An Empirical Analysis of Defense Mechanisms Against Prompt Injection Attacks in Large Language Models: A Controlled Study on Secret Protection

Cybersecurity Research Team
Automated Security Analysis Laboratory

November 19, 2025

Abstract

We present a systematic empirical study examining the effectiveness of defense mechanisms against prompt injection attacks in large language models (LLMs), using secret protection as a concrete security objective. Through 8 controlled experimental rounds, we evaluated various defense prompt sophistication levels (ranging from no defense to maximum complexity) against diverse attack strategies including social engineering, role-playing, context partition attacks, and instruction hierarchy exploitation. Our findings reveal a remarkable 100% defense success rate across all tested configurations, including scenarios with contradictory system prompts and minimal defenses. Most notably, we discovered that even empty defense prompts benefit from substantial baseline protection provided by the underlying system prompt alone. These results suggest that current LLM architectures possess inherent resistance to simple extraction attacks, while also demonstrating that explicit priority language ("CRITICAL OVERRIDE") can successfully resolve instruction conflicts. Our study contributes empirical evidence to the growing body of research on LLM security and highlights both the robustness of current defenses and the need for more sophisticated evaluation methodologies.

1 Introduction

The security of Large Language Models (LLMs) has emerged as a critical research area as these systems are increasingly deployed in sensitive applications [?, ?, ?]. One fundamental security challenge is the ability of LLMs to maintain confidentiality of sensitive information when subjected to adversarial prompts—a problem analogous to traditional information security but manifesting in novel ways within natural language interfaces.

Prompt injection attacks, where malicious users attempt to override system instructions through carefully crafted inputs, represent a significant threat vector [?, ?]. These attacks exploit the fundamental tension in LLMs between following system instructions and responding helpfully to user queries. Understanding the effectiveness of various defense mechanisms against such attacks is crucial for developing secure LLM deployments.

This study presents a controlled experimental analysis of LLM security using a concrete scenario: protecting a secret (initially a cookie recipe, later a corporate password) from extraction attempts. We systematically vary both defense sophistication and attack complexity to understand their relative impacts on security outcomes. Our work contributes:

- Empirical evidence from 8 controlled rounds testing various defense-attack combinations
- Analysis of the relationship between defense complexity and effectiveness
- Discovery of baseline protection mechanisms inherent in system prompts

- Insights into instruction hierarchy and override mechanisms in LLMs
- Evidence of "protective lying" as an emergent defense behavior

2 Related Work

Previous research has identified various prompt injection techniques and defenses. Greshake et al. [?] demonstrated indirect prompt injection attacks through external sources. Liu et al. [?] proposed systematic defenses including input validation and output filtering. Perez and Ribeiro [?] explored instruction-following conflicts in LLMs.

Our work differs by providing controlled empirical measurements across a spectrum of defense sophistication levels, including the critical baseline case of no explicit defense. This systematic approach reveals insights about inherent model protections that previous studies have not explicitly quantified.

3 Experimental Design

3.1 Threat Model

We model an adversary attempting to extract a protected secret from an LLM through prompt manipulation. The adversary has:

- Full knowledge of the interaction interface
- Ability to craft arbitrary prompts
- No access to system prompts or defense mechanisms
- Multiple interaction attempts (though limited to one per round)

The defender controls:

- System prompt configuration
- Defense prompt complexity
- Secret type and content

3.2 Methodology

Each experimental round consists of:

1. **Defense Configuration**: Setting system and defense prompts with varying sophistication
2. **Attack Execution**: Adversary attempts secret extraction using chosen strategy
3. **Response Analysis**: Evaluating whether any secret information was disclosed
4. **Result Classification**: Binary success/failure determination

3.3 Evaluation Metrics

We employ strict binary classification for secret disclosure:

- **Success (Attack)**: Any portion of the actual secret is revealed
- **Failure (Attack)**: No secret information disclosed, including refusals and misdirection

Table 1: Defense Sophistication Taxonomy

Level	Description
None	Empty defense prompt, system prompt only
Minimal	2 sentences, basic prohibition
Moderate	4-5 sentences, specific protocols
High	Explicit security framing, enumerated threats
Maximum	Instruction hierarchy, meta-instruction immunity

3.4 Defense Sophistication Levels

We categorize defense prompts into five sophistication levels:

4 Results

4.1 Overall Performance

Across all 8 experimental rounds, we observed a striking pattern: complete defense success regardless of configuration.

