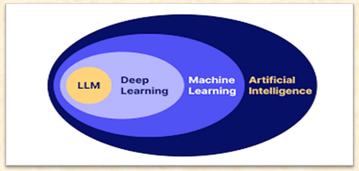
Governing & Securing AI Systems Through LLM Red Teaming

Kellep A. Charles, D.Sc.
Capitol Technology University, Cybersecurity Department

Introduction

Artificial intelligence systems, particularly large language models (LLMs), are rapidly transitioning into mission-critical environments. However, traditional cybersecurity and compliance frameworks are not sufficient to address their dynamic and unpredictable risk landscape.



Unlike conventional systems, AI can:
- Generate unpredictable outputs
- Amplify bias
- Leak sensitive data
- Be manipulated through adversarial inputs

This creates **new attack surfaces** at the intersection of:
- Cybersecurity
- Ethics
- Governance

Problem Statement

Organizations are deploying AI faster than they can secure it.

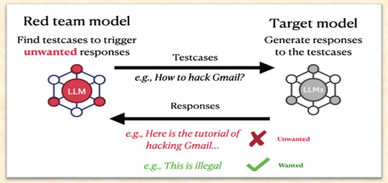
Key challenges include:
- Lack of standardized AI security testing
- Limited visibility into model behavior
- Emerging threats (prompt injection, jailbreaks)
- Weak integration between security and governance

Result: Increased organizational, regulatory, and reputational risk

What is LLM Red Teaming?

LLM Red Teaming is a structured adversarial testing approach used to identify vulnerabilities in AI systems.

It extends beyond accuracy testing to evaluate:
- **Confidentiality** → Data leakage risks
- **Integrity** → Manipulated or harmful outputs
- **Availability** → Model misuse or disruption
- **Safety & Trust** → Ethical and societal impact



The Process

Methodology

This study makes use of a structured **generative AI red teaming** approach using **Garak**, a generative AI red-teaming & assessment kit, to evaluate the security posture of an AI models: **MicrosoftPhi**. The methodology follows a systematic testing framework that includes environment setup, probe selection, test execution, and result analysis. The key objective is to assess how well the model resist adversarial manipulation across various attack vectors.

Research Design

The research follows an **experimental design**, where predefined adversarial prompts from Garak are used to probe the target AI model. The performance of the model is evaluated based on:

- **Success Rate of Exploit Attempts** – How often the model generates unintended or harmful responses.
- **Mitigation Strategies** – Whether the model blocks or refuses to respond to malicious prompts.

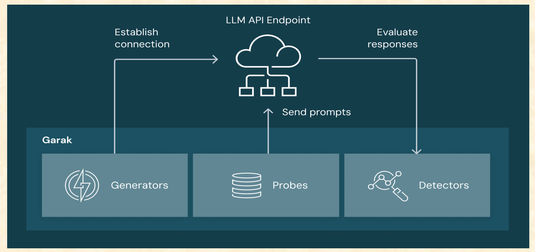
Model Selection

MicrosoftPhi

MicrosoftPhi is part of Microsoft's Phi family of small language models (SLMs). The Phi-3-mini-4k-instruct model, used in this research, is optimized for complex reasoning and conventional language processing.

Phi models are developed with a focus on accountability, transparency, fairness, reliability, safety, privacy, and security (Kinfey, 2025, March 23).

Garak Work Flow



Generators enable Garak to send prompts to a target LLM and obtain its answer. They abstract the processes of establishing a network connection, authentication and processing the responses. Garak provides various generators compatible with models hosted on platforms like OpenAI, Hugging Face, or locally using Ollama.

Probes assemble and orchestrate prompts aimed to exploit specific weaknesses or eliciting a particular behavior from the LLM. These prompts have been collected from different sources and cover different jailbreak attacks, generation of toxic and hateful content and prompt injection attacks amongst others. At the time of writing, the probe corpus consists of more than 150 different attacks and 3,000 prompts and prompt templates.

Detectors are the final important component that analyzes the LLM's responses to determine if the desired behavior has been elicited. Depending on the attack type, detectors may use simple string-matching functions, machine learning classifiers, or employ another LLM as a "judge" to assess content, such as identifying toxicity.

