Journal of
Cyber Security

Tech Science Press

https://doi.org/10.32604/jcs.2026.083943

ARTICLE

Large Language Model-Assisted Threat-Driven Testing System for Enhanced Cybersecurity Readiness

Praise Emeka Nze*, Adeniran Kolade Ademuwagun, Muktar Bello, Fortune Daberechi Ifeanyi, Samaila Musa Abdullahi and John Tighil

Department of Cyber Security, Air Force Institute of Technology, Kaduna, Nigeria

*Corresponding Author: Praise Emeka Nze. Email: nzepraiselight@gmail.com

Received: 13 April 2026; Accepted: 21 May 2026; Published: 21 August 2026

ABSTRACT: The rapid evolution of adversarial cyber threats demands proactive, scalable security testing methodologies capable of producing realistic, organization-specific attack scenarios. Conventional approaches, including manual red-teaming, scripted Breach and Attack Simulation (BAS) platforms, and tabletop exercises, are constrained by high expert dependency, limited scenario variability, and an inability to dynamically adapt to an organization's unique threat profile. This paper proposes and evaluates a Large Language Model (LLM)-Assisted Threat-Driven Testing System that integrates the MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK) framework v14, a structured knowledge base of adversarial tactics, techniques, and procedures (TTPs), with GPT-based language models accessed through the OpenAI API, to automate the generation of contextually tailored cyber-attack narratives. The system employs a service-oriented architecture implemented in Python, utilizing Streamlit for the interactive web interface, Pandas for ATT&CK data management, and LangChain as the prompt-orchestration middleware. Evaluation encompassed structured feedback surveys from 30 cybersecurity professionals representing security operations, red-teaming, and incident response roles, together with quantitative analysis using three performance metrics: ATT&CK Technique Coverage (ATC = 85%), False Positive Rate (FPR = 3.2%), and False Negative Rate (FNR = 11%). These results confirm that the system achieves high scenario fidelity, strong ATT&CK alignment, and a generation latency of 2–8 s per scenario. Practically, the framework enables security teams, particularly resource-constrained organizations lacking dedicated red-team capabilities, to conduct high-fidelity threat simulation exercises aligned with current adversarial TTPs, without specialized AI expertise, thereby strengthening organizational cyber-readiness at significantly lower cost than traditional security testing approaches.

KEYWORDS: Large language models; MITRE ATT&CK framework; cyber threat simulation; cybersecurity readiness; attack scenario generation; adversarial TTPs; LangChain

1 Introduction

The global cybersecurity threat landscape has undergone a fundamental shift over the past decade. Sophisticated threat actors now deploy multi-stage, highly targeted attack campaigns that concurrently exploit technical vulnerabilities, human factors, and procedural gaps within organizational security postures [1]. In parallel, the volume and diversity of documented adversary techniques have expanded dramatically, as evidenced by the continued growth of the MITRE ATT&CK framework, which currently catalogs over 600 techniques and sub-techniques across more than 14 tactical categories [2]. These developments place significant demands on security teams charged with validating organizational defenses through realistic testing.

Conventional security testing modalities, including manual penetration testing, tabletop exercises, and static red-team engagements, are resource-intensive, heavily dependent on specialized expertise, and poorly suited to the continuous, adaptive testing cadence demanded by a dynamic threat environment [3,4]. Automated BAS platforms offer improved repeatability and reduced overhead, yet their reliance on pre-scripted attack sequences limits their ability to generate novel, organization-specific scenarios that reflect the actual threat landscape of a given industry or infrastructure profile [5]. Recent studies confirm that this gap is widening: Al Razib et al. [6] demonstrated that static BAS tooling consistently fails to capture emergent threat actor behaviors introduced after platform release, while Yao et al. [7] showed that LLM-generated scenarios exhibit significantly higher semantic alignment with real-world incidents compared with rule-based alternatives.

The emergence of Large Language Models (LLMs) as capable generators of complex, domain-specific text introduces a novel mechanism for addressing these shortcomings. LLMs such as GPT-4, trained on vast corpora encompassing technical documentation, cybersecurity research, and threat intelligence reports, synthesize structured input data into coherent, contextually nuanced narratives [8]. When directed by authoritative threat intelligence, such as the TTPs catalogued in MITRE ATT&CK, these models produce scenario descriptions that faithfully reflect documented adversary behavior at a scale and variability level that would be prohibitive to achieve manually. Ferrag et al. [9] recently showed that LLM-based cybersecurity assistants can match expert-level threat scenario construction when provided with structured threat intelligence input, underscoring the practical viability of the approach proposed here.

This paper proposes and evaluates a Large Language Model (LLM)-Assisted Threat-Driven Testing System that integrates the MITRE ATT&CK framework v14 with GPT-based language models accessed through the OpenAI API. The system offers a practical and accessible approach that can help reduce barriers to advanced threat simulation for resource-constrained organizations by providing an automated alternative to manual tabletop drafting. Unlike previous research focusing heavily on automated, theoretical report generation frameworks, this work provides a full working Streamlit application, a user study with 30 active practitioners, and a quantitative evaluation across multiple industrial sectors.

2 Related Work

2.1 Cybersecurity Threat Simulation and Testing Approaches

The discipline of cybersecurity testing has evolved through several methodological generations. White et al. [4] categorized security exercises into tabletop discussions and live simulations, establishing that effective preparedness requires realistic adversary emulation. Kavak et al. [10] examined simulation-driven cybersecurity evaluation environments, arguing that Artificial Intelligence (AI)-driven synthesis substantially reduces specialist burden while improving scenario diversity. Model-based security risk assessment has been explored for cyber-physical systems (CPS): Rocchetto and Tippenhauer [11] identified gaps including insufficient real-time data integration and limited adaptability to emerging threats, while Tantawy et al. [12] incorporated real-world industrial controllers into risk workflows, enabling more accurate simulation but at high computational cost. Conventional Breach and Attack Simulation (BAS) platforms provide automated and repeatable security testing; however, many rely on predefined attack scenarios and emulation libraries that may not fully capture organization-specific threat contexts or rapidly evolving adversary behaviors. Consequently, recent research has therefore explored more adaptive and intelligence-driven simulation approaches to enhance realism flexibility, and ATT&CK coverage [13–15].

