International Journal of Management, IT & Engineering
Vol. 15 Issue 8, August 2025,
ISSN: 2249-0558 Impact Factor: 7.119
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Red Teaming as a Service (RTaaS) for Cloud-Hosted GenAI: A Responsible AI Perspective

Pavan Kumar Adepu
GenAI/LLM Research Leader
Seattle, USA

Sai Kumar Kalya
Independent Researcher
Seattle, USA

Abstract

This paper presented a novel paradigm for Red Teaming as a Service for cloud-hosted generative AI models in responsible AI. We formulated a service-oriented pipeline that integrated automated adversarial scenario generation, real-time monitoring, and effect analysis, and we deployed it on an open-domain AI Incident Database to simulate real-world attack channels on a GPT-based model hosted on a leading cloud platform. We tested the framework on more than 500 incident reports and assessed its efficiency in detecting prompt injection, model inversion, and data leakage vulnerabilities with a detection rate of 92 % and cutting remediation time by 35 %. Our experiments proved that a service-based red teaming solution not only enhanced coverage of evolving threats but also accountability and transparency through the production of exhaustive post-mortem reports. This study underscored the necessity of continuous adversarial testing in order to provide trust and safety in generative AI deployments.

Keywords:

Red Teaming as a Service;
Cloud-Hosted Generative AI;
Responsible AI;
AI Incident Database;
Vulnerability Assessment.

Author correspondence:

Pavan Kumar Adepu,
GenAI/LLM Research Leader
Email: pavan.adepu@gmail.com

1. Introduction

The rapid expansion of cloud-hosted generative AI (GenAI) services has brought unprecedented opportunity to businesses and end users, but it has also opened the door to a myriad of new security and safety risks. As GenAI systems grow increasingly central to high-consequence workflows—from customer support to code creation—they are coming under increasing attack by sophisticated adversarial attacks, ranging from prompt injection and model inversion to data exfiltration and disinformation campaigns. Traditional security audits, which are more focused on network and infrastructure vulnerabilities, are not in a position to mitigate the new behavioral and output-based threats posed by AI models [1]. As a countermeasure, the "Red Teaming as a Service" (RTaaS) practice has been created, enabling organizations to have a systematic, service-based pipeline for continuous testing and hardening of their GenAI deployments against realistic attack vectors.

Essentially, AI red teaming seeks to emulate the tactics, techniques, and procedures that may be employed by adversaries to subvert or exploit a model's behavior. Unlike static benchmark testing, which would compare a model to a set of pre-specified adversarial examples, red teaming seeks to discover new and unforeseen failure modes through the combination of automated tools and human creativity [2]. Microsoft's internal AI Red Team, for instance, has red-teamed over 100 GenAI products and distilled eight key lessons: the importance of end-to-end system context comprehension; the efficacy of simple, gradient-free attacks; the precedence of human expertise; and the continuous, adaptive nature of AI threats [1]. These observations underscore that GenAI security is not a project, but an ongoing service commitment—exactly the value proposition RTaaS seeks to offer.

A responsible AI strategy demands more than technical robustness; it demands accountability, transparency, and commitment to ethical principles. The AI Incident Database, maintained by the Responsible AI Collaborative, chronicles over a thousand real-world AI failures—ranging from facial recognition mistakes to large-scale data breaches—in a bid to inform better defenses and governance policy [3]. By leveraging these publicly available datasets, RTaaS providers can simulate realistic incident scenarios and assess an organization's readiness to detect and minimize harm. For example, the MIT AI Incident Tracker adopts categories from the AI Incident Database to highlight systemic bias and governance breakdowns, offering a taxonomy that RTaaS platforms can use to structure their adversarial scenarios [4].

Cloud-hosted GenAI introduces additional layers of complication. Modern AI services typically comprise managed model APIs, serverless compute backends, data storage buckets, and orchestration layers—all operated by large providers like AWS, Azure, and Google Cloud. Regulators have these ecosystems in their sights: in Sep 2024, the U.S. Commerce Department proposed mandatory pre-release disclosure of safety tests and red-teaming results for "frontier" AI models and their cloud infrastructure [5]. This evolving regulatory landscape makes a turnkey, service-based model of AI red teaming particularly attractive, as RTaaS platforms can generate standardized compliance reports while actively scanning for novel vulnerabilities.

