**1) Garak (NVIDIA)**

* GitHub: <https://github.com/NVIDIA/garak>
* Docs: <https://docs.nvidia.com/nemo/guardrails/latest/evaluation/llm-vulnerability-scanning.html>

Purpose: Garak is a red-teaming toolkit designed to evaluate LLMs against multiple categories of prompt-based and behavioral attacks. It simulates attack scenarios like prompt injection, malware generation, data leakage, and model inversion.

Inputs:

* Model type (OpenAI, HuggingFace, etc.)
* API keys (e.g., OPENAI\_API\_KEY)
* Probe list (promptinject, xss, dan, etc.)
* YAML config file (optional)

Outputs:

* Protection metrics by module (HTML, CSV)
* Matrix of success/fail for each probe (e.g., dan, encoding, xss)
* Side-by-side probe and model response logs

How to Use:

1. Clone the repo and install dependencies
2. Set API keys via environment variables (e.g., OPENAI\_API\_KEY)
3. Choose model and probe: garak model openai:gpt-4 probes xss
4. Export report: output html

Relevance:

* Matches Agentic AI Threat Classifications: Yes
* Prompt logs, exploit traces, jailbreak detection: Yes
* Confidence analysis over configs (bare\_llm, with\_gi\_dr, etc.): Yes

**2) Gandalf (Protect AI)**

Purpose: Gandalf focuses specifically on detecting prompt injection vulnerabilities. It sends crafted payloads to an LLM endpoint and logs if the prompt is executed, blocked, or leaked internal system logic.

Inputs:

* API endpoint for LLM
* YAML configuration

Outputs:

* JSON logs of prompt ➝ response ➝ status
* HTML summaries (optional)
* Flagged injections with timestamps

How to Use:

1. Install: pip install gandalf
2. Create a config file with the endpoint URL
3. Run: gandalf config config.yaml

Relevance:

* Matches Threat Categories: Prompt Injection
* Agent Logs: Partial
* Usable for CI/CD scanning

**3) Rebuff**

Purpose: Rebuff acts like a firewall for LLMs, sitting between the user and the model API. It detects and blocks risky input and monitors for sensitive output patterns.

Inputs:

* LLM endpoint wrapped by Rebuff
* Policy YAML configuration

Outputs:

* Real-time alert logs
* Policy violation traces
* Jailbreak attempt logs

How to Use:

1. Deploy Rebuff as a proxy (middleware)
2. Define guard rules in YAML
3. Logs available via API or dashboard

Relevance:

* Real-time filtering aligns with Safety Guardrails
* Detects jailbreaks, XSS, personal data exposure

**4) Llama Guard (Meta)**

* Hugging Face: <https://huggingface.co/meta-llama/LlamaGuard-7b>

Purpose: Llama Guard is a classification model trained to moderate LLM inputs and outputs. It detects harmful, biased, or policy-violating content.

Inputs:

* Prompt or response text string

Outputs:

* Label: Safe / Unsafe
* Risk Category: Violence, Hate, Misinformation, etc.
* Output format: JSON

How to Use:

1. Load via Hugging Face Transformers or API
2. Run classifier: predict(text)

Relevance:

* Helps in Bias/Fairness Auditing
* No logs or trace reports, but good as a filter

**5) PromptBench**

Purpose: PromptBench provides a benchmark to evaluate jailbreak resistance across models using standardized test prompts.

Inputs:

* Model name / API endpoint
* Test prompt set

Outputs:

* Jailbreak success rates
* Response flag categories
* Model-wise comparison

How to Use:

1. Clone repo and set up benchmark environment
2. Run CLI: python run\_benchmark.py ,model gpt4

Relevance:

* Useful to test new LLMs for vulnerability resistance
* Visual graphs, CSV results available

**6) AdvPromptGuard**

* Paper: <https://arxiv.org/abs/2307.15043>

Purpose: This tool generates stealthy prompt injection attacks to test an LLM's defense mechanisms.

Inputs:

* Prompt dataset
* Target LLM model or endpoint

Outputs:

* Logs of bypassed filters
* Injection success
* Prompt-response comparison

How to Use:

1. Install via pip
2. Provide target model and prompt set
3. Run scan

Relevance:

* Focused on obfuscated prompts
* Works well with Garak

**7) OpenPromptEval**

Purpose: This framework evaluates LLMs on bias, toxicity, hallucination and fairness using curated test prompts.

Inputs:

* Task to evaluate (bias, hallucination, etc.)
* LLM model name or API

Outputs:

* Bias graphs
* Fairness scores
* Class distribution

How to Use:

1. Clone repo
2. Run CLI: python eval.py ,task bias ,model mistral

Relevance:

* Great for fairness and explainability assessment
* Results used for compliance & safety auditing

**Summary Comparison Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool** | **Focus Area** | **Outputs** | **Integration Style** | **Threat Vectors Detected** |
| Garak | Multi-vector red teaming | HTML, CSV, logs | CLI / Batch | Prompt injection, xss, malware, etc. |
| Gandalf | Prompt injection | JSON, HTML logs | CLI / REST | Prompt hijacking |
| Rebuff | Input/output firewall | Real-time alerts/logs | Proxy / API middleware | Jailbreaks, PII, data leaks |
| Llama Guard | Moderation classifier | JSON labels per prompt | API / HuggingFace | Toxicity, misinformation |
| PromptBench | Jailbreak benchmarking | Success rate, graphs | CLI | Jailbreak resistance |
| AdvPromptGuard | Obfuscated injection | Attack logs, success rates | CLI | Evasion & stealth prompt injection |
| OpenPromptEval | Bias & hallucination | Scores, graphs | CLI | Bias, hallucination, fairness |

**Best LLM Vulnerability Tool – Depends on our Needs:**

| **Use Case** | **Recommended Tool** | **Why It’s Best** |
| --- | --- | --- |
| Broad LLM Red Teaming (multiple threat types) | Garak (NVIDIA) | It simulates 15+ types of attacks (prompt injection, XSS, data leakage, jailbreaks) and gives HTML/CSV reports. Ideal for coverage across categories. |
| Prompt Injection Testing | Gandalf (Protect AI) | Specialized in detecting prompt injection issues with simple YAML config. Easy to plug into pipelines. |
| Real-Time LLM Protection | Rebuff | Works like a firewall—detects & blocks threats in real time. Great for production deployments. |
| Input/Output Moderation | Llama Guard | Simple and lightweight. Helps classify unsafe prompts or outputs (hate speech, toxicity, etc.). |
| Jailbreak Resistance Benchmarking | PromptBench | Great for scoring LLMs against standard jailbreak attacks. Outputs success rates, works well for comparing models. |
| Advanced Obfuscated Attack Simulation | AdvPromptGuard | Sends stealthy, adversarial prompts. Useful for advanced red teaming. |
| Bias & Fairness Evaluation | OpenPromptEval | Focuses on bias, fairness, hallucinations. Produces readable fairness scores and impact visualizations. |

Garak is the best starting point It covers multiple probe types, outputs match many of your categories (like threat classification, jailbreak detection, explainability, supply chain risk) But,

Gandalf and Rebuff can be used as plug-in filters or layers on top of Garak.

OpenPromptEval is perfect if you want to add explainability/bias modules.