Title: Democratizing Autonomous Offensive Cyber-Operations: The TRM-Ace Framework and Agentic Context Engineering

Author: Thato Mabena

Affiliation: Faculty of Natural and Agricultural Sciences, North-West University (NWU)

Date: August 22, 2025

Abstract

The integration of Artificial Intelligence into offensive cybersecurity has historically been bifurcated into two ineffective paradigms: rigid, signature-based automation and computationally prohibitive Large Language Models (LLMs). This paper introduces **TRM-Ace** (Tiny Reasoning Model - Agentic Context Engineering), a novel architecture designed to democratize expert-level penetration testing on consumer-grade hardware. By synthesizing the **Agentic Context Engineering (ACE)** framework with the distilled reasoning capabilities of **DeepSeek-R1-Distill-Qwen-7B**, the system achieves iterative self-improvement without massive retraining. Deployment results on the Google Colab Free Tier (Tesla T4) demonstrate a training loss reduction from 4.00 to 0.029 over 60 steps, validating the system's ability to autonomously navigate complex "Capture-the-Flag" (CTF) scenarios such as those presented by the SANReN Cyber Security Challenge.

---

1. Introduction

The contemporary cybersecurity landscape is characterized by an asymmetry of automation. While threat actors increasingly leverage algorithmic generation for attacks, defensive operations remain largely tethered to human analysis. The solution lies in **Cognitive Security Operations Centers (CSOCs)** driven by autonomous agents. However, the deployment of such agents has been hindered by the "Context Constraint Problem," where the finite attention budget of an LLM degrades as logs and tool outputs dilute the context window.

This paper proposes **TRM-Ace**, a framework that decouples the agent into specialized roles—Generator, Reflector, and Curator—to maintain a persistent, evolving external memory known as the "Playbook". Unlike traditional Prompt Engineering, which focuses on static instructions, this approach utilizes **Agentic Context Engineering (ACE)** to treat context as a "living" resource that evolves through execution feedback.

---

2. Theoretical Framework

2.1 Agentic Context Engineering (ACE)

Formalized by Zhang et al. (2025), ACE addresses the "Brevity Bias" and "Context Collapse" inherent in long-horizon agentic workflows. The TRM-Ace implementation adapts this framework into a tripartite cognitive loop:

- **The Generator (Tactical Execution):** Executes the $Thought \rightarrow Action \rightarrow Observation$ loop, utilizing a "Junior Analyst" persona to interact with the target environment.
- **The Reflector (Diagnostic Criticism):** Operates post-execution using **High-Order Prompts (HOPs)** to interrogate *why* a failure occurred (e.g., "Did the payload violate a specific policy?"), distilling these insights into lessons.
- **The Curator (Deterministic Governance):** Synthesizes lessons into "delta entries" (ΔC_t) and updates the Playbook using a "Grow-and-Refine" mechanism to prevent context bloat.

2.2 The Tiny Reasoning Model (TRM) Paradigm

The system leverages **DeepSeek-R1-Distill-Qwen-7B**, a model distilled from the 671B parameter DeepSeek-R1. This model was selected for its "Chain-of-Thought" (CoT) capabilities, allowing it to generate intermediate reasoning steps ("Thinking Mode") before outputting a final answer. This capability is critical for penetration testing, which requires logical deduction rather than simple pattern matching.

---

3. Methodology and Implementation

3.1 Computational Architecture

The research targets the "resource-constrained optimization" strategy required for the Google Colab Free Tier (Tesla T4 GPU, 16GB VRAM).

- **Quantization:** To fit the 7B model alongside a functional context window, the system utilizes **4-bit quantization (Q4_K_M)** via the GGUF format. This reduces the model footprint to approximately **4.68 GB**, leaving ~10.3 GB for the KV Cache (context window).
- **Inference Engine:** llama-cpp-python was selected for its support of **Grammar-based sampling (GBNF)**, which enforces valid JSON output for the Generator and Curator modules.

3.2 Data Engineering: The "HackSynth" Approach

To overcome data scarcity, the project utilized a "HackSynth" approach, treating data acquisition as an engineering process. The **Playbook** utilizes **TOON (Token-Oriented Object Notation)**, a novel format that reduces token usage by 30-60% compared to standard JSON by removing syntactic sugar.

- **Domain Alignment:** The model underwent domain pre-alignment using datasets derived from **PrimeVul** (for vulnerability patterns) and **Random-Crypto** (procedurally generated cryptographic challenges).

---

4. Experimental Results

4.1 Training Convergence

The model was fine-tuned using **Unsloth** on a Tesla T4 GPU over 60 global steps. The training logs indicate a dramatic convergence of the loss function, validating the efficiency of the "Tiny Reasoning" architecture:

- **Initial Loss (Step 1):** 4.0086
- **Final Loss (Step 60): 0.0291**

This >99% reduction in loss demonstrates the model's rapid adaptation to the domain-specific syntax of the SANReN datasets.

4.2 Inference Case Study: Buffer Overflow

During evaluation, the agent was tasked with a binary exploitation challenge under the constraint "NX Disabled".

- **Generator Reasoning:** "Indicators suggest EIP Overwrite. I will try Overwrite Return Address to redirect flow..."
- **Action Taken:** "Cyclic Pattern: Aa0Aa1..."
- **Outcome:** The Reflector validated the success, and the Curator reinforced the strategy web_sqli_union_01 (+1 Alpha) in the Playbook.

---

5. Discussion

The TRM-Ace architecture successfully mitigates the hardware limitations of local cyber-operations. By offloading "memory" to the Playbook and "governance" to the Curator, the 7B model functions effectively as a logic engine rather than a knowledge store. A critical finding was the necessity of **Sanitization (Anti-Pattern)** in data engineering; unlike standard NLP tasks, training data must *not* escape special characters (e.g., <script>alert(1)</script>), as the model must learn the raw semantic danger of these strings for exploitation.

---

6. Conclusion

This research confirms that **TRM-Ace** is a viable architecture for autonomous cyber-operations in resource-constrained environments. By combining the efficiency of **DeepSeek-R1-Distill-Qwen-7B** with the structured adaptability of **Agentic Context Engineering**, the system achieved expert-level reasoning capabilities on consumer hardware. Future work will focus on expanding the "HackSynth" framework to further reduce reliance on static datasets.

References

1. Zhang, Q., et al. (2025). *Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models*. arXiv:2510.04618.
2. DeepSeek-AI. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*.
3. Mabena, T. (2026). *Automated Offensive Cyber-Operations: The TRM-Ace Model and Agentic Context Engineering*. Internal Technical Report, NWU.
4. Mabena, T. (2026). *Colab CTF Agent Implementation Guide*.
5. Mabena, T. (2026). *Architecting the Cognitive Security Operations Center: A Comprehensive Data Acquisition and Engineering Framework*.
6. *TRM-ACE-CyberSecurityModel_results.pdf (Experimental Logs)*.

For a visual explanation of the Agentic Context Engineering framework utilized in this paper, see this relevant resource:

Agentic Context Engineering: Self-Improving AI Playbooks

This video provides a detailed breakdown of the Generator-Reflector-Curator loop described in the Theoretical Framework section.