Seeing Through the Noise: Evaluating LLM Effectiveness in De-obfuscating Malicious Scripts

Author: Student Name, email@address (not .edu email alias)

Advisor: *Faculty Advisor Name*

*Accepted:*

# Abstract

* The problem of malicious script obfuscation
* Why traditional static/dynamic techniques have limitations
* How LLMs may offer a scalable or complementary alternative
* Outcomes of the research

## 1. Introduction

* Background on script obfuscation in malware

1.2. Review of prior work

* Overview of Patsakis et al. (2024): LLMs vs. Emotet scripts

### ****1.3. Key Technical Concepts****

* Introduce XOR encryption
  + What it is (e.g., byte-wise key-based transformation)
  + Why it's used by malware authors
  + Why it is hard for LLMs: no execution context, hidden semantics
* Other obfuscation methods? (e.g., base64, string splitting)

## 2. Research Method

### 2.1. Methodological Approach

* Comparative empirical evaluation
* Real-world dataset
* LLMs: GPT-4o, Gemini 2.5 Pro, Llama 4, Codestral 2508

### 2.2. Dataset Design

* XXX obfuscated PowerShell scripts from malware campaigns
* Criteria: known URLs/domains in decoded payloads

### 2.3. Evaluation Metrics

* URL/domain extraction accuracy
* Hallucination rate
* Structured prompt response consistency

### 2.4. Tools and Environment

* Discuss APIs, remote models and local models used and why
* Prompt templates used

## 3. Findings and Discussion

### 3.1. Accuracy of URL and Domain Extraction

* Discussion of accuracy and hallucinations

### 3.2. LLM Prompt Behavior

* Observations on temperature, model types (chat, code, thinking), and parameters
* Limitations of local model input length and training

### 3.4. Realism of Evaluation

* Challenges in using real malware vs synthetic obfuscation

## 4. Recommendations and Implications for Future Research

### 4.1. Recommendations for Practice

* Use LLMs as a fallback or enrichment in automated threat intelligence pipelines
* Validate LLM output

### 4.2. Implications for Future Research

* Can accuracy be improved with fine-tuning?
* Effectiveness across other scripting languages (e.g., VBA, PHP, JavaScript)

## 5. Conclusion

* Highlight the importance of de-obfuscation in malware analysis
* Summarize findings
* Further research/testing and operational integration?

## 6. References