GUC
German University in Cairo
الجامعة الألمانية بالقاهرة

A Self-Adaptive Agentic Moving Target Defense Architecture for Real-Time Cyber Threat Response

Master's Defense 6/7/2025

By Karim Ahmed

Under the Supervision of

Assoc. Prof. Dr. Mervat Abuelkheir
MET

Assoc. Prof. Dr. Maggie Mashaly
IET

Outline

- Introduction & Literature Review
- Problem Statement & Research Objectives
- Proposed Architecture
- Evaluation
- Conclusion

A Self-Adaptive Agentic Moving Target Defense Architecture for Real-Time Cyber Threat Response

Karim Ahmed
CSE Dept., MET Faculty
German University in Cairo
Cairo, Egypt
karim.abdel-aziz@guc.edu.eg

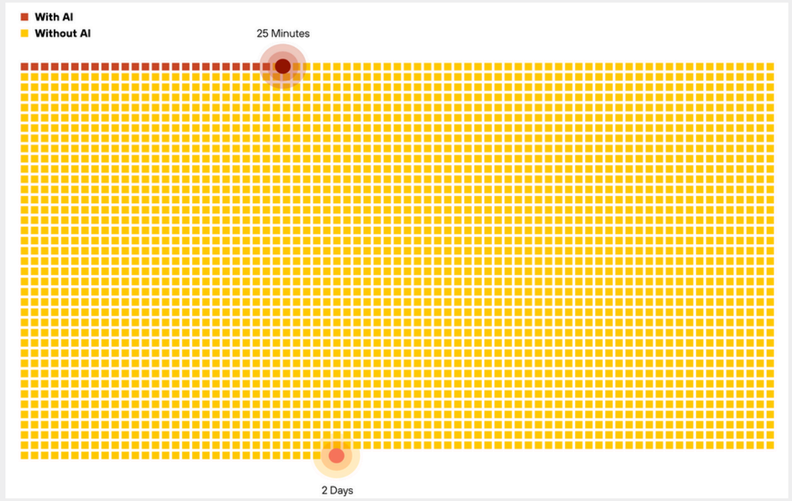
Maggie Mashaly
Networks Dept., IET Faculty
German University in Cairo
Cairo, Egypt
maggie.ezzat@guc.edu.eg

Mervat Abu-Elkheir
CSE Dept., MET Faculty
German University in Cairo
Cairo, Egypt
mervat.abuelkheir@guc.edu.eg

Submitted for review in proceedings of
IEEE Conference on Communications and Network Security 2025
Avignon, France

Introduction & Literature Review

Current Cybersecurity Landscape



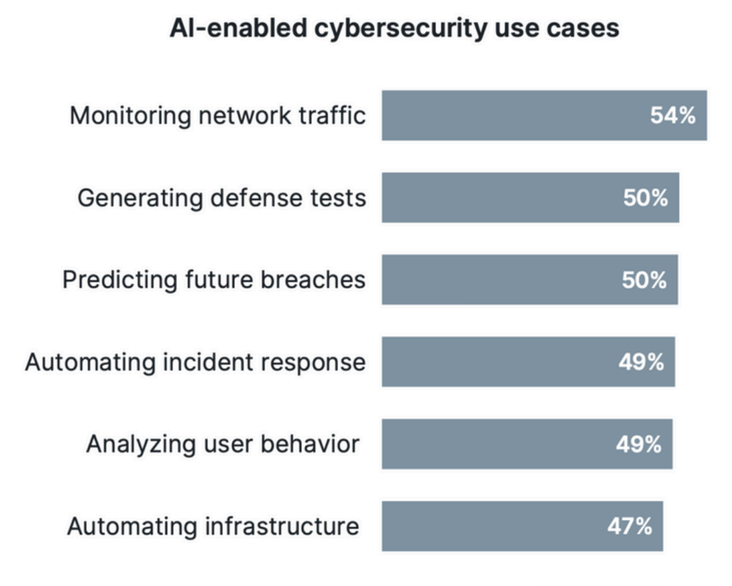
Illustrative timeline comparing manual and AI-assisted intrusion speed

Progressing Attack Speed

- Global attacks against organisations increased by 44% increase from 2023
  - with an average number of 1673 attacks per week
- 1 in 5 of the investigated cases by Palo Alto's Unit42, data exfiltration occurred in less than one hour since the compromise happened
- 54% of organisations surveyed by CompTIA are using or plan to use AI tools for monitoring network traffic
  - And 49% for automating incident response
- Time between detection and containment for the surveyed organisations, according to SANS 2025 IR Report:
  - 34.3% → between 1 to 5 hours
  - 22.7% → between 6 to 24 hours, while only
  - 19% → under an hour,
  - 5.4% → unknown
  - rest → more than 24 hours

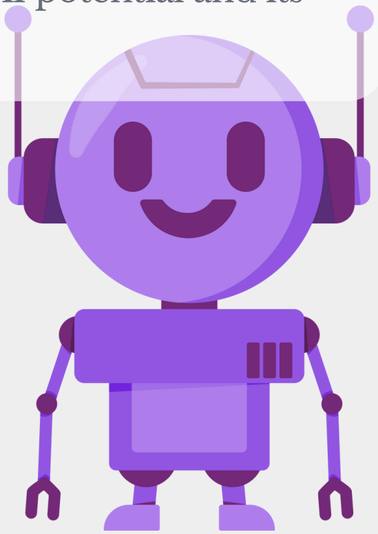
Introduction & Literature Review

Current Cybersecurity Landscape



AI-Assisted Offence

- **OpenAI identified accounts that used their models to generate attack scripts and research vulnerabilities for exploitation in 2024**
- 41% of surveyed firms by CompTIA, are still in the education or pilot phase of AI adoption, with 36% conducting only low-priority implementations.
  - reflecting the substantial gap between AI potential and its practical realization



Introduction & Literature Review

Moving Target Defense (MTD)

First conceptualized around 2009 as a paradigm shift from static, reactive cyber defenses to dynamic, proactive ones

The goal is to dynamically change system properties faster than attackers can exploit them, thereby thwarting reconnaissance and exploitation attempts in real-time.

