Gmail Carlos Denner <carlosdenner@gmail.com>

Communications of the ACM - Decision on Manuscript ID CACM-25-11-5646

1 message

Communications of the ACM <onbehalfof@manuscriptcentral.com> Tue, Jan 13, 2026 at 11:08 AM
Reply-To: eic@cacm.acm.org
To: carlosdenner@gmail.com
Cc: carlosdenner@gmail.com

13-Jan-2026

Dear Dr. Denner dos Santos:

Manuscript ID CACM-25-11-5646 titled "Prompt Injection Demystified: Building an LLM Firewall for Production LLM Systems" which you submitted to Communications of the ACM has been reviewed. The reviewer(s)'s comments are included at the bottom of this email.

Because of this reviewer(s) feedback, I decline to publish the manuscript in Communications of the ACM at this time. However, you may submit a new and revised manuscript to address the issues raised by these reviews.

Please note that resubmitting your manuscript does not guarantee eventual acceptance. Your resubmission will be subject to a complete re-review.

You cannot change the originally submitted version of the manuscript. Instead, please create a new manuscript and resubmit it.

Once you have revised your manuscript, go to https://mc.manuscriptcentral.com/cacm and enter the Author Center, where you will find your manuscript title listed under "Manuscripts with Decisions." Then click on "Create a Resubmission," located next to the manuscript number. Then, follow the steps for resubmitting your manuscript.

Because we are trying to facilitate the timely publication of manuscripts submitted to Communications of the ACM, your revised manuscript should be uploaded as soon as possible.

If you cannot submit your revision in a reasonable amount of time, we may have to consider your paper as a new submission. Please see deadlines on your ManuscriptCentral Author Dashboard. If you need a slight extension, please contact the Editor-in-Chief.

I look forward to seeing a resubmission that addresses the reviewer(s)' comments.

Sincerely,
James R. Larus
Editor-in-Chief, Communications of the ACM (CACM)
Professor Emeritus, EPFL

EIC:
I agree with the AE, the current article does not meet CACM standards. The topic, however, is an important one. I am changing the recommendation to Reject&Resubmit, which offer an opportunity to submit an **entirely new** article that is written to CACM standards and that address all of the issues raised in the review.

Co-Chair: Co-Chair, Practice
Comments to the Author:
Please see the Editor's remarks.

Associate Editor: van Deursen, Arie
Comments to the Author:
This is an interesting paper about prompt injection attacks.

In its current form it cannot be sent to reviewers, as they would reject it. This is partially due to the Q&A-like and tutorial-

like ("what you will learn") and somewhat chaotic writing style. In addition to that, the paper leaves out a lot of relevant information, while at the same time providing some very specific details (such as the Python normalizer function used), making it hard for the reader to appreciate the big picture.

Below I give my understanding of the paper, with some comments in square brackets, followed by a few thoughts on how to proceed.

The goal of the paper (but this isn't stated so explicitly) as I see it is to help the reader understand how to create simple, deterministic, easily audited defenses against prompt injection attacks.

To that end, the paper proposes a prompt filtering architecture (an "LLM Firewall") consisting of three steps:

1. An input normalizer taking care of unicode 'canonicalization', zero-width stripping, and homoglyph mapping to eliminate 'confusables';
2. The 'v1' signature filtering for 47 regex patterns identifying known injection markers (like 'ignore previous')
3. The 'v3' semantic filtering which uses an embedding similarity to 150 attack exemplars (no examples are given).

The paper also mentions a v2, structured heuristics, for "checking suspicious patterns like JSON fields with instruction-like text", but in the rest of the paper v2 is not mentioned. [what is the status?]

The firewall is based on a survey of 18 LLM security filings. [Which ones? Why these ones? Why not also include an analysis of academic literature (in ACM Digital Library?)]