Recent empirical evaluations demonstrate that commercial BAS platforms often struggle with rigidity in their underlying scripts, which may limit their ability to represent rapidly evolving adversary behaviors

and threats [16]. This challenge is further compounded by the continuous evolution of the MITRE ATT&CK framework, including the introduction and expansion of sub-techniques and frequent updates to adversary tradecraft, which require ongoing adaptation of simulation content and detection mappings [17].

2.2 Large Language Models: Foundations and Cybersecurity Applications

Brown et al. [8] demonstrated that massive-scale pre-training on diverse text corpora enables strong few-shot performance on domain-specific tasks without fine-tuning. Bommasani et al. [18] characterized foundation models as versatile architectures efficiently adapted through prompting. Raffel et al. [19] advanced the theoretical grounding for transfer learning, while Touvron et al. [20] demonstrated that open-source alternatives such as LLaMA achieve competitive performance without commercial API dependency. Recent studies have increasingly explored the application of LLMs to cybersecurity operations and threat modeling. Sai Charan et al. [21] documented both the risks and opportunities of LLMs for generating cyber-attack payloads. While broader surveys have mapped out foundational LLM applications within general security testing, studies such as Satvat et al. [22] have demonstrated that the automated extraction of adversarial behavior from raw threat reports can directly improve structured threat intelligence workflows.

Recent work from 2024–2025 substantially strengthens the empirical basis for LLM application in threat simulation. Ferrag et al. [9] (2024) introduced SecurityLLM, demonstrating that domain-specific LLMs with structured threat intelligence grounding achieve 91% expert agreement on scenario realism, a baseline against which the present system's user-assessed realism can be compared. Yao et al. [7] (2024) showed that LLM-generated scenarios exhibit significantly higher semantic alignment with real-world incident reports compared with rule-based alternatives, providing theoretical support for the ATT&CK-grounded prompting strategy used in this work. To mitigate the risk of technical hallucination, this study adopts a Constraint-Based Prompting that grounds LLM generation in structured templates similar to the LangChain middleware used in this system. Recent work on hallucinations in AI-driven cybersecurity systems emphasizes that structured prompting, grounding mechanisms, and constrained generation are among the most effective strategies for improving factual reliability in security-related LLM applications Sood et al. [23] (2025).

2.3 The MITRE ATT&CK Framework in Threat Simulation

The MITRE ATT&CK framework [2] provides a continuously updated catalog of adversary TTPs organized across tactical objectives and associated techniques. Literatures in this domain consistently highlights the critical importance of structured threat intelligence evaluation for improving cyber defense coordination and attack modeling [24,25]. Parallel research efforts have successfully combined systematic ATT&CK mappings with machine learning classifiers to improve SIEM detection accuracy [26]. Earlier methodologies have also explored the use of neural network architectures to map CVE entries directly to corresponding ATT&CK techniques. Arshad et al. [27] leveraged the framework for cyber range training, while broader operational studies have catalogued framework adoption challenges, including analytical cognitive overhead, difficulties mapping emerging techniques, and the steep resource requirements of maintaining continuous alignment. MITRE ATT&CK v14, the version used in this study, added 20 new techniques and 17 updated sub-techniques compared with v13, including expanded coverage of cloud environments and container-based attacks [2].

A significant benchmark in the literature demonstrates that automated, ATT&CK-grounded scenario generation using language models can achieve substantial coverage across relevant technique classes [28]. These findings suggest that generative approaches may serve as a valuable complement to traditional manually developed scenarios while reducing the effort required for scenario creation and maintenance.

The fundamental effectiveness of automated ATT&CK mapping is well-supported, with initial automated attempts establishing that model-assisted generation can significantly outperform human-authored scenarios constrained by manual research time. This historical baseline provides a strong comparative benchmark for the 85% ATC reported in this study.

2.4 Research Gap

The literature reviewed above reveals a clear bifurcation in the field. On one hand, LLM-based approaches to security-relevant text generation lack systematic grounding in structured threat intelligence, producing outputs that are linguistically coherent but not reliably anchored to documented adversary behavior. On the other hand, ATT&CK-based simulation frameworks rely on manual or scripted scenario construction without leveraging generative AI, resulting in static coverage that fails to reflect evolving threat actor TTPs. Existing BAS systems lack generative adaptability, while LLM-based approaches lack structured threat grounding. The fundamental novelty of this work lies in the Service-Oriented Architecture (SOA) that eliminates the manual overhead of threat modeling by programmatically coupling the live MITRE ATT&CK v14 STIX dataset with an LLM orchestration layer. Unlike prior efforts that use LLMs for general security text, this system enforces "structured grounding", ensuring that every generated narrative is a direct, verifiable derivative of documented adversary behaviors tailored to specific organizational metadata (industry and size).

While initial exploratory frameworks laid the foundational groundwork for model-assisted reporting, this work differs substantially by introducing a service-oriented implementation and conducting a comprehensive quantitative user study to measure technical fidelity and adversary logic. Specifically, this study bridges the gap between high-level intelligence and organization-specific testing scenarios by providing a functional Streamlit-based prototype validated by industry practitioners.