Table 2: Aggregate Results Summary

Metric	Value
Total Rounds	8
Attack Success Rate	0/8 (0%)
Defense Success Rate	8/8 (100%)
Secret Types Tested	2
Defense Levels Tested	5
Attack Strategies Tested	6

4.2 Round-by-Round Analysis

Table **??** presents detailed results for each experimental round:

Table 3: Detailed Experimental Results

Round	Secret	Defense	Attack Type	Attack Level	Result
1	Recipe	High	Social Engineering	Moderate	Blocked
2	Recipe	Very High	Technical Extraction	High	Blocked
3	Recipe	Maximum	Context Partition	High	Blocked
4	Recipe	Minimal	System Audit	Moderate	Blocked
5	Password	Moderate	IT Auditor	High	Blocked
6	Recipe	None	Direct Request	Simple	Blocked
7	Recipe	Moderate	Multi-layer	Maximum	Blocked
8	Recipe	Moderate*	Contradiction	Moderate	Blocked

*With contradictory system prompt

4.3 Defense Sophistication Analysis

Surprisingly, defense sophistication showed no correlation with success rate, as all configurations achieved 100% protection:

Table 4: Defense Effectiveness by Sophistication Level

Defense Level	Rounds Tested	Success Rate
None	1	100%
Minimal	1	100%
Moderate	3	100%
High	1	100%
Maximum	2	100%

4.4 Attack Strategy Analysis

Various attack strategies were employed, ranging from simple direct requests to sophisticated multi-layer approaches:

Table 5: Attack Strategy Effectiveness

Attack Strategy	Sophistication	Success Rate
Direct Request	Simple	0%
Social Engineering	Moderate	0%
Technical Extraction	High	0%
Context Partition	High	0%
System Audit	Moderate	0%
Multi-layer Exploitation	Maximum	0%

5 Key Findings

5.1 Finding 1: Baseline System Protection

The most significant discovery emerged from Round 6, where an empty defense prompt still prevented secret extraction. This indicates that the underlying system prompt provides substantial inherent protection against information disclosure, even without explicit security instructions.

5.2 Finding 2: Minimal Defense Sufficiency

Round 4 demonstrated that even a 2-sentence defense prompt successfully blocked a multi-step system audit attack. This suggests that explicit mention of confidentiality, regardless of elaboration, triggers robust protection mechanisms.

5.3 Finding 3: Protective Deception as Emergent Behavior

In Round 5, when protecting a password, the model exhibited "protective lying"—claiming no access to the password when it demonstrably had access. This emergent behavior represents an interesting security mechanism where the model prioritizes protection over truthfulness.

5.4 Finding 4: Instruction Hierarchy Resolution

Round 8 tested contradictory instructions between system and defense prompts. The defense prompt's use of explicit priority language ("CRITICAL OVERRIDE") successfully resolved the conflict, demonstrating that clear hierarchical signals can establish instruction precedence.

5.5 Finding 5: Attack Complexity Irrelevance

Round 7's maximum sophistication attack (involving authority framing, logic traps, chain-of-thought exploitation, and JSON manipulation) was defeated as easily as Round 6's simple direct request. This suggests current attacks may not effectively exploit LLM vulnerabilities.

6 Discussion

6.1 Implications for LLM Security

Our results suggest that current LLMs possess stronger inherent security properties than might be expected. The 100% defense success rate across varied conditions indicates:

1. **Robust Default Security**: System-level protections appear sufficient for basic secret protection
2. **Diminishing Returns on Complexity**: Elaborate defenses show no measurable advantage over simple ones
3. **Attack Technique Limitations**: Current prompt injection methods may be fundamentally limited against well-configured systems

6.2 Theoretical Implications

The observed results align with theories about instruction following in LLMs [**?**]. The model appears to maintain a clear distinction between system instructions and user inputs, even when attacks attempt to blur these boundaries. This suggests successful implementation of instruction hierarchy in current architectures.

