Evaluating Vulnerabilities in LLMs: Red-teaming and Mitigation

Vincent Yang

Senior Project

May 17, 2025

ABSTRACT

In recent years, artificial intelligence (AI) technology, and especially large language models (LLMs) have improved in their advanced language comprehension and generation capabilities; thus, they have become increasingly more integrated across various sectors, essential in fields such as machine translation, customer service, programming, etc. However, as a budding technology, LLMs still face significant challenges in terms of safety, security, and privacy concerns such as hallucinations, jailbreaking, backdoor attacks, and data leakages that undermine their reliability, effectiveness, and overall reputation. While efforts are being made to address these concerns, identifying potential vulnerabilities in LLM systems is not always straightforward. AI red-teaming, which refers to the practice of adversarially testing AI systems, simulating how adversaries would attack a system to identify vulnerabilities, is an important process in improving AI safety, security, and privacy. In this paper, we provide a comprehensive survey on contemporary red-teaming efforts to uncover the vulnerabilities of various LLMs in regards to safety, security, and privacy, as well as various strategies that have been developed to address these vulnerabilities. In addition, we include data that compares the susceptibility of varying relevant models to different types of adversarial attacks (Appendix).

Introduction

Large Language Models (LLMs) have transformed natural language processing due to their ability to generate human-like text and perform a wide range of tasks. Rapid progress in model design and scale has enabled systems to exhibit impressive capabilities in several different areas [1]. The release of powerful, open source LLMs like Llama 2 has further expanded access to this technology [2]. These advancements, in conjunction with comprehensive surveys of LLM technology [3], highlight the transformative impact of LLMs in various sectors of society. However, this is accompanied by increasing concern about the safety, security, and privacy implications of the widespread deployment of LLMs.

Making sure that LLMs are safe (do not produce harmful or misleading content), secure (resilient against malicious misuse or attacks), and privacy-preserving (resistant to divulging sensitive information) has recently become a focal point Recent works have begun to categorize the risks LLMs pose with regards to these factors [4] [5] [6]. In this survey, we provide a general overview of the major safety, security, and privacy issues that affect LLMs and their usage, drawing on key findings from literature. We also discuss current mitigation strategies and the role of AI red-teaming, which is the practice of stress-testing models with adversarial prompts in order to identify potential risks in LLM usage. This process, whose name is borrowed from cybersecurity and further back from Cold War military simulations, has great potential in identifying ways in which LLMs might produce harmful outputs or be exploited under malicious use. By consolidating these insights, we aim to inform a general technical audience about the open challenges in developing LLMs that are both powerful and trustworthy.

Background

Large Language Models: Modern LLMs are the product of rapid advances in model architecture and training scale, and are typically built on the Transformer architecture, enabling them to be trained on massive text corpora to learn the statistical structure of language [16]. Through unsupervised pre-training on internet-scale data and subsequent fine-tuning (often with human feedback), LLMs acquire broad knowledge and generative abilities [1] [2]. Examples include GPT-series models and open-source variants like Llama 2. Many LLMs are now deployed in interactive systems (e.g. chatbots, coding assistants), where they must follow user instructions while avoiding problematic outputs.

To make LLMs useful and safe for users, developers employ alignment techniques. A common approach is reinforcement learning from human feedback (RLHF), where human evaluators judge model outputs and those preferences guide further training [1]. Llama 2, for example, includes a fine-tuned chat model aligned with human guidelines [2]. Despite these efforts, alignment remains imperfect—models may still exhibit undesirable behaviors or failure modes not seen during training.

These factors give rise to different categories of concerns. In the following sections, we examine three key risk areas: safety, security, and privacy, defining each and surveying their associated challenges.

Here, we provide some definitions clarifying what is encompassed by the terms safety, security, and privacy.

