International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Adaptive Deception OS: An Operating System That Fakes Itself When Attacked

Priya V*, Uma Maheshwari S**
*(Student, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email:priyav.ug22.cs@francisxavier.ac.in)
** (Assistant Professor, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
Email: uma@francisxavier.ac.in)

----------**********----------

Abstract:

Because cybersecurity threats are constantly evolving, conventional defensive mechanisms are often inadequate against sophisticated and adaptive attacks. This paper presents Adaptive Deception OS, a virtualization-based cybersecurity simulation framework designed to protect a Windows host system by dynamically redirecting suspicious activity into an isolated Ubuntu-based decoy environment. The system continuously monitors for malicious indicators such as unauthorized command execution and suspicious file downloads. Upon detection, it automatically activates a deception protocol that simulates host lockdown, launches a virtual machine, and presents the attacker with a realistic but fabricated operating system environment containing synthetic sensitive files. This controlled redirection ensures that the original Windows host remains fully intact and uncompromised. The proposed implementation relies on event-driven behavioral monitoring and VirtualBox-based isolation without modifying the Windows kernel or executing real malicious payloads. Experimental evaluation demonstrates that virtualization-driven adaptive deception can effectively simulate real-time defensive redirection while preserving host integrity and maintaining ethical research boundaries. The results validate the feasibility of operating system–level self-faking mechanisms as a secure academic prototype for studying proactive containment strategies in cybersecurity environments.

Keywords — **Adaptive Deception, Virtualization Security, Intrusion Detection, Host Protection, Decoy Environment**

----------**********----------

I. INTRODUCTION

Over the past decade, cybersecurity threats have evolved significantly, with attackers employing advanced tactics such as ransomware, zero-day exploits, and social engineering. Traditional defenses like signature-based firewalls and antivirus software often rely on reactive responses and known threat patterns, allowing modern attacks to bypass them and cause system compromise and data loss.To address these challenges, researchers have explored proactive defense strategies, including deception technology. This approach misleads attackers by presenting fake systems, data, or vulnerabilities, redirecting malicious activity into controlled environments. This not only protects the main system but also collects valuable intelligence on attack methods.Virtualization has further enhanced deception-based defenses by enabling isolated environments—such as virtual machines and sandboxes—to safely emulate vulnerable systems and contain threats without risking the host operating system. However, most current implementations focus on network-level honeypots rather than host-level adaptive deception.This study introduces Adaptive Deception OS, a virtualization-based cybersecurity simulation that provides automated host protection through controlled deception. The

system monitors for suspicious behavior— like unauthorized downloads or malicious commands— and responds by simulating a system shutdown while redirecting the attacker to a separate Ubuntu-based decoy environment. This approach creates a realistic but fake operating system with fictitious sensitive files, successfully mimicking a breach without exposing real host data.Using VirtualBox automation and event-driven monitoring, the system triggers a deception protocol without relying on kernel changes or actual malware execution. The research demonstrates that automated virtualization based redirection is a viable host-level defense strategy, contributing a practical and ethical model for operating system deception in academic and research settings. It also lays the groundwork for future enhancements, including intelligent threat classification and real-time adaptive responses, supporting a shift from reactive to proactive security models.

II. OBJECTIVE

The primary objective of the Adaptive Deception OS project is to design and develop an intelligent cybersecurity system that enhances system protection through proactive defense mechanisms based on deception technology. Unlike traditional security approaches that rely on detection and prevention, this system aims to mislead attackers, monitor their activities, and safely analyze potential threats within a controlled environment.

One of the key objectives is to create a realistic and interactive deception environment that simulates a vulnerable operating system. By presenting attackers with a fake Linux terminal and a decoy Ubuntu-based environment, the system aims to divert malicious activities away from the actual host system, thereby preventing real damage and data loss.

Another important objective is to implement real-time monitoring of system activities such as command execution, USB device insertion, and file downloads. The system is designed to identify suspicious or malicious behavior using predefined patterns and heuristic analysis, enabling immediate response through alert generation and automated redirection to the honeypot environment.

The project also focuses on developing a centralized administrative dashboard that provides comprehensive visibility into all detected threats, attack logs, and system activities. This allows administrators to analyze attacker behavior, understand emerging threat patterns, and improve overall security strategies.

