AURA: Adaptive Unified Response Agent

Abhinav Anil K – 02
Abin Raj M K – 05
Amandeep S – 07
Anaswara K S – 12
Aswanth S – 21

Supervised by: Soubhagya V N

October 15, 2025

Group 01

Introduction

Background

- **Domain:** The project operates in the domain of **Proactive Cybersecurity**, a necessary evolution from traditional, reactive security models.
- **Project About:** AURA is an AI-driven framework designed to autonomously defend modern IT and IoT infrastructures by integrating threat intelligence, deception, and automated response.
- **Real-world Context:** In an era of sophisticated threats like Advanced Persistent Threats (APTs) and zero-day exploits, static defenses are failing. AURA addresses this vulnerability.

Relevance of the Topic

Why This Topic?

- **Current Relevance:** With the explosion of IoT devices and cloud adoption, the digital attack surface has grown exponentially, making intelligent, automated defense a critical need for all industries.
- **Real-world Problem:** AURA directly addresses the critical issue of "alert fatigue" in Security Operations Centers (SOCs) and the inability of legacy systems to stop novel attacks.
- **Positive Impact:** The project has the potential to significantly enhance the cyber resilience of critical sectors, including finance, healthcare, and smart city infrastructure.

Novelty and Innovation

What is New?

- **Unified Closed-Loop System:** AURA's primary innovation is its synergistic integration of OSINT, deception, and response into a single, autonomous feedback loop (Intelligence -¿ Deception -¿ Response).
- **Intelligence-Driven Deception:** Unlike static honeypots, AURA uses live threat intelligence to dynamically generate and configure decoys, making them highly credible and effective against targeted attacks.
- **Predictive Posture:** The framework moves beyond detection to prediction, using an AI core to forecast emerging threats based on real-world data.

Problem Statement

Problem Statement

To develop an AI-driven, adaptive cyber defense framework that seamlessly integrates intelligent intrusion detection, OSINT-based threat intelligence, and autonomous response mechanisms—leveraging machine learning and reinforcement learning to deliver real-time, context-aware, and unified protection against evolving cyber threats.

Project Objectives

Key Goals

- To develop an AI-based intrusion detection system for real-time threat detection.
- To integrate OSINT-driven threat intelligence for contextual awareness.
- To implement adaptive and autonomous response using reinforcement learning.
- To unify detection, intelligence, and response within a single framework.
- To provide real-time, explainable, and scalable cyber defense capabilities.