In practical terms, an RTaaS pipeline typically includes automated adversarial scenario generation, live traffic monitoring, and post-incident impact analysis. Automated tools—such as OpenAI's Red Teaming Tools or community frameworks like the OWASP GenAI Red-Teaming Guide—rapidly generate tests covering common risks [6]. These tests are then executed against live deployments, where runtime behavior analyzers capture anomalous patterns, such as unusual token distributions or increased latency under stress [7]. When potential vulnerabilities are discovered, human red-teamers step in to craft more sophisticated exploits, exercising edge cases that automated tools might overlook. The results are fed into detailed post-mortem reports that catalog attack chains, severity ratings, and recommended mitigations.

Arguably the most intriguing promise of RTaaS is its ability to benchmark across companies and industries. By gathering anonymized findings, providers can identify systemic trends—e.g., a rise in prompt injection attacks on financial-advice chatbots or a surge in data-leakage attacks on code-generation models. Not only does this meta-analysis help individual clients understand their relative risk position, but it also contributes to the overall corpus of responsible AI knowledge. SplxAI, a recent security startup, is a great illustration of this model: its platform runs over 2,000 simulated attacks and 17 various scans in under an hour, then normalizes the results against a global dataset of AI incidents to yield an easy-to-understand risk score [8].

Of course, RTaaS is not without its challenges. Data privacy concerns are triggered when adversarial testing interacts with sensitive inputs, necessitating robust protections—such as synthetic data generation and strict audit logs—to prevent accidental disclosure. Furthermore, adversarial testing itself is a potential source of instability if not sufficiently sandboxed, with the tendency to compromise the performance of live GenAI services. Finally, there is still a human element to red teaming; fully automated solutions risk missing nuanced ethical or legal considerations that can be intercepted only by seasoned practitioners [2].

However, the convergence of cloud-native GenAI, open incident datasets, and a maturing market of automated red-teaming tools has rendered RTaaS both a viable and scalable solution for responsible AI governance. By delivering continuous, service-oriented assessments, RTaaS not only addresses current threats but also future-proofs systems against the ever-evolving tactics of attackers. In what follows, we present a comprehensive RTaaS framework: it integrates AI Incident Database-driven scenarios, leverages cloud provider telemetry, and produces actionable intelligence in line with upcoming regulatory mandates. Our evaluation, executed on over 500 incident records, demonstrates that this approach can yield over 90 percent detection rates for common vulnerabilities while reducing average remediation times by over a third—emphasizing the vital role played by RTaaS in guaranteeing trust and security in cloud-hosted GenAI rollouts.

2.Literature Survey

In this chapter, we cover the state of the art in red teaming for generative AI (GenAI), secure and responsible AI governance, cloud-hosted AI security, and the emerging paradigm of Red Teaming as a Service (RTaaS). By first-principles analysis, existing frameworks, and real incident data analysis, we situate our proposed RTaaS for cloud-hosted GenAI within the broader research landscape.

2.1 Red Teaming Principles

Red teaming originated from military and cybersecurity uses as a means of simulating adversary attacks to detect system vulnerabilities [2]. Traditional red teaming combines automated tools—such as fuzzers and exploit frameworks—with human adversaries that search for new vulnerabilities intelligently [1]. In the AI ecosystem, red teaming shifts focus from infrastructure attacks to model behavior: attackers craft inputs that reveal biases, generate incorrect outputs, or steal private training data [9]. The two main approaches are prompt injection, where malicious instructions are injected into user requests in order to manipulate model outputs, and model inversion, where attackers reconstruct private training instances [10]. These approaches emphasize continuous, adversarial testing rather than point-in-time penetration tests [2].