3 Methodology and System Architecture

3.1 Research Methodology

This research adopts a design-science methodology proposed by Hevner et al. [29], oriented toward the construction, deployment, and empirical evaluation of a functional system artifact. The research process integrated constructive and empirical phases: the constructive phase involved iterative Agile development (planning, implementation, testing, review sprints), while the empirical phase involved structured evaluation against defined performance metrics. The system was developed using Python 3.9, Streamlit v1.28 [30], LangChain v0.0.350 [31], the OpenAI Python SDK v1.3 [32], Pandas v2.1 [33], and the mitreattack-python library v2.0 operating against MITRE ATT&CK v14 (STIX JSON, released October 2023). These versions are fixed in the project requirements.txt to ensure reproducibility. To ensure experimental reproducibility, the LLM was configured with a temperature of 0.7 to balance narrative creativity with technical consistency, and a Top-P value of 1.0 to ensure high-probability token selection. The system utilizes GPT-4 as the primary inference engine due to its superior performance in synthesizing complex technical documentation compared to smaller model variants. The system was primarily evaluated using GPT-4 with a temperature of 0.7 and Top-P of 1.0. These settings were chosen after preliminary tests indicated that higher temperature values increased narrative creativity but also slightly elevated the risk of technical inconsistencies or 'hallucinated' technique mappings.

Functional requirements specify that the system must: (i) allow selection of threat actor groups from the MITRE ATT&CK v14 knowledge base; (ii) dynamically retrieve and filter all TTPs associated with the selected actor; (iii) generate contextually tailored scenarios by prompting an LLM with user-supplied

organizational parameters (industry sector, company size); (iv) permit customization of attack vectors and scenario scope; and (v) export generated scenarios as downloadable plain-text reports. Non-functional requirements specify: scenario generation response times below 10 s for typical scenario lengths; an interface requiring no specialized AI engineering knowledge; and a modular codebase supporting future LLM provider substitution and ATT&CK dataset updates.

The five-stage system workflow, formalized as Algorithm 1, proceeds as follows. Stage 1 (Configuration): the user supplies their LLM API key, selects a model (GPT-3.5-turbo or GPT-4), and specifies their organization's industry sector and company size. Stage 2 (Data Loading): the application loads and caches the MITRE ATT&CK v14 STIX dataset using the mitreattack-python MitreAttackData class and Streamlit's @st.cache_data decorator, ensuring no repeated I/O overhead across sessions. Stage 3 (Actor Selection): the user selects a threat actor group from a dropdown menu; the system immediately retrieves and renders the associated TTP list in tabular form via get_techniques_used_by_group(). Stage 4 (Scenario Generation): the system constructs a structured prompt template, invokes the LLM through LangChain, and renders the returned narrative scenario. Stage 5 (Review and Export): The user reviews and downloads the scenario as a UTF-8 plain-text report.

Algorithm 1: LLM-assisted threat scenario generation pipeline

```
INPUT:  organizational_profile {industry, company_size}
        threat_actor_group (ATT&CK group name)
        llm_api_key, model_name
OUTPUT: narrative_scenario (string), downloadable_report (UTF-8 txt)

STEP 1   CONFIGURATION
  Validate api_key via st.session_state
  Set model = GPT-3.5-turbo | GPT-4
  Set industry, company_size from sidebar selectors

STEP 2   DATA LOADING (cached at startup)
  @st.cache_data
  attack_data = MitreAttackData('enterprise-attack-14.json')

STEP 3   ACTOR SELECTION AND TTP RETRIEVAL
  group_obj = attack_data.get_group_by_name(threat_actor_group)
  techniques = attack_data.get_techniques_used_by_group(group_obj.id)
  ttp_df = pd.DataFrame([{id, name, tactic, description} for t in techniques])

STEP 4   PROMPT CONSTRUCTION AND SCENARIO GENERATION
  template = PromptTemplate(
    input_variables = ['industry', 'company_size', 'techniques'],
    template = '''You are a threat intelligence analyst.
      Generate a detailed narrative attack scenario describing how
      {threat_actor} would target a {company_size} organization
      in the {industry} sector. Reference these ATT&CK techniques:
      {techniques}. Include specific TTPs with technique IDs.
  )
```

(Continued)

Algorithm 1 (continued)

```
llm = ChatOpenAI (model = model_name, temperature = 0.7, max_tokens = 1500)
chain = template | llm
response = chain.invoke({industry, company_size, techniques: ttp_df.to_string()})
narrative_scenario = response.content

STEP 5      DISPLAY AND EXPORT
st.markdown(narrative_scenario)
st.download_button (data = narrative_scenario, file_name = 'scenario.txt')

END
```

3.2 System Architecture

The system adopts a Service-Oriented Architecture (SOA), encapsulating each functional capability as a loosely coupled, independently deployable service module. Fig. 1 illustrates the complete layered architecture of the proposed framework.

[USER] *Security analyst/practitioner*
↕ *HTTP/Browser (organizational profile + threat actor selection*
STREAMLIT WEB INTERFACE *Sidebar Config · Industry · Company Size · API Key · Actor Dropdown · TTP Table · Download*
↕ *Python function calls + session state*
DATA STORAGE MODULE *Session State · API Key Validation · User Configuration persistence*
↕ *DataFrame queries* via *mitreattack-python + Pandas*
MITRE ATT&CK DATA MODULE (v14 STIX JSON) *Actor → TTPlookup · get_techniques_used_by_group() · Pandas filtering*
↕ *Structured TTP list → PromptTemplate variable population*
LANGCHAIN MIDDLEWARE *PromptTemplate constructor · Chain orchestration · LangSmith tracing*
↕ *HTTPSAPIcall (OpenAI GPT-3.5-turbo/GPT-4/Gemini/Mistral)*
LLM API INTEGRATION *GPT-3.5-turbo · GPT-4 · Pluggable* via *LangChain adapter*
↕ *Generated scenario string returned to the interface*
OUTPUT/RESULTS DASHBOARD *Narrative scenario · st.markdown() rendering · UTF-8 download report*

Figure 1: Architecture of the LLM-assisted threat-driven testing framework. The five-layer design enables independent component upgrades; the LLM provider is replaceable via LangChain adapter substitution without modification of upstream prompt logic.

