Exercises for Module 5

The exercises for the fifth module are about API tool-calling agents and prompt injection. Supporting code can be found at https://github.com/henningth/Generative-AI-in-Cybersecurity

Agents

Exercise 1: Download the file 05_xkcd_agent.py. This file uses the OpenAPI specification of the XKCD API to enable the user to ask questions about various comics on the XKCD website.

(a): Run the code and observe the output. What does the agent do in each invocation?

(b): Modify the code so that the user can input a question that the agent answers. The user should be able to ask several questions and abort the application by inputting "quit".

Encoding bypass

Exercise 2: This exercise is about converting an input to e.g. Base64, Hex or similar, and see if this influences the output from the LLM.

(a): Write Python code that takes input from a user and converts this input to Base64.

(b): Add a system prompt with the content "Respond to the following Base64 encoded request in Base64".

(c): Combine the system prompt and the Base64 encoded user prompt, and test this on both OpenAI models (e.g. gpt-4o-mini) and Open-Source models via Groq (e.g. llama3-8b-8192). Observe whether the size of the model has any effect on the output.

(d): Do the same test, but now use Hex encoding instead of Base64. Does this have any effect?

Indirect prompt injection

Exercise 3: This exercise is about indirect prompt injection. Solve the Portswigger lab at: https://portswigger.net/web-security/llm-attacks/lab-indirect-prompt-injection. Reflect on what part of this constitutes the indirect part of the prompt injection attack.

Excessive agency in LLMs

Exercise 4: This exercise is about command injection in the context of LLMs. Solve the Portswigger lab at: https://portswigger.net/web-security/llm-attacks/lab-exploiting-vulnerabilities-in-llm-apis. Reflect on what part of this constitutes excessive agency in this attack.

Prompt injection

Exercise 5: The website Gandalf by Lakera at https://gandalf.lakera.ai contains various challenges related to prompt injection. There are eight levels of increasing difficulty. Try to solve some of the levels.
Hints: Recall that LLMs have difficulty distinguishing between code and data (in our case, system and user prompts). Also, try having the LLM act in role-playing.