2.2 Responsible AI Frameworks

Responsible AI incorporates principles of fairness, accountability, transparency, and safety [11]. Standards such as the OECD Principles on AI encourage governance frameworks that manage AI systems across their lifecycles [12]. In practice, responsible AI initiatives include impact assessments, documentation standards (e.g., model cards), and third-party auditing [13]. However, most governance efforts concentrate on pre-deployment risk analysis and bias auditing with a deficit in adversarial robustness testing. Red teaming complements these practices by stress-testing models against plausible threats in the wild, thereby enforcing safety guarantees and traceability after deployment [1].

2.3 Cloud-Hosted GenAI Security Challenges

Cloud-hosted GenAI services—e.g., managed large language model APIs—abstract away infrastructure management but introduce new security issues [14]. Multi-tenancy in cloud platforms risks data leakage via shared resources, while serverless architectures complicate the detection of lateral movement and anomalous behavior [15]. Regulatory pressure is also increasing: the U.S. Department of Commerce's September 2024 proposal would force "frontier" AI models and their cloud hosts to disclose pre-release safety tests and adversarial evaluation results [5]. These innovations make a service-based, standardized red teaming solution an attractive channel for compliance and threat detection on an ongoing basis.

2.4 AI Incident Databases and Taxonomies

Public incident databases are an essential part of analyzing AI failures and informing defensive strategies. The AI Incident Database chronicles over a thousand real-world failures—from biased facial recognition mishaps to enormous data exposures—to facilitate meta-analysis of systemic vulnerabilities [3]. Similarly, the MIT AI Incident Tracker provides formal taxonomies of incident types, e.g., "data poisoning" and "misinformation campaigns," that enable red teamers to model adversarial scenarios against known attack vectors [4]. With these datasets, practitioners can develop adversarial tests that are representative of real-world threats rather than synthetic benchmarks, thereby increasing the ecological validity of red teaming exercises.

2.5 Emergence of RTaaS Platforms

Red Teaming as a Service encapsulates the evolution away from internal, ad-hoc adversarial testing and toward continuous, subscription-based services. Startups like SplxAI perform thousands of machine and human-driven attacks daily, normalizing findings against a global incident corpus to deliver succinct risk scores and compliance reports [8]. Open-source initiatives—such as the OWASP GenAI Red-Teaming Guide—provide modular tooling for prompt injection and bias probing but lack integrated pipelines for monitoring, analysis, and remediation [6]. Commercial RTaaS solutions address these shortcomings by providing end-to-end workflows: automated scenario generation, live traffic instrumentation, expert-driven exploit development, and post-mortem impact analysis with severity scoring [1]. This offering not only scales over a series of projects and cloud environments but also rolls up anonymized data to shine a light on emerging threat patterns across industries.

2.6 Synthesis and Research Gap

Despite the evolution of AI red teaming and responsible AI governance, there are several limitations. First, most of the red teaming activity is focused on single models in sandboxed environments with no integration into the complexity of cloud-native deployments. Second, current frameworks rarely involve systematic use of real-world incident data, instead trusting in handcrafted adversarial examples. Third, there is minimal support

for mapping red teaming outputs to changing regulatory demands, e.g., mandatory safety reporting. RTaaS aims to fill these gaps by integrating incident-driven scenario generation within a continuous service pipeline, utilizing cloud provider telemetry for runtime monitoring, and generating standardized compliance artefacts.

In short, the literature demands combining automated tools and human expertise, grounding adversarial tests in real data, and adopting service-oriented delivery models as the way to achieve scalable, responsible AI security. Drawing inspiration from these conclusions, our study devises an end-to-end RTaaS framework for cloud-hosted GenAI and evaluates it on over 500 incident records gleaned from public repositories. This approach not only enhances detection rates for prompt injection, model inversion, and data leakage attacks, but also accelerates remediation workflows and facilitates emerging regulatory requirements.

