Federal Office
for Information Security

Indirect Prompt Injections

Intrinsic Vulnerability in Application-Integrated AI Language Models

Document history

Version	*Date*	*Editor*	*Description*
Content	21th July 2023	TK24	This document provides an English translation of the cybersecurity warning “CSW-Nr. 2023-249034-1032, Version 1.0, 18.07.2023“ [BSI23-1]

Table 1: Document History

Federal Office for Information Security
P.O. Box 20 03 63
53133 Bonn
Internet: https://www.bsi.bund.de

Table of Contents

1 Context

Large Language Models (LLMs) are becoming increasingly popular. Amongst others, they can automatically process text documents and assist users by means of chatbots and autonomous agents. The functionality is continuously expanded. Using plugins, it is now possible for chatbots to automatically evaluate web pages or documents and to access programming environments or mailboxes. In many of the anticipated use cases, unverified data from insecure sources is processed.

In this case, LLMs are vulnerable to so-called *indirect prompt injections*: attackers can manipulate the data in these sources and place unwanted instructions for LLMs there. If LLMs access this data, the unwanted instructions may be executed. Attackers can thereby manipulate the behaviour of LLMs in a targeted manner. The potentially malicious commands can be encoded or hidden and may not be recognizable by users. In simple cases, this could be text on a web page with font size zero or hidden text in the transcript of a video. However, it is also possible to encode instructions so that they are still easily interpreted by LLMs but are difficult to read by humans (e.g., for example using ASCII code or similar). It might also be that a web server answers requests from chatbots with different content than human users receive from browser requests due to different call parameters.

The manufacturer OpenAI also points out this vulnerability in connection with the use of plugins in the ChatGPT product on 13th June 2023: “However, there are still open research questions. For example, a proof-of-concept exploit illustrates how untrusted data from a tool’s output can instruct the model to perform unintended actions.” [OAI23]

The risks following from the attack vector depend strongly on the specific use case and the conditions of use of the LLM, such as the action capabilities or rights. We outline possible consequences of such attacks for different use cases. All Examples are based on actual proof of concepts (PoCs):

- Use of an LLM to summarize or analyze text from external sources
  - Attackers could manipulate the result in a targeted manner
- Use of a chatbot accessing modified web pages
  - Results of queries could be manipulated in a targeted manner
  - The chatbot could exhibit undesirable behaviour and, for example, make legally questionable or undesirable statements
  - The chatbot could motivate users to access a (malicious) link
  - The chatbot could attempt to obtain sensitive information from users (e.g., credit card information)
  - Attackers could extract sensitive information from the chat history if, for example, the possibility to call web pages or display external images exists
  - The chatbot itself could call additional plugins and thus perform unwanted actions, such as:
    - Accessing e-mail account, summarizing recent emails and extracting information
    - Publishing private source code repositories
- Autonomous agent running locally in a Docker container and accessing an LLM via API:
  - Attackers could break out of the container and gain root privileges on the target system

After academia discussed this new vulnerability first in February 2023 [GRE23], the Federal Office for Information Security in Germany (BSI) has already addressed the attack vector in the paper “Large Language Models: Opportunities and Risks for Industry and Authorities” [BSI23-2]. Since the presentation and discussion of concrete PoCs for exploiting the vulnerability have been increasingly observed on the internet in recent months and, at the same time, the integration of language models in applications is progressing rapidly, BSI raises once again awareness for this new vulnerability class with this notification.

