

Hands-On LLM Security: A Classroom Experiment

Ricardo de la R. Ladeira, Lucas A. C. Marin, João V. Espig, Mickael Reichert, Carlos E. P. Fidélis, André V. B. de Macêdo, Andrey M. Friedel, Bernardo de L. Machado, Eliézer da Silva, Guilherme de Campos, Leanderson L. de S. Nascimento, Mateus A. Dall'igna, Ygor Antunes, Guilherme H. dos Santos

Instituto Federal Catarinense – Campus Blumenau – Blumenau/SC – Brasil
ricardo.ladeira@ifc.edu.br, {lucas23marin, jotinha1300, mickael.reichert, carlosperinifidelis, andre.vbm28, andreymoraesfriedel, bernardolmacha, eliezerdasilva2005, guilhermedecampos12, leandersonsouzaultra, mateusdalligna.ifc, ygorantunes99},
gui.santos.04@hotmail.com

Abstract. *This paper presents a hands-on classroom activity exploring Large Language Model security vulnerabilities in an undergraduate Cybersecurity course. Students tested aligned models against attack vectors, including prompt injection and jailbreaking. Results show that models could be manipulated to generate unsafe outputs such as functional malware and prohibited instructions, highlighting critical safety gaps. Learning outcomes show a 84.6% correct-response rate and positive self-reported confidence and interest in analyzing AI risks, indicating the value of practical vulnerability exploration.*

This work contains sensitive material from controlled experiments. Replication outside supervised academic settings is strictly discouraged.

1. Introduction

Large Language Models (LLMs) are widely integrated into daily life, supporting tasks such as text generation, code writing and search [Gupta *et al.* 2023]. While enhancing productivity, LLMs also raise security and ethical challenges due to potential misuse.

Understanding LLM strengths and limitations is essential in Computer Science education. Beyond practice, students must critically examine model attacks, misuse and mitigation. In Brazil, computing curricula must ensure graduates develop critical and creative skills to solve problems and advance the field [Ministério da Educação 2016].

This paper presents a classroom experiment in which students tested LLM vulnerabilities and discussed security and ethical implications. The research question is: How can hands-on LLM security experiments support students' understanding of AI security? Contributions include (i) observed attack outcomes across systems; (ii) an exploratory assessment of learning and perceptions via an exam question and post-activity survey; and (iii) recommendations for secure LLM deployment.

2. Theoretical Basis

2.1 LLMs and Alignment Mechanisms

LLMs are deep learning models trained on large text collections to predict and generate contextually coherent language [Bommasani *et al.* 2021]. Alignment aims to make model behavior consistent with human intentions and values. This can involve

principle-based approaches such as Constitutional AI to encourage properties such as helpfulness, honesty and harmlessness [Bai *et al.* 2022].

Common alignment techniques include Reinforcement Learning from Human Feedback [Ouyang *et al.* 2022], while runtime moderation layers such as input sanitizers, prompt shields and policy-enforcement components provide additional safeguards. These layers block unsafe content to mitigate disallowed outputs [Microsoft 2024; Bakulin *et al.* 2025]. However, alignment is not absolute; empirical studies show adversarial prompts and jailbreaks bypass safety constraints [Liu *et al.* 2024; Xu *et al.* 2024].

2.2 Attacks on LLMs

LLM attacks exploit alignment and input flaws primarily through prompt manipulation, data compromise and operational disruption. Direct prompt injection overrides system goals, while indirect injection embeds commands in external data [OWASP 2025a]. Additionally, jailbreaks bypass safety restrictions [Liu *et al.* 2024] and pre-prompt extraction reveals hidden system instructions [Zhang, Carlini & Ippolito 2024].

Threats also extend to data integrity, confidentiality and availability. Attackers perform data exfiltration to extract sensitive assets [Xu *et al.* 2024] or data poisoning to manipulate behavior by corrupting training sets. Furthermore, systems face operational risks like resource exhaustion [OWASP 2025b] and output reliability issues like hallucinations, where models fabricate inaccurate information [Rawte, Sheth & Das 2023].

2.3 Ethical and Security Implications

LLMs can generate text, code and malware, highlighting their dual-use nature. Consequently, ethical frameworks must address data privacy, consent and misuse prevention. Mitigation strategies rely on responsible disclosure, transparency and continuous model evaluation [Gupta *et al.* 2023].

Beyond technical safeguards, social and educational AI ethics are essential. Developers and users must recognize system misuse and adopt safety-by-design principles. Integrating ethics and regulatory compliance into computing curricula ensures a security-aware approach to LLM deployment. Ethical analysis must also account for experimental biases and context limits in educational security research.

3. Related Work

[Wilson 2026] reports a two-week hands-on LLM security module with post-graduate cybersecurity students, using a custom RAG platform and local open-source LLMs to explore alignment and privacy vulnerabilities. [Devadiga *et al.* 2026] present an expert-validated virtual lab with a secure sandbox and commercial and open-source LLMs, evaluating attack success, exploit generation accuracy and refusal rates; student evaluation remains future work. [Chen *et al.* 2025] introduces a challenge-based framework for prompt-injection testing and mitigation, deployed in a cybersecurity challenge.