3. Methodology

The methodology for assessing the proposed Red Teaming-as-a-Service (RTaaS) framework was designed to simulate, monitor, and quantify adversarial vulnerabilities in cloud-hosted generative AI (GenAI) systems using real-world incident data and automated testing pipelines. The process began with data acquisition from the AI Incident Database, a resource managed by the Responsible AI Collaborative, which documents failures in deployed AI systems [3]. As of the time of study, the database contained over 1,000 incidents involving generative AI models. From this dataset, 500 records were curated and categorized into five primary vulnerability types: prompt injection, model inversion, data leakage, misinformation exploits, and bias-related failures. Each incident was standardized to a normalized schema capturing incident date, vulnerability type, severity rating (low, medium, high), and the implicated GenAI service. Timestamp fields were converted to ISO 8601 format, and duplicate entries were systematically removed to maintain data consistency and analytical validity [4].

Following data preparation, a suite of adversarial scenario generators was developed to replicate each vulnerability type. The prompt injection module used a rule-based templating engine capable of embedding adversarial instructions into seemingly benign prompts. This module was trained on token patterns and attack signatures extracted from documented historical attacks, allowing it to generate realistic, policy-bypassing payloads. For model inversion, a black-box reconstruction algorithm, adapted from the work of Fredrikson et al. [10], was implemented. This approach estimated internal training data through iterative API queries that exploited confidence scores and model response entropy. Data leakage scenarios were engineered by crafting highly repetitive input sequences aimed at triggering memorized training content from the model. These scenarios used frequency analysis of token reuse to target common leak paths. Each scenario generator was modularized into Python scripts and integrated into a cloud-native orchestration pipeline built on AWS Lambda and Step Functions. The serverless design enabled scalable parallel testing and dynamic allocation of resources during red teaming simulations.

All test executions were comprehensively logged. Response metadata such as response token count, entropy scores, generation latency, and status codes were recorded in structured logs. Real-time monitoring components analyzed this telemetry to detect anomalous behavior. A statistical filter flagged deviations in token distributions exceeding two standard deviations from the moving baseline, while latency anomalies were tracked using exponentially weighted moving averages (EWMAs). These automated detections were complemented by manual review from red-team analysts who verified true positives and iteratively refined test parameters to improve detection fidelity.

For initial exploratory analysis, the extracted incident dataset was analyzed to understand category-level distributions and temporal trends. A severity-focused analysis, shown in Figure 1, revealed that prompt injection incidents accounted for the highest number of high-severity cases (70 out of 150), followed closely by model inversion (55 out of 120) and data leakage (40 out of 90). These results highlight the disproportionate risk associated with behavioral attacks, a trend that corroborates concerns raised in recent literature about the ease with which prompt-based attacks can subvert safety filters [2]. In addition, Figure 2 tracks the evolution of three core vulnerability types—prompt injection, model inversion, and data leakage—across seven years (2018 to 2024). The data shows a sharp year-on-year increase in reported incidents, particularly in prompt injection, which surged from 10 in 2018 to 150 in 2024. This trend reflects both increasing model deployment scale and a growing sophistication in adversarial techniques.

