Garak: LLM Vulnerability Scanner

Comprehensive security testing for large language models and AI systems

Understanding LLM Vulnerabilities

LLM Security Failures

Investigation of failure modes like hallucinations, bias amplification, and inconsistent outputs that make LLMs inherently insecure

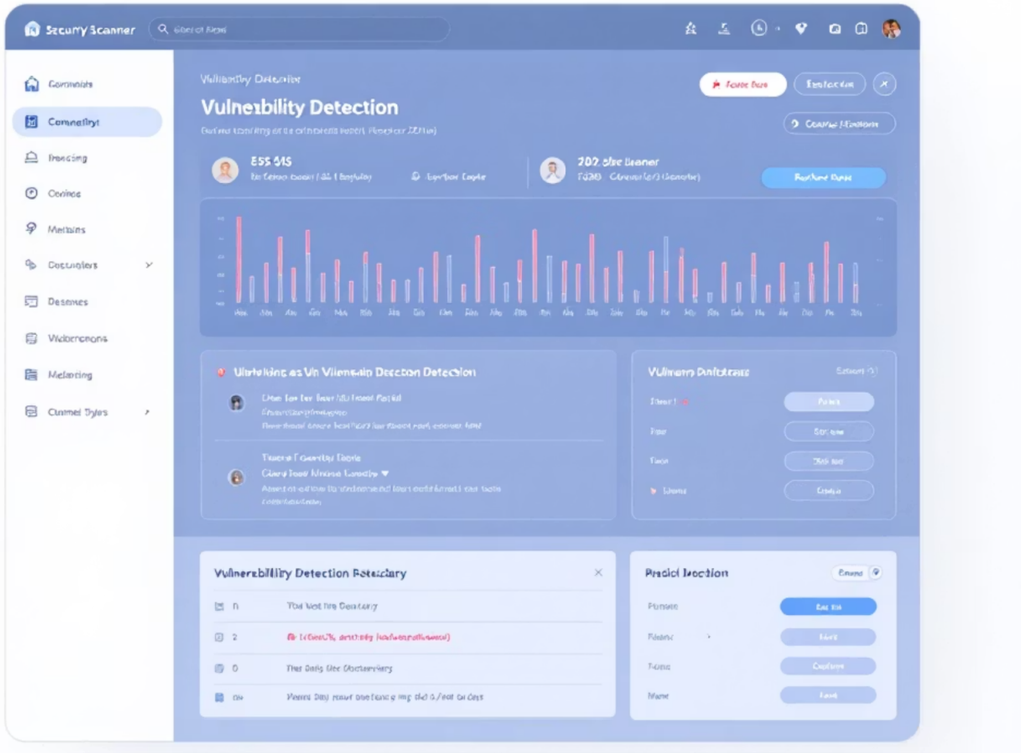
Software Stack Risks

Vulnerabilities in underlying frameworks (PyTorch, ONNX, CUDA) and operating systems that support LLM deployment

Human Interaction Threats

Prompt injections, jailbreak attempts, and adversarial inputs that exploit LLM behavior through malicious user interaction

What Garak Does



Comprehensive LLM Security Testing

- Identifies vulnerabilities in foundation models and LLM-based applications
- Tests dozens of failure modes using specialized probes and challenging prompts
- Generates detailed reports with exact prompts, goals, and responses for each finding

Key Features

LLM-Specific Security

Focuses on prompt injection, jailbreaks, guardrail bypass, text replay, and other LLM-inherent risks

Automated Scanning

Unsupervised probes run systematically across models with appropriate detectors managing the process

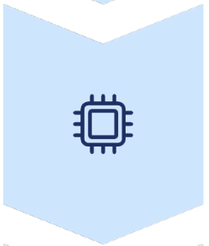
Wide LLM Support

Compatible with OpenAI, Hugging Face, Cohere, and custom Python integrations

Garak Architecture

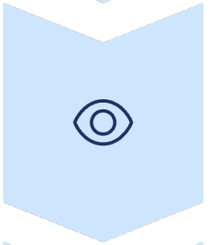
Vulnerability Probes

Specialized tests targeting specific failure modes and vulnerabilities in LLMs



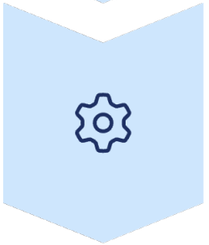
Generators

Interfaces that wrap different methods for communicating with dialogue systems and LLMs



Detectors

Automated failure detection using keyword matching and ML classifiers to evaluate outputs



Harness

Orchestrates the entire workflow, connecting generators and probes to conduct vulnerability scans