Additionally, the system aims to ensure safe and ethical experimentation by utilizing virtualization techniques, such as VirtualBox integration, to isolate the deception environment. This ensures that no real system resources are compromised while still providing a highly realistic simulation for attackers.

Finally, the project seeks to contribute to the advancement of cybersecurity by demonstrating how adaptive, deception-based strategies can transform traditional reactive defense models into proactive and intelligent security solutions.

In addition to these goals, the project aims to improve the adaptability and scalability of cybersecurity systems by integrating modular components that can evolve with emerging threats. By combining deception techniques with real-time monitoring and automated response mechanisms, the system provides a flexible framework that can be extended with advanced features such as AI-based threat detection and behavioral analysis. This ensures that the Adaptive Deception OS remains effective in handling dynamic attack patterns while supporting continuous improvement in modern cybersecurity practices.

III. METHODOLOGY

The overall processing pipeline integrates data acquisition, linguistic analysis, spatial modeling, and predictive intelligence into a unified framework for real-time disaster response. Information flows sequentially from social media collection and preprocessing to crisis understanding, hotspot identification, and graph-based demand estimation, ultimately supporting

automated alerts and visualization for decision-makers. This end-to-end movement from raw, unstructured inputs to actionable operational insights is summarized in Fig. 1, which presents the complete system architecture of the proposed system.

A. ***System Architecture:*** The suggested Adaptive Deception OS's operating workflow is shown in the system architecture shown in Fig. 1. A Behavior Monitoring Engine regularly examines system activity for questionable activity at the Windows Host System, where the process starts. The system continues to monitor normally if no anomaly is found. Nevertheless, the Deception Protocol is immediately triggered as soon as questionable activity is detected. At this point, two protective measures work simultaneously: VBoxManage launches an Ubuntu virtual machine, and a host lockdown overlay is shown to mimic system resource protection. A fake file system that imitates authentic host data while staying completely separate from the real Windows operating system is present in the Ubuntu Decoy Environment. This keeps the real host data safe and undamaged by redirecting all further attacker interactions to the fake environment. A virtualization-driven adaptive deception system appropriate for scholarly and research applications is demonstrated by this design. Additionally, the architecture is designed with a modular structure, allowing each component to operate independently while maintaining seamless integration across the system. This improves scalability and enables future enhancements such as AI-based threat classification and automated response optimization. The use of event-driven triggers ensures that system resources are efficiently utilized, activating defensive mechanisms only when necessary. Furthermore, comprehensive logging and monitoring provide valuable forensic data for analyzing attacker behavior and improving security policies. Overall, this architecture enhances system resilience by combining real-time detection, controlled deception, and secure isolation within a unified framework.