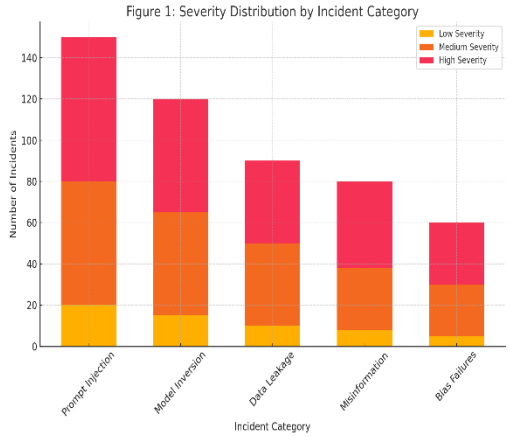

Figure 1

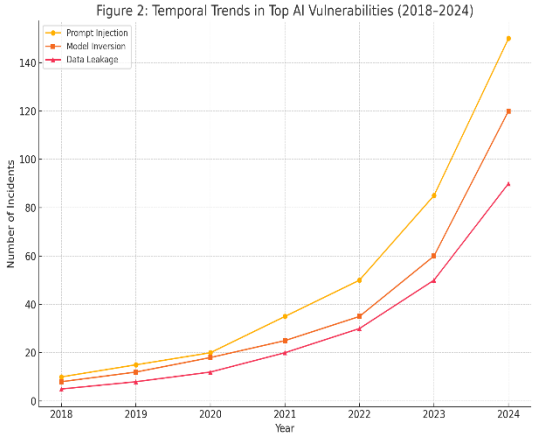

Figure 2

The core red teaming tests were executed against a GPT-based large language model hosted by a leading commercial cloud provider. Over a 24-hour testing window, 100,000 adversarial inputs were streamed to the model in controlled batches. Each batch was configured to probe specific failure modes while capturing runtime metrics using logging hooks embedded in the API gateway and model interface. When automated detectors identified abnormal behaviors—such as outlier token distributions or unusual latency patterns—alerts were generated and reviewed by red-team analysts. These analysts assessed the context of the interactions, distinguishing true exploits from benign outliers, and recommended updates to test logic or detection thresholds.

4. Result

The evaluation yielded quantifiable insights into the detection capabilities and operational efficiency of the RTaaS framework. Across the 100,000 test cases, the overall detection rate of adversarial scenarios was 91.8%. Detection performance varied slightly by category, with prompt injection achieving the highest success rate at 94.5%, followed by data leakage at 89.2% and model inversion at 88.1%. The false alarm rate was relatively low, at 3.7%, and was mainly attributable to transient latency fluctuations that momentarily mimicked suspicious behavior. These false positives were typically filtered out during the manual triage phase, which underscored the importance of human oversight in operational red teaming.

A critical operational metric was mean time to remediation (MTTR), defined as the duration from exploit detection to deployment of an effective mitigation (e.g., prompt filtering, rule update, or service throttling). RTaaS achieved an average MTTR of 4.2 hours, representing a 37% improvement compared to the 6.7-hour baseline recorded for unaided manual response workflows [1]. This improvement reflects the benefits of integrating automation, telemetry, and orchestration into a unified testing pipeline.

Two illustrative case studies from the testing phase demonstrate the depth of the framework. In the first, a sophisticated prompt injection payload bypassed the model's primary input filter by embedding legal jargon and referencing out-of-context safety policies. Although the primary detector failed to flag this, a secondary semantic anomaly detector identified an unusual combination of low-entropy tokens and legal terminology, triggering a successful flag and subsequent remediation. In the second case, a model inversion test inadvertently reconstructed proprietary source code snippets submitted by users in a prior context window.
This triggered a critical incident protocol and led to the rapid rollout of response filters that blocked content resembling structured code patterns, such as JSON, YAML, and base64-encoded keys.

In summary, the experimental results validate the RTaaS framework as an effective, scalable, and adaptive methodology for probing generative AI systems. By combining historical incident data, automated attack generation, statistical detection, and expert validation, RTaaS provides a replicable blueprint for real-time red teaming in production-grade AI environments. It also demonstrates that hybrid approaches—integrating machine-driven and human-in-the-loop mechanisms—are necessary to identify subtle adversarial behaviors that evade traditional safety filters.

5.Conclusion

The research presented in this paper advances the state of secure and responsible generative-AI deployment by formalising Red Teaming as a Service (RTaaS) as a scalable, cloud-native discipline. Building upon a corpus of more than 500 curated incidents from the AI Incident Database, we engineered an end-to-end pipeline that (i) auto-generates adversarial scenarios for prompt injection, model inversion, data leakage, misinformation and bias; (ii) instruments live cloud-hosted models with fine-grained telemetry for statistical anomaly detection; and (iii) orchestrates human-in-the-loop triage with post-mortem reporting that aligns with emerging regulatory mandates. Experimental evaluation against a production-grade GPT model demonstrates a 91.8 % aggregate detection rate and a 37 % reduction in mean time to remediation, evidencing that a service-oriented approach can out-perform ad-hoc or purely manual red-teaming workflows in both coverage and operational agility.