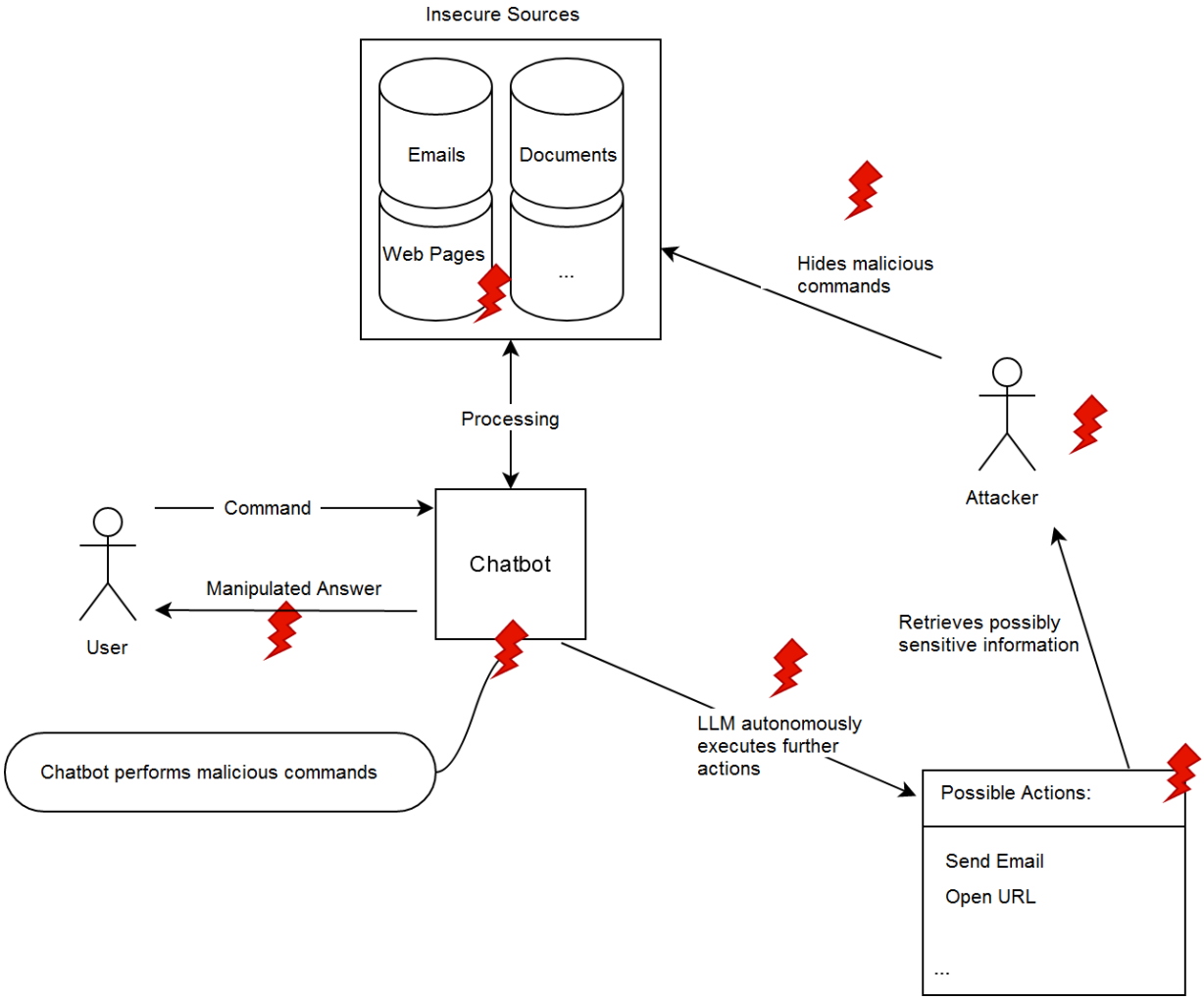

Figure 1 Indirect prompt injection with possible consequences in the context of chatbots

2 Assessment

The risks posed by indirect prompt injections are serious and arise when LLMs process information from insecure sources. The given examples are credible and the BSI has reproduced them partially. The impact of the vulnerability depends on the specific deployment scenario and the possible actions (or activated plugins) of the LLMs, such as access to sensitive data. Depending on the scenario, the impact of an attack can be considerable. However, the effort required to exploit the vulnerability and the impact can only be assessed in a specific individual case as part of a systematic risk analysis.

In human communication, texts can both convey information and issue commands. This ambiguity is now transferred to the IT sphere: Also in LLMs, there is no clear separation between data and instructions. Since this is an intrinsic weakness of the current technology, attacks of this type are fundamentally difficult to prevent. Currently, there is no known reliable and sustainably secure mitigation that does not also significantly limit functionality. The potential impact of the vulnerability is amplified if the LLM is used as a (partially) autonomous system that can independently perform actions with critical consequences.

Users usually have no way to detect such an attack by inspecting the sources themselves, since the commands can be both hidden and encoded.

3 Measures

When integrating LLMs into applications, a systematic risk analysis should be performed that explicitly assesses the risk posed by indirect prompt injections.

The risk can be reduced by excluding access to insecure sources or by performing human control and authorization before executing potentially critical actions of the LLM. This is also advisable because LLMs can currently hallucinate and make incorrect decisions even in the absence of attacks. To reduce the impact of a potential attack, actions may be restricted to be reversible or executed in a segregated environment ("sandbox"). In any case, the possible actions (or plugins) of LLMs should be limited to a minimum required for the use case. Performing targeted penetration tests (red teaming) can help to better assess the risks for a specific use scenario.

The manufacturer OpenAI also lists the following countermeasures in the context of the use of plugins in ChatGPT on 13th June 2023: "Developers can protect their applications by only consuming information from trusted tools and by including user confirmation steps before performing actions with real-world impact, such as sending an email, posting online or making a purchase." [OAI23]

Known attacks can be blocked by operators and manufacturers through filters. However, it is difficult to detect variations. Currently, mitigation measures are being discussed as well as tested by researchers, developers, and vendors to make exploiting the vulnerability more difficult, such as filtering and validating input or introducing roles for chatbots to better separate instructions and information [OWA23]. Overall, however, it should be emphasized that indirect prompt injections for LLMs are a relatively new type of vulnerability and no security best practices currently exist in this area.

It is important to make users and developers of LLMs aware of potential risks or limitations of the technology. One reason is that, similar to a social engineering attack, a manipulated chatbot may be able to credibly argue why a (malicious) action needs to be authorized.

The BSI publication "Large Language Models: Opportunities and Risks for Industry and Authorities" provides a compact overview of the opportunities and risks of LLMs [BSI23-2].

4 Bibliography

[BSI23-1] https://www.bsi.bund.de/SharedDocs/Cybersicherheitswarnungen/DE/2023/2023-249034-1032.pdf?__blob=publicationFile&v=3

[BSI23-2] https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/AI-in-lLanguage-processing.html?nn=910042

[GRE23] https://arxiv.org/abs/2302.12173

[OAI23] https://openai.com/blog/function-calling-and-other-api-updates

[OWA23] https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-2023-v05.pdf