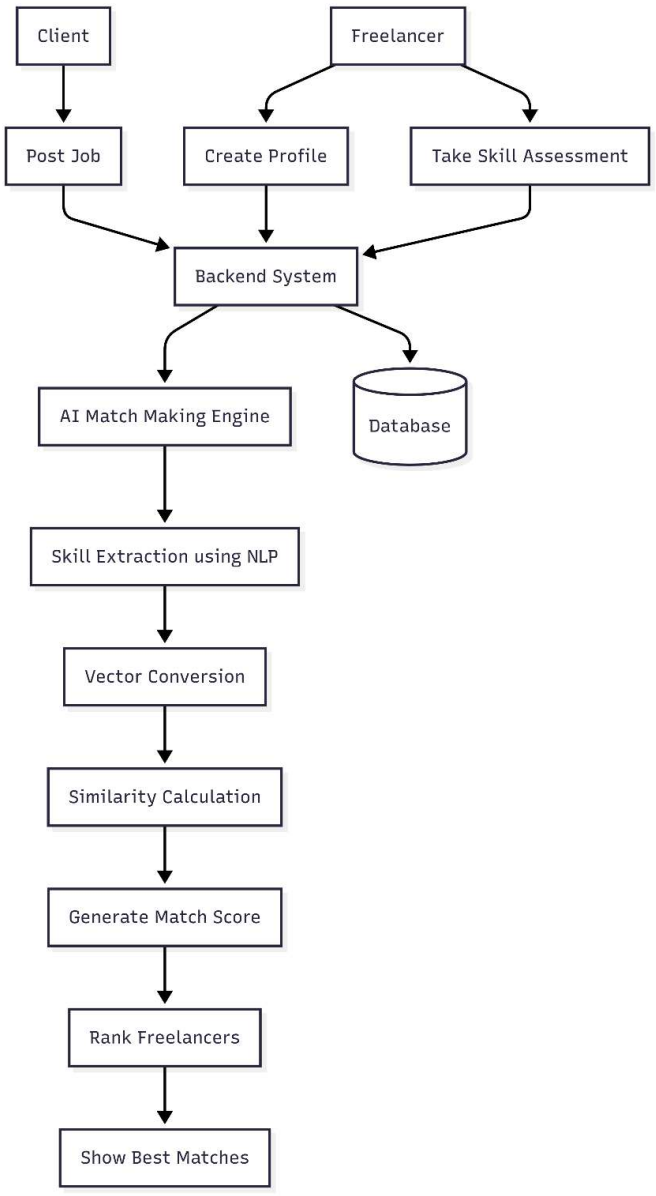

Fig. 1*. Block diagram of the proposed system*

B. ***Detection Engine:*** The Detection Engine serves as the primary defensive component of the Adaptive Deception OS, continuously monitoring the host system for suspicious activity using behavior-based analysis. Instead of relying on predefined malware signatures, it observes system-level indicators such as unauthorized file access, abnormal command execution, unusual process

behavior, and interactions with designated honeypot resources.

A background monitoring script analyzes event logs and activity records in real time, flagging potential threats when predefined behavioral thresholds—such as privilege escalation attempts or access to protected directories—are exceeded.Designed as a non-intrusive and event-driven architecture, the Detection Engine operates with user-level permissions and does not modify kernel components or intercept live network traffic. Upon confirming malicious behavior, it activates the Deception Protocol, triggering the lockdown overlay and virtualization-based redirection. Additionally, structured logging ensures forensic traceability and controlled trigger activation, allowing the system to distinguish between normal activity and coordinated intrusion attempts while maintaining strict operational boundaries.

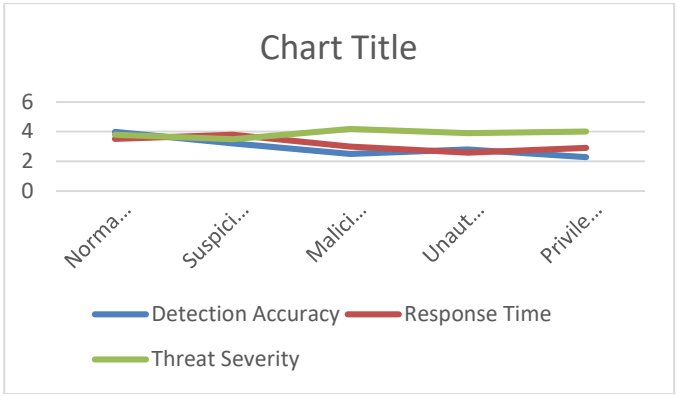

Fig 2: Detection Engine – Threat Analysis

C. *Deception Activation Layer*: The Deception Activation Layer functions as the response coordinator within the Adaptive Deception OS architecture. When the Detection Engine identifies suspicious or malicious activity, control is transferred to this layer to initiate a defensive transition. Instead of terminating the activity, the system shifts from a monitoring state to a deception state by updating the system status to UNDER_ATTACK and activating a simulated host lockdown overlay. This visual response indicates system protection while maintaining operational stability. Simultaneously, the layer logs the intrusion event and executes a VirtualBox command to launch the Ubuntu-based decoy environment. Designed as a modular and event-driven component, it activates only upon verified threats and does not modify kernel components or intercept live system calls. By bridging detection and virtualization-based redirection, the Deception Activation Layer ensures controlled containment and safely redirects attacker interaction into an isolated decoy environment.