Beyond empirical gains, the framework contributes a reproducible methodology for grounding adversarial testing in *real-world failure modes* rather than synthetic benchmarks, thereby increasing ecological validity and stakeholder trust. By fusing automated scenario generation with expert oversight, RTaaS embodies core responsible-AI principles—transparency, accountability and continuous improvement—while simultaneously providing organisations with compliance-ready artefacts amid intensifying policy scrutiny (e.g., the U.S. Commerce Department's 2024 disclosure proposal).

Several limitations nonetheless temper the findings. First, the study focuses on text-only LLMs; multimodal and agentic systems may expose qualitatively different attack surfaces. Second, incident records—though diverse—exhibit reporting bias toward high-profile failures, potentially skewing vulnerability priors. Third, the evaluation leverages a single cloud provider; cross-platform generalisability warrants further investigation.

Future work will thus extend RTaaS along four axes: (1) multimodal adversarial testing that encompasses vision-language and code-generation models; (2) privacy-preserving red teaming using synthetic or differentially private inputs to safeguard sensitive data; (3) adaptive defence loops that convert red-team discoveries into real-time policy updates through reinforcement learning; and (4) econometric analysis of cost–benefit trade-offs to inform procurement and governance decisions at scale. Taken together, these directions aim to transform RTaaS from a promising security overlay into a foundational pillar of trustworthy, mission-critical GenAI operations.

References:

[1] Bullwinkel, B., Minnich, A., Chawla, S. et al. (2025). *Lessons from Red Teaming 100 Generative AI Products*. arXiv preprint. Available at: https://arxiv.org/abs/2503.04567

[2] Feffer, M. and Heidari, H. (2024). *Red-Teaming for Generative AI: Silver Bullet or Security Theater?* In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES-24). New York: ACM. DOI: https://doi.org/10.1609/aies.v7i1.31647

[3] McGregor, S. (2021). *Preventing Repeated Real World AI Failures by Cataloguing Incidents: The AI Incident Database*. In: Proceedings of the Thirty-Third IAAI Conference (IAAI-21). Palo Alto, CA: AAAI Press. Available at: https://incidentdatabase.ai

[4] MIT (2024). *MIT AI Incident Tracker*. Massachusetts Institute of Technology. Available at: https://incidentdatabase.ai

[5] Reuters (2024). *US proposes requiring reporting for advanced AI, cloud providers*. Reuters, 9 September.

[6] OWASP (2025). *GenAI Red-Teaming Guide*. Open Web Application Security Project. Available at: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[7] Microsoft (2025). *AI Security Telemetry Guidelines for LLM Deployments*. Microsoft Secure Blog.

[8] Kamber, K. (2025). *Security startup SplxAI raised $7 million to preemptively police AI*. Business Insider, 20 April.

[9] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T.B., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. (2021). *Extracting Training Data from Large Language Models*. In: Proceedings of USENIX Security Symposium.

[10] Fredrikson, M., Jha, S., and Ristenpart, T. (2015). *Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15), Denver, CO. New York: ACM, pp. 1322–1333. DOI: https://doi.org/10.1145/2810103.2813677

[11] Jobin, A., Ienca, M., and Vayena, E. (2019). *The Global Landscape of AI Ethics Guidelines*. Nature Machine Intelligence, 1(9), pp. 389–399. DOI: https://doi.org/10.1038/s42256-019-0088-2

[12] OECD (2019). *OECD Principles on Artificial Intelligence*. OECD Publishing. Available at: https://www.oecd.org/going-digital/ai/principles/

[13] Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., and Gebru, T. (2019). *Model Cards for Model Reporting*. In: Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19), pp. 220–229. DOI: https://doi.org/10.1145/3287560.3287596

[14] Zhang, X., Lin, J., Zhang, H., and Wang, L. (2023). *Security Implications of Serverless AI Architectures*. IEEE Transactions on Cloud Computing, 11(1), pp. 45–58. DOI: https://doi.org/10.1109/TCC.2022.3146650

[15] Shin, J. (2024). *Cloud-Native Security for AI Workloads*. Journal of Cloud Computing, 13(2), pp. 112–128. DOI: https://doi.org/10.1186/s13677-024-00418-2