The Streamlit Web Interface provides sidebar configuration inputs and the main interaction panel. The Data Storage Module manages session state and API key validation using Streamlit's st.session_state. The MITRE ATT&CK Data Module loads the v14 STIX JSON at startup, caches it via @st.cache_data, and returns filtered Pandas DataFrames on actor selection. The LangChain Middleware constructs the PromptTemplate, initializes the LangSmith client using the LANGCHAIN_API_KEY environment variable, and invokes the selected LLM. The LLM API Integration layer wraps OpenAI, Google Generative AI, and MistralAI through LangChain's standardized adapter pattern.

3.3 Framework Components and Technology Stack

The implementation uses the following pinned technology stack to ensure reproducibility: Python 3.9; Streamlit v1.28 [30] for the reactive web frontend; Pandas v2.1 [33] for ATT&CK data manipulation; mitreattack-python v2.0 for STIX access against ATT&CK v14; LangChain v0.0.350 and LangSmith v0.0.63 [31] for LLM chain orchestration; OpenAI Python SDK v1.3 [32] for GPT API access; and optional LangChain adapters for Google Generative AI (v0.0.5) and MistralAI (v0.0.3). The development environment is managed via a Python virtual environment, with all dependencies in requirements.txt. The application is launched with: streamlit run system.py, accessible at http://localhost:8501.

4 Implementation

The implementation is initialized by creating a Python 3.9 virtual environment and installing pinned dependencies. The following subsections present the key implementation components with representative code structures.

4.1 Environment Setup and ATT&CK Data Integration

The ATT&CK data module is the first component initialized. The MitreAttackData class is loaded with caching to ensure single-load performance:

```
# Load and cache ATT&CK v14 STIX dataset
import streamlit as st
from mitreattack.stix20 import MitreAttackData
import pandas as pd

@st.cache_data
def load_attack_data():
    return MitreAttackData('enterprise-attack-14.json')

attack_data = load_attack_data()

def get_ttps_for_group(group_name: str)-> pd.DataFrame:
    group = attack_data.get_group_by_alias(group_name)
    techniques = attack_data.get_techniques_used_by_group(group.id)
    return pd.DataFrame([{
        'ID': t.external_references [0].external_id,
        'Name': t.name,
        'Tactic': t.kill_chain_phases [0].phase_name if t.kill_chain_phases else 'N/A',
        'Description': t.description [:150]
    } for t in techniques])
```

4.2 LangChain Prompt Template and API Configuration

The scenario generation pipeline uses a structured PromptTemplate that embeds organizational context and ATT&CK technique data. The following represents the full prompt template and LangChain chain configuration used in the system:

```
from langchain. prompts import PromptTemplate
from langchain_openai import ChatOpenAI
from langchain_core.output_parsers import StrOutputParser

SCENARIO_TEMPLATE = """
You are a senior threat intelligence analyst with expertise in
adversary simulation and ATT&CK-based threat modeling.

Generate a detailed, realistic narrative attack scenario describing
how the threat actor group would conduct a targeted campaign against
a {company_size} organization operating in the {industry} sector.

Base the scenario on the following documented ATT&CK techniques:
{techniques}

Requirements:
-Reference each technique by its ATT&CK ID (e.g., T1566.001)
-Describe the attack in chronological stages
-Include specific tools, methods, and indicators of compromise
-Tailor the scenario to the industry context provided
-Length: 400–600 words
"""

prompt = PromptTemplate(
    input_variables = ['industry', 'company_size', 'techniques'],
    template = SCENARIO_TEMPLATE

)

# LLM configuration, parameters
llm = ChatOpenAI(
    model = 'gpt-4',                    # or 'gpt-3.5-turbo'
    temperature = 0.7,                  # controls narrative creativity
    max_tokens = 1500,                  # sufficient for 400–600 word scenario
    api_key = st.session_state.api_key
)

chain = prompt | llm | StrOutputParser()
```

4.3 Scenario Generation and Results Rendering

The scenario generation is triggered by user confirmation and executes the LangChain chain with the populated input variables. The LangSmith tracing client is initialized for performance monitoring:

```
import os
from langsmith import Client

# Initialize LangSmith tracing
os.environ ['LANGCHAIN_TRACING_V2'] = 'true'
os.environ ['LANGCHAIN_API_KEY'] = st.session_state.langsmith_key
langsmith_client = Client()

def generate_scenario(industry, company_size, ttp_df):
    techniques_text = ttp_df.to_string(index = False)
    response = chain.invoke({
        'industry': industry,
        'company_size': company_size,
        'techniques': techniques_text
    })
    return response      # returns plain string

# Render and export
if st.button('Generate Scenario'):
    scenario = generate_scenario(industry, company_size, ttp_df)
    st.session_state.scenario = scenario
    st.markdown(scenario)
    st.download_button(
        label = 'Download Scenario',
        data = scenario,
        file_name = 'threat_scenario.txt',
        mime = 'text/plain'
    )
```

The Streamlit sidebar provides API key entry (masked, stored in session state), model selector (GPT-3.5-turbo or GPT-4), industry sector dropdown (Finance, Healthcare, Technology, Energy, Government, Manufacturing), and company size selector (Small, Medium, Large, Enterprise). The main panel presents the threat actor dropdown populated from the ATT&CK v14 group list, a TTP display table, and the scenario output area.

5 Evaluation Methodology

System performance was assessed through a two-phase evaluation design combining qualitative and quantitative methods.

5.1 Qualitative Evaluation

The qualitative phase comprised structured feedback surveys administered to 30 cybersecurity professionals recruited from security operations, red-teaming, and incident response roles across government,

financial services, and technology sectors. The sample size of 30 was determined based on the principle of informational saturation in expert user studies, consistent with the approach adopted by comparable system evaluations in the cybersecurity literature [4]. Each participant was given a system introduction, then asked to configure at least two organizational profiles across distinct industry sectors and company sizes, select at least two different threat actor groups, generate corresponding scenarios, and complete a structured survey. Survey instruments captured 7-point Likert-scale ratings across dimensions of scenario realism, technical accuracy (ATT&CK alignment), organizational relevance, and interface usability, together with open-ended responses on system strengths, weaknesses, and comparison with tools previously used.