Introduction & Literature Review

MTD - Principles and Techniques

5 layers model

- Dynamic Data
- Dynamic Software
- Dynamic Runtime Environment
  - Address Space Randomisation
  - Instruction Set Randomisation
- Dynamic Platforms
- Dynamic Networks

01

What to Move

How to Move

When to Move

Introduction & Literature Review

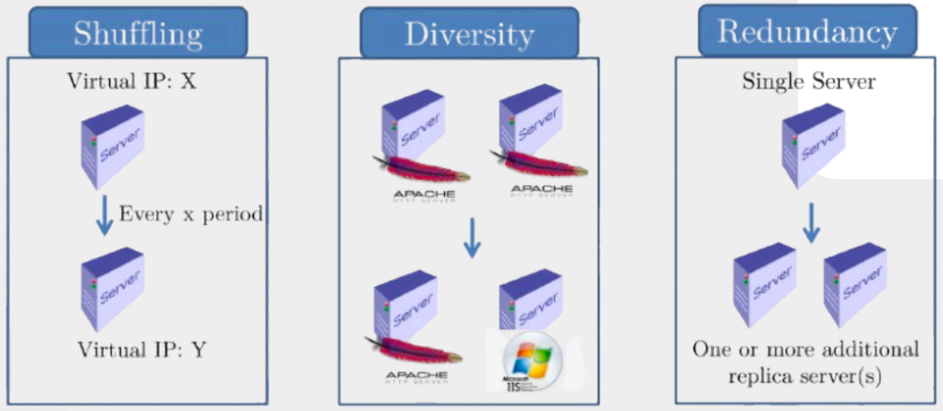
MTD - Principles and Techniques

SDR Taxonomy

- **Shuffling**
- **Diversity**
- **Redundancy**

What to Move | **How to Move** | When to Move

02



Introduction & Literature Review

MTD - Principles and Techniques

Triggering Strategies

- Fixed-Time
- Ad-hoc (Event-based)
- Hybrid

What to Move

How to Move

When to Move

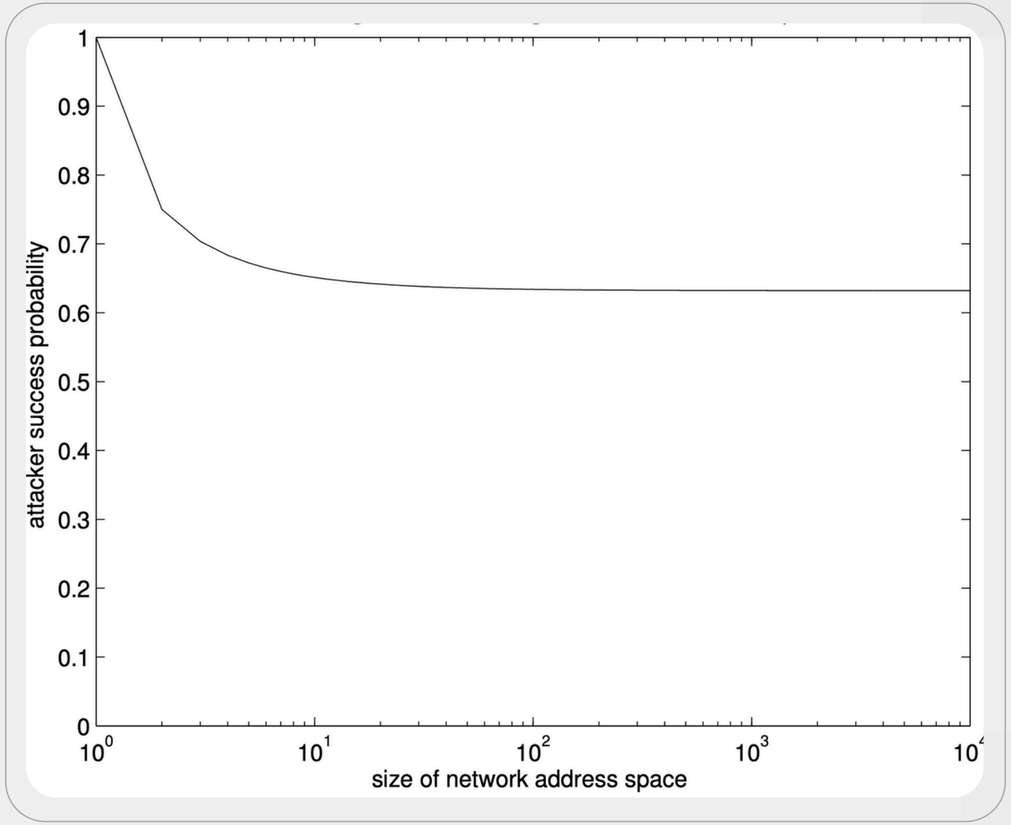
03

Introduction & Literature Review

MTD - MultiLayering

A single action is not enough

The attacker’s success rate plateaus at 64% even in significantly large address space. This success rate is defined as finding a single vulnerable machine in the targeted networks address space.



Attacker Success using Perfect Shuffling as Network Address Space Increases

Introduction & Literature Review

Intelligent MTD Techniques

Towards Affordable MTD

Lightweight models that are compressed for singular focus. Used mainly for edge and IoT devices

Towards Self-Adaptive MTD

Current Self-adaptive techniques use RNN to forecast emerging attack patterns. However, they face constant concept-drift and face high inference time.

