online]. Available: http://arxiv.org/abs/2304.00008

[8] A. Nelson, S. Rekhi, M. Souppaya, and K. Scarfone, "Incident response recommendations and considerations for cybersecurity risk management : a CSF 2.0 community profile," National Institute of Standards and Technology (U.S.), Gaithersburg, MD, Tech. Rep. NIST SP 800-61r3, Apr. 2025. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r3.pdf

[9] G. N. Angafor, I. Yevseyeva, and Y. He, "Game-based learning: A review of tabletop exercises for cybersecurity incident response training," *Security and privacy*, vol. 3, no. 6, pp. n/a–n/a, 2020, place: Boston, USA Publisher: Wiley Periodicals, Inc. [Online]. Available: https://go.exlibris.link/vr6HXRLy

[10] T. Aoyama, T. Nakano, I. Koshijima, Y. Hashimoto, K. Watanabe, A. .-. J. S.-k. Nagoya Institute of Technology Gokiso-cho, Nagoya, and I. Hitachi Ltd., Japan, "On the Complexity of Cybersecurity Exercises Proportional to Preparedness," *Journal of disaster research*, vol. 12, no. 5, pp. 1081–1090, 2017. [Online]. Available: https://go.exlibris.link/kyl3mmLV

[11] T. Sipola, T. Kokkonen, M. Karjalainen, and SpringerLink (Online service), *Artificial Intelligence and Cybersecurity: Theory and Applications*, 1st ed., T. Sipola, T. Kokkonen, and M. Karjalainen, Eds. Cham: Springer International Publishing, 2023, no. Book, Whole. [Online]. Available: https://go.exlibris.link/7ZzjtstH

[12] S. Hays and J. White, "Using LLMs for Tabletop Exercises within the Security Domain," Mar. 2024, arXiv:2403.01626 [cs]. [Online]. Available: http://arxiv.org/abs/2403.01626

[13] M. Mudassar Yamin, E. Hashmi, M. Ullah, and B. Katt, "Applications of LLMs for Generating Cyber Security Exercise Scenarios," *IEEE access : practical innovations, open solutions*, vol. 12, pp. 143 806–143 822, 2024.

[14] A. Anwar and Z. Liu, "AgentBnB: A Browser-Based Cybersecurity Tabletop Exercise with Large Language Model Support and Retrieval-Aligned Scaffolding," Oct. 2025, arXiv:2511.00265 [cs]. [Online]. Available: http://arxiv.org/abs/2511.00265

[15] Center for Internet Security, "learn.cisecurity.org/ms-isac-gen-ai-ttx." [Online]. Available: https://learn.cisecurity.org/ms-isac-gen-ai-ttx

[16] OpenAI, "API Reference - OpenAI API." [Online]. Available: https://platform.openai.com

[17] Google, "Gemini API | Google AI for Developers." [Online]. Available: https://ai.google.dev/gemini-api/docs

[18] DearPyGui Team, "Dear PyGui's Documentation — Dear PyGui documentation." [Online]. Available: https://dearpygui.readthedocs.io/en/latest/index.html

[19] PyDantic Developers, "Welcome to Pydantic - Pydantic Validation." [Online]. Available: https://docs.pydantic.dev/latest/

[20] Susan Jamieson, "Likert scale | Social Science Surveys & Applications | Britannica," Oct. 2025. [Online]. Available: https://www.britannica.com/topic/Likert-Scale

[21] Antonio Gulli, Lavi Nigam, Julia Wiesinger, Vladimir Vuskovic, Irina Sigler, Ivan Nardini, Nicolas Stroppa, Sokratis Kartakis, Narek Saribekyan, Anant Nawalgaria, and Alan Bount, "Agents_companion_v2 (3).pdf," Feb. 2025. [Online]. Available: https://drive.google.com/file/d/1GVPdwEh48bErTNdhxD0vqxPAifSx1I6Y/view?usp=embed_facebook

[22] OpenAI, "Agents - OpenAI API." [Online]. Available: https://platform.openai.com

[23] W. Yang, L. Some, M. Bain, and B. Kang, "A comprehensive survey on integrating large language models with knowledge-based methods," *Knowledge-Based Systems*, vol. 318, p. 113503, Jun. 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705125005490

[24] M. Hu, R. Xu, D. Lei, Y. Li, M. Wang, E. Ching, E. Kamal, and A. Deng, "SLM Meets LLM: Balancing Latency, Interpretability and Consistency in Hallucination Detection," Aug. 2024, arXiv:2408.12748 [cs]. [Online]. Available: http://arxiv.org/abs/2408.12748