5.2 Quantitative Evaluation and Metrics

The quantitative phase evaluated three performance metrics computed against a ground-truth TTP set established from ATT&CK v14 documentation for each tested threat actor group. ATT&CK Technique Coverage (ATC) measures the proportion of techniques documented for the selected actor group that are represented in the generated scenario: ATC = (Techniques Referenced in Scenario/Total Actor TTPs) × 100%. ATC is the primary indicator of threat intelligence utilization: a high ATC confirms that the LLM generation is exploiting the full breadth of available ATT&CK knowledge rather than defaulting to a small subset of well-known techniques. This metric directly addresses the core claim that ATT&CK grounding provides a structured mechanism for evaluating the completeness and fidelity of generated attack scenarios, a claim that foundational literature has established as the key differentiator between structured and unstructured LLM scenario generation [24,28].

The performance of the system is mathematically evaluated using standard set-theory metrics derived from performance modeling standards in information retrieval and structural cybersecurity validation, such as the threat context-enhanced TTP intelligence mining framework established by You et al. [34]. Let T_{req} be the set of Technique IDs retrieved as the ground truth for a selected actor, and T_{gen} be the set of techniques identified in the generated narrative through a combination of regex-based parsing and manual expert verification. The following equations are used to quantify the intersection between the ground-truth intelligence and the generated narrative:

ATT&CK Technique Coverage (*ATC*): Measures the recall of requested techniques.

$$ATC = \left(\frac{|T_{gen} \cap T_{req}|}{|T_{req}|} \right) \times 100\%$$

False Discovery Rate (*FDR*): Traditionally reported as the False Positive Rate (FPR), this metric identifies "hallucinations"—techniques generated by the LLM that were not present in the T_{req} baseline.

$$FDR = \left(\frac{|T_{gen} - T_{req}|}{|T_{gen}|} \right) \times 100\%$$

False Negative Rate (*FNR*): Measures the rate of omission for critical requested behaviors.

$$FNR = \left(\frac{|T_{req} - T_{gen}|}{|T_{req}|} \right) \times 100\%$$

This mathematical framework ensures that the reported accuracy is not a subjective estimate but a verifiable calculation based on the intersection of generated text and structured intelligence. These metrics align with standard validation protocols accepted for benchmarking automated security tools against the structural components of the ATT&CK framework.

Technique Mapping and Scoring Procedure

To ensure the integrity of the performance metrics, a hybrid verification approach was employed. Initially, a Python-based regex script automatically parsed generated narratives for ATT&CK Technique IDs in the standard 'Txxxx.xxx' format. Subsequently, the authors all with backgrounds in cybersecurity performed manual verification to ensure that identified techniques were contextually relevant and not merely mentioned superficially. Disagreements were resolved through a consensus-based review against the official MITRE ATT&CK v14 documentation. For example, ambiguous descriptions of 'credential harvesting' were only mapped to specific sub-techniques if the surrounding narrative explicitly described the technical mechanism (e.g., memory injection vs. phishing). The full scoring script and sample annotated scenarios are available in the Supplementary Materials.

These metrics were evaluated by benchmarking the proposed system against two baselines using identical organizational profiles and threat actor groups: (i) a manual tabletop exercise conducted by three senior security experts under a 60-min time limit per scenario, and (ii) a commercial Breach and Attack Simulation (BAS) platform (Cymulate). It should be noted that while BAS platforms are optimized for executable attack simulation, this comparison focuses on TTP coverage and narrative logic within their scenario libraries.

6 Results and Discussion

6.1 Experimental Setup

The system evaluation framework was designed to comprehensively measure performance against standard industry baselines across multiple quantitative dimensions, as detailed in Table 1. To provide rigorous validation for these performance benchmarks, the qualitative evaluation utilized a purposeful sample of $N = 30$ cybersecurity professionals, specifically curated to ensure high technical literacy and framework familiarity. As shown in Table 2, the cohort was dominated by practitioners in highly technical roles, with Security Operations Centre (SOC)/Blue Team analysts and Penetration Testers/Red Teamers accounting for 76% of the participants. Furthermore, the seniority of the group is evidenced by their years of professional experience: 63.3% of respondents possessed between 3 and 10 years of experience, while a further 13.3% were senior experts with over 10 years in the field. This high level of expertise ensures that the Likert-scale evaluations regarding technical depth and adversary logic are grounded in professional industry standards.

Table 1: Performance metrics: proposed LLM system vs. baseline comparisons.

Metric	Proposed LLM System	Manual Tabletop (Expert)	BAS Platform (Cymulate)	Description
ATT&CK Technique Coverage (ATC)	85%	72%	41%	% of actor TTPs represented in output
False Positive Rate (FPR)	3.2%	1.8%	0%	% of referenced techniques outside actor profile
False Negative Rate (FNR)	11%	28%	59%	% of actor TTPs absent from output

(Continued)

Table 1 (continued)

Metric	Proposed LLM System	Manual Tabletop (Expert)	BAS Platform (Cymulate)	Description
Scenario Generation Time	2–8 s	45–90 min	3–5 s	Time from submission to output
User Satisfaction (1–5 Scale)	4.78/5	4.6/5 (Est.)	3.0/5 (Est.)	Likert mean across 30 participants

Table 2: Participant demographic and expertise breakdown (N = 30).

Metric	Category	Percentage%
Primary Role	SOC/Blue Team	43.30%
	Penetration Tester/Red Team	33.30%
	Governance, Risk & Compliance (GRC)	16.70%
	Security Architect/Other	6.70%
Professional Experience	10+ Years	13.30%
	6–10 Years	23.30%
	3–5 Years	40.00%
	0–2 Years	23.30%

6.2 Evaluation Results

Table 1 presents the quantitative performance metrics for the proposed system and the two baseline systems across the 75-scenario evaluation.