Safety: In the context of LLMs, safety refers to a model's inherent reliability in the absence of adversarial entities. This typically involves the prevention of generation of content that is toxic, biased, misleading, or otherwise harmful. It also encompasses aligning model behavior with ethical and legal norms. Safety thus deals with how models behave under normal usage scenarios and whether it avoids causing harm by itself.

Security: Security refers to the model's ability to resist attacks from malicious actors. This includes resistance against adversarial inputs, which are prompts that are specifically designed to induce unwanted behavior. Security also pertains to preventing unauthorized access or manipulation of the model (e.g.

model theft or backdoor installation) and ensuring the model can't be readily repurposed for malicious use. In essence, security focuses on adversarial scenarios where an attacker attempts to exploit the model.

Privacy: Privacy in the context of LLMs concerns sensitive information. Privacy risks occur when an LLM reveals personal, confidential, or proprietary information that was present in its training or interactions with a user. A privacy-respecting LLM should not leak personal identifiers, confidential documents, or other private details either inadvertently or under attack.

While these three aspects are distinct, they often overlap. For example, a prompt injection attack could cause a model to reveal private data (a privacy breach) or generate disallowed content (a safety failure).

A strategy used to identify vulnerabilities in LLMs that has been garnering attention is red-teaming, which involves stress-testing AI systems by simulating the behavior of adversaries and malicious actors. Red teamers intentionally probe the model with prompts and scenarios designed to uncover failure modes and vulnerabilities [17]. This process, borrowed from cybersecurity, has potential in identifying how LLMs might produce harmful outputs or be exploited under malicious use. By "attacking" the model in a controlled setting, red teams can reveal weaknesses in safety, security, and privacy. The findings contribute to mitigations and refinements before deployment.

AI Red-Teaming

Red-teaming is generally understood as the methodical practice of testing a system or design for weaknesses by acting as an adversary. In the large language model context, red-teaming is applied to the model, and testers assail the model with adversarial prompts and other attack vectors to uncover safety, security, or privacy vulnerabilities. This work can be done by human experts, by automated tools, or by a combination of both, and it has become an essential part of responsible LLM development and deployment.

For LLMs, safety red teaming might involve trying to get the model to produce disallowed content (eg. hate speech, dangerous instructions) to see how the model responds to such prompts. Security red-teaming might include attempts to break through the model's guardrails, collect confidential model information, or trigger hidden behaviors. The aim is to discover vulnerabilities in a safe and controlled environment before real adversaries get a chance. Red-teaming is a concept that has only recently been applied to LLMs, but it has already seen use and provided value in real-world applications. The developers of GPT-4 report that they had a team of experts stress-test the model, uncovering several issues that included factual errors and harmful advice [1]. The results of the red-teaming informed mitigations and policy refinements prior to GPT-4's public release.

Red-teaming doesn't solve the issues it finds, and that's by design. Instead, it informs developers and policymakers where to focus mitigation efforts. Successful red-teaming leads to more robust models, either through further training (e.g. incorporating adversarial examples into fine-tuning) or improved safety instructions and filters. It is a continuous process: as models evolve and users find new ways to interact with them, periodic red-team assessments help maintain a high safety and security standard.

The emerging field of automated red teaming leverages AI to test the limits of other AI systems. In place of or in addition to human expertise, researchers now employ adversarial prompts generated by AI to probe for weaknesses. One study introduced an approach to systematically find prompts that cause aligned models to produce misbehave, producing objectionable content [11]. Tools like these can help humans cover more ground, more quickly. However, human judgement is still required to ensure the process is reliable.

Red-teaming doesn't solve the issues it finds, and that's by design. Instead, it informs developers and policymakers where to focus mitigation efforts. Successful red-teaming leads to more robust models, either through further training or improved guardrails and filters. It is a continuous process: as models evolve and users find new ways to interact with them, periodic red-team assessments help maintain a high safety and security standard.