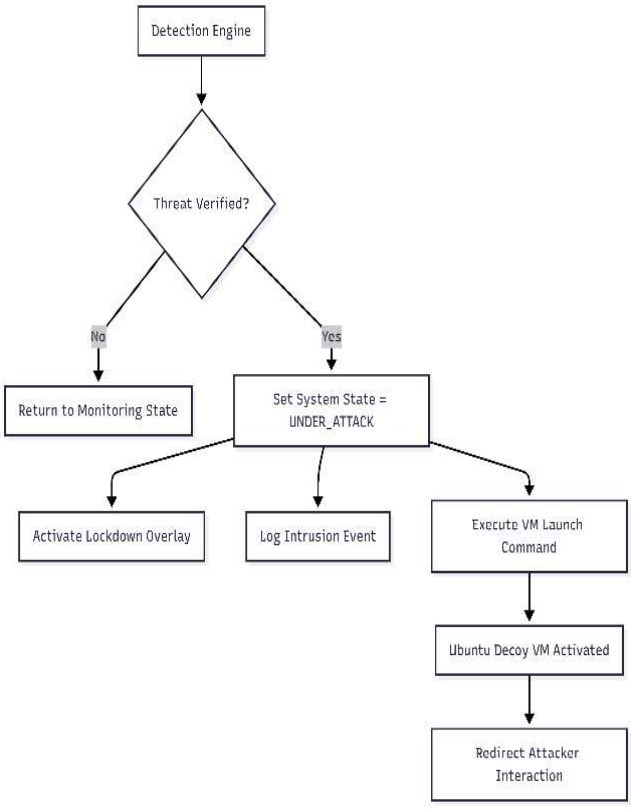

Fig.3. *Block diagram of the Deception Activation Layer*

D. *Virtualization-Based Redirection:* The Virtualization-Based Redirection module serves as the primary containment mechanism of the Adaptive Deception OS. Once malicious activity is confirmed, the system redirects attacker interaction into a sandboxed Ubuntu virtual machine instead of terminating processes or shutting down the host. This redirection is performed using VirtualBox automation, ensuring that suspicious activity is confined

within an isolated memory, CPU, and file system environment.Unlike traditional sandboxing approaches, this framework creates the illusion of a successful system compromise. The decoy virtual machine contains a fabricated Windows-like directory structure and synthetic sensitive files, while the real host remains unaffected. The process is automatic, event-driven, and does not involve kernel modification or system call interception. By separating detection from interaction, virtualization-based redirection ensures secure containment while preserving host integrity and enabling safe demonstration of adaptive deception principles.

E. Decoy Environment and Fake File Simulation: The Decoy Environment represents the final stage of the Adaptive Deception OS workflow. After redirection, the attacker is moved to an isolated Ubuntu virtual machine designed to mimic a compromised Windows system. This environment presents a fabricated directory structure and realistic-looking files while remaining completely separated from the actual host operating system.The decoy includes synthetic files such as credentials, financial documents, and project data that appear sensitive but contain no real information. Simulated command-line behavior and directory responses enhance realism, preventing immediate detection of deception. Strict isolation ensures no shared memory or direct file system linkage with the Windows host. By combining realistic simulation with virtualization-based containment, this module safely completes the deception cycle while protecting genuine system resources.

IV. RESULTS AND DISCUSSION

The Adaptive Deception OS prototype was experimentally assessed to evaluate its effectiveness in detecting anomalous host activities and automatically transitioning into a controlled deception state. The evaluation concentrated on three major dimensions: responsiveness of behavioral detection, reliability of virtualization-driven redirection, and preservation of host system integrity. Additionally, the synchronization between detection, activation, and decoy deployment was examined to determine overall operational stability.

1. CyberCore Landing Page

This interface demonstrates a user-friendly entry point for interacting with the system. It effectively communicates system capabilities while providing quick access to core modules. The integration of real-time metrics ensures immediate visibility into system status, improving usability and administrative control.

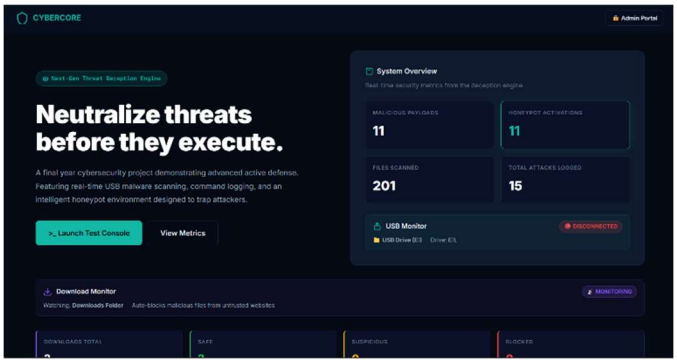

Fig.4. *CyberCore Landing Page – Adaptive Deception Engine Interface*

2. Security Dashboard

The dashboard confirms the system's ability to aggregate and present security data in a centralized manner. Metrics such as detected threats and scanned files validate the effectiveness of continuous monitoring. This visualization enhances situational awareness and supports faster decision-making.