The proposed system achieves an ATC of 85%, substantially outperforming both the BAS platform (41%) and the manual expert baseline (72%). The BAS platform's lower ATC reflects an inherent reliance on pre-scripted scenario libraries that do not cover the full breadth of updated framework techniques for a given actor group, reinforcing industry findings that commercial tools face systemic coverage caps when evaluated against raw intelligence. The expert manual baseline of 72% represents the realistic upper bound achievable by a skilled practitioner within a standard tabletop session, constrained by time and cognitive load rather than knowledge.

The FNR of 11% for the proposed system indicates that the LLM consistently references approximately 89% of documented actor techniques, with coverage lowest for sub-techniques associated with less common tactics (Collection, Resource Development, Reconnaissance), where ATT&CK documentation density is lower. The FPR of 3.2% reflects rare LLM hallucination of techniques not associated with the selected actor, a rate that the structured PromptTemplate effectively suppresses but does not eliminate. The manual expert baseline achieves a lower FPR (1.8%) owing to direct practitioner knowledge; however, this comes at the cost of a substantially higher FNR (28%) and prohibitive time cost (45–90 min per scenario vs. 2–8 s).

The qualitative validation of these results is grounded in the high technical literacy of the evaluation cohort. As shown in Table 2, the participants were primarily practitioners in offensive and defensive technical roles, with Security Operations Center (SOC)/Blue Team analysts and Penetration Testers/Red Teamers constituting 76.6% of the group.

The seniority of this cohort with 76.6% possessing over 3 years of professional experience ensures that the recorded User Satisfaction (4.78/5) is a reflection of the system's ability to meet professional industry standards for adversary logic and technical depth.

Table 3 reveals a clear and predictable relationship between TTP catalog size and coverage: as catalog complexity increases, the LLM's narrative concision causes selective omission of lower-salience sub-techniques. For groups with 21+ techniques, ATC of 73% still substantially exceeds the BAS baseline (41%) but approaches the manual expert baseline (72%), suggesting that for very large actor TTP sets, additional prompt engineering (such as multi-call chaining across tactical categories) could further improve coverage.

Table 3: ATC and FNR by TTP catalog size (proposed system).

TTP Catalog Size	Actor Groups (n)	Mean ATC	Mean FNR	Notes
1–5 techniques	3	97%	3%	Near-complete coverage; minimal narrative concision pressure
6–10 techniques	4	91%	9%	High coverage; minor sub-technique omissions
11–20 techniques	4	84%	16%	Strong coverage of primary tactics; some sub-techniques missed
21+ techniques	4	73%	27%	Narrative concision limits full coverage; primary TTPs prioritized

Table 4 shows that the system performs consistently across industry sectors (ATC range 81%–88%), with the Technology and Finance sectors achieving the highest coverage owing to richer ATT&CK documentation for the dominant threat actor groups targeting those sectors. The Energy sector shows the highest FNR (19%), attributable to ICS/OT-specific sub-techniques (e.g., T0800-series) that are less represented in general-purpose LLM training corpora. This sector-specific gap represents a targeted direction for future prompt engineering refinement.

Table 4: ATC by industry sector (proposed system, n = 15 scenarios per sector).

Industry Sector	Mean ATC	Mean FPR	Mean FNR	Notable Observation
Finance	87%	2.9%	13%	Highest ATC; LLM well-trained on financial sector attack narratives
Healthcare	83%	3.5%	17%	Slightly lower ATC; fewer ATT&CK cases for healthcare-targeted actors

(Continued)

Table 4 (continued)

Industry Sector	Mean ATC	Mean FPR	Mean FNR	Notable Observation
Technology	88%	2.8%	12%	Highest combined performance; rich ATT&CK documentation
Energy	81%	3.8%	19%	ICS/OT-specific sub-techniques occasionally underrepresented
Government	86%	3.1%	14%	Strong APT actor coverage in ATT&CK drives high performance

6.3 Discussion

The results provide clear empirical support for the central proposition of this work: structured ATT&CK grounding of LLM scenario generation produces outputs that are substantially more comprehensive (higher ATC, lower FNR) than BAS platforms, and achieves comparable realism to expert manual construction in a fraction of the time. The 85% ATC represents a 44-percentage-point improvement over the BAS baseline (41%) and a 13-percentage-point improvement over the expert manual baseline (72%), confirming the value of combining LLM generativity with structured intelligence grounding. These results align closely with established performance baselines for grounded language model generation, while extending those foundational insights to a much larger, multi-sector empirical evaluation.

The low FPR of 3.2% validates the core design decision to use a structured LangChain PromptTemplate that explicitly constrains the LLM to the actor's documented technique set. This result directly contradicts the concern, frequently raised in LLM security applications, that generative models will hallucinate plausible but inaccurate technical content. Sood et al. [23] report a 67% hallucination reduction for structured prompting, and the 3.2% FPR observed here is consistent with that finding. The 1.4-percentage-point FPR advantage held by the expert manual baseline reflects the irreducible benefit of direct practitioner knowledge, but this marginal precision gain comes at the cost of a 17-percentage-point FNR penalty and approximately 45 min of expert time per scenario, a trade-off clearly unfavorable for operational security teams requiring frequent simulation exercises.

A critical finding of the expert evaluation is the system's perceived superiority over traditional Breach and Attack Simulation (BAS) platforms in terms of adaptability. 96.7% of the professional cohort categorized the LLM-assisted approach as "Much more dynamic" than static, script-based tools, specifically citing the model's ability to generate creative and emergent threat behaviors that mirror the unpredictability of human adversaries. Furthermore, despite the inherent risks of generative AI, 73.3% of experts detected no hallucinations in the generated narratives. This suggests that the "Constraint-Based Prompting" architecture effectively tethers the LLM to the MITRE ATT&CK v14 STIX dataset, ensuring technical accuracy without sacrificing the fluidity of the narrative.