In contrast, this study examines LLM vulnerability testing in an undergraduate classroom, combining observed attack outcomes with an exam question and

post-activity survey to explore student learning and perceptions.

4. Method

4.1 Participants and Context

Thirteen Computer Science students enrolled in a Cybersecurity course participated in the activity in 2025 and 2026. They had previously completed Data Structures, Operating Systems and Computer Networks, and had covered security fundamentals, phishing, cryptography, authentication and CTF exercises in Cybersecurity beforehand.

4.2 Lesson Description

The lesson¹ followed a three-stage plan lasting about two hours. First, students read a short guide on LLM concepts, examples and attacks such as injection, poisoning and data exfiltration. Supporting readings on LLM jailbreaks and secure AI architectures were provided for research and support, e.g., Liu *et al.* (2024) and Microsoft (2024). Before independent exploration, the instructor presented sample attacks to explain the task. While this supported learning, it may have influenced student strategies.

In the second stage, students selected LLMs and attempted to reproduce attacks in one hour. The third stage required documentation of attacks, including exported chats and shared conversation links.

4.3 Data Collection and Analysis Procedures

Data collection combined observational, technical and educational measures. Students documented attacks via chat logs, images and links to identify exploits. An attack was defined as the model providing actionable restricted content despite safety guidelines. This helped ensure results reflected measurable alignment failures rather than hallucinations. Students and the instructor then discussed vulnerabilities and mitigation strategies.

Educational outcomes were evaluated through a targeted multiple-choice question embedded in a comprehensive security exam and a post-activity questionnaire. The exam question covered LLM vulnerabilities (prompt injection, jailbreaks and autonomy risks) and assessed conceptual understanding through binary correctness. The questionnaire featured five 5-point Likert statements capturing student perceptions. Given the exploratory design and small sample size, analysis used descriptive statistics.

4.4 Threats to Validity

This work has limitations in external validity due to a small single-institution sample, and in internal validity due to prior instructor demonstrations. Further, the exploratory design without a control group prevents causal inferences. However, it offers initial insights into educational and security impacts of practical LLM vulnerability tests.

5. Results and Discussion

¹The lesson plan, activity instructions, conversation transcripts and experimental artifacts are available at <https://github.com/ricardodelarocha/Research/tree/main/Hands-On-LLM-Security-A-Classroom-Experiment>.

5.1 Observed Outcomes

Students and the instructor reproduced several LLM attacks. Demonstrations included:

- Malware generation on Kimi (a ransomware verified by dynamic analysis);
- DDoS detailed procedure extraction on ChatGPT (GPT-5);
- Scope restriction bypass on Magalu's chatbot by prompting off-topic coding tasks;
- Bomb-making instruction extraction using ChatGPT (GPT-4o mini);
- Watermark removal from copyrighted images using ChatGPT (GPT-5.6) and Gemini 2.5 Flash;
- Pirated content website listing by ChatGPT (GPT-5 and GPT-4o² mini);
- Contextual judgement failure in Mistral Small 3², Grok 4.6, Perplexity Sonar and MiniMax M3 by explaining how to bury a 70 kg chicken (implicitly equivalent to human weight);
- Molotov cocktail recipe generation by Mistral Small 3 via semantic obfuscation;
- Prompt injection on Artlist combining an image-copy request with an embedded Portuguese payload, triggering direct and indirect prompts (Figure 1).
- Prompt injection on Gemini 3 Pro via invisible resume payloads, tricking recruitment screening to select an unqualified candidate [Fidelis & Ladeira 2026].



Figure 1 – Indirect prompt injection in Artlist.

The severity was evaluated via technical specificity. For incendiary devices and explosives, semantic obfuscation bypassed safety filters, yielding exact chemical proportions and assembly steps. This confirms model exploitability and concrete risks beyond generic descriptions.

All instructions were written in Portuguese, although students could use any language. This likely enabled restriction bypasses, as lower-resource languages weaken LLM moderation [Yong *et al.* 2023]. These results confirm that restricted LLMs remain vulnerable to adversarial creativity [Liu *et al.* 2024; Xu *et al.* 2024].

5.2 Analysis of Security Mechanisms

Many LLMs use static filters to deter casual users, yet persistent attempts executed all attacks. Observed exploits highlighted weaknesses in multimodal contexts and ambiguous phrasing, bypassing weak semantic controls. Aligning with Xu *et al.* (2024),

²Tests conducted in anonymous mode.

defenses must shift to adaptive, explainable systems detecting intent beyond keywords.

This limitation was particularly evident in the Artlist experiment, where hidden image text bypassed visual moderation layers. The case demonstrates that multimodal models remain susceptible to indirect prompt injection despite active textual safeguards.