Towards Optimized MTD

Training models on substantial amount of data and require exhaustive compute time to balance when and how to trigger MTD actions with minimal cost

Introduction & Literature Review
Large Language Models

Prompt Engineering

Training is great, but what if we avoid it → ICL

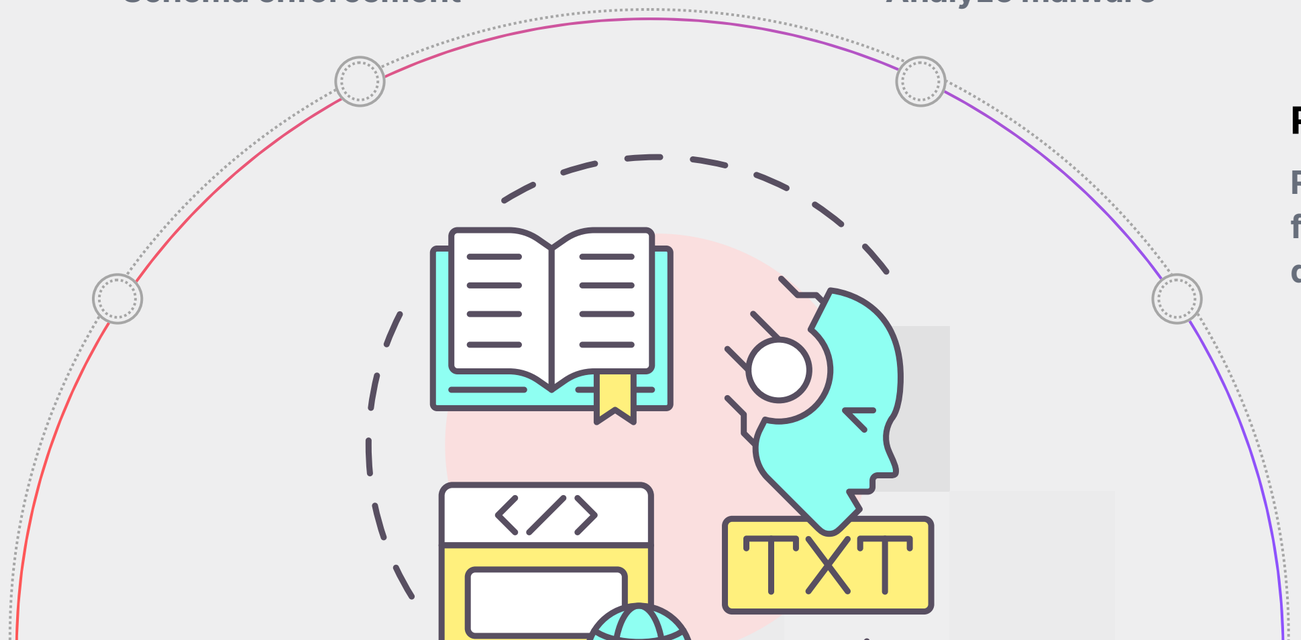
- Few-shots
- System message
- CoT
- Schema enforcement

LLM in Cybersecurity Now

- Threat detection
- Automate vulnerability management
- Assist incident response analysis
- Devise cyber deception strategies
- Analyze malware

Architecture

Transformer
Modern LLMs - NLP tasks
Ease of deployment



RAG

Real-time knowledge look-up from databases or documents

Problem Statement & Research Objectives

Literature Gap

- MTD frameworks lack cognitive orchestration
- AI-driven MTD is not leveraging LLMs
- LLMs in security remain underused for active defense

Research Objectives

- Design an MTD self-adaptive cyber defense architecture using LLMs
- Implement said Design using Prompt engineering (to avoid costly fine-tuning) and RAG for threat intelligence gathering
- Evaluate the implementation on: Latency, Success, plan accuracy, and agents' consistency.