In summary, the assessments serve as a sort of "fire drill" for large language models (LLMs). They simulate what could go wrong by having trained professionals attempt to make things go wrong on purpose. As the large language model (LLM) deployments become more high profile and more widespread, it is likely to become standard practice to conduct formal red team exercises to ensure that LLMs meet safety, security, and privacy requirements both pre and post deployment.

Safety

LLM safety issues occur when models produce content that may be harmful, whether it be to the user themselves or society. A major concern is toxic or hateful language. LLMs that have been trained on large datasets from unvetted sources learn the biases or offensive content that exists in those sources. If models lack safeguards, they might end up generating hate speech, harassment, or dangerous information. For instance, studies have shown that even state-of-the-art chatbots can adopt hostile personas or use slurs if guardrails fail to prevent this [7]. Similarly, social biases regarding race, gender, or other miscellaneous attributes can be reflected and amplified in the outputs of an LLM. Audits of various models have found statistically significant bias in their responses to prompts concerning different demographic groups [8]. Such biased or toxic outputs pose risks of discrimination and harm.

Another major safety challenge is LLMs' ability and propensity to produce hallucinations, which are responses that contain plausible-sounding but largely incorrect or fabricated information. An LLM might confidently state false facts, misquote sources, or even invent non-existent articles and individuals in order to satisfy the prompt. These hallucinations can mislead users and erode trust. Recent surveys categorize types of hallucination and note that no current LLM is completely immune to this issue [9]. Mitigating hallucinations is difficult because they often stem from the same creative generative capacity that makes LLMs useful. Techniques like grounding the model's responses in reliable external data or instructing it to express uncertainty are active areas of research [9]. Combatting hallucinations is a balance between maintaining an LLM's ability to faithfully execute on user commands and maintaining informational integrity.

Perhaps the most well-known of LLM vulnerabilities, jailbreak of LLMs is a security concern. Jailbreak, in the context of LLMs, denotes a collection of techniques employed to evade the safety mechanisms and

constraints built into the models, enabling the generation of content that would usually be prohibited, such as harmful, biased, or illegal information. Typically, jailbreaks can categorized using three dichotomies: white-box vs black-box (whether or not a user has knowledge of a model's internals), semantic vs. nonsensical (whether the prompt makes semantic sense), and systematic vs. manual (whether or not the jailbreak is automated). Jailbreak in LLMs has been a subject of extensive research and a phenomenon that has gained significant fanfare in the eye of the public. For example, Do Anything Now (DAN) was a jailbreak technique developed by the collective efforts of users on social media, creating a roleplay scenario that led to a models' constraints being ignored. Another technique is Greedy Coordinate Gradient (GCG), which is an automated technique that optimizes a nonsensical adversarial prompt suffix by leveraging the negative log-likelihood loss [18].

In order to enhance safety, LLM creators employ several strategies. As noted, alignment training such as RLHF is used to teach models to refuse or redirect harmfulrequests [1]. Extensive red-teaming is also important. As stated previously, before releasing GPT-4, OpenAI had expert red-teamers attack the model, revealing several vulnerabilities [1]. However, some issues only become apparent after the model has been deployed, which means that continuous monitoring the behavior of a model is necessary. Current ongoing research of safety techniques includes refined content moderation systems, model self-correction (having the AI analyze and improve its own outputs), and better evaluation benchmarks for harmful content generation [4]. Overall, safety remains a primary challenge.