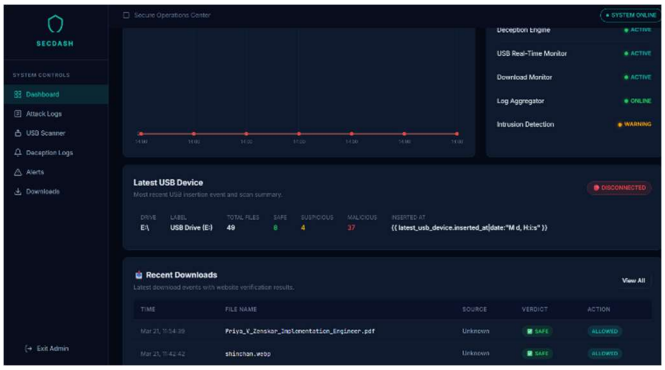

Fig. 5. Security Dashboard – System Monitoring Overview

3. USB Monitoring and Threat Detection Result

The results show that the system successfully detects and categorizes files from external devices. A higher number of malicious files indicates the system's capability to identify threats accurately. This highlights the importance of USB scanning in preventing external malware entry.

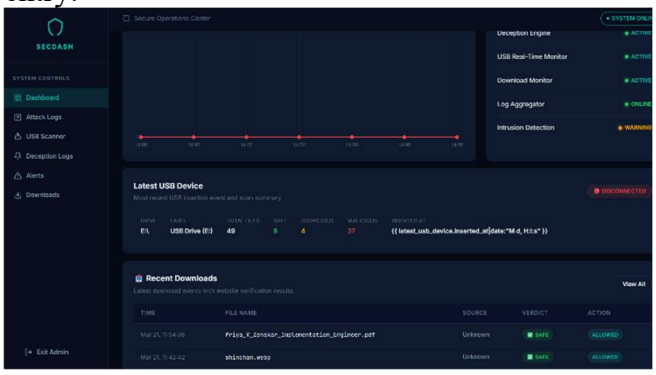

FIG. 6. USB MONITORING AND THREAT DETECTION RESULT

4. Honeypot Activation – Suspicious Activity Detection

The system successfully identifies malicious behavior and activates the deception protocol. The visual alert combined with automatic redirection ensures that attackers are isolated without affecting the host system. This confirms the effectiveness of the deception-based defense strategy.

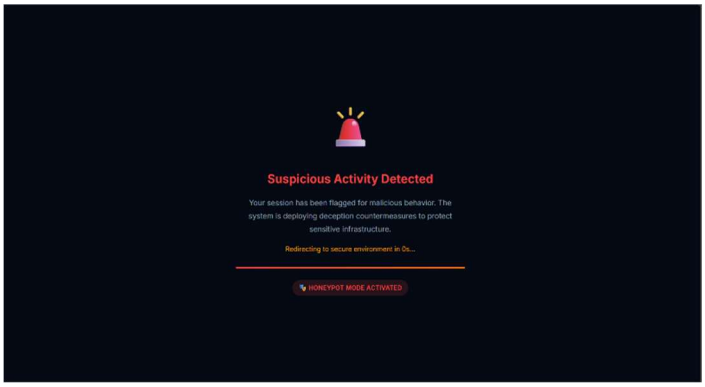

Fig. 7. Honeypot Activation – Suspicious Activity Detection

5. Fake OS Interface

This environment demonstrates the system's ability to mislead attackers by providing a convincing fake OS. Since all interactions occur within an isolated environment, real system data remains secure. This validates the core concept of adaptive deception and safe attack containment.

