Federal Office
for Information Security

Evasion Attacks on LLMs – A Checklist for LLM System Hardening

Facing Prompt Injections, Jailbreaks and Adversarial Attacks

The Checklist

This checklist provides the target audience with a structured list of tasks to be completed in chronological order. The checklist is divided into three thematic areas. First, a general understanding of LLMs, evasion attacks and countermeasures is to be established. Then attack vectors specific to the own use case are identified, followed by the selection and integration of appropriate countermeasures as a baseline approach.

However, it must be kept in mind that currently there is no single bullet proof solution for mitigating evasion attacks. Therefore, no warranty or liability is assumed for its completeness or effectiveness of the presented list. This checklist offers general guidance and must be adapted to the specific application context.

Foundation
☐ Understand the basics of how LLMs work!
☐ Understand, what evasion attacks are and how they work! Read papers and blogs! 🕮 🕮 🕮 🕮 🕮
☐ Analyze, develop and test practical examples! • Minigames and challenges - Test your AI hacking or defending skills! 🕮 🕮 🕮 🕮 • Learn about Red Teaming! 🕮 🕮 • Are you able to create your own attacks? Give it a try! (e.g., Indirect Prompt Injections via self-hosted LLM, ...)
☐ Learn about countermeasures! 🕮 • What countermeasures exist? • How do these countermeasures work?
☐ Raise awareness among the users of the LLM system.

Threat Modelling
☐ Describe your use case! • What is the task of the LLM? • What input and outputs do you need? • Which components and functionalities are necessary for the desired interaction? • Sketch your LLM system.
☐ Identify attack targets! • What impact do manipulated outputs of the LLM system have? • Are particularly sensitive data processed? E.g., personal data, confidential company data • What impact does a disruption of the LLM system's operational function have? • What actions can a manipulated LLM perform? Is there a risk of data leakage?
☐ "Lethal Trifecta", "Agents Rule of Two" - Check if your agent combines these three features; an attacker may trick it into accessing your private data, sending it to that attacker or writing actions. 🕮 🕮 🕮 • Has the LLM access to your **private data**? • Is the LLM exposed to **untrusted content**? • Has the LLM the ability to **externally communicate or to overwrite or change state through writing action**?
☐ Identify possible attackers! • Are the users of the LLM trustworthy? E.g., based on company affiliation or identity verification • Are external data sources and tool servers trustworthy?

□ Check your LLM system components (e.g., databases, functionalities and prompts)!

- Are there entry points for evasion attacks? 🕮
- Where are the components deployed?

□ Make safeguard requirements – Identify risks to mitigate!

LLM System Building and Hardening

□ Choose an LLM!

□ If applicable, use secure design patterns! 🕮 🕮

□ If you are in the case of the "Lethal Trifecta", “Agents Rule of Two” (see above):

- Check if it is possible to design your system so that the three features aren´t accessible to the LLM simultaneously.
- For example, if your LLM is able to process your private data and can also do internet research, you might check whether those two activities could be done in separated sessions. In this case, be careful about session-spanning information like long-term memory.

□ Consider the following questions, when selecting countermeasures for implementation!

- Which countermeasures seem practical?
- Which countermeasures are already implemented in the selected LLM / LLM system?
- Which system component should be protected by the countermeasure?
- Which countermeasures will be implemented in-house?
- Are there tools, libraries, and software packages available?

□ Adopt a multi-layered security approach by applying different, complementary countermeasures

- Baseline approach: Implement basic measures!
    - **AICTA – AI Cybersecurity Training and Awareness:** AI cybersecurity training conveys a demonstrated understanding of AI cybersecurity principles and background knowledge on role-specific countermeasures to stakeholders directly involved in the development, deployment, maintenance, cybersecurity or only use of AI systems. The training is tiered according to technical roles and responsibilities in AI system development, deployment and usage.
    - **SSM – Safety System Messages (Prompts):** Safety system messages are a type of system prompt that provides explicit instructions to mitigate against potential harms and guide systems to interact safely with users. When prompting it is important, among other things, to use a clear language, be concise, emphasize certain words and assure robustness. The effectiveness of the countermeasure requires appropriate training of the system prompt designer.
    - **RBP – Role-based Prompting:** Role-based prompting means, that the LLM is assigned a clearly defined and accountable role (e.g., user: "You are an ethical legal advisor"). Therefore, it becomes less likely to respond in ways that violate guidelines. It promotes a stronger alignment, reinforces the context (e.g., LLM: “As a doctor, I can’t give illegal advice”) and reduces ambiguity. The countermeasure is performed on the system prompt. The effectiveness of the countermeasure requires appropriate training of the prompt designer, including the user.

  - **HAG – Human Action Guardrail:** The human action guardrail is the process, where the user is engaged in authorizing a critical operation of the LLM, in stopping the current operational process, if potential threats are detected, or in ignoring the detected incident. The countermeasure is performed on the output side of the LLM, where the action is executed. The effectiveness of the countermeasure requires appropriate training of the user.
  - **HEF – Hypertext Element Filtering:** Hypertext element filtering is the process of detecting, highlighting or removing specific HTML elements (e.g., URLs, Links, email addresses or embedded program code), within text to prevent security risks or enforce content policies. The guardrail is performed on the user prompt, additional user data, data from other sources and the generated output of the LLM. It prevents the inclusion of malign underlying content.
  - **CS – Content Stripping:** Content Stripping aims on removing or simplifying unnecessary or irrelevant information. This can include metadata, hidden text (e.g., special Unicode characters), gibberish text, formatting data, headers and footers. It is performed on the user data, data from other sources and the generated output of the LLM. It mitigates risk of attacks, that are hidden in this unnecessary information.
  - **SIR – Sensitive Information Redaction:** Sensitive information redaction aims on identifying and redacting sensitive data (e.g., personal information, API keys). It is performed on the LLMs output. It reduces the leakage of sensitive or confidential information, that is embedded in the LLM through training or stored in additional databases, that serve as a knowledge base.
  - **LR – Labels and Reasoning of Data and Action:** The measure ensures that data generated by the LLM is clearly marked (e.g., via watermarking or readable indicators). In addition, the data basis for generated data and decision-making leading to executable actions should be presented in a transparent and traceable manner — for example, by referencing the sources that the model relies on in its reasoning.
  - **MAPM – Model Action Privilege Minimization:** Model action privilege minimization reduce the actions, that the LLM can potentially trigger, to the minimum set necessary for the use case of the LLM system. Additionally, the measure assures that they are executed with suitable rights and privileges. In particular, actions triggered by a request from a specific user should be performed within this user’s security context, thus inheriting their rights and privileges. The countermeasure mitigates the damage through a successful direct or indirect attack at the output-side of the LLM.
  - **SP – Structured Prompts:** The countermeasure ensures, that prompts are designed and processed by the LLM in a structured format to represent messages with metadata, distinguishing between different roles (e.g., user, assistant, system). This format helps the LLM better understand the context and flow of a conversation. The implementation lets the LLM act more robust against different kinds of Evasion attacks. Some developers and providers already offer this as a built-in feature of their models.

- Based on criticality of threats and use case consider implementing more measures!
  - Does the LLM system fulfill the fundamental criteria for determining its criticality?
  - Which threats have been identified but not yet sufficiently mitigated?

☐ Test the resulting LLM system!

- Do benchmarking!
- Perform Red Teaming!