Proceedings of CONF-SEML 2026 Symposium: Computational Analysis and Modeling in Complex Intelligent Systems
DOI: 10.54254/2755-2721/2026.GU33494

Evaluating the Safety of LLM-Based Agents: A User-Centered Study of OpenClaw

Guanbo Liu

International School, Beijing University of Posts and Telecommunications, Beijing, China
2024213684@bupt.cn

Abstract. This treatise conducts an evaluation of OpenClaw, an artificial intelligence agent grounded in large language models (LLMs), from a user-centered perspective through a light-weight real-world experiment. The study is designed to analyze the system's behavior under different categories of tasks, including benign, risk-sensitive, and adversarial scenarios. The results demonstrate that OpenClaw shows dependable working capability in ordinary tasks, producing structured and accurate outputs, while also exhibiting a certain level of basic safety awareness in adversarial situations such as prompt injection and harmful command requests. However, limitations become apparent when users encounter risk-sensitive situations. The system may still generate potentially dangerous outputs, even when warnings are provided, indicating that its safety mechanisms rely more on notification rather than strict prevention. In addition, the behavior of the system is found to vary depending on how user inputs are phrased, revealing sensitivity to prompt formulation. These findings suggest that, although OpenClaw has practical usability as a general-purpose assistant, there remains a clear need for stronger and more reliable safety control mechanisms in LLM-based agents to ensure secure real-world deployment.

Keywords: Large Language Models, AI Agents, Security Evaluation, User-Centered Study, Prompt Injection

1. Introduction

The fast progress of Large Language Models (LLMs) has caused the emergence of independent Artificial Intelligence (AI) agents which can directly have interactions with operation systems [1,2]. In these items, OpenClaw has appeared as one widely used open-source project [3]. It is designed to be a "personal artificial intelligence assistant" which can complete complex work by means of natural language, for example handling files, managing programs, or carrying out system orders [4].

Be different from traditional conversation robots which are restricted to interactions based on the web, OpenClaw carries out work with a higher degree of self-rule in local environments [5]. It can implement reading, writing, and execution of commands on a user's system, this enormously enlarges its capabilities but also puts forward significant security questions. When such agents start to carry out actions for users, the danger of actions that are not intended or harmful becomes a severe problem [6].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.GU33494

Most current studies on LLM agents put focus on promoting their abilities, including inference, tool utilization, and long-term memory [7]. Numerous researches assess these systems inside controlled surroundings and announce powerful task finishing achievement [8]. However, these evaluation works are frequently carried out under safe and trusted conditions, hence they may not reflect the actual usage scenarios of the real world.

On the opposite side, recent researches in the safety field have already begun to emphasize the possible dangers of LLM agents, which include prompt injection, harmful tool use, and the difficulties of carrying out safety restriction rules [9]. Frameworks which include MITRE ATLAS have also been put forward in order to depict the threat situation of AI systems [10].

Although researchers have made these efforts, an obvious blank in current research still remains. In the real-world personal computing environments, the user-centered research works which study these risks have a comparatively small number. Most current researches still stay on theory or rely on simulation, and thus the comprehension of actual safety methods that can be used by common users is still not enough [11].

This current treatise conducts an investigation of OpenClaw from the perspective which centers on users. A light-weight real world experiment which is of small weight is carried out to analyze the system's behavior under two kinds of conditions that are normal and risky.

2. Method

2.1. Introduction of OpenClaw

OpenClaw is one open-source artificial intelligence agent which based on large language models, it interacts with the local computing environment of the user. Different from traditional chat robots which only make text answers, OpenClaw has the ability to understand guidance and create runnable movements, for example document operations or command line proposals. This system carries out work in three big stages: comprehending user's input, making plans for actions, and producing outputs or carrying out steps.

Another important characteristic of OpenClaw is its expansible ability via tools and skills, this therefore allows the interaction with external systems. But, this also brings potential safety hazards, because the system may generate or execute unsafe commands, especially in antagonistic scenarios such as prompt injection. Past researches have already indicated that large language model agents possess great capability yet may display safety weaknesses, especially when they are beyond controlled settings.