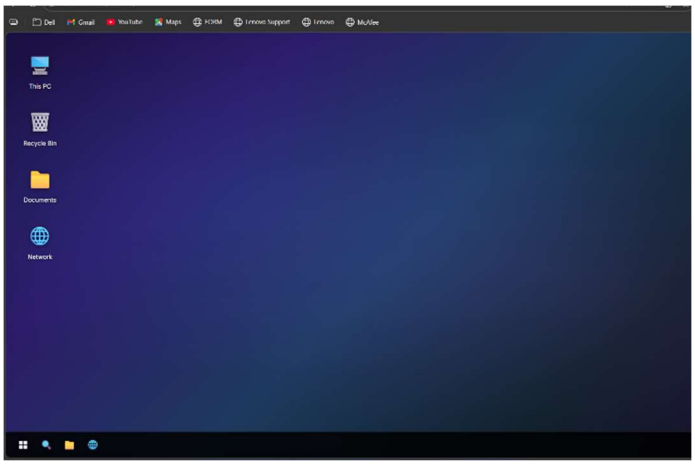

Fig. 7: Fake OS (Decoy Environment) Interface

V. REFERENCES

[1] M. ALMESHEKAH AND E. SPAFFORD, "TOWARD A THEORY OF CYBER DECEPTION," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 15, PP. 1316–1330, 2020.

[2] S. WANG, N. ZHANG AND R. BEYAH, "BEHAVIOR-BASED MALWARE DETECTION IN VIRTUALIZED SYSTEMS," IN IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 18, NO. 4, PP. 1719-1732, JULY-AUG. 2020.

[3] X. ZHANG, Y. LI AND J. SUN, "A SANDBOX-BASED MALWARE CONTAINMENT FRAMEWORK WITH REAL-TIME ANALYSIS," IEEE ACCESS, VOL. 8, PP. 134567-134580, 2020.

[4] F. JAJODIA, P. AMMANN, AND V. SWARUP, "MOVING TARGET DEFENSE: PRINCIPLES AND DESIGN," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 22, NO. 2, PP. 1166-1181, SECONDQUARTER 2020.

[5] H. CHEN, M. WANG, AND L. XIONG, "SECURE DECEPTION FOR CLOUD INFRASTRUCTURE: A COMPREHENSIVE FRAMEWORK," IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 9, NO. 1, PP. 117-130, JAN.–MAR. 2021.

[6] R. KUMAR AND R. SINGH, "ADAPTIVE INTRUSION RESPONSE FOR HOST LEVEL SECURITY," IN PROC. IEEE INTERNATIONAL CONF. ON CYBER SECURITY AND RESILIENCE, PP. 45-52, 2021.

[7] J. LIU, T. TAN, AND D. CHEN, "AUTOMATED DECEPTION ORCHESTRATION FOR CYBER DEFENSE," IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 18, NO. 3, PP. 3255-3267, SEPT. 2021.

[8] A. S. BALAKRISHNAN AND M. S. ISLAM, "VM ISOLATION AND SANDBOXING TECHNIQUES FOR SECURE HOST ENVIRONMENTS," IEEE ACCESS, VOL. 9, PP. 24533-24546, 2021.

[9] Z. YANG, J. LIU AND K. ZHAO, "HOST-LEVEL ANOMALY DETECTION USING BEHAVIORAL MODELING," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 17, PP. 1301-1313, 2022.

[10] G. T. NGUYEN AND S. KIM, "HYBRID HONEYPOT SYSTEMS FOR ADVANCED THREAT INTELLIGENCE," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 24, NO. 1, PP. 476-501, 2022.

[11] P. R. KUMAR, H. S. KAUR AND A. SHARMA, "VIRTUALIZATION BASED SECURITY TECHNIQUES FOR NEXT-GENERATION OPERATING SYSTEMS," IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSICS, VOL. 8, NO. 4, PP. 152-162, 2023.

[12] J. TAN, X. LIU AND B. WANG, "DECEPTION-DRIVEN DEFENSE IN CYBER PHYSICAL SYSTEMS: A SURVEY," IEEE ACCESS, VOL. 12, PP. 90857-90872, 2024.

[13] H. Q. TRAN AND B. LEE, "REAL-TIME BEHAVIORAL MONITORING AND RESPONSE MECHANISMS FOR HOST PROTECTION," IEEE SYSTEMS JOURNAL, VOL. 19, NO. 2, PP. 1520-1531, 2025.