Proposed System Architecture

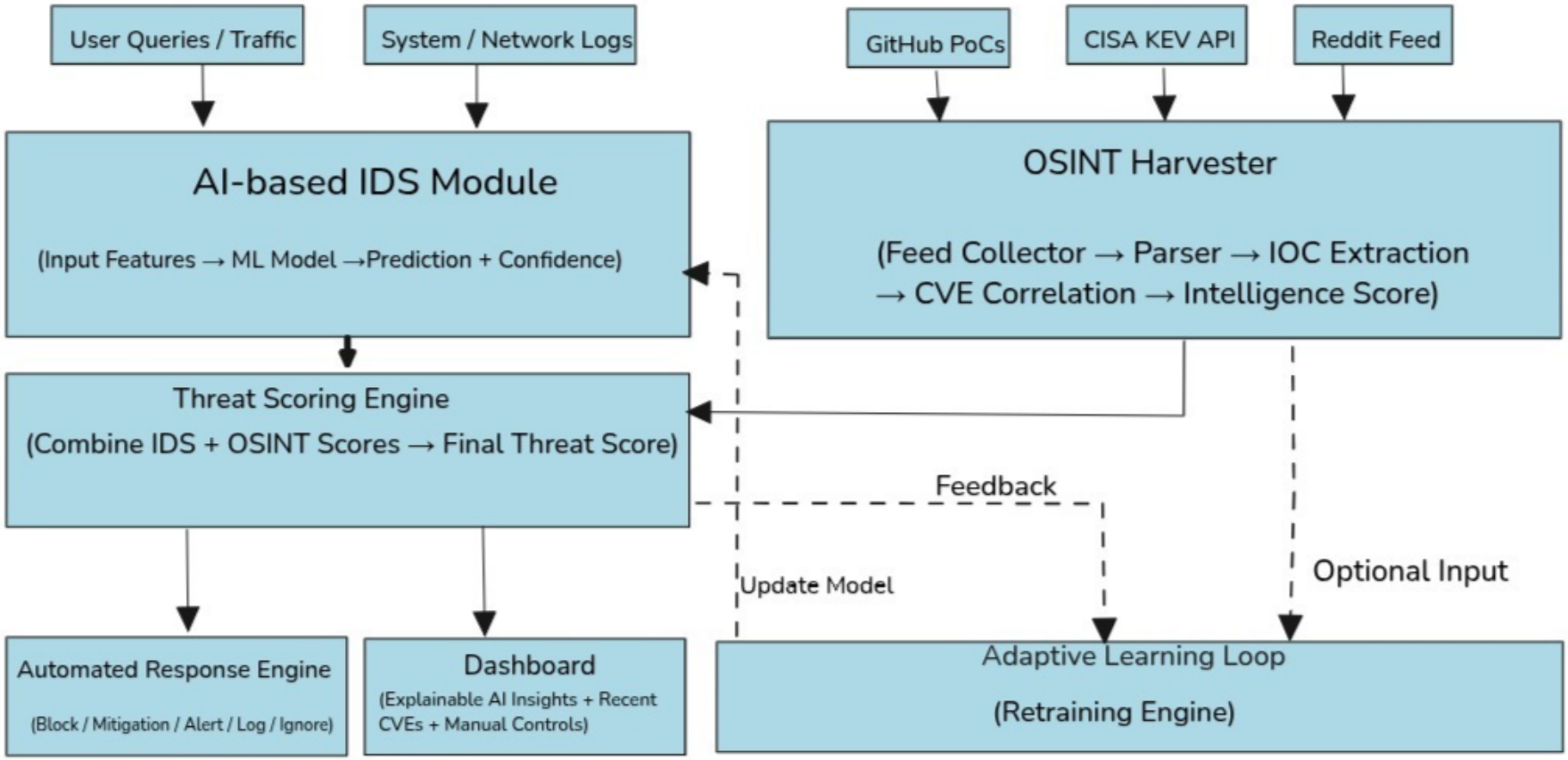

Figure: The Four Interconnected Layers of AURA.

Datasets

1. **Unified Multimodal Network Intrusion Detection System Dataset (UM-NIDS)**
   - DatasetSource:
     - Available IEE dataport: https://ieee-dataport.org/documents/unified-multimodal-network-intrusion-detection-systems-datasetfiles
   - Size:
     - Contains processed data from four benchmark datasets—CIC-IDS 2017, CIC-IoT 2023, UNSW-NB15, and CIC-DDoS 2019—with millions of network flow and payload records.
   - Features:
     - Combines network flow, payload, and contextual attributes.
     - Includes time-window patterns, attack types, and protocol metadata.
     - Provides labeled classes such as DoS, DDoS, Botnet, and normal traffic.
   - Preprocessing steps:
     - Data cleaning and normalization of flow and payload features.
     - Encoding categorical attributes (e.g., protocol, attack type).
     - Feature scaling using Min-Max or Standardization.
     - Balancing classes and applying train-test split (80:20).
   - Used in:
     - AURA IDS module for AI-driven intrusion detection.

Datasets

2 Live OSINT Threat Feed Dataset (Dynamic Data Collection)

- Source:
  - Real-time data from Reddit, GitHub, and CISA KEV API.
- Size:
  - Dynamic and continuously expanding based on live fetches.
- Features:
  - Extracted post text, PoC data, and CVE references.
  - Includes IOC indicators, timestamps, and source links.
  - Used for contextual threat correlation and intel scoring.
- Preprocessing steps:
  - Text cleaning and normalization.
  - NER-based IOC extraction.
  - Filtering via hybrid CNN + BiLSTM model.
  - Deduplication and structured JSON storage.
- Used in:
  - AURA OSINT module for real-time cyber threat intelligence.

Datasets

3 CNN-Based Attack Classification Dataset (Cowrie Honeypot Interaction Logs)

- Dataset Name:
  - `Cowrie Honeypot Dataset` — Medium-interaction SSH honeypot data
- Source:
  - Real-world attacker interactions captured by the Cowrie honeypot. `https://www.kaggle.com/datasets/xmlyna/cowrie-honeypot`
- Size:
  - Approximately 792 MB across all files.
  - Estimated row counts by file:
  - `ttylog.csv` — ∼40,000 rows (terminal session logs)
  - `cowrie.csv` — ∼800,000 rows (event-level logs).
- Features:
  - Shell commands, payloads, session duration, and attacker IP metadata.
  - Interaction sequences used to infer attacker intent and tactics.
  - Labels include attack types such as Reconnaissance, Exploitation, Persistence.

Datasets

- Preprocessing steps:
  - Tokenization and normalization of shell commands.
  - Embedding generation using Word2Vec or TF-IDF.
  - Session-level aggregation and structured JSON formatting.
  - Manual or heuristic-based labeling of attack categories.
- Used in:
  - Honeypot Adaptation Engine for real-time attack classification.
  - Guides dynamic honeypot behavior (e.g., service emulation, privilege escalation).
  - Feeds threat classification into AURA's Threat Scoring Engine.