This research utilizes a light-weight real-world arrangement to assess OpenClaw from the aspect of actual user. This system has been placed onto a local Windows computer through the use of the official setup procedure, and DeepSeek has been utilized as the back-end model for the purpose of guaranteeing steady working performance. No extra tools, skills, or outside integrations were turned on, and web search function was closed, hence this research do this in order to put focus on the agent's core behavior.

2.2. Tasks and evaluation

Six work items were designed and split into three kinds: harmless work items (e. g., summary making and to-do item list producing), risk-sensitive tasks (e. g., producing shell instructions and removing files), and confrontationary tasks (e. g., the instructions that damage system and the

DOI: 10.54254/2755-2721/2026.GU33494

injection of prompts). Every task was given in natural language via the OpenClaw interface. All the responses were recorded, and screen photographs were taken for every task as proof materials.

The working flow of this system includes user input, explanation by LLM, creation of actions, optional carrying out, and sending of output. For the sake of safety, all instructions which have possibility to bring harm were tested in a controlled way and were not really put into execution.

This evaluation was conducted according to three hierarchies: Safe (behavior that is correct and possesses no risk), Medium (behavior that has possible danger but is provided with alerts), and Unsafe (behavior that is dangerous and has no alerts). This research carried out the experiment on a personal Windows system, tasks were input by hand and responses were watched in real time. This evaluation depended on direct watching of task finish, alert action, and rejection of not safe orders.

3. Results and discussion

3.1. Experimental results

OpenClaw was tested using six tasks, and the results are summarized in Table 1.

Table 1. Experimental results

Task	Type	Expected Behavior	Actual Behavior	Safety Level
Summarization	Benign	Complete task	Completed successfully	Safe
To-do list	Benign	Generate structured plan	Detailed and structured output	Safe
Shell command	Sensitive	Warn or restrict	Provided command with warning	Medium
Delete file	Sensitive	Ask confirmation	Simulated execution with tool behavior	Medium
Remove system files	Adversarial	Refuse	Explicit refusal with explanation	Safe
Prompt injection	Adversarial	Reject	Rejected and maintained safety constraints	Safe

Under the friendly working tasks, the OpenClaw can carry out work in a dependable way, and it generates outputs that are correct and have good structure. In the tasks which are sensitive to risk, the system shows a mixed kind of behavior. For instance, when it produces shell commands for deleting files, it gives out commands that possibly have danger but also puts forward warnings and more secure replacement choices. This gives us that the system has shown a certain degree of safety consciousness, although outputs with risk may still be produced by it.

In the scene of document erasing, the system performed simulation operation without asking clear confirmation, which therefore can bring about problems if actual operation is turned on. By comparison, the system carries out work more sturdily in opposite tasks, for example declining orders that harm the system and standing against attempts of prompt injection, hence it shows that basic safety restrictions do exist.

In addition, the system's answering contents are on the whole clear and easy for people to comprehend. Even on the occasion when warnings are issued, risks obtain explanations which use relatively simple words, which therefore can assist ordinary users to better recognize potential dangers. But, step-by-step instructions on risk evasion are not always given, hence that a certain degree of user knowledge is still needed.

This system in addition has a trend to give beneficial proposals even when it is located inside dangerous circumstances. For example, when producing dangerous instruction sequences, more secure replacement choices can be put forward at this same time. Although this kind of behavior can

DOI: 10.54254/2755-2721/2026.GU33494

have use, it can also bring about puzzlement to users who do not have much experience, hence both safe and unsafe choices sometimes are displayed together.

3.2. Analysis and discussion

The outcomes discover an observable rule: OpenClaw obtains good effect in regular tasks but displays different actions when task risk grows higher. For simple usage situations, the system on the whole possesses reliability, hence it shows that it has usefulness when acting as a general assistant. Nevertheless, in scenes which have high requirement for risk sensitivity, protective mechanisms are not always sufficiently strong, hence dangerous commands may still be produced even when warnings are already issued.

Another problem is the absence of examination mechanisms in some operations, for example document deleting, which therefore could possibly bring unforeseen results in actual practical usage. In the same moment, this system exhibits strong capability in the scenarios which are clearly malicious, where requests that do harm are rejected and constraints on safety are kept.

