



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

An AI-Driven Penetration Testing Framework for Autonomous Offensive Security

LAUREA MAGISTRALE IN COMPUTER SCIENCE & ENGINEERING - INGEGNERIA INFORMATICA

Author: LORENZO LADINA

Advisor: PROF. STEFANO ZANERO

Academic year: 2025-2026

1. Introduction

The landscape of modern offensive security is undergoing a rapid transformation. As enterprise IT infrastructures grow increasingly complex, dynamic, and distributed, traditional penetration testing methodologies [7]—relying heavily on manual enumeration and static automated scanners—struggle to scale. While Large Language Models (LLMs) have demonstrated unprecedented text comprehension and reasoning capabilities in cybersecurity [5], deploying them as fully autonomous agents introduces severe operational and architectural bottlenecks:

- **Context Window Saturation (Vertical Bloat):** The ingestion of massive, unstructured raw security logs and verbose telemetry quickly exhausts the model's context window, severely degrading its logical reasoning.
- **Cognitive Drag (Horizontal Bloat):** Operating in continuous execution loops without state optimization forces the linear accumulation of command history, taxing the model's attention mechanism and drastically increasing latency and API costs.
- **Instruction-Data Confusion:** The lack of deterministic boundaries between trusted system instructions and untrusted envi-

ronmental data leaves autonomous agents highly vulnerable to Indirect Prompt Injection (IPI) attacks [4].

This thesis introduces a specialized middleware architecture designed to bridge the gap between high-level AI reasoning and low-level security execution, mitigating context saturation, limiting cognitive drag, and reducing environmental vulnerabilities through targeted orchestration

2. Proposed System Architecture

The framework implements a dynamic, closed-loop architecture that decouples strategic reasoning from practical execution. The framework mitigates context-handling challenges via two primary mediation layers:

2.1. Semantic Compression Layer

This component programmatically distills verbose tool outputs into highly dense, token-efficient security primitives before they reach the LLM. For instance, the Active Directory [2] module processes massive BloodHound JSON [6] arrays, stripping non-functional metadata to reconstruct a concise text-based graph of exploitable Access Control List (ACL) relationships (e.g., *GenericAll*, *WriteDacl*) [1]. Simi-

larly, the Web Semantic Wrapper strips away the visual presentation layer of raw HTML DOMs, extracting exclusively interactable forms, input fields, and action links.

2.2. Strategic Memory Management

To eliminate cognitive drag during long-duration engagements, the orchestrator maintains a dynamic *Recon Summary* instead of a raw, linear historical log. Backed by a local SQLite persistence layer, it enforces a strict routine of *Memory Hygiene* by filtering out redundant execution telemetry and explicitly highlighting immediate errors, forcing the model to reflect on execution failures and pivot strategies rather than entering infinite repetition loops.

3. Experimental Evaluation: Ablation Studies

The framework was empirically validated through systematic ablation studies across a virtualized Active Directory corporate network and an OWASP Top 10 vulnerable web application lab using Gemini 2.5 Flash and 3 Flash preview models as reasoning engine[3].

3.1. Semantic Compression Efficiency

In the Active Directory environment, raw log ingestion caused a prompt explosion peaking at 38,339 tokens, inducing attention collapse; the model bypassed deep graph relationships entirely, falling back on a basic Kerberoasting vector. Conversely, the semantically compressed framework stabilized the context footprint under 4,500 tokens—achieving an 85% reduction—allowing the agent to easily identify hidden ACL misconfigurations and execute a stealthier privilege escalation path.

3.2. Cognitive Drag Mitigation

The absence of the *Recon Summary* layer forced the model to expend up to 10,773 "Thinking Tokens" at its peak merely to maintain state awareness. Activating the Strategic Memory layer achieved an 81% reduction in cognitive overhead (down to 1,961 thinking tokens), demonstrating a crucial FinOps optimization for large-scale enterprise deployments.

3.3. Cross-Model Analysis and Domain Asymmetry

A suite of 200 automated benchmarking runs exposed a clear domain asymmetry. In highly structured network environments like Active Directory, lightweight models such as Gemini 2.5 Flash achieved a 100% success rate at a fraction of the cost. Conversely, unstructured web application exploitation required superior semantic depth; Gemini 2.5 Flash's success rate dropped to 52%, whereas Gemini 3 Flash maintained perfect effectiveness (100%), justifying its higher API cost.

4. Robustness and Security Analysis

Autonomous agents are inherently exposed to hostile environmental data. In baseline configurations (such as basic structural prompt), the agent exhibited a 100% vulnerability rate to Mission Hijacking via Indirect Prompt Injections (IPI) embedded within target payloads. The orchestrator counters this threat through a strict Defense-in-Depth strategy. By integrating the syntactic segregation of system instructions from untrusted data, automatic output escaping, semantic pruning to minimize the attack surface, and deterministic post-generation command validation against supported schemas, the framework successfully neutralizes malicious payloads, reducing the IPI success rate to a residual 10%.

5. Conclusions

This research demonstrates that the operational resilience, economic scalability, and security of autonomous offensive agents do not rely solely on the underlying foundation model's context capacity, but require a rigorous engineering of the orchestration middleware. The empirical validation confirms that addressing systemic bottlenecks—such as context saturation and cognitive drag—through Semantic Compression and Strategic Memory is mandatory to transition LLMs from passive assistants to active operational tools. Furthermore, by successfully isolating logical reasoning from data noise and structuralizing the execution loop, the system proactively mitigates severe environmental vulnerabilities like Indirect Prompt Injection. Ul-

timately, the proposed framework establishes a robust, maintainable, and tool-agnostic foundation for enterprise-grade automated penetration testing, paving the way for scalable, AI-driven Red Teaming operations.

References

- [1] Microsoft Corporation. How access control works in active directory domain services. Technical report, Microsoft Learn, 2020.
- [2] Microsoft Corporation. Active directory domain services overview. Technical report, Microsoft Learn, 2025.
- [3] Google Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2025.
- [4] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you’ve signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173*, 2023.
- [5] Andreas Happe and Jurgen Cito. Getting pwn’d by ai: Penetration testing with large language models. *arXiv preprint arXiv:2308.00121*, 2023.
- [6] SpecterOps. Bloodhound. <https://github.com/specterops/bloodhound>, 2026.
- [7] Blake Strom, Andy Applebaum, Doug Miller, Kathryn Nickels, Adam Pennington, and Cody Thomas. Mitre att&ck: Design and philosophy. Technical report, The MITRE Corporation, 2020.