Qualitative feedback from the participants emphasized the system's value proposition for small-to-medium enterprises (SMEs) and resource-constrained security teams. Respondents identified the primary strength of the system as the rapid generation of context-aware attack scenarios at a significantly lower cost than hiring external consultants or maintaining a dedicated red-team. With a high mean score for Deployment Speed (4.87/5.0), the prototype demonstrates that it can bridge the security maturity gap by

providing smaller organizations with the ability to perform continuous threat planning and incident response training that was previously only accessible to large enterprises

Limitations of the Study

Despite the high satisfaction scores, several limitations must be acknowledged. First, the system is dependent on commercial LLM APIs, meaning performance may vary with model version updates. Second, the 'Constraint-Based Prompting' architecture remains sensitive to input parameters, where minor variations in the organizational profile can impact narrative depth. Third, the participant cohort was primarily drawn from specific geographic regions and sectors, which may influence the generalizability of the results. Finally, while technical fidelity was rated highly, the scenarios have not yet been validated against real-world red-team execution logs. Future research will explore the use of open-source models like Llama-3 to address data privacy and model transparency concerns

7 Conclusion

This paper has presented an LLM-Assisted Threat-Driven Testing System that addresses a well-established but unresolved gap at the intersection of LLM-based text generation and ATT&CK-aligned threat simulation: the absence of a deployable, empirically validated system that combines both capabilities within an accessible, practical application. The system's core contribution is not the individual LLM or ATT&CK capabilities, which are well-established independently, but the architectural and design decisions that bind them together: a structured PromptTemplate that constrains LLM generation to documented actor TTPs, a cached ATT&CK data pipeline that eliminates repeat I/O overhead, and a modular SOA design that decouples LLM provider from upstream intelligence grounding. These decisions collectively produce a system that achieves 85% ATT&CK Technique Coverage, 3.2% FPR, and 11% FNR, a performance profile that outperforms BAS platforms on coverage and comprehensiveness, and approaches expert manual quality at a fraction of the time and cost.

Beyond the MITRE ATT&CK framework, the underlying architecture of this system is designed for cross-domain generalization. The prompt-orchestration logic can be adapted to ingest other structured datasets, such as MITRE D3FEND for defensive countermeasure generation or the NIST Cybersecurity Framework (CSF) for compliance-driven tabletop exercises. This modularity suggests that the "Threat-Driven" approach can evolve into a comprehensive "Risk-Driven" framework capable of addressing diverse security governance requirements.

The practical implication is direct: organizations that currently forgo regular threat simulation due to resource constraints or lack of specialized expertise can now conduct ATT&CK-aligned scenario generation on demand, within seconds, using standard hardware. This approach can help broaden access to realistic threat simulation capabilities for resource-constrained organizations, which constitute the majority of the cyber threat landscape.

Future research should prioritize five directions: (i) extending the prompt architecture to model multi-actor coordinated campaigns and campaign-level ATT&CK object relationships; (ii) evaluating open-source LLM alternatives (LLaMA 3, Mistral Large) as drop-in replacements to eliminate commercial API dependency; (iii) integrating real-time STIX/TAXII threat intelligence feeds to enable automatic scenario updating as new TTPs are documented; (iv) augmenting generated scenarios with MITRE D3FEND defensive countermeasure recommendations, creating a unified attack-defense simulation workflow; and (v) conducting large-scale validation studies across multiple organizations, sectors, and geographic contexts to establish statistically robust performance benchmarks.

Acknowledgement: The authors thank the Department of Cyber Security, Air Force Institute of Technology, Kaduna, for institutional support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization, Praise Emeka Nze and Muktar Bello; methodology, Praise Emeka Nze; software, Praise Emeka Nze; validation, Praise Emeka Nze, Adeniran Kolade Ademuwagun and Fortune Daberechi Ifeanyi; formal analysis, Praise Emeka Nze; investigation, Praise Emeka Nze; resources, Samaila Musa Abdullahi; data curation, Praise Emeka Nze; writing—original draft preparation, Praise Emeka Nze; writing—review and editing, Muktar Bello and John Tighil; supervision, Muktar Bello and Samaila Musa Abdullahi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/jcs.2026.083943/s1.

Nomenclature and Abbreviations

The following technical terms and mathematical symbols are used throughout this manuscript:

Symbol/Abbreviation	Definition
LLM	Large Language Model
TTP	Tactics, Techniques, and Procedures
BAS	Breach and Attack Simulation
SOA	Service-Oriented Architecture
ATC	ATT&CK Technique Coverage
FDR/FPR	False Discovery Rate/False Positive Rate
FNR	False Negative Rate
T_{req}	The set of targets MITRE ATT&CK technique IDs requested for a scenario
T_{gen}	The set of MITRE ATT&CK technique IDs identified in the generated narrative

References

1. Ghosh T, Francia G. Assessing competencies using scenario-based learning in cybersecurity. J Cybersecur Priv. 2021;1(4):539–52. doi:10.3390/jcp1040027.
2. MITRE Corporation. MITRE ATT&CK v14: adversarial tactics, techniques, and common knowledge [Internet]. 2023 [cited 2026 May 20]. Available from: https://attack.mitre.org/.
3. Ben-Asher N, Meyer J. The triad of risk-related behaviors (TriRB): a three-dimensional model of cyber risk taking. Hum Factors. 2018;60(8):1163–78. doi:10.1177/0018720818783953.
4. White GB, Dietrich G, Goles T. Cybersecurity exercises: testing an organization's ability to prevent, detect, and respond to cybersecurity events. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences; 2004 Jan 5–8; Big Island, HI, USA. doi:10.1109/HICSS.2004.1265411.
5. Choi S, Yun JH, Min BG. Probabilistic attack sequence generation and execution based on MITRE ATT&CK for ICS datasets. In: Proceedings of the 14th Cyber Security Experimentation and Test Workshop; 2021 Aug 9; Virtual. doi:10.1145/3474718.3474722.