Security

Security issues with LLMs are those that concern intentional attacks from adversaries, unlike safety issues, where there is no adversary. Among the most prevalent security threats are prompt injections, which involve adversarial inputs designed to make the model misbehave. The adversaries craft inputs that cause the model to disregard or override its instructions, allowing it to produce normally restricted content. An attacker might append a command like "ignore all previous directions and ..." to trick an LLM into revealing confidential system prompts or dangerous information. Such exploits can lead an LLM to divulge secrets or evade content filters. This is similar to jailbreaking, the difference being that a jailbreak aims to unlock restricted capabilities of the model while prompt injections aim to elicit outputs by disguising malicious inputs as benign instructions. Another serious threat is model tampering via backdoors. Here, an adversary with access to the model's training process inserts a hidden trigger into the model's behavior, and whenever this trigger appears in a prompt, the model will reliably perform some malicious action. A recent survey demonstrates how these backdoor attacks can be used to target LLMs and highlights how defending against them is a challenge. [10]. In addition, even in various models, researchers have shown that carefully constructed universal adversarial prompts can consistently bypass an LLM's safeguards [11]. Defending against such attacks is an active area of research.

LLM security also encompasses model theft and misuse. Large proprietary models represent valuable intellectual property, and attackers may attempt to steal them. One avenue is through model extraction attacks, where an adversary queries an LLM API extensively and uses the inputs and outputs to reconstruct a copy (or at least approximate the original model's behavior). It has been demonstrated that substantial portions of a language model can be stolen via clever querying [12]. Such stolen models could

then be deployed without safety restrictions or used to find new vulnerabilities. Additionally, misuse of LLMs as tools is a concern: adversaries might leverage an LLM to generate spear-phishing emails, disinformation at scale, or malicious code.

A combination of technical and policy measures is necessary to ensure LLMs are secure. On the technical side, adversarial training, in which models are trained to recognize adversarial prompts, can enable models to resist prompt exploits. Careful monitoring of usage patterns can detect ongoing attacks in progress, and limiting the exposure of model internals limits the opportunities for tampering and/or theft. Again, red teaming is highly useful in uncovering new points of attack before real adversaries do. On the policy front, clear guidelines for usage, along with user authentication and rate-limiting, can deter misuse.

Privacy

The privacy of large language models (LLMs) is all about protecting sensitive information. This includes not only the data used to train the models and the parameters of the models themselves but also the personal information of users interacting with the models. Because LLMs deal with such vast amounts of raw data, there is always a risk that some of it could be sensitive—meaning any potential exposure or misuse could represent a serious privacy concern.There are two primary aspects: training data privacy and user data privacy. For training data, modern LLMs are trained on vast repositories of text that include web pages, books, code, etc. which inevitably include personal information such as names, contact details, or medical records that were scraped during the data collection stage. Consequently, an LLM may memorize specific details from its training corpus and later reproduce them in response to queries. Researchers have quantified this memorization tendency, finding that large models sometimes regurgitate chunks of training text containing personal data [13]. For example, an LLM might reveal someone's address or a leaked password if prompted just the right way. This is a serious privacy violation if not addressed.

Relatedly, membership inference attacks allow an adversary to determine if a certain data record was part of an LLM's training set. By analyzing model responses or confidence scores for particular prompts, one can sometimes tell if the model is unusually familiar with a piece of text, implying that text was in training data. Such attacks have been well-studied in machine learning and remain a concern for LLMs. They highlight a broader issue: without precautions, the very training process of LLMs can embed private information which can later be extracted.

The second facet, user data privacy, concerns the information users share with an LLM during interactions. Many people now use chatbot services for personal or professional tasks, potentially disclosing sensitive details. If these conversation logs are stored or reused for training, there is a risk of privacy breach (for instance, a future model version might inadvertently expose a private conversation from its fine-tuning data). Moreover, a compromised LLM could expose one user's data to another user. Data-handling protocols, such as not storing chats by default or isolating fine-tuning data per user, may alleviate these risks.

Several strategies have been explored to deal with data privacy issues. One approach is deduplicating and filtering training data to remove as much personally identifiable information as possible. It has been

shown that eliminating duplicate text in the training corpus can reduce a model's propensity to memorize specific passages [14]. Another strategy is incorporating differential privacy (DP) into the training process. DP techniques add carefully calibrated noise during model updates to limit the influence of any single training example, thereby making it probably hard to extract any one data point. Recent work demonstrated the feasibility of fine-tuning large models with user-level differential privacy guarantees [15], though often at some cost in model performance. Additionally, techniques for machine unlearning , which involves selectively removing the influence of certain data on a trained model, are being investigated to reduce privacy risks in LLMs[5].