Problem Statement & Research Objectives

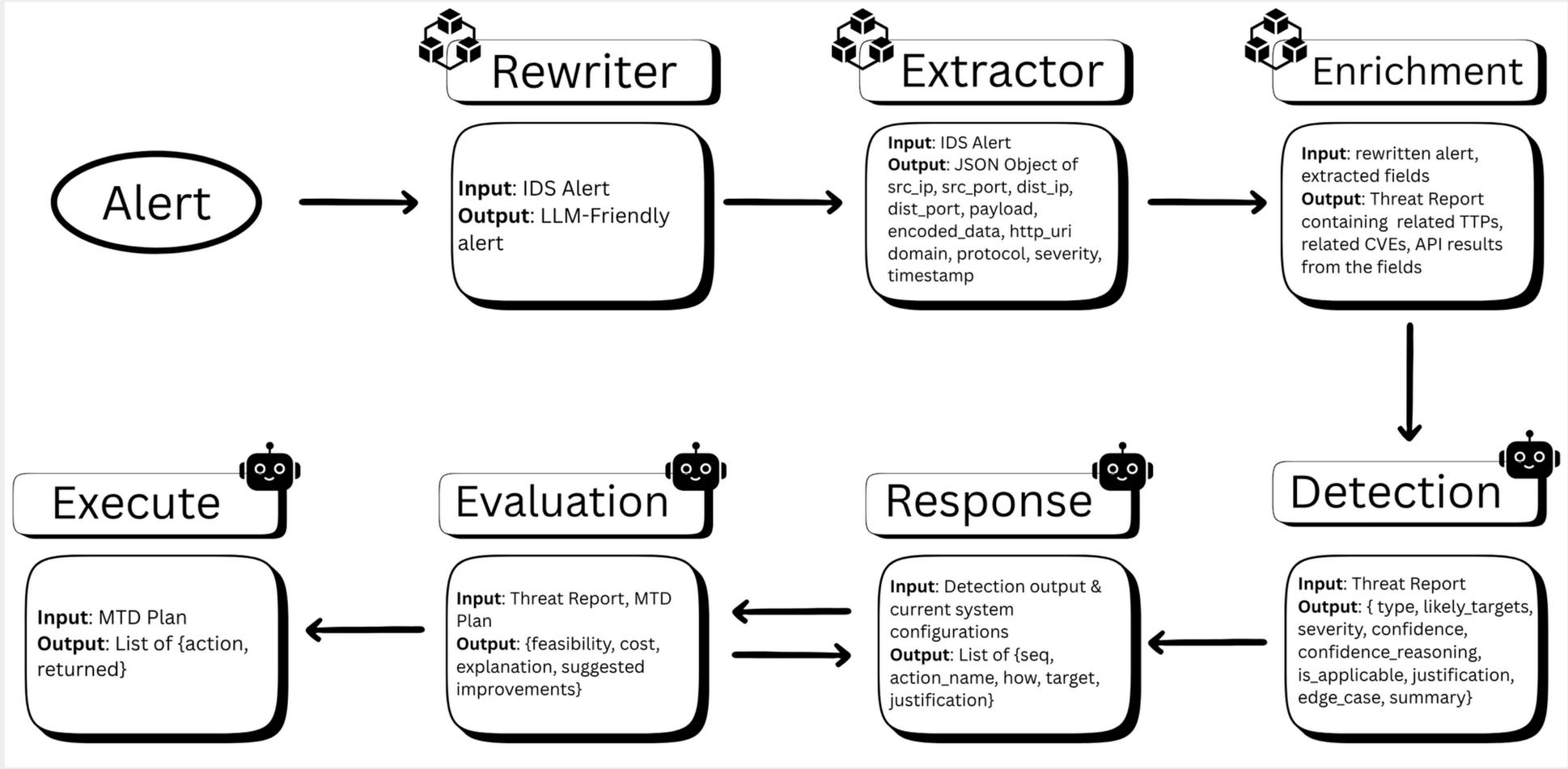
Research Motivation

- Quickening Attack speed vs. slow IR time
- Utilizing recent LLMs reasoning abilities
- Providing adaptable single-pane cognitive layer for automated responses

Research Questions

- Can an LLM+RAG–augmented agent orchestrate multi-layer MTD actions in the available narrow window?
- How accurate and sound are the defense plans generated by the LLM agent, and can the agent execute these plans consistently and deterministically across repeated runs?

Proposed Architecture



Proposed Architecture
Pipeline details

1

Alert Reception

The pipeline focuses on post-detection rather than detection and response.
An IDS sends an alert to the pipeline to start the process.

Alert

2

Initial preprocessing

Changing the alert language
Extraction of key fields

Rewriter

Input: IDS Alert
Output: LLM-Friendly alert

Extractor

Input: IDS Alert
Output: JSON Object of src_ip, src_port, dist_ip, dist_port, payload, encoded_data, http_uri domain, protocol, severity, timestamp

3

Enrichment process

Using RAG, we query the alert against TTP and CVE datasets to find matching entries for context enrichment
Using the extracted fields, we call external APIs for threat intelligence

Enrichment

Input: rewritten alert, extracted fields
Output: Threat Report containing related TTPs, related CVEs, API results from the fields

Proposed Architecture
Pipeline details

4

Detection Agent

Responsible for threat classification and MTD applicability assessment

Detection

Input: Threat Report
Output: { type, likely_targets, severity, confidence, confidence_reasoning, is_applicable, justification, edge_case, summary}

5

Response Agent

Responsible for MTD action planning

Response

Input: Detection output & current system configurations
Output: List of {seq, action_name, how, target, justification}

6

Evaluation Agent

Responsible for plan validation and improvement suggestions

Evaluation

Input: Threat Report, MTD Plan
Output: {feasibility, cost, explanation, suggested improvements}

Proposed Architecture
Pipeline details

7

Execute Agent

carries out the prescribed MTD techniques on the target system in real-time

Input: MTD Plan
Output: List of {action, returned}

IP Address Randomization

changing the network address of a host, thereby invalidating the attacker's knowledge of the target's location.

Port Randomization

remapping services to non-standard, random high ports, so that we can disrupt scans and exploits that assume default ports.

DNS Shuffling (Service Address Rotation)

randomize or rotate the association between domain names and IP addresses to confuse malware or scripts that rely on static DNS records

Traffic Obfuscation (Decoy Traffic Generation)

Traffic obfuscation aims to mask legitimate network traffic patterns or confuse adversaries by generating noise traffic

Proposed Architecture - Features

Modular and Independant

Each Agent/Module is standalone and can be used independently without calling the full pipeline.

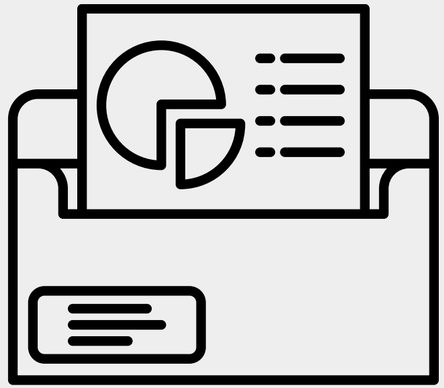
Enrichment

It doesn’t rely solely on the information in the alert, but calls for threat intel for better analysis

self-adaptability

This comes from a back and forth conversation that hapens between Response and Evaluation to reach a good plan

Evaluation



We ran 2 Attack Scenarios

- SSH-Brute Force Attack
- Reconnaissance Attempts

Utilizing 3 LLMs

- Llama3.2:7b
- qwen2.5:7b
- deepseek-r1:8b

Over two settings

- Cold-start
- Continuously running

We gathered information about the following:

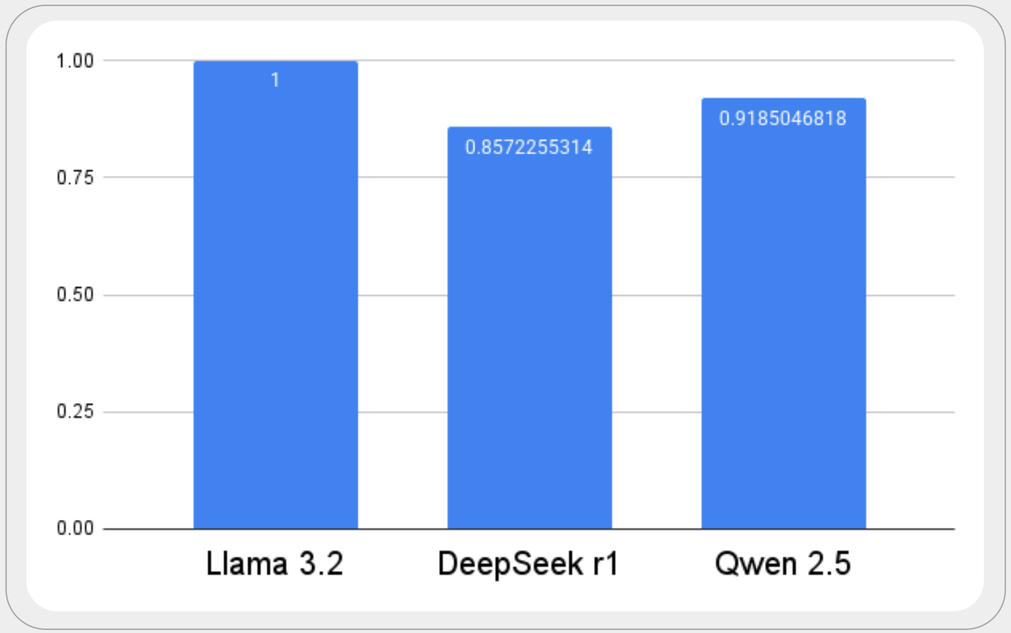
- Consistency:
  - cosine similarity on different outputs for the same input alert
  - Importance lies in replicating successful behaviour
- Run Time
  - Per Agent/Module
  - Overall

- RAG Similarity and Accuracy
  - Similarity threshold for retrieval and accuracy of returned documents
- Plan Soundness and Successful Execution
  - Correct choice of relevant actions
  - Success rate of execution

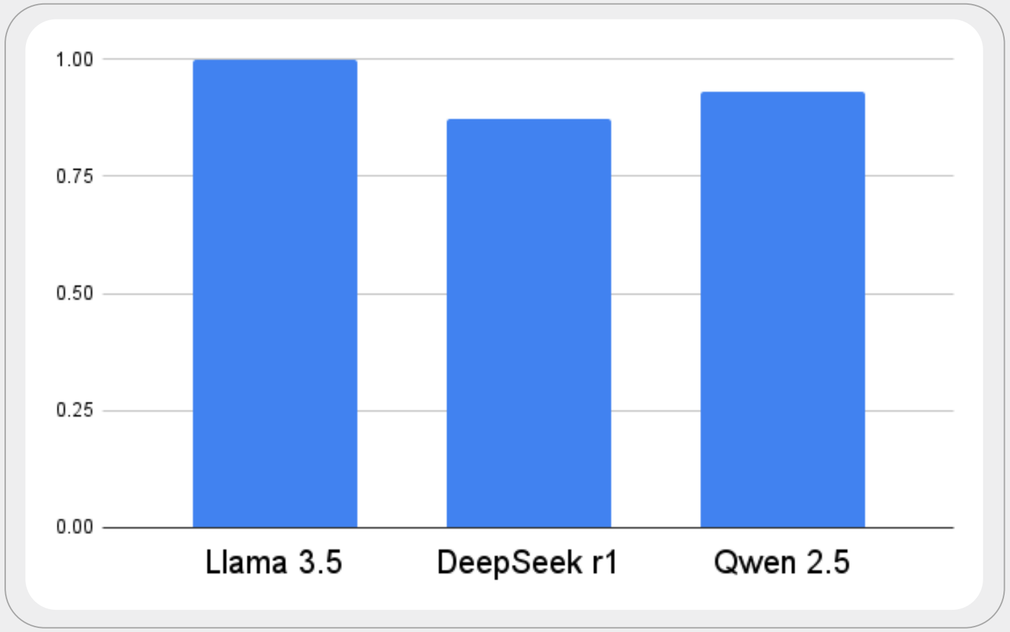
Evaluation

Consistency

Preprocessing modules (Rewrite and Enrichment) and Detection Agent, recorded 100% consistency across the three LLMs used



Response Agent Consistency across the three LLMs.