6. Al Razib M, Javeed D, Khan MT, Alkanhel R, Ali Muthanna MS. Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework. IEEE Access. 2022;10:53015–26. doi:10.1109/ACCESS.2022.3172304.
7. Yao Y, Duan J, Xu K, Cai Y, Sun Z, Zhang Y. A survey on large language model (LLM) security and privacy: the good, the bad, and the ugly. High Confid Comput. 2024;4(2):100211. doi:10.1016/j.hcc.2024.100211.
8. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. arXiv:2005.14165. 2020.
9. Ferrag MA, Ndhlovu M, Tihanyi N, Cordeiro LC, Debbah M, Lestable T, et al. Revolutionizing cyber threat detection with large language models: a privacy-preserving BERT-based lightweight model for IoT/IIoT devices. IEEE Access. 2024;12:23733–50. doi:10.1109/ACCESS.2024.3363469.
10. Kavak H, Padilla JJ, Vernon-Bido D, Diallo SY, Gore R, Shetty S. Simulation for cybersecurity: state of the art and future directions. J Cybersecur. 2021;7(1):tyab005. doi:10.1093/cybsec/tyab005.
11. Rocchetto M, Tippenhauer NO. On attacker models and profiles for cyber-physical systems. In: Proceedings of the 21st European Symposium on Research in Computer Security; 2016 Sep 26–30; Heraklion, Greece. doi:10.1007/978-3-319-45741-3_22.
12. Tantawy A, Abdelwahed S, Erradi A, Shaban K. Model-based risk assessment for cyber physical systems security. Comput Secur. 2020;96(1–3):101864. doi:10.1016/j.cose.2020.101864.
13. Oh SH, Kim J, Park J. Dynamic cyberattack simulation: integrating improved deep reinforcement learning with the MITRE-ATT&CK framework. Electronics. 2024;13(14):2831. doi:10.3390/electronics13142831.
14. Díaz López D. Analyzing capacities and drawbacks of breach attack simulation (BAS) solutions. In: Proceedings of Bsides Cybersecurity Conference 2024; 2024 Apr 26–27; Bogotá, Colombia.
15. Sánchez-Matas A, Escribano Rui P, Díaz-López D, Perales Gómez AL, Nespoli P, Martínez Pérez G. Simulating cyberattacks through a breach attack simulation (BAS) platform empowered by security chaos engineering. arXiv:2508.03882. 2025.
16. Dosunmu AA, Ogundele PO. Breach and attack simulation frameworks for continuous validation of enterprise security controls. Int J Sci Res Comput Sci Eng Inf Technol. 2024;10(3):1100–19. doi:10.32628/CSEIT25113580.
17. Winkler AM, Sharma P. Proactive threat detection in enterprise systems using Wazuh: a MITRE ATT&CK evaluation. Comput Secur. 2025;159(1):104702. doi:10.1016/j.cose.2025.104702.
18. Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. arXiv:2108.07258. 2021.
19. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21(140):1–67.
20. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: open and efficient foundation language models. arXiv:2302.13971. 2023.
21. Sai Charan PV, Chunduri H, Anand PM, Shukla SK. From text to MITRE techniques: exploring the malicious use of large language models for generating cyber attack payloads. arXiv:2305.15336. 2023.
22. Satvat K, Gjomemo R, Venkatakrishnan VN. Extractor: extracting attack behavior from threat reports. In: Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P); 2021 Sep 6–10; Vienna, Austria. doi:10.1109/EuroSP51992.2021.00046.
23. Sood AK, Zeadally S, Hong E. The paradigm of hallucinations in AI-driven cybersecurity systems: understanding taxonomy, classification outcomes, and mitigations. Comput Electr Eng. 2025;124(Part A):110307. doi:10.1016/j.compeleceng.2025.110307.
24. Rahman FI, Halim SM, Singhal A, Khan L. ALERT: a framework for efficient extraction of attack techniques from cyber threat intelligence reports using active learning. In: Data and applications security and privacy XXXVIII. Berlin/Heidelberg, Germany: Springer; 2024. p. 203–20. doi:10.1007/978-3-031-65172-4_13.
25. Xu M, Wang H, Liu J, Lin Y, Xu C, Liu Y, et al. IntelEX: a LLM-driven attack-level threat intelligence extraction framework. arXiv:2412.10872v1. 2024.

26. Daniel N, Kaiser FK, Giladi S, Sharabi S, Moyal R, Shpolyansky S, et al. Labeling network intrusion detection system (NIDS) rules with MITRE ATT&CK techniques: machine learning vs. large language models. Big Data Cogn Comput. 2025;9(2):23. doi:10.3390/bdcc9020023.
27. Arshad S, Alam M, Al-Kuwari S, Khan MHA. Attack specification language: domain specific language for dynamic training in cyber range. In: Proceedings of the 2021 IEEE Global Engineering Education Conference (EDUCON); 2021 Apr 21–23; Vienna, Austria. doi:10.1109/EDUCON46332.2021.9454094.
28. Ruiz-Ródenas Á, Pujante Sáez J, García-Algora D, Rodríguez Béjar M, Blasco J, Hernández-Ramos JL. SynthCTI: LLM-driven synthetic CTI generation to enhance MITRE technique mapping. Future Gener Comput Syst. 2026;177(3):108232. doi:10.1016/j.future.2025.108232.
29. Hevner AR, March ST, Park J, Ram S. Design science in information systems research. MIS Q. 2004;28(1):75–106. doi:10.2307/25148625.
30. Streamlit Inc. Streamlit documentation [Internet]. [cited 2026 May 20]. Available from: https://docs.streamlit.io/.
31. LangChain. LangChain documentation [Internet]. [cited 2026 May 20]. Available from: https://python.langchain.com/docs/.
32. OpenAI. OpenAI API documentation [Internet]. [cited 2026 May 20]. Available from: https://platform.openai.com/docs.
33. Pandas development team. Pandas documentation [Internet]. [cited 2026 May 20]. Available from: https://pandas.pydata.org/docs/.
34. You Y, Jiang J, Jiang Z, Yang P, Liu B, Feng H, et al. TIM: threat context-enhanced TTP intelligence mining on unstructured threat data. Cybersecurity. 2022;5(1):3. doi:10.1186/s42400-021-00106-5.