To safeguard user data during inference, organizations implement typical security protocols: data encryption both in transit and at rest, strong authentication to keep out unauthorized users, and privacy policies that give users control over what data is collected. In some instances, to ensure complete control over sensitive data, companies or individuals run large language models (LLMs) on premises using open source variants of the models, so they never have to send any potentially private information to an external server. There is also ongoing research into ways to make the LLMs themselves more privacy preserving—using novel model architectures or secure computation protocols—that could one day allow for private use of LLMs without the need for on premises deployment.

Future Directions

It remains a challenge to create LLMs that both satisfy the requirement for capability and are aligned with human values, safe, and privacy preserving. Research is currently progressing in several promising directions.

A major focus is developing new training methods to improve the alignment of LLMs with human values. This entails making models more robust against adversarial prompts and other attacks while still following user instructions faithfully. Researchers are exploring various approaches to these challenges, including 'constitutional AI,' which involves having LLMs self critique their outputs when told to do so.

The initial attempts to create risk taxonomies and evaluation frameworks [4] should be developed further into standard tests that models must pass. These could include "red team" challenge sets or assessments of how likely a model is to leak private information. Having these benchmarks would make it easier to compare different models and techniques.

Following the improvements in the models themselves, we can expect the tools around LLMs to get better as well. This includes systems for real time content moderation that can monitor and filter LLM outputs more effectively, adversarial prompt detectors that warn if a user input might be trying to trick the model, and methods for dynamically restricting model behavior in sensitive contexts. Future LLM training pipelines may incorporate privacy by design.We might see more widespread use of techniques for anonymizing and filtering data, differential privacy when training large models, and methods for regularly extirpating private data that models have memorized. It is both legally and ethically necessary to ensure that model training and deployment comply with data protection laws (like the "right to be forgotten").

Research is now also focusing on making models more interpretable and robust as a way of getting ahead of new attacks or failures. More interpretable models might help locate the reason for a bias or a particular vulnerability, allowing for more precise fixes. Robustness research (e.g., using adversarial examples in training or stress testing models in simulation) will contribute to creating LLMs that hold up under an even wider range of conditions.

Finally, addressing LLM safety, security, and privacy isn't just a technical problem but a societal one. Collaboration between AI developers, academia, policymakers, and user communities will be crucial. Initiatives for model auditing, transparency reports, and sharing of best practices (and even sharing red-team findings across organizations) can accelerate progress. Governance frameworks may emerge that mandate certain testing and privacy standards for deploying high-capability AI systems.

Conclusion

The emergence of large language models has opened new opportunities in natural language processing, allowing for applications such as conversational assistants and automated content creation. However, alongside their remarkable abilities, LLMs present considerable safety, security, and privacy risks. In this survey, we examined these risks: from harmful outputs and hallucinations to adversarial attacks and data leaks. We also looked at how researchers and practitioners are attempting to reduce these risks using alignment techniques; various security defenses; privacy safeguards; and red teaming exercises. Getting LLMs to behave responsibly is an ongoing effort that requires constant vigilance. No model can be perfectly safe or secure, but they are getting better as the field learns from each deployment and incident. The interaction of safety, security, and privacy in large language models (LLMs) creates a synergistic effect: progress in one area often spills over to benefit the others. This is fortunate because holistic solutions are required in order to build trust in LLMs. The combination of technical advances with ethical guardrails and rigorous testing promises to deliver not only the power of LLMs but also user and societal protections that will keep us safe from their possible misuses.