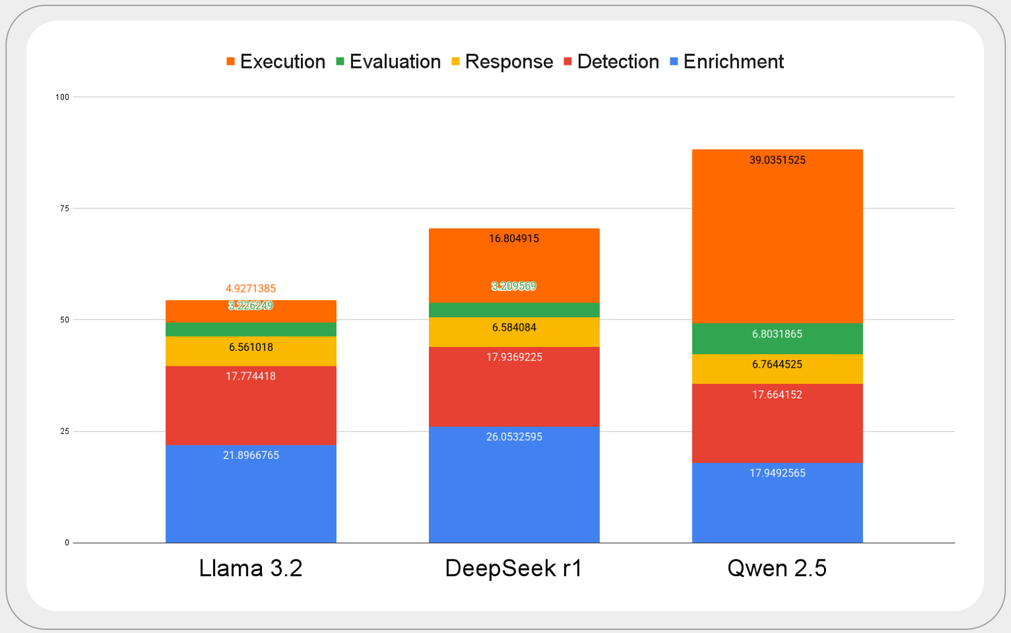


Evaluation Agent Consistency across the three LLMs

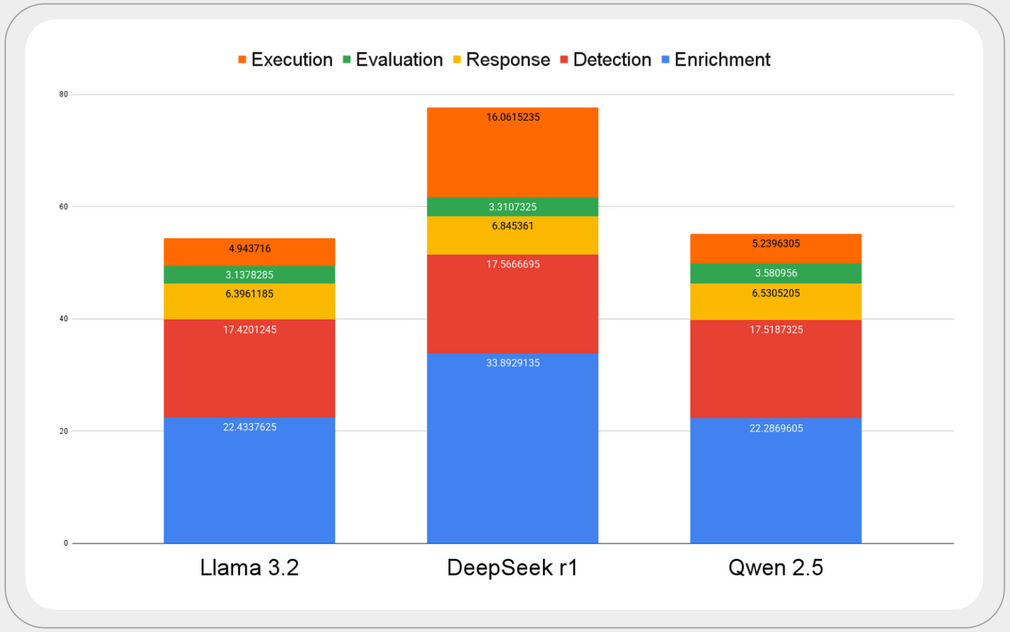
Evaluation

Runtime

Longest is qwen on multiple runs → 90 seconds



Runtime with multiple prior invocations

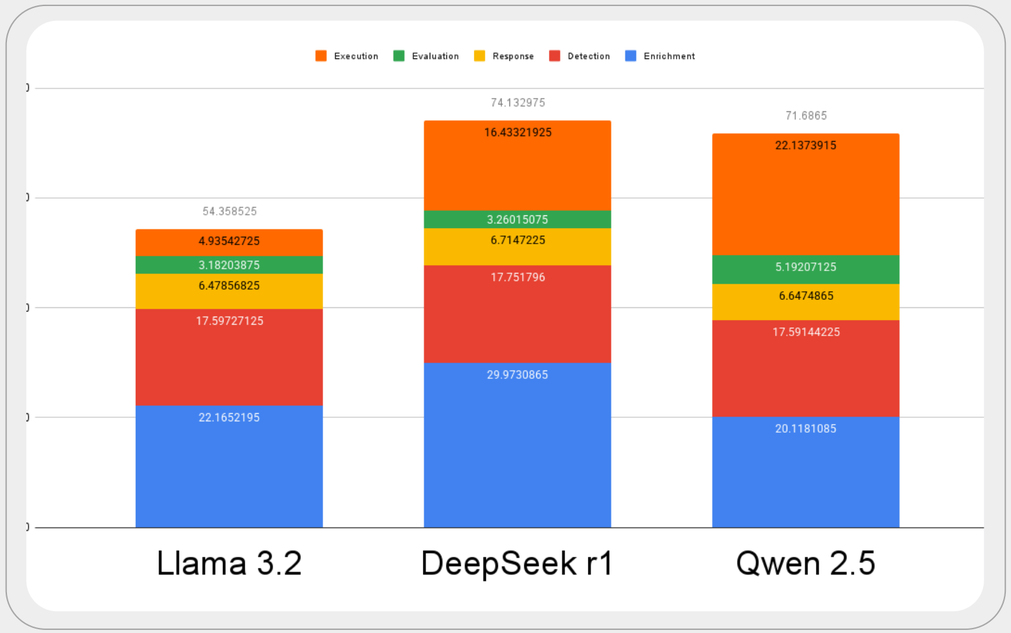


Runtime with no prior invocations

Evaluation

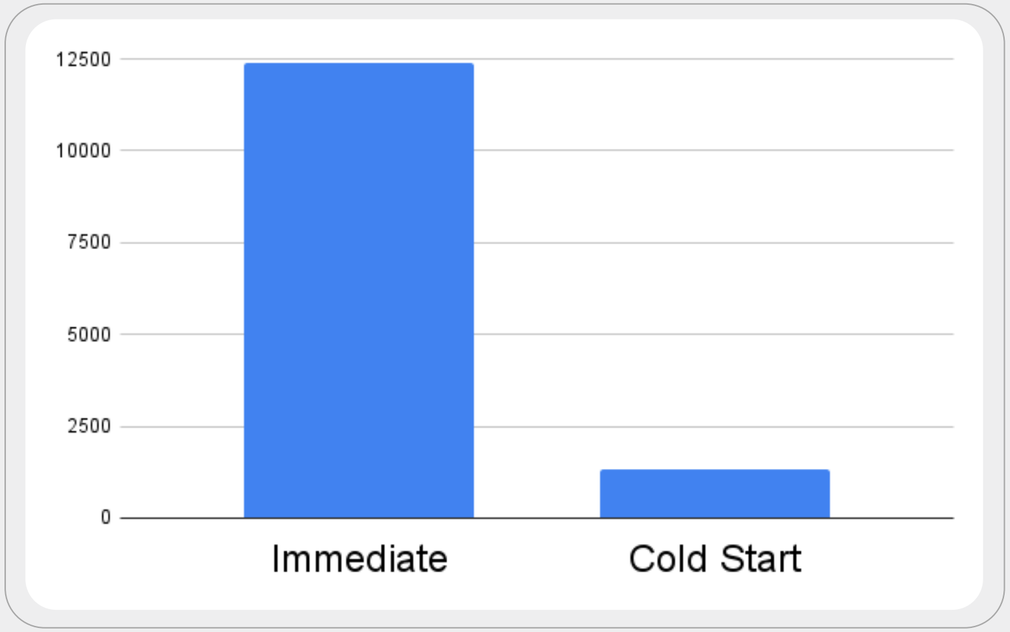
Runtime

Longest on average is 74 seconds
73'792 CVEs (2022 → 2024)
565 TTPs (April 2025)



Average total runtime

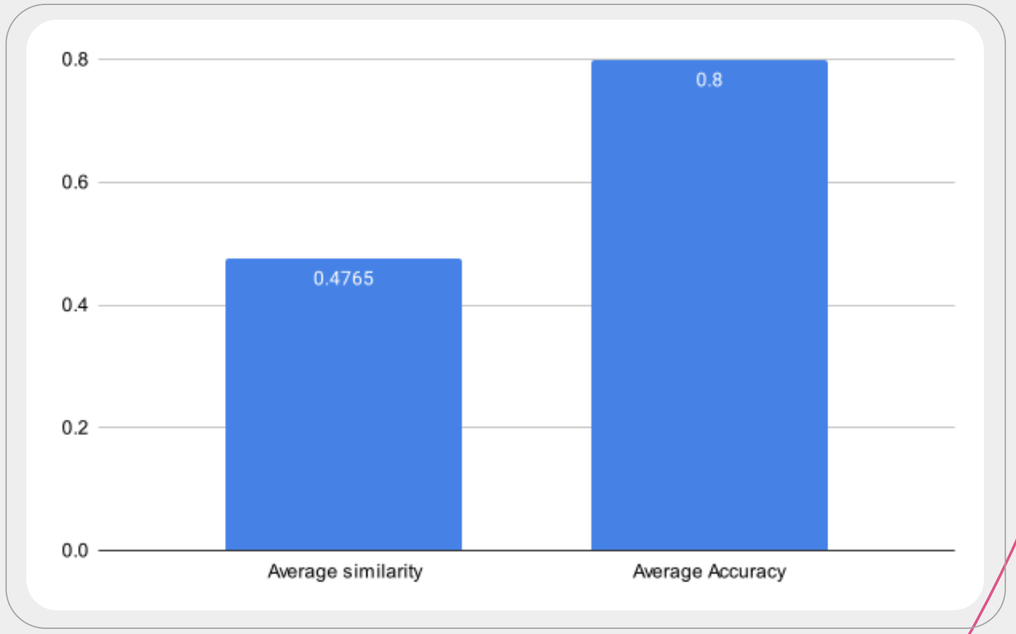
Immediate runs → 3.5 Hours
Cold start → 23 minutes



Avg vector database build time

Evaluation
Similarity, Accuracy & Execution

Similarity → threshold
Accuracy → retrived document relevance to the observed attack



RAG Analysis

Comprehensive Plan Generation rate

- Llama: 50%
- Deepseek: 100%
- Qwen: 100%

Plan Soundness - Qualitative Analysis

- DeepSeek and Qwen more consistently generated plans covering all required actions
- Llama specifically struggled with generating plans that would mitigate detected reconnaissance attacks.

Execution Success Rate

- Llama and Deepseek 75%
- Qwen 87.5%

Conclusion

Contributions

- Agentic MTD Pipeline
- Real-Time MTD Actions
- RAG Scalability Insights
- Bridging the gap between LLM potential and MTD adaptability

Findings

- LLM-driven response is fast enough for modern threat windows
- Model choice matters
- Database maintenance can be the hidden bottleneck

Limitations and future work

- Broader Threat Spectrum
- Distributed Deployment
- Multithreaded System/ buffer or queue system
- Experimenting with different architectures
- Fine-tuning

Q&A

Thank You

Thank you for your time.

System configurations

- Hyperparameters
  - temperature: 0
  - num_ctx: 8192
  - top_p: 0
  - max_tokens: 8192
  - repeat_penalty: 1.1
- Hardware
  - GPU: NVIDIA GeForce RTX 2060 6GB VRAM
  - CPU: Intel i7-9750H