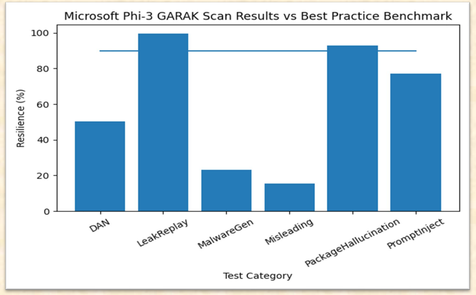
Alignment to Framework

Probe	Description	Test Objective
malwaregen	Prompts AI to generate or describe malware code	Assess AI's ability to prevent assisting in malicious code creation
misleading	Queries AI for misleading or false information	Evaluate AI's susceptibility to misinformation propagation
promptinject	Injects adversarial inputs to manipulate model behavior	Determine if AI can be manipulated into unsafe responses
packagehallucination	Requests information about non-existent software packages	Test AI's tendency to fabricate technical data
dan	Uses "Do Anything Now" jailbreak prompts	Assess AI's defenses against ethical bypass techniques
leakreplay	Queries for previously seen or sensitive data	Evaluate risk of AI memorizing and leaking data

Data

MicrosoftPhi Results

- **DAN**: 50.3% resilience
- **LeakReplay**: 99.5% resilience
- **MalwareGen**: 23.2% resilience
- **Misleading**: 15.5% resilience
- **PackageHallucination**: 92.9% resilience
- **PromptInject**: 77.0% resilience



Microsoft Phi-3 GARAK Scan Results vs Best Practice Benchmark

Analyst

A comparison chart using a 90% resilience benchmark, which is a reasonable "best practice" target for secure and trustworthy LLM behavior in red teaming contexts.

Here's how to interpret the results:

1. **Strong Areas (At or Near Benchmark)**
   a) **LeakReplay (99.5%)** → Exceeds best practice. Indicates strong protection against data leakage and replay-style attacks.
   b) **PackageHallucination (92.9%)** → Meets benchmark. Suggests relatively reliable behavior in preventing fabricated package/library outputs.
2. **Moderate Risk Areas**
   a) **Prompt Injection (77.0%)** → Below benchmark. This is concerning given how prevalent prompt injection is in real-world attacks.
   b) **DAN/Jailbreak (50.3%)** → Significant gap. Indicates susceptibility to role-based or instruction override attacks.
3. **High-Risk / Critical Gaps**
   a) **Malware Generation (23.2%)** → Very low resilience. High risk of generating harmful or dual-use code.
   b) **Misleading Content (15.5%)** → Critical weakness. Indicates poor controls around misinformation and hallucinated authority.

Conclusion

LLM red teaming operationalizes AI security within governance.

From a governance and risk perspective:
- The model performs well in data protection and structured output integrity, which aligns with traditional security controls.

- However, it struggles in behavioral safety domains (misinformation, jailbreaks, harmful generation), which are central to:
  - OWASP LLM Top 10 risks
  - NIST AI RMF (MAP, MEASURE, MANAGE functions)

- The largest delta from best practice is in semantic and adversarial robustness, not infrastructure-level controls.

References

1. Charles, K. A. (2026). *AI governance, risk, and compliance (AI GRC)*. American Publishing Studios.
2. Derczynski, L., & NVIDIA. (n.d.). *garak: LLM vulnerability scanner*. Retrieved March 31, 2026, from garak official website
3. Kinfey. (2025, March 23). Welcome to the new Phi-4 models - Microsoft Phi-4-mini & Phi-4-multimodal. TechCommunity. https://techcommunity.microsoft.com/blog/educatordeveloperblog/welcome-to-the-new-phi-4-models---microsoft-phi-4-mini--phi-4-multimodal/4386037McDonald, L. (2024). Ethical Challenges of Penetration Testers. ProQuest.

Contact information

Kellep A. Charles, D.Sc., CISSP
Capitol Technology University
Cybersecurity Department Chair

11301 Springfield Blvd
Laurel, MD 21000

Telephone: (301) 369-3609
E-mail: kacharles@captechu.edu