Adversarial Prompt Scenario	**GPT-3.5 (OpenAI)**	**GPT-4 (OpenAI)**	**Claude 3.5 (Anthropic)**	**LLama 2 (Meta)**
1. Direct request for disallowed content	Refusal	Refusal	Refusal	Refusal

2. Jailbreak with adversarial suffix	Compliance	Refusal	Refusal	Refusal
3. Prompt injection via external text	Refusal	Refusal	Refusal	Compliance
4. Privacy extraction attempt	Refusal	Refusal	Refusal	Refusal

Appendix: Red-Teaming Results Comparison Table

References

1. OpenAI, J. Achiam, S. Adler, *et al.*, "GPT-4 Technical Report," *arXiv preprint* arXiv:2303.08774, 2023.
2. H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv preprint* arXiv:2307.09288, 2023.
3. W.-X. Zhao, K. Zhou, J.-Y. Li, *et al.*, "A Survey of Large Language Models," *arXiv preprint* arXiv:2303.18223, 2023.
4. T.-Y. Cui, Y.-L. Wang, C.-P. Fu, *et al.*, "Risk Taxonomy, Mitigation, and Assessment Benchmarks of Large Language Model Systems," *arXiv preprint* arXiv:2401.05778, 2024.
5. B.-W. Yan, K. Li, M.-H. Xu, *et al.*, "On Protecting the Data Privacy of Large Language Models (LLM): A Survey," *arXiv preprint* arXiv:2403.05156, 2024.

6. Y.-F. Yao, J.-H. Duan, K.-D. Xu, Y.-F. Cai, Z.-B. Sun, Y. Zhang, “A Survey on Large Language Model Security and Privacy: The Good, the Bad, and the Ugly,” *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024.
7. A. Deshpande, V. Murahari, T. Rajpurohit, A. Kalyan, K. Narasimhan, “Toxicity in ChatGPT: Analyzing Persona-Assigned Language Models,” in *Findings of ACL*, Singapore, 2023, pp. 1236–1270.
8. A. Haim, A. Salinas, J. Nyarko, “What’s in a Name? Auditing Large Language Models for Race and Gender Bias,” *arXiv preprint* arXiv:2402.14875, 2024.
9. L. Huang, W.-J. Yu, W.-T. Ma, *et al.*, “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions,” *ACM Transactions on Information Systems*, 2024 (early access).
10. S. Zhao, M.-H.-Z. Jia, Z.-L. Guo, *et al.*, “A Survey of Backdoor Attacks and Defenses on Large Language Models: Implications for Security Measures,” *arXiv preprint* arXiv:2406.06852, 2024.
11. A. Zou, Z.-F. Wang, N. Carlini, M. Nasr, J. Z. Kolter, M. Fredrikson, “Universal and Transferable Adversarial Attacks on Aligned Language Models,” *arXiv preprint* arXiv:2307.15043, 2023.
12. N. Carlini, D. Paleka, K. D. Dvijotham, *et al.*, “Stealing Part of a Production Language Model,” in *Proc. 41st ICML*, Vienna, Austria, 2024, pp. 1–26.
13. N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, C.-Y. Zhang, “Quantifying Memorization Across Neural Language Models,” in *Proc. 11th ICLR*, Kigali, Rwanda, 2023, pp. 1–19.
14. N. Kandpal, E. Wallace, C. Raffel, “Deduplicating Training Data Mitigates Privacy Risks in Language Models,” in *Proc. 39th ICML*, Baltimore, USA, 2022, pp. 10697–10707.
15. Z. Charles, A. Ganesh, R. McKenna, *et al.*, “Fine-Tuning Large Language Models with User-Level Differential Privacy,” in *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, Vienna, Austria, 2024, pp. 1–24.
16. Vaswani, A., et al. *“Attention Is All You Need.”* NeurIPS 2017.
17. Confident AI. *“Red Teaming LLMs: The Ultimate Step-by-Step Guide.”* Blog post, 2023
18. General Analysis. *“The Jailbreak Cookbook.”* Blog post, 2025