Evaluation of Current Threat Vectors Posed by Prompt Injection Against Contemporary Large Language Models

A Research Proposal for MSc Cybersecurity, Privacy and Trust

Christopher Boylan – C00326765@setu.ie

RESEARCH MOTIVATION

Why This Research Matters

Large Language Models (LLMs) are rapidly being integrated into enterprise applications, from customer service chatbots to code generation tools. However, these powerful systems remain vulnerable to prompt injection attacks—a critical security gap that threatens data integrity, user privacy, and system reliability. Despite the urgency, there is a significant lack of independent, empirical research comparing the effectiveness of different mitigation strategies.

OWASP ranks prompt injection as the #1 security risk for LLM applications (2025)

OWASP GENAI SECURITY FRAMEWORK

Prompt Injection

Critical vulnerabilities identified by OWASP for GenAI systems

#1 CRITICAL THREAT

Prompt Injection

Malicious inputs that manipulate LLM behavior by overriding system instructions, leading to unauthorized actions, data leakage, privilege escalation, and complete system compromise. This represents the most severe and prevalent threat to LLM security.

Sensitive Information Disclosure

Unintended exposure of confidential data through LLM responses

Insecure Output Handling

Improper validation of LLM-generated content leading to vulnerabilities

ATTACK SURFACE ANALYSIS

Threat Vectors Under Investigation

Direct Injection

Adversarial prompts crafted by users to manipulate system behavior through jailbreaks, instruction override, and direct exploitation of model vulnerabilities

Indirect Injection

Malicious instructions embedded in third-party data sources including emails, websites, documents, and external APIs that poison the LLM's context

Mitigation Strategies: Multi-Layer Defense

Input Filtering
INPUT-LEVEL DEFENSE

Prompt Hardening
INPUT-LEVEL DEFENSE

Instruction Hierarchy
MODEL-LEVEL DEFENSE

Separate Guardrail Model
MODEL-LEVEL DEFENSE

Output Validation & Filtering
OUTPUT-LEVEL DEFENSE

Architectural Separation
SYSTEM-LEVEL DEFENSE

RESEARCH DESIGN

Rigorous Experimental Methodology

1. **Quantitative Experimental Design**
   Controlled testbed with contemporary LLM APIs (OpenAI, Anthropic, Google)
2. **Standardized Attack Corpus**
   Validated benchmarks from academic research and industry sources
3. **Comparative Evaluation**
   Systematic testing of mitigation effectiveness across attack vectors
4. **LLM-as-a-Judge Scoring**
   Objective, consistent evaluation methodology for attack success rates

RESEARCH IMPACT

Expected Contributions to the Field

Empirical Evidence: First independent comparative analysis of mitigation effectiveness across different attack vectors and contemporary LLM architectures

Practical Guidance: Evidence-based security recommendations for developers and practitioners deploying LLM applications in production environments