Environment

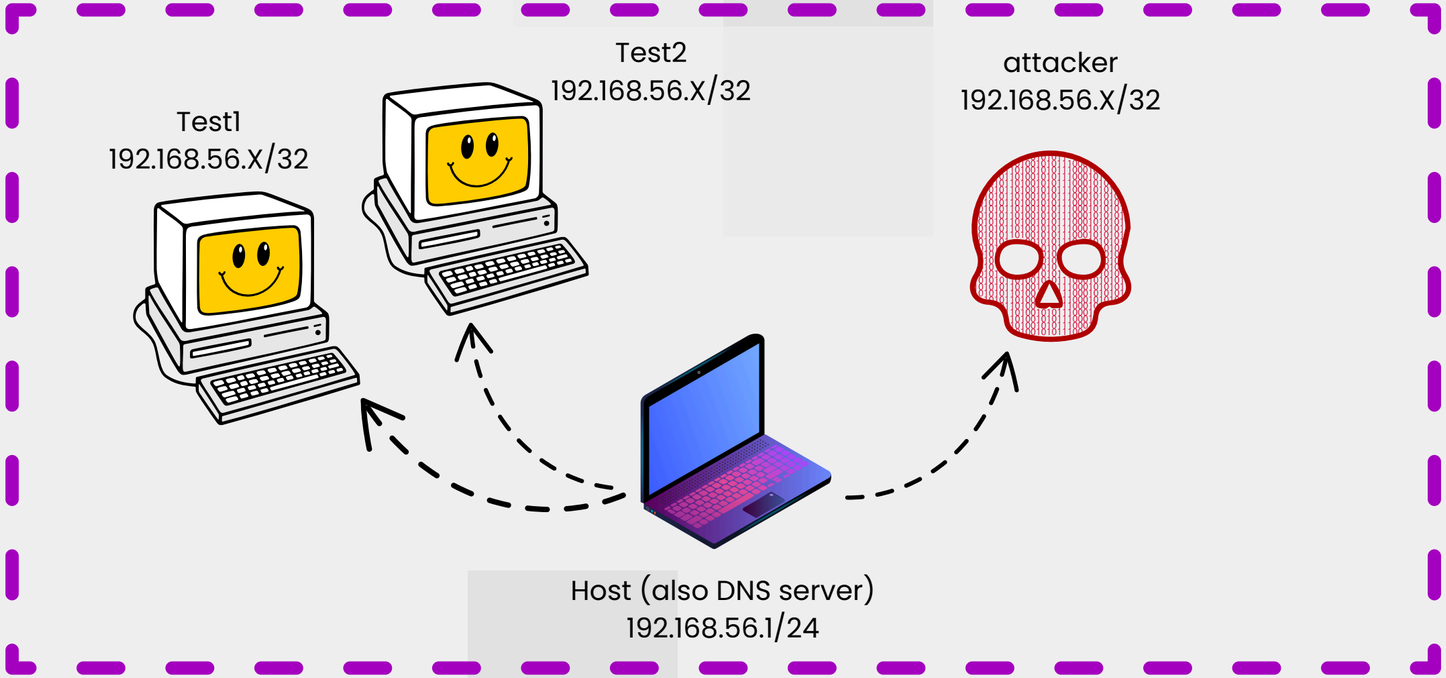


TABLE I. Agent Prompting Techniques.

Agent	*System Message Content*	*Other Techniques*
Rewrite	• What it is • What to do and not to do	• few-shots
Extractor	• What it is • What to do and not to do	• schema-based prompting
Detection	• What it is • What it does • Key input terminology • What is MTD • Objective	• schema-based prompting • few-shots • CoT
Response	• What it is • What it does • What is MTD • Objective • Available actions • What not to do	• schema-based prompting • few-shots • CoT
Evaluation	• What it is • What it does • What is MTD • Available actions • What not to do	• schema-based prompting • few-shots • CoT
Execution	• What it is • What it does • What is MTD • Available actions • What not to do	• schema-based prompting • CoT

Listing III.1 Example Snort rule that generates an alert when an external host sends a TFN-style ICMP Echo request to an internal host.

```
alert tcp any any -> any any alert icmp
    $EXTERNAL_NET any -> $HOME_NET any (msg:"
    PROTOCOL-ICMP TFN Probe"; icmp_id:678; itype
    :8; content:"1234"; fast_pattern:only;
    metadata:ruleset community; reference:cve
    ,2000-0138; classtype:attempted-dos; sid:221;
    rev:12;)
```

name "Disable Crypto Hardware"
id "T1600.002"
url "Consistency calculations"
platforms "Network"
kill chain phases "Defense Evasion"
description "Adversaries disable a network device's dedicated hardware encryption, which may enable them to leverage weaknesses in softwa... the strength of the cipher in software (e.g., [Reduce Key Space](T1600.001)). (Citation: Cisco Blog Legacy Device Attacks)"
data sources "File: File Modification"
detection "There is no documented method for defenders to directly identify behaviors that disable cryptographic hardware. Detection ef...m Image](T1601) and [Network Device CLI](T1059.008). Some detection methods require vendor support to aid in investigation."

TTP Example

cveId "CVE-2024-8158"
publishedYear 2024
descriptions_value "A bug in the 9p authentication implementation within lib9p allows an attacker with an existing valid user within the configu...m Plan 9 and is present in all versions of 9front and is remedied fully in commit 9645ae07eb66a59015e3e118d0024790c37400da."
product_affected_1 "9front"
vendor_affected_1 "9front"
references_tags "['patch']"
url "https://git.9front.org/plan9front/plan9front/07aa9bfeef55ca987d411115adcfbbd4390ecf34/commit.html"
description_problemType "CWE-639 Authorization Bypass Through User-Controlled Key"
problemTypes "CWE"

CVE Example

Consistency calculations

$$r_i^{(j,1)}, r_i^{(j,2)}, \ldots, r_i^{(j,R)}$$

$$\mathbf{v}_i^{(j,m)} = \mathrm{Enc}\left(r_i^{(j,m)}\right)$$

$\mathrm{sim}_{\cos}(\mathbf{v}_i^{(j,m)}, \mathbf{v}_i^{(j,n)})$ for $1 \le m < n \le R$, and define

$$C_j(x_i) = \frac{2}{R(R-1)} \sum_{1 \le m < n \le R} \mathrm{sim}_{\cos}\left(\mathbf{v}_i^{(j,m)}, \mathbf{v}_i^{(j,n)}\right).$$

The overall consistency for model j is

$$\overline{C}_j = \frac{1}{N} \sum_{i=1}^{N} C_j(x_i).$$