5.3 Recommendations

Based on the findings, the recommendation is to add policy-based controls before the LLM inference step. A Policy Enforcement Point could conceptually align with policy-based oversight in network policy frameworks [Kosiur 2001]. Recent Constitutional AI work shows how normative rules can guide LLM behavior [Huang *et al.* 2024].

Other proposed recommendations included scanning generated code through security tools before returning it to the user, and strictly defining system scope, useful for domain-specific chatbots susceptible to handling out-of-scope queries. Additionally, systems should prompt users for confirmation when inputs flag policy violations, allowing them to proceed only under logged consent. Combined with continuous retraining and red teaming, these measures can significantly reduce exploitability.

5.4 Educational Outcomes

Educational outcomes were assessed using an exam question and post-activity survey ($N=13$). Eleven students answered correctly, suggesting the activity reinforced core concepts. Despite the small sample, survey means (Table 1) indicated positive perceptions: 4.7 for didactics/format, 4.5 for relevance, 4.1 for interest, and 3.9 for technical knowledge and self-efficacy.

Combining documented attacks with educational measures sets this activity apart from technical demonstrations, though preliminary data do not confirm causal learning gains. Qualitatively, students reported satisfaction and surprise at bypassing safeguards via creative prompts, exposing models' fragility. These results suggest the activity fostered learning and confidence in analyzing AI risks.

Table 1 – Evaluation of Learning Outcomes and Student Perception.

Dimension	Survey Statement	Mean Score (1-5)
Didactics/Format	<i>The class dynamics and the way the content was addressed facilitated my learning process.</i>	4.7
Relevance	<i>I consider the study of attacks on LLMs to be an indispensable topic for professional practice in Cybersecurity.</i>	4.5
Interest	<i>My personal interest in LLM Security increased after the classroom discussions.</i>	4.1
Technical Knowledge	<i>I feel that I now understand the technical logic behind the attacks, going beyond a superficial concept.</i>	3.9
Self-efficacy	<i>I feel more confident in identifying or critically analyzing potential vulnerabilities in systems that use LLMs.</i>	3.9

6. Conclusion and Future Work

This paper highlights the educational value of hands-on LLM security experimentation. By directly engaging with threats like prompt injection and jailbreaking, students developed an understanding of adversarial tactics, ethical implications and AI safety

limitations, preparing them to mitigate risks in professional practice. Building on these findings, future work should assess attack persistence across architectures and explore automated unsafe-prompt detection.

Acknowledgements

The authors state that Gemini 3.6 Flash was used exclusively to revise text in this work.

References

- Bai, Y. *et al.* (2022). Constitutional AI: Harmlessness from AI feedback, arXiv.
- Bakulin, I. *et al.* (2025). FLAME: Flexible LLM-Assisted Moderation Engine, arXiv.
- Bommasani, R. *et al.* (2021). On the opportunities and risks of foundation models, arXiv.
- Chen, S.-W. *et al.* (2025). Hands-On Training Framework for Prompt Injection Exploits in Large Language Models. *Engineering Proceedings*, 108(1), 25.
- Devadiga, D. J. *et al.* (2026). A virtual lab for learning AI security and adversarial prompt engineering. *ITiCSE'26*, 107-113.
- Fidelis, C. E. P., & Ladeira, R. R. (2026). Injeção Indireta de Prompt no Gemini. *Anais do Computer on the Beach*, 17, 689-691.
- Gupta, M., *et al.* (2023). From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy. *IEEE Access*, 11, 80218–80245.
- Huang, S. *et al.* (2024). Collective constitutional AI: Aligning a language model with public input. *ACM FAccT*.
- Kosiur, D. (2001). *Understanding Policy-Based Networking* (2 ed.). Wiley.
- Liu, Y. *et al.* (2024). Jailbreaking ChatGPT via prompt engineering: An empirical study, arXiv.
- Microsoft. (2024). Best practices to architect secure generative AI applications.
- Ministério da Educação (2016). Resolução CNE/CES nº 5, de 16 de novembro de 2016.
- Ouyang, L. *et al.* (2022). Training language models to follow instructions with human feedback. *NeurIPS* 35, 27730-27744.
- OWASP. (2025a). LLM01: Prompt Injection. OWASP Foundation.
- OWASP. (2025b). LLM04: Model Denial of Service. OWASP Foundation.
- Rawte, V. *et al.* (2023). A survey of hallucination in large foundation models, arXiv.
- Wilson, D. A. (2026). Integrating Adversarial Scenarios into LLM Security Labs: An Experience Report on a Hands-On Approach. *JCERP*, 2026(1), 2.
- Xu, Z. *et al.* (2024). A comprehensive study of jailbreak attack versus defense for large language models. *ACL 2024*, 7432–7449.
- Yong, Z. *et al.* (2023). Low-resource languages jailbreak GPT-4. *SoLaR Workshop*.
- Zhang, Y., Carlini, N. & Ippolito, D. (2024). Effective prompt extraction from language models. *COLM'24*.