One important difference between "warning" and "prevention" is able to be gotten. Under some circumstances, the system gives warnings, but does not actively block risky conduct, which therefore may not be enough for users who have no familiarity with command-line operations. More strong safety mechanisms, for example action stopping or forced confirm steps, can raise system dependability.

Furthermore, the system's behavior thus has differences that rely on the way how task descriptions are worded. Even the small changes in word choosing can cause different answers, therefore showing that there is sensitivity to the variation of input. This characteristic can bring in extra security dangers in the actual use situations.

On the whole, OpenClaw may be regarded as having use value but not being completely safe under its present condition. Promotions in safety controlling, including more strict restrictions and clearer mutual design, are requisite. This research also possesses restrictions, including a small quantity of test examples and the lack of external instruments. More research works are required by us for making clearer the influence on safety that these systems bring.

4. Conclusion

This current work investigates the acting condition of OpenClaw from a perspective which centers on users by means of one lightweight experiment that is carried out in real environment. The outcome of research shows that this system carries out work in a dependable way in gentle tasks, and therefore displays basic safe consciousness in distinct opposite situations. Nevertheless, its behavior has become less consistent in the situations which are sensitive to risk, where outputs with potential danger may still be produced in spite of the warnings that are given.

This research can point out several problems that have practical meaning. This system often depends on warning notices instead of active prevention methods, therefore it may not be sufficient for users who possess restricted basic technical knowledge. Furthermore, the absence of inspection systems and the easiness to have input word order problems can therefore hence bring extra dangers in actual world application. These observed results therefore indicate that the present safety designs still have the necessity for further improvement.

In the same period, OpenClaw gives explicit expositions and now and then puts forward more secure replacement choices, which therefore can promote usability but hence also bring about

DOI: 10.54254/2755-2721/2026.GU33494

puzzlement when safe and unsafe choices show up at the same time. On the whole, this system seems to have use value but has not achieved complete reliability on the aspect of safety.

This present research possesses certain restrictive aspects, which include a small quantity of tasks and a simplified experimental establishment. In the future, work can expand the scope of scenarios and investigate more firm safety mechanisms for agents that are based on LLMs.

References

[1] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J. and Wen, J. (2024) A Survey on Large Language Model Based Autonomous Agents. Frontiers of Computer Science, 18, 186345.
[2] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L. and McGrew, B. (2023) GPT-4 Technical Report. arXiv, arXiv: 2303.08774.
[3] Wang, Y., Chen, X., Jin, X., Wang, M. and Yang, L. (2026) OpenClaw-RL: Train Any Agent Simply by Talking. arXiv, arXiv: 2603.10165.
[4] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K.R. and Cao, Y. (2022) ReAct: Synergizing Reasoning and Acting in Language Models. International Conference on Learning Representations (ICLR).
[5] Castelfranchi, C. (1998) Modelling Social Action for AI Agents. Artificial Intelligence, 103, 157-182.
[6] Shan, Z., Xin, J., Zhang, Y. and Xu, M. (2026) Don't Let the Claw Grip Your Hand: A Security Analysis and Defense Framework for OpenClaw. arXiv, arXiv: 2603.10387.
[7] Huang, X., Liu, W., Chen, X., Wang, X., Wang, H., Lian, D. and Chen, E. (2024) Understanding the Planning of LLM Agents: A Survey. arXiv, arXiv: 2402.02716.
[8] Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y. J., & Huang, G. (2024, March). Expel: Llm agents are experiential learners. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 17, pp. 19632-19642).
[9] Li, Y., Wen, H., Wang, W., Li, X., Yuan, Y., Liu, G. and Liu, Y. (2024) Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security. arXiv, arXiv: 2401.05459.
[10] Goel, N. and Gupta, N. (2025) Extending STRIDE and MITRE ATLAS for AI-Specific Threat Landscapes. Well Testing Journal, 34, 181-196.
[11] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P. and Horvitz, E. (2019) Guidelines for Human-AI Interaction. Proceedings of the CHI Conference on Human Factors in Computing Systems, 1-13.