Methodology

Stage 1 - Multi-Source Data Ingestion

Objective To collect and prepare real-time OSINT and internal network data for AI analysis.

1. **Implement the OSINT Harvester Module:**
   - Develop API Connectors for CISA KEV, GitHub, and Reddit.
   - Create parsers to extract essential information from raw data.
   - Schedule automated execution for near real-time intelligence.
2. **Implement the Real-Time Network Data Processor:**
   - Set up packet capture using tools like Scapy.
   - Develop a feature extractor to process raw packets into network flows and extract key features (duration, protocol, flags, etc.).
   - Normalize and vectorize data into a standardized numerical format for the AI model.

Methodology

Stage 2 - Parallel AI-Driven Threat Analysis

Objective To analyze OSINT and network data with dedicated AI models, generating initial threat predictions and intelligence scores.

1. **Develop & Deploy the NIDS Model:**
   - **Train Model:** Use the multimodal UM-NIDS dataset to train a CNN-BiLSTM model for traffic classification (e.g., Normal, DDoS, SQL Injection), optimizing for high accuracy.
   - **Deploy as API Service:** Package the trained model into a microservice (using Flask/FastAPI). This API will accept vectorized network data and output a threat prediction with a confidence score.

Methodology

Stage 2 - Parallel AI-Driven Threat Analysis

2. **Develop & Deploy the OSINT Model:**
   - **Train Model:** Use a custom-labeled dataset from OSINT data (GitHub, CISA, Reddit) to train a hybrid CNN-BiLSTM for NLP. The model learns to:
     - *Identify Key Terms (CNN):* Scan word embeddings to find significant threat keywords.
     - *Understand Context (BiLSTM):* Process feature sequences to understand grammatical structure and threat relevance.
   - **Deploy as API Service:** Package the model into a microservice. This API will accept parsed text and output an "Intelligence Score" quantifying the threat level.

Methodology

Stage 3 - Contextual Threat Fusion & Scoring

Objective To aggregate the independent outputs from the NIDS and OSINT models, fusing them into a single, context-aware "Final Threat Score."

1. **Develop Threat Scoring Engine:** Build a central microservice to receive predictions from both the NIDS and OSINT models.
2. **Implement Fusion Logic:** Create a weighted algorithm that correlates NIDS alerts with relevant OSINT intelligence, enriching the alert with external context.
3. **Calculate and Output Final Score:** The engine calculates a single, normalized "Final Threat Score" and passes it to the response stage.

Methodology

Stage 4 - Automated Response & Dynamic Deception

Objective To execute an automated, real-time response based on the Final Threat Score using a rule-based engine and a dynamic honeypot.

1. **Develop the Rule-Based Response Engine:**
   - Define a clear set of "if-then" rules based on the Final Threat Score (e.g., IF Score $>$ 90 THEN Block IP).
   - One key rule will be: IF the threat is novel and the score is high, THEN redirect to the honeypot.
2. **Implement the Dynamic Deception Core (Honeypot):**
   - Build a high-interaction honeypot that can dynamically mimic real services.
   - Integrate it with the response engine so that the "Redirect to Honeypot" action diverts attacker traffic to the decoy.
3. **Execute Automated Response:** A central script executes the chosen action by integrating with network control points like firewalls.

Methodology

Stage 5 - System Learning & Model Refinement

Objective To create a continuous feedback loop that uses the results of automated responses and security analyst input to periodically retrain and improve the AI models.

1. **Log All Actions and Outcomes:** Implement comprehensive logging for every alert, action, and outcome. Critically, log all attacker activity within the honeypot.
2. **Develop an Analyst Review Dashboard:** Create a web interface for a human analyst to review critical alerts and label the system's decisions as "True Positive" or "False Positive."
3. **Create a Model Retraining Pipeline:** Build an automated script that periodically collects this new, analyst-verified data to fine-tune and retrain both the NIDS and OSINT AI models.

Methodology

Stage 5 - System Learning & Model Refinement

4. **Refine Response Rules:** Use insights from the analyst review and honeypot logs to manually improve the logic in the Rule-Based Response Engine.