In section 3, for an evaluation, the paper uses 400 attack prompts (200 "RAG-borne" and 200 "schema smuggling"), 200 clean benign queries, 260 obfuscated bening queries, 65 novel jailbreak attacks (from jailbreak repositories -- multi-turn, context-confusion, semantic paraphrasing, goal hijacking), and 30 adverserially evolved attacks. [examples would help the reader appreciate these data sets]

There is a dataset on GitHub, but it seems incomplete. [the paper should give a much better intuition for what the dataset contains]

The paper describes a series of experiments:

P1: The 400 attack prompts were applied to LLaMa and Falcon. LLaMa complied with 32%-65%. [which ones are caught by both, and which ones by neither? Why? Examples?]

P2: To assess true positive rate and false alarm rate, v1 and v3 were applied to an even mixture of benign and attack prompts. No false alarms were raised. 57-80% true positive rates were found [what sort of cases were not found?]

P3: Filtering on v1 OR v3 still has 0% false alarms, and increases true positive rate to 87% [appearantly the two filters cover disjoint cases -- give examples? Discuss?]

P4: The similarity treshold of v3 has no effect in the range 0.1-0.7 [why??]

P5: To address obfuscation, a first step is adding the normalizer. The experiments on 260 obfuscated benign queries indicate a 23% false alarm hit without normalization, reduced to 10% to normalization. [in p6 this is described as a 'false alarm stress test' under "P6a"]

P6: From novel jailbreak attacks about half are caught, indicating the need to periodically update v1/v3 ("P6b"). The paper also mentions P6c, "30 adversarial attacks generated by mutating detected attacks until they evaded filters", but reports no results. [this is the kind of obfuscation that seems most interesting? Elaborate?]

P7: Filtering adds sub-millisecond latency

P8: Filtering scales linearly in terms of Memory / GPU consumption.

Section 4 discusses a setup of using the proposed LLM Firewall in both a production setting, and in a monitoring setting (for collecting threat intelligence). As explained in the caption of Fig.1, production uses just v3, and monitoring "adds v1 for higher recall."

Section 4 appears to repeat the experimental results of section 3 -- it is not clear why / what the difference is.

Section 5 suggests a two-week rollout plan for this firewall.

Section 6 describes a number of lessons.

Based on this summary, the following issues block publication:

- The paper needs a very clear and explicit goal. To be suitable for CACM this goal must include introducing the wider audience to the importance of prompt filtering, and keeping this wider audience engaged throughout the paper. Information for the subset of people interested in replicating the firewall, can be put in a dedicated section with reference to the github page.

- From this goal, a compelling paper structure should follow. I would suggest to have at least one running example with variations to illustrate all steps (e.g., attacking prompt, matching regex, matching embedding, ways to circumvent these, ...)

- In line with this goal, a section on the filtering state-of-the-art would be interesting. This could then also include an actual discussion of the 18 patent filings mentioned, and potentially some other relevant references.

- One result the paper seeks to present is the proposed "LLM Firewall". This deserves an explicit section. The proposed firewall is very simple, but it serves to illustrate a few points. This should be clarified in the paper.

- The paper should provide more information about the filtering steps in the firewall. This could be in the form of a table, with, say, 10 representative examples of patterns included (v1) or attacks considered (v3).

- The simple firewall is used to conduct a number of experiments. These serve to give the reader some intuition about potential precision and recall. But the paper should also make clear that the actual number found ("57%") are very sensitive to the precise actual experimental setup.

- The credibility of the numbers is directly related to the credibility of the datasets of attack/benign prompts used, in relation to the patterns / attack examplers included. These datasets should be described better, and be made fully available (not just on request).

- The deployment story (section 5) and the monitoring/production mode (section 4) is confusing at the moment. A single section presenting this as one possible deployment approach might be feasible.

- Lastly, the writing should be sanitized.

It is clear that this amounts to a complete rewrite of the paper. If the author is willing to do this, the resulting revision could then go to reviewers to hear from experts what they think of this paper.