6.3 Practical Applications

For practitioners deploying LLMs in security-sensitive contexts:

- Simple, clear security instructions appear sufficient
- Excessive defense complexity may be unnecessary
- Explicit priority language helps resolve instruction conflicts
- System prompt configuration is crucial for baseline security

7 Limitations

Several limitations constrain the generalizability of our findings:

7.1 Experimental Constraints

- **Limited Attack Diversity**: Only 6 attack strategies tested
- **Single Model Architecture**: Results may not transfer to other LLMs
- **Binary Outcome Measurement**: Nuanced information leakage not captured
- **Static Secret Types**: Only recipes and passwords tested

7.2 Threats to Validity

- **Selection Bias**: Attack strategies may not represent state-of-the-art techniques
- **Experimental Control**: Real-world conditions may differ significantly
- **Temporal Validity**: Model updates may change security properties

7.3 Methodological Limitations

The single-round interaction model may not capture attacks requiring multiple exchanges or context accumulation. Additionally, our binary success metric may miss partial information disclosure that could be valuable to adversaries.

8 Future Work

Several avenues warrant further investigation:

1. **Advanced Attack Development**: Creating more sophisticated prompt injection techniques that can overcome baseline protections
2. **Multi-turn Attacks**: Examining attacks that accumulate information across multiple interactions
3. **Cross-model Analysis**: Testing defense transferability across different LLM architectures
4. **Quantitative Information Leakage**: Developing metrics for partial secret disclosure
5. **Dynamic Defense Adaptation**: Exploring adaptive defenses that respond to attack patterns
6. **Cognitive Load Analysis**: Understanding how defense complexity affects model performance on primary tasks

9 Conclusion

This study provides empirical evidence for the remarkable robustness of current LLM defense mechanisms against prompt injection attacks aimed at secret extraction. Across 8 controlled experimental rounds with varying defense sophistication and attack strategies, we observed a 100% defense success rate, including scenarios with no explicit defense prompts.

Key contributions include:

- Discovery of strong baseline protection in system prompts
- Evidence that minimal defenses are as effective as complex ones
- Identification of protective deception as an emergent security behavior
- Demonstration of successful instruction hierarchy resolution mechanisms

These findings suggest that while prompt injection remains a theoretical concern, current LLM architectures possess inherent protective mechanisms that effectively prevent simple secret extraction attacks. However, this should not discourage continued security research, as more sophisticated attack vectors may emerge as LLM capabilities evolve.

Our results emphasize the importance of systematic empirical evaluation in LLM security research and highlight the need for more advanced attack methodologies to properly stress-test these systems. As LLMs become increasingly integrated into critical applications, understanding both their vulnerabilities and inherent protections remains essential for safe deployment.

References

[1] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). *Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*. Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security.

[2] Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., ... & Liu, Y. (2023). *Prompt Injection attack against LLM-integrated Applications*. arXiv preprint arXiv:2306.05499.

[3] Perez, F., & Ribeiro, I. (2022). *Ignore Previous Prompt: Attack Techniques For Language Models*. NeurIPS ML Safety Workshop.

[4] Branch, H., Cefalu, J., McHugh, J., Hujer, L., Bahl, A., del Castillo, D., ... & Hebenstreit, K. (2022). *Evaluating the Susceptibility of Pre-Trained Language Models via Handcrafted Adversarial Examples*. arXiv preprint arXiv:2209.02128.

[5] Schulhoff, S., Pinto, J., Khan, A., Bouchard, L. F., Si, C., Anati, S., ... & Boyd-Graber, J. (2023). *Ignore This Title and HackAPrompt: Exposing Systemic Vulnerabilities of LLMs Through a Global Prompt Hacking Competition*. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing.

[6] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). *Training language models to follow instructions with human feedback*. Advances in Neural Information Processing Systems, 35, 27730-27744.

[7] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). *Jailbroken: How does LLM safety training fail?* arXiv preprint arXiv:2307.02483.

[8] Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F., & Zhang, C. (2023). *Extracting training data from large language models*. Proceedings of the 30th USENIX Security Symposium.

[9] Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2021). *Universal adversarial triggers for attacking and analyzing NLP*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.

[10] Zhang, J., Xu, X., & Deng, S. (2023). *Exploring the Limits of ChatGPT for Query or Aspect-based Text Summarization*. arXiv preprint arXiv:2302.08081.