Synack

DATA SHEET

Synack's AI/LLM Pentesting

Synack's pentesting has evolved to test deployed Large Language Models (LLMs) within your attack surface, utilizing the skills of the Synack Red Team (SRT). The SRT consists of over 1,500 global, vetted researchers with a diversity of expertise.

There are many ways you may be using LLMs today. Examples of LLM use cases and outcomes include:

- Improved search results for internal and external tools
- Chatbots to help customers navigate your products and services
- Automated customer service experiences
- Language translation
- Classification and sorting of ingested content

Synack's AI/LLM Pentesting Methodology

Despite the wide range of use cases for LLMs, the Open Web Application Security Project (OWASP) has compiled 10 common and critical vulnerabilities that span potential abuses of an LLM.

Synack tests eight of the OWASP LLM Top 10, described below:

1. **Prompt Injection:** Prompt Injection describes a scenario where a particular input to the LLM produces an undesirable output. This can range from inappropriate responses from a chatbot to sensitive data exposure from a search bot.
2. **Insecure Output Handling:** If an LLM's output interacts with a plugin susceptible to common vulnerabilities like cross-site scripting or remote code execution, the LLM may be leveraged by an attacker as a tool to exploit the flaw.
3. **Training Data Poisoning:** If an LLM learns from user feedback and input, an attacker may purposefully poison the model by providing false or harmful input.
4. **Supply Chain:** An implementation of an LLM may involve calls to libraries or services that are vulnerable, for example, an outdated Python library.
5. **Sensitive Information Disclosure:** LLMs may leak sensitive information in a response or mistreat sensitive information that is inputted into the model.
6. **Insecure Plugin Design:** LLM plugins are called by models during interaction. If an attacker knows of a vulnerable plugin being called, they may craft specific input to exploit known vulnerabilities in that plugin.
7. **Excessive Agency:** An LLM has unnecessary permissions in an environment. For example, an LLM may need to read documents but may erroneously have write/edit permissions to the same documents.
8. **Model Theft:** An individual model may be trained on proprietary information, making the model itself unique IP. A copy of the model should not exist, however, attackers may be able to abuse the model in such a way that they are able to make a functional copy.

Deliverables: Coverage Reports

You will receive reports for each of the eight vulnerabilities tested for, including details around the researcher's methodology. The reports will be vetted by an internal team at Synack called Vulnerability Operations for quality.

These reports provide coverage and peace of mind that a researcher looked for these common and critical vulnerabilities on your LLM implementation.

Note that these missions are intended to be run in conjunction with a Synack14, Synack90 or Synack365 pentest covering the same scope. As such, you can expect all the features and deliverables that come with those tests, including coverage analytics, exploitable vulnerabilities and on-demand patch verification.