Algorithms/Models Used

Stage / Module	Algorithm / Model Used	Justification
AI-based IDS (NIDS)	Convolutional Neural Network (CNN) + BiLSTM Hybrid	Combines the CNN's ability to extract spatial features from network flows and payloads with the BiLSTM's strength in analyzing temporal sequences of events, ideal for detecting complex, multi-stage attacks.
OSINT Analysis	CNN-BiLSTM for NLP (with Word Embeddings)	The CNN identifies key threat-related keywords from text, while the BiLSTM understands grammatical context to accurately determine if the intelligence is security-relevant.
Threat Scoring	Weighted Scoring Algorithm / Rule-Based Logic	Fuses the NIDS prediction and OSINT intelligence scores into a single, context-aware Final Threat Score by correlating internal alerts with external threat data.
Automated Response	Rule-Based Response Engine	Provides a reliable and deterministic mechanism for executing predefined actions (e.g., Block, Quarantine, Redirect) based on the Final Threat Score, avoiding RL complexity.
Dynamic Deception	Dynamic High-Interaction Honeypot	Safely isolates and analyzes novel or persistent threats to collect intelligence on attacker tactics, techniques, and procedures (TTPs) without endangering production systems.
Model Refinement	Supervised Learning (Fine-Tuning)	Periodically retrains AI models using analyst-verified alerts (True/False Positives) to continuously improve detection accuracy and adapt to new threats.

Work So Far

- Finalized and validated the complete AURA system architecture.
- Built a functional intelligence prototype (osint-harvester) that gathers and analyzes live threat data.
- Implemented automated Indicators of Compromise (IOCs) extraction and exploit-to-CVE correlation in the prototype.
- Deployed a foundational static honeypot, establishing the core infrastructure for active deception.

Works To Be Done

- Connect the threat intelligence module to the honeypot to enable dynamic decoy generation.
- Develop the final module (Layer 4) to execute active countermeasures based on detected threats.
- Build the graph database to store and link threat intelligence for advanced analysis.
- Create the user interface for real-time system monitoring and visualization.
- Integrate all modules and perform end-to-end testing with simulated attacks.

Conclusion

Summary

- **Recap:** AURA is an innovative framework designed to address the critical need for proactive cybersecurity through its unique integration of intelligence, deception, and response.
- **Feasibility:** The project is highly feasible, leveraging well-established technologies in a novel architecture. Our phased work plan is on track.

References

1. M. Masunda, "Adaptive Threat Intelligence Systems for Real-Time Detection and Mitigation of Sophisticated Cyber Attacks in Enterprise Networks," *International Journal of Advance Research Publication and Reviews*, vol. 02, no. 05, pp. 470-491, May 2025.
2. T. O. Browne, M. Abedin, and M. J. M. Chowdhury, "A systematic review on research utilising artificial intelligence for open source intelligence (OSINT) applications," *Int. J. Inf. Secur.*, vol. 23, pp. 2911–2938, 2024.
3. P. Beltrán López, M. Gil Pérez, and P. Nespoli, "Cyber Deception: State of the art, Trends, and Open challenges," *arXiv preprint arXiv:2409.07194*, 2024.
4. S. A. Kareem, R. C. Sachan, and R. K. Malviya, "AI-Driven Adaptive Honeypots for Dynamic Cyber Threats," 2024.
5. Š. Grigaliūnas, et al., "Navigating the CISO's Mind by Integrating GenAI for Strategic Cyber Resilience," *Electronics*, vol. 14, no. 7, p. 1342, Mar. 2025.
6. S. Sreelakshmi, et al., "Enhancing Intrusion Detection Systems with Machine Learning," in *2024 2nd Int. Conf. Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 2024.

References

7. F. M. Anis and M. Hammoudeh, "AI-Powered Offensive Security: Automating Cyber Attacks with Large Language Models, Reinforcement Learning, and Generative Adversarial Networks," *Preprint*, May 2025.
8. M. Aminu, et al., "Enhancing Cyber Threat Detection through Real-time Threat Intelligence and Adaptive Defense Mechanisms," *Int. J. Comput. Appl. Technol. Res.*, vol. 13, no. 08, pp. 11-27, 2024.
9. K. I. Iyer, "Adaptive honeypots: Dynamic deception tactics in modern cyber defense," *Int. J. Sci. Res. Arch.*, vol. 04, no. 01, pp. 340–351, Dec. 2021.
10. J. Heluany, A. Amro, V. Gkioulos, and S. Katsikas, "Interplay of Digital Twins and Cyber Deception," in *2024 IEEE/ACM EnCyCriS Workshop*, Apr. 2024.
11. S. Shafee, A. Bessani, and P. M. Ferreira, "Evaluation of LLM-based chatbots for OSINT-based Cyber Threat Awareness," *Expert Syst. Appl.*, vol. 261, p. 125509, 2025.

12 S. Wali, Y. A. Farrukh, I. Khan, and N. D. Bastian, "Meta: Toward a Unified, Multimodal Dataset for Network Intrusion Detection Systems," *IEEE Data Descriptions*, vol. 1, 2024.

Thank You